PLEXUS

Sys5 UNIX Administrator's Reference Manual

98-05084.1 Ver. D November, 1986

Sys5 UNIX Administrator's Reference Manual

98-05084.1 Ver. D November, 1986

PLEXUS COMPUTERS, INC.
3833 North First Street
San Jose, CA 95134
408/943-9433

The information contained herein is subject to change without notice. Therefore, Plexus Computers, Inc. assumes no responsibility for the accuracy of the information presented in this document beyond its current release date.

Printed in the United States of America

1. INTRODUCTION

This manual supplements the information in the *UNIX Sys5 User's Reference Manual* and provides an easy reference volume for those who must administer a *UNIX* system. Accordingly, only those commands and descriptions deemed appropriate for system administrators have been included here.

This manual is divided into three sections:

- 1M. System Maintenance Commands and Application Programs
- 7. Special Files
- 8. System Maintenance Programs and Stand-alone Procedures

Throughout this volume, each reference of the form *name(1M), name(7),* or *name(8)*, refers to entries in this manual, while all other references to entries of the form *name(N)*, where *N* is a number possibly followed by a letter, refer to entry *name* in Section *N* of the *UNIX Sys5 Programmer's Reference Manual* or the *UNIX Sys5 User's Reference Manual.*

Section 1M *(System Maintenance Commands and Application Programs)* contains system maintenance programs such as *fsck, mkfs,* etc., which generally reside in the directory **/etc**; these entries carry a sub-section designation of *1M* for cross-referencing reasons.

Section 7 *(Special Files)* discusses the characteristics of each system file that actually refers to an input/output device. The names in this section generally refer to device names for the hardware, rather than to the names of the special files themselves.

Section 8 *(System Maintenance Programs)* discusses crash recovery, stand-alone procedures, facility descriptions, etc.

Each section consists of a number of independent entries of a page or so each. The name of the entry appears in the upper corners of its pages. Entries within each section are alphabetized, except for the introductory entry that begins each section. Some entries describe several routines, commands, etc., and in such cases, the entry appears only once, under its *major* name.

All entries have a common format, not all of whose parts always appear:

NAME gives the name(s) of the entry and briefly states its purpose.

SYNOPSIS summarizes the program being described. A few conventions are used, particularly in Section 1 (*Commands*):

Boldface strings are literals and are to be typed just as they appear.

Italic strings usually represent substitutable prototypes and program names found elsewhere in the manual. (They are underlined in the the typed versions of the entries.)

Square brackets (**[ ]**) around an argument prototype indicate that the argument is optional. When an argument prototype is given as *name* or *file*, it always refers to a *file* name.

Ellipses (**...**) are used to show that the previous argument prototype might be repeated.

A final convention is used by itself. An argument beginning with a minus (–), plus (+), or equal sign (=) is often a flag argument, even if it appears in a position where a file name could appear. Therefore, it is unwise to have files whose names begin with –, +, or =.

DESCRIPTION discusses the subject at hand.

FILES gives the file names that are built into the program.

SEE ALSO gives pointers to related information.

DIAGNOSTICS discusses the diagnostic indications that might be produced. Self-explanatory messages are not listed.

WARNINGS points out potential pitfalls.

BUGS gives known bugs, and sometimes, deficiencies. Occasionally the suggested fix is also described.

A table of contents precedes the first section. On most systems, all entries are available on-line via the *man(1)* command.

CONTENTS

1. COMMANDS AND APPLICATION PROGRAMS

1. COMMANDS AND APPLICATION PROGRAMS

intro................................introduction to commands and application programs
300..............................handle special functions of DASI 300 and 300s terminals
4014..paginator for the TEKTRONIX 4014 terminal
450..handle special functions of the DASI 450 terminal
acctcom..search and print process accounting file(s)
adb..absolute debugger
admin...create and administer SCCS files
ar......................................archive and library maintainer for portable archives
arcv.....................convert archive files from PDP-11 to common archive format
as...common assembler
asa..interpret ASA carriage control characters
at..execute commands at a later time
awk..pattern scanning and processing language
banner..make posters
bar...Berkeley archive and library maintainer
basename...deliver portions of path names
bbanner..print large banner on printer
bc...arbitrary-precision arithmetic language
bdiff...big diff
bfs...big file scanner
bls...list contents of directory
bs...a compiler/interpreter for modest-sized programs
cal...print calendar
calendar...reminder service
cat..concatenate and print files
cb..C program beautifier
cc..C compiler
cd...change working directory
cdc..change the delta commentary of an SCCS delta
cflow..generate C flow graph
chmod...change mode
chown...change owner or group
clear..clear terminal screen
cmp...compare two files
col..filter reverse line-feeds
comb...combine SCCS deltas
comm.......................................select or reject lines common to two sorted files
cp...copy, link or move files
cpio..copy file archives in and out
cpp...the C language preprocessor
crontab...user crontab file
crypt...encode/decode
csh......................................a shell (command interpreter) with C-like syntax
csplit...context split
ct...spawn getty to a remote terminal
ct...spawn getty to a remote terminal
ctags...create a tags file
ctrace...C program debugger

cu .... call another UNIX system
cu .... call another UNIX system
cut .... cut out selected fields of each line of a file
cxref .... generate C program cross-reference
date .... print and set the date
dc .... desk calculator
dd .... convert and copy a file
delta .... make a delta (change) to an SCCS file
deroff .... remove nroff/troff, tbl, and eqn constructs
dial .... dial a Racal-Vadic 3451 modem
diff .... differential file comparator
diff3 .... 3-way differential file comparison
diffmk .... mark differences between files
dircmp .... directory comparison
du .... summarize disk usage
dump .... dump selected parts of an object file
dx9700 .... prepare troff documents for the Xerox 9700 printer
echo .... echo arguments
ed .... text editor
edit .... text editor (variant of ex for casual users)
efl .... Extended Fortran Language
enable .... enable/disable LP printers
env .... set environment for command execution
eqn .... format mathematical text for nroff or troff
ex .... text editor
expr .... evaluate arguments as an expression
f77 .... Fortran 77 compiler
factor .... factor a number
file .... determine file type
find .... find files
fsplit .... split f77, ratfor, or efl files
gdev .... graphical device routines and filters
gdev .... graphical device routines and filters
ged .... graphical editor
ged .... graphical editor
get .... get a version of an SCCS file
getopt .... parse command options
graph .... draw a graph
graph .... draw a graph
graphics .... access graphical and numerical commands
graphics .... access graphical and numerical commands
greek .... select terminal filter
grep .... search a file for a pattern
gutil .... graphical utilities
gutil .... graphical utilities
head .... give first few lines of a stream
help .... ask for help
hp .... handle special functions of HP 2640 and 2621-series terminals
hyphen .... find hyphenated words
id .... print user and group IDs and names
ipcrm .... remove a message queue, semaphore set or shared memory id
ipcs .... report inter-process communication facilities status
join .... relational database operator

kill .... terminate a process
ld .... link editor for common object files
lex .... generate programs for simple lexical tasks
line .... read one line
lint .... a C program checker
login .... sign on
logname .... get login name
lorder .... find ordering relation for an object library
lp .... send/cancel requests to an LP line printer
lphold .... postpone printing, resume printing
lpstat .... print LP status information
ls .... list contents of directory
m4 .... macro processor
macref .... produce cross-reference listing of macro files
mail .... send mail to users or read mail
mailx .... interactive message processing system
make .... maintain, update, and regenerate groups of programs
makekey .... generate encryption key
man .... print entries in this manual
mesg .... permit or deny messages
mkdir .... make a directory
mkstr .... create an error message file by massaging C source
mm .... print/check documents formatted with the MM macros
mmlint .... sroff/MM nroff/MM document compatibility checker
mmt .... typeset documents, viewgraphs, and slides
more .... file perusal filter for crt viewing
newform .... change the format of a text file
newgrp .... log in to a new group
news .... print news items
nice .... run a command at low priority
nl .... line numbering filter
nm .... print name list of common object file
nohup .... run a command immune to hangups and quits
nroff .... format or typeset text
ocw .... prepare constant-width text for otroff
od .... octal dump
pack .... compress and expand files
passwd .... change login password
paste .... merge same lines of several files or subsequent lines of one file
pg .... file perusal filter for soft-copy terminals
pic .... troff preprocessor for drawing simple pictures
pr .... print files
printenv .... print out the environment
prof .... display profile data
prs .... print an SCCS file
ps .... report process status
ptx .... permuted index
pwd .... working directory name
ratfor .... rational Fortran dialect
regcmp .... regular expression compile
rm .... remove files or directories
rmdel .... remove a delta from an SCCS file
sact .... print current SCCS file editing activity

sag .... system activity graph
sag .... system activity graph
sar .... system activity reporter
scc .... C compiler for stand-alone programs
sccsdiff .... compare two versions of an SCCS file
script .... make typescript of terminal session
sdiff .... side-by-side difference program
sed .... stream editor
sh .... shell, the standard/restricted command programming language
size .... print section sizes of common object files
sleep .... suspend execution for an interval
sno .... SNOBOL interpreter
sort .... sort and/or merge files
spell .... find spelling errors
spline .... interpolate smooth curve
spline .... interpolate smooth curve
split .... split a file into pieces
sroff .... format text
stat .... statistical network useful with graphical commands
stat .... statistical network useful with graphical commands
strings .... find the printable strings in a object, or other binary, file
strip .... strip symbol and line number information from common object file
stty .... set the options for a terminal
style .... analyze surface characteristics of a document
su .... become super-user or another user
sum .... print checksum and block count of a file
sync .... update the super block
tabs .... set tabs on a terminal
tail .... deliver the last part of a file
tape .... tape manipulation
tar .... tape file archiver
tbl .... format tables for nroff or troff
tc .... troff output interpreter
tee .... pipe fitting
test .... condition evaluation command
time .... time a command
timex .... time a command; report process data and system activity
toc .... graphical table of contents routines
toc .... graphical table of contents routines
touch .... update access and modification times of a file
tplot .... graphics filters
tplot .... graphics filters
tput .... query terminfo database
tr .... translate characters
troff .... text formatting and typesetting
true .... provide truth values
tset .... set terminal modes
tsort .... topological sort
tty .... get the name of the terminal
umask .... set file-creation mode mask
uname .... print name of current UNIX system
unget .... undo a previous get of an SCCS file
uniq .... report repeated lines in a file

units....conversion program
uucp....UNIX system to UNIX system copy
uucp....UNIX system to UNIX system copy
uuencode....encode/decode a binary file for transmission via mail
uuencode....encode/decode a binary file for transmission via mail
uustat....uucp status inquiry and job control
uustat....uucp status inquiry and job control
uuto....public UNIX-to-UNIX system file copy
uuto....public UNIX-to-UNIX system file copy
uux....UNIX-to-UNIX system command execution
uux....UNIX-to-UNIX system command execution
val....validate SCCS file
vc....version control
vi....screen-oriented (visual) display editor based on ex
vtty....connect to a remote host via NOS
wait....await completion of process
wc....word count
what....identify SCCS files
who....who is on the system
write....write to another user
x9700....prepare nroff documents for the Xerox 9700 printer
xargs....construct argument list(s) and execute command
xstr....extract strings from C programs to implement shared strings
yacc....yet another compiler-compiler

1M. SYSTEM MAINTENANCE COMMANDS AND PROGRAMS

intro....system maintenance commands and application programs
accept....allow/prevent LP requests
acct....overview of accounting and miscellaneous accounting commands
acctcms....command summary from per-process accounting records
acctcon....connect-time accounting
acctmerg....merge or add total accounting files
acctprc....process accounting
acctsh....shell procedures for accounting
acpdmp....dump contents of Advanced Communication
brc....system initialization shell scripts
checkall....faster file system checking procedure
chroot....change root directory for a command
clri....clear i-node
copytape....make an image copy of a tape
cpset....install object files in binary directories
crash....examine system images
cron....clock daemon
dconfig....configure logical disks
dcopy....copy file systems for optimal access time
devnm....device name
df....report number of free disk blocks
diskusg....generate disk accounting data by user ID
dnld....download program files
dump....incremental file system dump
dumpdir....print the names of files on a dump tape
errdead....extract error records from dump

errdemon .... error-logging daemon
errpt .... process a report of logged errors
errstop .... terminate the error-logging daemon
fbackup .... make a fast tape backup of a file system
ff .... list file names and statistics for a file system
filesave .... daily/weekly UNIX system file system backup
finc .... fast incremental backup
frec .... recover files from a backup tape
fsck .... file system consistency check and interactive repair
fsdb .... file system debugger
fuser .... identify processes using a file or file structure
fwtmp .... manipulate connect accounting records
getty .... set terminal type, modes, speed, and line discipline
icpdmp .... dump contents of an Intelligent Communication
init .... process control initialization
install .... install commands
killall .... kill all active processes
link .... exercise link and unlink system calls
lpadmin .... configure the LP spooling system
lpsched .... start/stop the LP request scheduler and move requests
mirutil .... utility for connecting two identical
mkfs .... construct a file system
mknod .... build special file
mount .... mount and dismount file system
mvdir .... move a directory
ncheck .... generate names from i-numbers
non-btl .... reinstall MM macros without Bell Laboratories specific features
profiler .... operating system profiler
pwck .... password/group file checkers
ramdisk .... memory as disk
restor .... incremental file system restore
runacct .... run daily accounting
sadp .... disk access profiler
sar .... system activity report package
setmnt .... establish mount table
shutdown .... terminate all processing
sys .... System control and status program.
tic .... terminfo compiler
topq .... prioritize print queue
uucico .... file transport program for the uucp system
uuclean .... uucp spool directory clean-up
uusub .... monitor uucp network
uuxqt .... execute remote command requests
volcopy, labelit .... copy file systems with label checking
wall .... write to all users
whodo .... who is doing what

2. SYSTEM CALLS

2. SYSTEM CALLS

intro .... introduction to system calls and error numbers
access .... determine accessibility of a file

acct .... enable or disable process accounting
alarm .... set a process alarm clock
brk .... change data segment space allocation
chdir .... change working directory
chmod .... change mode of file
chown .... change owner and group of a file
chroot .... change root directory
close .... close a file descriptor
creat .... create a new file or rewrite an existing one
dup .... duplicate an open file descriptor
exec .... execute a file
fcntl .... file control
fork .... create a new process
getpid .... get process, process group, and parent process IDs
getuid .... get real user, effective user, real group, and effective group IDs
ioctl .... control device
kill .... send a signal to a process or a group of processes
link .... link to a file
lseek .... move read/write file pointer
mknod .... make a directory, or a special or ordinary file
mount .... mount a file system
msgctl .... message control operations
msgget .... get message queue
msgop .... message operations
nice .... change priority of a process
open .... open for reading or writing
pause .... suspend process until signal
pipe .... create an interprocess channel
plock .... lock process, text, or data in memory
profil .... execution time profile
ptrace .... process trace
read .... read from file
semctl .... semaphore control operations
semget .... get set of semaphores
semop .... semaphore operations
setpgrp .... set process group ID
setuid, setgid .... set user and group IDs
shmctl .... shared memory control operations
shmget .... get shared memory segment
shmop .... shared memory operations
signal .... specify what to do upon receipt of a signal
stat, fstat .... get file status
stime .... set time
sync .... update super-block
time .... get time
times .... get process and child process times
ulimit .... get and set user limits
umask .... set and get file creation mask
umount .... unmount a file system
uname .... get name of current UNIX system
unlink .... remove directory entry
ustat .... get file system statistics
utime .... set file access and modification times

wait .... wait for child process to stop or terminate
write .... write on a file

2S. STANDALONE SYSTEM CALLS

intro .... introduction to standalone system calls,
access .... determine accessibility of a file
brk .... change data segment space allocation
chdir .... change working directory
chmod .... change mode of file
close .... close a file descriptor
creat .... create a new special file
exit .... terminate process
float .... float and double routines
getargv .... display a program name and get arguments for
getpid .... get process ID
getuid .... get real user, effective user, real group, and effective group IDs
gtty .... get terminal characterisitcs
isatty .... returns a 1 if specified file descriptor is a terminal
kill .... send a signal to a process or a group of processes
lseek .... move read/write file pointer
mknod .... make a special file
mount .... mount a file system
nice .... change priority of a process
open .... open for reading or writing
read .... read from file
sleep .... suspend execution for interval
srcheof .... position to a specific file number on a tape
stat .... get file status
stime .... set time
stty .... set terminal characteristics
tell .... report the current value of a file pointer
time .... get time
umask .... set and get file creation mask
umount .... unmount a file system
ustat .... get file system statistics
write .... write on a file

3. SUBROUTINES

3C and 3S. C AND ASSEMBLER, STANDARD I/O LIBRARY ROUTINES

intro .... introduction to subroutines and libraries
a64l .... convert between long integer and base-64 ASCII string
abort .... generate an IOT fault
abs .... return integer absolute value
bsearch .... binary search a sorted table
clock .... report CPU time used
conv .... translate characters
crypt .... generate DES encryption
ctermid .... generate file name for terminal
ctime .... convert date and time to string
ctype .... classify characters

cuserid .... get character login name of the user
dial .... establish an out-going terminal line connection
drand48 .... generate uniformly distributed pseudo-random numbers
ecvt .... convert floating-point number to string
end .... last locations in program
fclose .... close or flush a stream
ferror .... stream status inquiries
fopen .... open a stream
fread .... binary input/output
frexp .... manipulate parts of floating-point numbers
fseek .... reposition a file pointer in a stream
ftw .... walk a file tree
getc .... get character or word from a stream
getcwd .... get path-name of current working directory
getenv .... return value for environment name
getgrent .... get group file entry
getlogin .... get login name
getopt .... get option letter from argument vector
getpass .... read a password
getpw .... get name from UID
getpwent .... get password file entry
gets .... get a string from a stream
getut .... access utmp file entry
hsearch .... manage hash search tables
l3tol .... convert between 3-byte integers and long integers
lsearch .... linear search and update
malloc .... main memory allocator
memory .... memory operations
mktemp .... make a unique file name
monitor .... prepare execution profile
nlist .... get entries from name list
perror .... system error messages
popen .... initiate pipe to/from a process
printf .... print formatted output
putc .... put character or word on a stream
putenv .... change or add value to environment
putpwent .... write password file entry
puts .... put a string on a stream
qsort .... quicker sort
rand .... simple random-number generator
scanf sscanf .... convert formatted input
setbuf .... assign buffering to a stream
setjmp .... non-local goto
sleep .... suspend execution for interval
ssignal .... software signals
stdio .... standard buffered input/output package
stdipc .... standard interprocess communication package
string .... string operations
strtod .... convert string to double-precision number
strtol .... convert string to integer
swab .... swap bytes
system .... issue a shell command
termlib .... terminal independent operation routines

tmpfile .... create a temporary file
tmpnam .... create a name for a temporary file
tsearch .... manage binary search trees
ttyname .... find name of a terminal
ttyslot .... find the slot in the utmp file of the current user
ungetc .... push character back into input stream
vprintf .... print formatted output of a varargs argument list

3M. MATHEMATICAL LIBRARY ROUTINES

bessel .... Bessel functions
erf .... error function and complementary error function
exp .... exponential, logarithm, power, square root functions
floor .... floor, ceiling, remainder, absolute value functions
gamma .... log gamma function
hypot .... Euclidean distance function
matherr .... error-handling function
sinh .... hyperbolic functions
trig .... trigonometric functions

3X. MISCELLANEOUS ROUTINES

assert .... verify program assertion
curses .... CRT screen handling and optimization package
ldahread .... read the archive header of a member of an archive file
ldclose, ldaclose .... close a common object file
ldfhread .... read the file header of a common object file
ldgetname ...... retrieve symbol name for common object file symbol table entry
ldlread .... manipulate line number entries of a common object file function
ldlseek .... seek to line number entries of a section of a common object file
ldohseek .... seek to the optional file header of a common object file
ldopen .... open a common object file for reading
ldrseek .... seek to relocation entries of a section of a common object file
ldshread .... read an indexed/named section header of a common object file
ldsseek .... seek to an indexed/named section of a common object file
ldtbindex ... compute the index of a symbol table entry of a common object file
ldtbread .... read an indexed symbol table entry of a common object file
ldtbseek .... seek to the symbol table of a common object file
logname .... return login name of user
malloc .... fast main memory allocator
plot .... graphics interface subroutines
regcmp .... compile and execute regular expression
sputl .... access long integer data in a machine-independent fashion

3F. FORTRAN ROUTINES

abort .... terminate Fortran program
abs .... Fortran absolute value
acos .... Fortran arccosine intrinsic function
aimag .... Fortran imaginary part of complex argument
aint .... Fortran integer part intrinsic function
asin .... Fortran arcsine intrinsic function
atan .... Fortran arctangent intrinsic function

atan2 .... Fortran arctangent intrinsic function
bool .... Fortran bitwise boolean functions
conjg .... Fortran complex conjugate intrinsic function
cos .... Fortran cosine intrinsic function
cosh .... Fortran hyperbolic cosine intrinsic function
dim .... positive difference intrinsic functions
dprod .... double precision product intrinsic function
exp .... Fortran exponential intrinsic function
ftype .... explicit Fortran type conversion
getarg .... return Fortran command-line argument
getenv .... return Fortran environment variable
iargc .... returns number of command line arguments passed to the program
index .... return location of Fortran substring
len .... return length of Fortran string
log .... Fortran natural logarithm intrinsic function
log10 .... Fortran common logarithm intrinsic function
max .... Fortran maximum-value functions
mclock .... return Fortran time accounting
min .... Fortran minimum-value functions
mod .... Fortran remaindering intrinsic functions
rand .... random number generator
round .... Fortran nearest integer functions
sign .... Fortran transfer-of-sign intrinsic function
signal .... specify Fortran action on receipt of a system signal
sin .... Fortran sine intrinsic function
sinh .... Fortran hyperbolic sine intrinsic function
sqrt .... Fortran square root intrinsic function
strcmp .... string comparison intrinsic functions
system .... issue a shell command from Fortran
tan .... Fortran tangent intrinsic function
tanh .... Fortran hyperbolic tangent intrinsic function

4. FILE FORMATS

intro .... introduction to file formats
L-devices .... link devices, connection information
L-dialcodes .... alphabetic dialing abbreviations file
L.cmds .... remote execution commands
L.sys .... link systems
USERFILE .... UUCP pathname permissions file
a.out .... common assembler and link editor output
acct .... per-process accounting file format
ar .... common archive file format
checklist .... list of file systems processed by fsck
core .... format of core image file
cpio .... format of cpio archive
dir .... format of directories
dump .... incremental dump tape format
errfile .... error-log file format
filehdr .... file header for common object files
fs .... format of system volume
fspec .... format specification in text files

gettydefs....................speed and terminal settings used by getty
gps....................graphical primitive string, format of graphical files
group....................group file
inittab....................script for the init process
inode....................format of an i-node
ioctl.syscon....................system console configuration file
issue....................issue identification file
ldfcn....................common object file access routines
linenum....................line number entries in a common object file
mnttab....................mounted file system table
passwd....................password file
plot....................graphics interface
profile....................setting up an environment at login time
reloc....................relocation information for a common object file
sccsfile....................format of SCCS file
scnhdr....................section header for a common object file
syms....................common object file symbol table format
term....................format of compiled term file.
termcap....................terminal capability data base
terminfo....................terminal capability data base
utmp....................utmp and wtmp entry formats

5. MISCELLANEOUS FACILITIES

intro....................introduction to miscellany
ascii....................map of ASCII character set
environ....................user environment
eqnchar....................special character definitions for eqn and neqn
fcntl....................file control options
font....................description files for device-independent troff
man....................macros for formatting entries in this manual
math....................math functions and constants
mm....................the MM macro package for formatting documents
mosd....................the OSDD adapter macro package for formatting documents
mptx....................the macro package for formatting a permuted index
mv....................a troff macro package for typesetting viewgraphs and slides
prof....................profile within a function
profile....................setting up an environment at login time
regexp....................regular expression compile and match routines
stat....................data returned by stat system call
term....................conventional names for terminals
troff....................description of output language
ttytype....................data base of terminal types by port
types....................primitive system data types
values....................machine-dependent values
varargs....................handle variable argument list

6. GAMES

intro....................introduction to games

arithmetic .... provide drill in number facts
back .... the game of backgammon
bj .... the game of black jack
craps .... the game of craps
hangman .... guess the word
maze .... generate a maze
moo .... guessing game
quiz .... test your knowledge
wump .... the game of hunt-the-wumpus

7. SPECIAL FILES

intro .... introduction to special files
err .... error-logging interface
ft .... IMSP streaming cartridge controller
icp .... Intelligent Communications Processor
mem .... core memory
mv .... a macro package for making view graphs
null .... the null file
pp .... parallel port interface
prf .... operating system profiler
pt .... IMSP cartridge controller
rm .... Cipher Microstreamer tape drive
rram .... allows memory to be used as a disk
tty .... general terminal interface

8. SYSTEM MAINTENANCE AND STANDALONE PROCEDURES

intro .... introduction to system maintenance procedures
cat .... concatenate and print files
crash .... what to do when the system crashes
dconfig .... configure logical disks
dd .... convert and copy a file
dformat .... disk formatter
du .... summarize disk usage
fbackup .... make a fast tape backup of a file system
fsck .... file system consistency check and interactive repair
fsdb .... file system debugger
help .... ask for help
ls .... list contents of directories
mkfs .... construct a file system
od .... octal dump
restor .... incremental file system restore

Sys5 UNIX Administrator's Reference Manual

98-05084.1 Ver. F August, 1987

PLEXUS COMPUTERS, INC.
3833 North First Street
San Jose, CA 95134
408/943-9433

The information contained herein is subject to change without notice. Therefore, Plexus Computers, Inc. assumes no responsibility for the accuracy of the information presented in this document beyond its current release date.

Printed in the United States of America

CONTENTS

1. COMMANDS AND APPLICATION PROGRAMS

1. COMMANDS AND APPLICATION PROGRAMS

intro ..... introduction to commands and application programs
300 ..... handle special functions of DASI 300 and 300s terminals
4014 ..... paginator for the TEKTRONIX 4014 terminal
450 ..... handle special functions of the DASI 450 terminal
acctcom ..... search and print process accounting file(s)
adb ..... absolute debugger
admin ..... create and administer SCCS files
ar ..... archive and library maintainer for portable archives
arcv ..... convert archive files from PDP-11 to common archive format
as ..... common assembler
asa ..... interpret ASA carriage control characters
at ..... execute commands at a later time
awk ..... pattern scanning and processing language
banner ..... make posters
bar ..... Berkeley archive and library maintainer
basename ..... deliver portions of path names
bbanner ..... print large banner on printer
bc ..... arbitrary-precision arithmetic language
bdiff ..... big diff
bfs ..... big file scanner
bls ..... list contents of directory
bs ..... a compiler/interpreter for modest-sized programs
cal ..... print calendar
calendar ..... reminder service
cat ..... concatenate and print files
cb ..... C program beautifier
cc ..... C compiler
cd ..... change working directory
cdc ..... change the delta commentary of an SCCS delta
cflow ..... generate C flow graph
chmod ..... change mode
chown ..... change owner or group
clear ..... clear terminal screen
cmp ..... compare two files
col ..... filter reverse line-feeds
comb ..... combine SCCS deltas
comm ..... select or reject lines common to two sorted files
copytape ..... make an image copy of a tape
cp ..... copy, link or move files
cpio ..... copy file archives in and out
cpp ..... the C language preprocessor
crontab ..... user crontab file
crypt ..... encode/decode
csh ..... a shell (command interpreter) with C-like syntax
csplit ..... context split
ct ..... spawn getty to a remote terminal
ctags ..... create a tags file
ctrace ..... C program debugger

cu .... call another UNIX system
cut .... cut out selected fields of each line of a file
cxref .... generate C program cross-reference
date .... print and set the date
dc .... desk calculator
dd .... convert and copy a file
delta .... make a delta (change) to an SCCS file
deroff .... remove nroff/troff, tbl, and eqn constructs
dial .... dial a Racal-Vadic 3451 modem
diff .... differential file comparator
diff3 .... 3-way differential file comparison
diffmk .... mark differences between files
dircmp .... directory comparison
du .... summarize disk usage
dump .... dump selected parts of an object file
dx9700 .... prepare troff documents for the Xerox 9700 printer
echo .... echo arguments
ed .... text editor
edit .... text editor (variant of ex for casual users)
efl .... Extended Fortran Language
enable .... enable/disable LP printers
env .... set environment for command execution
eqn .... format mathematical text for nroff or troff
ex .... text editor
expr .... evaluate arguments as an expression
f77 .... Fortran 77 compiler
factor .... factor a number
file .... determine file type
find .... find files
fsplit .... split f77, ratfor, or efl files
gdev .... graphical device routines and filters
ged .... graphical editor
get .... get a version of an SCCS file
getopt .... parse command options
graph .... draw a graph
graphics .... access graphical and numerical commands
greek .... select terminal filter
grep .... search a file for a pattern
gutil .... graphical utilities
head .... give first few lines of a stream
help .... ask for help
hp .... handle special functions of HP 2640 and 2621-series terminals
hyphen .... find hyphenated words
id .... print user and group IDs and names
ipcrm .... remove a message queue, semaphore set or shared memory id
ipcs .... report inter-process communication facilities status
join .... relational database operator
kill .... terminate a process
ld .... link editor for common object files
lex .... generate programs for simple lexical tasks
line .... read one line
lint .... a C program checker
login .... sign on

logname....get login name
lorder....find ordering relation for an object library
lp....send/cancel requests to an LP line printer
lphold....postpone printing, resume printing
lpstat....print LP status information
ls....list contents of directory
m4....macro processor
macref....produce cross-reference listing of macro files
mail....send mail to users or read mail
mailx....interactive message processing system
make....maintain, update, and regenerate groups of programs
makekey....generate encryption key
man....print entries in this manual
mesg....permit or deny messages
mkdir....make a directory
mkstr....create an error message file by massaging C source
mm....print/check documents formatted with the MM macros
mmlint....sroff/MM nroff/MM document compatibility checker
mmt....typeset documents, viewgraphs, and slides
more....file perusal filter for crt viewing
newform....change the format of a text file
newgrp....log in to a new group
news....print news items
nice....run a command at low priority
nl....line numbering filter
nm....print name list of common object file
nohup....run a command immune to hangups and quits
nroff....format or typeset text
ocw....prepare constant-width text for otroff
od....octal dump
pack....compress and expand files
passwd....change login password
paste....merge same lines of several files or subsequent lines of one file
pg....file perusal filter for soft-copy terminals
pic....troff preprocessor for drawing simple pictures
pr....print files
printenv....print out the environment
prof....display profile data
prs....print an SCCS file
ps....report process status
ptx....permuted index
pwd....working directory name
ratfor....rational Fortran dialect
regcmp....regular expression compile
rm....remove files or directories
rmdel....remove a delta from an SCCS file
sact....print current SCCS file editing activity
sag....system activity graph
sar....system activity reporter
scc....C compiler for stand-alone programs
sccsdiff....compare two versions of an SCCS file
script....make typescript of terminal session
sdiff....side-by-side difference program

sed .... stream editor
sh .... shell, the standard/restricted command programming language
size .... print section sizes of common object files
sleep .... suspend execution for an interval
sno .... SNOBOL interpreter
sort .... sort and/or merge files
spell .... find spelling errors
spline .... interpolate smooth curve
split .... split a file into pieces
sroff .... format text
stat .... statistical network useful with graphical commands
strings .... find the printable strings in a object, or other binary, file
strip .... strip symbol and line number information from common object file
stty .... set the options for a terminal
style .... analyze surface characteristics of a document
su .... become super-user or another user
sum .... print checksum and block count of a file
sync .... update the super block
tabs .... set tabs on a terminal
tail .... deliver the last part of a file
tape .... tape manipulation
tar .... tape file archiver
tbl .... format tables for nroff or troff
tc .... troff output interpreter
tee .... pipe fitting
test .... condition evaluation command
time .... time a command
timex .... time a command; report process data and system activity
toc .... graphical table of contents routines
touch .... update access and modification times of a file
tplot .... graphics filters
tput .... query terminfo database
tr .... translate characters
troff .... text formatting and typesetting
true .... provide truth values
tset .... set terminal modes
tsort .... topological sort
tty .... get the name of the terminal
umask .... set file-creation mode mask
uname .... print name of current UNIX system
unget .... undo a previous get of an SCCS file
uniq .... report repeated lines in a file
units .... conversion program
uucp .... UNIX system to UNIX system copy
uuencode .... encode/decode a binary file for transmission via mail
uustat .... uucp status inquiry and job control
uuto .... public UNIX-to-UNIX system file copy
uux .... UNIX-to-UNIX system command execution
val .... validate SCCS file
vc .... version control
vi .... screen-oriented (visual) display editor based on ex
vtty .... connect to a remote host via NOS
wait .... await completion of process

wc .... word count
what .... identify SCCS files
who .... who is on the system
write .... write to another user
x9700 .... prepare nroff documents for the Xerox 9700 printer
xargs .... construct argument list(s) and execute command
xstr .... extract strings from C programs to implement shared strings
yacc .... yet another compiler-compiler

1M. SYSTEM MAINTENANCE COMMANDS AND PROGRAMS

intro .... system maintenance commands and application programs
accept .... allow/prevent LP requests
acct .... overview of accounting and miscellaneous accounting commands
acctcms .... command summary from per-process accounting records
acctcon .... connect-time accounting
acctmerg .... merge or add total accounting files
acctprc .... process accounting
acctsh .... shell procedures for accounting
acpdmp .... dump contents of Advanced Communication
brc .... system initialization shell scripts
brdrst .... reset the VCP controller
cdconf .... configurable disk configuration utility
checkall .... faster file system checking procedure
chroot .... change root directory for a command
clri .... clear i-node
cpset .... install object files in binary directories
crash .... examine system images
cron .... clock daemon
dataio .... download program files
dconfig .... configure logical disks
dcopy .... copy file systems for optimal access time
devnm .... device name
df .... report number of free disk blocks
diskusg .... generate disk accounting data by user ID
dnld .... download program files
dump .... incremental file system dump
dumpdir .... print the names of files on a dump tape
errdead .... extract error records from dump
errdemon .... error-logging daemon
errpt .... process a report of logged errors
errstop .... terminate the error-logging daemon
fbackup .... make a fast tape backup of a file system
ff .... list file names and statistics for a file system
filesave .... daily/weekly UNIX system file system backup
finc .... fast incremental backup
frec .... recover files from a backup tape
fsck .... file system consistency check and interactive repair
fsdb .... file system debugger
fuser .... identify processes using a file or file structure
fwtmp .... manipulate connect accounting records
getty .... set terminal type, modes, speed, and line discipline
icpdmp .... dump contents of an Intelligent Communication

init .... process control initialization
install .... install commands
killall .... kill all active processes
link .... exercise link and unlink system calls
lpadmin .... configure the LP spooling system
lpsched .... start/stop the LP request scheduler and move requests
mirutil .... utility for connecting two identical
mkfs .... construct a file system
mknod .... build special file
mount .... mount and dismount file system
mvdir .... move a directory
ncheck .... generate names from i-numbers
non-btl .... reinstall MM macros without Bell Laboratories specific features
profiler .... operating system profiler
pwck .... password/group file checkers
ramdisk .... memory as disk
restor .... incremental file system restore
runacct .... run daily accounting
sadp .... disk access profiler
sar .... system activity report package
setmnt .... establish mount table
shutdown .... terminate all processing
sys .... System control and status program.
tic .... terminfo compiler
topq .... prioritize print queue
uucico .... file transport program for the uucp system
uuclean .... uucp spool directory clean-up
uusub .... monitor uucp network
uuxqt .... execute remote command requests
vconfig .... virtual terminal configuration
vcpdmp .... dump contents of VMEbus Comm. Proc. memory into a file
volcopy, labelit .... copy file systems with label checking
wall .... write to all users
whodo .... who is doing what

2. SYSTEM CALLS

2. SYSTEM CALLS

intro .... introduction to system calls and error numbers
access .... determine accessibility of a file
acct .... enable or disable process accounting
alarm .... set a process alarm clock
brk .... change data segment space allocation
chdir .... change working directory
chmod .... change mode of file
chown .... change owner and group of a file
chroot .... change root directory
close .... close a file descriptor
creat .... create a new file or rewrite an existing one
dup .... duplicate an open file descriptor
exec .... execute a file
exit .... terminate process

fcntl .... file control
fork .... create a new process
getpid .... get process, process group, and parent process IDs
getuid .... get real user, effective user, real group, and effective group IDs
ioctl .... control device
kill .... send a signal to a process or a group of processes
link .... link to a file
lockf .... provide exclusive file regions for reading or writing
lseek .... move read/write file pointer
mknod .... make a directory, or a special or ordinary file
mount .... mount a file system
msgctl .... message control operations
msgget .... get message queue
msgop .... message operations
nice .... change priority of a process
open .... open for reading or writing
pause .... suspend process until signal
pipe .... create an interprocess channel
plock .... lock process, text, or data in memory
profil .... execution time profile
ptrace .... process trace
read .... read from file
semctl .... semaphore control operations
semget .... get set of semaphores
semop .... semaphore operations
setpgrp .... set process group ID
setuid, setgid .... set user and group IDs
shmctl .... shared memory control operations
shmget .... get shared memory segment
shmop .... shared memory operations
signal .... specify what to do upon receipt of a signal
stat, fstat .... get file status
stime .... set time
sync .... update super-block
time .... get time
times .... get process and child process times
ulimit .... get and set user limits
umask .... set and get file creation mask
umount .... unmount a file system
uname .... get name of current UNIX system
unlink .... remove directory entry
ustat .... get file system statistics
utime .... set file access and modification times
wait .... wait for child process to stop or terminate
write .... write on a file

2S. STANDALONE SYSTEM CALLS

intro .... introduction to standalone system calls,
access .... determine accessibility of a file
brk .... change data segment space allocation
chdir .... change working directory
chmod .... change mode of file

close ....close a file descriptor
creat ....create a new special file
exit...terminate process
float ....float and double routines
getargv ....display a program name and get arguments for
getpid ....get process ID
getuid ....get real user, effective user, real group, and effective group IDs
gtty ....get terminal characterisitcs
isatty...................................returns a 1 if specified file descriptor is a terminal
kill ....send a signal to a process or a group of processes
lseek..move read/write file pointer
mknod...make a special file
mount ....mount a file system
nice ....change priority of a process
open..open for reading or writing
read ....read from file
sleep..suspend execution for interval
srcheof ....position to a specific file number on a tape
stat...get file status
stime ....set time
stty..set terminal characteristics
tell..report the current value of a file pointer
time ....get time
umask ....set and get file creation mask
umount..unmount a file system
ustat ....get file system statistics
write ....write on a file

3. SUBROUTINES

3C and 3S. C AND ASSEMBLER, STANDARD I/O LIBRARY ROUTINES

intro ....introduction to subroutines and libraries
a64l..............................convert between long integer and base-64 ASCII string
abort ....generate an IOT fault
abs ....return integer absolute value
bsearch ....binary search a sorted table
clock ....report CPU time used
conv ....translate characters
crypt ....generate DES encryption
ctermid ....generate file name for terminal
ctime ....convert date and time to string
ctype ....classify characters
cuserid ....get character login name of the user
dial...establish an out-going terminal line connection
drand48.....................generate uniformly distributed pseudo-random numbers
ecvt ....convert floating-point number to string
end..last locations in program
fclose ....close or flush a stream
ferror..stream status inquiries
fopen...open a stream
fread ....binary input/output
frexp...manipulate parts of floating-point numbers

fseek .... reposition a file pointer in a stream
ftw .... walk a file tree
getc .... get character or word from a stream
getcwd .... get path-name of current working directory
getenv .... return value for environment name
getgrent .... get group file entry
getlogin .... get login name
getopt .... get option letter from argument vector
getpass .... read a password
getpw .... get name from UID
getpwent .... get password file entry
gets .... get a string from a stream
getut .... access utmp file entry
hsearch .... manage hash search tables
l3tol .... convert between 3-byte integers and long integers
lsearch .... linear search and update
malloc .... main memory allocator
memory .... memory operations
mktemp .... make a unique file name
monitor .... prepare execution profile
nlist .... get entries from name list
perror .... system error messages
popen .... initiate pipe to/from a process
printf .... print formatted output
putc .... put character or word on a stream
putenv .... change or add value to environment
putpwent .... write password file entry
puts .... put a string on a stream
qsort .... quicker sort
rand .... simple random-number generator
scanf sscanf .... convert formatted input
setbuf .... assign buffering to a stream
setjmp .... non-local goto
sleep .... suspend execution for interval
ssignal .... software signals
stdio .... standard buffered input/output package
stdipc .... standard interprocess communication package
string .... string operations
strtod .... convert string to double-precision number
strtol .... convert string to integer
swab .... swap bytes
system .... issue a shell command
termlib .... terminal independent operation routines
tmpfile .... create a temporary file
tmpnam .... create a name for a temporary file
tsearch .... manage binary search trees
ttyname .... find name of a terminal
ttyslot .... find the slot in the utmp file of the current user
ungetc .... push character back into input stream
vprintf .... print formatted output of a varargs argument list

3M. MATHEMATICAL LIBRARY ROUTINES

bessel.......Bessel functions
erf.......error function and complementary error function
exp.......exponential, logarithm, power, square root functions
floor.......floor, ceiling, remainder, absolute value functions
gamma.......log gamma function
hypot.......Euclidean distance function
matherr.......error-handling function
sinh.......hyperbolic functions
trig.......trigonometric functions

3X. MISCELLANEOUS ROUTINES

assert.......verify program assertion
curses.......CRT screen handling and optimization package
ldahread.......read the archive header of a member of an archive file
ldclose, ldaclose.......close a common object file
ldfhread.......read the file header of a common object file
ldgetname.......retrieve symbol name for common object file symbol table entry
ldlread.......manipulate line number entries of a common object file function
ldlseek.......seek to line number entries of a section of a common object file
ldohseek.......seek to the optional file header of a common object file
ldopen.......open a common object file for reading
ldrseek.......seek to relocation entries of a section of a common object file
ldshread.......read an indexed/named section header of a common object file
ldsseek.......seek to an indexed/named section of a common object file
ldtbindex.......compute the index of a symbol table entry of a common object file
ldtbread.......read an indexed symbol table entry of a common object file
ldtbseek.......seek to the symbol table of a common object file
logname.......return login name of user
malloc.......fast main memory allocator
plot.......graphics interface subroutines
regcmp.......compile and execute regular expression

3F. FORTRAN ROUTINES

abort.......terminate Fortran program
abs.......Fortran absolute value
acos.......Fortran arccosine intrinsic function
aimag.......Fortran imaginary part of complex argument
aint.......Fortran integer part intrinsic function
asin.......Fortran arcsine intrinsic function
atan.......Fortran arctangent intrinsic function
atan2.......Fortran arctangent intrinsic function
bool.......Fortran bitwise boolean functions
conjg.......Fortran complex conjugate intrinsic function
cos.......Fortran cosine intrinsic function
cosh.......Fortran hyperbolic cosine intrinsic function
dim.......positive difference intrinsic functions
dprod.......double precision product intrinsic function
exp.......Fortran exponential intrinsic function
ftype.......explicit Fortran type conversion

getarg.....return Fortran command-line argument
getenv.....return Fortran environment variable
iargc.....returns number of command line arguments passed to the program
index.....return location of Fortran substring
len.....return length of Fortran string
log.....Fortran natural logarithm intrinsic function
log10.....Fortran common logarithm intrinsic function
max.....Fortran maximum-value functions
mclock.....return Fortran time accounting
min.....Fortran minimum-value functions
mod.....Fortran remaindering intrinsic functions
rand.....random number generator
round.....Fortran nearest integer functions
sign.....Fortran transfer-of-sign intrinsic function
signal.....specify Fortran action on receipt of a system signal
sin.....Fortran sine intrinsic function
sinh.....Fortran hyperbolic sine intrinsic function
sputl.....access long integer data in a machine-independent fashion
sqrt.....Fortran square root intrinsic function
strcmp.....string comparison intrinsic functions
system.....issue a shell command from Fortran
tan.....Fortran tangent intrinsic function
tanh.....Fortran hyperbolic tangent intrinsic function

4. FILE FORMATS

intro.....introduction to file formats
L-devices.....link devices, connection information
L-dialcodes.....alphabetic dialing abbreviations file
L.cmds.....remote execution commands
L.sys.....link systems
USERFILE.....UUCP pathname permissions file
a.out.....common assembler and link editor output
acct.....per-process accounting file format
ar..... common archive file format
checklist.....list of file systems processed by fsck
core.....format of core image file
cpio.....format of cpio archive
dialups.....list of dialup devices
dir.....format of directories
dump.....incremental dump tape format
d_passwd.....dialup password file
errfile.....error-log file format
filehdr.....file header for common object files
fs.....format of system volume
fspec.....format specification in text files
gettydefs.....speed and terminal settings used by getty
gps.....graphical primitive string, format of graphical files
group.....group file
inittab.....script for the init process
inode.....format of an i-node
ioctl.syscon.....system console configuration file

issue .... issue identification file
ldfcn .... common object file access routines
linenum .... line number entries in a common object file
mnttab .... mounted file system table
passwd .... password file
plot .... graphics interface
profile .... setting up an environment at login time
reloc .... relocation information for a common object file
sccsfile .... format of SCCS file
scnhdr .... section header for a common object file
syms .... common object file symbol table format
term .... format of compiled term file.
termcap .... terminal capability data base
terminfo .... terminal capability data base
utmp .... utmp and wtmp entry formats

5. MISCELLANEOUS FACILITIES

intro .... introduction to miscellany
ascii .... map of ASCII character set
environ .... user environment
eqnchar .... special character definitions for eqn and neqn
fcntl .... file control options
font .... description files for device-independent troff
man .... macros for formatting entries in this manual
math .... math functions and constants
mm .... the MM macro package for formatting documents
mosd .... the OSDD adapter macro package for formatting documents
mptx .... the macro package for formatting a permuted index
mv .... a troff macro package for typesetting viewgraphs and slides
prof .... profile within a function
profile .... setting up an environment at login time
regexp .... regular expression compile and match routines
stat .... data returned by stat system call
term .... conventional names for terminals
troff .... description of output language
ttytype .... data base of terminal types by port
types .... primitive system data types
values .... machine-dependent values
varargs .... handle variable argument list

6. GAMES

intro .... introduction to games
arithmetic .... provide drill in number facts
back .... the game of backgammon
bj .... the game of black jack
craps .... the game of craps
hangman .... guess the word
maze .... generate a maze

moo .... guessing game
quiz .... test your knowledge
wump .... the game of hunt-the-wumpus

7. SPECIAL FILES

intro .... introduction to special files
acp .... Advanced Communications Processor
ccb .... common circuits board driver
cd .... configurable disk drive
conslog .... copy of error messages from UNIX to console
dsk .... (optionally) mirrored disk driver
err .... error-logging interface
ft .... IMSP streaming cartridge controller
icp .... Intelligent Communications Processor
imsp .... Intelligent
mem .... core memory
mv .... a macro package for making view graphs
null .... the null file
od .... optical disk
pd .... IMSP disk controller
pp .... parallel port interface
prf .... operating system profiler
pt .... IMSP cartridge controller
rm .... Cipher Microstreamer tape drive
rram .... allows memory to be used as a disk
swap .... swap device
tty .... general terminal interface

8. SYSTEM MAINTENANCE AND STANDALONE PROCEDURES

intro .... introduction to system maintenance procedures
cat .... concatenate and print files
crash .... what to do when the system crashes
dconfig .... configure logical disks
dd .... convert and copy a file
dformat .... disk formatter
du .... summarize disk usage
fbackup .... make a fast tape backup of a file system
fsck .... file system consistency check and interactive repair
fsdb .... file system debugger
help .... ask for help
ls .... list contents of directories
mkfs .... construct a file system
od .... octal dump
restor .... incremental file system restore

PERMUTED INDEX

a isatty returns a	1 if specified file descriptor is	isatty(2S)
handle special functions of HP	2640 and 2621-series terminals	hp(1)
comparison diff3	3-way differential file	diff3(1)
handle special functions of DASI	300 and 300s terminals	300(1)
dial dial a Racal-Vadic	3451 modem	dial(1)
for the TEKTRONIX 4014 terminal	4014 paginator	4014(1)
of the DASI 450 terminal	450 handle special functions	450(1)
Fortran	77 compiler f77	f77(1)
troff documents for the Xerox	9700 printer dx9700 prepare	dx9700(1)
nroff documents for the Xerox	9700 printer x9700 prepare	x9700(1)
asa interpret	ASA carriage control characters	asa(1)
acpdmp dump contents of	Advanced Communication	acpdmp(1M)
acp	Advanced Communications Processor	acp(7)
maintainer bar	Berkeley archive and library	bar(1)
	C compiler	cc(1)
programs scc	C compiler for stand-alone	scc(1)
generate	C flow graph	cflow(1)
the	C language preprocessor	cpp(1)
	C program beautifier	cb(1)
lint	C program checker	lint(1)
generate	C program cross-reference cxref	cxref(1)
	C program debugger ctrace	ctrace(1)
report	CPU time used clock	clock(3C)
optimization package curses	CRT screen handling and	curses(3X)
shell (command interpreter) with	C-like syntax	csh(1)
dump contents of Advanced	Communication acpdmp	acpdmp(1M)
acp Advanced	Communications Processor	acp(7)
icp Intelligent	Communications Processor	icp(7)
handle special functions of the	DASI 450 terminal 450	450(1)
generate	DES encryption crypt	crypt(3C)
exits. write	EOT on the other terminal and	write(1)
	Euclidean distance function	hypot(3M)
	Extended Fortran Language	efl(1)
	Fortran 77 compiler	f77(1)
Extended	Fortran Language	efl(1)
	Fortran absolute value	abs(3F)
system signal signal specify	Fortran action on receipt of a	signal(3F)
function acos	Fortran arccosine intrinsic	acos(3F)
functio asin	Fortran arcsine intrinsic	asin(3F)
function atan2	Fortran arctangent intrinsic	atan2(3F)
function atan	Fortran arctangert intrinsic	atan(3F)
functions bool	Fortran bitwise boolean	bool(3F)
getarc return	Fortran command-line argument	getarc(3F)
intrinsic function log10	Fortran common logarithm	log10(3F)
intrinsic function conjg	Fortran complex conjugate	conjg(3F)
functions cos	Fortran cosine intrinsic	cos(3F)
rational	Fortran dialect ratfor	ratfor(1)
getenv return	Fortran environment variable	getenv(3F)
function exp	Fortran exponential intrinsic	exp(3F)
intrinsic function cosh	Fortran hyperbolic cosine	cosh(3F)
function sinh	Fortran hyperbolic sine intrinsic	sinh(3F)

intrinsic function tanh	Fortran hyperbolic tangent	tanh(3F)
complex argument aimag	Fortran imaginary part of	aimag(3F)
function aint	Fortran integer part intrinsic	aint(3F)
max	Fortran maximum-value functions	max(3F)
min	Fortran minimum-value functions	min(3F)
intrinsic function log	Fortran natural logarithm	log(3F)
functions round	Fortran nearest integer	round(3F)
terminate	Fortran program abort	abort(3F)
functions mod	Fortran remaindering intrinsic	mod(3F)
sin	Fortran sine intrinsic function	sin(3F)
function sqrt	Fortran square root intrinsic	sqrt(3F)
return length of	Fortran string len	len(3F)
return location of	Fortran substring index	index(3F)
issue a shell command from	Fortran system	system(3F)
function tan	Fortran tangent intrinsic	tan(3F)
return	Fortran time accounting mclock	mclock(3F)
intrinsic function sign	Fortran transfer-of-sign	sign(3F)
explicit	Fortran type conversion ftype	ftype(3F)
hp handle special functions of	HP 2640 and 2621-series terminals	hp(1)
disk accounting data by user	ID diskusg diskusg - generate	diskusg(1M)
set process group	ID setpgrp	setpgrp(2)
getpid get process	ID	getpid(2S)
and real and effective group	ID get real and effective user,	getuid(2S)
real group, and effective group	ID's getuid effective user,	getuid(2)
print user and group	IDs and names id	id(1)
process group, and parent process	IDs getpid get process,	getpid(2)
set user and group	IDs setuid	setuid(2)
pt	IMSP cartridge controller	pt(7)
pd	IMSP disk controller	pd(7)
controller ft	IMSP streaming cartridge	ft(7)
generate an	IOT fault abort	abort(3C)
icpdmp dump contents of an	Intelligent Communication	icpdmp(1M)
Processor icp	Intelligent Communications	icp(7)
Processor imsp	Intelligent Mass Storage	imsp(7)
connection information	L-devices link devices,	L-devices(4)
abbreviations file	L-dialcodes alphabetic dialing	L-dialcodes(4)
commands	L.cmds remote execution	L.cmds(4)
	L.sys link systems	L.sys(4)
send/cancel requests to an	LP line printer lp	lp(1)
enable/disable	LP printers enable	enable(1)
lpshut, lpmove start/stop the	LP request scheduler and	lpsched(1M)
configure the	LP spooling system lpadmin	lpadmin(1M)
print	LP status information lpstat	lpstat(1)
Extended Fortran	Language efl	efl(1)
documents mm the	MM macro package for formatting	mm(5)
documents formatted with the	MM macros mm print/check	mm(1)
imsp Intelligent	Mass Storage Processor	imsp(7)
formatting documents mosd the	OSDD adapter macro package for	mosd(5)
mdial dial the	P/75 onboard modem	dial(1)
execute a command on the	PCL network net	net(1)
acp Advanced Communications	Processor	acp(7)
icp Intelligent Communications	Processor	icp(7)
imsp Intelligent Mass Storage	Processor	imsp(7)

dial dial a	Racal-Vadic 3451 modem	dial(1)
change the delta commentary of an	SCCS delta cdc	cdc(1)
combine	SCCS deltas comb	comb(1)
make a delta (change) to an	SCCS file delta	delta(1)
print current	SCCS file editing activity sact	sact(1)
get a version of a	SCCS file get	get(1)
print an	SCCS file prs	prs(1)
remove a delta from an	SCCS file rmdel	rmdel(1)
compare two versions of an	SCCS file sccsdiff	sccsdiff(1)
format of	SCCS file sccsfile	sccsfile(4)
undo a previous get of an	SCCS file unget	unget(1)
validate	SCCS file val	val(1)
create and administer	SCCS files admin	admin(1)
identify	SCCS files what	what(1)
paginator for the	TEKTRONIX 4014 terminal	4014(1)
get name from	UID getpw	getpw(3C)
cu call another	UNIX system	cu(1C)
UNIX system to	UNIX system copy uucp	uucp(1)
filesave daily/weekly	UNIX system file system backup	filesave(1M)
uucp	UNIX system to UNIX system copy	uucp(1)
print name of current	UNIX system uname	uname(1)
get name of current	UNIX system uname	uname(2)
copy of error messages from	UNIX to console	conslog(7)
execution uux	UNIX-to-UNIX system command	uux(1)
uuto public	UNIX-to-UNIX system file copy	uuto(1)
USERFILE	UUCP pathname permissions file	USERFILE(4)
prepare troff documents for the	Xerox 9700 printer dx9700	dx9700(1)
prepare nroff documents for the	Xerox 9700 printer x9700	x9700(1)
Processor	acp Advanced Communications	acp(7)
assembler and link editor output	a.out common	a.out(4)
integer and base-64 ASCII string	a64l convert between long	a64l(3C)
L-dialcodes alphabetic dialing	abbreviations file	L-dialcodes(4)
generate an IOT fault	abort	abort(3C)
terminate Fortran program	abort	abort(3F)
return integer absolute value	abs	abs(3C)
Fortran absolute value	abs	abs(3F)
	absolute debugger adb	adb(1)
return integer	absolute value abs	abs(3C)
Fortran	absolute value abs	abs(3F)
floor, ceiling, remainder,	absolute value functions floor	floor(3M)
a file touch update	access and modification times of	touch(1)
utime set file	access and modification times	utime(2)
of a file	access determine accessibility	access(2S)
commands graphics	access graphical and numerical	graphics(1)
machine-independent fashion.	access long integer data in a	sputl(3X)
disk	access profiler sadp	sadp(1M)
common object file	access routines ldfcn	ldfcn(4)
copy file systems for optimal	access time dcopy	dcopy(1M)
	access utmp file entry getut	getut(3C)
determine accessibility of a file	access	access(2)
enable or disable process	accounting acct	acct(2)
connect-time	accounting acctcon	acctcon(1M)
of accounting and miscellaneous	accounting commands acct	acct(1M)

diskusg - generate disk	accounting data by user ID	diskusg(1M)
per-process	accounting file format acct	acct(4)
search and print process	accounting file(s) acctcom	acctcom(1)
merge or add total	accounting files acctmerg	acctmerg(1M)
return Fortran time	accounting mclock	mclock(3F)
command summary from per-process	accounting records acctcms	acctcms(1M)
run daily	accounting runacct	runacct(1M)
or disable process accounting	acct enable	acct(2)
miscellaneous accounting commands	acct overview of accounting and	acct(1M)
accounting file format	acct per-process	acct(4)
per-process accounting records	acctcms command summary from	acctcms(1M)
print process accounting file(s)	acctcom search and	acctcom(1)
connect-time accounting	acctcon	acctcon(1M)
or add total accounting files	acctmerg merge	acctmerg(1M)
arccosine intrinsic function	acos Fortran	acos(3F)
Advanced Communication	acpdmp dump contents of	acpdmp(1M)
signal signal specify Fortran	action on receipt of a system	signal(3F)
kill all	active processes killall	killall(1M)
system	activity graph sag	sag(1)
system	activity report package sar	sar(1M)
system	activity report sail	sail(1)
print current SCCS file editing	activity sact	sact(1)
report process data and system	activity timex time a command;	timex(1)
formatting mosd the OSDD	adapter macro package for	mosd(5)
absolute debugger	adb	adb(1)
acctmerg merge or	add total accounting files	acctmerg(1M)
change or	add value to environment putenv	putenv(3C)
create and	administer SCCS files admin	admin(1)
part of complex argument	aimag Fortran imaginary	aimag(3F)
integer part intrinsic function	aint Fortran	aint(3F)
set a process	alarm clock alarm	alarm(2)
change data segment space	allocation brk	brk(2)
brk change data segment space	allocation	brk(2S)
main memory	allocator malloc	malloc(3C)
fast main memory	allocator malloc	malloc(3X)
disk rram	allows memory to be used as a	rram(7)
abbreviations file L-dialcodes	alphabetic dialing	L-dialcodes(4)
of a document	analyze surface characteristics	style(1)
sort	and/or merge files sort	sort(1)
common archive file format	ar	ar(4)
maintainer for portable archives	ar archive and library	ar(1)
language bc	arbitrary-precision arithmetic	bc(1)
Fortran	arccosine intrinsic function acos	acos(3F)
for portable archives ar	archive and library maintainer	ar(1)
bar Berkeley	archive and library maintainer	bar(1)
format of cpio	archive cpio	cpio(4)
common	archive file format ar	ar(4)
archive header of a member of an	archive file ldahread read the	ldahread(3X)
common archive format convert	archive files	arcv(1)
files	archive format arcv archive	arcv(1)
archive file ldahread read the	archive header of a member of an	ldahread(3X)
tape file	archiver tar	tar(1)
library maintainer for portable	archives ar archive and	ar(1)

copy file	archives in and out cpio	cpio(1)
Fortran	arcsine intrinsic functio asin	asin(3F)
atan2 Fortran	arctangent intrinsic function	atan2(3F)
atan Fortran	arctangert intrinsic function	atan(3F)
archive format	arcv convert archive files from	arcv(1)
Fortran imaginary part of complex	argument aimag	aimag(3F)
return Fortran command-line	argument getarc	getarc(3F)
handle variable	argument list varargs	varargs(5)
formatted output of a varargs	argument list vprintf print	vprintf(3S)
command xargs construct	argument list(s) and execute	xargs(1)
get option letter from	argument vector getopt	getopt(3C)
evaluate	arguments as an expression expr	expr(1)
echo	arguments echo	echo(1)
display a program name and get	arguments for getargv	getargv(2S)
arbitrary-precision	arithmetic language bc	bc(1)
provide drill in number facts	arithmetic	arithmetic(6)
map of ASCII character set	ascii	ascii(5)
Fortran arcsine intrinsic functio	asin	asin(3F)
	ask for help	help(1)
help	ask for help	help(8)
a.out common	assembler and link editor output	a.out(4)
common	assembler as	as(1)
verify program	assertion assert	assert(3X)
setbuf, setvbuf	assign buffering to a stream	setbuf(3S)
arctangent intrinsic function	atan2 Fortran	atan2(3F)
	await completion of process wait	wait(1)
scanning and processing language	awk pattern	awk(1)
the game of	backgammon back	back(6)
UNIX system file system	backup filesave daily/weekly	filesave(1M)
fast incremental	backup finc	finc(1M)
fbackup fast tape	backup of a file system	fbackup(1M)
fbackup make a fast tape	backup of a file system	fbackup(8)
recover files from a	backup tape frec	frec(1M)
make posters	banner	banner(1)
print large	banner on printer	bbanner(1)
library maintainer	bar Berkeley archive and	bar(1)
ttytype data	base of terminal types by port	ttytype(5)
terminal capability data	base terminfo	terminfo(4)
convert between long integer and	base-64 ASCII string a64l	a64l(3C)
(visual) display editor	based on ex vi screen-oriented	vi(1)
deliver portions of path names	basename	basename(1)
printer	bbanner print large banner on	bbanner(1)
arithmetic language	bc arbitrary-precision	bc(1)
big diff	bdiff	bdiff(1)
C program	beautifier cb	cb(1)
su	become super-user or another user	su(1)
big file scanner	bfs	bfs(1)
	big diff bdiff	bdiff(1)
	big file scanner bfs	bfs(1)
install object files in	binary directories cpset	cpset(1M)
	binary input/output fread	fread(3S)
bsearch	binary search a sorted table	bsearch(3C)
manage	binary search trees tsearch	tsearch(3C)

Fortran	bitwise boolean functions bool	bool(3F)
the game of	black jack bj	bj(6)
print checksum and	block count of a file sum	sum(1)
update the super	block sync	sync(1)
report number of free disk	blocks df	df(1M)
Fortran bitwise boolean functions	bool	bool(3F)
initialization shell scripts	brc system	brc(1M)
reset the VCP controller	brdrst	brdrst(1M)
allocation	brk change data segment space	brk(2S)
data segment space allocation	brk change	brk(2)
for modest-sized programs	bs a compiler/interpreter	bs(1)
binary search a sorted table	bsearch	bsearch(3C)
stdio standard	buffered input/output package	stdio(3S)
assign	buffering to a stream setbuf, setvbuf	setbuf(3S)
	build special file mknod	mknod(1M)
swap	bytes swab	swab(3C)
vcpdmp dump contents of VMEbus	Communication Processor's	vcpdmp(1M)
crm	caching reel-to-reel tape driver	rm(7)
print calendar	cal	cal(1)
desk	calculator dc	dc(1)
print	calendar cal	cal(1)
reminder service	calendar	calendar(1)
cu	call another UNIX system	cu(1C)
data returned by stat system	call stat	stat(5)
exercise link and unlink system	calls link	link(1M)
terminal	capability data base terminfo	terminfo(4)
interpret ASA	carriage control characters asa	asa(1)
ft IMSP streaming	cartridge controller	ft(7)
pt IMSP	cartridge controller	pt(7)
text editor (variant of ex for	casual users) edit	edit(1)
files	cat concatenate and print	cat(8)
concatenate and print files	cat	cat(1)
C program beautifier	cb	cb(1)
C compiler	cc	cc(1)
driver	ccb common circuits board	ccb(7)
change working directory	cd	cd(1)
delta commentary of an SCCS delta	cdc change the	cdc(1)
value functions floor floor,	ceiling, remainder, absolute	floor(3M)
generate C flow graph	cflow	cflow(1)
allocation brk	change data segment space	brk(2)
allocation brk	change data segment space	brk(2S)
	change login password passwd	passwd(1)
	change mode chmod	chmod(1)
	change mode of file chmod	chmod(2)
chmod	change mode of file	chmod(2S)
environment putenv	change or add value to	putenv(3C)
file chown	change owner and group of a	chown(2)
nice	change priority of a process	nice(2)
nice	change priority of a process	nice(2S)
	change root directory chroot	chroot(2)
command chroot	change root directory for a	chroot(1M)
an SCCS delta cdc	change the delta commentary of	cdc(1)
file newform	change the format of a text	newform(1)

make a delta	(change) to an SCCS file delta	delta(1)
	change working directory cd	cd(1)
	change working directory chdir	chdir(2)
chdir	change working directory	chdir(2S)
create an interprocess	channel pipe	pipe(2)
ungetc push	character back into input stream	ungetc(3S)
neqn eqnchar special	character definitions for eqn and	eqnchar(5)
cuserid get	character login name of the user	cuserid(3S)
getc get	character or word from a stream	getc(3S)
putc put	character or word on a stream	putc(3S)
map of ASCII	character set ascii	ascii(5)
gtty get terminal	characterisitcs	gtty(2S)
analyze surface	characteristics of a document	style(1)
stty set terminal	characteristics	stty(2S)
interpret ASA carriage control	characters asa	asa(1)
translate	characters conv	conv(3C)
classify	characters ctype	ctype(3C)
translate	characters tr	tr(1)
	chdir change working directory	chdir(2S)
change working directory	chdir	chdir(2)
file system consistency	check and interactive repair fsck	fsck(1M)
fsck file system consistency	check and interactive repair	fsck(8)
file system checking procedure	checkall faster	checkall(1M)
a C program	checker lint	lint(1)
nroff/MM document compatibility	checker mmlint sroff/MM	mmlint(1)
password/group file	checkers pwck	pwck(1M)
faster file system	checking procedure checkall	checkall(1M)
copy file systems with label	checking volcopy	volcopy(1M)
of file systems processed by fsck	checklist list	checklist(4)
file sum print	checksum and block count of a	sum(1)
get process and	child process times	times(2)
terminate wait wait for	child process to stop or	wait(2)
	chmod change mode of file	chmod(2S)
change mode	chmod	chmod(1)
change mode of file	chmod	chmod(2)
change owner or group	chown	chown(1)
change root directory	chroot	chroot(2)
root directory for a command	chroot change	chroot(1M)
ccb common	circuits board driver	ccb(7)
	classify characters ctype	ctype(3C)
uucp spool directory	clean-up uuclean	uuclean(1M)
	clear i-node clri	clri(1M)
set a process alarm	clock alarm	alarm(2)
cron	clock daemon cron	cron(1M)
report CPU time used	clock	clock(3C)
ldclose	close a common object file	ldclose(3X)
	close a file descriptor close	close(2)
close	close a file descriptor	close(2S)
	close or flush a stream fclose	fclose(3S)
close a file descriptor	close	close(2)
clear i-node	clri	clri(1M)
compare two files	cmp	cmp(1)
filter reverse line-feeds	col	col(1)

	combine SCCS deltas comb	comb(1)
lines common to two sorted files	comm select or reject	comm(1)
run a	command at low priority nice	nice(1)
change root directory for a	command chroot	chroot(1M)
set environment for	command execution env	env(1)
UNIX-to-UNIX system	command execution uux	uux(1)
issue a shell	command from Fortran system	system(3F)
quits nohup run a	command immune to hangups and	nohup(1)
syntax a shell	(command interpreter) with C-like	csh(1)
execute a	command on the PCL network net	net(1)
parse	command options getopt	getopt(1)
shell, the standard/restricted	command programming language	sh(1)
per-process accounting records	command summary from	acctcms(1M)
issue a shell	command system	system(3S)
condition evaluation	command test	test(1)
time a	command time	time(1)
argument list(s) and execute	command xargs construct	xargs(1)
return Fortran	command-line argument getarc	getarc(3F)
system activity timex time a	command; report process data and	timex(1)
and miscellaneous accounting	commands acct of accounting	acct(1M)
execute	commands at a later time at	at(1)
access graphical and numerical	commands graphics	graphics(1)
install	commands install	install(1M)
network useful with graphical	commands stat statistical	stat(1)
change the delta	commentary of an SCCS delta	cdc(1)
	common archive file format	ar(4)
archive files	common archive format	arcv(1)
convert archive files	common archive format	arcv(1)
output a.out	common assembler and link editor	a.out(4)
	common assembler as	as(1)
ccb	common circuits board driver	ccb(7)
function log10 Fortran	common logarithm intrinsic	log10(3F)
routines ldfcn	common object file access	ldfcn(4)
ldopen open a	common object file for reading	ldopen(3X)
ldlread line number entries of a	common object file function	ldlread(3X)
close a	common object file ldclose	ldclose(3X)
read the file header of a	common object file ldfhread	ldfhread(3X)
number entries of a section of a	common object file ldlseek line	ldlseek(3X)
to the optional file header of a	common object file ldohseek seek	ldohseek(3X)
entries of a section of a	common object file ldrseek	ldrseek(3X)
section header of a	common object file ldshread	ldshread(3X)
to an indexed/named section of a	common object file ldsseek seek	ldsseek(3X)
of a symbol table entry of a	common object file ldtbindex	ldtbindex(3X)
indexed symbol table entry of a	common object file ldtbread an	ldtbread(3X)
seek to the symbol table of a	common object file ldtbseek	ldtbseek(3X)
line number entries in a	common object file linenum	linenum(4)
print name list of	common object file nm	nm(1)
relocation information for a	common object file reloc	reloc(4)
section header for a	common object file scnhdr	scnhdr(4)
line number information from a	common object file strip and	strip(1)
entry retrieve symbol name for	common object file symbol table	ldgetname(3X)
format syms	common object file symbol table	syms(4)
file header for	common object files filehdr	filehdr(4)

link editor for	common object files ld	ld(1)
print section sizes of	common object files size	size(1)
select or reject lines	common to two sorted files comm	comm(1)
ipcs report inter-process	communication facilities status	ipcs(1)
standard interprocess	communication package stdipc	stdipc(3C)
differential file	comparator diff	diff(1)
	compare two files cmp	cmp(1)
file sccsdiff	compare two versions of an SCCS	sccsdiff(1)
strcmp string	comparision intrinsic functions	strcmp(3F)
3-way differential file	comparison diff3	diff3(1)
directory	comparison dircmp	dircmp(1)
expression regcmp	compile and execute regular	regcmp(3X)
regular expression	compile and match routines	regexp(5)
format of	compiled term file term	term(4)
C	compiler cc	cc(1)
Fortran 77	compiler f77	f77(1)
scc C	compiler for stand-alone programs	scc(1)
terminfo	compiler tic	tic(1M)
yet another	compiler-compiler yacc	yacc(1)
modest-sized programs bs a	compiler/interpreter for	bs(1)
error function and	complementary error function	erf(3M)
await	completion of process	wait(1)
Fortran imaginary part of	complex argument	aimag(3F)
function conjg Fortran	complex conjugate intrinsic	conjg(3F)
	compress and expand files	pack(1)
table entry of a common object	compute the index of a symbol	ldtbindex(3X)
	concatenate and print files cat	cat(1)
cat	concatenate and print files	cat(8)
test	condition evaluation command	test(1)
ioctl.syscon system console	configuration file	ioctl.syscon(4)
vconfig virtual terminal	configuration	vconfig(1M)
dconfig	configure logical disks	dconfig(1M)
dconfig	configure logical disks	dconfig(8)
lpadmin	configure the LP spooling system	lpadmin(1M)
conjugate intrinsic function	conjg Fortran complex	conjg(3F)
conjg Fortran complex	conjugate intrinsic function	conjg(3F)
	connect-time accounting acctcon	acctcon(1M)
an out-going terminal line	connection dial establish	dial(3C)
L-devices link devices,	connection information	L-devices(4)
repair fsck,dfsck file system	consistency check and interactive	fsck(8)
fsck, dfsck file system	consistency check and interactive	fsck(1M)
messages from UNIX to console	conslog copy of error	conslog(7)
ioctl.syscon system	console configuration file	ioctl.syscon(4)
of error messages from UNIX to	console conslog provides copy	conslog(7)
ocw prepare	constant-width text for otroff	ocw(1)
math functions and	constants math	math(5)
	construct a file system mkfs	mkfs(1M)
mkfs	construct a file system	mkfs(8)
execute command xargs	construct argument list(s) and	xargs(1)
remove nroff/troff, tbl, and eqn	constructs deroff	deroff(1)
ls list	contents of directories	ls(8)
list	contents of directory ls	ls(1)
graphical table of	contents routines toc	toc(1)

	context split csplit	csplit(1)
interpret ASA carriage	control characters asa	asa(1)
	control device ioctl	ioctl(2)
file	control fcntl	fcntl(2)
process	control initialization init	init(1M)
message	control operations msgctl	msgctl(2)
semaphore	control operations semctl	semctl(2)
shared memory	control operations shmctl	shmctl(2)
file	control options fcntl	fcntl(5)
uucp status inquiry and job	control uustat	uustat(1)
version	control vc	vc(1)
ft IMSP streaming cartridge	controller	ft(7)
pd IMSP disk	controller	pd(7)
pt IMSP cartridge	controller	pt(7)
translate characters	conv	conv(3C)
terminals term	conventional names for	term(5)
explicit Fortran type	conversion ftype	ftype(3F)
	conversion program units	units(1)
	convert and copy a file dd	dd(1)
dd	convert and copy a file	dd(8)
common archive format	convert archive files from	arcv(1)
and long integers l3tol	convert between 3-byte integers	l3tol(3C)
base-64 ASCII string a64l	convert between long integer and	a64l(3C)
ctime	convert date and time to string	ctime(3C)
string ecvt	convert floating-point number to	ecvt(3C)
	convert formatted input scanf	scanf(3S)
strtol	convert string to integer	strtol(3C)
double-precision number strtod	convert string to	strtod(3C)
convert and	copy a file dd	dd(1)
dd convert and	copy a file	dd(8)
cpio	copy file archives in and out	cpio(1)
access time dcopy	copy file systems for optimal	dcopy(1M)
checking volcopy	copy file systems with label	volcopy(1M)
copytape make an image	copy of a tape	copytape(1)
to console conslog	copy of error messages from UNIX	conslog(7)
UNIX system to UNIX system	copy uucp	uucp(1)
public UNIX-to-UNIX system file	copy uuto	uuto(1)
link or move files, cp	copy	cp(1)
a tape	copytape make an image copy of	copytape(1)
format of	core image file core	core(4)
	core memory mem	mem(7)
cosine intrinsic function	cosh Fortran hyperbolic	cosh(3F)
Fortran hyperbolic	cosine intrinsic function cosh	cosh(3F)
Fortran	cosine intrinsic functions cos	cos(3F)
print checksum and block	count of a file sum	sum(1)
word	count wc	wc(1)
copy, link or move files	cp	cp(1)
format of	cpio archive cpio	cpio(4)
copy file archives in and out	cpio	cpio(1)
the C language preprocessor	cpp	cpp(1)
files in binary directories	cpset install object	cpset(1M)
the game of	craps craps	craps(6)
examine system images	crash	crash(1M)

to do when the system crashes	crash what	crash(8)
file or rewrite an existing one	creat create a new	creat(2)
file tmpnam	create a name for a temporary	tmpnam(3S)
existing one creat	create a new file or rewrite an	creat(2)
	create a new process fork	fork(2)
creat	create a new special file	creat(2S)
	create a temporary file tmpfile	tmpfile(3S)
pipe	create an interprocess channel	pipe(2)
admin	create and administer SCCS files	admin(1)
set and get file	creation mask umask	umask(2)
umask set and get file	creation mask	umask(2S)
driver	crm caching reel-to-reel tape	rm(7)
	cron - clock daemon cron	cron(1M)
user crontab file	crontab	crontab(1)
generate C program	cross-reference cxref	cxref(1)
files macref produce	cross-reference listing of macro	macref(1)
encode/decode	crypt	crypt(1)
generate DES encryption	crypt	crypt(3C)
interpreter) with C-like syntax	csh a shell (command	csh(1)
context split	csplit	csplit(1)
spawn getty to a remote terminal	ct	ct(1)
generate file name for terminal	ctermid	ctermid(3S)
convert date and time to string	ctime	ctime(3C)
C program debugger	ctrace	ctrace(1)
classify characters	ctype	ctype(3C)
UNIX system	cu call another	cu(1C)
activity sact print	current SCCS file editing	sact(1)
print name of	current UNIX system uname	uname(1)
get name of	current UNIX system uname	uname(2)
the slot in the utmp file of the	current user ttyslot find	ttyslot(3C)
tell report the	current value of a file pointer	tell(2S)
get path-name of	current working directory getcwd	getcwd(3C)
handling and optimization package	curses CRT screen	curses(3X)
interpolate smooth	curve spline	spline(1)
character login name of the user	cuserid get	cuserid(3S)
line of a file cut	cut out selected fields of each	cut(1)
C program cross-reference	cxref generate	cxref(1)
cron - clock	daemon cron	cron(1M)
error-logging	daemon errdemon	errdemon(1M)
terminate the error-logging	daemon errstop	errstop(1M)
run	daily accounting runacct	runacct(1M)
system backup filesave	daily/weekly UNIX system file	filesave(1M)
time a command; report process	data and system activity timex	timex(1)
port ttytype	data base of terminal types by	ttytype(5)
port ttytype	data base of terminal types by	ttytype(5)
terminal capability	data base terminfo	terminfo(4)
generate disk accounting	data by user ID diskusg diskusg	diskusg(1M)
fashion. access long integer	data in a machine-independent	sputl(3X)
lock process, text, or	data in memory plock	plock(2)
display profile	data prof	prof(1)
stat	data returned by stat system call	stat(5)
change	data segment space allocation brk	brk(2)
brk change	data segment space allocation	brk(2S)

primitive system	data types types	types(5)
relational	database operator join	join(1)
query terminfo	database tput	tput(1)
download program files	dataio	dataio(1M)
convert	date and time to string ctime	ctime(3C)
print and set the date	date	date(1)
desk calculator	dc	dc(1)
lpadmin	configure the LP spooling system	lpadmin(1M)
configure logical disks	dconfig	dconfig(1M)
configure logical disks	dconfig	dconfig(8)
systems for optimal access time	dcopy copy file	dcopy(1M)
	dd convert and copy a file	dd(8)
convert and copy a file	dd	dd(1)
absolute	debugger adb	adb(1)
C program	debugger ctrace	ctrace(1)
file system	debugger fsdb	fsdb(1M)
fsdb file system	debugger	fsdb(8)
eqnchar special character	definitions for eqn and neqn	eqnchar(5)
basename	deliver portions of path names	basename(1)
tail	deliver the last part of a file	tail(1)
delta make a	delta (change) to an SCCS file	delta(1)
the delta commentary of an SCCS	delta cdc change	cdc(1)
cdc change the	delta commentary of an SCCS delta	cdc(1)
remove a	delta from an SCCS file rmdel	rmdel(1)
a delta (change) to an SCCS file	delta make	delta(1)
combine SCCS	deltas comb	comb(1)
permit or	deny messages mesg	mesg(1)
tbl, and eqn constructs	deroff remove nroff/troff,	deroff(1)
device-independent troff font	description files for	font(5)
troff	description of output language	troff(5)
close a file	descriptor close	close(2)
duplicate an open file	descriptor dup	dup(2)
returns a 1 if specified file	descriptor is a isatty	isatty(2S)
close close a file	descriptor	close(2S)
	desk calculator dc	dc(1)
access	determine accessibility of a file	access(2)
access	determine accessibility of a file	access(2S)
	determine file type file	file(1)
control	device ioctl	ioctl(2)
	device name devnm	devnm(1M)
graphical	device routines and filters gdev	gdev(1)
font description files for	device-independent troff	font(5)
L-devices link	devices, connection information	L-devices(4)
device name	devnm	devnm(1M)
report number of free disk blocks	df	df(1M)
	dformat disk formatter	dformat(8)
consistency check and interactive, fsck	dfsck file system	fsck(8)
consistency check and interactive, fsck	dfsck file system	fsck(1M)
dial	dial a Racal-Vadic 3451 modem	dial(1)
mdial	dial the P/75 onboard modem	dial(1)
terminal line connection	dial establish an out-going	dial(3C)
rational Fortran	dialect ratfor	ratfor(1)
L-dialcodes alphabetic	dialing abbreviations file	L-dialcodes(4)

d_passwd	dialup password file	d_passwd(4)
dialups, list of	dialup devices	dialups(4)
big	diff bdiff	bdiff(1)
differential file comparator	diff	diff(1)
differential file comparison	diff3 3-way	diff3(1)
dim positive	difference intrinsic functions	dim(3F)
side-by-side	difference program sdiff	sdiff(1)
mark	differences between files diffmk	diffmk(1)
	differential file comparator diff	diff(1)
diff3 3-way	differential file comparison	diff3(1)
mark differences between files	diffmk	diffmk(1)
difference intrinsic functions	dim positive	dim(3F)
format of directories	dir	dir(4)
directory comparison	dircmp	dircmp(1)
install object files in binary	directories cpset	cpset(1M)
format of	directories dir	dir(4)
remove files or	directories rm	rm(1)
ls list contents of	directories	ls(8)
change working	directory cd	cd(1)
change working	directory chdir	chdir(2)
change root	directory chroot	chroot(2)
uucp spool	directory clean-up uuclean	uuclean(1M)
	directory comparison dircmp	dircmp(1)
remove	directory entry unlink	unlink(2)
change root	directory for a command chroot	chroot(1M)
get path-name of current working	directory getcwd	getcwd(3C)
list contents of	directory ls	ls(1)
make a	directory mkdir	mkdir(1)
move a	directory mvdir	mvdir(1M)
working	directory name pwd	pwd(1)
ordinary file mknod make a	directory or a special or	mknod(2)
chdir change working	directory	chdir(2S)
enable or	disable process accounting acct	acct(2)
type, modes, speed, and line	discipline getty set terminal	getty
od optical	disk	od(7)
rod raw interface to optical	disk	od(7)
	disk access profiler sadp	sadp(1M)
diskusg diskusg - generate	disk accounting data by user ID	diskusg(1M)
report number of free	disk blocks df	df(1M)
pd IMSP	disk controller	pd(7)
dsk optionally mirrored	disk driver	dsk(7)
dformat	disk formatter	dformat(8)
summarize	disk usage du	du(1)
du summarize	disk usage	du(8)
ramdisk memory as	disk	ramdisk(1M)
allows memory to be used as a	disk rram	rram(7)
configure logical	disks	dconfig(1M)
configure logical	disks	dconfig(8)
accounting data by user ID	diskusg - generate disk	diskusg(1M)
mount and	dismount file system mount	mount(1M)
screen-oriented (visual)	display editor based on ex vi	vi(1)
	display profile data prof	prof(1)
Euclidean	distance function hypot	hypot(3M)

drand48 generate uniformly	distributed pseudo-random numbers	drand48(3C)
download program files	dataio	dataio(1M)
download program files	dnld	dnld(1M)
mmlint sroff/MM nroff/MM	document compatibility checker	mmlint(1)
surface characteristics of a	document analyze	style(1)
macros mm print/check	documents formatted with the MM ...	mm(1)
MM macro package for formatting	documents mm the	mm(5)
macro package for formatting	documents the OSDD adapter	mosd(5)
mmt typeset	documents, viewgraphs, and slides .	mmt(1)
who is	doing what whodo	whodo(1M)
intrinsic function dprod	double precision product	dprod(3F)
float and	double routines	float(2S)
convert string to	double-precision number strtod	strtod(3C)
dnld	download program files	dnld(1M)
	d_passwd dialup password file	d_passwd(4)
check and interactive fsck,	dfsck file system consistency	fsck(1M)
product intrinsic function	dprod double precision	dprod(3F)
distributed pseudo-random numbers	drand48 generate uniformly	drand48(3C)
	draw a graph graph	graph(1)
pic troff preprocessor for	drawing simple pictures	pic(1)
provide	drill in number facts arithmetic	arithmetic(6)
ccb common circuits board	driver	ccb(7)
dsk optionally mirrored disk	driver	dsk(7)
crm caching reel-to-reel tape	driver	rm(7)
rrm raw reel-to-reel tape	driver	rm(7)
driver	dsk optionally mirrored disk	dsk(7)
raw (unbuffered) version of	dsk, rdsk	dsk(7)
	du summarize disk usage	du(8)
summarize disk usage	du	du(1)
Communication acpdmp	dump contents of Advanced	acpdmp(1M)
Communication Processor's	dump contents of VMEbus	vcpdmp(1M)
Intelligent Communication icpdmp	dump contents of an	icpdmp(1M)
extract error records from	dump errdead	errdead(1M)
octal	dump od	od(1)
object file dump	dump selected parts of an	dump(1)
dump incremental	dump tape format	dump(4)
od octal	dump	od(8)
selected parts of an object file	dump dump	dump(1)
duplicate an open file descriptor	dup	dup(2)
for the Xerox 9700 printer	dx9700 prepare troff documents	dx9700(1)
	echo arguments echo	echo(1)
floating-point number to string	ecvt convert	ecvt(3C)
text editor	ed	ed(1)
(variant of ex for casual users)	edit text editor	edit(1)
print current SCCS file	editing activity sact	sact(1)
users edit text	editor (variant of ex for casual	edit(1)
screen-oriented (visual) display	editor based on ex vi	vi(1)
text	editor ex	ex(1)
link	editor for common object files ld	ld(1)
graphic	editor ged	ged(1)
common assembler and link	editor output a.out	a.out(4)
stream	editor sed	sed(1)
split f77, ratfor, or	efl files fsplit	fsplit(1)

Extended Fortran Language	efl	efl(1)
accounting acct	enable or disable process	acct(2)
enable	enable/disable LP printers	enable(1)
for/ uuencode,uudecode	encode/decode a binary file	uuencode(1c)
	encode/decode crypt	crypt(1)
generate DES	encryption crypt	crypt(3C)
generate	encryption key makekey	makekey(1)
lst iocations in program	end	end(3C)
get	entries from name list nlist	nlist(3C)
linenum line number	entries in a common object file	linenum(4)
print	entries in this manual man	man(1)
manipulate line number	entries of a common object file	ldlread(3X)
object file seek to relocation	entries of a section of a common	ldrseek(3X)
utmp and wtmp	entry formats utmp	utmp(4)
get group file	entry getgrent	getgrent(3C)
get password file	entry getpwent	getpwent(3C)
access utmp file	entry getut	getut(3C)
common object file symbol table	entry ldgetname symbol name for	ldgetname(3X)
the index of a symbol table	entry of a common object file	ldtbindex(3X)
read an indexed symbol table	entry of a common object file	ldtbread(3X)
write password file	entry putpwent	putpwent(3C)
remove directory	entry unlink	unlink(2)
environment for command execution	env set	env(1)
user environment	environ	environ(5)
setting up an	environment at login time	profile(4)
profile setting up an	environment at login time	profile(5)
user	environment environ	environ(5)
env set	environment for command execution	env(1)
return value for	environment name	getenv(3C)
change or add value to	environment putenv	putenv(3C)
return Fortran	environment variable getenv	getenv(3F)
special character definitions for	eqn and neqn eqnchar	eqnchar(5)
remove nroff/troff, tbl, and	eqn constructs deroff	deroff(1)
for nroff or troff	eqn format mathematical text	eqn(1)
and complementary error function	erf error function	erf(3M)
error-logging interface	err	err(7)
extract error records from dump	errdead	errdead(1M)
error-logging daemon	errdemon	errdemon(1M)
error-log file format	errfile	errfile(4)
complementary error function erf	error function and	erf(3M)
copy of	error messages from UNIX to	conslog(7)
system	error messages perror	perror(3C)
extract	error records from dump errdead	errdead(1M)
	error-handling function matherr	matherr(3M)
	error-log file format errfile	errfile(4)
	error-logging daemon errdemon	errdemon(1M)
terminate the	error-logging daemon errstop	errstop(1M)
	error-logging interface err	err(7)
process a report of logged	errors errpt	errpt(1M)
find spelling	errors spell	spell(1)
process a report of logged errors	errpt	errpt(1M)
the error-logging daemon	errstop terminate	errstop(1M)
line connection dial	establish an out-going terminal	dial(3C)

	establish mount table setmnt	setmnt(1M)
expression expr	evaluate arguments as an	expr(1)
test condition	evaluation command	test(1)
text editor (variant of	ex for casual users) edit	edit(1)
(visual) display editor based on	ex vi screen-oriented	vi(1)
	examine system images crash	crash(1M)
reading or lockf provide	exclusive file regions for	lockf(2)
execute a file	exec	exec(2)
network net	execute a command on the PCL	net(1)
	execute a file exec	exec(2)
construct argument list(s) and	execute command xargs	xargs(1)
at	execute commands at a later time	at(1)
compile and	execute regular expression	regcmp(3X)
uuxqt	execute remote command requests	uuxqt(1M)
L.cmds remote	execution commands	L.cmds(4)
set environment for command	execution env	env(1)
suspend	execution for interval sleep	sleep(3C)
prepare	execution profile monitor	monitor(3C)
	execution time profile profil	profil(2)
UNIX-to-UNIX system command	execution uux	uux(1)
calls link	exercise link and unlink system	link(1M)
create a new file or rewrite an	existing one creat	creat(2)
	exit terminate process	exit(2S)
terminate process	exit	exit(2)
EOT on the other terminal and	exits. write	write(1)
exponential intrinsic function	exp Fortran	exp(3F)
power, square root function	exp exponential, logarithm,	exp(3M)
compress and	expand files pack	pack(1)
conversion ftype	explicit Fortran type	ftype(3F)
exp Fortran	exponential intrinsic function	exp(3F)
square root function exp	exponential, logarithm, power,	exp(3M)
arguments as an expression	expr evaluate	expr(1)
routines regexp regular	expression compile and match	regexp(5)
regular	expression compile regcmp	regcmp(1)
evaluate arguments as an	expression expr	expr(1)
compile and execute regular	expression regcmp	regcmp(3X)
errdead	extract error records from dump	errdead(1M)
Fortran 77 compiler	f77	f77(1)
split	f77, ratfor, or efl files fsplit	fsplit(1)
inter-process communication	facilities status ipcs report	ipcs(1)
	factor a number factor	factor(1)
provide drill in number	facts arithmetic	arithmetic(6)
data in a machine-independent	fashion. sputl long integer	sputl(3X)
	fast incremental backup finc	finc(1M)
	fast main memory allocator	malloc(3X)
fbackup	fast tape backup of a file system	fbackup(1M)
fbackup make a	fast tape backup of a file system	fbackup(8)
procedure checkall	faster file system checking	checkall(1M)
generate an IOT	fault abort	abort(3C)
backup of a file system	fbackup make a fast tape	fbackup(1M)
backup of a file system	fbackup make a fast tape	fbackup(8)
close or flush a stream	fclose	fclose(3S)
file control options	fcntl	fcntl(5)

stream status inquiries	ferror	ferror(3S)
and statistics for a file system	ff list file names	ff(1M)
cut out selected	fields of each line of a file cut	cut(1)
times utime set	file access and modification	utime(2)
common object	file access routines ldfcn	ldfcn(4)
determine accessibility of a	file access	access(2)
tape	file archiver tar	tar(1)
copy	file archives in and out cpio	cpio(1)
password/group	file checkers pwck	pwck(1M)
change mode of	file chmod	chmod(2)
change owner and group of a	file chown	chown(2)
differential	file comparator diff	diff(1)
3-way differential	file comparison diff3	diff3(1)
	file control fcntl	fcntl(2)
	file control options fcntl	fcntl(5)
public UNIX-to-UNIX system	file copy uuto	uuto(1)
format of core image	file core	core(4)
set and get	file creation mask umask	umask(2)
umask set and get	file creation mask	umask(2S)
user crontab	file crontab	crontab(1)
selected fields of each line of a	file cut cut out	cut(1)
convert and copy a	file dd	dd(1)
make a delta (change) to an SCCS	file delta	delta(1)
close a	file descriptor close	close(2)
duplicate an open	file descriptor dup	dup(2)
returns a 1 if specified	file descriptor is a isatty	isatty(2S)
close close a	file descriptor	close(2S)
dump selected parts of an object	file dump	dump(1)
print current SCCS	file editing activity sact	sact(1)
get group	file entry getgrent	getgrent(3C)
get password	file entry getpwent	getpwent(3C)
access utmp	file entry getut	getut(3C)
write password	file entry putpwent	putpwent(3C)
execute a	file exec	exec(2)
search a	file for a pattern grep	grep(1)
open a common object	file for reading ldopen	ldopen(3X)
per-process accounting	file format acct	acct(4)
common archive	file format ar	ar(4)
error-log	file format errfile	errfile(4)
number entries of a common object	file function ldlread line	ldlread(3X)
get a version of a SCCS	file get	get(1)
group	file group	group(4)
files filehdr	file header for common object	filehdr(4)
file ldfhread read the	file header of a common object	ldfhread(3X)
file seek to the optional	file header of a common object	ldohseek(3X)
split a	file into pieces split	split(1)
issue identification	file issue	issue(4)
header of a member of an archive	file ldahread read the archive	ldahread(3X)
close a common object	file ldclose	ldclose(3X)
file header of a common object	file ldfhread read the	ldfhread(3X)
of a section of a common object	file ldlseek line number entries	ldlseek(3X)
file header of a common object	file ldohseek to the optional	ldohseek(3X)
of a section of a common object	file ldrseek relocation entries	ldrseek(3X)

section header of a common object	file ldshread an indexed/named	ldshread(3X)
section of a common object	file ldsseek an indexed/named	ldsseek(3X)
table entry of a common object	file ldtbindex index of a symbol	ldtbindex(3X)
table entry of a common object	file ldtbread an indexed symbol	ldtbread(3X)
symbol table of a common object	file ldtbseek seek to the	ldtbseek(3X)
number entries in a common object	file linenum line	linenum(4)
link to a	file link	link(2)
build special	file mknod	mknod(1M)
or a special or ordinary	file mknod make a directory	mknod(2)
generate	file name for terminal ctermid	ctermid(3S)
make a unique	file name mktemp	mktemp(3C)
file system ff list	file names and statistics for a	ff(1M)
change the format of a text	file newform	newform(1)
print name list of common object	file nm	nm(1)
the null	file null	null(7)
find the slot in the utmp	file of the current user ttyslot	ttyslot(3C)
identify processes using a	file or file structure fuser	fuser(1M)
creat create a new	file or rewrite an existing one	creat(2)
password	file passwd	passwd(4)
files or subsequent lines of one	file paste same lines of several	paste(1)
more, page	file perusal filter for crt viewing	more(1)
terminals pg	file perusal filter for soft-copy	pg(1)
reposition a	file pointer in a stream fseek	fseek(3S)
move read/write	file pointer lseek	lseek(2)
lseek move read/write	file pointer	lseek(2S)
report the current value of a	file pointer tell	tell(2S)
print an SCCS	file prs	prs(1)
read from	file read	read(2)
lockf provide exclusive	file regions for reading or	lockf(2)
information for a common object	file reloc relocation	reloc(4)
remove a delta from an SCCS	file rmdel	rmdel(1)
big	file scanner bfs	bfs(1)
compare two versions of an SCCS	file sccsdiff	sccsdiff(1)
format of SCCS	file sccsfile	sccsfile(4)
header for a common object	file scnhdr section	scnhdr(4)
get	file status stat	stat(2)
stat get	file status	stat(2S)
information from a common object	file strip and line number	strip(1)
processes using a file or	file structure fuser identify	fuser(1M)
checksum and block count of a	file sum print	sum(1)
symbol name for common object	file symbol table entry ldgetname	ldgetname(3X)
common object	file symbol table format syms	syms(4)
daily/weekly UNIX system	file system backup filesave	filesave(1M)
checkall faster	file system checking procedure	checkall(1M)
interactive fsck, dfsck	file system consistency check and	fsck(1M)
and interactive fsck, dfsck	file system consistency check	fsck(8)
	file system debugger fsdb	fsdb(1M)
fsdb	file system debugger	fsdb(8)
file names and statistics for a	file system ff list	ff(1M)
construct a	file system mkfs	mkfs(1M)
mount and dismount	file system mount	mount(1M)
mount a	file system mount	mount(2)
incremental	file system restore	restor(1M)

restor incremental	file system restore	restor(8)
get	file system statistics ustat	ustat(2)
ustat get	file system statistics	ustat(2S)
mounted	file system table mnttab	mnttab(4)
unmount a	file system umount	umount(2)
mount a	file system	mount(2S)
unmount a	file system	umount(2S)
make a fast tape backup of a	file system fbackup	fbackup(1M)
time dcopy copy	file systems for optimal access	dcopy(1M)
checklist list of	file systems processed by fsck	checklist(4)
volcopy copy	file systems with label checking	volcopy(1M)
deliver the last part of a	file tail	tail(1)
create a temporary	file tmpfile	tmpfile(3S)
create a name for a temporary	file tmpnam	tmpnam(3S)
and modification times of a	file touch update access	touch(1)
uucp system uucico	file transport program for	uucico
walk a	file tree ftw	ftw(3C)
determine	file type file	file(1)
undo a previous get of an SCCS	file unget	unget(1)
report repeated lines in a	file uniq	uniq(1)
validate SCCS	file val	val(1)
write on a	file write	write(2)
creat create a new special	file	creat(2S)
determine file type	file	file(1)
mknod make a special	file	mknod(2S)
read read from	file	read(2S)
write on a	file	write(2S)
format of compiled term	file term	term(4)
and print process accounting	file(s) acctcom search	acctcom(1)
set	file-creation mode mask umask	umask(1)
header for common object files	filehdr file	filehdr(4)
dataio download program	files	dataio(1M)
merge or add total accounting	files acctmerg	acctmerg(1M)
create and administer SCCS	files admin	admin(1)
concatenate and print	files cat	cat(1)
compare two	files cmp	cmp(1)
reject lines common to two sorted	files comm select or	comm(1)
copy, link or move	files cp	cp(1)
mark differences between	files diffmk	diffmk(1)
file header for common object	files filehdr	filehdr(4)
recover	files from a backup tape frec	frec(1M)
format specification in text	files fspec	fspec(4)
split f77, ratfor, or efl	files fsplit	fsplit(1)
string, format of graphical	files gps graphical primitive	gps(4)
install object	files in binary directories cpset	cpset(1M)
link editor for common object	files ld	ld(1)
remove	files or directories rm	rm(1)
merge same lines of several	files or subsequent lines of one	paste(1)
compress and expand	files pack	pack(1)
print	files pr	pr(1)
section sizes of common object	files size print	size(1)
sort and/or merge	files sort	sort(1)
identify SCCS	files what	what(1)

archive format convert archive	files	arcv(1)
UNIX system file system backup	filesave daily/weekly	filesave(1M)
more, page file perusal	filter for crt viewing	more(1)
file perusal	filter for soft-copy terminals pg	pg(1)
select terminal	filter greek	greek(1)
line numbering	filter nl	nl(1)
	filter reverse line-feeds col	col(1)
graphical device routines and	filters gdev	gdev(1)
graphics	filters tplot	tplot(1)
fast incremental backup	finc	finc(1M)
	find hyphenated words hyphen	hyphen(1)
	find name of a terminal ttyname	ttyname(3C)
object library lorder	find ordering relation for an	lorder(1)
	find spelling errors spell	spell(1)
an object, or other	find the printable strings in	strings(1)
the current user ttyslot	find the slot in the utmp file of	ttyslot(3C)
find files	find	find(1)
	float and double routines	float(2S)
ecvt convert	floating-point number to string	ecvt(3C)
manipulate parts of	floating-point numbers frexp	frexp(3S)
absolute value functions floor	floor, ceiling, remainder,	floor(3M)
generate C	flow graph cflow	cflow(1)
close or	flush a stream fclose	fclose(3S)
open a stream	fopen	fopen(3S)
create a new process	fork	fork(2)
per-process accounting file	format acct	acct(4)
common archive file	format ar	ar(4)
common archive	format arcv archive files	arcv(1)
error-log file	format errfile	errfile(4)
nroff or troff eqn	format mathematical text for	eqn(1)
	format of SCCS file sccsfile	sccsfile(4)
change the	format of a text file newform	newform(1)
	format of an i-node inode	inode(4)
term	format of compiled term file	term(4)
	format of core image file core	core(4)
	format of cpio archive cpio	cpio(4)
	format of directories dir	dir(4)
graphical primitive string,	format of graphical files gps	gps(4)
	format of system volume fs	fs(4)
nroff	format or typeset text	nroff(1)
files fspec	format specification in text	fspec(4)
common object file symbol table	format syms	syms(4)
tbl	format tables for nroff or troff	tbl(1)
sroff	format text	sroff(1)
utmp and wtmp entry	formats utmp	utmp(4)
convert	formatted input scanf	scanf(3S)
argument list vprintf print	formatted output of a varargs	vprintf(3S)
print	formatted output printf	printf(3S)
dformat disk	formatter	dformat(8)
mptx the macro package for	formatting a permuted index	mptx(5)
troff text	formatting and typesetting	troff(1)
mm the MM macro package for	formatting documents	mm(5)
OSDD adapter macro package for	formatting documents mosd the	mosd(5)

man macros for	formatting entries in this manual	man(5)
binary input/output	fread	fread(3S)
recover files from a backup tape	frec	frec(1M)
report number of	free disk blocks df	df(1M)
parts of floating-point numbers	frexp manipulate	frexp(3S)
format of system volume	fs	fs(4)
list of file systems processed by	fsck checklist	checklist(4)
check and interactive repair	fsck, dfsck file system consistency ...	fsck(8)
check and interactive repair	fsck, dfsck file system consistency ...	fsck(1M)
	fsdb file system debugger	fsdb(8)
file system debugger	fsdb	fsdb(1M)
a file pointer in a stream	fseek reposition	fseek(3S)
specification in text files	fspec format	fspec(4)
split f77, ratfor, or efl files	fsplit	fsplit(1)
controller	ft IMSP streaming cartridge	ft(7)
walk a file tree	ftw	ftw(3C)
explicit Fortran type conversion	ftype	ftype(3F)
Fortran arcsine intrinsic	functio asin	asin(3F)
Fortran arccosine intrinsic	function acos	acos(3F)
Fortran integer part intrinsic	function aint	aint(3F)
function error	function and complementary error ...	erf(3M)
Fortran arctangert intrinsic	function atan	atan(3F)
Fortran arctangent intrinsic	function atan2	atan2(3F)
complex conjugate intrinsic	function conjg Fortran	conjg(3F)
hyperbolic cosine intrinsic	function cosh Fortran	cosh(3F)
precision product intrinsic	function dprod double	dprod(3F)
function and complementary error	function erf error	erf(3M)
Fortran exponential intrinsic	function exp	exp(3F)
logarithm, power, square root	function exp exponential, ...	exp(3M)
log gamma	function gamma	gamma(3M)
Euclidean distance	function hypot	hypot(3M)
entries of a common object file	function ldlread line number	ldlread(3X)
natural logarithm intrinsic	function log Fortran	log(3F)
common logarithm intrinsic	function log10 Fortran	log10(3F)
error-handling	function matherr	matherr(3M)
profile within a	function prof	prof(5)
transfer-of-sign intrinsic	function sign Fortran	sign(3F)
Fortran sine intrinsic	function sin	sin(3F)
Fortran hyperbolic sine intrinsic	function sinh	sinh(3F)
Fortran square root intrinsic	function sqrt	sqrt(3F)
Fortran tangent intrinsic	function tan	tan(3F)
hyperbolic tangent intrinsic	function tanh Fortran	tanh(3F)
math	functions and constants math	math(5)
Fortran bitwise boolean	functions bool	bool(3F)
Fortran cosine intrinsic	functions cos	cos(3F)
positive difference intrinsic	functions dim	dim(3F)
remainder, absolute value	functions floor floor, ceiling, ...	floor(3M)
Fortran maximum-value	functions max	max(3F)
Fortran minimum-value	functions min	min(3F)
Fortran remaindering intrinsic	functions mod	mod(3F)
terminals 300 handle special	functions of DASI 300 and 300s	300(1)
2621-series handle special	functions of HP 2640 and	hp(1)
terminal 450 handle special	functions of the DASI 450	450(1)

Fortran nearest integer	functions round	round(3F)
hyperbolic	functions sinh	sinh(3M)
string comparision intrinsic	functions strcmp	strcmp(3F)
trigonometric	functions trig	trig(3M)
using a file or file structure	fuser identify processes	fuser(1M)
guessing	game moo	moo(6)
the	game of backgammon back	back(6)
the	game of black jack bj	bj(6)
the	game of craps craps	craps(6)
the	game of hunt-the-wumpus wump	wump(6)
log gamma function	gamma	gamma(3M)
device routines and filters	gdev graphical	gdev(1)
graphic editor	ged	ged(1)
tty	general terminal interface	tty(7)
	generate C flow graph cflow	cflow(1)
cross-reference cxref	generate C program	cxref(1)
	generate DES encryption crypt	crypt(3C)
	generate a maze maze	maze(6)
	generate an IOT fault abort	abort(3C)
user ID diskusg diskusg	generate disk accounting data by	diskusg(1M)
	generate encryption key makekey	makekey(1)
ctermid	generate file name for terminal	ctermid(3S)
ncheck	generate names from i-numbers	ncheck(1M)
lexical tasks lex	generate programs for simple	lex(1)
pseudo-random numbers drand48	generate uniformly distributed	drand48(3C)
simple random-number	generator rand	rand(3C)
random number	generator rand	rand(3F)
	get a string from a stream gets	gets(3S)
get	get a version of a SCCS file	get(1)
	get and set user limits ulimit	ulimit(2)
user cuserid	get character login name of the	cuserid(3S)
stream getc	get character or word from a	getc(3S)
nlist	get entries from name list	nlist(3C)
	get file status stat	stat(2)
stat	get file status	stat(2S)
ustat	get file system statistics	ustat(2)
ustat	get file system statistics	ustat(2S)
	get group file entry getgrent	getgrent(3C)
	get login name getlogin	getlogin(3C)
	get login name logname	logname(1)
	get message queue msgget	msgget(2)
	get name from UID getpw	getpw(3C)
uname	get name of current UNIX system	uname(2)
vector getopt	get option letter from argument	getopt(3C)
getpwent	get password file entry	getpwent(3C)
working directory getcwd	get path-name of current	getcwd(3C)
getpid	get process ID	getpid(2S)
times	get process and child process	times(2)
parent process IDs getpid	get process, process group, and	getpid(2)
and real and effective getuid	get real and effective user,	getuid(2S)
real group, and effective group	get real user, effective user,	getuid(2)
	get set of semaphores semget	semget(2)
shmget	get shared memory segment	shmget(2)

gtty	get terminal characterisitcs	gtty(2S)
tty	get the name of the terminal	tty(1)
	get time time	time(2)
time	get time	time(2S)
Fortran command-line argument	getarc return	getarc(3F)
and get arguments for	getargv display a program name	getargv(2S)
character or word from a stream	getc get	getc(3S)
of current working directory	getcwd get path-name	getcwd(3C)
return value for environment name	getenv	getenv(3C)
Fortran environment variable	getenv return	getenv(3F)
get group file entry	getgrent	getgrent(3C)
get login name	getlogin	getlogin(3C)
parse command options	getopt	getopt(1)
letter from argument vector	getopt get option	getopt(3C)
read a password	getpass	getpass(3C)
	getpid get process ID	getpid(2S)
group, and parent process IDs	getpid get process, process	getpid(2)
get name from UID	getpw	getpw(3C)
get password file entry	getpwent	getpwent(3C)
get a string from a stream	gets	gets(3S)
spawn	getty to a remote terminal ct	ct(1)
terminal settings used by getty	gettydefs speed and	gettydefs(4)
user, and real and effective	getuid get real and effective	getuid(2S)
group, and effective group ID's	getuid effective user, real	getuid(2)
access utmp file entry	getut	getut(3C)
setjmp, longjmp non-local	goto	setjmp(3C)
string, format of graphical files	gps graphical primitive	gps(4)
generate C flow	graph cflow	cflow(1)
system activity	graph sag	sag(1)
draw a graph	graph	graph(1)
	graphic editor ged	ged(1)
graphics access	graphical and numerical commands	graphics(1)
statistical network useful with	graphical commands stat	stat(1)
filters gdev	graphical device routines and	gdev(1)
primitive string, format of	graphical files gps graphical	gps(4)
routines toc	graphical table of contents	toc(1)
	graphical utilities gutil	gutil(1)
	graphics filters tplot	tplot(1)
	graphics interface plot	plot(4)
plot	graphics interface subroutines	plot(3X)
graphical and numerical commands	graphics access	graphics(1)
a macro package for making view	graphs mv	mv(7)
select terminal filter	greek	greek(1)
search a file for a pattern	grep	grep(1)
set process	group ID setpgrp	setpgrp(2)
user, and real and effective	group ID get real and effective	getuid(2S)
user, real group, and effective	group ID's getuid effective	getuid(2)
print user and	group IDs and names id	id(1)
set user and	group IDs setuid	setuid(2)
change owner or	group chown	chown(1)
get	group file entry getgrent	getgrent(3C)
	group file group	group(4)
log in to a new	group newgrp	newgrp(1)

change owner and	group of a file chown	chown(2)
send a signal to a process or a	group of processes kill	kill(2)
group file	group	group(4)
real user, effective user, real	group, and effective group ID's	getuid(2)
getpid get process, process	group, and parent process IDs	getpid(2)
maintain, update, and regenerate	groups of programs make	make(1)
characterisitcs	gtty get terminal	gtty(2S)
	guess the word hangman	hangman(6)
	guessing game moo	moo(6)
graphical utilities	gutil	gutil(1)
300 and 300s terminals 300	handle special functions of DASI	300(1)
2640 and 2621-series terminals	handle special functions of HP	hp(1)
DASI 450 terminal 450	handle special functions of the	450(1)
varargs	handle variable argument list	varargs(5)
curses CRT screen	handling and optimization package	curses(3X)
guess the word	hangman	hangman(6)
run a command immune to	hangups and quits	nohup(1)
manage	hash search tables hsearch	hsearch(3C)
scnhdr section	header for a common object file	scnhdr(4)
filehdr file	header for common object files	filehdr(4)
ldfhread read the file	header of a common object file	ldfhread(3X)
seek to the optional file	header of a common object file	ldohseek(3X)
read an indexed/named section	header of a common object file	ldshread(3X)
file ldahread read the archive	header of a member of an archive	ldahread(3X)
ask for help	help	help(8)
ask for help	help	help(1)
HP 2640 and 2621-series terminals	hp handle special functions of	hp(1)
manage hash search tables	hsearch	hsearch(3C)
the game of	hunt-the-wumpus wump	wump(6)
function cosh Fortran	hyperbolic cosine intrinsic	cosh(3F)
	hyperbolic functions sinh	sinh(3M)
function sinh Fortran	hyperbolic sine intrinsic	sinh(3F)
function tanh Fortran	hyperbolic tangent intrinsic	tanh(3F)
find hyphenated words	hyphen	hyphen(1)
Euclidean distance function	hypot	hypot(3M)
	iargc iargc	iargc(3F)
clear	i-node clri	clri(1M)
format of an	i-node inode	inode(4)
generate names from	i-numbers ncheck	ncheck(1M)
Processor	icp Intelligent Communications	icp(7)
Intelligent Communication	icpdmp dump contents of an	icpdmp(1M)
semaphore set or shared memory	id ipcrm remove a message queue,	ipcrm(1)
user and group IDs and names	id print	id(1)
issue	identification file issue	issue(4)
	identify SCCS files what	what(1)
or file structure fuser	identify processes using a file	fuser(1M)
format of core	image file core	core(4)
make an	image copy of a tape	copytape(1)
	core memory mem	mem(7)
examine system	images crash	crash(1M)
argument aimag Fortran	imaginary part of complex	aimag(3F)
run a command	immune to hangups and quits	nohup(1)
Processor	imsp Intelligent Mass Storage	imsp(7)

fast	incremental backup finc	finc(1M)
dump	incremental dump tape format	dump(4)
	incremental file system restore	restor(1M)
restor	incremental file system restore	restor(8)
termlib terminal	independent operation routines	termlib(3C)
a common object compute the	index of a symbol table entry of	ldtbindex(3X)
permuted	index ptx	ptx(1)
package for formatting a permuted	index mptx the macro	mptx(5)
location of Fortran substring	index return	index(3F)
common object file read an	indexed symbol table entry of a	ldtbread(3X)
a common object file read an	indexed/named section header of	ldshread(3X)
common object file seek to an	indexed/named section of a	ldsseek(3X)
file reloc relocation	information for a common object	reloc(4)
strip symbol and line number	information from a common object	strip(1)
print LP status	information lpstat	lpstat(1)
process control initialization		init(1M)
process control	initialization init	init(1M)
system	initialization shell scripts brc	brc(1M)
popen	initiate pipe to/from a process	popen(3S)
script for the init process	inittab	inittab(4)
format of an i-node	inode	inode(4)
convert formatted	input scanf	scanf(3S)
push character back into	input stream ungetc	ungetc(3S)
binary	input/output fread	fread(3S)
standard buffered	input/output package stdio	stdio(3S)
stream status	inquiries ferror	ferror(3S)
uucp status	inquiry and job control uustat	uustat(1)
directories cpset	install object files in binary	cpset(1M)
install commands	install	install(1M)
return	integer absolute value abs	abs(3C)
a64l convert between long	integer and base-64 ASCII string	a64l(3C)
access long	integer data in a	sputl(3X)
Fortran nearest	integer functions round	round(3F)
aint Fortran	integer part intrinsic function	aint(3F)
convert string to	integer strtol	strtol(3C)
convert between 3-byte	integers and long integers l3tol	l3tol(3C)
facilities status ipcs report	inter-process communication	ipcs(1)
system mailx	interactive message processing	mailx(1)
file system consistency check and	interactive repair fsck	fsck(8)
file system consistency check	interactive repair fsck, dfsck	fsck(1M)
tty general terminal	interface	tty(7)
error-logging	interface err	err(7)
graphics	interface plot	plot(4)
graphics	interface subroutines plot	plot(3X)
pp parallel port	interface	pp(7)
	interpolate smooth curve spline	spline(1)
characters asa	interpret ASA carriage control	asa(1)
SNOBOL	interpreter sno	sno(1)
tc troff output	interpreter	tc(1)
create an	interprocess channel pipe	pipe(2)
package stdipc standard	interprocess communication	stdipc(3C)
suspend execution for an	interval sleep	sleep(1)
suspend execution for	interval sleep	sleep(3C)

Fortran arcsine	intrinsic functio asin	asin(3F)
Fortran arccosine	intrinsic function acos	acos(3F)
Fortran integer part	intrinsic function aint	aint(3F)
Fortran arctangert	intrinsic function atan	atan(3F)
Fortran arctangent	intrinsic function atan2	atan2(3F)
Fortran complex conjugate	intrinsic function conjg	conjg(3F)
Fortran hyperbolic cosine	intrinsic function cosh	cosh(3F)
double precision product	intrinsic function dprod	dprod(3F)
Fortran exponential	intrinsic function exp	exp(3F)
Fortran natural logarithm	intrinsic function log	log(3F)
Fortran common logarithm	intrinsic function log10	log10(3F)
Fortran transfer-of-sign	intrinsic function sign	sign(3F)
Fortran sine	intrinsic function sin	sin(3F)
Fortran hyperbolic sine	intrinsic function sinh	sinh(3F)
Fortran square root	intrinsic function sqrt	sqrt(3F)
Fortran tangent	intrinsic function tan	tan(3F)
Fortran hyperbolic tangent	intrinsic function tanh	tanh(3F)
Fortran cosine	intrinsic functions cos	cos(3F)
positive difference	intrinsic functions dim	dim(3F)
Fortran remaindering	intrinsic functions mod	mod(3F)
string comparision	intrinsic functions strcmp	strcmp(3F)
control device	ioctl	ioctl(2)
configuration file	ioctl.syscon system console	ioctl.syscon(4)
semaphore set or shared memory id	ipcrm remove a message queue,	ipcrm(1)
communication facilities status	ipcs report inter-process	ipcs(1)
specified file descriptor is a	isatty returns a 1 if	isatty(2S)
Fortran system	issue a shell command from	system(3F)
	issue a shell command system	system(3S)
issue identification file	issue	issue(4)
print news	items news	news(1)
uucp status inquiry and	job control uustat	uustat(1)
relational database operator	join	join(1)
generate encryption	key makekey	makekey(1)
	kill all active processes killall	killall(1M)
process or a group of processes	kill send a signal to a	kill(2S)
terminate a process	kill	kill(1)
a process or a group of processes	kill send a signal to	kill(2)
kill all active processes	killall	killall(1M)
test your	knowledge quiz	quiz(6)
3-byte integers and long integers	l3tol convert between	l3tol(3C)
copy file systems with	label checking volcopy	volcopy(1M)
pattern scanning and processing	language awk	awk(1)
arbitrary-precision arithmetic	language bc	bc(1)
the C	language preprocessor cpp	cpp(1)
command programming	language sh standard/restricted	sh(1)
troff description of output	language	troff(5)
print	large banner on printer	bbanner(1)
execute commands at a	later time at	at(1)
editor for common object files	ld link	ld(1)
of a member of an archive file	ldahread read the archive header	ldahread(3X)
close a common object file	ldclose	ldclose(3X)
object file access routines	ldfcn common	ldfcn(4)
header of a common object file	ldfhread read the file	ldfhread(3X)

object file symbol table entry	ldgetname symbol name for common	ldgetname(3X)
of a common object file function	ldlread line number entries	ldlread(3X)
a section of a common object file	ldlseek line number entries of	ldlseek(3X)
header of a common object file	ldohseek to the optional file	ldohseek(3X)
a common object file for reading	ldopen open	ldopen(3X)
a section of a common object file	ldrseek to relocation entries of	ldrseek(3X)
header of a common object file	ldshread indexed/named section	ldshread(3X)
section of a common object file	ldsseek to an indexed/named	ldsseek(3X)
entry of a common object file	ldtbindex of a symbol table	ldtbindex(3X)
entry of a common object file	ldtbread an indexed symbol table	ldtbread(3X)
table of a common object file	ldtbseek seek to the symbol	ldtbseek(3X)
return length of Fortran string	len	len(3F)
getopt get option	letter from argument vector	getopt(3C)
generate programs for simple	lexical tasks lex	lex(1)
ordering relation for an object	library lorder find	lorder(1)
archives ar archive and	library maintainer for portable	ar(1)
bar Berkeley archive and	library maintainer	bar(1)
get and set user	limits ulimit	ulimit(2)
establish an out-going terminal	line connection dial	dial(3C)
read one	line line	line(1)
object file linenum	line number entries in a common	linenum(4)
object file manipulate	line number entries of a common	ldlread(3X)
of a common object seek to	line number entries of a section	ldlseek(3X)
common object strip symbol and	line number information from a	strip(1)
	line numbering filter nl	nl(1)
cut out selected fields of each	line of a file cut	cut(1)
send/cancel requests to an LP	line printer lp	lp(1)
read one line	line	line(1)
filter reverse	line-feeds col	col(1)
	linear search and update lsearch	lsearch(3C)
entries in a common object file	linenum line number	linenum(4)
comm select or reject	lines common to two sorted files	comm(1)
report repeated	lines in a file uniq	uniq(1)
subsequent lines of merge same	lines of several files or	paste(1)
exercise	link and unlink system calls link	link(1M)
information L-devices	link devices, connection	L-devices(4)
files ld	link editor for common object	ld(1)
common assembler and	link editor output a.out	a.out(4)
copy,	link or move files cp	cp(1)
L.sys	link systems	L.sys(4)
link to a file	link	link(2)
link and unlink system calls	link exercise	link(1M)
a C program checker	lint	lint(1)
ls	list contents of directories	ls(8)
	list contents of directory ls	ls(1)
for a file system ff	list file names and statistics	ff(1M)
get entries from name	list nlist	nlist(3C)
print name	list of common object file nm	nm(1)
dialups	list of dialup devices	dialups(4)
fsck checklist	list of file systems processed by	checklist(4)
handle variable argument	list varargs	varargs(5)
output of a varargs argument	list vprintf print formatted	vprintf(3S)
construct argument	list(s) and execute command	xargs(1)

macref produce cross-reference	listing of macro files	macref(1)
copy, link or move files	ln, cp, mv	cp(1)
index return	location of Fortran substring	index(3F)
lst	locations in program end	end(3C)
memory plock	lock process, text, or data in	plock(2)
regions for reading or writing	lockf provide exclusive file	lockf(2)
	log gamma function gamma	gamma(3M)
	log in to a new group newgrp	newgrp(1)
logarithm intrinsic function	log Fortran natural	log(3F)
logarithm intrinsic function	log10 Fortran common	log10(3F)
Fortran natural	logarithm intrinsic function log	log(3F)
function exp exponential,	logarithm, power, square root	exp(3M)
process a report of	logged errors errpt	errpt(1M)
configure	logical disks	dconfig(1M)
configure	logical disks	dconfig(8)
get	login name getlogin	getlogin(3C)
get	login name logname	logname(1)
get character	login name of the user cuserid	cuserid(3S)
return	login name of user logname	logname(3X)
change	login password passwd	passwd(1)
setting up an environment at	login time profile	profile(4)
setting up an environment at	login time profile	profile(5)
sign on	login	login(1)
get login name	logname	logname(1)
return login name of user	logname	logname(3X)
setjmp,	longjmp non-local goto	setjmp(3C)
relation for an object library	lorder find ordering	lorder(1)
run a command at	low priority nice	nice(1)
requests to an LP line printer	lp send/cancel	lp(1)
configure the LP spooling system	lpadmin	lpadmin(1M)
resume printing	lphold postpone printing,	lphold(1)
start/stop LP request scheduler	lpsched, lpshut, lpmove	lpsched(1M)
configure the	LP spooling system lpadmin	lpadmin(1M)
print LP status information	lpstat	lpstat(1)
directories	ls list contents of	ls(8)
list contents of directory	ls	ls(1)
linear search and update	lsearch	lsearch(3C)
pointer	lseek move read/write file	lseek(2S)
move read/write file pointer	lseek	lseek(2)
	lst locations in program end	end(3C)
macro processor	m4	m4(1)
	machine-dependent values values	values(5)
access long integer data in a	machine-independent fashion.	sputl(3X)
listing of macro files	macref produce cross-reference	macref(1)
permuted index mptx the	macro package for formatting a	mptx(5)
documents mm the MM	macro package for formatting	mm(5)
mosd the OSDD adapter	macro package for formatting	mosd(5)
graphs mv a	macro package for making view	mv(7)
viewgraphs and mv a troff	macro package for typesetting	mv(5)
	macro processor m4	m4(1)
in this manual man	macros for formatting entries	man(5)
send	mail to users or read mail mail	mail(1)
binary file for transmission via	mail /encode/decode a	uuencode(1c)

message processing system	mailx interactive	mailx(1)
	main memory allocator malloc	malloc(3C)
fast	main memory allocator malloc	malloc(3X)
regenerate groups of programs	maintain, update, and	make(1)
ar archive and library	maintainer for portable archives	ar(1)
SCCS file delta	make a delta (change) to an	delta(1)
	make a directory mkdir	mkdir(1)
or ordinary file mknod	make a directory or a special	mknod(2)
mknod	make a special file	mknod(2S)
	make a unique file name mktemp	mktemp(3C)
of a tape	make an image copy	copytape(1)
	core memory mem	mem(7)
examine system	images crash	crash(1M)
	make posters banner	banner(1)
generate encryption key	makekey	makekey(1)
mv a macro package for	making view graphs	mv(7)
main memory allocator	malloc	malloc(3C)
fast main memory allocator	malloc	malloc(3X)
tsearch	manage binary search trees	tsearch(3C)
	manage hash search tables	hsearch(3C)
a common object file function	manipulate line number entries of	ldlread(3X)
floating-point numbers frexp	manipulate parts of	frexp(3S)
print entries in this	manual man	man(1)
	map of ASCII character set ascii	ascii(5)
diffmk	mark differences between files	diffmk(1)
set file-creation mode	mask umask	umask(1)
set and get file creation	mask umask	umask(2)
set and get file creation	mask umask	umask(2S)
regular expression compile and	match routines regexp	regexp(5)
math functions and constants	math	math(5)
error-handling function	matherr	matherr(3M)
Fortran	maximum-value functions max	max(3F)
generate a	maze maze	maze(6)
return Fortran time accounting	mclock	mclock(3F)
modem	mdial dial the P/75 onboard	dial(1)
core memory	mem	mem(7)
read the archive header of a	member of an archive file	ldahread(3X)
main	memory allocator malloc	malloc(3C)
fast main	memory allocator malloc	malloc(3X)
ramdisk	memory as disk	ramdisk(1M)
shared	memory control operations	shmctl(2)
queue, semaphore set or shared	memory id remove a message	ipcrm(1)
core	memory mem	mem(7)
	memory operations memory	memory(3C)
shared	memory operations shmop	shmop(2)
lock process, text, or data in	memory plock	plock(2)
get shared	memory segment shmget	shmget(2)
rram allows	memory to be used as a disk	rram(7)
memory operations	memory	memory(3C)
sort and/or	merge files sort	sort(1)
files acctmerg	merge or add total accounting	acctmerg(1M)
or subsequent lines of one	merge same lines of several files	paste(1)
permit or deny messages	mesg	mesg(1)

msgctl	message control operations	msgctl(2)
	message operations msgop	msgop(2)
interactive	message processing system	mailx(1)
get	message queue msgget	msgget(2)
shared memory id remove a	message queue, semaphore set or ...	ipcrm(1)
copy of error	messages from UNIX to console	conslog(7)
permit or deny	messages mesg	mesg(1)
system error	messages perror	perror(3C)
Fortran minimum-value functions	min	min(3F)
dsk optionally	mirrored disk driver	dsk(7)
two identical	mirutil utility for connecting	mirutil(1M)
overview of accounting and	miscellaneous accounting commands	acct(1M)
make a directory	mkdir	mkdir(1)
	mkfs construct a file system	mkfs(8)
construct a file system	mkfs	mkfs(1M)
	mknod make a special file	mknod(2S)
build special file	mknod	mknod(1M)
or a special or ordinary file	mknod make a directory	mknod(2)
make a unique file name	mktemp	mktemp(3C)
formatted with the MM macros	mm print/check documents	mm(1)
formatting documents	mm the MM macro package for	mm(5)
document compatibility checker	mmlint sroff/MM nroff/MM	mmlint(1)
viewgraphs, and slides	mmt typeset documents,	mmt(1)
mounted file system table	mnttab	mnttab(4)
remaindering intrinsic functions	mod Fortran	mod(3F)
change	mode chmod	chmod(1)
set file-creation	mode mask umask	umask(1)
chmod change	mode of file	chmod(2S)
dial a Racal-Vadic 3451	modem	dial(1)
mdial dial the P/75 onboard	modem	dial(1)
a compiler/interpreter for	modest-sized programs bs	bs(1)
touch update access and	modification times of a file	touch(1)
set file access and	modification times utime	utime(2)
	monitor uucp network uusub	uusub(1M)
prepare execution profile	monitor	monitor(3C)
guessing game	moo	moo(6)
for crt viewing	more, page file perusal filter	more(1)
package for formatting documents	mosd the OSDD adapter macro	mosd(5)
mount	mount a file system	mount(2S)
establish	mount table setmnt	setmnt(1M)
mount and dismount file system	mount	mount(1M)
mount a file system	mount	mount(2)
mnttab	mounted file system table	mnttab(4)
	move a directory mvdir	mvdir(1M)
copy, link or	move files cp	cp(1)
lseek	move read/write file pointer	lseek(2)
lseek	move read/write file pointer	lseek(2S)
LP request scheduler and	move requests start/stop the	lpsched(1M)
formatting a permuted index	mptx the macro package for	mptx(5)
message control operations	msgctl	msgctl(2)
get message queue	msgget	msgget(2)
message operations	msgop	msgop(2)
copy, link or move files	mv, ln, cp	cp(1)

view graphs	mv a macro package for making	mv(7)
typesetting viewgraphs and	mv a troff macro package for	mv(5)
move a directory	mvdir	mvdir(1M)
device	name devnm	devnm(1M)
create a	name for a temporary file	tmpnam(3S)
symbol table retrieve symbol	name for common object file	ldgetname(3X)
generate file	name for terminal ctermid	ctermid(3S)
get	name from UID getpw	getpw(3C)
return value for environment	name getenv	getenv(3C)
get login	name getlogin	getlogin(3C)
get entries from	name list nlist	nlist(3C)
nm print	name list of common object file	nm(1)
get login	name logname	logname(1)
make a unique file	name mktemp	mktemp(3C)
find	name of a terminal ttyname	ttyname(3C)
print	name of current UNIX system	uname(1)
get	name of current UNIX system	uname(2)
get the	name of the terminal tty	tty(1)
get character login	name of the user cuserid	cuserid(3S)
return login	name of user logname	logname(3X)
working directory	name pwd	pwd(1)
system ff list file	names and statistics for a file	ff(1M)
deliver portions of path	names basename	basename(1)
conventional	names for terminals term	term(5)
generate	names from i-numbers ncheck	ncheck(1M)
print user and group IDs and	names id	id(1)
function log Fortran	natural logarithm intrinsic	log(3F)
generate names from i-numbers	ncheck	ncheck(1M)
Fortran	nearest integer functions round	round(3F)
character definitions for eqn and	neqn eqnchar special	eqnchar(5)
a command on the PCL network	net execute	net(1)
commands stat statistical	network useful with graphical	stat(1)
monitor uucp	network uusub	uusub(1M)
change the format of a text file	newform	newform(1)
log in to a new group	newgrp	newgrp(1)
print	news items news	news(1)
process	nice change priority of a	nice(2S)
run a command at low priority	nice	nice(1)
change priority of a process	nice	nice(2)
line numbering filter	nl	nl(1)
get entries from name list	nlist	nlist(3C)
name list of common object file	nm print	nm(1)
immune to hangups and quits	nohup run a command	nohup(1)
setjmp, longjmp	non-local goto	setjmp(3C)
9700 printer x9700 prepare	nroff documents for the Xerox	x9700(1)
	nroff format or typeset text	nroff(1)
tbl format tables for	nroff or troff	tbl(1)
checker mmlint sroff/MM	nroff/MM document compatibility	mmlint(1)
constructs deroff remove	nroff/troff, tbl, and eqn	deroff(1)
the	null file null	null(7)
file linenum line	number entries in a common object	linenum(4)
file function manipulate line	number entries of a common object	ldlread(3X)
common object seek to line	number entries of a section of a	ldlseek(3X)

factor a	number factor	factor(1)
provide drill in	number facts arithmetic	arithmetic(6)
random	number generator rand	rand(3F)
object strip symbol and line	number info from a common	strip(1)
report	number of free disk blocks df	df(1M)
string to double-precision	number strtod convert	strtod(3C)
convert floating-point	number to string ecvt	ecvt(3C)
line	numbering filter nl	nl(1)
distributed pseudo-random	numbers drand48 uniformly	drand48(3C)
parts of floating-point	numbers frexp manipulate	frexp(3S)
access graphical and	numerical commands graphics	graphics(1)
common	object file access routines ldfcn	ldfcn(4)
dump selected parts of an	object file dump	dump(1)
open a common	object file for reading ldopen	ldopen(3X)
line number entries of a common	object file function ldlread	ldlread(3X)
close a common	object file ldclose	ldclose(3X)
read the file header of a common	object file ldfhread	ldfhread(3X)
entries of a section of a common	object file ldlseek line number	ldlseek(3X)
optional file header of a common	object file ldohseek seek to the	ldohseek(3X)
entries of a section of a common	object file ldrseek relocation	ldrseek(3X)
section header of a common	object file ldshread	ldshread(3X)
section of a common	object file ldsseek	ldsseek(3X)
a symbol table entry of a common	object file ldtbindex index of	ldtbindex(3X)
symbol table entry of a common	object file ldtbread an indexed	ldtbread(3X)
to the symbol table of a common	object file ldtbseek seek	ldtbseek(3X)
line number entries in a common	object file linenum	linenum(4)
print name list of common	object file nm	nm(1)
information for a common	object file reloc relocation	reloc(4)
section header for a common	object file scnhdr	scnhdr(4)
number information from a common	object file strip and line	strip(1)
retrieve symbol name for common	object file symbol table entry	ldgetname(3X)
syms common	object file symbol table format	syms(4)
file header for common	object files filehdr	filehdr(4)
directories cpset install	object files in binary	cpset(1M)
link editor for common	object files ld	ld(1)
print section sizes of common	object files size	size(1)
find ordering relation for an	object library lorder	lorder(1)
od	octal dump	od(8)
text for otroff	ocw prepare constant-width	ocw(1)
octal dump	od	od(8)
octal dump	od	od(1)
	od optical disk	od(7)
mdial dial the P/75	onboard modem	dial(1)
reading ldopen	open a common object file for	ldopen(3X)
	open a stream fopen	fopen(3S)
duplicate an	open file descriptor dup	dup(2)
open	open for reading or writing	open(2S)
open for reading or writing	open	open(2)
profiler	operating system profiler	profiler(1M)
terminal independent	operation routines	termlib(3C)
memory	operations memory	memory(3C)
message control	operations msgctl	msgctl(2)
message	operations msgop	msgop(2)

semaphore control	operations semctl	semctl(2)
semaphore	operations semop	semop(2)
shared memory control	operations shmctl	shmctl(2)
shared memory	operations shmop	shmop(2)
string	operations string	string(3C)
relational database	operator join	join(1)
od	optical disk	od(7)
rod raw interface to	optical disk	od(7)
copy file systems for	optimal access time dcopy	dcopy(1M)
CRT screen handling and	optimization package curses	curses(3X)
vector getopt get	option letter from argument	getopt(3C)
object file seek to the	optional file header of a common	ldohseek(3X)
driver dsk	optionally mirrored disk	dsk(7)
file control	options fcntl	fcntl(5)
set the	options for a terminal stty	stty(1)
parse command	options getopt	getopt(1)
library lorder find	ordering relation for an object	lorder(1)
make a directory or a special or	ordinary file mknod	mknod(2)
prepare constant-width text for	otroff ocw	ocw(1)
connection dial establish an	out-going terminal line	dial(3C)
common assembler and link editor	output a.out	a.out(4)
tc troff	output interpreter	tc(1)
troff description of	output language	troff(5)
vprintf print formatted	output of a varargs argument list	vprintf(3S)
print formatted	output printf	printf(3S)
miscellaneous accounting	overview of accounting and	acct(1M)
change	owner and group of a file chown	chown(2)
change	owner or group chown	chown(1)
compress and expand files	pack	pack(1)
screen handling and optimization	package curses CRT	curses(3X)
mv a macro	package for making view graphs	mv(7)
system activity report	package sar	sar(1M)
standard buffered input/output	package stdio	stdio(3S)
interprocess communication	package stdipc standard	stdipc(3C)
crt viewing ,	page file perusal filter for	more(1)
4014 terminal 4014	paginator for the TEKTRONIX	4014(1)
pp	parallel port interface	pp(7)
get process, process group, and	parent process IDs getpid	getpid(2)
	parse command options getopt	getopt(1)
Fortran integer	part intrinsic function aint	aint(3F)
deliver the last	part of a file tail	tail(1)
Fortran imaginary	part of complex argument aimag	aimag(3F)
dump selected	parts of an object file dump	dump(1)
frexp manipulate	parts of floating-point numbers	frexp(3S)
change login password	passwd	passwd(1)
password file	passwd	passwd(4)
get	password file entry getpwent	getpwent(3C)
write	password file entry putpwent	putpwent(3C)
read a	password getpass	getpass(3C)
pwck	password/group file checkers	pwck(1M)
or subsequent lines of one file	paste lines of several files	paste(1)
deliver portions of	path names	basename(1)
directory getcwd get	path-name of current working	getcwd(3C)

USERFILE UUCP	pathname permissions file	USERFILE(4)
search a file for a	pattern grep	grep(1)
language awk	pattern scanning and processing	awk(1)
suspend process until signal	pause	pause(2)
	pd IMSP disk controller	pd(7)
format acct	per-process accounting file	acct(4)
acctcms command summary from	per-process accounting records	acctcms(1M)
USERFILE UUCP pathname	permissions file	USERFILE(4)
	permit or deny messages mesg	mesg(1)
	permuted index ptx	ptx(1)
macro package for formatting a	permuted index mptx the	mptx(5)
system error messages	perror	perror(3C)
page file	perusal filter for crt viewing	more(1)
terminals pg file	perusal filter for soft-copy	pg(1)
drawing simple pictures	pic troff preprocessor for	pic(1)
preprocessor for drawing simple	pictures pic troff	pic(1)
split a file into	pieces split	split(1)
initiate	pipe to/from a process popen	popen(3S)
create an interprocess channel	pipe	pipe(2)
process, text, or data in memory	plock lock	plock(2)
graphics interface subroutines	plot	plot(3X)
graphics interface	plot	plot(4)
reposition a file	pointer in a stream fseek	fseek(3S)
move read/write file	pointer lseek	lseek(2)
lseek move read/write file	pointer	lseek(2S)
the current value of a file	pointer tell report	tell(2S)
initiate pipe to/from a process	popen	popen(3S)
pp parallel	port interface	pp(7)
data base of terminal types by	port ttytype	ttytype(5)
and library maintainer for	portable archives ar archive	ar(1)
deliver	portions of path names basename	basename(1)
number on a tape srcheof	position to a specific file	srcheof(2S)
functions dim	positive difference intrinsic	dim(3F)
make	posters banner	banner(1)
printing lphold	postpone printing, resume	lphold(1)
exponential, logarithm,	power, square root function	exp(3M)
	pp parallel port interface	pp(7)
print files	pr	pr(1)
function dprod double	precision product intrinsic	dprod(3F)
monitor	prepare execution profile	monitor(3C)
Xerox 9700 printer x9700	prepare nroff documents for the	x9700(1)
Xerox 9700 printer dx9700	prepare troff documents for the	dx9700(1)
the C language	preprocessor cpp	cpp(1)
pictures pic troff	preprocessor for drawing simple	pic(1)
unget undo a	previous get of an SCCS file	unget(1)
graphical files gps graphical	primitive string, format of	gps(4)
types	primitive system data types	types(5)
lpstat	print LP status information	lpstat(1)
	print an SCCS file prs	prs(1)
	print and set the date date	date(1)
	print calendar cal	cal(1)
of a file sum	print checksum and block count	sum(1)
activity sact	print current SCCS file editing	sact(1)

man	print entries in this manual	man(1)
concatenate and	print files cat	cat(1)
	print files pr	pr(1)
cat concatenate and	print files	cat(8)
varargs argument list vprintf	print formatted output of a	vprintf(3S)
	print formatted output printf	printf(3S)
on printer	print large banner	bbanner(1)
file nm	print name list of common object	nm(1)
system uname	print name of current UNIX	uname(1)
	print news items news	news(1)
acctcom search and	print process accounting file(s)	acctcom(1)
topq prioritize	print queue	topq(1M)
object files size	print section sizes of common	size(1)
names id	print user and group IDs and	id(1)
other strings find the	printable strings in an object, or	strings(1)
with the MM macros mm	print/check documents formatted	mm(1)
requests to an LP line	printer lp send/cancel	lp(1)
documents for the Xerox 9700	printer prepare nroff	x9700(1)
documents for the Xerox 9700	printer prepare troff	dx9700(1)
enable/disable LP	printers enable	enable(1)
print formatted output	printf	printf(3S)
postpone printing, resume	printing	lphold(1)
topq	prioritize print queue	topq(1M)
run a command at low	priority nice	nice(1)
change	priority of a process nice	nice(2)
nice change	priority of a process	nice(2S)
faster file system checking	procedure checkall	checkall(1M)
getpid get	process ID	getpid(2S)
process group, and parent	process IDs getpid get process,	getpid(2)
errpt	process a report of logged errors	errpt(1M)
errors errpt	process a report of logged	errpt(1M)
enable or disable	process accounting acct	acct(2)
acctcom search and print	process accounting file(s)	acctcom(1)
set a	process alarm clock alarm	alarm(2)
get	process and child process times	times(2)
init	process control initialization	init(1M)
timex time a command; report	process data and system activity	timex(1)
terminate	process exit	exit(2)
create a new	process fork	fork(2)
set	process group ID setpgrp	setpgrp(2)
IDs getpid get process,	process group, and parent process	getpid(2)
script for the init	process inittab	inittab(4)
terminate a	process kill	kill(1)
change priority of a	process nice	nice(2)
kill send a signal to a	process or a group of processes	kill(2)
initiate pipe to/from a	process popen	popen(3S)
report	process status ps	ps(1)
get process and child	process times	times(2)
wait for child	process to stop or terminate wait	wait(2)
	process trace ptrace	ptrace(2)
suspend	process until signal pause	pause(2)
await completion of	process wait	wait(1)
exit terminate	process	exit(2S)

nice change priority of a	process	nice(2S)
parent process IDs getpid get	process, process group, and	getpid(2)
plock lock	process, text, or data in memory	plock(2)
list of file systems	processed by fsck checklist	checklist(4)
signal to a process or a group of	processes kill send a	kill(2)
kill all active	processes killall	killall(1M)
structure fuser identify	processes using a file or file	fuser(1M)
pattern scanning and	processing language awk	awk(1)
terminate all	processing shutdown	shutdown(1M)
interactive message	processing system mailx	mailx(1)
macro	processor m4	m4(1)
double precision	product intrinsic function dprod	dprod(3F)
display profile data	prof	prof(1)
profile within a function	prof	prof(5)
execution time profile	profil	profil(2)
display	profile data prof	prof(1)
prepare execution	profile monitor	monitor(3C)
execution time	profile profil	profil(2)
environment at login time	profile setting up an	profile(5)
	profile within a function prof	prof(5)
up an environment at login time	profile setting	profile(4)
disk access	profiler sadp	sadp(1M)
operating system profiler	profiler	profiler(1M)
terminate Fortran	program abort	abort(3F)
verify	program assertion assert	assert(3X)
C	program beautifier cb	cb(1)
a C	program checker lint	lint(1)
generate C	program cross-reference cxref	cxref(1)
C	program debugger ctrace	ctrace(1)
lst locations in	program end	end(3C)
dataio download	program files	dataio(1M)
for getargv display a	program name and get arguments	getargv(2S)
side-by-side difference	program sdiff	sdiff(1)
conversion	program units	units(1)
the standard/restricted command	programming language sh shell,	sh(1)
for modest-sized	programs bs compiler/interpreter	bs(1)
lex generate	programs for simple lexical tasks	lex(1)
update, and regenerate groups of	programs make maintain,	make(1)
arithmetic	provide drill in number facts	arithmetic(6)
for reading or writing lockf	provide exclusive file regions	lockf(2)
	provide truth values true	true(1)
print an SCCS file	prs	prs(1)
report process status	ps	ps(1)
generate uniformly distributed	pseudo-random numbers drand48	drand48(3C)
	pt IMSP cartridge controller	pt(7)
process trace	ptrace	ptrace(2)
permuted index	ptx	ptx(1)
copy uuto	public UNIX-to-UNIX system file	uuto(1)
stream ungetc	push character back into input	ungetc(3S)
	put a string on a stream puts	puts(3S)
stream putc	put character or word on a	putc(3S)
or add value to environment	putenv change	putenv(3C)
write password file entry	putpwent	putpwent(3C)

put a string on a stream	puts	puts(3S)
password/group file checkers	pwck	pwck(1M)
working directory name	pwd	pwd(1)
quicker sort	qsort	qsort(3C)
	query terminfo database tput	tput(1)
get message	queue msgget	msgget(2)
topq prioritize print	queue	topq(1M)
memory id remove a message	queue, semaphore set or shared	ipcrm(1)
	quicker sort qsort	qsort(3C)
a command immune to hangups and	quits nohup run	nohup(1)
test your knowledge	quiz	quiz(6)
	ramdisk memory as disk	ramdisk(1M)
simple random-number generator	rand	rand(3C)
random number generator	rand	rand(3F)
rational Fortran dialect	ratfor	ratfor(1)
split f77,	ratfor, or efl files fsplit	fsplit(1)
	rational Fortran dialect ratfor	ratfor(1)
rod	raw interface to optical disk	od(7)
rrm	raw reel-to-reel tape driver	rm(7)
rdsk	raw (unbuffered) version of dsk	dsk(7)
of dsk	rdsk raw (unbuffered) version	dsk(7)
	read a password getpass	getpass(3C)
entry of a common object file	read an indexed symbol table	ldtbread(3X)
header of a common object file	read an indexed/named section	ldshread(3X)
	read from file read	read(2)
read	read from file	read(2S)
send mail to users or	read mail mail	mail(1)
	read one line line	line(1)
member of an archive file	read the archive header of a	ldahread(3X)
object file ldfhread	read the file header of a common	ldfhread(3X)
read from file	read	read(2)
lockf exclusive file regions for	reading or writing	lockf(2)
move	read/write file pointer lseek	lseek(2)
lseek move	read/write file pointer	lseek(2S)
open a common object file for	reading ldopen	ldopen(3X)
open for	reading or writing open	open(2)
open open for	reading or writing	open(2S)
get real and effective user, and	real and effective group ID	getuid(2S)
real effective group get	real and effective user,	getuid(2)
specify what to do upon	receipt of a signal	signal(2)
specify Fortran action on	receipt of a system signal	signal(3F)
from per-process accounting	records acctcms command summary	acctcms(1M)
extract error	records from dump errdead	errdead(1M)
tape frec	recover files from a backup	frec(1M)
crm caching	reel-to-reel tape driver	rm(7)
rrm raw	reel-to-reel tape driver	rm(7)
regular expression compile	regcmp	regcmp(1)
and execute regular expression	regcmp compile	regcmp(3X)
make maintain, update, and	regenerate groups of programs	make(1)
compile and match routines	regexp regular expression	regexp(5)
lockf provide exclusive file	regions for reading or writing	lockf(2)
match routines regexp	regular expression compile and	regexp(5)
regcmp	regular expression compile	regcmp(1)

compile and execute	regular expression regcmp	regcmp(3X)
files comm select or	reject lines common to two sorted	comm(1)
lorder find ordering	relation for an object library	lorder(1)
join	relational database operator	join(1)
for a common object file	reloc relocation information	reloc(4)
of a common object seek to	relocation entries of a section	ldrseek(3X)
common object file reloc	relocation information for a	reloc(4)
functions floor, ceiling,	remainder, absolute value	floor(3M)
mod Fortran	remaindering intrinsic functions	mod(3F)
	reminder service calendar	calendar(1)
uuxqt execute	remote command requests	uuxqt(1M)
L.cmds	remote execution commands	L.cmds(4)
spawn getty to a	remote terminal ct	ct(1)
file rmdel	remove a delta from an SCCS	rmdel(1)
semaphore set or shared memory	remove a message queue,	ipcrm(1)
	remove directory entry unlink	unlink(2)
	remove files or directories rm	rm(1)
constructs deroff	remove nroff/troff, tbl, and eqn	deroff(1)
consistency check and interactive	repair dfsck file system	fsck(1M)
consistency check and interactive	repair fsck, dfsck file system	fsck(8)
consistency check and interactive	repair fsck file system	fsck(1M)
consistency check and interactive	repair fsck file system	fsck(8)
report	repeated lines in a file uniq	uniq(1)
	report CPU time used clock	clock(3C)
communication facilities	report inter-process	ipcs(1)
blocks df	report number of free disk	df(1M)
process a	report of logged errors errpt	errpt(1M)
system activity	report package sar	sar(1M)
activity timex time a command;	report process data and system	timex(1)
	report process status ps	ps(1)
uniq	report repeated lines in a file	uniq(1)
system activity	report sail	sail(1)
stream fseek	reposition a file pointer in a	fseek(3S)
requests start/stop the LP	request scheduler and move	lpsched(1M)
send/cancel	requests to an LP line printer lp	lp(1)
uuxqt execute remote command	requests	uuxqt(1M)
brdrst	reset the VCP controller	brdrst(1M)
restore	restor incremental file system	restor(8)
incremental file system	restore	restor(1M)
lphold postpone printing,	resume printing	lphold(1)
object file symbol table entry	retrieve symbol name for common	ldgetname(3X)
argument getarc	return Fortran command-line	getarc(3F)
variable getenv	return Fortran environment	getenv(3F)
mclock	return Fortran time accounting	mclock(3F)
abs	return integer absolute value	abs(3C)
len	return length of Fortran string	len(3F)
substring index	return location of Fortran	index(3F)
logname	return login name of user	logname(3X)
name getenv	return value for environment	getenv(3C)
data	returned by stat system call stat	stat(5)
filter	reverse line-feeds col	col(1)
create a new file or	rewrite an existing one creat	creat(2)
remove files or directories	rm	rm(1)

remove a delta from an SCCS file	rmdel	rmdel(1)
disk	rod raw interface to optical	od(7)
change	root directory chroot	chroot(2)
chroot change	root directory for a command	chroot(1M)
logarithm, power, square	root function exp exponential,	exp(3M)
Fortran square	root intrinsic function sqrt	sqrt(3F)
Fortran nearest integer functions	round	round(3F)
graphical device	routines and filters gdev	gdev(1)
common object file access	routines ldfcn	ldfcn(4)
expression compile and match	routines regexp regular	regexp(5)
graphical table of contents	routines toc	toc(1)
float and double	routines	float(2S)
as a disk	rram allows memory to be used	rram(7)
driver	rrm raw reel-to-reel tape	rm(7)
nice	run a command at low priority	nice(1)
and quits nohup	run comm immune to hngup	nohup(1)
run daily accounting	runacct	runacct(1M)
SCCS file editing activity	sact print current	sact(1)
disk access profiler	sadp	sadp(1M)
system activity graph	sag	sag(1)
system activity report	sail	sail(1)
system activity report package	sar	sar(1M)
convert formatted input	scanf	scanf(3S)
big file	scanner bfs	bfs(1)
awk pattern	scanning and processing language	awk(1)
programs	scc C compiler for stand-alone	scc(1)
two versions of an SCCS file	sccsdiff compare	sccsdiff(1)
format of SCCS file	sccsfile	sccsfile(4)
start/stop the LP request	scheduler and move requests	lpsched(1M)
header for a common object file	scnhdr section	scnhdr(4)
package curses CRT	screen handling and optimization	curses(3X)
editor based on ex vi	screen-oriented (visual) display	vi(1)
inittab	script for the init process	inittab(4)
system initialization shell	scripts brc	brc(1M)
side-by-side difference program	sdiff	sdiff(1)
grep	search a file for a pattern	grep(1)
binary	search a sorted table bsearch	bsearch(3C)
accounting file(s) acctcom	search and print process	acctcom(1)
linear	search and update lsearch	lsearch(3C)
manage hash	search tables hsearch	hsearch(3C)
manage binary	search trees tsearch	tsearch(3C)
object file scnhdr	section header for a common	scnhdr(4)
file read an indexed/named	section header of a common object	ldshread(3X)
seek to line number entries of a	section of a common object file	ldlseek(3X)
seek to relocation entries of a	section of a common object file	ldrseek(3X)
seek to an indexed/named	section of a common object file	ldsseek(3X)
files size print	section sizes of common object	size(1)
stream editor	sed	sed(1)
section of a common object	seek to line number entries of a	ldlseek(3X)
section of a common object file	seek to relocation entries of a	ldrseek(3X)
of a common object file	seek to the optional file header	ldohseek(3X)
common object file ldtbseek	seek to the symbol table of a	ldtbseek(3X)
get shared memory	segment shmget	shmget(2)

change data	segment space allocation brk	brk(2)
brk change data	segment space allocation	brk(2S)
to two sorted files comm	select or reject lines common	comm(1)
	select terminal filter greek	greek(1)
file cut cut out	selected fields of each line of a	cut(1)
dump dump	selected parts of an object file	dump(1)
semctl	semaphore control operations	semctl(2)
	semaphore operations semop	semop(2)
ipcrm remove a message queue,	semaphore set or shared memory id	ipcrm(1)
get set of	semaphores semget	semget(2)
semaphore control operations	semctl	semctl(2)
get set of semaphores	semget	semget(2)
semaphore operations	semop	semop(2)
group of processes kill	send a signal to a process or a	kill(2)
a group of processes kill	send a signal to a process or	kill(2S)
mail	send mail to users or read mail	mail(1)
line printer lp	send/cancel requests to an LP	lp(1)
reminder	service calendar	calendar(1)
	set a process alarm clock alarm	alarm(2)
umask	set and get file creation mask	umask(2)
umask	set and get file creation mask	umask(2S)
map of ASCII character	set ascii	ascii(5)
execution env	set environment for command	env(1)
modification times utime	set file access and	utime(2)
umask	set file-creation mode mask	umask(1)
get	set of semaphores semget	semget(2)
remove a message queue, semaphore	set or shared memory id ipcrm	ipcrm(1)
	set process group ID setpgrp	setpgrp(2)
	set tabs on a terminal tabs	tabs(1)
stty	set terminal characteristics	stty(2S)
print and	set the date date	date(1)
stty	set the options for a terminal	stty(1)
	set time stime	stime(2)
stime	set time	stime(2S)
	set user and group IDs setuid	setuid(2)
get and	set user limits ulimit	ulimit(2)
assign buffering to a stream	setbuf, setvbuf	setbuf(3S)
establish mount table	setmnt	setmnt(1M)
non-local goto	setjmp, longjmp	setjmp(3C)
set process group ID	setpgrp	setpgrp(2)
login time profile	setting up an environment at	profile(4)
login time profile	setting up an environment at	profile(5)
speed and terminal	settings used by getty gettydefs	gettydefs(4)
set user and group IDs	setuid	setuid(2)
stream setbuf,	setvbuf assign buffering to a	setbuf(3S)
of one merge same lines of	several files or subsequent lines	paste(1)
command programming language	sh the standard/restricted	sh(1)
shmctl	shared memory control operations	shmctl(2)
a message queue, semaphore set or	shared memory id ipcrm remove	ipcrm(1)
	shared memory operations shmop	shmop(2)
get	shared memory segment shmget	shmget(2)
issue a	shell command from Fortran sys	system(3F)
issue a	shell command system	system(3S)

system initialization	shell scripts brc	brc(1M)
command programming language sh	shell, the standard/restricted	sh(1)
shared memory control operations	shmctl	shmctl(2)
get shared memory segment	shmget	shmget(2)
shared memory operations	shmop	shmop(2)
terminate all processing	shutdown	shutdown(1M)
sdiff	side-by-side difference program	sdiff(1)
	sign on login	login(1)
intrinsic function	sign Fortran transfer-of-sign	sign(3F)
suspend process until	signal pause	pause(2)
processes kill send a	signal to a process or a group of	kill(2)
processes kill send a	signal to a process or a group of	kill(2S)
on receipt of a system signal	signal specify Fortran action	signal(3F)
to do upon receipt of a signal	signal specify what	signal(2)
software	signals ssignal	ssignal(3C)
generate programs for	simple lexical tasks lex	lex(1)
troff preprocessor for drawing	simple pictures pic	pic(1)
rand	simple random-number generator	rand(3C)
Fortran	sine intrinsic function sin	sin(3F)
Fortran hyperbolic	sine intrinsic function sinh	sinh(3F)
hyperbolic functions	sinh	sinh(3M)
sine intrinsic function	sinh Fortran hyperbolic	sinh(3F)
print section	sizes of common object files size	size(1)
interval	sleep suspend execution for	sleep(2S)
suspend execution for an interval	sleep	sleep(1)
suspend execution for interval	sleep	sleep(3C)
for typesetting viewgraphs and	slides a troff macro package	mv(5)
documents, viewgraphs, and	slides mmt typeset	mmt(1)
current user ttyslot find the	slot in the utmp file of the	ttyslot(3C)
interpolate	smooth curve spline	spline(1)
SNOBOL interpreter	sno	sno(1)
file perusal filter for	soft-copy terminals pg	pg(1)
	software signals ssignal	ssignal(3C)
	sort and/or merge files sort	sort(1)
quicker	sort qsort	qsort(3C)
topological	sort tsort	tsort(1)
or reject lines common to two	sorted files comm select	comm(1)
binary search a	sorted table bsearch	bsearch(3C)
change data segment	space allocation brk	brk(2)
brk change data segment	space allocation	brk(2S)
terminal ct	spawn getty to a remote	ct(1)
for eqn and neqn eqnchar	special character definitions	eqnchar(5)
build	special file mknod	mknod(1M)
create a new	special file	creat(2S)
mknod make a	special file	mknod(2S)
300s terminals 300 handle	special functions of DASI 300 and	300(1)
2621-series terminals handle	special functions of HP 2640 and	hp(1)
terminal 450 handle	special functions of the DASI 450	450(1)
make a directory or a	special or ordinary file mknod	mknod(2)
format	specification in text files fspec	fspec(4)
isatty returns a 1 if	specified file descriptor is a	isatty(2S)
of a system signal signal	specify Fortran action on receipt	signal(3F)
of a signal signal	specify what to do upon receipt	signal(2)

by getty gettydefs	speed and terminal settings used	gettydefs(4)
find	spelling errors spell	spell(1)
interpolate smooth curve	spline	spline(1)
	split a file into pieces split	split(1)
context	split csplit	csplit(1)
fsplit	split f77, ratfor, or efl files	fsplit(1)
uucp	spool directory clean-up uuclean	uuclean(1M)
configure the LP	spooling system lpadmin	lpadmin(1M)
in a machine-independent fashion.	sputl access long integer data	sputl(3X)
square root intrinsic function	sqrt Fortran	sqrt(3F)
exponential, logarithm, power,	square root function exp	exp(3M)
sqrt Fortran	square root intrinsic function	sqrt(3F)
file number on a tape	srcheof position to a specific	srcheof(2S)
	sroff format text	sroff(1)
compatibility checker mmlint	sroff/MM nroff/MM document	mmlint(1)
software signals	ssignal	ssignal(3C)
scc C compiler for	stand-alone programs	scc(1)
package stdio	standard buffered input/output	stdio(3S)
communication package stdipc	standard interprocess	stdipc(3C)
programming shell, the	standard/restricted command	sh(1)
scheduler and move requests	start/stop the LP request	lpsched(1M)
	stat get file status	stat(2S)
get file status	stat	stat(2)
data returned by stat system call	stat	stat(5)
useful with graphical commands	stat statistical network	stat(1)
list file names and	statistics for a file system ff	ff(1M)
get file system	statistics ustat	ustat(2)
ustat get file system	statistics	ustat(2S)
print LP	status information lpstat	lpstat(1)
stream	status inquiries ferror	ferror(3S)
uustat uucp	status inquiry and job control	uustat(1)
communication facilities	status ipcs report inter-process	ipcs(1)
system control and	status program	sys(1M)
report process	status ps	ps(1)
get file	status stat	stat(2)
stat get file	status	stat(2S)
buffered input/output package	stdio standard	stdio(3S)
communication package	stdipc standard interprocess	stdipc(3C)
	stime set time	stime(2S)
set time	stime	stime(2)
wait for child process to	stop or terminate wait	wait(2)
comparision intrinsic functions	strcmp string	strcmp(3F)
	stream editor sed	sed(1)
close or flush a	stream fclose	fclose(3S)
open a	stream fopen	fopen(3S)
reposition a file pointer in a	stream fseek	fseek(3S)
get character or word from a	stream getc	getc(3S)
get a string from a	stream gets	gets(3S)
put character or word on a	stream putc	putc(3S)
put a string on a	stream puts	puts(3S)
assign buffering to a	stream setbuf, setvbuf	setbuf(3S)
	stream status inquiries ferror	ferror(3S)
push character back into input	stream ungetc	ungetc(3S)

ft IMSP	streaming cartridge controller	ft(7)
long integer and base-64 ASCII	string a64l convert between	a64l(3C)
functions strcmp	string comparision intrinsic	strcmp(3F)
convert date and time to	string ctime	ctime(3C)
convert floating-point number to	string ecvt	ecvt(3C)
get a	string from a stream gets	gets(3S)
return length of Fortran	string len	len(3F)
put a	string on a stream puts	puts(3S)
strtod convert	string to double-precision number	strtod(3C)
convert	string to integer strtol	strtol(3C)
string operations	string	string(3C)
gps graphical primitive	string, format of graphical files	gps(4)
find the printable	strings in an object, or binary, file	strings(1)
information from a common	strip symbol and line number	strip(1)
string to double-precision number	strtod convert	strtod(3C)
convert string to integer	strtol	strtol(3C)
processes using a file or file	structure fuser identify	fuser(1M)
characteristics	stty set terminal	stty(2S)
set the options for a terminal	stty	stty(1)
become super-user or another user	su	su(1)
graphics interface	subroutines plot	plot(3X)
same lines of several files or	subsequent lines of one file	paste(1)
return location of Fortran	substring index	index(3F)
and block count of a file	sum print checksum	sum(1)
	summarize disk usage du	du(1)
du	summarize disk usage	du(8)
accounting records command	summary from per-process	acctcms(1M)
update the	super block sync	sync(1)
update	super-block sync	sync(2)
become	super-user or another user su	su(1)
document analyze	surface characteristics of a	style(1)
interval sleep	suspend execution for an	sleep(1)
sleep	suspend execution for interval	sleep(2S)
sleep	suspend execution for interval	sleep(3C)
pause	suspend process until signal	pause(2)
swap bytes	swab	swab(3C)
swap	swap device	swap(7)
information from a strip	symbol and line number	strip(1)
file symbol table retrieve	symbol name for common object	ldgetname(3X)
name for common object file	symbol table entry ldgetname	ldgetname(3X)
object compute the index of a	symbol table entry of a common	ldtbindex(3X)
object file read an indexed	symbol table entry of a common	ldtbread(3X)
common object file	symbol table format syms	syms(4)
file ldtbseek seek to the	symbol table of a common object	ldtbseek(3X)
object file symbol table format	syms common	syms(4)
update the super block	sync	sync(1)
update super-block	sync	sync(2)
call another UNIX	system	cu(1C)
	system activity graph sag	sag(1)
sar	system activity report package	sar(1M)
	system activity report sail	sail(1)
command; report process data and	system activity timex time a	timex(1)
daily/weekly UNIX system file	system backup filesave	filesave(1M)

data returned by stat	system call stat	stat(5)
exercise link and unlink	system calls link	link(1M)
checkall faster file	system checking procedure	checkall(1M)
UNIX-to-UNIX	system command execution uux	uux(1)
interactive repair fsck file	system consistency check and	fsck(1M)
interactive repair fsck file	system consistency check and	fsck(8)
file ioctl.syscon	system console configuration	ioctl.syscon(4)
sys	system control and status program	sys(1M)
UNIX system to UNIX	system copy uucp	uucp(1)
what to do when the	system crashes crash	crash(8)
primitive	system data types types	types(5)
file	system debugger fsdb	fsdb(1M)
fsdb file	system debugger	fsdb(8)
	system error messages perror	perror(3C)
names and statistics for a file	system ff list file	ff(1M)
public UNIX-to-UNIX	system file copy uuto	uuto(1)
filesave daily/weekly UNIX	system file system backup	filesave(1M)
examine	system images crash	crash(1M)
scripts brc	system initialization shell	brc(1M)
configure the LP spooling	system lpadmin	lpadmin(1M)
interactive message processing	system mailx	mailx(1)
construct a file	system mkfs	mkfs(1M)
mount and dismount file	system mount	mount(1M)
mount a file	system mount	mount(2)
operating	system profiler profiler	profiler(1M)
incremental file	system restore	restor(1M)
Fortran action on receipt of a	system signal signal specify	signal(3F)
get file	system statistics ustat	ustat(2)
ustat get file	system statistics	ustat(2S)
mounted file	system table mnttab	mnttab(4)
UNIX	system to UNIX system copy uucp	uucp(1)
unmount a file	system umount	umount(2)
print name of current UNIX	system uname	uname(1)
get name of current UNIX	system uname	uname(2)
format of	system volume fs	fs(4)
who is on the	system who	who(1)
mount a file	system	mount(2S)
issue a shell command	system	system(3S)
a shell command from Fortran	system issue	system(3F)
dcopy copy file	systems for optimal access time	dcopy(1M)
checklist list of file	systems processed by fsck	checklist(4)
volcopy copy file	systems with label checking	volcopy(1M)
binary search a sorted	table bsearch	bsearch(3C)
for common object file symbol	table entry ldgetname name	ldgetname(3X)
compute the index of a symbol	table entry of a common object	ldtbindex(3X)
file read an indexed symbol	table entry of a common object	ldtbread(3X)
common object file symbol	table format syms	syms(4)
mounted file system	table mnttab	mnttab(4)
ldtbseek seek to the symbol	table of a common object file	ldtbseek(3X)
graphical	table of contents routines toc	toc(1)
establish mount	table setmnt	setmnt(1M)
tbl format	tables for nroff or troff	tbl(1)
manage hash search	tables hsearch	hsearch(3C)

set tabs on a terminal	tabs	tabs(1)
deliver the last part of a file	tail	tail(1)
Fortran	tangent intrinsic function tan	tan(3F)
tangent intrinsic function	tanh Fortran hyperbolic	tanh(3F)
fbackup make a fast	tape backup of a file system	fbackup(1M)
fbackup make a fast	tape backup of a file system	fbackup(8)
make an image copy of a	tape copytape	
crm caching reel-to-reel	tape driver	rm(7)
rrm raw reel-to-reel	tape driver	rm(7)
	tape file archiver tar	tar(1)
dump incremental dump	tape format	dump(4)
recover files from a backup	tape frec	frec(1M)
tape file archiver	tar	tar(1)
programs for simple lexical	tasks lex generate	lex(1)
troff	tbl format tables for nroff or	tbl(1)
deroff remove nroff/troff,	tbl, and eqn constructs	deroff(1)
	tc troff output interpreter	tc(1)
of a file pointer	tell report the current value	tell(2S)
create a	temporary file tmpfile	tmpfile(3S)
create a name for a	temporary file tmpnam	tmpnam(3S)
format of compiled term file	term	term(4)
conventional names for terminals	term	term(5)
data base	termcap terminal capability	termcap(4)
paginator for the TEKTRONIX 4014	terminal 4014	4014(1)
special functions of the DASI 450	terminal 450 handle	450(1)
EOT on the other	terminal and exits. write	write(1)
termcap	terminal capability data base	termcap(4)
terminfo	terminal capability data base	terminfo(4)
gtty get	terminal characterisitcs	gtty(2S)
stty set	terminal characteristics	stty(2S)
vconfig virtual	terminal configuration	vconfig(1M)
spawn getty to a remote	terminal ct	ct(1)
generate file name for	terminal ctermid	ctermid(3S)
select	terminal filter greek	greek(1)
routines termlib	terminal independent operation	termlib(3C)
tty general	terminal interface	tty(7)
establish an out-going	terminal line connection dial	dial(3C)
gettydefs speed and	terminal settings used by getty	gettydefs(4)
set the options for a	terminal stty	stty(1)
set tabs on a	terminal tabs	tabs(1)
get the name of the	terminal tty	tty(1)
find name of a	terminal ttyname	ttyname(3C)
ttytype data base of	terminal types by port	ttytype(5)
functions of DASI 300 and 300s	terminals 300 handle special	300(1)
of HP 2640 and 2621-series	terminals hp special functions	hp(1)
file perusal filter for soft-copy	terminals pg	pg(1)
conventional names for	terminals term	term(5)
	terminate Fortran program abort	abort(3F)
	terminate a process kill	kill(1)
shutdown	terminate all processing	shutdown(1M)
	terminate process exit	exit(2)
exit	terminate process	exit(2S)
daemon errstop	terminate the error-logging	errstop(1M)

wait for child process to stop or	terminate wait	wait(2)
	terminfo compiler tic	tic(1M)
query	terminfo database tput	tput(1)
terminal capability data base	terminfo	terminfo(4)
operation routines	termlib terminal independent	termlib(3C)
command	test condition evaluation	test(1)
	test your knowledge quiz	quiz(6)
condition evaluation command	test	test(1)
	text editor ed	ed(1)
	text editor ex	ex(1)
casual users edit	text editor variant of ex for	edit(1)
change the format of a	text file newform	newform(1)
format specification in	text files fspec	fspec(4)
eqn format mathematical	text for nroff or troff	eqn(1)
ocw prepare constant-width	text for otroff	ocw(1)
troff	text formatting and typesetting	troff(1)
nroff format or typeset	text	nroff(1)
sroff format	text	sroff(1)
lock process,	text, or data in memory plock	plock(2)
	the C language preprocessor cpp	cpp(1)
	the game of backgammon back	back(6)
	the game of black jack bj	bj(6)
	the game of craps craps	craps(6)
wump	the game of hunt-the-wumpus	wump(6)
	the null file null	null(7)
terminfo compiler	tic	tic(1M)
	time a command time	time(1)
data and system activity timex	time a command; report process	timex(1)
update access and modification	times of a file touch	touch(1)
set file access and modification	times utime	utime(2)
get process and child process	times	times(2)
process data and system activity	timex time a command; report	timex(1)
create a temporary file	tmpfile	tmpfile(3S)
a name for a temporary file	tmpnam create	tmpnam(3S)
initiate pipe	to/from a process popen	popen(3S)
table of contents routines	toc graphical	toc(1)
	topological sort tsort	tsort(1)
	topq prioritize print queue	topq(1M)
merge or add	total accounting files acctmerg	acctmerg(1M)
and modification times of a file	touch update access	touch(1)
graphics filters	tplot	tplot(1)
query terminfo database	tput	tput(1)
translate characters	tr	tr(1)
process	trace ptrace	ptrace(2)
function sign Fortran	transfer-of-sign intrinsic	sign(3F)
	translate characters conv	conv(3C)
	translate characters tr	tr(1)
system uucico file	transport program for the uucp	uucico(1M)
walk a file	tree ftw	ftw(3C)
manage binary search	trees tsearch	tsearch(3C)
trigonometric functions	trig	trig(3M)
language	troff description of output	troff(5)
9700 printer dx9700 prepare	troff documents for the Xerox	dx9700(1)

typesetting viewgraphs mv a	troff macro package for	mv(5)
tc	troff output interpreter	tc(1)
simple pictures pic	troff preprocessor for drawing	pic(1)
typesetting	troff text formatting and	troff(1)
tbl format tables for nroff or	troff	tbl(1)
mathematical text for nroff or	troff eqn format	eqn(1)
files for device-independent	troff font description	font(5)
provide truth values	true	true(1)
manage binary search trees	tsearch	tsearch(3C)
topological sort	tsort	tsort(1)
get the name of the terminal	tty	tty(1)
general terminal interface	tty	tty(7)
find name of a terminal	ttyname	ttyname(3C)
the utmp file of the current user	ttyslot find the slot in	ttyslot(3C)
types by port	ttytype data base of terminal	ttytype(5)
explicit Fortran	type conversion ftype	ftype(3F)
determine file	type file	file(1)
primitive system data	types types	types(5)
and slides mmt	typeset documents, viewgraphs,	mmt(1)
nroff format or	typeset text	nroff(1)
mv a troff macro package for	typesetting viewgraphs and slides	mv(5)
troff text formatting and	typesetting	troff(1)
get and set user limits	ulimit	ulimit(2)
creation mask	umask set and get file	umask(2S)
set file-creation mode mask	umask	umask(1)
	umount unmount a file system	umount(2S)
unmount a file system	umount	umount(2)
print name of current UNIX system	uname	uname(1)
get name of current UNIX system	uname	uname(2)
rdsk raw	(unbuffered) version of dsk	dsk(7)
a previous get of an SCCS file	unget undo	unget(1)
character back into input stream	ungetc push	ungetc(3S)
pseudo-random numbers generate	uniformly distributed	drand48(3C)
report repeated lines in a file	uniq	uniq(1)
make a	unique file name mktemp	mktemp(3C)
conversion program	units	units(1)
exercise link and	unlink system calls link	link(1M)
remove directory entry	unlink	unlink(2)
	unmount a file system umount	umount(2)
umount	unmount a file system	umount(2S)
suspend process	until signal pause	pause(2)
times of a file touch	update access and modification	touch(1)
linear search and	update lsearch	lsearch(3C)
	update super-block sync	sync(2)
	update the super block sync	sync(1)
programs make maintain,	update, and regenerate groups of	make(1)
specify what to do	upon receipt of a signal signal	signal(2)
summarize disk	usage du	du(1)
du summarize disk	usage	du(8)
stat statistical network	useful with graphical commands	stat(1)
generate disk accounting data by	user ID diskusg diskusg	diskusg(1M)
print	user and group IDs and names id	id(1)
set	user and group IDs setuid	setuid(2)

	user crontab file crontab	crontab(1)
get character login name of the	user cuserid	cuserid(3S)
	user environment environ	environ(5)
get and set	user limits ulimit	ulimit(2)
return login name of	user logname	logname(3X)
become super-user or another	user su	su(1)
in the utmp file of the current	user ttyslot find the slot	ttyslot(3C)
getuid get real and effective	user, and real and effective	getuid(2S)
group get real user, effective	user, real group, and effective	getuid(2)
send mail to	users or read mail mail	mail(1)
write to all	users wall	wall(1M)
editor (variant of ex for casual	users) edit text	edit(1)
fuser identify processes	using a file or file structure	fuser(1M)
statistics	ustat get file system	ustat(2S)
get file system statistics	ustat	ustat(2)
graphical	utilities gutil	gutil(1)
identical mirutil	utility for connecting two	mirutil(1M)
access and modification times	utime set file	utime(2)
utmp	utmp and wtmp entry formats	utmp(4)
access	utmp file entry getut	getut(3C)
ttyslot find the slot in the	utmp file of the current user	ttyslot(3C)
for the uucp system	uucico file transport program	uucico(1M)
uucp spool directory clean-up	uuclean	uuclean(1M)
monitor	uucp network uusub	uusub(1M)
control uustat	uucp status inquiry and job	uustat(1)
file transport program for the	uucp system uucico	uucico(1M)
UNIX system to UNIX system copy	uucp	uucp(1)
encode/decode a binary file for/	uuencode,uudecode	uuencode(1c)
status inquiry and job control	uustat uucp	uustat(1)
monitor uucp network	uusub	uusub(1M)
UNIX-to-UNIX system file copy	uuto public	uuto(1)
system command execution	uux UNIX-to-UNIX	uux(1)
requests	uuxqt execute remote command	uuxqt(1M)
brdrst reset the	VCP controller	brdrst(1M)
vcpdmp dump contents of	VMEbus Communication Processor's	vcpdmp(1M)
validate SCCS file	val	val(1)
	validate SCCS file val	val(1)
return integer absolute	value abs	abs(3C)
Fortran absolute	value abs	abs(3F)
return	value for environment name getenv	getenv(3C)
ceiling, remainder, absolute	value functions floor floor,	floor(3M)
tell report the current	value of a file pointer	tell(2S)
change or add	value to environment putenv	putenv(3C)
provide truth	values true	true(1)
machine-dependent values	values	values(5)
print formatted output of a	varargs argument list vprintf	vprintf(3S)
handle variable argument list	varargs	varargs(5)
return Fortran environment	variable getenv	getenv(3F)
version control	vc	vc(1)
virtual terminal configuration	vconfig	vconfig(1M)
Communication Processor's	vcpdmp, dump contents of VMEbus	vcpdmp(1M)
get option letter from argument	vector getopt	getopt(3C)
	verify program assertion assert	assert(3X)

	version control vc	vc(1)
get a	version of a SCCS file get	get(1)
compare two	versions of an SCCS file sccsdiff	sccsdiff(1)
display editor based on ex	vi screen-oriented (visual)	vi(1)
mv a macro package for making	view graphs	mv(7)
troff macro package for typesetting	viewgraphs and mv	mv(5)
mmt typeset documents,	viewgraphs, and slides	mmt(1)
vconfig	virtual terminal configuration	vconfig(1M)
file systems with label checking	volcopy copy	volcopy(1M)
format of system	volume fs	fs(4)
output of a varargs argument list	vprintf print formatted	vprintf(3S)
terminate wait	wait for child process to stop or	wait(2)
await completion of process	wait	wait(1)
	walk a file tree ftw	ftw(3C)
write to all users	wall	wall(1M)
word count	wc	wc(1)
crashes crash	what to do when the system	crash(8)
	who is on the system who	who(1)
who is doing what	whodo	whodo(1M)
profile	within a function prof	prof(5)
	word count wc	wc(1)
get character or	word from a stream getc	getc(3S)
guess the	word hangman	hangman(6)
put character or	word on a stream putc	putc(3S)
find hyphenated	words hyphen	hyphen(1)
change	working directory cd	cd(1)
change	working directory chdir	chdir(2)
get path-name of current	working directory getcwd	getcwd(3C)
	working directory name pwd	pwd(1)
chdir change	working directory	chdir(2S)
write	write on a file	write(2S)
putpwent	write password file entry	putpwent(3C)
	write to all users wall	wall(1M)
write on a file	write	write(2)
on the other terminal and exits.	write EOT	write(1)
open for reading or	writing open	open(2)
open for reading or	writing	open(2S)
file regions for reading or	writing provide exclusive	lockf(2)
utmp and	wtmp entry formats utmp	utmp(4)
the game of hunt-the-wumpus	wump	wump(6)
for the Xerox 9700 printer	x9700 prepare nroff documents	x9700(1)
list(s) and execute command	xargs construct argument	xargs(1)
yacc	yet another compiler-compiler	yacc(1)

NAME

brc, bcheckrc, rc, powerfail – system initialization shell scripts

SYNOPSIS

/etc/brc

/etc/bcheckrc

/etc/rc

/etc/powerfail

DESCRIPTION

Except for *powerfail* , these shell procedures are executed via entries in */etc/inittab* by *init (1M)* when the system is changed out of *SINGLE USER* mode. *Powerfail* is executed whenever a system power failure is detected.

The *brc* procedure clears the mounted file system table, **/etc/mnttab** (see *mnttab (4)),* and loads any programmable micro-processors with their appropriate scripts.

The *bcheckrc* procedure performs all the necessary consistency checks to prepare the system to change into multi-user mode. It will prompt to set the system date and to check the file systems with *fsck (1M).*

The *rc* procedure starts all system daemons before the terminal lines are enabled for multi-user mode. In addition, file systems are mounted and accounting, error logging, system activity logging and the Remote Job Entry (RJE) system are activated in this procedure.

The *powerfail* procedure is invoked when the system detects a power failure condition. Its chief duty is to reload any programmable micro-processors with their appropriate scripts, if suitable. It also logs the fact that a power failure occurred.

SEE ALSO

fsck(1M), init(1M), shutdown(1M), inittab(4), mnttab(4).

NAME

brdrst – reset the VCP controller

SYNOPSIS

/etc/dconfig/brdrst vcpname

DESCRIPTION

brdrst is used to reset a VCP controller board while the UNIX system is running. **vcpname** is the name of the VCP controller found in the **/dev** directory. Possible names are from **vc0** to **vc8**.

FILES

/etc/inittab

SEE ALSO

/dev/dataio
/etc/vconfig/vconfig

CAUTION

Before resetting a VCP controller, KILL all processes that are sleeping on that controller. Also, turn off all getty entries in the **/etc/inittab** file for that VCP.

NAME

cdconf – configurable disk configuration utility

SYNOPSIS

cdconf conf_dev stripe_size slave off sects slave off sects ...

DESCRIPTION

The **cdconf** command is used to configure one minor device of the configurable disk driver. It would normally be used in **/etc/rc** to do the configurations before mounting file systems. It can be used to reconfigure a device that was previously configured, provided that no process has it open and it is not a mounted file system.

The configurable disk driver allows several logical disks to be joined into one logical disk. Thus the configured device can be larger than any physical disk. The configured device can be the concatenation of the slave devices or it can be striped among them. Striping has the advantage of distributing the accesses more evenly among the slave devices. This allows better throughput if they are on different physical devices. The stripe size defines the granularity at which sections in the configured device alternate among the slave devices.

The configuration is completely determined by the parameters passed on the **cdconf** command line. Up to 16 slave devices can be specified. Each one is defined by a group of three parameters, **slave,** **off**, and **sects**.

Numeric parameters are decimal unless they start with a zero, which indicates octal, or unless they start with *0x*, which indicates hex.

The parameters are interpreted as follows:

conf_dev This is the name of a character special file for the configurable disk driver. This, and the corresponding block device, will be configured.

stripe_size This is the stripe size in 512 byte sectors. A value of zero means the slave devices are concatenated in their order of appearance, rather than striped. Picking the proper stripe size can have a significant effect on performance. It should be evenly divisable by the usual access size, otherwise some accesses will require two disk requests.

slave This is the name of a block special file to include in the configured device.

off This is the offset, in sectors, to the first usable sector on the slave device. If the slave device is an entire physical disk (such as **/dev/dsk/4x0**) the offset should be at least two to avoid damaging the data maintained by **dconfig**. This applies to any slave device which begins at sector zero of a physical disk.

sects This is the number of 512 byte sectors to include in the configured device. The sum of **off** and **sects** must be less than or equal to the size of the slave device as defined with **dconfig**. If **sects** is not a multiple of the stripe size it will be rounded down.

DIAGNOSTICS

If there are no errors nothing is printed and the command terminates with an exit code of zero. Any errors will be reported on the standard error output and the exit code will be one.

Only the superuser can use this command.

When a device is configured, the last 512 byte sector of each slave device is read to ensure it will work. If a slave device is not large enough as a result of the values given to **dconfig**, then the error message will be "A slave device is not big enough". If the sector was not readable for some other reason, such as the drive is powered off, then the error message will be "Last sector of a slave device was not readable".

EXAMPLE

```
cdconf /dev/rcd/a 8 /dev/dsk/0s3 0 715316 /dev/dsk/1s0 2 835314
                                              /dev/dsk/2s0      2
835314
```

This creates a logical disk of almost 1.2 gigabytes on a machine with three drives of 835316 usable sectors each. The stripe size is 4Kb (eight sectors of 512 bytes). The first 60Mb (120000 sectors) of drive zero are used for the root file system and swap space. The next 715312 sectors are part of the configured device and the last four sectors are lost to rounding. Drives one and two each contribute 835312 sectors. The first two sectors are reserved for **dconfig** and the last two are lost to rounding.

The striping causes the following mapping of sectors in **/dev/cd/a** to slave devices

0-7	:	/dev/dsk/0s3	0-7
8-15	:	/dev/dsk/1s0	2-9
16-23	:	/dev/dsk/2s0	2-9
24-31	:	/dev/dsk/0s3	8-15
32-39	:	/dev/dsk/1s0	10-17

FILES

/dev/cd/[a-h] Block special files for configurable disk driver
/dev/rcd/[a-h] Character special files for configurable disk driver

SEE ALSO

cd(7)

NAME

checkall – faster file system checking procedure

SYNOPSIS

/etc/checkall

DESCRIPTION

The *checkall* procedure is a prototype and must be modified to suit local conditions. The following will serve as an example:

```
# check the root file system by itself
fsck /dev/dsk/0s0

# dual fsck of drives 0 and 1
dfsck /dev/rdsk/0s[12345] – /dev/rdsk/1s1
```

In the above example (where **/dev/rdsk/1s1** is 320K blocks and **/dev/rdsk/0s[12345]** are each 65K or less), a previous sequential *fsck* took 19 minutes. The *checkall* procedure takes 11 minutes.

Dfsck is a program that permits an operator to interact with two *fsck (1M)* programs at once. To aid in this, *dfsck* will print the file system name for each message to the operator. When answering a question from *dfsck* , the operator must prefix the response with a **1** or a **2** (indicating that the answer refers to the first or second file system group).

Due to the file system load balancing required for dual checking, the *dfsck (1M)* command should always be executed through the *checkall* shell procedure.

In a practical sense, the file systems are divided as follows:

```
dfsck file_systems_on_drive_0 – file_systems_on_drive_1
dfsck file_systems_on_drive_2 – file_systems_on_drive_3
  . . .
```

A three-drive system can be handled by this more concrete example (assumes two large file systems per drive):

```
dfsck /dev/dsk/3s1 /dev/dsk/0s[14] – /dev/dsk/1s[14]
/dev/dsk/3s4
```

Note that the first file system on drive 3 is first in the *filesystems1* list and is last in the *filesystems2* list assuring that references to that drive will not overlap at execution time.

WARNINGS

1. Do not use *dfsck* to check the *root* file system.
2. On a check that requires a scratch file (see **–t** above), be careful not to use the same temporary file for the two groups (this is sure to scramble the file systems).
3. The *dfsck* procedure is useful only if the system is set up for multiple physical I/O buffers.

SEE ALSO

fsck(1M).

Setting Up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide .*

NAME

cron - clock daemon

SYNOPSIS

/etc/cron

DESCRIPTION

Cron executes commands at specified dates and times. Regularly scheduled commands can be specified according to instructions found in crontab files; users can submit their own crontab file via the *crontab* command. Commands which are to be executed only once may be submitted via the *at* command. Since *cron* never exits, it should only be executed once. This is best done by running *cron* from the initialization process through the file **/etc/rc** (see *init*(1M)).

Cron only examines crontab files and at command files during process initialization and when a file changes. This reduces the overhead of checking for new or changed files at regularly scheduled intervals.

FILES

/usr/lib/cron	main cron directory
/usr/lib/cron/log	accounting information
/usr/spool/cron	spool area

SEE ALSO

at(1), crontab(1), sh(1), init(1M).

DIAGNOSTICS

A history of all actions taken by cron are recorded in **/usr/lib/cron/log.**

NAME

dataio – download program files

SYNOPSIS

/etc/dataio [options]

DESCRIPTION

This program transfers program files from the UNIX system to either the EH 4A/BPS4 prom programmer or a DATA I/O prom programmer or a Plexus system that is running a debugging program. The program options are as follows, where *xxxx* is a hex number:

−a *xxxx* Sets *xxxx* as the base address for text relocation. This address is also sent to the Plexus monitor if the program is in that mode.

−b *xxxx* Sets *xxxx* as the base address for bss relocation. This address is also sent to the Plexus monitor if the program is in that mode.

−w Initializes the EH-4A prom programmer, does the download, and programs the prom.

−c Puts a checksum (so that the words will sum to 0) at location 0x0ffe. Used for making proms so that they can be checked for integrity.

−t *info* If the output file is a tty then *info* is used to set up the terminals options. This is done by first opening the terminal and then issuing an *stty* command to it with *info* as the parameters.

−o *outf* Sets the output file name to *outf*.

−f *inf* Sets the input file name to *inf*.

−k *promsize*
Determines the size of the proms being programmed. (*promsize* is multiplied by 1024 to get the actual prom size.)

−i *il, cno* Indicates the interleave factor *il* and the set number *cno* to program. If *cno* is 0, then all sets will be programmed.

−p Sets the program to output data in the format used by the EH prom programmer.

−z Sets the program to output data in the format used by the Plexus monitor.

−s *xxxx* Sets the segment number sent to the Plexus monitor.

−u Used for downloading UNIX thru the boot program,

−v Used for the 2732As.

−y *xxxx* Sets the communications address for loading the SIOC.

−dv *n m* Download VCP.

n starting ACP number [0-3]

m number of ACPs such that n + m <= 3

−A *prombase*
Sets *prombase* as the base address for programming proms.

−C Puts a checksum (so that the words will sum to 0) in the last word of data that is downloaded. Used for making proms for only part of a file that can be checked for data integrity.

−B Used for 4B/BPS4 prom programmer.

−D Used for the DATA I/C 29A programmer.

−F *xxyy* Used only for the DATA I/O programmer and must be present if the -D switch is *xx* is the family and *yy* is the pinout code (e.g. 1924 for 2732DC).

−L Object file header contains LONGs as in 68000 type object files. The default options are:

-a 0000
-b 0000
-t 9600
-o /dev/tty15
-f vcpprom
-l
-p
-s 0000
-A 0000

FILES

/dev/tty15

NOTES

This is a Plexus command; it is not part of standard System V.

SEE ALSO

vconfig(1M)

NAME

dconfig – configure logical disks

SYNOPSIS

/etc/dconfig - for use under UNIX

dconfig - for running program from release tape only

/stand/dconfig - for standalone use (UNIX not running) only

DESCRIPTION

Dconfig allows you to change the Sys5 default logical disk address assignments and the default UNIX device mapping. It also can be used to verify the logical disk configuration, change the system nodename for **uucp** and **uname**, or change the primary bootname.

Dconfig has both regular (**/etc/dconfig**) and standalone (**/stand/dconfig**) versions. Plexus release tapes also contain a copy of **dconfig**. The arguments to **/etc/dconfig** (the regular version) differ from those for the standalone and tape versions. **/etc/dconfig** expects the special files defined in the **/dev** directory as arguments, while the standalone version and the release tape version both use built-in special filenames as described in the user's manual for your system.

Dconfig prompts for responses, and gives the current values for each parameter in brackets. A <return> leaves the values the same; a <return> in response to a yes or no question defaults to "no". Unlike most Sys5 programs, **dconfig** expects response in terms of 512-byte sectors, rather than 1024 byte blocks.

If **dconfig** for any reason (e.g., permissions) cannot access the disk you type, it continues to give the "Disk?" prompt. For more complete information and examples, see the chapter on standalone programs in your user's manual.

NOTES

This is a Plexus command. It is not part of standard System V.

Dconfig should not be run on disks containing a raw file system which starts at block 0 of the physical disk, as it will ruin the data in the raw file system.

Dconfig cannot use the first two blocks on a disk in a file system other than the first logical one. That is, if you have two disks, the file system size declarations for */dev/dsk/0s0* and */dev/dsk/0s1* must start at sector 0; 0s2-0s15 must not use sectors 0 and 1. On the second disk, the file system size declarations for */dev/dsk/1s0 (/dev/dsk/0s16)* and */dev/dsk/1s1 (/dev/dsk/0s17)* must start at sector 0; 1s2-1s15 (0s18-0s31) must not use sectors 0 and 1.

/etc/dconfig should be used only to examine, and not change, data.

SEE ALSO

uname(1).

NAME

errdead – extract error records from dump

SYNOPSIS

/etc/errdead dumpfile [namelist]

DESCRIPTION

When hardware errors are detected by the system, an error record that contains information pertinent to the error is generated. If the error-logging daemon *errdemon (1M)* is not active or if the system crashes before the record can be placed in the error file, the error information is held by the system in a local buffer. *Errdead* examines a system dump (or memory), extracts such error records, and passes them to *errpt (1M)* for analysis.

The *dumpfile* specifies the file (or memory) that is to be examined. The system namelist is specified by *namelist ;* if not given, **/unix** is used.

FILES

/unix	system namelist
/usr/bin/errpt	analysis program
/usr/tmp/errXXXXXX	temporary file

DIAGNOSTICS

Diagnostics may come from either *errdead* or *errpt .* In either case, they are intended to be self-explanatory.

SEE ALSO

errdemon(1M), errpt(1M).

NAME

errdemon – error-logging daemons

SYNOPSIS

/usr/lib/errdemon [*errfile* [*consfile* [*eccfile*]]]]

DESCRIPTION

The error logging daemons *errdemon* collects data from the operating system by reading the special files **/dev/error** and **/dev/conslog.** One process is created to read each special file. All console messages printed by the kernel are read from **/dev/conslog** and appended to *consfile.* If *consfile* is not specified then **/usr/adm/console_log** is used. Binary error records are read from **/dev/error** and appended to *errfile.* If *errfile* is not specified when the daemon is activated, */usr/adm/errfile* is used. To simplify analysis of memory errors, single bit memory errors are reported in *eccfile* as ascii messages. No other analysis of the error records is done by *errdemon ;* that responsibility is left to *errpt (1M).* If *eccfile* is not specified then **/usr/adm/ecclog** is used. The error-logging daemons are terminated by sending them a software kill signal (see *kill (1)).* Only the super-user may start the daemons, and only one daemon per special file may be active at any time.

FILES

/dev/error	source of binary error records
/usr/adm/errfile	repository for binary error records
/dev/conslog	source of console output
/usr/adm/console_log	repository of console output
/usr/adm/ecclog	ascii log of single bit memory errors

DIAGNOSTICS

The diagnostics produced by *errdemon* are intended to be self-explanatory.

SEE ALSO

errpt(1M), errstop(1M), kill(1), err(7).

FILES

/etc/inittab
/etc/utmp
/etc/wtmp
/etc/ioctl.syscon
/dev/syscon
/dev/systty

SEE ALSO

getty(1M), login(1), sh(1), who(1), kill(2), inittab(4), utmp(4).

DIAGNOSTICS

If *init* finds that it is continuously respawning an entry from **/etc/inittab** more than 10 times in 2 minutes, it will assume that there is an error in the command string, and generate an error message on the system console, and refuse to respawn this entry until either 5 minutes has elapsed or it receives a signal from a user *init* (*telinit*). This prevents *init* from eating up system resources when someone makes a typographical error in the *inittab* file or a program is removed that is referenced in the *inittab.*

NAME

install – install commands

SYNOPSIS

/etc/install [**−c** dira] [**−f** dirb] [**−i**] [**−n** dirc] [**−o**] [**−s**] file [dirx ...]

DESCRIPTION

Install is a command most commonly used in "makefiles" (see *make (1))* to install a *file* (updated target file) in a specific place within a file system. Each *file* is installed by copying it into the appropriate directory, thereby retaining the mode and owner of the original command. The program prints messages telling the user exactly what files it is replacing or creating and where they are going.

If no options or directories (*dirx* ...) are given, *install* will search a set of default directories (**/bin**, **/usr/bin** , **/etc** , **/lib** , and **/usr/lib** , in that order) for a file with the same name as *file* . When the first occurrence is found, *install* issues a message saying that it is overwriting that file with *file* , and proceeds to do so. If the file is not found, the program states this and exits without further action.

If one or more directories (*dirx* ...) are specified after *file* , those directories will be searched before the directories specified in the default list.

The meanings of the options are:

−c *dira* — Installs a new command (*file*) in the directory specified by *dira* , only if it is not found. If it is found, *install* issues a message saying that the file already exists, and exits without overwriting it. May be used alone or with the **−s** option.

−f *dirb* — Forces *file* to be installed in given directory, whether or not one already exists. If the file being installed does not already exist, the mode and owner of the new file will be set to **755** and **bin**, respectively. If the file already exists, the mode and owner will be that of the already existing file. May be used alone or with the **−o** or **−s** options.

−i — Ignores default directory list, searching only through the given directories (*dirx* ...). May be used alone or with any other options other than **−c** and **−f** .

−n *dirc* — If *file* is not found in any of the searched directories, it is put in the directory specified in *dirc* . The mode and owner of the new file will be set to **755** and **bin**, respectively. May be used alone or with any other options other than **−c** and **−f** .

−o — If *file* is found, this option saves the "found" file by copying it to **OLD***file* in the directory in which it was found. This option is useful when installing a normally text busy file such as **/bin/sh** or **/etc/getty** , where the existing file cannot be removed. May be used alone or with any other options other than **−c** .

−s Suppresses printing of messages other than error messages. May be used alone or with any other options.

IMPORTANT

These commands are for source code customers only.

SEE ALSO

make(1).

NAME

uuxqt – execute remote command requests

SYNOPSIS

/usr/lib/uucp/uuxqt [**-s** system] [**-x** debug_level]

DESCRIPTION

uuxqt is the program that executes remote job requests from remote systems generated by the use of the *uux* command. (*Mail* uses *uux* for remote mail requests.) *uuxqt* searches the spool directories looking for *X*.files. For each *X*. file, *uuxqt* checks to see if all the required data files are available and accessible, and file commands are permitted for requesting system.

The **-x** *debug_level* is a single digit between 0 and 9. Higher numbers give more detailed debugging information.

FILES

/usr/lib/uucp/L.sys
/usr/lib/uucp/L.cmds
/usr/spool/uucp/*
/usr/spool/locks/LCK*

SEE ALSO

uucico(1M).
uucp(1C), uustat(1C), uux(1C), mail(1) in the *Sys5 UNIX User's Reference Manual*.

NAME

vconfig – virtual terminal configuration

SYNOPSIS

vconfig [–c] [–f] [l] [d] filename
vconfig [–p] [l] [d] vcp_no
vconfig [–s] [l] [d] major_device_no

DESCRIPTION

vconfig is a utility program which allows the system administrator to allocate a proper device name in the **/dev** directory to a physical port on a Plexus VCP communications controller. The three major functions are: to configure a kernel resident system device table, to obtain the types of communication ports in a VCP, to obtain information from kernel's system device table for a particular major device number.

COMMANDS

f — Fill the kernel system device table with the contents of the file named *filename*.

c — Configure the contents in *filename* with the information received from all the VCP's. The kernel system device table will only be updated if the entry in *filename* matches that returned by the VCP.

s — Show the content in the kernel system device table for a major device.

p — Show all of the physical ports available from each VCP.

OPTION

l — The **l** option causes the logging function to be turned on. The logfile is found in **/etc/vconfig/log**.

d — The same messages as in the **l** option are displayed to the user's terminal.

CAUTION

The **c** and **f** commands should be executed only during boot-up time, since any changes to the configuration file will potentially cause the loss of use of the already configured lines.

NAME

vcpdmp – dump contents of VMEbus Communication Processor's memory into a file

SYNOPSIS

/etc/vcpdmp /dev/vc*X file*

Where: *X* = 0, 1, 2, or 3

DESCRIPTION

vcpdmp reads the contents of a VMEbus Communication Processor's (VCP) memory and dumps it to the specified file. The VCP must be in the reset mode. That is, it has been reset with a system reset for **vcpdmp** to work properly. This command can only be executed in single-user state before the VCP is downloaded with a new kernel. The VCP device names are **/dev/vc0**, **/dev/vc1**, **/dev/vc2**, and **/dev/vc3**.

FILES

dataio(1m)

NOTES

This is a Plexus command. It is not part of standard *System V*.

BUGS

Ensure that there is enough disk space before running this program. It requires approximately 1Mb.

NAME

intro – introduction to special files

DESCRIPTION

This section describes various special files that refer to specific hardware peripherals and UNIX system device drivers. The names of the entries are generally derived from names for the hardware, as opposed to the names of the special files themselves. Characteristics of both the hardware device and the corresponding UNIX system device driver are discussed where applicable.

Tape device file names are in the following format:

/dev/{r}mt/(c#d)#[hml]{n}

where **r** indicates a raw device, **c#d** indicates the controller number (which is optionally specified by the system administrator), **#** is the device number, **hml** indicates the density (**h** (high) for 6250 bpi, **m** (medium) for 1600 bpi, and **l** (low density) for 800 bpi), and **n** indicates no rewind on close. (e.g., **/dev/mt/2mn**)

Disk device file names are in the following format:

/dev/{r}dsk/(r)(c#d)#s#

where **r** indicates a raw interface to the disk, the second **r** indicates that this disk is on a remote system, the **c#d** indicates the controller number (which is optionally specified by the system administrator), and **#s#** indicates the drive and section numbers, respectively.

BUGS

While the names of the entries *generally* refer to vendor hardware names, in certain cases these names are seemingly arbitrary for various historical reasons.

NAME

acp – Advanced Communications Processor

DESCRIPTION

The *acp* provides communications links between your Plexus system and serial devices (terminals and modems) and parallel devices (parallel printers). It allows segments of the operating system and customized communications and terminal handling programs to be downloaded to the ACP and executed locally.

Each ACP has 512Kb of memory, sixteen serial ports, and one parallel (Centronics-type) port. The serial ports are RS232C compatible and have the modem control lines necessary to support standard asynchronous or synchronous protocols at software-selectable rates up to 19.2K baud. All 16 ports can operate at the maximum rate simultaneously since each port has its own DMA channel.

FILES

/dev/ac[0-4]

SEE ALSO

acpdmp(1M), dnld(1M), tty(7).

NAME

ccb – common circuits board driver

DESCRIPTION

The *ccb* provides access to the functions of the common circuits board (P/75 only) or a limited set of commands simulating these functions (all other systems). *ioctl* calls to this device can be used to query or control processor action. The *ioctl* calls and structure are defined in **/usr/include/sys/ccb.h**.

SEE ALSO

sys(1M)

FILES

/dev/ccb

NAME

cd – configurable disk drive

DESCRIPTION

The configurable disk driver allows several logical disks to be joined into one logical disk which can then be larger than a physical disk. The configured device may be either striped (alternated) among its component logical disk partitions, or a concatenation of them.

For example, if a configured device consists of logical disks 1, 2, and 3, and it has a stripe size of 4K. Addresses in the configured device would then map to the disks as follows:

0x0000 – 0x0fff : disk 1 bytes 0x0000 –0x0fff

0x1000 – 0x1fff : disk 2 bytes 0x0000 – 0x0fff

0x2000 – 0x2fff : disk 3 bytes 0x0000 – 0x0fff

0x3000 – 0x3fff : disk 1 bytes 0x1000 – 0x1fff

0x4000 – 0x4fff : disk 2 bytes 0x1000 – 0x1fff

The **CD_CONFIG ioctl** call configures a device. It accepts file descriptor numbers returned by **open**(2) so that the configuration command can deal with file names rather than major and minor device numbers. This also insures that the devices have already been opened successfully. Its argument points to the following structure:

```
struct cd_conf {

    int    ssize;    /* Stripe size in 512 byte sectors. If zero there is no
                        striping and the partitions are simply concatenated to
                        define the configured device. This size should be at
                        least as big as the most common disk request. */

    int    pcnt;     /* The number of partitions being joined.

    struct part {     /* This structure defines the partitions being joined.
                       The order of this table defines the ordering in the
                       logical address space of the configured device. */

        int   file;     /* File descriptor returned by a call to open. The
                           file must be a special file for a block device. */

        int   offset;   /* The number of sectors at the beginning of the
                           partition to not use. Note that it is important that
                           physical sector 0 of a drive should not be included
                           in a configured device. Sector zero always contains
                           driver configuration information which must not be
                           damaged. */

        int   sectors;  /* The number of 512 byte sectors to use from this
                           file. If this is not an exact multiple of the stripe
                           size then the remainder will be waster. */

    } parts[MAXPART];

};
```

The **CD_DECONFIG ioctl** call deconfigures a device so that it may be reconfigured. It is necessary to close the configured device for a deconfigure to be complete. When the deconfigure call is made a flag is set to indicate that the device should be deconfigured on its next close. This flag is cleared on any opens. When the driver gets its close call it closes the slave devices. After that the deivce may be reconfigured. Since the close will not occur if there are multiple opens, it is not possible to reconfigure a device while it is in use as the deconfigure will have no effect, and the open to attempt a reconfigure will clear the deconfigure flag.

FILES

/dev/cd/[a-h]
/dev/rcd/[a-h]
/dev/plx/cdconf

SEE ALSO

cdconf(1M)

NAME

conslog – provides copy of error messages from UNIX to console

DESCRIPTION

The *conslog* device provides a copy of all error messages printed by UNIX to the console. Messages are generally read by *errdemon*(1M) and written to **/usr/adm/console_log**. Thus, a transcription of all system errors can be maintained.

NOTES

Only messages printed by the operating system show up in *conslog*; user error messages written to the console are not recorded.

BUGS

The error daemon must be started in *rc* or these error records will be lost.

FILES

/dev/conslog
/usr/adm/console_log transcription file

SEE ALSO

errdemon(1M).

NAME

dsk – (optionally) mirrored disk driver
rdsk –raw (unbuffered) version of dsk

DESCRIPTION

dsk and *rdsk* devices provide access to the default disk devices on a system. On systems with only one type of disk controller, these devices refer to that controller. Systems with more than one type of disk controller use this device to access the controller supporting mirroring (the EMSP in systems with both an EMSP and an IMSP).

Read and write operations to the raw (*rdsk*) interfaces must begin on a 512 byte boundary, and be in multiples of 512 bytes long.

I/O to the buffered (*dsk*) devices uses the UNIX buffering features, and thus may specify arbitrary block lengths and locations.

FILES

/dev/dsk/*X*s*X*
/dev/rdsk/*X*s*X*

SEE ALSO

pc(7).

NAME

err – error-logging interface

DESCRIPTION

Minor device 0 of the *err* driver is the interface between a process and the system's error-record collection routines. The driver may be opened only for reading by a single process with super-user permissions. Each read causes an entire error record to be retrieved; the record is truncated if the read request is for less than the record's length.

FILES

/dev/error special file

SEE ALSO

errdemon(1M).

NAME

ft – IMSP streaming cartridge controller

DESCRIPTION

This is a pseudo driver which will stream I/O between a cartridge tape drive and an IMSP-controlled disk. It uses a disk partition (logical disk) as a scratch buffer area. The disk partition is a small (1-2 megabyte) logical disk created using *dconfig*. It must not overlap a currently active file system. See Section 1M of this manual and the *UNIX Sys5 Administrator's Guide* for more information on *dconfig*.

By convention, the files **/dev/rft/0m** and **/dev/rft/0mn** are used to access the cartridge in streaming mode. Accessing with **/dev/rft/0m** rewinds the cartridge when this special file is closed. Accessing with **/dev/rft/0mn** does not rewind the cartridge when the file is closed.

The reads and writes take place asynchronously, occurring when the buffer area is filled. Therefore, errors which occur might not be reported until the tape device is closed. You must be careful not to attempt to write more to the tape cartridge than it can hold. Errors reported might relate either to the disk or the tape. Tape errors are described in *pt*(7), disk errors in *pd*(7).

The major number for these files is 25. The minor number for **/dev/rft/0m** is the same as the number of the disk partition you are using for the scratch buffer. The minor number for **/dev/rft/0mn** is gotten by adding 128 to the minor number of **dev/rft/0m**. For example, if you are using **/dev/dsk/0s15** for your scratch buffer area, the relevant minor number is 15. To create nodes for the two *ft* devices use the following commands:

mknod /dev/rft/0m c 25 15
mknod /dev/rft/0mn c 25 143 (128 + 15)

FILES

/dev/rft/0m
/dev/rft/0mn
/dev/dsk/?s?

WARNING

Be very careful that the disk partition used for the scratch buffer does not overlap an active file system. File system corruption might occur if it does.

Do not attempt to write more than a cartridge can hold.

This device will not work with disks controlled by an EMSP.

SEE ALSO

dconfig(1M), mknod(1M), pd(7), pt(7).

NAME

icp – Intelligent Communications Processor

DESCRIPTION

The *icp* is a special device that allows access to the memory of the Intelligent Communications Processor (ICP). Reading from the device resets the ICP. Writing to the device overwrites the memory.

The ICP provides communication links between your Plexus system and serial devices (terminals and modems) and parallel devices (parallel printers). It is a 16-bit processor module designed to handle serial and parallel I/O tasks in Plexus systems. The ICP provides the buffering and processing required to support high-speed communications with terminals, modems, printers, and other serial devices.

The ICP is controlled by a 16-bit processor with 48Kb of memory. Each ICP controls eight RS232C serial ports and one parallel (Centronics-type) port. Each serial port has full modem support and a maximum transfer rate of 19.2K baud. The serial ports are capable of supporting the asynchronous and bisynchronous protocols.

FILES

/dev/ic[0-4]

BUGS

Reading from the ICP resets it and kills all terminals actively using it.

SEE ALSO

dnld(1M), icpdmp(1M), tty(7).

NAME

imsp – Intelligent Mass Storage Processor

DESCRIPTION

The *imsp* is a special device that allows access to the memory of the Intelligent Mass Storage Processor (IMSP). Reading from the device returns data from the IMSP's local memory. Writing to the device overwrites the IMSP's local memory.

The IMSP is an intelligent disk and tape controller that contains its own Z8001 microprocessor. It receives commands from the CPU to move blocks of data between system memory (RAM) and the disk drives or cartridge tape drive. The processor's 256Kb address space is organized as follows: 16Kb local ROM, 128Kb local RAM, and 64Kb shared RAM. The 128Kb of local RAM is used to buffer a number of sectors, to decrease the number of disk accesses when the system experiences a heavy processing load. These buffers store the information from the disk and pass it to each process as if it were the only process using disk.

The IMSP uses an industry-standard SMD type disk interface and also controls the cartridge tape drive (QIC02 or QIC24 type cartridge tape interface). The intelligent cartridge tape drive performs many of the functions normally required of a tape controller. it communicates with the IMSP over eight data lines and eight control lines.

FILES

/dev/im[0-3]
/dev/pd/*XsX*
/dev/dsk/*XsX*

BUGS

Writing to the IMSP can cause it to hang. This may crash UNIX and destroy file systems.

NAME

null – the null file

DESCRIPTION

Data written on a null special file is discarded.

Reads from a null special file always return 0 bytes.

FILES

/dev/null

NAME

od – optical disk rod – raw interface to optical disk

DESCRIPTION

The Plexus optical disk is a Write-Once Read-Many (WORM) optical disk designed to archive and retrieve very large amounts of data (1Gbyte per disk side). It can access up to four standalone optical disk drives.

ioctl calls and return structures are defined in **/usr/include/sys/od.h.**

FILES

/dev/od
/dev/rod
/usr/include/sys/od.h

SEE ALSO

odconf(1M), odls(1M), and odstat(1M) in your *Optical Disk User's Manual*.

NAME

pd – IMSP disk controller

DESCRIPTION

The IMSP disk/tape controller and associated driver code access up to four disks. Each disk is subdivided into 16 logical volumes. By convention, /dev/dsk/0s[0-15] refer to the logical volumes of physical disk 0, /dev/dsk/1s[0-15] refer to the logical volumes of physical disk 1, and so on.

The *dsk* files access the disk via the systems's normal buffering mechanism and may be read and written without regard to physical disk records. There is also a 'raw' interface which provides for direct transmission between the disk and the user's read or write buffer. A single read or write call results in exactly one I/O operation and therefore raw I/O is considerably more efficient when many words are transmitted. The names of the raw *pd* files begin with *rdsk* and end with a number that selects the same logical disk volume as the corresponding *dsk* file.

In raw I/O the buffer must begin on a 512 byte boundary.

FILES

/dev/dsk
/dev/im0

NOTES

This is a Plexus device. It is not part of standard *System V.*

SEE ALSO

imsp(7).

DIAGNOSTICS

The IMSP controller may produce the following error messages:

0x0201	Reserved for controller busy
0x0301	Command undefined
0x0401	Command cannot be done
0x0501	Bad CAB parameters
0x0f01	Firmware bug encountered
0x0601	Internal command interrupts
0x0701	Parity error occurred
0x0801	PROM checksum error
0x1103	Disk protected from writing
0x1203	Disk not ready
0x1303	Disk drive fault indicated
0x1403	Disk failed to select
0x1503	Disk operation timeout error
0x1603	Disk failed in formatting
0x1703	Disk seek error

0x1803	Disk ECC error in id field
0x1903	Disk ECC error in data field
0x1b03	Disk limits not defined
0x1c03	Disk unable to locate track

NAME

pp – parallel port interface

DESCRIPTION

The parallel port interface enables access to the parallel port on the Intelligent Communications Processor (ICP) or the Advanced Communications Processor (ACP). Each ACP and ICP has one parallel port interface. The parallel port interface is a write-only device. It is also a raw device, i.e., the operating system does no processing of data written to it.

Pp has no *stty*-like features. If your printer does not handle tabs and new-line characters, you need to write a filter to use this device.

FILES

/dev/pp[0-3]

SEE ALSO

acp(7), icp(7), lp(7), tty(7).

NAME

prf – operating system profiler

DESCRIPTION

The file provides access to activity information in the operating system. Writing the file loads the measurement facility with text addresses to be monitored. Reading the file returns these addresses and a set of counters indicative of activity between adjacent text addresses.

The recording mechanism is driven by the system clock and samples the program counter at line frequency. Samples that catch the operating system are matched against the stored text addresses and increment corresponding counters for later processing.

The file is a pseudo-device with no associated hardware.

FILES

/dev/prf

SEE ALSO

profiler(1M).

NAME

pt – IMSP cartridge controller

DESCRIPTION

The IMSP disk/tape controller and associated driver code allow access to a cartridge tape. The cartridge can be accessed only in raw mode (i.e., as a character device), and can be rewound or left at the current position. These options are available based on the minor device number of the special file used to access it. If the cartridge is not to be rewound, it is positioned after the filemark at the end of the current file.

If the 04 bit is on in the minor device number, the cartridge is not rewound when closed.

By convention, the files **/dev/rpt/0m** and **/dev/rpt/0mn** are used to access the cartridge in raw mode. Accessing **/dev/rpt/0m** rewinds the cartridge when this special file is closed. Accessing **/dev/rpt/0mn** does not rewind the cartridge when the file is closed. Each *read* or *write* call reads or writes the next record on the cartridge. All records on a cartridge are 512 bytes long and all reads and writes must be in multiples of 512 bytes. An error is returned otherwise. The I/O buffer used in the *read*(2) or *write*(2) system call should begin on a word boundary and the count should be even. Seeks are ignored. A zero byte count is returned when a file mark is read, but another read will fetch the first record of the new file.

The cartridge drive can be accessed in high speed mode. However, this mode is effectively limited to skipping forward over files on the cartridge and to I/O between the cartridge and a disk attached to the same IMSP controller. High speed mode is accessed via *ioctl*(2) system calls. The arguments to the *ioctl* are:

fildes File descriptor returned from an *open* (2) of the special tape file **/dev/rpt/0m** or **/dev/rpt/0mn**.

request A special command for the cartridge drive. These commands are defined in **/usr/include/sys/imsc.h** and some are described below.

arg A pointer to a structure of the type "ptcmd" as defined in **/usr/include/sys/imsc.h**.

Some of the members of *ptcmd* are:

dknum Major/minor device number of the IMSP disk being read or written to (if applicable) as returned by *stat*(2) system call (st_rdev).

blkno Starting sector number on logical disk to be read/written. Sectors on disk are 512 bytes long and numbered starting at 0. Note sector addresses are relative to the logical, not the physical disk.

blkcnt The number of 512-byte records to be read from or written to cartridge.

Some of the more useful *ioctl* requests for the cartridge as defined in **/usr/include/sys/imsc.h** are:

C_IRECALL Read from cartridge and write to disk. The cartridge and disk must be on same IMSP controller. The system returns in *ptcmd.blkcnt* the number of 512-byte records not read. This is zero if the system reads all the records requested.

C_ISAVE Read from disk and write to tape. The cartridge and disk must be on same IMSP controller. The system returns in *ptcmd.blkcnt* the number of 512-byte records not read. This is zero if the system reads all the record images (sectors) requested.

C_IWEOF Write EOF mark on cartridge.

C_IREW Rewinds the cartridge.

C_IMOVE Position to file *blkcnt* on cartridge.

Writing multiple files on cartridge should be done all at once, i.e., without rewinding the cartridge. Once a cartridge has been rewound, positioning to the end of a file on the cartridge and then writing to the cartridge may overwrite data. For example, once the cartridge has been rewound, positioning to the end of file 2 and writing to the cartridge may overwrite portions of file 2.

Neither the hardware or the software implement or support an end-of-tape marker on the cartridge.

FILES

/dev/rpt/0m generic
/dev/rpt/0mn no rewind

SEE ALSO

rmt(7).

DIAGNOSTICS

The IMSP controller produces error diagnostics in the following form:

sys3: error on PT, minor 0
sys3: bn = *bbbb* er = 0x*nnnn*, 0x*mmmm*

where *bbbb* is a block number. The first set of "er" numbers (*nnnn*) gives status. The second set of "er" numbers (*mmmm*) describes errors. Each set of "er" numbers is discussed separately below.

Status Bytes

There are two meaningful bytes of status (*nnnn*); these are the third and fourth bytes of a 32 bit word. Because the status representation is "zero-true", if the third byte is all ones, the system construes the entire word as a negative number and prepends "ffff" to the two status bytes. This leading "ffff" can be ignored.

The meaning of each bit of the status bytes is listed below. Examples follow.

Byte 0

Bit 7	Status byte 0 contains information
Bit 6	Cartridge not in place
Bit 5	Drive not online
Bit 4	Write protected
Bit 3	End of media
Bit 2	Unrecoverable data error
Bit 1	BIE not located
Bit 0	File mark detected

Byte 1

Bit 7	Status byte 1 contains information
Bit 6	Illegal command
Bit 5	No data detected
Bit 4	8 or more read retries
Bit 3	Beginning of media
Bit 2	Reserved
Bit 1	Reserved
Bit 0	Reset/Power-up occurred

For example, the error

```
sys3: error on PT, minor 0
sys3: bn = 2345 er = 0x7b77, 0x1604
```

shows two bytes of status. The first byte is "7b", which means (remember zero indicates true) status byte 0 is meaningful and unrecoverable data error. The second byte is "77", which means status byte 1 is meaningful and beginning of media.

The second "er" number (0x1604) is described below.

The error

```
sys3: error on PT, minor 0
sys3: bn = bbbb er = 0xffffff76, 0x1604
```

shows the first byte of status to be "ff" (status byte 0 contains no information). The second byte is "76", which means status byte 1 contains information, and illegal command. The first four "f's" result from the system construing the status word as negative; they can be ignored.

Error Bytes

The following list shows the possible values for the error status (the second "er" number, or *mmmm* above):

0x0201	Reserved for controller busy
0x0301	Command undefined
0x0401	Command cannot be done
0x0501	Bad CAB parameters

0x0f01	Firmware bug encountered
0x0601	Internal command interrupts
0x0701	Parity error occurred
0x0801	PROM checksum error
0x1004	End of file reached
0x1304	An exception other than an end-of-file error
0x1504	Tape timeout error
0x1604	Error during recall
0x1704	Error during save
0x1804	Error received while attempting to get status from the tape drive
0x1904	During exception state, a command other than *rstat* was received
0x2004	No tape drive present
0x2104	Timeout during wait recall
0x2204	Timeout during wait save
0x2304	Timeout during stat tape
0x2404	Timeout during stat tape
0x2504	Timeout during command tape
0x2604	Timeout during command tape
0x2704	Timeout during ready tape
0x2804	Tape drive inconsistent at start of tape command
0x1505	Timeout on Host bus request

NAME

crm – caching reel-to-reel tape driver
rrm – raw reel-to-reel tape driver

DESCRIPTION

crm

crm provides access to a caching reel-to-reel tape driver. It uses an in-core cache to allow accesses to continue at near-streaming tape speeds. See FILES, below, for the list of acceptable *crm* devices.

The major device number for *crm* is 28. The minor device number is the same as for the standard tape driver.

An open will fail if any other tape device is open (ENXIO). Once opened, no other tape drives can be opened. Open may fail because the cache could not be dynamically allocated. If there is insufficient idle memory, the open will return ENOMEM. This condition may be temporary in nature, and the open may succeed if system activity is reduced.

crm does not support any *ioctl* calls.

Write errors will not be reported until the write physically happens. Thus, the call getting the error may be as much as 128K bytes beyond the occurrence of the problem. An error may never be reported if the close occurs before the physical write. This does not apply to the EOT error. Records will be written beyond the EOT mark. The record that is rejected for EOT will not be written to tape, but an EOF will be written on the close.

Tapes with variable record sizes may not be readable. The first record read after open or encountering an EOF mark is used to determine the size of all subsequent reads. All records to the next EOF must be equal to or less than this first record's length. Standard UNIX utilities write fixed length records, so this limitation should not be a problem.

If read returns an EIO error because of a bad spot on the tape, more reads may be issued to continue after the bad record.

rrm

The Cipher Microstreamer magnetic tape can be accessed in blocked or raw mode and can be rewound or left at the current position. These options are available based on the minor device number of the special file used to access it. When the special file is closed, the tape can be rewound or not (see below). If the special file was open for writing, two end-of-files are written. If the tape is not to be rewound, it is positioned with the head between the two tapemarks.

If the 04 bit is on in the minor device number, the tape is not rewound when closed.

If the 010 bit is on in the minor device number, the tape is set to high speed mode (100 in/sec). By convention, **/dev/rrm/0mn** accesses the tape in high speed mode.

By convention, the file **/dev/mt0** accesses the tape in blocked mode. A tape accessed in block mode consists of a series of 1024-byte records terminated by an end-of-file. As much as it can, the system makes it possible, if inefficient, to treat the tape like any other file. Seeks have their usual meaning and it is possible to read or write a byte at a time. Writing in very small units is inadvisable, however, because it tends to create monstrous record gaps.

Use **/dev/mt0** to access the tape in a way compatible with ordinary files. However, when foreign tapes are to be dealt with, and especially when long records are to be read or written, the 'raw' interface is more appropriate. By convention, the files **/dev/rpt/0m** and **/dev/rpt/0mn** are used to access the tape in raw mode. Accessing **/dev/rpt/0m** rewinds the tape when **/dev/rpt/0mn** is closed. Accessing **/dev/rpt/0mn** does not rewind the tape when **/dev/rpt/0mn** is closed.

Each *read* or *write* call reads or writes the next record on the tape. For writes, the record has the same length as the buffer given. During a read, the record size is passed back as the number of bytes read, provided it is no greater than the number of bytes requested; if the record is longer than the number of bytes requested, an error is returned. On the other hand, if the number of bytes requested is larger than the actual record size, there is a delay of 1-2 seconds between the reading of each record.

In raw tape I/O, the buffer must begin on a word boundary and the count must be even. Seeks are ignored. A zero byte count is returned when a tape mark is read, but another read will fetch the first record of the next tape file.

The tape drive can be run in high speed mode; however, this is really only usable for fast forward or reverse skipping of file marks. The files used for high speed mode are denoted by an 'h' just before the unit number.

If you want to write your own program for tape manipulation on the *rm* device, there is an *ioctl*(2) interface for controlling the tape drive. The file */usr/include/sys/rm.h* lists the commands that can be issued. These all begin with "C_" (capital C followed by an underbar). The only *ioctl* request type allowed for this device is RMPOSN ("*rm* position"). The *ioctl* call structure is

```
struct rmcmd_struct {
  unsigned rm_cmd;      /* the command C_<option> */
  unsigned rm_cnt;      /* count, useful for commands
                           such as SRCHEOF */
  unsigned rm_status;   /* physical device status returned */
  };
```

The status value is found by adding all the relevant values in the "status fields" portion of *rm.h*. Status is determined by the output status field, which consists of two bytes arranged as follows:

15	14	13	12 11 10 9 8	7	6	5	4	3	2	1
E	C	R	ERROR	FM	OL	LP	EOT	R	FB	P

where

Byte 0	Not used
P	(Write Protect) The tape does not have a write enable ring.
FB	(Formatter Busy) The Formatter is busy.
R	(Ready) The selected drive is ready.
EOT	(End of Tape) The EOT marker was detected.
LP	(Load Point) The tape is at load point.
OL	(On Line) The drive is on line.
FM	(Filemark) A filemark was detected on this operation.
E	(Entered) Execution has begun.
C	(Complete) The command has completed successfully.
R	(Retry) At least one Retry was executed .
ERROR	This 5-bit field specifies an error code when a non-recoverable error is encountered. Error codes are listed under DIAGNOSTICS below.

For example, the value "C068" means the tape is online at load point, ready, and previous command has completed.

The following program fragment illustrates the use of **ioctl** to rewind the tape.

```
#include "sys/rm.h"
#include "fcntl.h"
int fildes;     /* file descriptor, returned by open */

fildes = open("/dev/rpt/0m",O_RDWR);

rmcmd.cmd = C_REW;
rmcmd.cnt = 1;
rmcmd.status = -1;

ioctl(fildes, RMPOSN, &rmcmd);
```

FILES

/dev/crm/0m	
/dev/crm/0mn	(no rewind)
/dev/rrm/0m	
/dev/rrm/0hm	(high speed)
/dev/rrm/0mn	(no rewind)
/dev/rrm/0hmn	(no rewind, high speed)
/usr/include/sys/rm.h	

SEE ALSO

tape(1), rm(7).
ioctl(2) in the *Sys5 UNIX Programmer's Reference Manual.*

DIAGNOSTICS

The tape controller issues the following codes for unrecoverable errors detected during execution of a command. The code is returned in the Command Status byte, bits 8-12.

Code	Description
00	No unrecoverable error.
01	Timed out waiting for expected Data Busy false.
02	Timed out waiting for expected Data Busy false, Formatter Busy false and Ready True.
03	Timed out waiting for expected Ready false.
04	Timed out waiting for expected Ready true.
05	Timed out waiting for expected Data Busy true.
06	A memory time-out occurred during a system memory reference.
07	A blank tape was encountered where data was expected.
08	An error occurred in the micro-diagnostic.
09	An unexpected EOT was encountered during a forward operation, or Load Point during a reverse operation.
0A	A hard or soft error occurred that could not be eliminated by retry.
0B	A read overflow or write overflow occurred. This error indicates that the FIFO was empty when data was requested by the tape during a write, or full when the tape presented a byte during a read.
0C	Not used.
0D	A read parity error occurred on the byte interface between the drive and the controller.
0E	An error was detected during calculation of the checksum on the PROM.
0F	A tape time-out occurred, because the tape drive did not supply an expected read or write strobe. This error occurs when you attempt to read a larger record than was written. It may also occur during a write if the tape is damaged.

10 Tape not ready.

11 A write was attempted on a tape without a write-enable ring.

12 Not used.

13 The diagnostic mode jumper was not installed while attempting to execute a Diagnostic command.

14 An attempt was made to link from a command that does not allow linking.

15 An unexpected filemark was encountered during a tape read.

16 An error in specifying a parameter was detected by the controller. The usual cause is a byte count that is either zero or too large.

17 Not used.

18 An unidentifiable hardware error occurred.

19 A streaming read or write operation was terminated by the operating system or disk.

The tape driver sends the code FFFF to the screen when the block size requested is smaller than the actual block size on the tape.

BUGS

Some other systems (including previous releases of Plexus software) will not be able to read records written beyond the EOT, but both Plexus Sys5 tape drivers (**rrm** and **crm**) will read beyond EOT.

NAME

rram, ram – allows memory to be used as a disk.

DESCRIPTION

Allocates memory to one of up to eight devices which allow the memory to be used as a disk.

The directory **/dev/rram** contains the raw devices used as parameters to the utility, and **/dev/ram** contains block devices which can be configured to be mounted file systems.

Nodes in **/dev/rram** are character type (**c**) devices. The major number to use is 7. Minor numbers range from 0 to 7.

Nodes in **/dev/ram** are block type (**b**) devices. The major number is 3.

Blocks from these devices do not stay in the buffer pool. Their buffers are reused immediately to allow the buffer pool to be used by disk devices.

Usage is as follows:

mknod /dev/ram/*devname* **b 3** *devnumber*

mknod /dev/rram/*devname* **c 7** *devnumber*

FILES

/dev/ram/*devname*
/dev/rram/*devname*

SEE ALSO

mknod(1m), ramdisk(1m).

NAME

swap – swap device

DESCRIPTION

swap is a block special device that corresponds to the file system containing the swap area (default **/dev/dsk/0s2**. Reading from the *swap* device returns data from the swap area.

swap is used by the command *ps*(1) to read the data from swapped processes.

BUGS

The device numbers for **/dev/swap** must agree with what was entered in *dconfig* for *swapdev*.

FILES

/dev/swap

PLEXUS COMPUTERS

Reader Comment Form

Company Name :

Your Name (Optional) :

Manual Name : **Sys5 UNIX Administrator's Reference Manual**

Publication Number : **98-05084.1 Ver. F**

Please let us know if anything in this manual is unclear, incomplete, or inaccurate.

1. Should any information be included or removed?

____

____

2. Please specify the page and nature of any error(s) found in this document.

____

____

3. Other Comments

____

____

____

Please mail this form to :

Technical Publications Department
Plexus Computers, Inc.
3833 North First St.
San Jose, CA 95134

PERMUTED INDEX

make a delta	(change) to an SCCS file delta	delta(1)
edit text editor	(variant of ex for casual users)	edit(1)
status program.	/etc/sys System control and	sys(1m)
a isatty returns a	1 if specified file descriptor is	isatty(2s)
handle special functions of HP	2640 and 2621-series terminals hp	hp(1)
comparison diff3	3-way differential file	diff3(1)
handle special functions of DASI	300 and 300s terminals 300	300(1)
dial dial a Racal-Vadic	3451 modem	dial(1)
for the TEKTRONIX 4014 terminal	4014 paginator	4014(1)
of the DASI 450 terminal	450 handle special functions	450(1)
Fortran	77 compiler f77	f77(1)
troff documents for the Xerox	9700 printer dx9700 prepare	dx9700(1)
nroff documents for the Xerox	9700 printer x9700 prepare	x9700(1)
asa interpret	ASA carriage control characters	asa(1)
acpdmp dump contents of	Advanced Communication	acpdmp(1m)
maintainer bar	Berkeley archive and library	bar(1)
	C compiler cc	cc(1)
programs scc	C compiler for stand-alone	scc(1)
generate	C flow graph cflow	cflow(1)
the	C language preprocessor cpp	cpp(1)
	C program beautifier cb	cb(1)
a	C program checker lint	lint(1)
generate	C program cross-reference cxref	cxref(1)
	C program debugger ctrace	ctrace(1)
report	CPU time used clock	clock(3C)
optimization package curses	CRT screen handling and	curses(3X)
rm	Cipher Microstreamer tape drive	rm(7)
dump contents of Advanced	Communication acpdmp	acpdmp(1m)
icp Intelligent	Communications Processor	icp(7)
handle special functions of the	DASI 450 terminal 450	450(1)
generate	DES encryption crypt	crypt(3C)
exits. write	EOT on the other terminal and	write(1)
	Euclidean distance function hypot	hypot(3M)
	Extended Fortran Language efl	efl(1)
	Fortran 77 compiler f77	f77(1)
Extended	Fortran Language efl	efl(1)
	Fortran absolute value abs	abs(3F)
system signal signal specify	Fortran action on receipt of a	signal(3F)
function acos	Fortran arccosine intrinsic	acos(3F)
functio asin	Fortran arcsine intrinsic	asin(3F)
function atan2	Fortran arctangent intrinsic	atan2(3F)
function atan	Fortran arctangert intrinsic	atan(3F)
functions bool	Fortran bitwise boolean	bool(3F)
getarc return	Fortran command-line argument	getarc(3F)
intrinsic function log10	Fortran common logarithm	log10(3F)
intrinsic function conjg	Fortran complex conjugate	conjg(3F)
functions cos	Fortran cosine intrinsic	cos(3F)
rational	Fortran dialect ratfor	ratfor(1)
getenv return	Fortran environment variable	getenv(3F)
function exp	Fortran exponential intrinsic	exp(3F)
intrinsic function cosh	Fortran hyperbolic cosine	cosh(3F)
function sinh	Fortran hyperbolic sine intrinsic	sinh(3F)
intrinsic function tanh	Fortran hyperbolic tangent	tanh(3F)

complex argument aimag	Fortran imaginary part of	aimag(3F)
function aint	Fortran integer part intrinsic	aint(3F)
max	Fortran maximum-value functions	max(3F)
min	Fortran minimum-value functions	min(3F)
intrinsic function log	Fortran natural logarithm	log(3F)
functions round	Fortran nearest integer	round(3F)
terminate	Fortran program abort	abort(3F)
functions mod	Fortran remaindering intrinsic	mod(3F)
sin	Fortran sine intrinsic function	sin(3F)
function sqrt	Fortran square root intrinsic	sqrt(3F)
return length of	Fortran string len	len(3F)
return location of	Fortran substring index	index(3F)
issue a shell command from	Fortran system	system(3F)
function tan	Fortran tangent intrinsic	tan(3F)
return	Fortran time accounting mclock	mclock(3F)
intrinsic function sign	Fortran transfer-of-sign	sign(3F)
explicit	Fortran type conversion ftype	ftype(3F)
hp handle special functions of	HP 2640 and 2621-series terminals	hp(1)
disk accounting data by user	ID diskusg diskusg - generate	diskusg(1m)
set process group	ID setpgrp	setpgrp(2)
getpid get process	ID	getpid(2s)
and real and effective group	ID get real and effective user,	getuid(2s)
real group, and effective group	ID's getuid effective user,	getuid(2)
print user and group	IDs and names id	id(1)
process group, and parent process	IDs getpid get process,	getpid(2)
set user and group	IDs setuid	setuid(2)
pt	IMSP cartridge controller	pt(7)
controller ft	IMSP streaming cartridge	ft(7)
generate an	IOT fault abort	abort(3C)
icpdmp dump contents of an	Intelligent Communication	icpdmp(1m)
Processor icp	Intelligent Communications	icp(7)
connection information	L-devices link devices,	L-devices(4)
abbreviations file	L-dialcodes alphabetic dialing	L-dialcodes(4)
commands	L.cmds remote execution	L.cmds(4)
	L.sys link systems	L.sys(4)
send/cancel requests to an	LP line printer lp	lp(1)
enable/disable	LP printers enable	enable(1)
requests start/stop the	LP request scheduler and move	lpsched(1m)
configure the	LP spooling system lpadmin	lpadmin(1m)
print	LP status information lpstat	lpstat(1)
Extended Fortran	Language efl	efl(1)
documents mm the	MM macro package for formatting	mm(5)
documents formatted with the	MM macros mm print/check	mm(1)
rm Cipher	Microstreamer tape drive	rm(7)
formatting documents mosd the	OSDD adapter macro package for	mosd(5)
mdial dial the	P/75 onboard modem	dial(1)
execute a command on the	PCL network net	net(1)
icp Intelligent Communications	Processor	icp(7)
dial dial a	Racal-Vadic 3451 modem	dial(1)
change the delta commentary of an	SCCS delta cdc	cdc(1)
combine	SCCS deltas comb	comb(1)
make a delta (change) to an	SCCS file delta	delta(1)
print current	SCCS file editing activity sact	sact(1)
get a version of a	SCCS file get	get(1)
print an	SCCS file prs	prs(1)

remove a delta from an	SCCS file rmdel	rmdel(1)
compare two versions of an	SCCS file sccsdiff	sccsdiff(1)
format of	SCCS file sccsfile	sccsfile(4)
undo a previous get of an	SCCS file unget	unget(1)
validate	SCCS file val	val(1)
create and administer	SCCS files admin	admin(1)
identify	SCCS files what	what(1)
sys	System control and status program	sys(1m)
program. /etc/sys	System control and status	sys(1m)
status program	System control and	sys(1m)
paginator for the	TEKTRONIX 4014 terminal 4014	4014(1)
get name from	UID getpw	getpw(3C)
UNIX system to	UNIX system copy uucp	uucp(1)
call another	UNIX system cu	cu(1)
filesave daily/weekly	UNIX system file system backup	filesave(1m)
uucp	UNIX system to UNIX system copy	uucp(1)
print name of current	UNIX system uname	uname(1)
get name of current	UNIX system uname	uname(2)
execution uux	UNIX-to-UNIX system command	uux(1)
uuto public	UNIX-to-UNIX system file copy	uuto(1)
permissions file	USERFILE UUCP pathname	USERFILE(4)
USERFILE	UUCP pathname permissions file	USERFILE(4)
prepare troff documents for the	Xerox 9700 printer dx9700	dx9700(1)
prepare nroff documents for the	Xerox 9700 printer x9700	x9700(1)
	a C program checker lint	lint(1)
modest-sized programs bs	a compiler/interpreter for	bs(1)
view graphs mv	a macro package for making	mv(7)
typesetting viewgraphs and mv	a troff macro package for	mv(5)
assembler and link editor output	a.out common	a.out(4)
integer and base-64 ASCII string	a64l convert between long	a64l(3C)
L-dialcodes alphabetic dialing	abbreviations file	L-dialcodes(4)
generate an IOT fault	abort	abort(3C)
terminate Fortran program	abort	abort(3F)
return integer absolute value	abs	abs(3C)
Fortran absolute value	abs	abs(3F)
	absolute debugger adb	adb(1)
return integer	absolute value abs	abs(3C)
Fortran	absolute value abs	abs(3F)
floor, ceiling, remainder,	absolute value functions floor	floor(3M)
a file touch update	access and modification times of	touch(1)
utime set file	access and modification times	utime(2)
of a file	access determine accessibility	access(2s)
commands graphics	access graphical and numerical	graphics(1)
machine-independent fashion.	access long integer data in a	sputl(3X)
disk	access profiler sadp	sadp(1m)
common object file	access routines ldfcn	ldfcn(4)
copy file systems for optimal	access time dcopy	dcopy(1m)
	access utmp file entry getut	getut(3C)
determine accessibility of a file	access	access(2)
determine	accessibility of a file access	access(2)
access determine	accessibility of a file	access(2s)
enable or disable process	accounting acct	acct(2)
connect-time	accounting acctcon	acctcon(1m)
of accounting and miscellaneous	accounting commands acct	acct(1m)
diskusg - generate disk	accounting data by user ID	diskusg(1m)

per-process	accounting file format acct	acct(4)
search and print process	accounting file(s) acctcom	acctcom(1)
merge or add total	accounting files acctmerg	acctmerg(1m)
return Fortran time	accounting mclock	mclock(3F)
command summary from per-process	accounting records acctcms	acctcms(1m)
run daily	accounting runacct	runacct(1m)
or disable process accounting	acct enable	acct(2)
miscellaneous accounting commands	acct overview of accounting and	acct(1m)
accounting file format	acct per-process	acct(4)
per-process accounting records	acctcms command summary from	acctcms(1m)
print process accounting file(s)	acctcom search and	acctcom(1)
connect-time accounting	acctcon	acctcon(1m)
or add total accounting files	acctmerg merge	acctmerg(1m)
arccosine intrinsic function	acos Fortran	acos(3F)
Advanced Communication	acpdmp dump contents of	acpdmp(1m)
signal signal specify Fortran	action on receipt of a system	signal(3F)
kill all	active processes killall	killall(1m)
system	activity graph sag	sag(1)
system	activity report package sar	sar(1m)
system	activity report sail	sail(1)
print current SCCS file editing	activity sact	sact(1)
report process data and system	activity timex time a command;	timex(1)
formatting mosd the OSDD	adapter macro package for	mosd(5)
absolute debugger	adb	adb(1)
acctmerg merge or	add total accounting files	acctmerg(1m)
change or	add value to environment putenv	putenv(3C)
create and	administer SCCS files admin	admin(1)
part of complex argument	aimag Fortran imaginary	aimag(3F)
integer part intrinsic function	aint Fortran	aint(3F)
set a process	alarm clock alarm	alarm(2)
change data segment space	allocation brk	brk(2)
brk change data segment space	allocation	brk(2s)
main memory	allocator malloc	malloc(3C)
fast main memory	allocator malloc	malloc(3X)
disk rram	allows memory to be used as a	rram(7)
abbreviations file L-dialcodes	alphabetic dialing	L-dialcodes(4)
of a document	analyze surface characteristics	style(1)
sort	and/or merge files sort	sort(1)
common archive file format	ar	ar(4)
maintainer for portable archives	ar archive and library	ar(1)
language bc	arbitrary-precision arithmetic	bc(1)
Fortran	arccosine intrinsic function acos	acos(3F)
for portable archives ar	archive and library maintainer	ar(1)
bar Berkeley	archive and library maintainer	bar(1)
format of cpio	archive cpio	cpio(4)
common	archive file format ar	ar(4)
archive header of a member of an	archive file ldahread read the	ldahread(3X)
common archive format convert	archive files	arcv(1)
files	archive format arcv archive	arcv(1)
archive file ldahread read the	archive header of a member of an	ldahread(3X)
tape file	archiver tar	tar(1)
library maintainer for portable	archives ar archive and	ar(1)
copy file	archives in and out cpio	cpio(1)
Fortran	arcsine intrinsic functio asin	asin(3F)
atan2 Fortran	arctangent intrinsic function	atan2(3F)

atan Fortran	arctangert intrinsic function	atan(3F)
archive format	arcv convert archive files from	arcv(1)
Fortran imaginary part of complex	argument aimag	aimag(3F)
return Fortran command-line	argument getarc	getarc(3F)
handle variable	argument list varargs	varargs(5)
formatted output of a varargs	argument list vprintf print	vprintf(3S)
command xargs construct	argument list(s) and execute	xargs(1)
get option letter from	argument vector getopt	getopt(3C)
evaluate	arguments as an expression expr	expr(1)
echo	arguments echo	echo(1)
display a program name and get	arguments for getargv	getargv(2s)
arbitrary-precision	arithmetic language bc	bc(1)
provide drill in number facts	arithmetic	arithmetic(6)
map of ASCII character set	ascii	ascii(5)
Fortran arcsine intrinsic functio	asin	asin(3F)
	ask for help help	help(1)
help	ask for help	help(8)
a.out common	assembler and link editor output	a.out(4)
common	assembler as	as(1)
verify program	assertion assert	assert(3X)
setbuf	assign buffering to a stream	setbuf(3S)
arctangent intrinsic function	atan2 Fortran	atan2(3F)
	await completion of process wait	wait(1)
scanning and processing language	awk pattern	awk(1)
the game of	backgammon back	back(6)
UNIX system file system	backup filesave daily/weekly	filesave(1m)
fast incremental	backup finc	finc(1m)
fbackup fast tape	backup of a file system	fbackup(1m)
fbackup make a fast tape	backup of a file system	fbackup(8)
recover files from a	backup tape frec	frec(1m)
make posters	banner	banner(1)
library maintainer	bar Berkeley archive and	bar(1)
ttytype data	base of terminal types by port	ttytype(5)
terminal capability data	base terminfo	terminfo(4)
convert between long integer and	base-64 ASCII string a64l	a64l(3C)
(visual) display editor	based on ex vi screen-oriented	vi(1)
deliver portions of path names	basename	basename(1)
arithmetic language	bc arbitrary-precision	bc(1)
big diff	bdiff	bdiff(1)
C program	beautifier cb	cb(1)
su	become super-user or another user	su(1)
big file scanner	bfs	bfs(1)
	big diff bdiff	bdiff(1)
	big file scanner bfs	bfs(1)
install object files in	binary directories cpset	cpset(1m)
	binary input/output fread	fread(3S)
bsearch	binary search a sorted table	bsearch(3C)
manage	binary search trees tsearch	tsearch(3C)
Fortran	bitwise boolean functions bool	bool(3F)
the game of	black jack bj	bj(6)
print checksum and	block count of a file sum	sum(1)
update the super	block sync	sync(1)
report number of free disk	blocks df	df(1m)
Fortran bitwise boolean functions	bool	bool(3F)
initialization shell scripts	brc system	brc(1m)

allocation	brk change data segment space	brk(2s)
data segment space allocation	brk change	brk(2)
for modest-sized programs	bs a compiler/interpreter	bs(1)
binary search a sorted table	bsearch	bsearch(3C)
stdio standard	buffered input/output package	stdio(3S)
assign	buffering to a stream setbuf	setbuf(3S)
	build special file mknod	mknod(1m)
swap	bytes swab	swab(3C)
print calendar	cal	cal(1)
desk	calculator dc	dc(1)
print	calendar cal	cal(1)
reminder service	calendar	calendar(1)
	call another UNIX system cu	cu(1)
data returned by stat system	call stat	stat(5)
exercise link and unlink system	calls link	link(1m)
terminal	capability data base terminfo	terminfo(4)
interpret ASA	carriage control characters asa	asa(1)
ft IMSP streaming	cartridge controller	ft(7)
pt IMSP	cartridge controller	pt(7)
text editor (variant of ex for	casual users) edit	edit(1)
files	cat concatenate and print	cat(8)
concatenate and print files	cat	cat(1)
C program beautifier	cb	cb(1)
C compiler	cc	cc(1)
change working directory	cd	cd(1)
delta commentary of an SCCS delta	cdc change the	cdc(1)
value functions floor floor,	ceiling, remainder, absolute	floor(3M)
generate C flow graph	cflow	cflow(1)
allocation brk	change data segment space	brk(2)
allocation brk	change data segment space	brk(2s)
	change login password passwd	passwd(1)
	change mode chmod	chmod(1)
	change mode of file chmod	chmod(2)
chmod	change mode of file	chmod(2s)
environment putenv	change or add value to	putenv(3C)
file chown	change owner and group of a	chown(2)
nice	change priority of a process	nice(2)
nice	change priority of a process	nice(2s)
	change root directory chroot	chroot(2)
command chroot	change root directory for a	chroot(1m)
an SCCS delta cdc	change the delta commentary of	cdc(1)
file newform	change the format of a text	newform(1)
	change working directory cd	cd(1)
	change working directory chdir	chdir(2)
chdir	change working directory	chdir(2s)
create an interprocess	channel pipe	pipe(2)
ungetc push	character back into input stream	ungetc(3S)
neqn eqnchar special	character definitions for eqn and	eqnchar(5)
cuserid get	character login name of the user	cuserid(3S)
getc get	character or word from a stream	getc(3S)
putc put	character or word on a stream	putc(3S)
map of ASCII	character set ascii	ascii(5)
gtty get terminal	characterisitcs	gtty(2s)
analyze surface	characteristics of a document	style(1)
stty set terminal	characteristics	stty(2s)

interpret ASA carriage control	characters asa	asa(1)
translate	characters conv	conv(3C)
classify	characters ctype	ctype(3C)
translate	characters tr	tr(1)
	chdir change working directory	chdir(2s)
change working directory	chdir	chdir(2)
file system consistency	check and interactive repair fsck	fsck(1m)
fsck file system consistency	check and interactive repair	fsck(8)
file system checking procedure	checkall faster	checkall(1m)
a C program	checker lint	lint(1)
nroff/MM document compatibility	checker mmlint sroff/MM	mmlint(1)
password/group file	checkers pwck	pwck(1m)
faster file system	checking procedure checkall	checkall(1m)
copy file systems with label	checking volcopy	volcopy(1m)
of file systems processed by fsck	checklist list	checklist(4)
file sum print	checksum and block count of a	sum(1)
get process and	child process times	times(2)
terminate wait wait for	child process to stop or	wait(2)
	chmod change mode of file	chmod(2s)
change mode	chmod	chmod(1)
change mode of file	chmod	chmod(2)
change owner or group	chown	chown(1)
change root directory	chroot	chroot(2)
root directory for a command	chroot change	chroot(1m)
	classify characters ctype	ctype(3C)
uucp spool directory	clean-up uuclean	uuclean(1m)
	clear i-node clri	clri(1m)
set a process alarm	clock alarm	alarm(2)
cron	clock daemon cron	cron(1m)
report CPU time used	clock	clock(3C)
ldclose	close a common object file	ldclose(3X)
	close a file descriptor close	close(2)
close	close a file descriptor	close(2s)
	close or flush a stream fclose	fclose(3S)
close a file descriptor	close	close(2)
clear i-node	clri	clri(1m)
compare two files	cmp	cmp(1)
filter reverse line-feeds	col	col(1)
	combine SCCS deltas comb	comb(1)
lines common to two sorted files	comm select or reject	comm(1)
run a	command at low priority nice	nice(1)
change root directory for a	command chroot	chroot(1m)
set environment for	command execution env	env(1)
UNIX-to-UNIX system	command execution uux	uux(1)
issue a shell	command from Fortran system	system(3F)
quits nohup run a	command immune to hangups and	nohup(1)
execute a	command on the PCL network net	net(1)
parse	command options getopt	getopt(1)
shell, the standard/restricted	command programming language sh	sh(1)
per-process accounting records	command summary from	acctcms(1m)
issue a shell	command system	system(3S)
condition evaluation	command test	test(1)
time a	command time	time(1)
argument list(s) and execute	command xargs construct	xargs(1)
return Fortran	command-line argument getarc	getarc(3F)

system activity timex time a	command; report process data and	timex(1)
and miscellaneous accounting	commands acct of accounting	acct(1m)
execute	commands at a later time at	at(1)
access graphical and numerical	commands graphics	graphics(1)
install	commands install	install(1m)
network useful with graphical	commands stat statistical	stat(1)
change the delta	commentary of an SCCS delta cdc	cdc(1)
	common archive file format ar	ar(4)
archive files	common archive format arcv	arcv(1)
convert archive files	common archive format	arcv(1)
output a.out	common assembler and link editor	a.out(4)
	common assembler as	as(1)
function log10 Fortran	common logarithm intrinsic	log10(3F)
routines ldfcn	common object file access	ldfcn(4)
ldopen open a	common object file for reading	ldopen(3X)
ldlread line number entries of a	common object file function	ldlread(3X)
close a	common object file ldclose	ldclose(3X)
read the file header of a	common object file ldfhread	ldfhread(3X)
number entries of a section of a	common object file ldlseek line	ldlseek(3X)
to the optional file header of a	common object file ldohseek seek	ldohseek(3X)
entries of a section of a	common object file ldrseek	ldrseek(3X)
section header of a	common object file ldshread	ldshread(3X)
to an indexed/named section of a	common object file ldsseek seek	ldsseek(3X)
of a symbol table entry of a	common object file ldtbindex	ldtbindex(3X)
indexed symbol table entry of a	common object file ldtbread an	ldtbread(3X)
seek to the symbol table of a	common object file ldtbseek	ldtbseek(3X)
line number entries in a	common object file linenum	linenum(4)
print name list of	common object file nm	nm(1)
relocation information for a	common object file reloc	reloc(4)
section header for a	common object file scnhdr	scnhdr(4)
line number information from a	common object file strip and	strip(1)
entry retrieve symbol name for	common object file symbol table	ldgetname(3X)
format syms	common object file symbol table	syms(4)
file header for	common object files filehdr	filehdr(4)
link editor for	common object files ld	ld(1)
print section sizes of	common object files size	size(1)
select or reject lines	common to two sorted files comm	comm(1)
ipcs report inter-process	communication facilities status	ipcs(1)
standard interprocess	communication package stdipc	stdipc(3C)
differential file	comparator diff	diff(1)
	compare two files cmp	cmp(1)
file sccsdiff	compare two versions of an SCCS	sccsdiff(1)
strcmp string	comparision intrinsic functions	strcmp(3F)
3-way differential file	comparison diff3	diff3(1)
directory	comparison dircmp	dircmp(1)
expression regcmp	compile and execute regular	regcmp(3X)
regular expression	compile and match routines regexp	regexp(5)
format of	compiled term file. term	term(4)
C	compiler cc	cc(1)
Fortran 77	compiler f77	f77(1)
scc C	compiler for stand-alone programs	scc(1)
terminfo	compiler tic	tic(1m)
yet another	compiler-compiler yacc	yacc(1)
modest-sized programs bs a	compiler/interpreter for	bs(1)
error function and	complementary error function erf	erf(3M)

await	completion of process wait	wait(1)
Fortran imaginary part of	complex argument aimag	aimag(3F)
function conjg Fortran	complex conjugate intrinsic	conjg(3F)
	compress and expand files pack	pack(1)
table entry of a common object	compute the index of a symbol	ldtbindex(3X)
	concatenate and print files cat	cat(1)
cat	concatenate and print files	cat(8)
test	condition evaluation command	test(1)
ioctl.syscon system console	configuration file	ioctl.syscon(4)
	configure logical disks	dconfig(1m)
dconfig	configure logical disks	dconfig(8)
lpadmin	configure the LP spooling system	lpadmin(1m)
conjugate intrinsic function	conjg Fortran complex	conjg(3F)
conjg Fortran complex	conjugate intrinsic function	conjg(3F)
	connect-time accounting acctcon	acctcon(1m)
an out-going terminal line	connection dial establish	dial(3C)
L-devices link devices,	connection information	L-devices(4)
repair fsck file system	consistency check and interactive	fsck(1m)
repair fsck file system	consistency check and interactive	fsck(8)
ioctl.syscon system	console configuration file	ioctl.syscon(4)
ocw prepare	constant-width text for otroff	ocw(1)
math functions and	constants math	math(5)
	construct a file system mkfs	mkfs(1m)
mkfs	construct a file system	mkfs(8)
execute command xargs	construct argument list(s) and	xargs(1)
remove nroff/troff, tbl, and eqn	constructs deroff	deroff(1)
ls list	contents of directories	ls(8)
list	contents of directory ls	ls(1)
graphical table of	contents routines toc	toc(1)
	context split csplit	csplit(1)
interpret ASA carriage	control characters asa	asa(1)
	control device ioctl	ioctl(2)
file	control fcntl	fcntl(2)
process	control initialization init	init(1m)
message	control operations msgctl	msgctl(2)
semaphore	control operations semctl	semctl(2)
shared memory	control operations shmctl	shmctl(2)
file	control options fcntl	fcntl(5)
uucp status inquiry and job	control uustat	uustat(1)
version	control vc	vc(1)
ft IMSP streaming cartridge	controller	ft(7)
pt IMSP cartridge	controller	pt(7)
tty	controlling terminal interface	tty(7)
translate characters	conv	conv(3C)
terminals term	conventional names for	term(5)
explicit Fortran type	conversion ftype	ftype(3F)
	conversion program units	units(1)
	convert and copy a file dd	dd(1)
dd	convert and copy a file	dd(8)
common archive format	convert archive files from	arcv(1)
and long integers l3tol	convert between 3-byte integers	l3tol(3C)
base-64 ASCII string a64l	convert between long integer and	a64l(3C)
ctime	convert date and time to string	ctime(3C)
string ecvt	convert floating-point number to	ecvt(3C)
	convert formatted input scanf	scanf(3S)

strtol	convert string to integer	strtol(3C)
double-precision number strtod	convert string to	strtod(3C)
convert and	copy a file dd	dd(1)
dd convert and	copy a file	dd(8)
cpio	copy file archives in and out	cpio(1)
access time dcopy	copy file systems for optimal	dcopy(1m)
checking volcopy	copy file systems with label	volcopy(1m)
UNIX system to UNIX system	copy uucp	uucp(1)
public UNIX-to-UNIX system file	copy uuto	uuto(1)
	copy, link or move files cp	cp(1)
format of	core image file core	core(4)
	core memory mem	mem(7)
cosine intrinsic function	cosh Fortran hyperbolic	cosh(3F)
Fortran hyperbolic	cosine intrinsic function cosh	cosh(3F)
Fortran	cosine intrinsic functions cos	cos(3F)
print checksum and block	count of a file sum	sum(1)
word	count wc	wc(1)
copy, link or move files	cp	cp(1)
format of	cpio archive cpio	cpio(4)
copy file archives in and out	cpio	cpio(1)
the C language preprocessor	cpp	cpp(1)
files in binary directories	cpset install object	cpset(1m)
the game of	craps craps	craps(6)
examine system images	crash	crash(1m)
to do when the system crashes	crash what	crash(8)
file or rewrite an existing one	creat create a new	creat(2)
file tmpnam	create a name for a temporary	tmpnam(3S)
existing one creat	create a new file or rewrite an	creat(2)
	create a new process fork	fork(2)
creat	create a new special file	creat(2s)
	create a temporary file tmpfile	tmpfile(3S)
pipe	create an interprocess channel	pipe(2)
admin	create and administer SCCS files	admin(1)
set and get file	creation mask umask	umask(2)
umask set and get file	creation mask	umask(2s)
	cron - clock daemon cron	cron(1m)
user crontab file	crontab	crontab(1)
generate C program	cross-reference cxref	cxref(1)
files macref produce	cross-reference listing of macro	macref(1)
encode/decode	crypt	crypt(1)
generate DES encryption	crypt	crypt(3C)
context split	csplit	csplit(1)
spawn getty to a remote terminal	ct	ct(1)
generate file name for terminal	ctermid	ctermid(3S)
convert date and time to string	ctime	ctime(3C)
C program debugger	ctrace	ctrace(1)
classify characters	ctype	ctype(3C)
call another UNIX system	cu	cu(1)
activity sact print	current SCCS file editing	sact(1)
print name of	current UNIX system uname	uname(1)
get name of	current UNIX system uname	uname(2)
the slot in the utmp file of the	current user ttyslot find	ttyslot(3C)
tell report the	current value of a file pointer	tell(2s)
get path-name of	current working directory getcwd	getcwd(3C)
handling and optimization package	curses CRT screen	curses(3X)

interpolate smooth	curve spline	spline(1)
character login name of the user	cuserid get	cuserid(3S)
line of a file cut	cut out selected fields of each	cut(1)
C program cross-reference	cxref generate	cxref(1)
cron - clock	daemon cron	cron(1m)
error-logging	daemon errdemon	errdemon(1m)
terminate the error-logging	daemon errstop	errstop(1m)
run	daily accounting runacct	runacct(1m)
system backup filesave	daily/weekly UNIX system file	filesave(1m)
time a command; report process	data and system activity timex	timex(1)
port ttytype	data base of terminal types by	ttytype(5)
port ttytype	data base of terminal types by	ttytype(5)
terminal capability	data base terminfo	terminfo(4)
generate disk accounting	data by user ID diskusg diskusg	diskusg(1m)
fashion. access long integer	data in a machine-independent	sputl(3X)
lock process, text, or	data in memory plock	plock(2)
display profile	data prof	prof(1)
stat	data returned by stat system call	stat(5)
change	data segment space allocation brk	brk(2)
brk change	data segment space allocation	brk(2s)
primitive system	data types types	types(5)
relational	database operator join	join(1)
query terminfo	database tput	tput(1)
convert	date and time to string ctime	ctime(3C)
print and set the	date date	date(1)
print and set the date	date	date(1)
desk calculator	dc	dc(1)
disks	dconfig configure logical	dconfig(8)
systems for optimal access time	dcopy copy file	dcopy(1m)
	dd convert and copy a file	dd(8)
convert and copy a file	dd	dd(1)
absolute	debugger adb	adb(1)
C program	debugger ctrace	ctrace(1)
file system	debugger fsdb	fsdb(1m)
fsdb file system	debugger	fsdb(8)
eqnchar special character	definitions for eqn and neqn	eqnchar(5)
basename	deliver portions of path names	basename(1)
tail	deliver the last part of a file	tail(1)
delta make a	delta (change) to an SCCS file	delta(1)
the delta commentary of an SCCS	delta cdc change	cdc(1)
cdc change the	delta commentary of an SCCS delta	cdc(1)
remove a	delta from an SCCS file rmdel	rmdel(1)
a delta (change) to an SCCS file	delta make	delta(1)
combine SCCS	deltas comb	comb(1)
permit or	deny messages mesg	mesg(1)
tbl, and eqn constructs	deroff remove nroff/troff,	deroff(1)
device-independent troff font	description files for	font(5)
troff	description of output language	troff(5)
close a file	descriptor close	close(2)
duplicate an open file	descriptor dup	dup(2)
returns a 1 if specified file	descriptor is a isatty	isatty(2s)
close close a file	descriptor	close(2s)
	desk calculator dc	dc(1)
access	determine accessibility of a file	access(2)
access	determine accessibility of a file	access(2s)

	determine file type file	file(1)
control	device ioctl	ioctl(2)
	device name devnm	devnm(1m)
graphical	device routines and filters gdev	gdev(1)
font description files for	device-independent troff	font(5)
L-devices link	devices, connection information	L-devices(4)
device name	devnm	devnm(1m)
report number of free disk blocks	df	df(1m)
	dformat disk formatter	dformat(8)
dial	dial a Racal-Vadic 3451 modem	dial(1)
mdial	dial the P/75 onboard modem	dial(1)
terminal line connection	dial establish an out-going	dial(3C)
rational Fortran	dialect ratfor	ratfor(1)
L-dialcodes alphabetic	dialing abbreviations file	L-dialcodes(4)
big	diff bdiff	bdiff(1)
differential file comparator	diff	diff(1)
differential file comparison	diff3 3-way	diff3(1)
dim positive	difference intrinsic functions	dim(3F)
side-by-side	difference program sdiff	sdiff(1)
mark	differences between files diffmk	diffmk(1)
	differential file comparator diff	diff(1)
diff3 3-way	differential file comparison	diff3(1)
mark differences between files	diffmk	diffmk(1)
difference intrinsic functions	dim positive	dim(3F)
format of directories	dir	dir(4)
directory comparison	dircmp	dircmp(1)
install object files in binary	directories cpset	cpset(1m)
format of	directories dir	dir(4)
remove files or	directories rm	rm(1)
ls list contents of	directories	ls(8)
change working	directory cd	cd(1)
change working	directory chdir	chdir(2)
change root	directory chroot	chroot(2)
uucp spool	directory clean-up uuclean	uuclean(1m)
	directory comparison dircmp	dircmp(1)
remove	directory entry unlink	unlink(2)
change root	directory for a command chroot	chroot(1m)
get path-name of current working	directory getcwd	getcwd(3C)
list contents of	directory ls	ls(1)
make a	directory mkdir	mkdir(1)
move a	directory mvdir	mvdir(1m)
working	directory name pwd	pwd(1)
ordinary file mknod make a	directory or a special or	mknod(2)
chdir change working	directory	chdir(2s)
enable or	disable process accounting acct	acct(2)
type, modes, speed, and line	discipline getty set terminal	getty
	disk access profiler sadp	sadp(1m)
diskusg diskusg - generate	disk accounting data by user ID	diskusg(1m)
report number of free	disk blocks df	df(1m)
dformat	disk formatter	dformat(8)
summarize	disk usage du	du(1)
du summarize	disk usage	du(8)
ramdisk memory as	disk	ramdisk(1m)
allows memory to be used as a	disk rram	rram(7)
configure logical	disks	dconfig(1m)

accounting data by user ID	diskusg - generate disk	diskusg(1m)
mount and	dismount file system mount	mount(1m)
screen-oriented (visual)	display editor based on ex vi	vi(1)
	display profile data prof	prof(1)
Euclidean	distance function hypot	hypot(3M)
drand48 generate uniformly	distributed pseudo-random numbers	drand48(3C)
mmlint sroff/MM nroff/MM	document compatibility checker	mmlint(1)
surface characteristics of a	document analyze	style(1)
macros mm print/check	documents formatted with the MM	mm(1)
MM macro package for formatting	documents mm the	mm(5)
macro package for formatting	documents the OSDD adapter	mosd(5)
mmt typeset	documents, viewgraphs, and slides	mmt(1)
who is	doing what whodo	whodo(1m)
intrinsic function dprod	double precision product	dprod(3F)
float and	double routines	float(2s)
convert string to	double-precision number strtod	strtod(3C)
product intrinsic function	dprod double precision	dprod(3F)
distributed pseudo-random numbers	drand48 generate uniformly	drand48(3C)
	draw a graph graph	graph(1)
pic troff preprocessor for	drawing simple pictures	pic(1)
provide	drill in number facts arithmetic	arithmetic(6)
rm Cipher Microstreamer tape	drive	rm(7)
	du summarize disk usage	du(8)
summarize disk usage	du	du(1)
Communication acpdmp	dump contents of Advanced	acpdmp(1m)
Intelligent Communication icpdmp	dump contents of an	icpdmp(1m)
extract error records from	dump errdead	errdead(1m)
octal	dump od	od(1)
object file dump	dump selected parts of an	dump(1)
dump incremental	dump tape format	dump(4)
od octal	dump	od(8)
selected parts of an object file	dump dump	dump(1)
duplicate an open file descriptor	dup	dup(2)
for the Xerox 9700 printer	dx9700 prepare troff documents	dx9700(1)
	echo arguments echo	echo(1)
floating-point number to string	ecvt convert	ecvt(3C)
text editor	ed	ed(1)
(variant of ex for casual users)	edit text editor	edit(1)
print current SCCS file	editing activity sact	sact(1)
users edit text	editor (variant of ex for casual	edit(1)
screen-oriented (visual) display	editor based on ex vi	vi(1)
text	editor ex	ex(1)
link	editor for common object files ld	ld(1)
graphic	editor ged	ged(1)
common assembler and link	editor output a.out	a.out(4)
stream	editor sed	sed(1)
split f77, ratfor, or	efl files fsplit	fsplit(1)
Extended Fortran Language	efl	efl(1)
accounting acct	enable or disable process	acct(2)
enable/disable LP printers	enable	enable(1)
enable	enable/disable LP printers	enable(1)
for/ uuencode,uudecode	encode/decode a binary file	uuencode(1c)
	encode/decode crypt	crypt(1)
generate DES	encryption crypt	crypt(3C)
generate	encryption key makekey	makekey(1)

lst locations in program	end	end(3C)
get	entries from name list nlist	nlist(3C)
linenum line number	entries in a common object file	linenum(4)
print	entries in this manual man	man(1)
manipulate line number	entries of a common object file	ldlread(3X)
object file seek to relocation	entries of a section of a common	ldrseek(3X)
utmp and wtmp	entry formats utmp	utmp(4)
get group file	entry getgrent	getgrent(3C)
get password file	entry getpwent	getpwent(3C)
access utmp file	entry getut	getut(3C)
common object file symbol table	entry ldgetname symbol name for	ldgetname(3X)
the index of a symbol table	entry of a common object file	ldtbindex(3X)
read an indexed symbol table	entry of a common object file	ldtbread(3X)
write password file	entry putpwent	putpwent(3C)
remove directory	entry unlink	unlink(2)
environment for command execution	env set	env(1)
user environment	environ	environ(5)
setting up an	environment at login time profile	profile(4)
profile setting up an	environment at login time	profile(5)
user	environment environ	environ(5)
env set	environment for command execution	env(1)
return value for	environment name getenv	getenv(3C)
change or add value to	environment putenv	putenv(3C)
return Fortran	environment variable getenv	getenv(3F)
special character definitions for	eqn and neqn eqnchar	eqnchar(5)
remove nroff/troff, tbl, and	eqn constructs deroff	deroff(1)
for nroff or troff	eqn format mathematical text	eqn(1)
and complementary error function	erf error function	erf(3M)
error-logging interface	err	err(7)
extract error records from dump	errdead	errdead(1m)
error-logging daemon	errdemon	errdemon(1m)
error-log file format	errfile	errfile(4)
complementary error function erf	error function and	erf(3M)
system	error messages perror	perror(3C)
extract	error records from dump errdead	errdead(1m)
	error-handling function matherr	matherr(3M)
	error-log file format errfile	errfile(4)
	error-logging daemon errdemon	errdemon(1m)
terminate the	error-logging daemon errstop	errstop(1m)
	error-logging interface err	err(7)
process a report of logged	errors errpt	errpt(1m)
find spelling	errors spell	spell(1)
process a report of logged errors	errpt	errpt(1m)
the error-logging daemon	errstop terminate	errstop(1m)
line connection dial	establish an out-going terminal	dial(3C)
	establish mount table setmnt	setmnt(1m)
expression expr	evaluate arguments as an	expr(1)
test condition	evaluation command	test(1)
text editor (variant of	ex for casual users) edit	edit(1)
(visual) display editor based on	ex vi screen-oriented	vi(1)
	examine system images crash	crash(1m)
execute a file	exec	exec(2)
network net	execute a command on the PCL	net(1)
	execute a file exec	exec(2)
construct argument list(s) and	execute command xargs	xargs(1)

at	execute commands at a later time	at(1)
compile and	execute regular expression regcmp	regcmp(3X)
uuxqt	execute remote command requests	uuxqt(1m)
L.cmds remote	execution commands	L.cmds(4)
set environment for command	execution env	env(1)
suspend	execution for interval sleep	sleep(3C)
prepare	execution profile monitor	monitor(3C)
	execution time profile profil	profil(2)
UNIX-to-UNIX system command	execution uux	uux(1)
calls link	exercise link and unlink system	link(1m)
create a new file or rewrite an	existing one creat	creat(2)
	exit terminate process	exit(2s)
terminate process	exit	exit(2)
EOT on the other terminal and	exits. write	write(1)
exponential intrinsic function	exp Fortran	exp(3F)
power, square root function	exp exponential, logarithm,	exp(3M)
compress and	expand files pack	pack(1)
conversion ftype	explicit Fortran type	ftype(3F)
exp Fortran	exponential intrinsic function	exp(3F)
square root function exp	exponential, logarithm, power,	exp(3M)
arguments as an expression	expr evaluate	expr(1)
routines regexp regular	expression compile and match	regexp(5)
regular	expression compile regcmp	regcmp(1)
evaluate arguments as an	expression expr	expr(1)
compile and execute regular	expression regcmp	regcmp(3X)
errdead	extract error records from dump	errdead(1m)
Fortran 77 compiler	f77	f77(1)
split	f77, ratfor, or efl files fsplit	fsplit(1)
inter-process communication	facilities status ipcs report	ipcs(1)
	factor a number factor	factor(1)
provide drill in number	facts arithmetic	arithmetic(6)
data in a machine-independent	fashion. sputl long integer	sputl(3X)
	fast incremental backup finc	finc(1m)
	fast main memory allocator malloc	malloc(3X)
fbackup	fast tape backup of a file system	fbackup
fbackup make a	fast tape backup of a file system	fbackup(8)
procedure checkall	faster file system checking	checkall(1m)
generate an IOT	fault abort	abort(3C)
backup of a file system	fbackup make a fast tape	fbackup(1m)
backup of a file system	fbackup make a fast tape	fbackup(8)
close or flush a stream	fclose	fclose(3S)
file control options	fcntl	fcntl(5)
stream status inquiries	ferror	ferror(3S)
and statistics for a file system	ff list file names	ff(1m)
cut out selected	fields of each line of a file cut	cut(1)
times utime set	file access and modification	utime(2)
common object	file access routines ldfcn	ldfcn(4)
determine accessibility of a	file access	access(2)
tape	file archiver tar	tar(1)
copy	file archives in and out cpio	cpio(1)
password/group	file checkers pwck	pwck(1m)
change of of	file chmod	chmod(2)
change owner and group of a	file chown	chown(2)
differential	file comparator diff	diff(1)
3-way differential	file comparison diff3	diff3(1)

	file control fcntl	fcntl(2)
	file control options fcntl	fcntl(5)
public UNIX-to-UNIX system	file copy uuto	uuto(1)
format of core image	file core	core(4)
set and get	file creation mask umask	umask(2)
umask set and get	file creation mask	umask(2s)
user crontab	file crontab	crontab(1)
selected fields of each line of a	file cut cut out	cut(1)
convert and copy a	file dd	dd(1)
make a delta (change) to an SCCS	file delta	delta(1)
close a	file descriptor close	close(2)
duplicate an open	file descriptor dup	dup(2)
returns a 1 if specified	file descriptor is a isatty	isatty(2s)
close close a	file descriptor	close(2s)
dump selected parts of an object	file dump	dump(1)
print current SCCS	file editing activity sact	sact(1)
get group	file entry getgrent	getgrent(3C)
get password	file entry getpwent	getpwent(3C)
access utmp	file entry getut	getut(3C)
write password	file entry putpwent	putpwent(3C)
execute a	file exec	exec(2)
search a	file for a pattern grep	grep(1)
open a common object	file for reading ldopen	ldopen(3X)
per-process accounting	file format acct	acct(4)
common archive	file format ar	ar(4)
error-log	file format errfile	errfile(4)
number entries of a common object	file function ldlread line	ldlread(3X)
get a version of a SCCS	file get	get(1)
group	file group	group(4)
files filehdr	file header for common object	filehdr(4)
file ldfhread read the	file header of a common object	ldfhread(3X)
file seek to the optional	file header of a common object	ldohseek(3X)
split a	file into pieces split	split(1)
issue identification	file issue	issue(4)
header of a member of an archive	file ldahread read the archive	ldahread(3X)
close a common object	file ldclose	ldclose(3X)
file header of a common object	file ldfhread read the	ldfhread(3X)
of a section of a common object	file ldlseek line number entries	ldlseek(3X)
file header of a common object	file ldohseek to the optional	ldohseek(3X)
of a section of a common object	file ldrseek relocation entries	ldrseek(3X)
section header of a common object	file ldshread an indexed/named	ldshread(3X)
section of a common object	file ldsseek an indexed/named	ldsseek(3X)
table entry of a common object	file ldtbindex index of a symbol	ldtbindex(3X)
table entry of a common object	file ldtbread an indexed symbol	ldtbread(3X)
symbol table of a common object	file ldtbseek seek to the	ldtbseek(3X)
number entries in a common object	file linenum line	linenum(4)
link to a	file link	link(2)
build special	file mknod	mknod(1m)
or a special or ordinary	file mknod make a directory	mknod(2)
generate	file name for terminal ctermid	ctermid(3S)
make a unique	file name mktemp	mktemp(3C)
file system ff list	file names and statistics for a	ff(1m)
change the format of a text	file newform	newform(1)
print name list of common object	file nm	nm(1)
the null	file null	null(7)

find the slot in the utmp	file of the current user ttyslot	ttyslot(3C)
identify processes using a	file or file structure fuser	fuser(1m)
creat create a new	file or rewrite an existing one	creat(2)
password	file passwd	passwd(4)
files or subsequent lines of one	file paste same lines of several	paste(1)
terminals pg	file perusal filter for soft-copy	pg(1)
reposition a	file pointer in a stream fseek	fseek(3S)
move read/write	file pointer lseek	lseek(2)
lseek move read/write	file pointer	lseek(2s)
report the current value of a	file pointer tell	tell(2s)
print an SCCS	file prs	prs(1)
read from	file read	read(2)
information for a common object	file reloc relocation	reloc(4)
remove a delta from an SCCS	file rmdel	rmdel(1)
big	file scanner bfs	bfs(1)
compare two versions of an SCCS	file sccsdiff	sccsdiff(1)
format of SCCS	file sccsfile	sccsfile(4)
header for a common object	file scnhdr section	scnhdr(4)
get	file status stat	stat(2)
stat get	file status	stat(2s)
information from a common object	file strip and line number	strip(1)
processes using a file or	file structure fuser identify	fuser(1m)
checksum and block count of a	file sum print	sum(1)
symbol name for common object	file symbol table entry ldgetname	ldgetname(3X)
common object	file symbol table format syms	syms(4)
daily/weekly UNIX system	file system backup filesave	filesave(1m)
checkall faster	file system checking procedure	checkall(1m)
interactive repair fsck	file system consistency check and	fsck(1m)
and interactive repair fsck	file system consistency check	fsck(8)
	file system debugger fsdb	fsdb(1m)
fsdb	file system debugger	fsdb(8)
file names and statistics for a	file system ff list	ff(1m)
construct a	file system mkfs	mkfs(1m)
mount and dismount	file system mount	mount(1m)
mount a	file system mount	mount(2)
incremental	file system restore	restor(1m)
restor incremental	file system restore	restor(8)
get	file system statistics ustat	ustat(2)
ustat get	file system statistics	ustat(2s)
mounted	file system table mnttab	mnttab(4)
unmount a	file system umount	umount(2)
mount a	file system	mount(2s)
unmount a	file system	umount(2s)
make a fast tape backup of a	file system fbackup	fbackup(1m)
time dcopy copy	file systems for optimal access	dcopy(1m)
checklist list of	file systems processed by fsck	checklist(4)
volcopy copy	file systems with label checking	volcopy(1m)
deliver the last part of a	file tail	tail(1)
create a temporary	file tmpfile	tmpfile(3S)
create a name for a temporary	file tmpnam	tmpnam(3S)
and modification times of a	file touch update access	touch(1)
uucp system uucico	file transport program for	uucico
walk a	file tree ftw	ftw(3C)
determine	file type file	file(1)
undo a previous get of an SCCS	file unget	unget(1)

report repeated lines in a	file uniq	uniq(1)
validate SCCS	file val	val(1)
write on a	file write	write(2)
creat create a new special	file	creat(2s)
determine file type	file	file(1)
mknod make a special	file	mknod(2s)
read read from	file	read(2s)
write on a	file	write(2s)
and print process accounting	file(s) acctcom search	acctcom(1)
set	file-creation mode mask umask	umask(1)
format of compiled term	file. term	term(4)
header for common object files	filehdr file	filehdr(4)
merge or add total accounting	files acctmerg	acctmerg(1m)
create and administer SCCS	files admin	admin(1)
concatenate and print	files cat	cat(1)
compare two	files cmp	cmp(1)
reject lines common to two sorted	files comm select or	comm(1)
copy, link or move	files cp	cp(1)
mark differences between	files diffmk	diffmk(1)
file header for common object	files filehdr	filehdr(4)
recover	files from a backup tape frec	frec(1m)
format specification in text	files fspec	fspec(4)
split f77, ratfor, or efl	files fsplit	fsplit(1)
string, format of graphical	files gps graphical primitive	gps(4)
install object	files in binary directories cpset	cpset(1m)
link editor for common object	files ld	ld(1)
remove	files or directories rm	rm(1)
merge same lines of several	files or subsequent lines of one	paste(1)
compress and expand	files pack	pack(1)
print	files pr	pr(1)
section sizes of common object	files size print	size(1)
sort and/or merge	files sort	sort(1)
identify SCCS	files what	what(1)
archive format convert archive	files	arcv(1)
UNIX system file system backup	filesave daily/weekly	filesave(1m)
file perusal	filter for soft-copy terminals pg	pg(1)
select terminal	filter greek	greek(1)
line numbering	filter nl	nl(1)
	filter reverse line-feeds col	col(1)
graphical device routines and	filters gdev	gdev(1)
graphics	filters tplot	tplot(1)
fast incremental backup	finc	finc(1m)
	find hyphenated words hyphen	hyphen(1)
	find name of a terminal ttyname	ttyname(3C)
object library lorder	find ordering relation for an	lorder(1)
	find spelling errors spell	spell(1)
the current user ttyslot	find the slot in the utmp file of	ttyslot(3C)
find files	find	find(1)
	float and double routines	float(2s)
ecvt convert	floating-point number to string	ecvt(3C)
manipulate parts of	floating-point numbers frexp	frexp(3S)
absolute value functions floor	floor, ceiling, remainder,	floor(3M)
generate C	flow graph cflow	cflow(1)
close or	flush a stream fclose	fclose(3S)
open a stream	fopen	fopen(3S)

create a new process	fork	fork(2)
per-process accounting file	format acct	acct(4)
common archive file	format ar	ar(4)
common archive	format arcv archive files	arcv(1)
error-log file	format errfile	errfile(4)
nroff or troff eqn	format mathematical text for	eqn(1)
	format of SCCS file sccsfile	sccsfile(4)
change the	format of a text file newform	newform(1)
	format of an i-node inode	inode(4)
term	format of compiled term file.	term(4)
	format of core image file core	core(4)
	format of cpio archive cpio	cpio(4)
	format of directories dir	dir(4)
graphical primitive string,	format of graphical files gps	gps(4)
	format of system volume fs	fs(4)
nroff	format or typeset text	nroff(1)
files fspec	format specification in text	fspec(4)
common object file symbol table	format syms	syms(4)
tbl	format tables for nroff or troff	tbl(1)
sroff	format text	sroff(1)
utmp and wtmp entry	formats utmp	utmp(4)
convert	formatted input scanf	scanf(3S)
argument list vprintf print	formatted output of a varargs	vprintf(3S)
print	formatted output printf	printf(3S)
dformat disk	formatter	dformat(8)
mptx the macro package for	formatting a permuted index	mptx(5)
troff text	formatting and typesetting	troff(1)
mm the MM macro package for	formatting documents	mm(5)
OSDD adapter macro package for	formatting documents mosd the	mosd(5)
man macros for	formatting entries in this manual	man(5)
binary input/output	fread	fread(3S)
recover files from a backup tape	frec	frec(1m)
report number of	free disk blocks df	df(1m)
parts of floating-point numbers	frexp manipulate	frexp(3S)
format of system volume	fs	fs(4)
list of file systems processed by	fsck checklist	checklist(4)
check and interactive repair	fsck file system consistency	fsck(8)
check and interactive repair	fsck file system consistency	fsck(1m)
	fsdb file system debugger	fsdb(8)
file system debugger	fsdb	fsdb(1m)
a file pointer in a stream	fseek reposition	fseek(3S)
specification in text files	fspec format	fspec(4)
split f77, ratfor, or efl files	fsplit	fsplit(1)
controller	ft IMSP streaming cartridge	ft(7)
walk a file tree	ftw	ftw(3C)
explicit Fortran type conversion	ftype	ftype(3F)
Fortran arcsine intrinsic	functio asin	asin(3F)
Fortran arccosine intrinsic	function acos	acos(3F)
Fortran integer part intrinsic	function aint	aint(3F)
function erf error	function and complementary error	erf(3M)
Fortran arctangert intrinsic	function atan	atan(3F)
Fortran arctangent intrinsic	function atan2	atan2(3F)
complex conjugate intrinsic	function conjg Fortran	conjg(3F)
hyperbolic cosine intrinsic	function cosh Fortran	cosh(3F)
precision product intrinsic	function dprod double	dprod(3F)

function and complementary error	function erf error	erf(3M)
Fortran exponential intrinsic	function exp	exp(3F)
logarithm, power, square root	function exp exponential,	exp(3M)
log gamma	function gamma	gamma(3M)
Euclidean distance	function hypot	hypot(3M)
entries of a common object file	function ldlread line number	ldlread(3X)
natural logarithm intrinsic	function log Fortran	log(3F)
common logarithm intrinsic	function log10 Fortran	log10(3F)
error-handling	function matherr	matherr(3M)
profile within a	function prof	prof(5)
transfer-of-sign intrinsic	function sign Fortran	sign(3F)
Fortran sine intrinsic	function sin	sin(3F)
Fortran hyperbolic sine intrinsic	function sinh	sinh(3F)
Fortran square root intrinsic	function sqrt	sqrt(3F)
Fortran tangent intrinsic	function tan	tan(3F)
hyperbolic tangent intrinsic	function tanh Fortran	tanh(3F)
math	functions and constants math	math(5)
Fortran bitwise boolean	functions bool	bool(3F)
Fortran cosine intrinsic	functions cos	cos(3F)
positive difference intrinsic	functions dim	dim(3F)
remainder, absolute value	functions floor floor, ceiling,	floor(3M)
Fortran maximum-value	functions max	max(3F)
Fortran minimum-value	functions min	min(3F)
Fortran remaindering intrinsic	functions mod	mod(3F)
terminals 300 handle special	functions of DASI 300 and 300s	300(1)
2621-series handle special	functions of HP 2640 and	hp(1)
terminal 450 handle special	functions of the DASI 450	450(1)
Fortran nearest integer	functions round	round(3F)
hyperbolic	functions sinh	sinh(3M)
string comparision intrinsic	functions strcmp	strcmp(3F)
trigonometric	functions trig	trig(3M)
using a file or file structure	fuser identify processes	fuser(1m)
guessing	game moo	moo(6)
the	game of backgammon back	back(6)
the	game of black jack bj	bj(6)
the	game of craps craps	craps(6)
the	game of hunt-the-wumpus wump	wump(6)
log gamma function	gamma	gamma(3M)
device routines and filters	gdev graphical	gdev(1)
graphic editor	ged	ged(1)
tty	general terminal interface	tty(7)
	generate C flow graph cflow	cflow(1)
cross-reference cxref	generate C program	cxref(1)
	generate DES encryption crypt	crypt(3C)
	generate a maze maze	maze(6)
	generate an IOT fault abort	abort(3C)
user ID diskusg diskusg	generate disk accounting data by	diskusg(1m)
	generate encryption key makekey	makekey(1)
ctermid	generate file name for terminal	ctermid(3S)
ncheck	generate names from i-numbers	ncheck(1m)
lexical tasks lex	generate programs for simple	lex(1)
pseudo-random numbers drand48	generate uniformly distributed	drand48(3C)
simple random-number	generator rand	rand(3C)
random number	generator rand	rand(3F)
	get a string from a stream gets	gets(3S)

get	get a version of a SCCS file	get(1)
	get and set user limits ulimit	ulimit(2)
user cuserid	get character login name of the	cuserid(3S)
stream getc	get character or word from a	getc(3S)
nlist	get entries from name list	nlist(3C)
	get file status stat	stat(2)
stat	get file status	stat(2s)
ustat	get file system statistics	ustat(2)
ustat	get file system statistics	ustat(2s)
	get group file entry getgrent	getgrent(3C)
	get login name getlogin	getlogin(3C)
	get login name logname	logname(1)
	get message queue msgget	msgget(2)
	get name from UID getpw	getpw(3C)
uname	get name of current UNIX system	uname(2)
vector getopt	get option letter from argument	getopt(3C)
getpwent	get password file entry	getpwent(3C)
working directory getcwd	get path-name of current	getcwd(3C)
getpid	get process ID	getpid(2s)
times	get process and child process	times(2)
parent process IDs getpid	get process, process group, and	getpid(2)
and real and effective getuid	get real and effective user,	getuid(2s)
real group, and effective group	get real user, effective user,	getuid(2)
	get set of semaphores semget	semget(2)
shmget	get shared memory segment	shmget(2)
gtty	get terminal characterisitcs	gtty(2s)
tty	get the name of the terminal	tty(1)
	get time time	time(2)
time	get time	time(2s)
Fortran command-line argument	getarc return	getarc(3F)
and get arguments for	getargv display a program name	getargv(2s)
character or word from a stream	getc get	getc(3S)
of current working directory	getcwd get path-name	getcwd(3C)
return value for environment name	getenv	getenv(3C)
Fortran environment variable	getenv return	getenv(3F)
get group file entry	getgrent	getgrent(3C)
get login name	getlogin	getlogin(3C)
parse command options	getopt	getopt(1)
letter from argument vector	getopt get option	getopt(3C)
read a password	getpass	getpass(3C)
	getpid get process ID	getpid(2s)
group, and parent process IDs	getpid get process, process	getpid(2)
get name from UID	getpw	getpw(3C)
get password file entry	getpwent	getpwent(3C)
get a string from a stream	gets	gets(3S)
and terminal settings used by	getty gettydefs speed	gettydefs(4)
spawn	getty to a remote terminal ct	ct(1)
terminal settings used by getty	gettydefs speed and	gettydefs(4)
user, and real and effective	getuid get real and effective	getuid(2s)
group, and effective group ID's	getuid effective user, real	getuid(2)
access utmp file entry	getut	getut(3C)
string, format of graphical files	gps graphical primitive	gps(4)
generate C flow	graph cflow	cflow(1)
draw a	graph graph	graph(1)
system activity	graph sag	sag(1)

draw a graph	graph	graph(1)
	graphic editor ged	ged(1)
graphics access	graphical and numerical commands	graphics(1)
statistical network useful with	graphical commands stat	stat(1)
filters gdev	graphical device routines and	gdev(1)
primitive string, format of	graphical files gps graphical	gps(4)
routines toc	graphical table of contents	toc(1)
	graphical utilities gutil	gutil(1)
	graphics filters tplot	tplot(1)
	graphics interface plot	plot(4)
plot	graphics interface subroutines	plot(3X)
graphical and numerical commands	graphics access	graphics(1)
a macro package for making view	graphs mv	mv(7)
select terminal filter	greek	greek(1)
search a file for a pattern	grep	grep(1)
set process	group ID setpgrp	setpgrp(2)
user, and real and effective	group ID get real and effective	getuid(2s)
user, real group, and effective	group ID's getuid effective	getuid(2)
print user and	group IDs and names id	id(1)
set user and	group IDs setuid	setuid(2)
change owner or	group chown	chown(1)
get	group file entry getgrent	getgrent(3C)
	group file group	group(4)
log in to a new	group newgrp	newgrp(1)
change owner and	group of a file chown	chown(2)
send a signal to a process or a	group of processes kill	kill(2)
group file	group	group(4)
real user, effective user, real	group, and effective group ID's	getuid(2)
getpid get process, process	group, and parent process IDs	getpid(2)
maintain, update, and regenerate	groups of programs make	make(1)
characterisitcs	gtty get terminal	gtty(2s)
	guess the word hangman	hangman(6)
	guessing game moo	moo(6)
graphical utilities	gutil	gutil(1)
300 and 300s terminals 300	handle special functions of DASI	300(1)
2640 and 2621-series terminals	handle special functions of HP	hp(1)
DASI 450 terminal 450	handle special functions of the	450(1)
varargs	handle variable argument list	varargs(5)
curses CRT screen	handling and optimization package	curses(3X)
guess the word	hangman	hangman(6)
run a command immune to	hangups and quits	nohup(1)
manage	hash search tables hsearch	hsearch(3C)
scnhdr section	header for a common object file	scnhdr(4)
filehdr file	header for common object files	filehdr(4)
ldfhread read the file	header of a common object file	ldfhread(3X)
seek to the optional file	header of a common object file	ldohseek(3X)
read an indexed/named section	header of a common object file	ldshread(3X)
file ldahread read the archive	header of a member of an archive	ldahread(3X)
	help ask for help	help(8)
ask for help	help	help(1)
HP 2640 and 2621-series terminals	hp handle special functions of	hp(1)
manage hash search tables	hsearch	hsearch(3C)
the game of	hunt-the-wumpus wump	wump(6)
function cosh Fortran	hyperbolic cosine intrinsic	cosh(3F)
	hyperbolic functions sinh	sinh(3M)

function sinh Fortran	hyperbolic sine intrinsic	sinh(3F)
function tanh Fortran	hyperbolic tangent intrinsic	tanh(3F)
find hyphenated words	hyphen	hyphen(1)
Euclidean distance function	hypot	hypot(3M)
clear	i-node clri	clri(1m)
format of an	i-node inode	inode(4)
generate names from	i-numbers ncheck	ncheck(1m)
	iargc iargc	iargc(3F)
Processor	icp Intelligent Communications	icp(7)
Intelligent Communication	icpdmp dump contents of an	icpdmp(1m)
semaphore set or shared memory	id ipcrm remove a message queue,	ipcrm(1)
user and group IDs and names	id print	id(1)
issue	identification file issue	issue(4)
	identify SCCS files what	what(1)
or file structure fuser	identify processes using a file	fuser(1m)
format of core	image file core	core(4)
examine system	images crash	crash(1m)
argument aimag Fortran	imaginary part of complex	aimag(3F)
run a command	immune to hangups and quits nohup	nohup(1)
fast	incremental backup finc	finc(1m)
dump	incremental dump tape format	dump(4)
	incremental file system restore	restor(1m)
restor	incremental file system restore	restor(8)
termlib terminal	independent operation routines	termlib(3c)
a common object compute the	index of a symbol table entry of	ldtbindex(3X)
permuted	index ptx	ptx(1)
package for formatting a permuted	index mptx the macro	mptx(5)
location of Fortran substring	index return	index(3F)
common object file read an	indexed symbol table entry of a	ldtbread(3X)
a common object file read an	indexed/named section header of	ldshread(3X)
common object file seek to an	indexed/named section of a	ldsseek(3X)
file reloc relocation	information for a common object	reloc(4)
strip symbol and line number	information from a common object	strip(1)
print LP status	information lpstat	lpstat(1)
script for the	init process inittab	inittab(4)
process control initialization	init	init(1m)
process control	initialization init	init(1m)
system	initialization shell scripts brc	brc(1m)
popen	initiate pipe to/from a process	popen(3S)
script for the init process	inittab	inittab(4)
format of an i-node	inode	inode(4)
convert formatted	input scanf	scanf(3S)
push character back into	input stream ungetc	ungetc(3S)
binary	input/output fread	fread(3S)
standard buffered	input/output package stdio	stdio(3S)
stream status	inquiries ferror	ferror(3S)
uucp status	inquiry and job control uustat	uustat(1)
directories cpset	install object files in binary	cpset(1m)
install commands	install	install(1m)
return	integer absolute value abs	abs(3C)
a64l convert between long	integer and base-64 ASCII string	a64l(3C)
access long	integer data in a	sputl(3X)
Fortran nearest	integer functions round	round(3F)
aint Fortran	integer part intrinsic function	aint(3F)
convert string to	integer strtol	strtol(3C)

convert between 3-byte	integers and long integers l3tol	l3tol(3C)
facilities status ipcs report	inter-process communication	ipcs(1)
system mailx	interactive message processing	mailx(1)
file system consistency check and	interactive repair fsck	fsck(1m)
file system consistency check and	interactive repair fsck	fsck(8)
error-logging	interface err	err(7)
graphics	interface plot	plot(4)
graphics	interface subroutines plot	plot(3X)
controlling terminal	interface tty	tty(7)
pp parallel port	interface	pp(7)
tty general terminal	interface	tty(7)
	interpolate smooth curve spline	spline(1)
characters asa	interpret ASA carriage control	asa(1)
SNOBOL	interpreter sno	sno(1)
tc troff output	interpreter	tc(1)
create an	interprocess channel pipe	pipe(2)
package stdipc standard	interprocess communication	stdipc(3C)
suspend execution for an	interval sleep	sleep(1)
suspend execution for	interval sleep	sleep(3C)
Fortran arcsine	intrinsic functio asin	asin(3F)
Fortran arccosine	intrinsic function acos	acos(3F)
Fortran integer part	intrinsic function aint	aint(3F)
Fortran arctangert	intrinsic function atan	atan(3F)
Fortran arctangent	intrinsic function atan2	atan2(3F)
Fortran complex conjugate	intrinsic function conjg	conjg(3F)
Fortran hyperbolic cosine	intrinsic function cosh	cosh(3F)
double precision product	intrinsic function dprod	dprod(3F)
Fortran exponential	intrinsic function exp	exp(3F)
Fortran natural logarithm	intrinsic function log	log(3F)
Fortran common logarithm	intrinsic function log10	log10(3F)
Fortran transfer-of-sign	intrinsic function sign	sign(3F)
Fortran sine	intrinsic function sin	sin(3F)
Fortran hyperbolic sine	intrinsic function sinh	sinh(3F)
Fortran square root	intrinsic function sqrt	sqrt(3F)
Fortran tangent	intrinsic function tan	tan(3F)
Fortran hyperbolic tangent	intrinsic function tanh	tanh(3F)
Fortran cosine	intrinsic functions cos	cos(3F)
positive difference	intrinsic functions dim	dim(3F)
Fortran remaindering	intrinsic functions mod	mod(3F)
string comparision	intrinsic functions strcmp	strcmp(3F)
control device	ioctl	ioctl(2)
configuration file	ioctl.syscon system console	ioctl.syscon(4)
semaphore set or shared memory id	ipcrm remove a message queue,	ipcrm(1)
communication facilities status	ipcs report inter-process	ipcs(1)
specified file descriptor is a	isatty returns a 1 if	isatty(2s)
Fortran system	issue a shell command from	system(3F)
	issue a shell command system	system(3S)
	issue identification file issue	issue(4)
issue identification file	issue	issue(4)
print news	items news	news(1)
uucp status inquiry and	job control uustat	uustat(1)
relational database operator	join	join(1)
generate encryption	key makekey	makekey(1)
	kill all active processes killall	killall(1m)
process or a group of processes	kill send a signal to a	kill(2s)

terminate a process	kill	kill(1)
a process or a group of processes	kill send a signal to	kill(2)
kill all active processes	killall	killall(1m)
test your	knowledge quiz	quiz(6)
3-byte integers and long integers	l3tol convert between	l3tol(3C)
copy file systems with	label checking volcopy	volcopy(1m)
pattern scanning and processing	language awk	awk(1)
arbitrary-precision arithmetic	language bc	bc(1)
the C	language preprocessor cpp	cpp(1)
command programming	language sh standard/restricted	sh(1)
troff description of output	language	troff(5)
execute commands at a	later time at	at(1)
editor for common object files	ld link	ld(1)
of a member of an archive file	ldahread read the archive header	ldahread(3X)
close a common object file	ldclose	ldclose(3X)
object file access routines	ldfcn common	ldfcn(4)
header of a common object file	ldfhread read the file	ldfhread(3X)
object file symbol table entry	ldgetname symbol name for common	ldgetname(3X)
of a common object file function	ldlread line number entries	ldlread(3X)
a section of a common object file	ldlseek line number entries of	ldlseek(3X)
header of a common object file	ldohseek to the optional file	ldohseek(3X)
a common object file for reading	ldopen open	ldopen(3X)
a section of a common object file	ldrseek to relocation entries of	ldrseek(3X)
header of a common object file	ldshread indexed/named section	ldshread(3X)
section of a common object file	ldsseek to an indexed/named	ldsseek(3X)
entry of a common object file	ldtbindex of a symbol table	ldtbindex(3X)
entry of a common object file	ldtbread an indexed symbol table	ldtbread(3X)
table of a common object file	ldtbseek seek to the symbol	ldtbseek(3X)
return length of Fortran string	len	len(3F)
getopt get option	letter from argument vector	getopt(3C)
generate programs for simple	lexical tasks lex	lex(1)
ordering relation for an object	library lorder find	lorder(1)
archives ar archive and	library maintainer for portable	ar(1)
bar Berkeley archive and	library maintainer	bar(1)
get and set user	limits ulimit	ulimit(2)
establish an out-going terminal	line connection dial	dial(3C)
read one	line line	line(1)
object file linenum	line number entries in a common	linenum(4)
object file manipulate	line number entries of a common	ldlread(3X)
of a common object seek to	line number entries of a section	ldlseek(3X)
common object strip symbol and	line number information from a	strip(1)
	line numbering filter nl	nl(1)
cut out selected fields of each	line of a file cut	cut(1)
send/cancel requests to an LP	line printer lp	lp(1)
read one line	line	line(1)
filter reverse	line-feeds col	col(1)
	linear search and update lsearch	lsearch(3C)
entries in a common object file	linenum line number	linenum(4)
comm select or reject	lines common to two sorted files	comm(1)
report repeated	lines in a file uniq	uniq(1)
subsequent lines of merge same	lines of several files or	paste(1)
exercise	link and unlink system calls link	link(1m)
information L-devices	link devices, connection	L-devices(4)
files ld	link editor for common object	ld(1)
common assembler and	link editor output a.out	a.out(4)

copy,	link or move files cp	cp(1)
L.sys	link systems	L.sys(4)
	link to a file link	link(2)
link to a file	link	link(2)
link and unlink system calls	link exercise	link(1m)
a C program checker	lint	lint(1)
ls	list contents of directories	ls(8)
	list contents of directory ls	ls(1)
for a file system ff	list file names and statistics	ff(1m)
get entries from name	list nlist	nlist(3C)
print name	list of common object file nm	nm(1)
fsck checklist	list of file systems processed by	checklist(4)
handle variable argument	list varargs	varargs(5)
output of a varargs argument	list vprintf print formatted	vprintf(3S)
construct argument	list(s) and execute command xargs	xargs(1)
macref produce cross-reference	listing of macro files	macref(1)
index return	location of Fortran substring	index(3F)
lst	locations in program end	end(3C)
memory plock	lock process, text, or data in	plock(2)
	log gamma function gamma	gamma(3M)
	log in to a new group newgrp	newgrp(1)
logarithm intrinsic function	log Fortran natural	log(3F)
logarithm intrinsic function	log10 Fortran common	log10(3F)
Fortran natural	logarithm intrinsic function log	log(3F)
function exp exponential,	logarithm, power, square root	exp(3M)
process a report of	logged errors errpt	errpt(1m)
configure	logical disks	dconfig(1m)
get	login name getlogin	getlogin(3C)
get	login name logname	logname(1)
get character	login name of the user cuserid	cuserid(3S)
return	login name of user logname	logname(3X)
change	login password passwd	passwd(1)
setting up an environment at	login time profile	profile(4)
setting up an environment at	login time profile	profile(5)
sign on	login	login(1)
get login name	logname	logname(1)
return login name of user	logname	logname(3X)
relation for an object library	lorder find ordering	lorder(1)
run a command at	low priority nice	nice(1)
requests to an LP line printer	lp send/cancel	lp(1)
configure the LP spooling system	lpadmin	lpadmin(1m)
resume printing	lphold postpone printing,	lphold(1)
scheduler and move requests	lpsched the LP request	lpsched(1m)
print LP status information	lpstat	lpstat(1)
directories	ls list contents of	ls(8)
list contents of directory	ls	ls(1)
linear search and update	lsearch	lsearch(3C)
pointer	lseek move read/write file	lseek(2s)
move read/write file pointer	lseek	lseek(2)
	lst locations in program end	end(3C)
macro processor	m4	m4(1)
	machine-dependent values values	values(5)
access long integer data in a	machine-independent fashion.	sputl(3X)
listing of macro files	macref produce cross-reference	macref(1)
permuted index mptx the	macro package for formatting a	mptx(5)

documents mm the MM	macro package for formatting	mm(5)
mosd the OSDD adapter	macro package for formatting	mosd(5)
graphs mv a	macro package for making view	mv(7)
viewgraphs and mv a troff	macro package for typesetting	mv(5)
	macro processor m4	m4(1)
in this manual man	macros for formatting entries	man(5)
send	mail to users or read mail mail	mail(1)
binary file for transmission via	mail /encode/decode a	uuencode(1c)
message processing system	mailx interactive	mailx(1)
	main memory allocator malloc	malloc(3C)
fast	main memory allocator malloc	malloc(3X)
regenerate groups of programs	maintain, update, and	make(1)
ar archive and library	maintainer for portable archives	ar(1)
SCCS file delta	make a delta (change) to an	delta(1)
	make a directory mkdir	mkdir(1)
or ordinary file mknod	make a directory or a special	mknod(2)
mknod	make a special file	mknod(2s)
	make a unique file name mktemp	mktemp(3C)
	make posters banner	banner(1)
generate encryption key	makekey	makekey(1)
mv a macro package for	making view graphs	mv(7)
main memory allocator	malloc	malloc(3C)
fast main memory allocator	malloc	malloc(3X)
tsearch	manage binary search trees	tsearch(3C)
	manage hash search tables hsearch	hsearch(3C)
a common object file function	manipulate line number entries of	ldlread(3X)
floating-point numbers frexp	manipulate parts of	frexp(3S)
print entries in this	manual man	man(1)
	map of ASCII character set ascii	ascii(5)
diffmk	mark differences between files	diffmk(1)
set file-creation mode	mask umask	umask(1)
set and get file creation	mask umask	umask(2)
set and get file creation	mask umask	umask(2s)
regular expression compile and	match routines regexp	regexp(5)
math functions and constants	math	math(5)
error-handling function	matherr	matherr(3M)
Fortran maximum-value functions	max	max(3F)
Fortran	maximum-value functions max	max(3F)
generate a	maze maze	maze(6)
return Fortran time accounting	mclock	mclock(3F)
modem	mdial dial the P/75 onboard	dial(1)
core memory	mem	mem(7)
read the archive header of a	member of an archive file	ldahread(3X)
main	memory allocator malloc	malloc(3C)
fast main	memory allocator malloc	malloc(3X)
ramdisk	memory as disk	ramdisk(1m)
shared	memory control operations shmctl	shmctl(2)
queue, semaphore set or shared	memory id ipcrm remove a message	ipcrm(1)
core	memory mem	mem(7)
	memory operations memory	memory(3C)
shared	memory operations shmop	shmop(2)
lock process, text, or data in	memory plock	plock(2)
get shared	memory segment shmget	shmget(2)
rram allows	memory to be used as a disk	rram(7)
memory operations	memory	memory(3C)

sort and/or	merge files sort	sort(1)
files acctmerg	merge or add total accounting	acctmerg(1m)
or subsequent lines of one	merge same lines of several files	paste(1)
permit or deny messages	mesg	mesg(1)
msgctl	message control operations	msgctl(2)
	message operations msgop	msgop(2)
interactive	message processing system mailx	mailx(1)
get	message queue msgget	msgget(2)
shared memory id remove a	message queue, semaphore set or	ipcrm(1)
permit or deny	messages mesg	mesg(1)
system error	messages perror	perror(3C)
Fortran minimum-value functions	min	min(3F)
two identical	mirutil utility for connecting	mirutil(1m)
overview of accounting and	miscellaneous accounting commands	acct(1m)
make a directory	mkdir	mkdir(1)
	mkfs construct a file system	mkfs(8)
construct a file system	mkfs	mkfs(1m)
	mknod make a special file	mknod(2s)
build special file	mknod	mknod(1m)
or a special or ordinary file	mknod make a directory	mknod(2)
make a unique file name	mktemp	mktemp(3C)
formatted with the MM macros	mm print/check documents	mm(1)
formatting documents	mm the MM macro package for	mm(5)
document compatibility checker	mmlint sroff/MM nroff/MM	mmlint(1)
viewgraphs, and slides	mmt typeset documents,	mmt(1)
mounted file system table	mnttab	mnttab(4)
remaindering intrinsic functions	mod Fortran	mod(3F)
change	mode chmod	chmod(1)
set file-creation	mode mask umask	umask(1)
chmod change	mode of file	chmod(2s)
dial a Racal-Vadic 3451	modem	dial(1)
mdial dial the P/75 onboard	modem	dial(1)
a compiler/interpreter for	modest-sized programs bs	bs(1)
touch update access and	modification times of a file	touch(1)
set file access and	modification times utime	utime(2)
	monitor uucp network uusub	uusub(1m)
prepare execution profile	monitor	monitor(3C)
guessing game	moo	moo(6)
package for formatting documents	mosd the OSDD adapter macro	mosd(5)
mount	mount a file system	mount(2s)
mount	mount and dismount file system	mount(1m)
establish	mount table setmnt	setmnt(1m)
mount and dismount file system	mount	mount(1m)
mount a file system	mount	mount(2)
mnttab	mounted file system table	mnttab(4)
	move a directory mvdir	mvdir(1m)
copy, link or	move files cp	cp(1)
lseek	move read/write file pointer	lseek(2)
lseek	move read/write file pointer	lseek(2s)
the LP request scheduler and	move requests lpsched start/stop	lpsched(1m)
formatting a permuted index	mptx the macro package for	mptx(5)
message control operations	msgctl	msgctl(2)
get message queue	msgget	msgget(2)
message operations	msgop	msgop(2)
view graphs	mv a macro package for making	mv(7)

typesetting viewgraphs and	mv a troff macro package for	mv(5)
move a directory	mvdir	mvdir(1m)
device	name devnm	devnm(1m)
create a	name for a temporary file tmpnam	tmpnam(3S)
symbol table retrieve symbol	name for common object file	ldgetname(3X)
generate file	name for terminal ctermid	ctermid(3S)
get	name from UID getpw	getpw(3C)
return value for environment	name getenv	getenv(3C)
get login	name getlogin	getlogin(3C)
get entries from	name list nlist	nlist(3C)
nm print	name list of common object file	nm(1)
get login	name logname	logname(1)
make a unique file	name mktemp	mktemp(3C)
find	name of a terminal ttyname	ttyname(3C)
print	name of current UNIX system	uname(1)
get	name of current UNIX system	uname(2)
get the	name of the terminal tty	tty(1)
get character login	name of the user cuserid	cuserid(3S)
return login	name of user logname	logname(3X)
working directory	name pwd	pwd(1)
system ff list file	names and statistics for a file	ff(1m)
deliver portions of path	names basename	basename(1)
conventional	names for terminals term	term(5)
generate	names from i-numbers ncheck	ncheck(1m)
print user and group IDs and	names id	id(1)
function log Fortran	natural logarithm intrinsic	log(3F)
generate names from i-numbers	ncheck	ncheck(1m)
Fortran	nearest integer functions round	round(3F)
character definitions for eqn and	neqn eqnchar special	eqnchar(5)
a command on the PCL network	net execute	net(1)
execute a command on the PCL	network net	net(1)
commands stat statistical	network useful with graphical	stat(1)
monitor uucp	network uusub	uusub(1m)
change the format of a text file	newform	newform(1)
log in to a new group	newgrp	newgrp(1)
print	news items news	news(1)
process	nice change priority of a	nice(2s)
run a command at low priority	nice	nice(1)
change priority of a process	nice	nice(2)
line numbering filter	nl	nl(1)
get entries from name list	nlist	nlist(3C)
name list of common object file	nm print	nm(1)
immune to hangups and quits	nohup run a command	nohup(1)
9700 printer x9700 prepare	nroff documents for the Xerox	x9700(1)
	nroff format or typeset text	nroff(1)
tbl format tables for	nroff or troff	tbl(1)
checker mmlint sroff/MM	nroff/MM document compatibility	mmlint(1)
constructs deroff remove	nroff/troff, tbl, and eqn	deroff(1)
the	null file null	null(7)
file linenum line	number entries in a common object	linenum(4)
file function manipulate line	number entries of a common object	ldlread(3X)
common object seek to line	number entries of a section of a	ldlseek(3X)
factor a	number factor	factor(1)
provide drill in	number facts arithmetic	arithmetic(6)
random	number generator rand	rand(3F)

object strip symbol and line	number info from a common	strip(1)
report	number of free disk blocks df	df(1m)
string to double-precision	number strtod convert	strtod(3C)
convert floating-point	number to string ecvt	ecvt(3C)
line	numbering filter nl	nl(1)
distributed pseudo-random	numbers drand48 uniformly	drand48(3C)
parts of floating-point	numbers frexp manipulate	frexp(3S)
access graphical and	numerical commands graphics	graphics(1)
common	object file access routines ldfcn	ldfcn(4)
dump selected parts of an	object file dump	dump(1)
open a common	object file for reading ldopen	ldopen(3X)
line number entries of a common	object file function ldlread	ldlread(3X)
close a common	object file ldclose	ldclose(3X)
read the file header of a common	object file ldfhread	ldfhread(3X)
entries of a section of a common	object file ldlseek line number	ldlseek(3X)
optional file header of a common	object file ldohseek seek to the	ldohseek(3X)
entries of a section of a common	object file ldrseek relocation	ldrseek(3X)
section header of a common	object file ldshread	ldshread(3X)
section of a common	object file ldsseek	ldsseek(3X)
a symbol table entry of a common	object file ldtbindex index of	ldtbindex(3X)
symbol table entry of a common	object file ldtbread an indexed	ldtbread(3X)
to the symbol table of a common	object file ldtbseek seek	ldtbseek(3X)
line number entries in a common	object file linenum	linenum(4)
print name list of common	object file nm	nm(1)
information for a common	object file reloc relocation	reloc(4)
section header for a common	object file scnhdr	scnhdr(4)
number information from a common	object file strip and line	strip(1)
retrieve symbol name for common	object file symbol table entry	ldgetname(3X)
syms common	object file symbol table format	syms(4)
file header for common	object files filehdr	filehdr(4)
directories cpset install	object files in binary	cpset(1m)
link editor for common	object files ld	ld(1)
print section sizes of common	object files size	size(1)
find ordering relation for an	object library lorder	lorder(1)
od	octal dump	od(8)
text for otroff	ocw prepare constant-width	ocw(1)
	od octal dump	od(8)
octal dump	od	od(1)
mdial dial the P/75	onboard modem	dial(1)
reading ldopen	open a common object file for	ldopen(3X)
	open a stream fopen	fopen(3S)
duplicate an	open file descriptor dup	dup(2)
open	open for reading or writing	open(2)
open	open for reading or writing	open(2s)
open for reading or writing	open	open(2)
profiler	operating system profiler	profiler(1m)
terminal independent	operation routines	termlib(3c)
memory	operations memory	memory(3C)
message control	operations msgctl	msgctl(2)
message	operations msgop	msgop(2)
semaphore control	operations semctl	semctl(2)
semaphore	operations semop	semop(2)
shared memory control	operations shmctl	shmctl(2)
shared memory	operations shmop	shmop(2)
string	operations string	string(3C)

relational database	operator join	join(1)
copy file systems for	optimal access time dcopy	dcopy(1m)
CRT screen handling and	optimization package curses	curses(3X)
vector getopt get	option letter from argument	getopt(3C)
object file seek to the	optional file header of a common	ldohseek(3X)
file control	options fcntl	fcntl(5)
set the	options for a terminal stty	stty(1)
parse command	options getopt	getopt(1)
library lorder find	ordering relation for an object	lorder(1)
make a directory or a special or	ordinary file mknod	mknod(2)
prepare constant-width text for	otroff ocw	ocw(1)
connection dial establish an	out-going terminal line	dial(3C)
common assembler and link editor	output a.out	a.out(4)
tc troff	output interpreter	tc(1)
troff description of	output language	troff(5)
vprintf print formatted	output of a varargs argument list	vprintf(3S)
print formatted	output printf	printf(3S)
miscellaneous accounting	overview of accounting and	acct(1m)
change	owner and group of a file chown	chown(2)
change	owner or group chown	chown(1)
compress and expand files	pack	pack(1)
screen handling and optimization	package curses CRT	curses(3X)
mv a macro	package for making view graphs	mv(7)
system activity report	package sar	sar(1m)
standard buffered input/output	package stdio	stdio(3S)
interprocess communication	package stdipc standard	stdipc(3C)
4014 terminal 4014	paginator for the TEKTRONIX	4014(1)
pp	parallel port interface	pp(7)
get process, process group, and	parent process IDs getpid	getpid(2)
	parse command options getopt	getopt(1)
Fortran integer	part intrinsic function aint	aint(3F)
deliver the last	part of a file tail	tail(1)
Fortran imaginary	part of complex argument aimag	aimag(3F)
dump selected	parts of an object file dump	dump(1)
frexp manipulate	parts of floating-point numbers	frexp(3S)
change login password	passwd	passwd(1)
password file	passwd	passwd(4)
get	password file entry getpwent	getpwent(3C)
write	password file entry putpwent	putpwent(3C)
	password file passwd	passwd(4)
read a	password getpass	getpass(3C)
change login	password passwd	passwd(1)
pwck	password/group file checkers	pwck(1m)
or subsequent lines of one file	paste lines of several files	paste(1)
deliver portions of	path names basename	basename(1)
directory getcwd get	path-name of current working	getcwd(3C)
USERFILE UUCP	pathname permissions file	USERFILE(4)
search a file for a	pattern grep	grep(1)
language awk	pattern scanning and processing	awk(1)
suspend process until signal	pause	pause(2)
format acct	per-process accounting file	acct(4)
acctcms command summary from	per-process accounting records	acctcms(1m)
USERFILE UUCP pathname	permissions file	USERFILE(4)
	permit or deny messages mesg	mesg(1)
	permuted index ptx	ptx(1)

macro package for formatting a	permuted index mptx the	mptx(5)
system error messages	perror	perror(3C)
terminals pg file	perusal filter for soft-copy	pg(1)
drawing simple pictures	pic troff preprocessor for	pic(1)
preprocessor for drawing simple	pictures pic troff	pic(1)
split a file into	pieces split	split(1)
initiate	pipe to/from a process popen	popen(3S)
create an interprocess channel	pipe	pipe(2)
process, text, or data in memory	plock lock	plock(2)
graphics interface subroutines	plot	plot(3X)
graphics interface	plot	plot(4)
reposition a file	pointer in a stream fseek	fseek(3S)
move read/write file	pointer lseek	lseek(2)
lseek move read/write file	pointer	lseek(2s)
the current value of a file	pointer tell report	tell(2s)
initiate pipe to/from a process	popen	popen(3S)
pp parallel	port interface	pp(7)
data base of terminal types by	port ttytype	ttytype(5)
and library maintainer for	portable archives ar archive	ar(1)
deliver	portions of path names basename	basename(1)
number on a tape srcheof	position to a specific file	srcheof(2s)
functions dim	positive difference intrinsic	dim(3F)
make	posters banner	banner(1)
printing lphold	postpone printing, resume	lphold(1)
exponential, logarithm,	power, square root function exp	exp(3M)
	pp parallel port interface	pp(7)
print files	pr	pr(1)
function dprod double	precision product intrinsic	dprod(3F)
monitor	prepare execution profile	monitor(3C)
Xerox 9700 printer x9700	prepare nroff documents for the	x9700(1)
Xerox 9700 printer dx9700	prepare troff documents for the	dx9700(1)
the C language	preprocessor cpp	cpp(1)
pictures pic troff	preprocessor for drawing simple	pic(1)
unget undo a	previous get of an SCCS file	unget(1)
graphical files gps graphical	primitive string, format of	gps(4)
types	primitive system data types	types(5)
lpstat	print LP status information	lpstat(1)
	print an SCCS file prs	prs(1)
	print and set the date date	date(1)
	print calendar cal	cal(1)
of a file sum	print checksum and block count	sum(1)
activity sact	print current SCCS file editing	sact(1)
man	print entries in this manual	man(1)
concatenate and	print files cat	cat(1)
	print files pr	pr(1)
cat concatenate and	print files	cat(8)
varargs argument list vprintf	print formatted output of a	vprintf(3S)
	print formatted output printf	printf(3S)
file nm	print name list of common object	nm(1)
system uname	print name of current UNIX	uname(1)
	print news items news	news(1)
acctcom search and	print process accounting file(s)	acctcom(1)
topq prioritize	print queue	topq(1m)
object files size	print section sizes of common	size(1)
names id	print user and group IDs and	id(1)

with the MM macros mm	print/check documents formatted	mm(1)
requests to an LP line	printer lp send/cancel	lp(1)
documents for the Xerox 9700	printer prepare nroff	x9700(1)
documents for the Xerox 9700	printer prepare troff	dx9700(1)
enable/disable LP	printers enable	enable(1)
print formatted output	printf	printf(3S)
postpone printing, resume	printing	lphold(1)
topq	prioritize print queue	topq(1m)
run a command at low	priority nice	nice(1)
change	priority of a process nice	nice(2)
nice change	priority of a process	nice(2s)
faster file system checking	procedure checkall	checkall(1m)
getpid get	process ID	getpid(2s)
process group, and parent	process IDs getpid get process,	getpid(2)
errpt	process a report of logged errors	errpt(1m)
errors errpt	process a report of logged	errpt(1m)
enable or disable	process accounting acct	acct(2)
acctcom search and print	process accounting file(s)	acctcom(1)
set a	process alarm clock alarm	alarm(2)
get	process and child process times	times(2)
init	process control initialization	init(1m)
timex time a command; report	process data and system activity	timex(1)
terminate	process exit	exit(2)
create a new	process fork	fork(2)
set	process group ID setpgrp	setpgrp(2)
IDs getpid get process,	process group, and parent process	getpid(2)
script for the init	process inittab	inittab(4)
terminate a	process kill	kill(1)
change priority of a	process nice	nice(2)
kill send a signal to a	process or a group of processes	kill(2)
initiate pipe to/from a	process popen	popen(3S)
report	process status ps	ps(1)
get process and child	process times	times(2)
wait for child	process to stop or terminate wait	wait(2)
	process trace ptrace	ptrace(2)
suspend	process until signal pause	pause(2)
await completion of	process wait	wait(1)
exit terminate	process	exit(2s)
nice change priority of a	process	nice(2s)
parent process IDs getpid get	process, process group, and	getpid(2)
plock lock	process, text, or data in memory	plock(2)
list of file systems	processed by fsck checklist	checklist(4)
signal to a process or a group of	processes kill send a	kill(2)
kill all active	processes killall	killall(1m)
structure fuser identify	processes using a file or file	fuser(1m)
pattern scanning and	processing language awk	awk(1)
terminate all	processing shutdown	shutdown(1m)
interactive message	processing system mailx	mailx(1)
macro	processor m4	m4(1)
double precision	product intrinsic function dprod	dprod(3F)
display profile data	prof	prof(1)
profile within a function	prof	prof(5)
execution time profile	profil	profil(2)
display	profile data prof	prof(1)
prepare execution	profile monitor	monitor(3C)

execution time	profile profil	profil(2)
environment at login time	profile setting up an	profile(5)
	profile within a function prof	prof(5)
up an environment at login time	profile setting	profile(4)
disk access	profiler sadp	sadp(1m)
operating system profiler	profiler	profiler(1m)
terminate Fortran	program abort	abort(3F)
verify	program assertion assert	assert(3X)
C	program beautifier cb	cb(1)
a C	program checker lint	lint(1)
generate C	program cross-reference cxref	cxref(1)
C	program debugger ctrace	ctrace(1)
lst locations in	program end	end(3C)
for getargv display a	program name and get arguments	getargv(2s)
side-by-side difference	program sdiff	sdiff(1)
conversion	program units	units(1)
the standard/restricted command	programming language sh shell,	sh(1)
for modest-sized	programs bs compiler/interpreter	bs(1)
lex generate	programs for simple lexical tasks	lex(1)
update, and regenerate groups of	programs make maintain,	make(1)
arithmetic	provide drill in number facts	arithmetic(6)
	provide truth values true	true(1)
print an SCCS file	prs	prs(1)
report process status	ps	ps(1)
generate uniformly distributed	pseudo-random numbers drand48	drand48(3C)
	pt IMSP cartridge controller	pt(7)
process trace	ptrace	ptrace(2)
permuted index	ptx	ptx(1)
copy uuto	public UNIX-to-UNIX system file	uuto(1)
stream ungetc	push character back into input	ungetc(3S)
	put a string on a stream puts	puts(3S)
stream putc	put character or word on a	putc(3S)
or add value to environment	putenv change	putenv(3C)
write password file entry	putpwent	putpwent(3C)
put a string on a stream	puts	puts(3S)
password/group file checkers	pwck	pwck(1m)
working directory name	pwd	pwd(1)
quicker sort	qsort	qsort(3C)
	query terminfo database tput	tput(1)
get message	queue msgget	msgget(2)
topq prioritize print	queue	topq(1m)
memory id remove a message	queue, semaphore set or shared	ipcrm(1)
	quicker sort qsort	qsort(3C)
a command immune to hangups and	quits nohup run	nohup(1)
test your knowledge	quiz	quiz(6)
	ramdisk memory as disk	ramdisk(1m)
simple random-number generator	rand	rand(3C)
random number generator	rand	rand(3F)
simple	random-number generator rand	rand(3C)
rational Fortran dialect	ratfor	ratfor(1)
split f77,	ratfor, or efl files fsplit	fsplit(1)
	rational Fortran dialect ratfor	ratfor(1)
	read a password getpass	getpass(3C)
entry of a common object file	read an indexed symbol table	ldtbread(3X)
header of a common object file	read an indexed/named section	ldshread(3X)

	read from file read	read(2)
read	read from file	read(2s)
send mail to users or	read mail mail	mail(1)
	read one line line	line(1)
member of an archive file	read the archive header of a	ldahread(3X)
object file ldfhread	read the file header of a common	ldfhread(3X)
read from file	read	read(2)
move	read/write file pointer lseek	lseek(2)
lseek move	read/write file pointer	lseek(2s)
open a common object file for	reading ldopen	ldopen(3X)
open for	reading or writing open	open(2)
open open for	reading or writing	open(2s)
get real and effective user, and	real and effective group ID	getuid(2s)
get real and effective user,	real and effective group ID's	getuid(2)
get real and errective user,	real and effective group,	getuid(2)
real effective group get	real and effective user,	getuid(2)
specify what to do upon	receipt of a signal	signal(2)
specify Fortran action on	receipt of a system signal	signal(3F)
from per-process accounting	records acctcms command summary	acctcms(1m)
extract error	records from dump errdead	errdead(1m)
tape frec	recover files from a backup	frec(1m)
regular expression compile	regcmp	regcmp(1)
and execute regular expression	regcmp compile	regcmp(3X)
make maintain, update, and	regenerate groups of programs	make(1)
compile and match routines	regexp regular expression	regexp(5)
match routines regexp	regular expression compile and	regexp(5)
	regular expression compile regcmp	regcmp(1)
regcmp	regular expression compile	regcmp(1)
compile and execute	regular expression regcmp	regcmp(3X)
files comm select or	reject lines common to two sorted	comm(1)
lorder find ordering	relation for an object library	lorder(1)
join	relational database operator	join(1)
for a common object file	reloc relocation information	reloc(4)
of a common object seek to	relocation entries of a section	ldrseek(3X)
common object file reloc	relocation information for a	reloc(4)
functions floor, ceiling,	remainder, absolute value	floor(3M)
mod Fortran	remaindering intrinsic functions	mod(3F)
	reminder service calendar	calendar(1)
uuxqt execute	remote command requests	uuxqt(1m)
L.cmds	remote execution commands	L.cmds(4)
spawn getty to a	remote terminal ct	ct(1)
file rmdel	remove a delta from an SCCS	rmdel(1)
semaphore set or shared memory	remove a message queue,	ipcrm(1)
	remove directory entry unlink	unlink(2)
	remove files or directories rm	rm(1)
constructs deroff	remove nroff/troff, tbl, and eqn	deroff(1)
consistency check and interactive	repair fsck file system	fsck(1m)
consistency check and interactive	repair fsck file system	fsck(8)
report	repeated lines in a file uniq	uniq(1)
	report CPU time used clock	clock(3C)
communication facilities	report inter-process	ipcs(1)
blocks df	report number of free disk	df(1m)
process a	report of logged errors errpt	errpt(1m)
system activity	report package sar	sar(1m)
activity timex time a command;	report process data and system	timex(1)

	report process status ps	ps(1)
uniq	report repeated lines in a file	uniq(1)
system activity	report sail	sail(1)
stream fseek	reposition a file pointer in a	fseek(3S)
requests start/stop the LP	request scheduler and move	lpsched(1m)
send/cancel	requests to an LP line printer lp	lp(1)
uuxqt execute remote command	requests	uuxqt(1m)
restore	restor incremental file system	restor(8)
incremental file system	restore	restor(1m)
lphold postpone printing,	resume printing	lphold(1)
object file symbol table entry	retrieve symbol name for common	ldgetname(3X)
argument getarc	return Fortran command-line	getarc(3F)
variable getenv	return Fortran environment	getenv(3F)
mclock	return Fortran time accounting	mclock(3F)
abs	return integer absolute value	abs(3C)
len	return length of Fortran string	len(3F)
substring index	return location of Fortran	index(3F)
logname	return login name of user	logname(3X)
name getenv	return value for environment	getenv(3C)
data	returned by stat system call stat	stat(5)
filter	reverse line-feeds col	col(1)
create a new file or	rewrite an existing one creat	creat(2)
drive	rm Cipher Microstreamer tape	rm(7)
remove files or directories	rm	rm(1)
remove a delta from an SCCS file	rmdel	rmdel(1)
change	root directory chroot	chroot(2)
chroot change	root directory for a command	chroot(1m)
logarithm, power, square	root function exp exponential,	exp(3M)
Fortran square	root intrinsic function sqrt	sqrt(3F)
Fortran nearest integer functions	round	round(3F)
graphical device	routines and filters gdev	gdev(1)
common object file access	routines ldfcn	ldfcn(4)
expression compile and match	routines regexp regular	regexp(5)
graphical table of contents	routines toc	toc(1)
float and double	routines	float(2s)
as a disk	rram allows memory to be used	rram(7)
nice	run a command at low priority	nice(1)
and quits nohup	run comm immune to hngup	nohup(1)
run daily accounting	runacct	runacct(1m)
SCCS file editing activity	sact print current	sact(1)
disk access profiler	sadp	sadp(1m)
system activity graph	sag	sag(1)
system activity report	sail	sail(1)
system activity report package	sar	sar(1m)
convert formatted input	scanf	scanf(3S)
big file	scanner bfs	bfs(1)
awk pattern	scanning and processing language	awk(1)
programs	scc C compiler for stand-alone	scc(1)
two versions of an SCCS file	sccsdiff compare	sccsdiff(1)
format of SCCS file	sccsfile	sccsfile(4)
start/stop the LP request	scheduler and move requests	lpsched(1m)
header for a common object file	scnhdr section	scnhdr(4)
package curses CRT	screen handling and optimization	curses(3X)
editor based on ex vi	screen-oriented (visual) display	vi(1)
inittab	script for the init process	inittab(4)

system initialization shell	scripts brc	brc(1m)
side-by-side difference program	sdiff	sdiff(1)
grep	search a file for a pattern	grep(1)
binary	search a sorted table bsearch	bsearch(3C)
accounting file(s) acctcom	search and print process	acctcom(1)
linear	search and update lsearch	lsearch(3C)
manage hash	search tables hsearch	hsearch(3C)
manage binary	search trees tsearch	tsearch(3C)
object file scnhdr	section header for a common	scnhdr(4)
file read an indexed/named	section header of a common object	ldshread(3X)
seek to line number entries of a	section of a common object file	ldlseek(3X)
seek to relocation entries of a	section of a common object file	ldrseek(3X)
seek to an indexed/named	section of a common object file	ldsseek(3X)
files size print	section sizes of common object	size(1)
stream editor	sed	sed(1)
section of a common object	seek to line number entries of a	ldlseek(3X)
section of a common object file	seek to relocation entries of a	ldrseek(3X)
of a common object file	seek to the optional file header	ldohseek(3X)
common object file ldtbseek	seek to the symbol table of a	ldtbseek(3X)
get shared memory	segment shmget	shmget(2)
change data	segment space allocation brk	brk(2)
brk change data	segment space allocation	brk(2s)
to two sorted files comm	select or reject lines common	comm(1)
	select terminal filter greek	greek(1)
file cut cut out	selected fields of each line of a	cut(1)
dump dump	selected parts of an object file	dump(1)
semctl	semaphore control operations	semctl(2)
	semaphore operations semop	semop(2)
ipcrm remove a message queue,	semaphore set or shared memory id	ipcrm(1)
get set of	semaphores semget	semget(2)
semaphore control operations	semctl	semctl(2)
get set of semaphores	semget	semget(2)
semaphore operations	semop	semop(2)
group of processes kill	send a signal to a process or a	kill(2)
a group of processes kill	send a signal to a process or	kill(2s)
mail	send mail to users or read mail	mail(1)
line printer lp	send/cancel requests to an LP	lp(1)
reminder	service calendar	calendar(1)
	set a process alarm clock alarm	alarm(2)
umask	set and get file creation mask	umask(2)
umask	set and get file creation mask	umask(2s)
map of ASCII character	set ascii	ascii(5)
execution env	set environment for command	env(1)
modification times utime	set file access and	utime(2)
umask	set file-creation mode mask	umask(1)
get	set of semaphores semget	semget(2)
remove a message queue, semaphore	set or shared memory id ipcrm	ipcrm(1)
	set process group ID setpgrp	setpgrp(2)
	set tabs on a terminal tabs	tabs(1)
stty	set terminal characteristics	stty(2s)
print and	set the date date	date(1)
stty	set the options for a terminal	stty(1)
	set time stime	stime(2)
stime	set time	stime(2s)
	set user and group IDs setuid	setuid(2)

get and	set user limits ulimit	ulimit(2)
assign buffering to a stream	setbuf	setbuf(3S)
establish mount table	setmnt	setmnt(1m)
set process group ID	setpgrp	setpgrp(2)
login time profile	setting up an environment at	profile(4)
login time profile	setting up an environment at	profile(5)
speed and terminal	settings used by getty gettydefs	gettydefs(4)
set user and group IDs	setuid	setuid(2)
of one merge same lines of	several files or subsequent lines	paste(1)
command programming language	sh the standard/restricted	sh(1)
shmctl	shared memory control operations	shmctl(2)
a message queue, semaphore set or	shared memory id ipcrm remove	ipcrm(1)
	shared memory operations shmop	shmop(2)
get	shared memory segment shmget	shmget(2)
issue a	shell command from Fortran sys	system(3F)
issue a	shell command system	system(3S)
system initialization	shell scripts brc	brc(1m)
command programming language sh	shell, the standard/restricted	sh(1)
shared memory control operations	shmctl	shmctl(2)
get shared memory segment	shmget	shmget(2)
shared memory operations	shmop	shmop(2)
terminate all processing	shutdown	shutdown(1m)
sdiff	side-by-side difference program	sdiff(1)
	sign on login	login(1)
intrinsic function	sign Fortran transfer-of-sign	sign(3F)
suspend process until	signal pause	pause(2)
processes kill send a	signal to a process or a group of	kill(2)
processes kill send a	signal to a process or a group of	kill(2s)
on receipt of a system signal	signal specify Fortran action	signal(3F)
to do upon receipt of a signal	signal specify what	signal(2)
software	signals ssignal	ssignal(3C)
generate programs for	simple lexical tasks lex	lex(1)
troff preprocessor for drawing	simple pictures pic	pic(1)
rand	simple random-number generator	rand(3C)
Fortran	sine intrinsic function sin	sin(3F)
Fortran hyperbolic	sine intrinsic function sinh	sinh(3F)
hyperbolic functions	sinh	sinh(3M)
sine intrinsic function	sinh Fortran hyperbolic	sinh(3F)
print section	sizes of common object files size	size(1)
interval	sleep suspend execution for	sleep(2s)
suspend execution for an interval	sleep	sleep(1)
suspend execution for interval	sleep	sleep(3C)
for typesetting viewgraphs and	slides a troff macro package	mv(5)
documents, viewgraphs, and	slides mmt typeset	mmt(1)
current user ttyslot find the	slot in the utmp file of the	ttyslot(3C)
interpolate	smooth curve spline	spline(1)
SNOBOL interpreter	sno	sno(1)
file perusal filter for	soft-copy terminals pg	pg(1)
	software signals ssignal	ssignal(3C)
	sort and/or merge files sort	sort(1)
quicker	sort qsort	qsort(3C)
topological	sort tsort	tsort(1)
or reject lines common to two	sorted files comm select	comm(1)
binary search a	sorted table bsearch	bsearch(3C)
change data segment	space allocation brk	brk(2)

brk change data segment	space allocation	brk(2s)
terminal ct	spawn getty to a remote	ct(1)
for eqn and neqn eqnchar	special character definitions	eqnchar(5)
build	special file mknod	mknod(1m)
create a new	special file	creat(2s)
mknod make a	special file	mknod(2s)
300s terminals 300 handle	special functions of DASI 300 and	300(1)
2621-series terminals handle	special functions of HP 2640 and	hp(1)
terminal 450 handle	special functions of the DASI 450	450(1)
make a directory or a	special or ordinary file mknod	mknod(2)
format	specification in text files fspec	fspec(4)
isatty returns a 1 if	specified file descriptor is a	isatty(2s)
of a system signal signal	specify Fortran action on receipt	signal(3F)
of a signal signal	specify what to do upon receipt	signal(2)
by getty gettydefs	speed and terminal settings used	gettydefs(4)
find	spelling errors spell	spell(1)
interpolate smooth curve	spline	spline(1)
	split a file into pieces split	split(1)
context	split csplit	csplit(1)
fsplit	split f77, ratfor, or efl files	fsplit(1)
uucp	spool directory clean-up uuclean	uuclean(1m)
configure the LP	spooling system lpadmin	lpadmin(1m)
in a machine-independent fashion.	sputl access long integer data	sputl(3X)
square root intrinsic function	sqrt Fortran	sqrt(3F)
exponential, logarithm, power,	square root function exp	exp(3M)
sqrt Fortran	square root intrinsic function	sqrt(3F)
file number on a tape	srcheof position to a specific	srcheof(2s)
	sroff format text	sroff(1)
compatibility checker mmlint	sroff/MM nroff/MM document	mmlint(1)
software signals	ssignal	ssignal(3C)
scc C compiler for	stand-alone programs	scc(1)
package stdio	standard buffered input/output	stdio(3S)
communication package stdipc	standard interprocess	stdipc(3C)
programming shell, the	standard/restricted command	sh(1)
scheduler and move requests	start/stop the LP request	lpsched(1m)
	stat get file status	stat(2s)
data returned by	stat system call stat	stat(5)
get file status	stat	stat(2)
data returned by stat system call	stat	stat(5)
useful with graphical commands	stat statistical network	stat(1)
graphical commands stat	statistical network useful with	stat(1)
list file names and	statistics for a file system ff	ff(1m)
get file system	statistics ustat	ustat(2)
ustat get file system	statistics	ustat(2s)
print LP	status information lpstat	lpstat(1)
stream	status inquiries ferror	ferror(3S)
uustat uucp	status inquiry and job control	uustat(1)
communication facilities	status ipcs report inter-process	ipcs(1)
System control and	status program	sys(1m)
report process	status ps	ps(1)
get file	status stat	stat(2)
stat get file	status	stat(2s)
buffered input/output package	stdio standard	stdio(3S)
communication package	stdipc standard interprocess	stdipc(3C)
	stime set time	stime(2s)

set time	stime	stime(2)
wait for child process to	stop or terminate wait	wait(2)
comparision intrinsic functions	strcmp string	strcmp(3F)
	stream editor sed	sed(1)
close or flush a	stream fclose	fclose(3S)
open a	stream fopen	fopen(3S)
reposition a file pointer in a	stream fseek	fseek(3S)
get character or word from a	stream getc	getc(3S)
get a string from a	stream gets	gets(3S)
put character or word on a	stream putc	putc(3S)
put a string on a	stream puts	puts(3S)
assign buffering to a	stream setbuf	setbuf(3S)
	stream status inquiries ferror	ferror(3S)
push character back into input	stream ungetc	ungetc(3S)
ft IMSP	streaming cartridge controller	ft(7)
long integer and base-64 ASCII	string a64l convert between	a64l(3C)
functions strcmp	string comparision intrinsic	strcmp(3F)
convert date and time to	string ctime	ctime(3C)
convert floating-point number to	string ecvt	ecvt(3C)
get a	string from a stream gets	gets(3S)
return length of Fortran	string len	len(3F)
put a	string on a stream puts	puts(3S)
	string operations string	string(3C)
strtod convert	string to double-precision number	strtod(3C)
convert	string to integer strtol	strtol(3C)
string operations	string	string(3C)
gps graphical primitive	string, format of graphical files	gps(4)
information from a common	strip symbol and line number	strip(1)
from a common object file	strip line number information	strip(1)
string to double-precision number	strtod convert	strtod(3C)
convert string to integer	strtol	strtol(3C)
processes using a file or file	structure fuser identify	fuser(1m)
characteristics	stty set terminal	stty(2s)
set the options for a terminal	stty	stty(1)
become super-user or another user	su	su(1)
graphics interface	subroutines plot	plot(3X)
same lines of several files or	subsequent lines of one file	paste(1)
return location of Fortran	substring index	index(3F)
and block count of a file	sum print checksum	sum(1)
	summarize disk usage du	du(1)
du	summarize disk usage	du(8)
accounting records command	summary from per-process	acctcms(1m)
update the	super block sync	sync(1)
update	super-block sync	sync(2)
become	super-user or another user su	su(1)
document analyze	surface characteristics of a	style(1)
interval sleep	suspend execution for an	sleep(1)
sleep	suspend execution for interval	sleep(2s)
sleep	suspend execution for interval	sleep(3C)
pause	suspend process until signal	pause(2)
swap bytes	swab	swab(3C)
information from a strip	symbol and line number	strip(1)
file symbol table retrieve	symbol name for common object	ldgetname(3X)
name for common object file	symbol table entry ldgetname	ldgetname(3X)
object compute the index of a	symbol table entry of a common	ldtbindex(3X)

object file read an indexed	symbol table entry of a common	ldtbread(3X)
common object file	symbol table format syms	syms(4)
file ldtbseek seek to the	symbol table of a common object	ldtbseek(3X)
object file symbol table format	syms common	syms(4)
update the super block	sync	sync(1)
update super-block	sync	sync(2)
	system activity graph sag	sag(1)
sar	system activity report package	sar(1m)
	system activity report sail	sail(1)
command; report process data and	system activity timex time a	timex(1)
daily/weekly UNIX system file	system backup filesave	filesave(1m)
data returned by stat	system call stat	stat(5)
exercise link and unlink	system calls link	link(1m)
checkall faster file	system checking procedure	checkall(1m)
UNIX-to-UNIX	system command execution uux	uux(1)
interactive repair fsck file	system consistency check and	fsck(1m)
interactive repair fsck file	system consistency check and	fsck(8)
file ioctl.syscon	system console configuration	ioctl.syscon(4)
UNIX system to UNIX	system copy uucp	uucp(1)
what to do when the	system crashes crash	crash(8)
call another UNIX	system cu	cu(1)
primitive	system data types types	types(5)
file	system debugger fsdb	fsdb(1m)
fsdb file	system debugger	fsdb(8)
	system error messages perror	perror(3C)
names and statistics for a file	system ff list file	ff(1m)
public UNIX-to-UNIX	system file copy uuto	uuto(1)
filesave daily/weekly UNIX	system file system backup	filesave(1m)
examine	system images crash	crash(1m)
scripts brc	system initialization shell	brc(1m)
configure the LP spooling	system lpadmin	lpadmin(1m)
interactive message processing	system mailx	mailx(1)
construct a file	system mkfs	mkfs(1m)
mount and dismount file	system mount	mount(1m)
mount a file	system mount	mount(2)
operating	system profiler profiler	profiler(1m)
incremental file	system restore	restor(1m)
Fortran action on receipt of a	system signal signal specify	signal(3F)
get file	system statistics ustat	ustat(2)
ustat get file	system statistics	ustat(2s)
mounted file	system table mnttab	mnttab(4)
UNIX	system to UNIX system copy uucp	uucp(1)
unmount a file	system umount	umount(2)
print name of current UNIX	system uname	uname(1)
get name of current UNIX	system uname	uname(2)
format of	system volume fs	fs(4)
who is on the	system who	who(1)
mount a file	system	mount(2s)
issue a shell command	system	system(3S)
a shell command from Fortran	system issue	system(3F)
dcopy copy file	systems for optimal access time	dcopy(1m)
checklist list of file	systems processed by fsck	checklist(4)
volcopy copy file	systems with label checking	volcopy(1m)
binary search a sorted	table bsearch	bsearch(3C)
for common object file symbol	table entry ldgetname name	ldgetname(3X)

compute the index of a symbol	table entry of a common object	ldtbindex(3X)
file read an indexed symbol	table entry of a common object	ldtbread(3X)
common object file symbol	table format syms	syms(4)
mounted file system	table mnttab	mnttab(4)
ldtbseek seek to the symbol	table of a common object file	ldtbseek(3X)
graphical	table of contents routines toc	toc(1)
establish mount	table setmnt	setmnt(1m)
tbl format	tables for nroff or troff	tbl(1)
manage hash search	tables hsearch	hsearch(3C)
set tabs on a terminal	tabs	tabs(1)
deliver the last part of a file	tail	tail(1)
Fortran	tangent intrinsic function tan	tan(3F)
tangent intrinsic function	tanh Fortran hyperbolic	tanh(3F)
fbackup make a fast	tape backup of a file system	fbackup(1m)
fbackup make a fast	tape backup of a file system	fbackup(8)
rm Cipher Microstreamer	tape drive	rm(7)
	tape file archiver tar	tar(1)
dump incremental dump	tape format	dump(4)
recover files from a backup	tape frec	frec(1m)
tape file archiver	tar	tar(1)
programs for simple lexical	tasks lex generate	lex(1)
troff	tbl format tables for nroff or	tbl(1)
deroff remove nroff/troff,	tbl, and eqn constructs	deroff(1)
	tc troff output interpreter	tc(1)
of a file pointer	tell report the current value	tell(2s)
create a	temporary file tmpfile	tmpfile(3S)
create a name for a	temporary file tmpnam	tmpnam(3S)
format of compiled term file.	term	term(4)
conventional names for terminals	term	term(5)
data base	termcap terminal capability	termcap(4)
paginator for the TEKTRONIX 4014	terminal 4014	4014(1)
special functions of the DASI 450	terminal 450 handle	450(1)
EOT on the other	terminal and exits. write	write(1)
termcap	terminal capability data base	termcap(4)
terminfo	terminal capability data base	terminfo(4)
gtty get	terminal characterisitcs	gtty(2s)
stty set	terminal characteristics	stty(2s)
spawn getty to a remote	terminal ct	ct(1)
generate file name for	terminal ctermid	ctermid(3S)
select	terminal filter greek	greek(1)
routines termlib	terminal independent operation	termlib(3c)
controlling	terminal interface tty	tty(7)
tty general	terminal interface	tty(7)
establish an out-going	terminal line connection dial	dial(3C)
gettydefs speed and	terminal settings used by getty	gettydefs(4)
set the options for a	terminal stty	stty(1)
set tabs on a	terminal tabs	tabs(1)
get the name of the	terminal tty	tty(1)
find name of a	terminal ttyname	ttyname(3C)
ttytype data base of	terminal types by port	ttytype(5)
functions of DASI 300 and 300s	terminals 300 handle special	300(1)
of HP 2640 and 2621-series	terminals hp special functions	hp(1)
file perusal filter for soft-copy	terminals pg	pg(1)
conventional names for	terminals term	term(5)
	terminate Fortran program abort	abort(3F)

	terminate a process kill	kill(1)
shutdown	terminate all processing	shutdown(1m)
	terminate process exit	exit(2)
exit	terminate process	exit(2s)
daemon errstop	terminate the error-logging	errstop(1m)
wait for child process to stop or	terminate wait	wait(2)
	terminfo compiler tic	tic(1m)
query	terminfo database tput	tput(1)
terminal capability data base	terminfo	terminfo(4)
operation routines	termlib terminal independent	termlib(3c)
command	test condition evaluation	test(1)
	test your knowledge quiz	quiz(6)
condition evaluation command	test	test(1)
	text editor ed	ed(1)
	text editor ex	ex(1)
casual users edit	text editor variant of ex for	edit(1)
change the format of a	text file newform	newform(1)
format specification in	text files fspec	fspec(4)
eqn format mathematical	text for nroff or troff	eqn(1)
ocw prepare constant-width	text for otroff	ocw(1)
troff	text formatting and typesetting	troff(1)
nroff format or typeset	text	nroff(1)
sroff format	text	sroff(1)
lock process,	text, or data in memory plock	plock(2)
	the C language preprocessor cpp	cpp(1)
	the game of backgammon back	back(6)
	the game of black jack bj	bj(6)
	the game of craps craps	craps(6)
wump	the game of hunt-the-wumpus	wump(6)
	the null file null	null(7)
terminfo compiler	tic	tic(1m)
	time a command time	time(1)
data and system activity timex	time a command; report process	timex(1)
update access and modification	times of a file touch	touch(1)
set file access and modification	times utime	utime(2)
get process and child process	times	times(2)
process data and system activity	timex time a command; report	timex(1)
create a temporary file	tmpfile	tmpfile(3S)
a name for a temporary file	tmpnam create	tmpnam(3S)
initiate pipe	to/from a process popen	popen(3S)
table of contents routines	toc graphical	toc(1)
	topological sort tsort	tsort(1)
	topq prioritize print queue	topq(1m)
merge or add	total accounting files acctmerg	acctmerg(1m)
and modification times of a file	touch update access	touch(1)
graphics filters	tplot	tplot(1)
query terminfo database	tput	tput(1)
translate characters	tr	tr(1)
process	trace ptrace	ptrace(2)
function sign Fortran	transfer-of-sign intrinsic	sign(3F)
	translate characters conv	conv(3C)
	translate characters tr	tr(1)
system uucico file	transport program for the uucp	uucico(1m)
walk a file	tree ftw	ftw(3C)
manage binary search	trees tsearch	tsearch(3C)

trigonometric functions	trig	trig(3M)
language	troff description of output	troff(5)
9700 printer dx9700 prepare	troff documents for the Xerox	dx9700(1)
typesetting viewgraphs mv a	troff macro package for	mv(5)
tc	troff output interpreter	tc(1)
simple pictures pic	troff preprocessor for drawing	pic(1)
typesetting	troff text formatting and	troff(1)
tbl format tables for nroff or	troff	tbl(1)
mathematical text for nroff or	troff eqn format	eqn(1)
files for device-independent	troff font description	font(5)
provide truth values	true	true(1)
provide	truth values true	true(1)
manage binary search trees	tsearch	tsearch(3C)
topological sort	tsort	tsort(1)
	tty general terminal interface	tty(7)
get the name of the terminal	tty	tty(1)
controlling terminal interface	tty	tty(7)
find name of a terminal	ttyname	ttyname(3C)
the utmp file of the current user	ttyslot find the slot in	ttyslot(3C)
types by port	ttytype data base of terminal	ttytype(5)
explicit Fortran	type conversion ftype	ftype(3F)
determine file	type file	file(1)
primitive system data	types types	types(5)
and slides mmt	typeset documents, viewgraphs,	mmt(1)
nroff format or	typeset text	nroff(1)
mv a troff macro package for	typesetting viewgraphs and slides	mv(5)
troff text formatting and	typesetting	troff(1)
get and set user limits	ulimit	ulimit(2)
creation mask	umask set and get file	umask(2s)
set file-creation mode mask	umask	umask(1)
	umount unmount a file system	umount(2s)
unmount a file system	umount	umount(2)
print name of current UNIX system	uname	uname(1)
get name of current UNIX system	uname	uname(2)
file unget	undo a previous get of an SCCS	unget(1)
a previous get of an SCCS file	unget undo	unget(1)
character back into input stream	ungetc push	ungetc(3S)
pseudo-random numbers generate	uniformly distributed	drand48(3C)
report repeated lines in a file	uniq	uniq(1)
make a	unique file name mktemp	mktemp(3C)
conversion program	units	units(1)
exercise link and	unlink system calls link	link(1m)
remove directory entry	unlink	unlink(2)
	unmount a file system umount	umount(2)
umount	unmount a file system	umount(2s)
suspend process	until signal pause	pause(2)
times of a file touch	update access and modification	touch(1)
linear search and	update lsearch	lsearch(3C)
	update super-block sync	sync(2)
	update the super block sync	sync(1)
programs make maintain,	update, and regenerate groups of	make(1)
specify what to do	upon receipt of a signal signal	signal(2)
summarize disk	usage du	du(1)
du summarize disk	usage	du(8)
stat statistical network	useful with graphical commands	stat(1)

generate disk accounting data by	user ID diskusg diskusg	diskusg(1m)
print	user and group IDs and names id	id(1)
set	user and group IDs setuid	setuid(2)
	user crontab file crontab	crontab(1)
get character login name of the	user cuserid	cuserid(3S)
	user environment environ	environ(5)
get and set	user limits ulimit	ulimit(2)
return login name of	user logname	logname(3X)
become super-user or another	user su	su(1)
in the utmp file of the current	user ttyslot find the slot	ttyslot(3C)
getuid get real and effective	user, and real and effective	getuid(2s)
and effective group get real	user, effective user, real group,	getuid(2)
group get real and effective,	user, real and effective	getuid(2)
group get real user, effective	user, real group, and effective	getuid(2)
send mail to	users or read mail mail	mail(1)
write to all	users wall	wall(1m)
editor (variant of ex for casual	users) edit text	edit(1)
fuser identify processes	using a file or file structure	fuser(1m)
statistics	ustat get file system	ustat(2s)
get file system statistics	ustat	ustat(2)
graphical	utilities gutil	gutil(1)
identical mirutil	utility for connecting two	mirutil(1m)
access and modification times	utime set file	utime(2)
utmp	utmp and wtmp entry formats	utmp(4)
access	utmp file entry getut	getut(3C)
ttyslot find the slot in the	utmp file of the current user	ttyslot(3C)
for the uucp system	uucico file transport program	uucico(1m)
uucp spool directory clean-up	uuclean	uuclean(1m)
monitor	uucp network uusub	uusub(1m)
uuclean	uucp spool directory clean-up	uuclean(1m)
control uustat	uucp status inquiry and job	uustat(1)
file transport program for the	uucp system uucico	uucico(1m)
UNIX system to UNIX system copy	uucp	uucp(1)
encode/decode a binary file for/	uuencode,uudecode	uuencode(1c)
status inquiry and job control	uustat uucp	uustat(1)
monitor uucp network	uusub	uusub(1m)
UNIX-to-UNIX system file copy	uuto public	uuto(1)
system command execution	uux UNIX-to-UNIX	uux(1)
requests	uuxqt execute remote command	uuxqt(1m)
validate SCCS file	val	val(1)
	validate SCCS file val	val(1)
return integer absolute	value abs	abs(3C)
Fortran absolute	value abs	abs(3F)
return	value for environment name getenv	getenv(3C)
ceiling, remainder, absolute	value functions floor floor,	floor(3M)
tell report the current	value of a file pointer	tell(2s)
change or add	value to environment putenv	putenv(3C)
provide truth	values true	true(1)
machine-dependent	values values	values(5)
machine-dependent values	values	values(5)
print formatted output of a	varargs argument list vprintf	vprintf(3S)
handle variable argument list	varargs	varargs(5)
handle	variable argument list varargs	varargs(5)
return Fortran environment	variable getenv	getenv(3F)
version control	vc	vc(1)

get option letter from argument	vector getopt	getopt(3C)
	verify program assertion assert	assert(3X)
	version control vc	vc(1)
get a	version of a SCCS file get	get(1)
compare two	versions of an SCCS file sccsdiff	sccsdiff(1)
display editor based on ex	vi screen-oriented (visual)	vi(1)
mv a macro package for making	view graphs	mv(7)
mmt typeset documents,	viewgraphs, and slides	mmt(1)
file systems with label checking	volcopy copy	volcopy(1m)
format of system	volume fs	fs(4)
output of a varargs argument list	vprintf print formatted	vprintf(3S)
terminate wait	wait for child process to stop or	wait(2)
await completion of process	wait	wait(1)
	walk a file tree ftw	ftw(3C)
write to all users	wall	wall(1m)
word count	wc	wc(1)
crashes crash	what to do when the system	crash(8)
	who is doing what whodo	whodo(1m)
	who is on the system who	who(1)
who is doing what	whodo	whodo(1m)
profile	within a function prof	prof(5)
	word count wc	wc(1)
get character or	word from a stream getc	getc(3S)
guess the	word hangman	hangman(6)
put character or	word on a stream putc	putc(3S)
find hyphenated	words hyphen	hyphen(1)
change	working directory cd	cd(1)
change	working directory chdir	chdir(2)
get path-name of current	working directory getcwd	getcwd(3C)
	working directory name pwd	pwd(1)
chdir change	working directory	chdir(2s)
write	write on a file	write(2s)
putpwent	write password file entry	putpwent(3C)
	write to all users wall	wall(1m)
write on a file	write	write(2)
on the other terminal and exits.	write EOT	write(1)
open for reading or	writing open	open(2)
open for reading or	writing	open(2s)
utmp and	wtmp entry formats utmp	utmp(4)
the game of hunt-the-wumpus	wump	wump(6)
for the Xerox 9700 printer	x9700 prepare nroff documents	x9700(1)
list(s) and execute command	xargs construct argument	xargs(1)
yacc	yet another compiler-compiler	yacc(1)

NAME

intro – introduction to system maintenance commands and application programs

DESCRIPTION

This section describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes. The commands in this section should be used along with those listed in Section 1 of the *UNIX System User Reference Manual* and Sections 2, 3, 4, and 5 of the *UNIX System Programmer Reference Manual*. References to other manual entries not of the form *name (1M), name (7)* or *name (8)* refer to entries of the above manuals.

COMMAND SYNTAX

Unless otherwise noted, commands described in this section accept options and other arguments according to the following syntax:

name [*option*(*s*)] [*cmdarg*(*s*)]
where:

name	The name of an executable file.
option	– *noargletter (s)* or, – *argletter* <> *optarg* where <> is optional white space.
noargletter	A single letter representing an option without an argument.
argletter	A single letter representing an option requiring an argument.
optarg	Argument (character string) satisfying preceding *argletter* .
cmdarg	Path name (or other command argument) *not* beginning with – or, – by itself indicating the standard input.

SEE ALSO

getopt(1), getopt(3C).
"UNIX System User Reference Manual" .
"UNIX System Programmer Reference Manual" .
"UNIX System Administrator Guide" .

DIAGNOSTICS

Upon termination, each command returns two bytes of status, one supplied by the system and giving the cause for termination, and (in the case of "normal" termination) one supplied by the program (see *wait (2)* and *exit (2)).* The former byte is 0 for normal termination; the latter is customarily 0 for successful execution and non-zero to indicate troubles such as erroneous parameters, bad or inaccessible data, or other inability to cope with the task at hand. It is called variously "exit code", "exit status", or "return code", and is described only where special conventions are involved.

BUGS

Regretfully, many commands do not adhere to the aforementioned syntax.

NAME

accept, reject – allow/prevent LP requests

SYNOPSIS

/usr/lib/accept destinations
/usr/lib/reject [**–r**[reason]] destinations

DESCRIPTION

Accept allows *lp (1)* to accept requests for the named *destinations* . A *destination* can be either a printer or a class of printers. Use *lpstat (1)* to find the status of *destinations* .

Reject prevents *lp (1)* from accepting requests for the named *destinations* . A *destination* can be either a printer or a class of printers. Use *lpstat (1)* to find the status of *destinations* . The following option is useful with *reject* .

–r [*reason*] Associates a *reason* with preventing *lp* from accepting requests. This *reason* applies to all printers mentioned up to the next **–r** option. *Reason* is reported by *lp* when users direct requests to the named *destinations* and by *lpstat (1).* If the **–r** option is not present or the **–r** option is given without a *reason* , then a default *reason* will be used.

FILES

/usr/spool/lp/*

SEE ALSO

enable(1), lp(1), lpadmin(1M), lpsched(1M), lpstat(1).

NAME

acctdisk, acctdusg, accton, acctwtmp – overview of accounting and miscellaneous accounting commands

SYNOPSIS

/usr/lib/acct/acctdisk

/usr/lib/acct/acctdusg [**-u** file] [**-p** file]

/usr/lib/acct/accton [file]

/usr/lib/acct/acctwtmp "reason"

DESCRIPTION

Accounting software is structured as a set of tools (consisting of both C programs and shell procedures) that can be used to build accounting systems. *Acctsh (1M)* describes the set of shell procedures built on top of the C programs.

Connect time accounting is handled by various programs that write records into **/etc/utmp** , as described in *utmp (4).* The programs described in *acctcon (1M)* convert this file into session and charging records, which are then summarized by *acctmerg (1M).*

Process accounting is performed by the UNIX system kernel. Upon termination of a process, one record per process is written to a file (normally **/usr/adm/pacct).** The programs in *acctprc (1M)* summarize this data for charging purposes; *acctcms (1M)* is used to summarize command usage. Current process data may be examined using *acctcom (1).*

Process accounting and connect time accounting (or any accounting records in the format described in *acct (4))* can be merged and summarized into total accounting records by *acctmerg* (see **tacct** format in *acct (4)). Prtacct* (see *acctsh (1M))* is used to format any or all accounting records.

Acctdisk reads lines that contain user ID, login name, and number of disk blocks and converts them to total accounting records that can be merged with other accounting records.

Acctdusg reads its standard input (usually from **find / -print)** and computes disk resource consumption (including indirect blocks) by login. If **-u** is given, records consisting of those file names for which *acctdusg* charges no one are placed in *file* (a potential source for finding users trying to avoid disk charges). If **-p** is given, *file* is the name of the password file. This option is not needed if the password file is **/etc/passwd .** (See *diskusg*(1M) for more details.)

Accton alone turns process accounting off. If *file* is given, it must be the name of an existing file, to which the kernel appends process accounting records (see *acct (2)* and *acct (4)).*

Acctwtmp writes a *utmp (4)* record to its standard output. The record contains the current time and a string of characters that describe the *reason* . A record type of ACCOUNTING is assigned (see *utmp (4)). Reason* must be a string of 11 or less characters, numbers, **$** , or spaces. For example, the following are suggestions for use in reboot and shutdown procedures, respectively:

```
acctwtmp `uname` >> /etc/wtmp
acctwtmp "file save" >> /etc/wtmp
```

FILES

/etc/passwd	used for login name to user ID conversions
/usr/lib/acct	holds all accounting commands listed in sub-class 1M of this manual
/usr/adm/pacct	current process accounting file
/etc/wtmp	login/logoff history file

SEE ALSO

acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), diskusg(1M), fwtmp(1M), runacct(1M), acctcom(1), acct(2), acct(4), utmp(4).

UNIX Accounting System in the *Sys5 UNIX Administrator Guide* .

NAME

acctcms – command summary from per-process accounting records

SYNOPSIS

/usr/lib/acct/acctcms [options] files

DESCRIPTION

Acctcms reads one or more *files* , normally in the form described in *acct (4).* It adds all records for processes that executed identically-named commands, sorts them, and writes them to the standard output, normally using an internal summary format. The *options* are:

- **–a** Print output in ASCII rather than in the internal summary format. The output includes command name, number of times executed, total kcore-minutes, total CPU minutes, total real minutes, mean size (in K), mean CPU minutes per invocation, "hog factor", characters transferred, and blocks read and written, as in *acctcom (1).* Output is normally sorted by total kcore-minutes.
- **–c** Sort by total CPU time, rather than total kcore-minutes.
- **–j** Combine all commands invoked only once under "***other".
- **–n** Sort by number of command invocations.
- **–s** Any file names encountered hereafter are already in internal summary format.
- **–t** Process all records as total accounting records. The default internal summary format splits each field into prime and non-prime time parts. This option combines the prime and non-prime time parts into a single field that is the total of both, and provides upward compatibility with old (i.e., UNIX System V) style **acctcms** internal summary format records.

The following options may be used only with the **-a** option.

- **–p** Output a prime-time-only command summary.
- **–o** Output a non-prime (offshift) time only command summary.

When **–p** and **–o** are used together, a combination prime and non-prime time report is produced. All the output summaries will be total usage except number of times executed, CPU minutes, and real minutes which will be split into prime and non-prime.

A typical sequence for performing daily command accounting and for maintaining a running total is:

```
acctcms file ... >today
cp total previoustotal
acctcms –s today previoustotal >total
acctcms –a –s today
```

SEE ALSO

acct(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acctcom(1), acct(2), acct(4), utmp(4).

BUGS

Unpredictable output results if **-t** is used on new style internal summary format files, or if it is not used with old style internal summary format files.

NAME

acctcon1, acctcon2 – connect-time accounting

SYNOPSIS

/usr/lib/acct/acctcon1 [options]

/usr/lib/acct/acctcon2

DESCRIPTION

Acctcon1 converts a sequence of login/logoff records read from its standard input to a sequence of records, one per login session. Its input should normally be redirected from **/etc/wtmp** . Its output is ASCII, giving device, user ID, login name, prime connect time (seconds), non-prime connect time (seconds), session starting time (numeric), and starting date and time. The *options* are:

–p Print input only, showing line name, login name, and time (in both numeric and date/time formats).

–t *Acctcon1* maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session still in progress. The **–t** flag causes it to use, instead, the last time found in its input, thus assuring reasonable and repeatable numbers for non-current files.

–l *file* *File* is created to contain a summary of line usage showing line name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logoffs. This file helps track line usage, identify bad lines, and find software and hardware oddities. Hang-up, termination of *login (1)* and termination of the login shell each generate logoff records, so that the number of logoffs is often three to four times the number of sessions. See *init (1M)* and *utmp (4).*

–o *file* *File* is filled with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.

Acctcon2 expects as input a sequence of login session records and converts them into total accounting records (see **tacct** format in *acct (4)).*

EXAMPLES

These commands are typically used as shown below. The file **ctmp** is created only for the use of *acctprc (1M)* commands:

```
acctcon1 -t -l lineuse -o reboots <wtmp | sort +1n +2 >ctmp
acctcon2 <ctmp | acctmerg >ctacct
```

FILES

/etc/wtmp

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), init(1M), login(1), runacct(1M), acct(2), acct(4), utmp(4).

BUGS

The line usage report is confused by date changes. Use *wtmpfix* (see *fwtmp*(1M)) to correct this situation.

NAME

acctmerg – merge or add total accounting files

SYNOPSIS

/usr/lib/acct/acctmerg [options] [file] . . .

DESCRIPTION

Acctmerg reads its standard input and up to nine additional files, al in the **tacct** format (see *acct (4))* or an ASCII version thereof. I merges these inputs by adding records whose keys (normally use ID and name) are identical, and expects the inputs to be sorted or those keys. *Options* are:

- **–a** Produce output in ASCII version of **tacct .**
- **–i** Input files are in ASCII version of **tacct .**
- **–p** Print input with no processing.
- **–t** Produce a single record that totals all input.
- **–u** Summarize by user ID, rather than user ID and name.
- **–v** Produce output in verbose ASCII format, with more precise notation for floating point numbers.

The following sequence is useful for making "repairs" to any file kept in this format:

EXAMPLES

acctmerg –v <file1 >file2
edit file2 as desired . . .
acctmerg –i <file2 >file1

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctcon(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct(4), utmp(4).

NAME

acctprc1, acctprc2 – process accounting

SYNOPSIS

/usr/lib/acct/acctprc1 [ctmp]

/usr/lib/acct/acctprc2

DESCRIPTION

Acctprc1 reads input in the form described by *acct (4),* adds login names corresponding to user IDs, then writes for each process an ASCII line giving user ID, login name, prime CPU time (tics), non-prime CPU time (tics), and mean memory size (in memory segment units). If **ctmp** is given, it is expected to contain a list of login sessions, in the form described in *acctcon (1M),* sorted by user ID and login name. If this file is not supplied, it obtains login names from the password file. The information in **ctmp** helps it distinguish among different login names that share the same user ID.

Acctprc2 reads records in the form written by *acctprc1* , summarizes them by user ID and name, then writes the sorted summaries to the standard output as total accounting records.

These commands are typically used as shown below:

```
acctprc1 ctmp </usr/adm/pacct | acctprc2 >ptacct
```

FILES

/etc/passwd

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctsh(1M), cron(1M), fwtmp(1M), runacct(1M), acct(2), acct(4), utmp(4).

BUGS

Although it is possible to distinguish among login names that share user IDs for commands run normally, it is difficult to do this for those commands run from *cron (1M),* for example. More precise conversion can be done by faking login sessions on the console via the *acctwtmp* program in *acct (1M).*

CAVEAT

A memory segment of the mean memory size is a unit of measure for the number of bytes in a logical memory segment on a particular processor. For example, on a PDP-11/70 this measure would be in 64-byte units, while on a VAX11/780 it would be in 512-byte units.

NAME

chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup, turnacct – shell procedures for accounting

SYNOPSIS

/usr/lib/acct/chargefee login-name number

/usr/lib/acct/ckpacct [blocks]

/usr/lib/acct/dodisk [-o] [files ...]

/usr/lib/acct/lastlogin

/usr/lib/acct/monacct number

/usr/lib/acct/nulladm file

/usr/lib/acct/prctmp

/usr/lib/acct/prdaily [-l] [-c] [mmdd]

/usr/lib/acct/prtacct file ["heading"]

/usr/lib/acct/runacct [mmdd] [mmdd state]

/usr/lib/acct/shutacct ["reason"]

/usr/lib/acct/startup

/usr/lib/acct/turnacct on | off | switch

DESCRIPTION

Chargefee can be invoked to charge a *number* of units to *login-name* . A record is written to **/usr/adm/fee** , to be merged with other accounting records during the night.

Ckpacct should be initiated via *cron*(1M). It periodically checks the size of **/usr/adm/pacct** . If the size exceeds *blocks*, 1000 by default, *turnacct* will be invoked with argument *switch* . If the number of free disk blocks in the **/usr** file system falls below 500, *ckpacct* will automatically turn off the collection of process accounting records via the **off** argument to *turnacct* . When at least this number of blocks is restored, the accounting will be activated again. This feature is sensitive to the frequency at which *ckpacct* is executed, usually by *cron* .

Dodisk should be invoked by *cron* to perform the disk accounting functions. By default, it will do disk accounting on the special files in **/etc/checklist.** If the **–o** flag is used, it will do a slower version of disk accounting by login directory. *Files* specify the one or more filesystem names where disk accounting will be done. If *files* are used, disk accounting will be done on these filesystems only. If the **–o** flag is used, *files* should be mount points of mounted filesystem. If omitted, they should be the special file names of mountable filesystems.

Lastlogin is invoked by *runacct* to update **/usr/adm/acct/sum/loginlog** , which shows the last date on which each person logged in.

Monacct should be invoked once each month or each accounting period. *Number* indicates which month or period it is. If *number* is not given, it defaults to the current month (01–12). This default is useful if *monacct* is to executed via *cron (1M)* on the first day of each month. *Monacct* creates summary files in **/usr/adm/acct/fiscal** and restarts summary files in **/usr/adm/acct/sum** .

Nulladm creates *file* with mode 664 and insures that owner and group are **adm** . It is called by various accounting shell procedures.

Prctmp can be used to print the session record file (normally **/usr/adm/acct/nite/ctmp** created by *acctcon1* (see *acctcon (1M)).*

Prdaily is invoked by *runacct* to format a report of the previous day's accounting data. The report resides in **/usr/adm/acct/sum/rprt***mmdd* where *mmdd* is the month and day of the report. The current daily accounting reports may be printed by typing *prdaily .* Previous days' accounting reports can be printed by using the *mmdd* option and specifying the exact report date desired. The **–l** flag prints a report of exceptional usage by login id for the specifed date. Previous daily reports are cleaned up and therefore inaccessible after each invocation of *monacct* . The **–c** flag prints a report of exceptional resource usage by command, and may be used on current day's accounting data only.

Prtacct can be used to format and print any total accounting (**tacct**) file.

Runacct performs the accumulation of connect, process, fee, and disk accounting on a daily basis. It also creates summaries of command usage. For more information, see *runacct*(1M).

Shutacct should be invoked during a system shutdown (usually in **/etc/shutdown)** to turn process accounting off and append a "reason" record to **/etc/wtmp** .

Startup should be called by **/etc/rc** to turn the accounting on whenever the system is brought up.

Turnacct is an interface to *accton* (see *acct (1M))* to turn process accounting **on** or **off**. The **switch** argument turns accounting off, moves the current **/usr/adm/pacct** to the next free name in **/usr/adm/pacct***incr* (where *incr* is a number starting with **1** and incrementing by one for each additional **pacct** file), then turns accounting back on again. This procedure is called by *ckpacct* and thus can be taken care of by the *cron* and used to keep **pacct** to a reasonable size.

FILES

/usr/adm/fee	accumulator for fees
/usr/adm/pacct	current file for per-process accounting
/usr/adm/pacct*	used if pacct gets large and during execution of daily accounting procedure
/etc/wtmp	login/logoff summary
/usr/lib/acct/ptelus.awk	contains the limits for exceptional usage by login id
/usr/lib/acct/ptecms.awk	contains the limits for exceptional usage by command name
/usr/adm/acct/nite	working directory
/usr/lib/acct	holds all accounting commands listed in sub-class 1M of this manual
/usr/adm/acct/sum	summary directory, should be saved

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctprc(1M), cron(1M), diskusg(1M), fwtmp(1M), runacct(1M), acct(2), acct(4), utmp(4).

NAME

acpdmp – dump contents of Advanced Communication Processor's memory into a file

SYNOPSIS

/etc/acpdmp /dev/ac*X* *file*
where:
X = 0, 1, 2, 3, or 4

DESCRIPTION

Acpdmp reads the contents of an Advanced Communication Processor's (ACP) memory and dumps it to the specified file. The ACP must be in the reset mode. That is, it has been reset with a system reset for *acpdmp* to work properly. This command can only be executed in single-user state before the ACP is downloaded with a new kernel. The ACP device names are /dev/ac0, /dev/ac1, /dev/ac2, /dev/ac3, /dev/ac4.

If there are ACPs and Intelligent Communication Processor's (ICPs) in a system, the ICPs are the lowest numbered devices. ACP device numbers are one greater than the last ICP device number.

EXAMPLE 1: 3 ICPs and 1 ACP

icp0
icp1
icp2
acp3

EXAMPLE 2: 0 ICPs and 3 ACPs

acp0
acp1
acp2

FILES

dnld(1M)

NOTES

This is a Plexus command. It is not part of standard System V.
The total number of ACPs and ICPs in a system must not exceed 5.

BUGS

Ensure that there is enough disk space before running this program.
It requires approximately 516K.

NAME

brc, bcheckrc, rc, powerfail – system initialization shell scripts

SYNOPSIS

/etc/brc

/etc/bcheckrc

/etc/rc

/etc/powerfail

DESCRIPTION

Except for *powerfail* , these shell procedures are executed via entries in */etc/inittab* by *init (1M)* when the system is changed out of *SINGLE USER* mode. *Powerfail* is executed whenever a system power failure is detected.

The *brc* procedure clears the mounted file system table, **/etc/mnttab** (see *mnttab (4)),* and loads any programmable micro-processors with their appropriate scripts.

The *bcheckrc* procedure performs all the necessary consistency checks to prepare the system to change into multi-user mode. It will prompt to set the system date and to check the file systems with *fsck (1M).*

The *rc* procedure starts all system daemons before the terminal lines are enabled for multi-user mode. In addition, file systems are mounted and accounting, error logging, system activity logging and the Remote Job Entry (RJE) system are activated in this procedure.

The *powerfail* procedure is invoked when the system detects a power failure condition. Its chief duty is to reload any programmable micro-processors with their appropriate scripts, if suitable. It also logs the fact that a power failure occurred.

SEE ALSO

fsck(1M), init(1M), shutdown(1M), inittab(4), mnttab(4).

NAME

checkall – faster file system checking procedure

SYNOPSIS

/etc/checkall

DESCRIPTION

The *checkall* procedure is a prototype and must be modified to suit local conditions. The following will serve as an example:

```
# check the root file system by itself
fsck /dev/dsk/0s0

# dual fsck of drives 0 and 1
dfsck /dev/rdsk/0s[12345] – /dev/rdsk/1s1
```

In the above example (where **/dev/rdsk/1s1** is 320K blocks and **/dev/rdsk/0s[12345]** are each 65K or less), a previous sequential *fsck* took 19 minutes. The *checkall* procedure takes 11 minutes.

Dfsck is a program that permits an operator to interact with two *fsck (1M)* programs at once. To aid in this, *dfsck* will print the file system name for each message to the operator. When answering a question from *dfsck* , the operator must prefix the response with a **1** or a **2** (indicating that the answer refers to the first or second file system group).

Due to the file system load balancing required for dual checking, the *dfsck (1M)* command should always be executed through the *checkall* shell procedure.

In a practical sense, the file systems are divided as follows:

```
dfsck file_systems_on_drive_0 – file_systems_on_drive_1
dfsck file_systems_on_drive_2 – file_systems_on_drive_3
  . . .
```

A three-drive system can be handled by this more concrete example (assumes two large file systems per drive):

```
dfsck /dev/dsk/3s1 /dev/dsk/0s[14] – /dev/dsk/1s[14]
/dev/dsk/3s4
```

Note that the first file system on drive 3 is first in the *filesystems1* list and is last in the *filesystems2* list assuring that references to that drive will not overlap at execution time.

WARNINGS

1. Do not use *dfsck* to check the *root* file system.
2. On a check that requires a scratch file (see **–t** above), be careful not to use the same temporary file for the two groups (this is sure to scramble the file systems).

3. The *dfsck* procedure is useful only if the system is set up for multiple physical I/O buffers.

SEE ALSO

fsck(1M).

Setting Up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide* .

NAME

chroot – change root directory for a command

SYNOPSIS

/etc/chroot newroot command

DESCRIPTION

The given command is executed *relative to the new root* . The meaning of any initial slashes (/) in path names is changed for a command and any of its children to *newroot* . Furthermore, the initial working directory is *newroot* .

Notice that:

chroot newroot command >x

will create the file **x** relative to the original root, not the new one.

This command is restricted to the super-user.

The new root path name is always relative to the current root: even if a *chroot* is currently in effect, the *newroot* argument is relative to the current root of the running process.

SEE ALSO

chdir(2).

BUGS

One should exercise extreme caution when referencing special files in the new root file system.

NAME

clri – clear i-node

SYNOPSIS

/etc/clri file-system i-number ...

DESCRIPTION

Clri writes zeros on the 64 bytes occupied by the i-node numbered *i-number* . *File-system* must be a special file name referring to a device containing a file system. After *clri* is executed, any blocks in the affected file will show up as "missing" in an *fsck (1M)* of the *file-system* . This command should only be used in emergencies and extreme care should be exercised.

Read and write permission is required on the specified *file-system* device. The i-node becomes allocatable.

The primary purpose of this routine is to remove a file which for some reason appears in no directory. If it is used to *zap* an i-node which does appear in a directory, care should be taken to track down the entry and remove it. Otherwise, when the i-node is reallocated to some new file, the old entry will still point to that file. At that point removing the old entry will destroy the new file. The new entry will again point to an unallocated i-node, so the whole cycle is likely to be repeated again and again.

SEE ALSO

fsck(1M), fsdb(1M), ncheck(1M), fs(4).

BUGS

If the file is open, *clri* is likely to be ineffective.

NAME

copytape – make an image copy of a tape

SYNOPSIS

/usr/plx/copytape [**–rwv**] [**–p numfiles**] [**–f filenum**] [**–d descfile**] [**–i**] srcfile [**–o**] dstfile

DESCRIPTION

Copytape is used for duplicating tapes. It preserves blocking and file marks. The *–r* option specifies that *srcfile* (presumably a tape) is to be read and its data placed on *dstfile*. If not otherwise specified, standard output contains the blocking and file mark information. The *–w* option (default) specifies that *srcfile* is to be read and *dstfile* (presumably a tape) is to be written according to information given as standard input.

The *–v* option (used with the *–r* option) specifies that variable size blocks may occur within a tape file.

The *–p* option must be used for the streaming tape drive, and the number of files to be read must be specified. A raw disk file system (e.g, **/dev/rdk3**), as opposed to a file, MUST be used when the streaming tape drive *–p* option is specified. On Plexus systems, *–p* works only if the system has installed an IMSP board. As a result, this option will not work on a P/15 or P/20.

The *–f* option specifies that a single file is to be read from or written to tape. The *filenum* selects the file from the *srcfile*, starting with file number 0.

–i signals the input file, while *–o* means the output file.

EXAMPLES

The command

```
copytape -r /dev/rmt0 tapeimage > descfile
```

makes an image of the tape in drive 0 in the file *tapeimage* while creating a description file called *descfile.* By loading a new tape and issuing the command

```
copytape -w tapeimage /dev/nrmt0 < descfile
```

an exact image of the tape will be created. Notice that **/dev/nrmt0** is used instead of **/dev/rmt0**. This is required so that the tape will not rewind between files. Also notice that *tapefile* may be very large, and that there must be enough room in the file system to hold it before this will work. It is also possible to use logical disk drives (e.g., **/dev/dk5**), but this can be extremely dangerous if used incorrectly. Note that a cartridge tape will operate in streaming mode only if a raw logical disk is specified.

NOTES

This command is a Plexus feature; it is not part of standard Sys5.

BUGS

The *–v* option doesn't work for streaming cartridge tape drives.
The *–p* option doesn't work for Plexus P/15 or P/20 systems.

NAME

cpset – install object files in binary directories

SYNOPSIS

cpset [-o] object directory [mode owner group]

DESCRIPTION

Cpset is used to install the specified *object* file in the given *directory* . The *mode* , *owner* , and *group* , of the destination file may be specified on the command line. If this data is omitted, two results are possible:

If the user of *cpset* has administrative permissions (that is, the user's numerical ID is less than 100), the following defaults are provided:

mode – 0755

owner – bin

group – bin

If the user is not an administrator, the default, owner, and group of the destination file will be that of the invoker.

An optional argument of **–o** will force *cpset* to move *object* to **OLD***object* in the destination directory before installing the new object.

For example:

cpset echo /bin 0755 bin bin

cpset echo /bin

cpset echo /bin/echo

All the examples above have the same effect (assuming the user is an administrator). The file **echo** will be copied into **/bin** and will be given **0755, bin, bin** as the mode, owner, and group, respectively.

Cpset utilizes the file **/usr/src/destinations** to determine the final destination of a file. The locations file contains pairs of pathnames separated by spaces or tabs. The first name is the "official" destination (for example: **/bin/echo).** The second name is the new destination. For example, if *echo* is moved from **/bin** to **/usr/bin** , the entry in **/usr/src/destinations** would be:

/bin/echo /usr/bin/echo

When the actual installation happens, *cpset* verifies that the "old" pathname does not exist. If a file exists at that location, *cpset* issues a warning and continues. This file does not exist on a distribution tape; it is used by sites to track local command movement. The procedures used to build the source will be responsible for defining the "official" locations of the source.

Cross Generation

The environment variable **ROOT** will be used to locate the destination file (in the form **$ROOT/usr/src/destinations**). This is necessary in the cases where cross generation is being done on a production system.

SEE ALSO

install(1M), make(1), mk(8).

NAME

crash – examine system images

SYNOPSIS

/**etc**/**crash** [system] [namelist]

DESCRIPTION

Crash is an interactive utility for examining an operating system core image. It has facilities for interpreting and formatting the various control structures in the system and certain miscellaneous functions that are useful when perusing a dump.

The arguments to *crash* are the file name where the *system* image can be found and a *namelist* file to be used for symbol values.

The default values are /**dev**/**mem** and /**unix** ; hence, *crash* with no arguments can be used to examine an active system. If a *system* image file is given, it is assumed to be a system core dump and the default process is set to be that of the process running at the time of the crash. This is determined by a value stored in a fixed location by the dump mechanism.

COMMANDS

Input to *crash* is typically of the form:

command [*options*] [structures to be printed].

When allowed, *options* will modify the format of the printout. If no specific structure elements are specified, all valid entries will be used. As an example, **proc – 12 15 3** would print process table slots 12, 15, and 3 in a long format, while **proc** would print the entire process table in standard format.

In general, those commands that perform I/O with addresses assume hexadecimal on 32-bit machines and octal on 16-bit machines.

The current repertory consists of:

user [list of process table entries]

Aliases: **uarea , u_area , u .**

Print the user structure of the named process as determined by the information contained in the process table entry. If no entry number is given, the information from the last executing process will be printed. Swapped processes produce an error message.

trace [–r] [list of process table entries]

Aliases: **t .**

Generate a kernel stack trace of the current process. If the **–r** option is used, the trace begins at the saved stack frame pointer in **kfp .** Otherwise the trace starts at the bottom of the stack and attempts to find valid stack frames deeper in the stack. If no entry number is given, the information from the last executing process will be printed.

kfp [stack frame pointer]

Aliases: **r5** , **fp** .

Print the program's idea of the start of the current stack frame (set initially from a fixed location in the dump) if no argument is given, or set the frame pointer to the supplied value.

stack [list of process table entries]

Aliases: **stk** , **s** , **kernel** , **k** .

Format a dump of the kernel stack of a process. The addresses shown are virtual system data addresses rather than true physical locations. If no entry number is given, the information from the last executing process will be printed.

proc [–[r]] [list of process table entries]

Aliases: **ps** , **p** .

Format the process table. The **–r** option causes only runnable processes to be printed. The – alone generates a longer listing.

pcb [list of process table entries]

Print the process control block of the current process. The process control block is a part of the user area (VAX-11/780 only). If no entry number is given, the information from the last executing process will be printed.

i-node [–] [list of i-node table entries]

Aliases: **ino** , **i** .

Format the i-node table. The – option will also print the i-node data block addresses.

file [list of file table entries]

Aliases: **files** , **f** .

Format the file table.

mount [list of mount table entries]

Aliases: **mnt** , **m** .

Format the mount table.

text [list of text table entries]

Aliases: **txt** , **x** .

Format the text table.

tty [type] [–] [list of tty entries]

Aliases: **term** (also **dz** and **dh** are aliases on DEC

machines).
Print the tty structures. The *type* argument determines which structure will be used (such as **kl** , **dh** , **dz** , or **dzb** on DEC equipment; **tn83** , **tn74** , or **tn4** on the 3B 20S computers). No default *type* is provided. However, once specified, the last *type* is remembered. The – option prints the *stty (1)* parameters for the given line.

stat Print certain statistics found in the dump. These include the panic string (if a panic occurred), time of crash, system name, and the registers saved in low memory by the dump mechanism.

var Aliases: **tunables , tunable , tune , v .**
Print the tunable system parameters.

buf [list of buffer headers]
Aliases: **hdr , bufhdr .**
Format the system buffer headers.

buffer [format] [list of buffers]
Alias: **b .**
Print the data in a system buffer according to *format* . If *format* is omitted, the previous *format* is used. Valid formats include **decimal** , **octal** , **hex** , **character** , **byte** , **directory** , **i-node** , and **write** . The last creates a file in the current directory (see *FILES*) containing the buffer data.

callout Aliases: **calls , call , c , timeout , time , tout .**
Print all entries in the callout table.

map [list of map names]
Format the named system map structures.

nm [list of symbols]
Print symbol value and type as found in the *namelist* file.

ts [list of text addresses]
Find the closest text symbols to the given addresses.

ds [list of data addresses]
Find the closest data symbols to the given addresses.

od [symbol name or address] [count] [format]
Aliases: **dump , rd .**
Dump *count* data values starting at the symbol value or address given according to *format* . Allowable formats are **octal** , **longoct** , **decimal** , **longdec** , **character** , **hex** , or **byte .**

!	Escape to shell.
q	Exit from *crash* .
?	Print synopsis of commands.

ALIASES

There are built-in aliases for many of the *formats* as well as those listed for the commands. Some of them are:

byte	b.
character	char, c.
decimal	dec, e.
directory	direct, dir, d.
hexadecimal	hexadec, hex, h, x.
i-node	ino , i.
longdec	ld, D.
longoct	lo, O.
octal	oct, o.
write	w.

FILES

/usr/include/sys/*.h	header files for table and structure info
/dev/mem	default system image file
/unix	default namelist file
buf.#	files created containing buffer data

SEE ALSO

mount(1M), nm(1), ps(1), sh(1), stty(1), crash(8).

BUGS

Most flags are abbreviated and will have little meaning to the uninitiated user. A source listing of the system header files at hand would be most useful while using *crash.*

Stack tracing of the current process on a running system does not work.

NAME

cron - clock daemon

SYNOPSIS

/etc/cron

DESCRIPTION

Cron executes commands at specified dates and times. Regularly scheduled commands can be specified according to instructions found in crontab files; users can submit their own crontab file via the *crontab* command. Commands which are to be executed only once may be submitted via the *at* command. Since *cron* never exits, it should only be executed once. This is best done by running *cron* from the initialization process through the file **/etc/rc** (see *init*(1M)).

Cron only examines crontab files and at command files during process initialization and when a file changes. This reduces the overhead of checking for new or changed files at regularly scheduled intervals.

FILES

/usr/lib/cron	main cron directory
/usr/lib/cron/log	accounting information
/usr/spool/cron	spool area

SEE ALSO

at(1), crontab(1), sh(1), init(1M).

DIAGNOSTICS

A history of all actions taken by cron are recorded in **/usr/lib/cron/log.**

NAME

dconfig – configure logical disks

SYNOPSIS

/etc/dconfig - for use under UNIX

dconfig - for running program from release tape only

/stand/dconfig - for standalone use (UNIX not running) only

DESCRIPTION

Dconfig allows you to change the Sys5 default logical disk address assignments and the default UNIX device mapping. It also can be used to verify the logical disk configuration, change the system nodename for **uucp** and **uname**, or change the primary bootname.

Dconfig has both regular (**/etc/dconfig**) and standalone (**/stand/dconfig**) versions. Plexus release tapes also contain a copy of **dconfig**. The arguments to **/etc/dconfig** (the regular version) differ from those for the standalone and tape versions. **/etc/dconfig** expects the special files defined in the **/dev** directory as arguments, while the standalone version and the release tape version both use built-in special filenames as described in the user's manual for your system.

Dconfig prompts for responses, and gives the current values for each parameter in brackets. A <return> leaves the values the same; a <return> in response to a yes or no question defaults to "no". Unlike most Sys5 programs, **dconfig** expects response in terms of 512-byte sectors, rather than 1024 byte blocks.

If **dconfig** for any reason (e.g., permissions) cannot access the disk you type, it continues to give the "Disk?" prompt. For more complete information and examples, see the chapter on standalone programs in your user's manual.

NOTES

This is a Plexus command. It is not part of standard System V.

Dconfig should not be run on disks containing a raw file system which starts at block 0 of the physical disk, as it will ruin the data in the raw file system.

Dconfig cannot use the first two blocks on a disk in a file system other than the first logical one. That is, if you have two disks, the file system size declarations for */dev/dsk/0s0* and */dev/dsk/0s1* must start at sector 0; 0s2-0s15 must not use sectors 0 and 1. On the second disk, the file system size declarations for */dev/dsk/1s0 (/dev/dsk/0s16)* and */dev/dsk/1s1 (/dev/dsk/0s17)* must start at sector 0; 1s2-1s15 (0s18-0s31) must not use sectors 0 and 1.

/etc/dconfig should be used only to examine, and not change, data.

SEE ALSO

uname(1).

NAME

dcopy – copy file systems for optimal access time

SYNOPSIS

/etc/dcopy [–**s**X] [–**a**n] [–**d**] [–**v**] [–**f**fsize[:isize]] inputfs outputfs

DESCRIPTION

Dcopy copies file system *inputfs* to *outputfs* . *Inputfs* is the existing file system; *outputfs* is an appropriately sized file system, to hold the reorganized result. For best results *inputfs* should be the raw device and *outputfs* should be the block device. *Dcopy* should be run on unmounted file systems (in the case of the root file system, copy to a new pack). With no arguments, *dcopy* copies files from *inputfs* compressing directories by removing vacant entries, and spacing consecutive blocks in a file by the optimal rotational gap. The possible options are

–**s***X* supply device information for creating an optimal organization of blocks in a file. The forms of *X* are the same as the **–s** option of *fsck (1M).*

–**a***n* place the files not accessed in *n* days after the free blocks of the destination file system (default for *n* is 7). If no *n* is specified then no movement occurs.

–d leave order of directory entries as is (default is to move sub-directories to the beginning of directories).

–v currently reports how many files were processed, and how big the source and destination freelists are.

–**f***fsize*[:*isize*]
specify the *outputfs* file system and inode list sizes (in blocks). If the option (or *:isize*) is not given, the values from the *inputfs* are used.

Dcopy catches interrupts and quits and reports on its progress. To terminate *dcopy* send a quit signal, and *dcopy* will no longer catch interrupts or quits.

SEE ALSO

fsck(1M), mkfs(1M), ps(1).

NAME

devnm – device name

SYNOPSIS

/etc/devnm [names]

DESCRIPTION

Devnm identifies the special file associated with the mounted file system where the argument *name* resides. (As a special case, both the block device name and the swap device name are printed for the argument name / if swapping is done on the same disk section as the **root** file system.) Argument names must be full path names.

This command is most commonly used by **/etc/rc** (see *brc (1M))* to construct a mount table entry for the **root** device.

EXAMPLE

The command:

    /etc/devnm /usr

produces

    dsk/0s1 /usr

if **/usr** is mounted on **/dev/dsk/0s1 .**

FILES

/dev/dsk/*
/etc/mnttab

SEE ALSO

brc(1M), setmnt(1M).

NAME

df – report number of free disk blocks

SYNOPSIS

df [**–t**] [**–f**] [file-systems]

DESCRIPTION

Df prints out the number of free blocks and free i-nodes available for on-line file systems by examining the counts kept in the super-blocks; *file-systems* may be specified either by device name (e.g., **/dev/dsk/0s1**) or by mounted directory name (e.g., **/usr**). If the *file-systems* argument is unspecified, the free space on all of the mounted file systems is printed.

The **–t** flag causes the total allocated block figures to be reported as well.

If the **–f** flag is given, only an actual count of the blocks in the free list is made (free i-nodes are not reported). With this option, *df* will report on raw devices.

FILES

/dev/dsk/*
/etc/mnttab

SEE ALSO

fs(4), mnttab(4).

NAME

diskusg - generate disk accounting data by user ID

SYNOPSIS

diskusg [options] [files]

DESCRIPTION

Diskusg generates intermediate disk accounting information from data in *files,* or the standard input if omitted. *Diskusg* output lines on the standard output, one per user, in the following format:

uid login #blocks

where

uid -	the numerical user ID of the user.
login -	the login name of the user; and
#blocks -	the total number of disk blocks allocated to this user.

Diskusg normally reads only the i-nodes of file systems for disk accounting. In this case, *files* are the special filenames of these devices.

Diskusg recognizes the following options:

-s the input data is already in *diskusg* output format. *Diskusg* combines all lines for a single user into a single line.

-v verbose. Print a list on standard error of all files that are charged to no one.

-i *fnmlist* ignore the data on those file systems whose file system name is in *fnmlist. Fnmlist* is a list of file system names separated by commas or enclose within quotes. *Diskusg* compares each name in this list with the file system name stored in the volume ID (see *labelit(1M)).*

-p *file* use *file* as the name of the password file to generate login names. /**etc**/**passwd** is used by default.

-u *file* write records to *file* of files that are charged to no one. Records consist of the special file name, the i-node number, and the user ID.

The output of *diskusg* is normally the input to *acctdisk* (see *acct (1M))* which generates total accounting records that can be merged with other accounting records. *Diskusg* is normally run in *dodisk* (see *acctsh (1M)).*

EXAMPLES

The following will generate daily disk accounting information:

```
for i in /dev/rp00 /dev/rp01 /dev/rp10 /dev/rp11; do
        diskusg $i > dtmp.`basename $i` &
done
wait
diskusg -s dtmp.* | sort +0n +1 | acctdisk > disktacct
```

FILES

/etc/passwd   used for user ID to login name conversions

SEE ALSO

acct(1M), acctsh(1M), acct(4)

NAME

dnld – download program files

SYNOPSIS

/etc/dnld [options]

DESCRIPTION

This program transfers program files from the UNIX system to either the EH 4A/BPS4 prom programmer or a DATA I/O PROM programmer or a Plexus system that is running a debugging program. The program options are as follows, where *xxxx* is a hex number:

–a *xxxx* Sets *xxxx* as the base address for text relocation. This address is also sent to the Plexus monitor if the program is in that mode.

–b *xxxx* Sets *xxxx* as the base address for bss relocation. This address is also sent to the Plexus monitor if the program is in that mode.

–i Initializes the EH-4A PROM programmer, does the *dnld*, and programs the PROM.

–c Puts a checksum (so that the words will sum to 0) at location 0x0ffe. Used for making PROMs so that they can be checked for integrity.

–t *info* If the output file is a tty then *info* is used to set up the terminals options. This is done by first opening the terminal and then issuing an *stty* command to it with *info* as the parameters.

–o *outf* Sets the output file name to *outf*.

–da Ignored if **-da** is specified.

–f *inf* Sets the input file name to *inf*.

–k *promsize*
Determines the size of the proms being programmed.

–l Causes the low byte of each instruction in *inf* to be output to *outf*. Used only for prom programming.

–h Causes the high byte of each instruction in *inf* to be output to *outf*. Used only for prom programming.

–p Sets the program to output data in the format used by the EH prom programmer.

–z Sets the program to output data in the format used by the Plexus monitor.

–s *xxxx* Sets the segment number sent to the Plexus monitor.

–u Used for downloading UNIX thru the boot program,

–v Used for the 2732As.

–y *xxxx* Sets the communications address for loading the SIOC.

–d*x* Download IMSP, remote kernel, ACP, or ICP where *x* is:

d download IMSP

r download P15/P20 remote kernel

a*nm* download ACP

n starting ACP number [0-4]
m number of ACPs such that n + m <= 5

s download ICP

no option (blank) default: download ICP

–B Used for 4B/BPS4 PROM programmer.

–D Used for the DATA I/O 29A programmer.

–F *xxyy* Used only for the DATA I/O programmer and must be present if the -D switch is. *xx* is the family and *yy* is the pinout code (e.g. 1924 for 2732DC).

–L Object file header contains LONGs as in 68000 type object files.

The default options are:

-a 0000
-b 0000
-t 1200
-o /dev/promio
-f a.out
-l
-p
-s 0000
-y f800

FILES

/dev/promio

NOTES

This is a Plexus command; it is not part of standard System V.

SEE ALSO

icpdmp(1M)

BUGS

Some of the options may not work for programming PROMs.

NAME

dump – incremental file system dump

SYNOPSIS

/etc/dump [*key* [*arguments*] *file-system*]

DESCRIPTION

Dump copies to magnetic tape all files changed after a certain date in the *file-system*. The *key* specifies the date and other options about the dump.

Key consists of characters from the set **0123456789fusd**.

f Place the dump on the next *argument* file instead of the tape.

u If the dump completes successfully, write the date of the beginning of the dump on file **/etc/ddate**. This file records a separate date for each file system and each dump level.

0–9 This number is the "dump level". All files modified since the last date stored in the file **/etc/ddate** for the same file system at lesser levels will be dumped. If no date is determined by the level, the beginning of time is assumed; thus the option **0** causes the entire file system to be dumped. The default level is 9.

s The size of the dump tape is specified in feet. The number of feet is taken from the *argument* after the tape device name.

EXAMPLE: **/etc/dump/ fsd /dev/rpt/0m 2000 /dev/rdsk/0s***X*

where: *X* is 0 - 15

When using cartridge tapes, for a 20Mb taype, use 1000 for a 45Mb cartridge use 2000, and for 60Mb cartridges use 2500. When using reel-to-reel tapes, use the tape length in feet minus 100 (thus, for a 2400 foot tape, 2300 is the correct size). You should subtract 100 to allow for bad spots on the tape.

When the specified size is reached, the dump will wait for reels or cartridges to be changed. The default size is 2,300 feet.

d The density of the tape, expressed in BPI, is taken from the next *argument*. This is used in calculating the amount of tape used per write. The default is 1600.

If no arguments are given, the *key* is assumed to be **9u** and a default file system is dumped to the default tape.

Performing Dumps

Now a short suggestion on how to perform dumps. Start with a full level-0 dump: **/etc/dump 0u**. Next, periodic level-9 dumps should be made on an exponential progression of tapes. (Sometimes called Tower of Hanoi: 1, 2, 1, 3, 1, 2, 1, 4, ...; tape 1 used every other time, tape 2 is used every fourth, tape 3 is used every eighth, etc.): **/etc/dump 9u**. When the level-9 incremental approaches a full tape (about 78,000 blocks at 1600 BPI blocked 10 1024-byte blocks per record), a level-1 dump should be made: **/etc/dump 1u**. After this, the exponential series should progress as if uninterrupted. These level-9 dumps are based on the level-1 dump, which is based on the level-0 full dump. This progression of levels of dumps can be carried as far as desired.

FILES

/etc/ddate: record dump dates of file system/level.

/dev/rpt/0m	(cartrige tape - rewind)
/dev/rpt/0mn	(cartridge tape - no rewind)
/dev/rrm/0m	(9-track tape - rewind)
/dev/rrm/0mn	(9-track tape - no rewind)

The default file system varies with installation.

SEE ALSO

cpio(1), dumpdir(1M), restor(1M), volcopy(1M), dump(5).

DIAGNOSTICS

If the dump requires more than one tape, it will ask you to change tapes. Reply with a new-line after this has been done.

WARNING

Sizes are based on 1600 BPI blocked tape. The raw magnetic tape device has to be used to approach these densities. Read errors on the file system are ignored. Write errors on the magnetic tape are usually fatal.

Dump does not use true end-of-tape, but calculates the available storage from the size option. If this size is too large, dump will fail with a write error.

NAME

dumpdir – print the names of files on a dump tape

SYNOPSIS

dumpdir [**f** filename]

DESCRIPTION

Dumpdir is used to read magtapes dumped with the *dump* command and list the names and inode numbers of all the files and directories on the tape.

The **f** option makes *filename* the name of the tape instead of the default.

FILES

Default tape unit varies with installation.
rst*

SEE ALSO

dump(1M), restor(1M)

DIAGNOSTICS

If the dump extends over more than one tape, it may ask you to change tapes. Reply with a new-line when the next tape has been mounted.

BUGS

There is redundant information on the tape that could be used in case of tape reading problems. Unfortunately, *dumpdir* doesn't use it.

Dumpdir cannot report correctly on a file having a very long directory path (greater than 15 directories).

NAME

errdead – extract error records from dump

SYNOPSIS

/etc/errdead dumpfile [namelist]

DESCRIPTION

When hardware errors are detected by the system, an error record that contains information pertinent to the error is generated. If the error-logging daemon *errdemon (1M)* is not active or if the system crashes before the record can be placed in the error file, the error information is held by the system in a local buffer. *Errdead* examines a system dump (or memory), extracts such error records, and passes them to *errpt (1M)* for analysis.

The *dumpfile* specifies the file (or memory) that is to be examined. The system namelist is specified by *namelist ;* if not given, /**unix** is used.

FILES

/unix	system namelist
/usr/bin/errpt	analysis program
/usr/tmp/errXXXXXX	temporary file

DIAGNOSTICS

Diagnostics may come from either *errdead* or *errpt .* In either case, they are intended to be self-explanatory.

SEE ALSO

errdemon(1M), errpt(1M).

NAME

errdemon – error-logging daemon

SYNOPSIS

/usr/lib/errdemon [file]

DESCRIPTION

The error logging daemon *errdemon* collects error records from the operating system by reading the special file **/dev/error** and places them in *file* . If *file* is not specified when the daemon is activated, **/usr/adm/errfile** is used. Note that *file* is created if it does not exist; otherwise, error records are appended to it, so that no previous error data is lost. No analysis of the error records is done by *errdemon* ; that responsibility is left to *errpt (1M).* The error-logging daemon is terminated by sending it a software kill signal (see *kill (1)).* Only the super-user may start the daemon, and only one daemon may be active at any time.

FILES

/dev/error	source of error records
/usr/adm/errfile	repository for error records

DIAGNOSTICS

The diagnostics produced by *errdemon* are intended to be self-explanatory.

SEE ALSO

errpt(1M), errstop(1M), kill(1), err(7).

NAME

errpt – process a report of logged errors

SYNOPSIS

errpt [options] [files]

DESCRIPTION

Errpt processes data collected by the error logging mechanism (*errdemon*(1M)) and generates a report of that data. The default report is a summary of all errors posted in the files named. Options apply to all files and are described below. If no files are specified, *errpt* attempts to use **/usr/adm/errfile** as *file* .

A summary report notes the options that may limit its completeness, records the time stamped on the earliest and latest errors encountered, and gives the total number of errors of one or more types. Each device summary contains the total number of unrecovered errors, recovered errors, errors unabled to be logged, I/O operations on the device, and miscellaneous activities that occurred on the device. The number of times that *errpt* has difficulty reading input data is included as read errors.

Any detailed report contains, in addition to specific error information, all instances of the error logging process being started and stopped, and any time changes (via *date (1)*) that took place during the interval being processed. A summary of each error type included in the report is appended to a detailed report.

A report may be limited to certain records in the following ways:

–s *date* Ignore all records posted earlier than *date*, where *date* has the form *mmddhhmmyy*, consistent in meaning with the *date (1)* command.

–e *date* Ignore all records posted later than *date* , whose form is as described above.

–a Produce a detailed report that includes all error types.

–d *devlist* A detailed report is limited to data about devices given in *devlist* , where *devlist* can be one of two forms: a list of device identifiers separated from one another by a comma, or a list of device identifiers enclosed in double quotes and separated from one another by a comma and/or more spaces. *Errpt* is familiar with the common form of identifiers. Additional identifiers are **int** and **mem** which include detailed reports of stray-interrupt and memory-parity type errors, respectively.

−p *n* Limit the size of a detailed report to *n* pages.

−f In a detailed report, limit the reporting of bloc device errors to unrecovered errors.

FILES

/usr/adm/errfile default error file

SEE ALSO

date(1), errdead(1M), errdemon(1M), errfile(4).

NOTE

errpt(1m) is not implemented in Plexus Version 1.4 of UNIX Sys5. It is scheduled to be implemented in Plexus Version 1.5 of UNIX Sys5.

NAME

errstop – terminate the error-logging daemon

SYNOPSIS

/etc/errstop [namelist]

DESCRIPTION

The error-logging daemon *errdemon (1M)* is terminated by using *errstop* . This is accomplished by executing *ps (1)* to determine the daemon's identity and then sending it a software kill signal (see *signal (2));* **/unix** is used as the system namelist if none is specified. Only the super-user may use *errstop* .

FILES

/unix default system namelist

DIAGNOSTICS

The diagnostics produced by *errstop* are intended to be self-explanatory.

SEE ALSO

errdemon(1M), ps(1), kill(2), signal(2).

NAME

fbackup - make a fast tape backup of a file system

SYNOPSIS

fbackup - for running the program from a release tape only

/stand/fbackup - for standalone (no UNIX) use only

DESCRIPTION

The standalone program **fbackup** makes a fast (intermittently streaming) copy of data on disk to tape, or data on tape to disk. It is usually used to make a copy of a file system. **Fbackup** is faster than **dump** and writes in a format that is understood by **dd** (i.e., it is a byte-by-byte copy), so you should use **fbackup** rather than **dump** if you need the speed.

Fbackup prompts for its arguments. **Fbackup** writes to 9-track tape in block sizes of 16K bytes per record.

To use **fbackup**, you need to know the starting disk address of the file system, and its length in 512-byte disk sectors. To find this out, use **dconfig**(8).

NOTES

This is a Plexus program. It is not part of standard SYSTEM V.

SEE ALSO

Plexus User's Manual

BUGS

Fbackup accepts unsupported combinations of disk and tape and proceeds to copy between a supported combination.

NAME

ff – list file names and statistics for a file system

SYNOPSIS

/etc/ff [options] special

DESCRIPTION

Ff reads the i-list and directories of the *special* file, assuming it to be a file system, saving i-node data for files which match the selection criteria. Output consists of the path name for each saved i-node, plus any other file information requested using the print *options* below. Output fields are positional. The output is produced in i-node order; fields are separated by tabs. The default line produced by *ff* is:

path-name i-number

With all *options* enabled, output fields would be:

path-name i-number size uid

The argument *n* in the *option* descriptions that follow is used as a decimal integer (optionally signed), where +*n* means more than *n* , –*n* means less than *n* , and *n* means exactly *n* . A day is defined as a 24 hour period.

–I Do not print the i-node number after each path name.

–l Generate a supplementary list of all path names for multiply linked files.

–p *prefix* The specified *prefix* will be added to each generated path name. The default is **.** .

–s Print the file size, in bytes, after each path name.

–u Print the owner's login name after each path name.

–a *n* Select if the i-node has been accessed in *n* days.

–m *n* Select if the i-node has been modified in *n* days.

–c *n* Select if the i-node has been changed in *n* days.

–n *file* Select if the i-node has been modified more recently than the argument *file* .

–i *i-node-list* Generate names for only those i-nodes specified in *i-node-list* .

EXAMPLES

To generate a list of the names of all files on a specified file system:

```
ff –l /dev/diskroot
```

To produce an index of files and i-numbers which are on a file system and have been modified in the last 24 hours:

```
ff –m –1 /dev/diskusr > /log/incbackup/usr/tuesday
```

To obtain the path names for i-nodes 451 and 76 on a specified file system:

```
ff –i 451,76 /dev/rdsk/0s7
```

SEE ALSO

finc(1M), find(1), frec(1M), ncheck(1M).

BUGS

Only a single path name out of any possible ones will be generated for a multiply linked i-node, unless the **–l** option is specified. When **–l** is specified, no selection criteria apply to the names generated. All possible names for every linked file on the file system will be included in the output.

On very large file systems, memory may run out before *ff* does.

NAME

filesave, tapesave – daily/weekly UNIX system file system backup

SYNOPSIS

/etc/filesave.?
/etc/tapesave

DESCRIPTION

These shell scripts are provided as models. They are designed to provide a simple, interactive operator environment for file backup. *Filesave.?* is for daily disk-to-disk backup and *tapesave* is for weekly disk-to-tape.

The suffix **.?** can be used to name another system where two (or more) machines share disk drives (or tape drives) and one or the other of the systems is used to perform backup on both.

SEE ALSO

shutdown(1M), volcopy(1M).

NAME

finc – fast incremental backup

SYNOPSIS

finc [selection-criteria] file-system raw-tape

DESCRIPTION

Finc selectively copies the input *file-system* to the output *raw-tape* . The cautious will want to mount the input *file-system* read-only to insure an accurate backup, although acceptable results can be obtained in read-write mode. The tape must be previously labelled by *labelit* (see *volcopy (1M)).* The selection is controlled by the *selection-criteria* , accepting only those i-nodes/files for whom the conditions are true.

It is recommended that production of a *finc* tape be preceded by the *ff* command, and the output of *ff* be saved as an index of the tape's contents. Files on a *finc* tape may be recovered with the *frec* command.

The argument **n** in the *selection-criteria* which follow is used as a decimal integer (optionally signed), where +*n* means more than *n* , –*n* means less than *n* , and *n* means exactly *n* . A day is defined as a 24 hours.

–a *n* True if the file has been accessed in *n* days.

–m *n* True if the file has been modified in *n* days.

–c *n* True if the i-node has been changed in *n* days.

–n *file* True for any file which has been modified more recently than the argument *file* .

EXAMPLES

To write a tape consisting of all files from file-system /**usr** modified in the last 48 hours:

```
finc –m –2 /dev/rdiskusr /dev/rmt/0m
```

SEE ALSO

cpio(1), ff(1M), frec(1M), volcopy(1M).

NAME

frec – recover files from a backup tape

SYNOPSIS

/etc/frec [**–p** path] [**–f** reqfile] raw-tape i-number:name ...

DESCRIPTION

Frec recovers files from the specified *raw-tape* backup tape written by *volcopy*(1M) or *finc*(1M), given their *i-numbers*. The data for each recovery request will be written into the file given by *name*.

The **–p** option allows you to specify a default prefixing *path* different from your current working directory. This will be prefixed to any *names* that are not fully qualified, i.e., that do not begin with / or ./. If any directories are missing in the paths of recovery *names* they will be created.

–p *path* Specifies a prefixing *path* to be used to fully qualify any names that do not start with / or ./.

–f *reqfile* Specifies a file which contains recovery requests. The format is i-number:newname, one per line.

EXAMPLES

To recover a file, i-number 1216 when backed-up, into a file named **junk** in your current working directory:

```
frec /dev/rmt/0m 1216:junk
```

To recover files with i-numbers 14156, 1232, and 3141 into files **/usr/src/cmd/a**, **/usr/src/cmd/b** and **/usr/joe/a.c**:

```
frec –p /usr/src/cmd /dev/rmt/0m 14156:a 1232:b
3141:/usr/joe/a.c
```

SEE ALSO

cpio(1), ff(1M), finc(1M), volcopy(1M).

BUGS

While paving a path (i.e., creating the intermediate directories contained in a pathname) *frec* can only recover i-node fields for those directories contained on the tape and requested for recovery.

NAME

fsck, dfsck – file system consistency check and interactive repair

SYNOPSIS

/etc/fsck [**–y**] [**–n**] [**–s**X] [**–S**X] [**–t** file] [**–q**] [**–D**] [**–f**] [file-systems]

/etc/dfsck [options1] filsys1 **...** – [options2] filsys2 **...**

DESCRIPTION

Fsck

Fsck audits and interactively repairs inconsistent conditions for UNIX system files. If the file system is consistent then the number of files, number of blocks used, and number of blocks free are reported. If the file system is inconsistent the operator is prompted for concurrence before each correction is attempted. It should be noted that most corrective actions will result in some loss of data. The amount and severity of data lost may be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond **yes** or **no**. If the operator does not have write permission *fsck* will default to a **–n action.**

Fsck has more consistency checks than its predecessors *check* , *dcheck* , *fcheck* , and *icheck* combined.

The following options are interpreted by *fsck* .

–y Assume a yes response to all questions asked by *fsck* .

–n Assume a no response to all questions asked by *fsck* ; do not open the file system for writing.

–s *X* Ignore the actual free list and (unconditionally) reconstruct a new one by rewriting the super-block of the file system. The file system should be unmounted while this is done; if this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately afterwards. This precaution is necessary so that the old, bad, in-core copy of the superblock will not continue to be used, or written on the file system.

The **–s** *X* option allows for creating an optimal free-list organization. The following forms of *X* are supported for the following devices:

–s3 (RP03)
–s4 (RP04, RP05, RP06)
–sBlocks-per-cylinder:Blocks-to-skip (for anything else)

If *X* is not given, the values used when the file system was created are used. If these values were not specified, then the value *400*:*7* is used.

–S *X* Conditionally reconstruct the free list. This option is like **–s** *X* above except that the free list is rebuilt only if there were no discrepancies discovered in the file system. Using **–S** will force a no response to all questions asked by *fsck* . This option is useful for forcing free list reorganization on uncontaminated file systems.

–t If *fsck* cannot obtain enough memory to keep its tables, it uses a scratch file. If the **–t** option is specified, the file named in the next argument is used as the scratch file, if needed. Without the **–t flag,** *fsck* will prompt the operator for the name of the scratch file. The file chosen should not be on the file system being checked, and if it is not a special file or did not already exist, it is removed when *fsck* completes.

–q Quiet *fsck.* Do not print size-check messages in Phase 1. Unreferenced **fifos** will silently be removed. If *fsck* requires it, counts in the superblock will be automatically fixed and the free list salvaged.

–D Directories are checked for bad blocks. Useful after system crashes.

–f Fast check. Check block and sizes (Phase 1) and check the free list (Phase 5). The free list will be reconstructed (Phase 6) if it is necessary.

If no *file-systems* are specified, *fsck* will read a list of default file systems from the file **/etc/checklist** .

Inconsistencies checked are as follows:

- Blocks claimed by more than one i-node or the free list.
- Blocks claimed by an i-node or the free list outside the range of the file system.
- Incorrect link counts.
- Size checks:
  - Incorrect number of blocks.
  - Directory size not 16-byte aligned.
- Bad i-node format.
- Blocks not accounted for anywhere.
- Directory checks:
  - File pointing to unallocated i-node.
  - I-node number out of range.
- Super Block checks:
  - More than 65536 i-nodes.
  - More blocks for i-nodes than there are in the file system.
- Bad free block list format.
- Total free block and/or free i-node count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the **lost+found** directory, if the files are nonempty. The user will be notified if the file or directory is empty or not. If it is empty, *fsck* will silently remove them. *Fsck* will force the reconnection of nonempty directories. The name assigned is the i-node number. The only restriction is that the directory **lost+found** must preexist in the root of the file system being checked and must have empty slots in which entries can be made. This is accomplished by making **lost+found** , copying a number of files to the directory, and then removing them (before *fsck* is executed).

Checking the raw device is almost always faster and should be used with everything but the *root* file system.

Dfsck

Dfsck allows two file system checks on two different drives simultaneously. *options1* and *options2* are used to pass options to *fsck* for the two sets of file systems. A – is the separator between the file system groups.

The *dfsck* program permits an operator to interact with two *fsck (1M)* programs at once. To do this, *dfsck* prints the file system name for each message to the operator. When answering a question from *dfsck* , the operator must prefix the response with a **1** or a **2** (indicating that the answer refers to the first or second file system group).

Do not use *dfsck* to check the *root* file system.

FILES

/etc/checklist	contains default list of file systems to check.
/etc/checkall	optimizing *dfsck* shell file.

SEE ALSO

checkall(1M), clri(1M), ncheck(1M), crash(8).
checklist(4), fs(4) in the *Sys5 UNIX Programmer's Reference Manual.*
Setting Up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide* .

BUGS

I-node numbers for **.** and **..** in each directory should be checked for validity.

DIAGNOSTICS

The diagnostics produced by *fsck* are intended to be self-explanatory.

NAME

fsdb – file system debugger

SYNOPSIS

/etc/fsdb special [–]

DESCRIPTION

Fsdb can be used to patch up a damaged file system after a crash. It has conversions to translate block and i-numbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an i-node. These greatly simplify the process of correcting control block entries or descending the file system tree.

Fsdb contains several error-checking routines to verify i-node and block addresses. These can be disabled if necessary by invoking *fsdb* with the optional – argument or by the use of the **O** symbol. (*Fsdb* reads the i-size and f-size entries from the superblock of the file system as the basis for these checks.)

Numbers are considered decimal by default. Octal numbers must be prefixed with a zero. During any assignment operation, numbers are checked for a possible truncation error due to a size mismatch between source and destination.

Fsdb reads a block at a time and will therefore work with raw as well as block I/O. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block.

The symbols recognized by *fsdb* are:

#	absolute address
i	convert from i-number to i-node address
b	convert to block address
d	directory slot offset
+, –	address arithmetic
q	quit
>, <	save, restore an address
=	numerical assignment
=+	incremental assignment
=–	decremental assignment
="	character string assignment
O	error checking flip flop
p	general print facilities
f	file print facility
B	byte mode
W	word mode
D	double word mode

!	escape to shell

The print facilities generate a formatted output in various styles. The current address is normalized to an appropriate boundary before printing begins. It advances with the printing and is left at the address of the last item printed. The output can be terminated at any time by typing the delete character. If a number follows the **p** symbol, that many entries are printed. A check is made to detect block boundary overflows since logically sequential blocks are generally not physically sequential. If a count of zero is used, all entries to the end of the current block are printed. The print options available are:

i	print as i-nodes
d	print as directories
o	print as octal words
e	print as decimal words
c	print as characters
b	print as octal bytes

The **f** symbol is used to print data blocks associated with the current i-node. If followed by a number, that block of the file is printed. (Blocks are numbered from zero.) The desired print option letter follows the block number, if present, or the **f** symbol. This print facility works for small as well as large files. It checks for special devices and that the block pointers used to find the data are not zero.

Dots, tabs, and spaces may be used as function delimiters but are not necessary. A line with just a new-line character will increment the current address by the size of the data type last printed. That is, the address is set to the next byte, word, double word, directory entry or i-node, allowing the user to step through a region of a file system. Information is printed in a format appropriate to the data type. Bytes, words and double words are displayed with the octal address followed by the value in octal and decimal. A **.B** or **.D** is appended to the address for byte and double word values, respectively. Directories are printed as a directory slot offset followed by the decimal i-number and the character representation of the entry name. I-nodes are printed with labeled fields describing each element.

The following mnemonics are used for i-node examination and refer to the current working i-node:

md	mode
ln	link count
uid	user ID number
gid	group ID number
sz	file size

a#	data block numbers (0 – 12)
at	access time
mt	modification time
maj	major device number
min	minor device number

EXAMPLES

386i	prints i-number 386 in an i-node format. This now becomes the current working i-node.
ln=4	changes the link count for the working i-node to 4.
ln=+1	increments the link count by 1.
fc	prints, in ASCII, block zero of the file associated with the working i-node.
2i.fd	prints the first 32 directory entries for the root i-node of this file system.
d5i.fc	changes the current i-node to that associated with the 5th directory entry (numbered from zero) found from the above command. The first logical block of the file is then printed in ASCII.
512B.p0o	prints the superblock of this file system in octal.
2i.a0b.d7=3	changes the i-number for the seventh directory slot in the root directory to 3. This example also shows how several operations can be combined on one command line.
d7.nm="name"	changes the name field in the directory slot to the given string. Quotes are optional when used with **nm** if the first character is alphabetic.
a2b.p0d	prints the third block of the current i-node as directory entries.

SEE ALSO

fsck(1M), dir(4), fs(4).

NAME

fuser – identify processes using a file or file structure

SYNOPSIS

/etc/fuser [**–ku**] files [–] [[**–ku**] files]

DESCRIPTION

Fuser lists the process IDs of the processes using the *files* specified as arguments. For block special devices, all processes using any file on that device are listed. The process ID is followed by **c** , **p** or **r** if the process is using the file as its current directory, the parent of its current directory (only when in use by the system), or its root directory, respectively. If the **–u** option is specified, the login name, in parentheses, also follows the process ID. In addition, if the **–k** option is specified, the **SIGKILL** signal is sent to each process. Only the super-user can terminate another user's process (see *kill (2)).* Options may be respecified between groups of files. The new set of options replaces the old set, with a lone dash canceling any options currently in force.

The process IDs are printed as a single line on the standard output, separated by spaces and terminated with a single new line. All other output is written on standard error.

EXAMPLES

fuser –ku /dev/dsk/1s?

will terminate all processes that are preventing disk drive one from being unmounted if typed by the super-user, listing the process ID and login name of each as it is killed.

fuser –u /etc/passwd

will list process IDs and login names of processes that have the password file open.

fuser –ku /dev/dsk/1s? –u /etc/passwd

will do both of the above examples in a single command line.

FILES

/unix	for namelist
/dev/kmem	for system image
/dev/mem	also for system image

SEE ALSO

mount(1M), ps(1), kill(2), signal(2).

NAME

fwtmp, wtmpfix – manipulate connect accounting records

SYNOPSIS

/usr/lib/acct/fwtmp [–ic]
/usr/lib/acct/wtmpfix [files]

DESCRIPTION

Fwtmp

Fwtmp reads from the standard input and writes to the standard output, converting binary records of the type found in **wtmp** to formatted ASCII records. The ASCII version is useful to enable editing, via *ed*(1), bad records or general purpose maintenance of the file.

The argument **–ic** is used to denote that input is in ASCII form, and output is to be written in binary form.

Wtmpfix

Wtmpfix examines the standard input or named files in **wtmp** format, corrects the time/date stamps to make the entries consistent, and writes to the standard output. A – can be used in place of *files* to indicate the standard input. If time/date corrections are not performed, *acctcon1* will fault when it encounters certain date-change records.

Each time the date is set, a pair of date change records are written to **/etc/wtmp** . The first record is the old date denoted by the string **old time** placed in the line field and the flag **OLD_TIME** placed in the type field of the **<utmp.h>** structure. The second record specifies the new date and is denoted by the string **new time** placed in the line field and the flag **NEW_TIME** placed in the type field. *Wtmpfix* uses these records to synchronize all time stamps in the file.

In addition to correcting time/date stamps, *wtmpfix* will check the validity of the name field to ensure that it consists solely of alphanumeric characters or spaces. If it encounters a name that is considered invalid, it will change the login name to **INVALID** and write a diagnostic to the standard error. In this way, *wtmpfix* reduces the chance that *acctcon1* will fail when processing connect accounting records.

FILES

/etc/wtmp
/usr/include/utmp.h

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), runacct(1M), ed(1), acct(2), acct(4), utmp(4).

NAME

getty – set terminal type, modes, speed, and line discipline

SYNOPSIS

/etc/getty [**–h**] [**–t** timeout] line [speed [type [linedisc]]]
/etc/getty **–c** file

DESCRIPTION

Getty is a program that is invoked by *init (1M).* It is the second process in the series, (*init-getty-login-shell*) that ultimately connects a user with the UNIX system. Initially *getty* generates a system identification message from the values returned by the *uname (2)* system call. Then, if **/etc/issue** exists, it outputs this to the user's terminal, followed finally by the login message field for the entry it is using from **/etc/gettydefs** . *Getty* reads the user's login name and invokes the *login (1)* command with the user's name as argument. While reading the name, *getty* attempts to adapt the system to the speed and type of terminal being used.

Line is the name of a tty line in **/dev** to which *getty* is to attach itself. *Getty* uses this string as the name of a file in the **/dev** directory to open for reading and writing. Unless *getty* is invoked with the **–h** flag, *getty* will force a hangup on the line by setting the speed to zero before setting the speed to the default or specified speed. The **–t** flag plus *timeout* in seconds, specifies that *getty* should exit if the open on the line succeeds and no one types anything in the specified number of seconds. The optional second argument, *speed* , is a label to a speed and tty definition in the file **/etc/gettydefs** . This definition tells *getty* at what speed to initially run, what the login message should look like, what the initial tty settings are, and what speed to try next should the user indicate that the speed is inappropriate (by typing a <*break*> character). The default *speed* is 300 baud. The optional third argument, *type* , is a character string describing to *getty* what type of terminal is connected to the line in question. *Getty* understands the following types:

none	default
vt61	DEC vt61
vt100	DEC vt100
hp45	Hewlett-Packard HP45
c100	Concept 100

The default terminal is **none**; i.e., any crt or normal terminal unknown to the system. Also, for terminal type to have any meaning, the virtual terminal handlers must be compiled into the operating system. They are available, but not compiled in the default condition. The optional fourth argument, *linedisc* , is a character string describing which line discipline to use in communicating with the terminal. Again the hooks for line disciplines are available in the operating system but there is only one presently available, the default line discipline, **LDISC0** .

When given no optional arguments, *getty* sets the *speed* of the interface to 300 baud, specifies that raw mode is to be used (awaken on every character), that echo is to be suppressed, either parity allowed, new-line characters will be converted to carriage return-line feed, and tab expansion performed on the standard output. It types the login message before reading the user's name a character at a time. If a null character (or framing error) is received, it is assumed to be the result of the user pushing the "break" key. This will cause *getty* to attempt the next *speed* in the series. The series that *getty* tries is determined by what it finds in **/etc/gettydefs** .

The user's name is terminated by a new-line or carriage-return character. The latter results in the system being set to treat carriage returns appropriately (see *ioctl (2)).*

The user's name is scanned to see if it contains any lower-case alphabetic characters; if not, and if the name is non-empty, the system is told to map any future upper-case characters into the corresponding lower-case characters.

In addition to the standard UNIX system erase and kill characters (**#** and @), *getty* also understands **\b** and **^U**. If the user uses a **\b** as an erase, or **^U** as a kill character, *getty* sets the standard erase character and/or kill character to match.

Getty also understands the "standard" ESS2 protocols for erasing, killing and aborting a line, and terminating a line. If *getty* sees the ESS erase character, _, or kill character, **$**, or abort character, **&**, or the ESS line terminators, / or !, it arranges for this set of characters to be used for these functions.

Finally, *login* is called with the user's name as an argument. Additional arguments may be typed after the login name. These are passed to *login* , which will place them in the environment (see *login (1)).*

A check option is provided. When *getty* is invoked with the **-c** option and *file*, it scans the file as if it were scanning **/etc/gettydefs** and prints out the results to the standard output. If there are any unrecognized modes or improperly constructed entries, it reports these. If the entries are correct, it prints out the values of the various flags. See *ioctl (2)* to interpret the values. Note that some values are added to the flags automatically.

FILES

/etc/gettydefs
/etc/issue

SEE ALSO

ct(1C), init(1M), login(1), ioctl(2), gettydefs(4), inittab(4), tty(7).

BUGS

While *getty* does understand simple single character quoting conventions, it is not possible to quote the special control characters that *getty* uses to determine when the end of the line has been reached, which protocol is being used, and what the erase character is. Therefore it is not possible to login via *getty* and type a **#**, **@**, **/**, **!**, **_**, backspace, **^U**, **^D**, or **&** as part of your login name or arguments. They will always be interpreted as having their special meaning as described above.

NAME

icpdmp – dump contents of an Intelligent Communication Processor's memory into a file

SYNOPSIS

/etc/icpdmp /dev/ic*X* *file*
where:
X = 0, 1, 2, 3, or 4

DESCRIPTION

Icpdmp reads the contents of an Intelligent Communication Processor's (ICP) memory and dumps it to the specified file. The ICP must be in the reset mode. That is, it has been reset with a system reset for *icpdmp* to work properly. This command can only be executed in single-user state before the ICP is downloaded with a new kernel. The ICP device names are /dev/ic0, /dev/ic1, /dev/ic2, /dev/ic3, /dev/ic4.

If there are ICPs and Advanced Communication Processor's (ACPs) in a system, the ICPs are the lowest numbered devices. ACP device numbers are one greater than the last ICP device number.

EXAMPLE : 3 ICPs and 1 ACP

icp0
icp1
icp2
acp3

FILES

dnld(1M)

NOTES

This is a Plexus command; it is not part of standard System V.
The total number of ACPs and ICPs in a system must not exceed 5.

DIAGNOSTICS

BUGS

Ensure that there is enough disk space before running this program. It requires approximately 64K.

NAME

init, telinit – process control initialization

SYNOPSIS

/etc/init [**0123456SsQq**]

/etc/telinit [**0123456sSQqabc**]

DESCRIPTION

Init

Init is a general process spawner. Its primary role is to create processes from a script stored in the file **/etc/inittab** (see *inittab (4)).* This file usually has *init* spawn *getty 's* on each line that a user may log in on. It also controls autonomous processes required by any particular system.

Init considers the system to be in a *run-level* at any given time. A *run-level* can be viewed as a software configuration of the system where each configuration allows only a selected group of processes to exist. The processes spawned by *init* for each of these *run-levels* is defined in the *inittab* file. *Init* can be in one of eight *run-levels*, **0–6** and **S** or **s** . The *run-level* is changed by having a privileged user run **/etc/init** (which is linked to */etc/telinit).* This user-spawned *init* sends appropriate signals to the orginal *init* spawned by the operating system when the system was rebooted, telling it which *run-level* to change to.

Init is invoked inside the UNIX system as the last step in the boot procedure. The first thing *init* does is to look for **/etc/inittab** and see if there is an entry of the type *initdefault* (see *inittab (4)).* If there is, *init* uses the *run-level* specified in that entry as the initial *run-level* to enter. If this entry is not in *inittab* or *inittab* is not found, *init* requests that the user enter a *run-level* from the virtual system console, **/dev/syscon.** If an **S** (**s**) is entered, *init* goes into the *SINGLE USER* level. This is the only *run-level* that doesn't require the existence of a properly formatted *inittab* file. If **/etc/inittab** doesn't exist, then by default the only legal *run-level* that *init* can enter is the *SINGLE USER* level. In the *SINGLE USER* level the virtual console terminal **/dev/syscon** is opened for reading and writing and the command **/bin/su** is invoked immediately. To exit from the *SINGLE USER run-level* one of two options can be elected. First, if the shell is terminated (via an end-of-file), *init* will reprompt for a new *run-level.* Second, the *init* or *telinit* command can signal *init* and force it to change the *run-level* of the system.

When attempting to boot the system, failure of *init* to prompt for a new *run-level* may be due to the fact that the device **/dev/syscon** is linked to a device other than the physical system teletype (**/dev/systty**). If this occurs, *init* can be forced to relink **/dev/syscon** by typing a delete on the system teletype which is collocated with the processor.

When *init* prompts for the new *run-level*, the operator may enter only one of the digits **0** through **6** or the letters **S** or **s** . If **S** is entered *init* operates as previously described in *SINGLE USER* mode with the additional result that **/dev/syscon** is linked to the user's terminal line, thus making it the virtual system console. A message is generated on the physical console, **/dev/systty** , saying where the virtual terminal has been relocated.

When *init* comes up initially and whenever it switches out of *SINGLE USER* state to normal run states, it sets the *ioctl (2)* states of the virtual console, **/dev/syscon** , to those modes saved in the file **/etc/ioctl.syscon** . This file is written by *init* whenever *SINGLE USER* mode is entered. If this file does not exist when *init* wants to read it, a warning is printed and default settings are assumed.

If a **0** through **6** is entered *init* enters the corresponding *run-level .* Any other input will be rejected and the user will be re-prompted. If this is the first time *init* has entered a *run-level* other than *SINGLE USER , init* first scans *inittab* for special entries of the type *boot* and *bootwait .* These entries are performed, providing the *run-level* entered matches that of the entry before any normal processing of *inittab* takes place. In this way any special initialization of the operating system, such as mounting file systems, can take place before users are allowed onto the system. The *inittab* file is scanned to find all entries that are to be processed for that *run-level* .

Run-level **2** is usually defined by the user to contain all of the terminal processes and daemons that are spawned in the multi-user environment.

In a multi-user environment, the *inittab* file is usually set up so that *init* will create a process for each terminal on the system.

For terminal processes, ultimately the shell will terminate because of an end-of-file either typed explicitly or generated as the result of hanging up. When *init* receives a child death signal, telling it that a process it spawned has died, it records the fact and the reason it died in **/etc/utmp** and **/etc/wtmp** if it exists (see *who (1)).* A history of the processes spawned is kept in **/etc/wtmp** if such a file exists.

To spawn each process in the *inittab* file, *init* reads each entry and for each entry which should be respawned, it forks a child process. After it has spawned all of the processes specified by the *inittab* file, *init* waits for one of its descendant processes to die, a powerfail signal, or until *init* is signaled by *init* or *telinit* to change the system's *run-level*. When one of the above three conditions occurs, *init* re-examines the *inittab* file. New entries can be added to the *inittab* file at any time; however, *init* still waits for one of the above three conditions to occur. To provide for an instantaneous response the **init Q** or **init q** command can wake *init* to re-examine the *inittab* file.

If *init* receives a *powerfail* signal (*SIGPWR*) and is not in *SINGLE USER* mode, it scans *inittab* for special powerfail entries. These entries are invoked (if the *run-levels* permit) before any further processing takes place. In this way *init* can perform various cleanup and recording functions whenever the operating system experiences a power failure.

When *init* is requested to change *run-levels* (via *telinit*), *init* sends the warning signal (**SIGTERM**) to all processes that are undefined in the target *run-level* . *Init* waits 20 seconds before forcibly terminating these processes via the kill signal (**SIGKILL**).

Telinit

Telinit , which is linked to */etc/init* , is used to direct the actions of *init* . It takes a one-character argument and signals *init* via the kill system call to perform the appropriate action. The following arguments serve as directives to *init* .

- **0–6** tells *init* to place the system in one of the *run-levels* **0–6** .
- **a , b , c** tells *init* to process only those **/etc/inittab** file entries having the **a** , **b** or **c** *run-level* set.
- **Q , q** tells *init* to re-examine the **/etc/inittab** file.
- **s , S** tells *init* to enter the single user environment. When this level change is effected, the virtual system teletype, **/dev/syscon** , is changed to the terminal from which the command was executed.

Telinit can only be run by someone who is super-user or a member of group **sys**.

FILES

/etc/inittab
/etc/utmp
/etc/wtmp
/etc/ioctl.syscon
/dev/syscon
/dev/systty

SEE ALSO

getty(1M), login(1), sh(1), who(1), kill(2), inittab(4), utmp(4).

DIAGNOSTICS

If *init* finds that it is continuously respawning an entry from **/etc/inittab** more than 10 times in 2 minutes, it will assume that there is an error in the command string, and generate an error message on the system console, and refuse to respawn this entry until either 5 minutes has elapsed or it receives a signal from a user *init* (*telinit*). This prevents *init* from eating up system resources when someone makes a typographical error in the *inittab* file or a program is removed that is referenced in the *inittab.*

NAME

install – install commands

SYNOPSIS

/etc/install [**–c** dira] [**–f** dirb] [**–i**] [**–n** dirc] [**–o**] [**–s**] file [dirx ...]

DESCRIPTION

Install is a command most commonly used in "makefiles" (see *make (1))* to install a *file* (updated target file) in a specific place within a file system. Each *file* is installed by copying it into the appropriate directory, thereby retaining the mode and owner of the original command. The program prints messages telling the user exactly what files it is replacing or creating and where they are going.

If no options or directories (*dirx ...*) are given, *install* will search a set of default directories (**/bin**, **/usr/bin** , **/etc** , **/lib** , and **/usr/lib** , in that order) for a file with the same name as *file* . When the first occurrence is found, *install* issues a message saying that it is overwriting that file with *file* , and proceeds to do so. If the file is not found, the program states this and exits without further action.

If one or more directories (*dirx ...*) are specified after *file* , those directories will be searched before the directories specified in the default list.

The meanings of the options are:

–c *dira* Installs a new command (*file*) in the directory specified by *dira* , only if it is not found. If it is found, *install* issues a message saying that the file already exists, and exits without overwriting it. May be used alone or with the **–s** option.

–f *dirb* Forces *file* to be installed in given directory, whether or not one already exists. If the file being installed does not already exist, the mode and owner of the new file will be set to **755** and **bin**, respectively. If the file already exists, the mode and owner will be that of the already existing file. May be used alone or with the **–o** or **–s** options.

–i Ignores default directory list, searching only through the given directories (*dirx ...*). May be used alone or with any other options other than **–c** and **–f** .

–n *dirc* If *file* is not found in any of the searched directories, it is put in the directory specified in *dirc* . The mode and owner of the new file will be set to **755** and **bin**, respectively. May be used alone or with any other options other than **–c** and **–f** .

–o If *file* is found, this option saves the "found" file by copying it to **OLD***file* in the directory in which it was found. This option is useful when installing a normally text busy file such as **/bin/sh** or **/etc/getty** , where the existing file cannot be removed. May be used alone or with any other options other than **–c** .

–s Suppresses printing of messages other than error messages. May be used alone or with any other options.

SEE ALSO

make(1).

NAME

killall – kill all active processes

SYNOPSIS

/etc/killall [signal]

DESCRIPTION

Killall is a procedure used by **/etc/shutdown** to kill all active processes not directly related to the shutdown procedure.

Killall is chiefly used to terminate all processes with open files so that the mounted file systems will be unbusied and can be unmounted.

Killall sends *signal* (see *kill (1))* to all remaining processes not belonging to the above group of exclusions. If no *signal* is specified, a default of **9** is used.

FILES

/etc/shutdown

SEE ALSO

fuser(1M), kill(1), ps(1), shutdown(1M), signal(2).

NAME

link, unlink – exercise link and unlink system calls

SYNOPSIS

/etc/link file1 file2
/etc/unlink file

DESCRIPTION

Link and *unlink* perform their respective system calls on their arguments, abandoning all error checking. These commands may only be executed by the super-user, who (it is hoped) knows what he or she is doing.

SEE ALSO

rm(1), link(2), unlink(2).

NAME

lpadmin – configure the LP spooling system

SYNOPSIS

/usr/lib/lpadmin –p printer [options]
/usr/lib/lpadmin –x dest
/usr/lib/lpadmin –d [dest]

DESCRIPTION

Lpadmin configures LP spooling systems to describe printers, classes and devices. It is used to add and remove destinations, change membership in classes, change devices for printers, change printer interface programs and to change the system default destination. *Lpadmin* may not be used when the LP scheduler, *lpsched*, is running, except where noted below.

Exactly one of the **–p** , **–d** or **–x** options must be present for every legal invocation of *lpadmin* .

–d [*dest*] makes *dest*, an existing destination, the system default destination. If *dest* is not supplied, there is no system default destination. This option may be used when *lpsched* is running. No other *options* are allowed with **–d** .

–x*dest* removes destination *dest* from the LP system. If *dest* is a printer and is the only member of a class, then the class will be deleted, also. No other *options* are allowed with **–x** .

–p*printer* names a *printer* to which all of the *options* below refer. If *printer* does not exist, it will be created.

The following *options* are only useful with **–p** and may appear in any order. In the following examples the printer will be referred to as *P*.

–c*class* inserts printer *P* into the specified *class* . *Class* will be created if it does not already exist.

–e*printer* copies an existing *printer's* interface program to be the new interface program for *P* .

–h indicates that the device associated with *P* is hardwired. This *option* is assumed when creating a new printer unless the **–l** *option* is supplied.

–i*interface* establishes a new interface program for *P* . *Interface* is the pathname of the new program.

–l indicates that the device associated with *P* is a login terminal. The LP scheduler, *lpsched*, automatically disables all login terminals each time it is started. Before re-enabling *P* , its current *device* should be established using *lpadmin* .

–m*model* selects a model interface program for *P* . *Model* is one of the model interface names supplied with the LP software (see *Models* below).

–r*class* removes printer *P* from the specified *class* . If *P* is the last member of the *class* , then the *class* will be removed.

–v*device* associates a new *device* with printer *P* . *Device* is the pathname of a file that is writable by the LP administrator, *lp* . Note that there is nothing to stop an administrator from associating the same *device* with more than one *printer* . If only the **–p** and **–v** *options* are supplied, then *lpadmin* may be used while the scheduler is running.

Restrictions.

When creating a new printer, the **–v** option and one of the **–e** , **–i** or **–m** options must be supplied. Only one of the **–e** , **–i** or **–m** options may be supplied. The **–h** and **–l** keyletters are mutually exclusive. Printer and class names may be no longer than 14 characters and must consist entirely of the characters **A - Z** , **a - z** , **0 - 9** and _ (underscore).

Models.

Model printer interface programs are supplied with the LP software. They are shell procedures which interface between *lpsched (1M)* and devices. All models reside in the directory **/usr/spool/lp/model** and may be used as is with *lpadmin* **–m** . Models should have 644 permission if owned by lp & bin, or 664 permission if owned by bin & bin. Alternatively, LP administrators may modify copies of models and then use *lpadmin* **–i** to associate them with printers. The following list describes the *models* and lists the options which they may be given on the *lp* command line using the **–o** keyletter:

dumb interface for a line printer without special functions and protocol. Form feeds are assumed. This is a good model to copy and modify for printers which do not have models.

1640 DIABLO 1640 terminal running at 1200 baud, using XON/XOFF protocol. Options:

- **–12** 12-pitch (10-pitch is the default)
- **–f** do not use the *450 (1)* filter. The output has been pre-processed by either *450 (1)* or the *nroff (1)* 450 driving table.

hp Hewlett-Packard 2631A line printer at 2400 baud. Options:

- **–c** compressed print
- **–e** expanded print

prx Printronix P300 or P600 printer using XON/XOFF protocol at 1200 baud.

EXAMPLES

1. Assuming there is an existing Hewlett-Packard 2631A line printer named *hp2* , it will use the **hp** model interface after the command:

   /usr/lib/lpadmin –php2 –mhp

2. To obtain compressed print on *hp2* , use the command:

   lp –dhp2 –o–c files

3. A DIABLO 1640 printer called *st1* can be added to the LP configuration with the command:

   /usr/lib/lpadmin –pst1 –v/dev/tty20 –m1640

4. An *nroff (1)* document may be printed on *st1* in any of the following ways:

   ```
   nroff –T450 files | lp –dst1 –of
   nroff –T450–12 files | lp –dst1 –of
   nroff –T37 files | col | lp –dst1
   ```

5. The following command prints the password file on *st1* in 12-pitch:

   lp –dst1 –o12 /etc/passwd

   NOTE: the **–12** option to the **1640** model should never be used in conjunction with *nroff (1)* .

FILES

/usr/spool/lp/*

SEE ALSO

accept(1M), lpsched(1M), enable(1), lp(1), lpstat(1), nroff(1).

NAME

lpsched, lpshut, lpmove – start/stop the LP request scheduler and move requests

SYNOPSIS

/usr/lib/lpsched
/usr/lib/lpshut
/usr/lib/lpmove requests dest
/usr/lib/lpmove dest1 dest2

DESCRIPTION

Lpsched schedules requests taken by *lp (1)* for printing on line printers.

Lpshut shuts down the line printer scheduler. All printers that are printing at the time *lpshut* is invoked will stop printing. Requests that were printing at the time a printer was shut down will be reprinted in their entirety after *lpsched* is started again. All LP commands perform their functions even when *lpsched* is not running.

Lpmove moves requests that were queued by *lp (1)* between LP destinations. This command may be used only when *lpsched* is not running.

The first form of the command moves the named *requests* to the LP destination, *dest* . *Requests* are request ids as returned by *lp (1)* . The second form moves all requests for destination *dest1* to destination *dest2* . As a side effect, *lp (1)* will reject requests for *dest1* .

Note that *lpmove* never checks the acceptance status (see *accept (1M))* for the new destination when moving requests.

FILES

/usr/spool/lp/*

SEE ALSO

accept(1M), enable(1), lp(1), lpadmin(1M), lpstat(1).

NAME

mirutil – utility for connecting two identical disks as a mirrored pair.

SYNOPSIS

/etc/mirutil
/etc/mirutil [**-s**] [**-druv**] *pdn* [**-m**] *pdn sdn*
where:
pdn = primary drive number
sdn = secondary drive number

DESCRIPTION

This utility is run from the UNIX shell. Each of the seven available functions can be selected from the **mirutil** menu. From the **mirutil** menu, enter only the first letter of the option, either upper or lower case.

All of the functions except *Quit* can be invoked directly from the shell. Either upper or lower case is accepted.

-s *pdn* Status displays the current status of all disks on the system.

-m *pdn sdn* Mirror assigns (or mirrors) two disks as a mirrored pair. The primary drive will be mirrored by the secondary drive. Enter the primary drive number first, followed by the secondary drive number.

-d *pdn* Disconnect temporarily separates a pair of disks which are set up as a mirror. While disconnected they will not be kept in sync. The drives are not unmirrored and can be reconnected at any time with the Reconnect option.

-r *pdn* Reconnect reconnects a disconnected mirrored disk.

-u *pdn* Unmirror unmirrors the disks of an existing mirrored pair. The unmirror option permanently detaches two drives and erases the mirror information for this pair.

-v *pdn* Verify verifies that a mirrored pair of drives is in agreement. The verify option checks the mirrored pair, sector by sector. You may use the drive that displays `In Use` under the `Status` column while the verify operation is taking place.

q Quit exits the **mirutil** menu and returns the operating system prompt. This option is only invoked from the **mirutil** menu.

You must have root or superuser status for all functions *except* Status and Quit.

FILES

/dev/rdsk/0s0

SEE ALSO

Sys5 Administrator's Handbook

NAME

mkfs – construct a file system

SYNOPSIS

/etc/mkfs special blocks[:inodes] [gap blocks/cyl]
/etc/mkfs special proto [gap blocks/cyl]

DESCRIPTION

Mkfs constructs a file system by writing on the special file according to the directions found in the remainder of the command line. The command waits 10 seconds before starting to construct the file system. If the second argument is given as a string of digits, *mkfs* builds a file system with a single empty directory on it. The size of the file system is the value of *blocks* interpreted as a decimal number. The boot program is left uninitialized. If the optional number of inodes is not given, the default is the number of *logical* blocks divided by 4.

If the second argument is a file name that can be opened, *mkfs* assumes it to be a prototype file *proto*, and will take its directions from that file. The prototype file contains tokens separated by spaces or new-lines. The first token is the name of a file to be copied onto block zero as the bootstrap program. The second token is a number specifying the size of the created file system in *physical* disk blocks. Typically it will be the number of blocks on the device, perhaps diminished by space for swapping. The next token is the number of i-nodes in the file system. The maximum number of i-nodes configurable is 65500. The next set of tokens comprise the specification for the root file. File specifications consist of tokens giving the mode, the user ID, the group ID, and the initial contents of the file. The syntax of the contents field depends on the mode.

The mode token for a file is a 6-character string. The first character specifies the type of the file. (The characters **–bcd** specify regular, block special, character special and directory files respectively.) The second character of the type is either **u** or – to specify set-user-id mode or not. The third is **g** or – for the set-group-id mode. The rest of the mode is a three digit octal number giving the owner, group, and other read, write, execute permissions (see *chmod*(1)).

Two decimal number tokens come after the mode; they specify the user and group ID's of the owner of the file.

If the file is a regular file, the next token is a pathname whence the contents and size are copied. If the file is a block or character special file, two decimal number tokens follow which give the major and minor device numbers. If the file is a directory, *mkfs* makes the entries . and .. and then reads a list of names and (recursively) files specifications for the entries in the directory. The scan is terminated with the token **$**.

A sample prototype specification follows:

```
/stand/diskboot
4872 110
d—777 3 1
usr     d—777 3 1
        sh      ——755 3 1 /bin/sh
        ken     d—755 6 1
                $
        b0      b—644 3 1 0 0
        c0      c—644 3 1 0 0
        $
$
```

In both command syntaxes, the rotational *gap* and the number of *blocks/cyl* can be specified. The *default* will be used if the supplied *gap* and *blocks/cyl* are considered illegal values or if a short argument count occurs. Your User's Manual lists the default values for your system.

The best gap factor should be calculated as:

gap = (sectors per track / 2) + number of heads

If you are using a Xylogics disk (P/60 and P/75 only) and do not enter the gap size and blocks/cyl, a warning will be displayed. See the Plexus user's guide for your system for information on gap size and blocks per cylinder.

At any time during the program you can <DEL> to cancel the program and start over.

A new flag value has been added to help **mkfs** to do its job quickly and quietly. A **-q** before the device name prevents it from sleeping, or from printing any warnings or statistics.

SEE ALSO

chmod(1), dir(4), fs(4).

BUGS

If a prototype is used, it is not possible to initialize a file larger than 64K bytes, nor is there a way to specify links.

NAME

mknod – build special file

SYNOPSIS

/etc/mknod name **c** | **b** major minor
/etc/mknod name **p**

DESCRIPTION

Mknod makes a directory entry and corresponding i-node for a special file. The first argument is the *name* of the entry. In the first case, the second is **b** if the special file is block-type (disks, tape) or **c** if it is character-type (other devices). The last two arguments are numbers specifying the *major* device type and the *minor* device (e.g., unit, drive, or line number), which may be either decimal or octal.

The assignment of major device numbers is specific to each system. They have to be dug out of the system source file **conf.c .**

Mknod can also be used to create fifo's (a.k.a named pipes) (second case in *SYNOPSIS* above).

SEE ALSO

mknod(2).

NAME

mount, umount – mount and dismount file system

SYNOPSIS

/etc/mount [special directory [**–r**]]

/etc/umount special

DESCRIPTION

Mount announces to the system that a removable file system is present on the device *special* . The *directory* must exist already; it becomes the name of the root of the newly mounted file system.

These commands maintain a table of mounted devices. If invoked with no arguments, *mount* prints the table.

The optional last argument indicates that the file is to be mounted read-only. Physically write-protected and magnetic tape file systems must be mounted in this way or errors will occur when access times are updated, whether or not any explicit write is attempted.

Umount announces to the system that the removable file system previously mounted on device *special* is to be removed.

FILES

/etc/mnttab   mount table

SEE ALSO

setmnt(1M), mount(2), mnttab(4).

DIAGNOSTICS

Mount issues a warning if the file system to be mounted is currently mounted under another name.

Umount complains if the special file is not mounted or if it is busy. The file system is busy if it contains an open file or some user's working directory.

BUGS

Some degree of validation is done on the file system; however, it is generally unwise to mount garbage file systems.

NAME

mvdir – move a directory

SYNOPSIS

/etc/mvdir dirname name

DESCRIPTION

Mvdir moves directories within a file system. *Dirname* must be a directory; *name* must not exist. Neither name may be a sub-set of the other (**/x/y** cannot be moved to **/x/y/z** , nor vice versa).

Only super-user can use *mvdir* .

SEE ALSO

mkdir(1).

NAME

ncheck – generate names from i-numbers

SYNOPSIS

/etc/ncheck [**–i** numbers] [**–a**] [**–s**] [file-system]

DESCRIPTION

Ncheck with no argument generates a path-name vs. i-number list of all files on a set of default file systems. Names of directory files are followed by /. . The **–i** option reduces the report to only those files whose i-numbers follow. The **–a** option allows printing of the names **.** and **..** , which are ordinarily suppressed. The **–s** option reduces the report to special files and files with set-user-ID mode; it is intended to discover concealed violations of security policy.

A file system may be specified.

The report is in no useful order, and probably should be sorted.

SEE ALSO

fsck(1M), sort(1).

DIAGNOSTICS

When the file system structure is improper, **??** denotes the "parent" of a parentless file and a path-name beginning with **...** denotes a loop.

NAME

non-btl – reinstall MM macros without Bell Laboratories specific features

SYNOPSIS

sh non-btl.sh

DESCRIPTION

The *non-btl.sh* command will modify and re-install the source for the Memorandum Macros (used with *nroff* and *troff*) when Bell Labs specific macros are not desired.

Specifically, use of the *non-btl.sh* command will remove the **.TM**, **.PM, .CS** macros, and the }**2** string (which normally contains the name "Bell Laboratories") from the macro package. After running *non-btl.sh*, use of these features will have no effect.

This command does not remove the source for these features from the macro file, but does erase their definition. Those users who wish to tailor the macro package to their own environment may choose to not run *non-btl.sh*, but to modify the definition of the affected macros and string to their own specifications. Remember to re-install the macros after they are modified.

IMPORTANT

The *non-btl.sh* command is found in the directory **/usr/src/cmd/text/macros.d**, and may be run only by the super-user.

NAME

prfld, prfstat, prfdc, prfsnap, prfpr – operating system profiler

SYNOPSIS

/etc/prfld [namelist]
/etc/prfstat on
/etc/prfstat off
/etc/prfdc file [period [off_hour]]
/etc/prfsnap file
/etc/prfpr file [cutoff [namelist]]

DESCRIPTION

Prfld , *prfstat* , *prfdc* , *prfsnap* , and *prfpr* form a system of programs to facilitate an activity study of the UNIX operating system.

Prfld is used to initialize the recording mechanism in the system. It generates a table containing the starting address of each system subroutine as extracted from *namelist* .

Prfstat is used to enable or disable the sampling mechanism. Profiler overhead is less than 1% as calculated for 500 text addresses. *Prfstat* will also reveal the number of text addresses being measured.

Prfdc and *prfsnap* perform the data collection function of the profiler by copying the current value of all the text address counters to a file where the data can be analyzed. *Prfdc* will store the counters into *file* every *period* minutes and will turn off at *off_hour* (valid values for *off_hour* are **0 – 24).** *Prfsnap* collects data at the time of invocation only, appending the counter values to *file* .

Prfpr formats the data collected by *prfdc* or *prfsnap* . Each text address is converted to the nearest text symbol (as found in *namelist)* and is printed if the percent activity for that range is greater than *cutoff* .

FILES

/dev/prf	interface to profile data and text addresses
/unix	default for namelist file

SEE ALSO

prf(7).

NAME

pwck, grpck – password/group file checkers

SYNOPSIS

/etc/pwck [file]
/etc/grpck [file]

DESCRIPTION

Pwck scans the password file and notes any inconsistencies. The checks include validation of the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. The criteria for determining a valid login name is derived from Setting up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide* . The default password file is **/etc/passwd .**

Grpck verifies all entries in the group file. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the password file. The default group file is **/etc/group .**

FILES

/etc/group
/etc/passwd

SEE ALSO

group(4), passwd(4).
Setting Up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide .*

DIAGNOSTICS

Group entries in **/etc/group** with no login names are flagged.

NAME

ramdisk – memory as disk

SYNOPSIS

/usr/plx/ramdisk *devname* [**-p** | **size** [*k* | *m*]]

DESCRIPTION

This is a new driver that allows memory to be used as a disk. A new utility exists to allocate memory to one of up to eight devices. There are two new directories for the nodes: **/dev/rram** contains the raw devices which are mostly used as paramters to the utility, **/dev/ram** contains block devices which can be configured to be mounted file systems.

devname — name of a node for the ram device. Usually in **/dev/rram**.

size [k | m] — number of bytes of memory to allocate to the device. The optional *k* multiplies the number by 1024. The optional *m* multiplies by 1024*1024.

-p — causes the device to be used for **pipedev**. It is doubtful that this is a performance improvement. The device must already be a mounted file system or an error results. If the block device is completely closed (i.e. unmounted) **pipedev** will revert to its value at boot time.

FILES

/dev/rram

SEE ALSO

mkfs(1m), rram(7).

NAME

restor – incremental file system restore

SYNOPSIS

restor *key* [*arguments*]

DESCRIPTION

Restor is used to read magnetic tapes dumped with the *dump* command. A *dump* followed by a *mkfs* and a *restor* is used to change the size of a file system.

In the standalone version of this program, a final +*n* argument advances the tape *n* files before executing the *restor*. To space forward *n* files in the online version type

```
/usr/plx/tape srcheof n
```

before typing the *restor* command.

The *key* specifies what is to be done. *Key* is one of the characters **rRxt**, optionally combined with **f**.

f Use the first *argument* as the name of the tape instead of the default.

r or **R** The tape is read and loaded into the file system specified in *argument*. If the key is **R**, *restor* asks which tape of a multi-volume set to start on. This allows *restor* to be interrupted and then restarted (an *fsck* must be done before the restart). The **r** option should only be used to restore a complete dump tape onto a clear file system, or to restore an incremental dump tape onto a file system so created. Thus:

```
/etc/mkfs /dev/dsk/0s1 18000
restor r /dev/dsk/0s1
```

is a typical sequence to restore a complete dump. Another *restor* can be done to get an incremental dump in on top of this.

x Each file on the tape named by an *argument* is extracted. The file name has all "mount" prefixes removed; for example, if **/usr** is a mounted file system, **/usr/bin/lpr** is named **/bin/lpr** on the tape. The extracted file is placed in a file with a numeric name supplied by *restor* (actually the inode number). In order to keep the amount of tape read to a minimum, the following procedure is recommended:

1. Mount volume 1 of the set of dump tapes.
2. Type the *restor* command.

3. *Restor* will announce whether or not it found the files, give the numeric name that it will assign to the file, and rewind the tape.

4. It then asks you to "mount the desired tape volume". Type the number of the volume. On a multi-volume dump the recommended procedure is to mount the last through the first volumes, in that order. *Restor* checks to see if any of the requested files are on the mounted tape (or a later tape–thus the reverse order) and doesn't read through the tape if no files are. If you are working with a single-volume dump or if the number of files being restored is large, respond to the query with **1** and *restor* will read the tapes in sequential order.

t Print the date the tape was written and the date the file system was dumped from.

FILES

/dev/rpt/0m	(cartrige tape - rewind)
/dev/rpt/0mn	(cartrige tape - no rewind)
/dev/rrm/0m	(9-track tape - rewind)
/dev/rrm/0mn	(9-track tape - no rewind)
rst*	

NOTES

This command has a standalone version.

SEE ALSO

dump(1M), dumpdir(1M), fsck(1M), mkfs(1M).

DIAGNOSTICS

There are various diagnostics involved with reading the tape and writing the disk. There are also diagnostics if the i-list or the free list of the file system is not large enough to hold the dump.

If the dump extends over more than one tape, it may ask you to change tapes. Reply with a new-line when the next tape has been mounted.

BUGS

There is redundant information on the tape that could be used in case of tape reading problems. Unfortunately, *restor* doesn't use it. The **x** option of the standalone version does not work.

The Sys5 version of *restor* cannot read multiple volume dumps made with the Sys3 version of *dump*. If you have multiple volume dumps of a Sys3 file system, use the standalone *restor* on your old Sys3 release tape to load the dump onto your new Sys5 file system. Then use the Sys5 version of */etc/dump* to make a new backup.

NAME

runacct – run daily accounting

SYNOPSIS

/usr/lib/acct/runacct [mmdd [state]]

DESCRIPTION

Runacct is the main daily accounting shell procedure. It is normally initiated via *cron (1M). Runacct* processes connect, fee, disk, and process accounting files. It also prepares summary files for *prdaily* or billing purposes.

Runacct takes care not to damage active accounting files or summary files in the event of errors. It records its progress by writing descriptive diagnostic messages into **active** . When an error is detected, a message is written to **/dev/console** , mail (see *mail*(1)) is sent to **root and adm** , and *runacct* terminates. *Runacct* uses a series of lock files to protect against re-invocation. The files **lock** and **lock1** are used to prevent simultaneous invocation, and **last-date** is used to prevent more than one invocation per day.

Runacct breaks its processing into separate, restartable *states* using **statefile** to remember the last *state* completed. It accomplishes this by writing the *state* name into **statefile** . *Runacct* then looks in **statefile** to see what it has done and to determine what to process next. *States* are executed in the following order:

SETUP	Move active accounting files into working files.
WTMPFIX	Verify integrity of **wtmp** file, correcting date changes if necessary.
CONNECT1	Produce connect session records in **ctmp.h** format.
CONNECT2	Convert **ctmp.h** records into **tacct.h** format.
PROCESS	Convert process accounting records into **tacct.h** format.
MERGE	Merge the connect and process accounting records.
FEES	Convert output of *chargefee* into **tacct.h** format and merge with connect and process accounting records.
DISK	Merge disk accounting records with connect, process, and fee accounting records.
MERGETACCT	Merge the daily total accounting records in **daytacct** with the summary total accounting records in **/usr/adm/acct/sum/tacct** .

CMS	Produce command summaries.
USEREXIT	Any installation-dependent accounting programs can be included here.
CLEANUP	Cleanup temporary files and exit.

To restart *runacct* after a failure, first check the **active** file for diagnostics, then fix up any corrupted data files such as **pacct or wtmp** . The **lock** files and **lastdate** file must be removed before *runacct* can be restarted. The argument *mmdd* is necessary if *runacct* is being restarted, and specifies the month and day for which *runacct* will rerun the accounting. Entry point for processing is based on the contents of **statefile** ; to override this, include the desired *state* on the command line to designate where processing should begin.

EXAMPLES

To start *runacct*, enter:
nohup runacct 2> /usr/adm/acct/nite/fd2log & ®.in -5

To restart *runacct*, enter:
nohup runacct 0601 2>> /usr/adm/acct/nite/fd2log & ®.in -5

To restart *runacct* at a specific *state*, enter:
nohup runacct 0601 MERGE 2>> /usr/adm/acct/nite/fd2log & ®.in -5

FILES

/etc/wtmp
/usr/adm/pacct*
/usr/src/cmd/acct/tacct.h
/usr/src/cmd/acct/ctmp.h
/usr/adm/acct/nite/active
/usr/adm/acct/nite/daytacct
/usr/adm/acct/nite/lock
/usr/adm/acct/nite/lock1
/usr/adm/acct/nite/lastdate
/usr/adm/acct/nite/statefile
/usr/adm/acct/nite/ptacct*.*mmdd*

SEE ALSO

acct(1M), acctcms(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), cron(1M), fwtmp(1M).
mail(1) in the *Sys5 UNIX User's Reference Manual.*
acct(2), acct(4), utmp(4) in the Sys5 UNIX Programmer's Reference Manual.
Sys5 UNIX Accounting System in the *Sys5 UNIX Administrator's Guide* .

BUGS

Normally it is not a good idea to restart *runacct* in the **SETUP** *state* .

Run **SETUP** manually and restart via:

runacct *mmdd* **WTMPFIX**

If *runacct* failed in the **PROCESS** *state,* remove the last **ptacct** file because it will not be complete.

NAME

sadp – disk access profiler

SYNOPSIS

sadp [**–th**] [**–d** device[–drive]] s [n]

DESCRIPTION

Sadp reports disk access location and seek distance, in tabular or histogram form. It samples disk activity once every second during an interval of *s* seconds. This is done repeatedly if *n* is specified. Cylinder usage and disk distance are recorded in units of 8 cylinders.

Valid values of *device* are **rp06, rm05,** and **disk.** *Drive* specifies the disk drives and it may be:

> a drive number in the range supported by *device* ,
> two numbers separated by a minus (indicating an inclusive range),

or

> a list of drive numbers separated by commas.

Up to 8 disk drives may be reported. The **–d** option may be omitted, if only one *device* is present.

The **–t** flag causes the data to be reported in tabular form. The **–h** flag produces a histogram on the printer of the data. Default is **–t** .

EXAMPLE

The command:

```
sadp –d rp06–0 900 4
```

will generate 4 tabular reports, each describing cylinder usage and seek distance of rp06 disk drive 0 during a 15-minute interval.

FILES

/dev/kmem

NAME

sa1, sa2, sadc – system activity report package

SYNOPSIS

/usr/lib/sa/sadc [t n] [ofile]

/usr/lib/sa/sa1 [t n]

/usr/lib/sa/sa2 [**–ubdycwaqvmA**] [**–s** time] [**–e** time] [**–i** sec]

DESCRIPTION

System activity data can be accessed at the special request of a user (see *sar (1))* and automatically on a routine basis as described here. The operating system contains a number of counters that are incremented as various system actions occur. These include CPU utilization counters, buffer usage counters, disk and tape I/O activity counters, TTY device activity counters, switching and system-call counters, file-access counters, queue activity counters, and counters for inter-process communications.

Sadc and shell procedures, *sa1* and *sa2* , are used to sample, save, and process this data.

Sadc , the data collector, samples system data *n* times every *t* seconds and writes in binary format to *ofile* or to standard output. If *t* and *n* are omitted, a special record is written. This facility is used at system boot time to mark the time at which the counters restart from zero. The **/etc/rc** entry:

```
su sys –c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`"
```

writes the special record to the daily data file to mark the system restart.

The shell script *sa1* , a variant of *sadc* , is used to collect and store data in binary file **/usr/adm/sa/sa***dd* where *dd* is the current day. The arguments *t* and *n* cause records to be written *n* times at an interval of *t* seconds, or once if omitted. The entries in **crontab** (see *cron (1M)):*

```
0 * * * 0,6 su sys –c "/usr/lib/sa/sa1"
0 8–17 * * 1–5 su sys –c "/usr/lib/sa/sa1 1200 3"
0 18–7 * * 1–5 su sys –c "/usr/lib/sa/sa1"
```

will produce records every 20 minutes during working hours and hourly otherwise.

The shell script *sa2* , a variant of *sar (1),* writes a daily report in file **/usr/adm/sa/sar***dd*. The options are explained in *sar (1).* The **crontab** entry:

> 5 18 * * 1–5 su adm –c "/usr/lib/sa/sa2 –s 8:00 –e 18:01 –i 3600 –A"

will report important activities hourly during the working day.

The structure of the binary daily data file is:

```
struct sa {
        struct sysinfo si;  /* see /usr/include/sys/sysinfo.h */
        int  szi-node;      /* current entries of i-node table */
        int  szfile;        /* current entries of file table */
        int  sztext;        /* current entries of text table */
        int  szproc;        /* current entries of proc table */
        int  mszi-node;     /* size of i-node table */
        int  mszfile;       /* size of file table */
        int  msztext;       /* size of text table */
        int  mszproc;       /* size of proc table */
        long  i-nodeovf;    /* cumul. overflows of i-node table */
        long  fileovf;      /* cumul. overflows of file table */
        long  textovf;      /* cumul. overflows of text table */
        long  procovf;      /* cumul. overflows of proc table */
        time_t  ts;         /* time stamp, seconds */
        long  devio[NDEVS][4];      /* device info for up to NDEVS units
#define IO_OPS          0           /* cumul. I/O requests */
#define IO_BCNT         1           /* cumul. blocks transferred */
#define IO_ACT          2           /* cumul. drive busy time in ticks */
#define IO_RESP         3           /* cumul. I/O resp time in ticks */
};
```

FILES

/usr/adm/sa/sa*dd*	daily data file
/usr/adm/sa/sar*dd*	daily report file
/tmp/sa.adrfl	address file

SEE ALSO

cron(1M), sag(1G), sar(1), timex(1).

NAME

setmnt – establish mount table

SYNOPSIS

/etc/setmnt

DESCRIPTION

Setmnt creates the **/etc/mnttab** table (see *mnttab (4)),* which is needed for both the *mount (1M)* and *umount* commands. *Setmnt* reads standard input and creates a *mnttab* entry for each line. Input lines have the format:

filesys node

where *filesys* is the name of the file system's *special file* (e.g., "dsk/?s?") and *node* is the root name of that file system. Thus *filesys* and *node* become the first two strings in the *mnttab (4)* entry.

FILES

/etc/mnttab

SEE ALSO

mount(1M), mnttab(4).

BUGS

Evil things will happen if *filesys* or *node* are longer than 32 characters.
Setmnt silently enforces an upper limit on the maximum number of *mnttab* entries.

NAME

shutdown – terminate all processing

SYNOPSIS

/etc/shutdown [seconds]

DESCRIPTION

Shutdown is part of the UNIX system operation procedures. Its primary function is to terminate all currently running processes in an orderly and cautious manner. *Seconds* is the number of seconds the system delays between the shutdown warning and the beginning of the shutdown procedure. The procedure is designed to interact with the operator (i.e., the person who invoked *shutdown). Shutdown* may instruct the operator to perform some specific tasks, or to supply certain responses before execution can resume. *Shutdown* goes through the following steps:

All users logged on the system are notified to log off the system by a broadcasted message. The operator may display his/her own message at this time. Otherwise, the standard file-save message is displayed. Default time before system shuts down is 60 seconds.

If the operator wishes to run the file-save procedure, *shutdown* unmounts all file systems.

All file systems' super blocks are updated before the system is to be stopped (see *sync (1)).* This must be done before re-booting the system, to insure file system integrity. The most common error diagnostic that will occur is *device busy* . This diagnostic happens when a particular file system could not be unmounted.

SEE ALSO

mount(1M), sync(1).

NAME

/etc/sys – System control and status program.

SYNOPSIS

/etc/sys command

DESCRIPTION

Sys performs system control functions as well as returning system status. The following commands are recognized by *sys.*

Commands that return a value. (No other action is taken.)

stat Prints the value (in decimal) of the status port on the common circuits board. (P/75 only.)

warm Returns zero (true) if the ambient air temperature sensor is tripped. (P/75 only.)

hot Returns zero (true) if any of the exhaust air temperature sensors are tripped. (P/75 only.)

ups1 Returns zero (true) if the uninterruptible power supply line one is active. (P/75 only.)

ups2 Returns zero (true) if the uninterruptible power supply line two is active. (P/75 only.)

keyoff Returns zero (true) if the system keyswitch is turned off. (P/75 only.)

autoboot Returns zero (true) if the autoboot switch is set on the CPU board.

switches Prints the value (in decimal) of the CPU board switch settings.

initstate Prints the value (in decimal) of the current init state. (Single-user = 0.)

The following commands perform a specific action. Super-user privileges are required.

safeon Turn on the "safe" mode. All buffers are written to disk when updated.

safeoff Turn off the "safe" mode. Normal operation of buffers.

poweroff Power off the system for the time specified in the common circuits board switch settings. (Shutdown is performed gracefully, using the same facilities as the "keyswitch off" sequence.) (P/75 only.)

reset Immediately resets the system. (Selftest is executed.) Note that this is not a "graceful" function; *Sync* should be typed before invoking this function.

off When the system is performing a "graceful shutdown" due to a "keyswitch off" or *sys poweroff* command, this command will remove power from the system. (P/75 only.)

debug Call the system debugger. (Execution of UNIX is halted.)

NOTE

/etc/sys accesses the file /etc/ccb when it is invoked. */etc/ccb* contains a TZ variable which you should set to your time zone as you did in */etc/profile*.

BUGS

P/75 only commands executed on any other system will be silently ignored.

NAME

tic – terminfo compiler

SYNOPSIS

tic [**–v** [*n*]] file ...

DESCRIPTION

Tic translates terminfo files from the source format into the compiled format. The results are placed in the directory **/usr/lib/terminfo**.

The **–v** (verbose) option causes *tic* to output trace information showing its progress. If the optional integer is appended, the level of verbosity can be increased.

Tic compiles all terminfo descriptions in the given files. When a **use**= field is discovered, *tic* searches first the current file, then the master file, which is "./terminfo.src".

If the environment variable TERMINFO is set, the results are placed there instead of **/usr/lib/terminfo**.

Some limitations: total compiled entries cannot exceed 4096 bytes. The name field cannot exceed 128 bytes.

FILES

/usr/lib/terminfo/*/* compiled terminal capability data base

SEE ALSO

curses(3X), terminfo(4).

BUGS

Instead of searching **./terminfo.src**, it should check for an existing compiled entry.

NAME

topq – prioritize print queue

SYNOPSIS

/usr/lib/topq [id]

DESCRIPTION

Topq places the request whose identification number is *id* at the top of the print queue, whether or not **lpsched** is running. Only super-user can use **topq**.

SEE ALSO

enable(1), lp(1), lphold(1), lprun(1), lpstat(1),
accept(1m), lpadmin(1m), lpsched(1m) in the *Sys5 UNIX Administrator's Reference Manual.*

NAME

uucico – file transport program for the uucp system

SYNOPSIS

/usr/lib/uucp/uucico [**–r** role_number] [**–x** debug_level] **–s** system_name

DESCRIPTION

Uucico is the file transport program for *uucp* work file transfers. Role numbers for the **–r** are the digit 1 for master mode or 0 for slave mode (default). The **–r** option should be specified as the digit 1 for master mode when *uucico* is started by a program or *cron*. *Uux* and *uucp* both queue jobs that will be transferred by *uucico*. *Uucico* is usually started by *uucp*, but it can be done manually for debugging purposes. A single digit must be used for the **–x** option, with higher numbers for more debugging, and the mode number must be 1.

FILES

/usr/lib/uucp/L.sys
/usr/lib/uucp/L-devices
/usr/spool/uucp/*
/usr/spool/uucppublic/*

SEE ALSO

cron(1M).
uucp(1C), uustat(1C), uux(1C) in *Sys5 UNIX User's Reference Manual*.

NAME

uuclean – uucp spool directory clean-up

SYNOPSIS

/usr/lib/uucp/uuclean [options]

DESCRIPTION

Uuclean will scan the spool directory for files with the specified prefix and delete all those which are older than the specified number of hours.

The following options are available.

–d*directory* Clean *directory* instead of the spool directory. If *directory* is not a valid spool directory it cannot contain "work files" i.e., files whose names start with "C.". These files have special meaning to *uuclean* pertaining to *uucp* job statistics.

–p*pre* Scan for files with *pre* as the file prefix. Up to 10 **–p** arguments may be specified. A **–p** without any *pre* following will cause all files older than the specified time to be deleted.

–n*time* Files whose age is more than *time* hours will be deleted if the prefix test is satisfied. (default time is 72 hours)

–w*file* The default action for *uuclean* is to remove files which are older than a specified time (see **–n** option). The **–w** option is used to find those files older than *time* hours, however, the files are not deleted. If the argument *file* is present the warning is placed in *file* , otherwise, the warnings will go to the standard output.

–s*sys* Only files destined for system *sys* are examined. Up to 10 **–s** arguments may be specified.

–m*file* The **–m** option sends mail to the owner of the file when it is deleted. If a *file* is specified then an entry is placed in *file.*

uuclean is also used in *uucp* daemon shell scripts: *uudemon.hr* completes jobs waiting on the local system and merges status reports into the log file; *uudemon.day* cleans the spool directory and merges daily log files with weekly log files; *uudemon.wk* maintains the weekly log and removes files older than two weeks.

This program is typically started by *cron (1M).*

FILES

/usr/lib/uucp	directory with commands used internally by *uuclean*
/usr/spool/uucp	spool directory

SEE ALSO

cron(1M), uucp(1C), uux(1C).

NAME

uusub – monitor uucp network

SYNOPSIS

/usr/lib/uucp/uusub [options]

DESCRIPTION

Uusub(1M) defines a *uucp* subnetwork and monitors the connection and traffic among the members of the subnetwork. The following options are available:

–a*sys*	Add *sys* to the subnetwork.
–d*sys*	Delete *sys* from the subnetwork.
–l	Report the statistics on connections.
–r	Report the statistics on traffic amount.
–f	Flush the connection statistics.
–u*hr*	Gather the traffic statistics over the past *hr* hours.
–c*sys*	Exercise the connection to the system *sys* . If *sys* is specified as **all** , then exercise the connection to all the systems in the subnetwork.

The meanings of the connections report are:

sys #call #ok time #dev #login #nack #other

where *sys* is the remote system name, *#call* is the number of times the local system tries to call *sys* since the last flush was done, and *#ok* is the number of successful connections, *time* is the latest successful connect time, *#dev* is the number of unsuccessful connections because of no available device (e.g., ACU), *#login* is the number of unsuccessful connections because of login failure, *#nack* is the number of unsuccessful connections because of no response (e.g. line busy, system down), and *#other* is the number of unsuccessful connections because of other reasons.

The meanings of the traffic statistics are:

sfile sbyte rfile rbyte

where *sfile* is the number of files sent and *sbyte* is the number of bytes sent over the period of time indicated in the latest *uusub* command with the **–u***hr* option. Similarly, *rfile* and *rbyte* are the numbers of files and bytes received.

The command:

uusub –c all –u 24

is typically started by *cron (1M)* once a day.

NAME

volcopy, labelit – copy file systems with label checking

SYNOPSIS

/etc/volcopy [options] fsname special1 volname1 special2 volname2

/etc/labelit special [fsname volume [**–n**]]

DESCRIPTION

Volcopy makes a literal copy of the file system using a blocksize matched to the device. *Options* are:

–a invoke a verification sequence requiring a positive operator response instead of the standard 10-second delay before the copy is made

–s (default) invoke the **DEL if wrong** verification sequence.

Other *options* are used only with tapes:

–bpi density bits-per-inch (i.e., **800 / 1600 / 6250),**

–feet size size of reel in feet (i.e., **1200 / 2400),**

–reel num beginning reel number for a restarted copy,

–buf use double buffered I/O.

The program requests length and density information if it is not given on the command line or is not recorded on an input tape label. If the file system is too large to fit on one reel, *volcopy* will prompt for additional reels. Labels of all reels are checked. Tapes may be mounted alternately on two or more drives. If *volcopy* is interrupted, it will ask if the user wants to quit or wants a shell. In the latter case, the user can perform other operations (e.g.,: *labelit)* and return to *volcopy* by exiting the new shell.

The *fsname* argument represents the mounted name (e.g.,: **root** , **u1** , etc.) of the filsystem being copied.

The *special* should be the physical disk section or tape (e.g.,: **/dev/rdsk/1s5** , **/dev/rmt/0m** , etc.).

The *volname* is the physical volume name (e.g.,: **pk3** , **t0122** , etc.) and should match the external label sticker. Such label names are limited to six or fewer characters. *Volname* may be – to use the existing volume name.

Special1 and *volname1* are the device and volume from which the copy of the file system is being extracted. *Special2* and *volname2* are the target device and volume.

Fsname and *volname* are recorded in the last 12 characters of the superblock (**char fsname[6], volname[6];**).

Labelit can be used to provide initial labels for unmounted disk or tape file systems. With the optional arguments omitted, *labelit* prints current label values. The **–n** option provides for initial labeling of new tapes only (this destroys previous contents).

FILES

/etc/log/filesave.log   a record of file systems/volumes copied

SEE ALSO

sh(1), fs(4).

BUGS

Only device names beginning **/dev/rmt/** are treated as tapes.

NAME

wall – write to all users

SYNOPSIS

/etc/wall

DESCRIPTION

Wall reads its standard input until an end-of-file. It then sends this message to all currently logged-in users preceded by:

Broadcast Message from ...

It is used to warn all users, typically prior to shutting down the system.

The sender must be super-user to override any protections the users may have invoked (see *mesg (1)).*

FILES

/dev/tty*

SEE ALSO

mesg(1), write(1).

DIAGNOSTICS

"Cannot send to ..." when the open on a user's tty file fails.

NAME

whodo – who is doing what

SYNOPSIS

/etc/whodo

DESCRIPTION

Whodo produces merged, reformatted, and dated output from the *who (1)* and *ps (1)* commands.

FILES

etc/passwd

SEE ALSO

ps(1), who(1).

NAME

intro – introduction to special files

DESCRIPTION

This section describes various special files that refer to specific hardware peripherals and UNIX system device drivers. The names of the entries are generally derived from names for the hardware, as opposed to the names of the special files themselves. Characteristics of both the hardware device and the corresponding UNIX system device driver are discussed where applicable.

Tape device file names are in the following format:

/dev/{r}mt/(c#d)#[hml]{n}

where **r** indicates a raw device, **c#d** indicates the controller number (which is optionally specified by the system administrator), **#** is the device number, **hml** indicates the density (**h** (high) for 6250 bpi, **m** (medium) for 1600 bpi, and **l** (low density) for 800 bpi), and **n** indicates no rewind on close. (e.g., **/dev/mt/2mn**)

Disk device file names are in the following format:

/dev/{r}dsk/(r)(c#d)#s#

where **r** indicates a raw interface to the disk, the second **r** indicates that this disk is on a remote system, the **c#d** indicates the controller number (which is optionally specified by the system administrator), and **#s#** indicates the drive and section numbers, respectively.

BUGS

While the names of the entries *generally* refer to vendor hardware names, in certain cases these names are seemingly arbitrary for various historical reasons.

NAME

err – error-logging interface

DESCRIPTION

Minor device 0 of the *err* driver is the interface between a process and the system's error-record collection routines. The driver may be opened only for reading by a single process with super-user permissions. Each read causes an entire error record to be retrieved; the record is truncated if the read request is for less than the record's length.

FILES

/dev/error special file

SEE ALSO

errdemon(1M).

NAME

ft – IMSP streaming cartridge controller

DESCRIPTION

This is a pseudo driver which will stream I/O between a cartridge tape drive and an IMSP-controlled disk. It uses a disk partition (logical disk) as a scratch buffer area. The disk partition is a small (1-2 megabyte) logical disk created using *dconfig*. It must not overlap a currently active file system. See Section 1M of this manual and the *UNIX Sys5 Administrator's Guide* for more information on *dconfig*.

By convention, the files **/dev/rft/0m** and **/dev/rft/0mn** are used to access the cartridge in streaming mode. Accessing with **/dev/rft/0m** rewinds the cartridge when this special file is closed. Accessing with **/dev/rft/0mn** does not rewind the cartridge when the file is closed.

The reads and writes take place asynchronously, occurring when the buffer area is filled. Therefore, errors which occur might not be reported until the tape device is closed. You must be careful not to attempt to write more to the tape cartridge than it can hold. Errors reported might relate either to the disk or the tape. Tape errors are described in *pt*(7), disk errors in *pd*(7).

The major number for these files is 25. The minor number for **/dev/rft/0m** is the same as the number of the disk partition you are using for the scratch buffer. The minor number for **/dev/rft/0mn** is gotten by adding 128 to the minor number of **dev/rft/0m**. For example, if you are using **/dev/dsk/0s15** for your scratch buffer area, the relevant minor number is 15. To create nodes for the two *ft* devices use the following commands:

mknod /dev/rft/0m c 25 15
mknod /dev/rft/0mn c 25 143 (128 + 15)

FILES

/dev/rft/0m
/dev/rft/0mn
/dev/dsk/?s?

WARNING

Be very careful that the disk partition used for the scratch buffer does not overlap an active file system. File system corruption might occur if it does.

Do not attempt to write more than a cartridge can hold.

This device will not work with disks controlled by an EMSP.

SEE ALSO

dconfig(1M), mknod(1M), pd(7), pt(7).

NAME

icp – Intelligent Communications Processor

DESCRIPTION

The *icp* is a special device that allows access to the memory of the Intelligent Communications Processor (ICP). Reading from the device resets the ICP. Writing to the device overwrites the memory.

FILES

/dev/ic[0-4]

BUGS

Reading from the ICP resets it and kills all terminals actively using it.

SEE ALSO

dnld(1m), icpdmp(1m).

NAME

mem, kmem – core memory
mbiomem, mbmem – Multibus memory
liomem – local I/O device memory

DESCRIPTION

Mem is a special file that is an image of the core memory of the computer. It may be used, for example, to examine, and even to patch the system.

Byte addresses in *mem* are interpreted as memory addresses. References to non-existent locations cause errors to be returned.

Examining and patching device registers is likely to lead to unexpected results when read-only or write-only bits are present.

The file *kmem* is the same as *mem* except that kernel virtual memory rather than physical memory is accessed.

Mbiomem is a special file that is an image of the Multibus I/O address space.

Mbmem is a special file that is an image of the Multibus memory address space.

Liomem is a special file that ia an image of the local I/O device address space. This can be used, for example, to reference the clock chip or the SIO chip.

FILES

/dev/mem
/dev/kmem
/dev/mbiomem
/dev/mbmem
/dev/liomem

NAME

mv – a macro package for making view graphs

SYNOPSIS

mvt [options] [files]

troff −mv [options] [files]

DESCRIPTION

This package provides an easy-to-use facility for making view graphs and projection slides in a variety of formats. A dozen or so macros are provided that accomplish most of the formatting tasks needed in making transparencies. All of the facilities of *troff*(1), *eqn*(1), and *tbl*(1) are available for more difficult tasks. The output can be previewed on most terminals, and, in particular, on the Tektronix 4014 and on the Versatec printer. See the reference below for further details.

FILES

/usr/lib/tmac/tmac.v

SEE ALSO

eqn(1), mvt(1), tbl(1), troff(1).

A Macro Package for View Graphs and Slides by T. A. Dolotta and D. W. Smith (in preparation).

NAME

null – the null file

DESCRIPTION

Data written on a null special file is discarded.

Reads from a null special file always return 0 bytes.

FILES

/dev/null

NAME

pp – parallel port interface

DESCRIPTION

The parallel port interface enables access to the parallel port on the Intelligent Communications Processor (ICP). Each ICP has one parallel port interface. The parallel port interface is a write-only device. It is also a raw device, i.e., the operating system does no processing of data written to it.

Pp has no *stty*-like features. If your printer does not handle tabs and new-line characters, you need to write a filter to use this device.

FILES

/dev/pp[0-3]

SEE ALSO

tty(7), icp(7)

NAME

prf – operating system profiler

DESCRIPTION

The file provides access to activity information in the operating system. Writing the file loads the measurement facility with text addresses to be monitored. Reading the file returns these addresses and a set of counters indicative of activity between adjacent text addresses.

The recording mechanism is driven by the system clock and samples the program counter at line frequency. Samples that catch the operating system are matched against the stored text addresses and increment corresponding counters for later processing.

The file is a pseudo-device with no associated hardware.

FILES

/dev/prf

SEE ALSO

profiler(1M).

NAME

pt – IMSP cartridge controller

DESCRIPTION

The IMSP disk/tape controller and associated driver code allow access to a cartridge tape. The cartridge can be accessed only in raw mode (i.e., as a character device), and can be rewound or left at the current position. These options are available based on the minor device number of the special file used to access it. If the cartridge is not to be rewound, it is positioned after the filemark at the end of the current file.

If the 04 bit is on in the minor device number, the cartridge is not rewound when closed.

By convention, the files **/dev/rpt/0m** and **/dev/rpt/0mn** are used to access the cartridge in raw mode. Accessing **/dev/rpt/0m** rewinds the cartridge when this special file is closed. Accessing **/dev/rpt/0mn** does not rewind the cartridge when the file is closed. Each *read* or *write* call reads or writes the next record on the cartridge. All records on a cartridge are 512 bytes long and all reads and writes must be in multiples of 512 bytes. An error is returned otherwise. The I/O buffer used in the *read*(2) or *write*(2) system call should begin on a word boundary and the count should be even. Seeks are ignored. A zero byte count is returned when a file mark is read, but another read will fetch the first record of the new file.

The cartridge drive can be accessed in high speed mode. However, this mode is effectively limited to skipping forward over files on the cartridge and to I/O between the cartridge and a disk attached to the same IMSP controller. High speed mode is accessed via *ioctl*(2) system calls. The arguments to the *ioctl* are:

fildes File descriptor returned from an *open*(2) of the special tape file **/dev/rpt/0m** or **/dev/rpt/0mn**.

request A special command for the cartridge drive. These commands are defined in **/usr/include/sys/imsc.h** and some are described below.

arg A pointer to a structure of the type "ptcmd" as defined in **/usr/include/sys/imsc.h**.

Some of the members of **ptcmd** are:

dknum Major/minor device number of the IMSP disk being read or written to (if applicable) as returned by *stat*(2) system call (st_rdev).

blkno Starting sector number on logical disk to be read/written. Sectors on disk are 512 bytes long and numbered starting at 0. Note sector addresses are relative to the logical, not the physical disk.

blkcnt The number of 512-byte records to be read from or written to cartridge.

Some of the more useful *ioctl* requests for the cartridge as defined in **/usr/include/sys/imsc.h** are:

C_IRECALL Read from cartridge and write to disk. The cartridge and disk must be on same IMSP controller. The system returns in **ptcmd.blkcnt** the number of 512-byte records not read. This is zero if the system reads all the records requested.

C_ISAVE Read from disk and write to tape. The cartridge and disk must be on same IMSP controller. The system returns in **ptcmd.blkcnt** the number of 512-byte records not read. This is zero if the system reads all the record images (sectors) requested.

C_IWEOF Write EOF mark on cartridge.

C_IREW Rewinds the cartridge.

C_IMOVE Position to file **blkcnt** on cartridge.

Writing multiple files on cartridge should be done all at once, i.e., without rewinding the cartridge. Once a cartridge has been rewound, positioning to the end of a file on the cartridge and then writing to the cartridge may overwrite data. For example, once the cartridge has been rewound, positioning to the end of file 2 and writing to the cartridge may overwrite portions of file 2.

Neither the hardware or the software implement or support an end-of-tape marker on the cartridge.

FILES

/dev/rpt/0m
/dev/rpt/0mn

DIAGNOSTICS

The IMSP controller produces error diagnostics in the following form:

> sys3: error on PT, minor 0
> sys3: bn = *bbbb* er = 0x*nnnn*, 0x*mmmm*

where *bbbb* is a block number. The first set of "er" numbers (*nnnn*) gives status. The second set of "er" numbers (*mmmm*) describes errors. Each set of "er" numbers is discussed separately below.

Status Bytes

There are two meaningful bytes of status (*nnnn*); these are the third and fourth bytes of a 32 bit word. Because the status representation is "zero-true", if the third byte is all ones, the system construes the entire word as a negative number and prepends "ffff" to the two status bytes. This leading "ffff" can be ignored.

The meaning of each bit of the status bytes is listed below. Examples follow.

Byte 0

Bit 7	Status byte 0 contains information
Bit 6	Cartridge not in place
Bit 5	Drive not online
Bit 4	Write protected
Bit 3	End of media
Bit 2	Unrecoverable data error
Bit 1	BIE not located
Bit 0	File mark detected

Byte 1

Bit 7	Status byte 1 contains information
Bit 6	Illegal command
Bit 5	No data detected
Bit 4	8 or more read retries
Bit 3	Beginning of media
Bit 2	Reserved
Bit 1	Reserved
Bit 0	Reset/Power-up occurred

For example, the error

```
sys3: error on PT, minor 0
sys3: bn = 2345 er = 0x7b77, 0x1604
```

shows two bytes of status. The first byte is "7b", which means (remember zero indicates true) status byte 0 is meaningful and unrecoverable data error. The second byte is "77", which means status byte 1 is meaningful and beginning of media.

The second "er" number (0x1604) is described below.

The error

```
sys3: error on PT, minor 0
sys3: bn = bbbb er = 0xffffff76, 0x1604
```

shows the first byte of status to be "ff" (status byte 0 contains no information). The second byte is "76", which means status byte 1 contains information, and illegal command. The first four "f's" result from the system construing the status word as negative; they can be ignored.

Error Bytes

The following list shows the possible values for the error status (the second "er" number, or *mmmm* above):

0x0201	Reserved for controller busy
0x0301	Command undefined
0x0401	Command cannot be done
0x0501	Bad CAB parameters
0x0f01	Firmware bug encountered
0x0601	Internal command interrupts
0x0701	Parity error occurred
0x0801	PROM checksum error
0x1004	End of file reached
0x1304	An exception other than an end-of-file error
0x1504	Tape timeout error
0x1604	Error during recall
0x1704	Error during save
0x1804	Error received while attempting to get status from the tape drive
0x1904	During exception state, a command other than *rstat* was received
0x2004	No tape drive present
0x2104	Timeout during wait recall
0x2204	Timeout during wait save
0x2304	Timeout during stat tape
0x2404	Timeout during stat tape
0x2504	Timeout during command tape
0x2604	Timeout during command tape
0x2704	Timeout during ready tape
0x2804	Tape drive inconsistent at start of tape command
0x1505	Timeout on Host bus request

NAME

rm – Cipher Microstreamer tape drive

DESCRIPTION

The Cipher Microstreamer magnetic tape can be accessed in blocked or raw mode and can be rewound or left at the current position. These options are available based on the minor device number of the special file used to access it. When the special file is closed, the tape can be rewound or not (see below). If the special file was open for writing, two end-of-files are written. If the tape is not to be rewound, it is positioned with the head between the two tapemarks.

If the 04 bit is on in the minor device number, the tape is not rewound when closed.

If the 010 bit is on in the minor device number, the tape is set to high speed mode (100 in/sec). By convention, **/dev/nrrmh0** accesses the tape in high speed mode.

By convention, the file **/dev/mt0** accesses the tape in blocked mode. A tape accessed in block mode consists of a series of 1024-byte records terminated by an end-of-file. As much as it can, the system makes it possible, if inefficient, to treat the tape like any other file. Seeks have their usual meaning and it is possible to read or write a byte at a time. Writing in very small units is inadvisable, however, because it tends to create monstrous record gaps.

Use **/dev/mt0** to access the tape in a way compatible with ordinary files. However, when foreign tapes are to be dealt with, and especially when long records are to be read or written, the 'raw' interface is more appropriate. By convention, the files **/dev/rpt/0m** and **/dev/rpt/0mn** are used to access the tape in raw mode. Accessing **/dev/rpt/0m** rewinds the tape when **/dev/rpt/0mn** is closed. Accessing **/dev/rpt/0mn** does not rewind the tape when **/dev/rpt/0mn** is closed.

Each *read* or *write* call reads or writes the next record on the tape. For writes, the record has the same length as the buffer given. During a read, the record size is passed back as the number of bytes read, provided it is no greater than the number of bytes requested; if the record is longer than the number of bytes requested, an error is returned. On the other hand, if the number of bytes requested is larger than the actual record size, there is a delay of 1-2 seconds between the reading of each record.

In raw tape I/O, the buffer must begin on a word boundary and the count must be even. Seeks are ignored. A zero byte count is returned when a tape mark is read, but another read will fetch the first record of the new tape file.

The tape drive can be run in high speed mode; however, this is really only usable for fast forward or reverse skipping of file marks. The files used for high speed mode are denoted by an 'h' just before the unit number.

If you want to write your own program for tape manipulation on the *rm* device, there is an **ioctl**(2) interface for controlling the tape drive. The file */usr/include/sys/rm.h* lists the commands that can be issued. These all begin with "C_" (capital C followed by an underbar). The only **ioctl** request type allowed for this device is RMPOSN ("*rm* position"). The **ioctl** call structure is

```
struct rmcmd_struct {
  unsigned rm_cmd;      /* the command C_<option> */
  unsigned rm_cnt;      /* count, useful for commands
                           such as SRCHEOF */
  unsigned rm_status;   /* physical device status returned */
  };
```

The status value is found by adding all the relevant values in the "status fields" portion of *rm.h*. Status is determined by the output status field, which consists of two bytes arranged as follows:

15	14	13	12 11 10 9 8	7	6	5	4	3	2	1
E	C	R	ERROR	FM	OL	LP	EOT	R	FB	P

where

Byte 0	Not used
P	(Write Protect) The tape does not have a write enable ring.
FB	(Formatter Busy) The Formatter is busy.
R	(Ready) The selected drive is ready.
EOT	(End of Tape) The EOT marker was detected.
LP	(Load Point) The tape is at load point.
OL	(On Line) The drive is on line.
FM	(Filemark) A filemark was detected on this operation
E	(Entered) Execution has begun.
C	(Complete) The command has completed successfully.
R	(Retry) At least one Retry was executed .
ERROR	This 5-bit field specifies an error code when a non-recoverable error is encountered. Error codes are listed under DIAGNOSTICS below.

For example, the value "C068" means the tape is online at load point, ready, and previous command has completed.

The following program fragment illustrates the use of **ioctl** to rewind the tape.

```
#include "sys/rm.h"
#include "fcntl.h"
int fildes;    /* file descriptor, returned by open */

fildes = open("/dev/rpt/0m",O_RDWR);

rmcmd.cmd = C_REW;
rmcmd.cnt = 1;
rmcmd.status = -1;

ioctl(fildes, RMPOSN, &rmcmd);
```

FILES

/dev/mt0
/dev/rpt/0m
/dev/rpt/0mn
/dev/nrrmh0
/usr/include/sys/rm.h

SEE ALSO

tape(1), ioctl(2).

DIAGNOSTICS

The tape controller issues the following codes for unrecoverable errors detected during execution of a command. The code is returned in the Command Status byte, bits 8-12.

Code	Description
00	No unrecoverable error.
01	Timed out waiting for expected Data Busy false.
02	Timed out waiting for expected Data Busy false, Formatter Busy false and Ready True.
03	Timed out waiting for expected Ready false.
04	Timed out waiting for expected Ready true.
05	Timed out waiting for expected Data Busy true.
06	A memory time-out occurred during a system memory reference.
07	A blank tape was encountered where data was expected.
08	An error occurred in the micro-diagnostic.
09	An unexpected EOT was encountered during a forward operation, or Load Point during a reverse operation.

0A A hard or soft error occurred that could not be eliminated by retry.

0B A read overflow or write overflow occurred. This error indicates that the FIFO was empty when data was requested by the tape during a write, or full when the tape presented a byte during a read.

0C Not used.

0D A read parity error occurred on the byte interface between the drive and the controller.

0E An error was detected during calculation of the checksum on the PROM.

0F A tape time-out occurred, because the tape drive did not supply an expected read or write strobe. This error occurs when you attempt to read a larger record than was written. It may also occur during a write if the tape is damaged.

10 Tape not ready.

11 A write was attempted on a tape without a write-enable ring.

12 Not used.

13 The diagnostic mode jumper was not installed while attempting to execute a Diagnostic command.

14 An attempt was made to link from a command that does not allow linking.

15 An unexpected filemark was encountered during a tape read.

16 An error in specifying a parameter was detected by the controller. The usual cause is a byte count that is either zero or too large.

17 Not used.

18 An unidentifiable hardware error occurred.

19 A streaming read or write operation was terminated by the operating system or disk.

The tape driver sends the code FFFF to the screen when the block size requested is smaller than the actual block size on the tape.

NAME

rram, ram – allows memory to be used as a disk.

DESCRIPTION

Allocates memory to one of up to eight devices which allow the memory to be used as a disk.

The directory **/dev/rram** contains the raw devices used as parameters to the utility, and **/dev/ram** contains block devices which can be configured to be mounted file systems.

Nodes in */dev/rram* are character type (**c**) devices. The major number to use is 7. Minor numbers range from 0 to 7.

Nodes in */dev/ram* are block type (**b**) devices. The major number is 3.

Blocks from these devices do not stay in the buffer pool. Their buffers are reused immediately to allow the buffer pool to be used by disk devices.

Usage is as follows:

mknod /dev/ram/*devname* **b 3** *devnumber*

mknod /dev/rram/*devname* **c 7** *devnumber*

FILES

/dev/ram/*devname*
/dev/rram/*devname*

SEE ALSO

mknod(1m), ramdisk(1m).

NAME

tty – general terminal interface

DESCRIPTION

This section describes both a particular special file and the general nature of the terminal interface.

The file **/dev/tty** is, in each process, a synonym for the control terminal associated with the process group of that process, if any. It is useful for programs or shell sequences that wish to be sure of writing messages on the terminal no matter how output has been redirected. It can also be used for programs that demand the name of a file for output, when typed output is desired and it is tiresome to find out what terminal is currently in use.

As for terminals in general: all of the asynchronous communications ports use the same general interface, no matter what hardware is involved. The remainder of this section discusses the common features of this interface.

When a terminal file is opened, it normally causes the process to wait until a connection is established. In practice, users' programs seldom open these files; they are opened by *getty*(8) and become a user's standard input, output, and error files. The very first terminal file opened by the process group leader of a terminal file not already associated with a process group becomes the *control terminal* for that process group. The control terminal plays a special role in handling quit and interrupt signals, as discussed below. The control terminal is inherited by a child process during a *fork*(2). A process can break this association by changing its process group using *setpgrp*(2).

A terminal associated with one of these files ordinarily operates in full-duplex mode. Characters may be typed at any time, even while output is occurring, and are only lost when the system's character input buffers become completely full, which is rare, or when the user has accumulated the maximum allowed number of input characters that have not yet been read by some program. Currently, this limit is 512 characters. When the input limit is reached, all the saved characters are thrown away without notice.

Normally, terminal input is processed in units of lines. A line is delimited by a new-line (ASCII LF) character, an end-of-file (ASCII EOT) character, or an end-of-line character. This means that a program attempting to read will be suspended until an entire line has been typed. Also, no matter how many characters are requested in the read call, at most one line will be returned. It is not, however, necessary to read a whole line at once; any number of characters may be requested in a read, even one, without losing information.

During input, erase and kill processing is normally done. By default, the character # erases the last character typed, except that it will not erase beyond the beginning of the line. By default, the character @ kills (deletes) the entire input line, and optionally outputs a new-line character. Both these characters operate on a key-stroke basis, independently of any backspacing or tabbing that may have been done. Both the erase and kill characters may be entered literally by preceding them with the escape character (\). In this case the escape character is not read. The erase and kill characters may be changed.

Certain characters have special functions on input. These functions and their default character values are summarized as follows:

INTR (Rubout or ASCII DEL) generates an *interrupt* signal which is sent to all processes with the associated control terminal. Normally, each such process is forced to terminate, but arrangements may be made either to ignore the signal or to receive a trap to an agreed-upon location; see *signal*(2).

QUIT (Control-| or ASCII FS) generates a *quit* signal. Its treatment is identical to the interrupt signal except that, unless a receiving process has made other arrangements, it will not only be terminated but a core image file (called **core**) will be created in the current working directory.

ERASE (#) erases the preceding character. It will not erase beyond the start of a line, as delimited by a NL, EOF, or EOL character.

KILL (@) deletes the entire line, as delimited by a NL, EOF, or EOL character.

EOF (Control-d or ASCII EOT) may be used to generate an end-of-file from a terminal. When received, all the characters waiting to be read are immediately passed to the program, without waiting for a new-line, and the EOF is discarded. Thus, if there are no characters waiting, which is to say the EOF occurred at the beginning of a line, zero characters will be passed back, which is the standard end-of-file indication.

NL (ASCII LF) is the normal line delimiter. It can not be changed or escaped.

EOL (ASCII NUL) is an additional line delimiter, like NL. It is not normally used.

STOP (Control-s or ASCII DC3) can be used to temporarily suspend output. It is useful with CRT terminals to prevent output from disappearing before it can be read. While output is suspended, STOP characters are ignored and not read.

START (Control-q or ASCII DC1) is used to resume output which has been suspended by a STOP character. While output is not suspended, START characters are ignored and not read. The start/stop characters can not be changed or escaped.

The character values for INTR, QUIT, ERASE, KILL, EOF, and EOL may be changed to suit individual tastes. The ERASE, KILL, and EOF characters may be escaped by a preceding \ character, in which case no special function is done.

When the carrier signal from the data-set drops, a *hangup* signal is sent to all processes that have this terminal as the control terminal. Unless other arrangements have been made, this signal causes the processes to terminate. If the hangup signal is ignored, any subsequent read returns with an end-of-file indication. Thus programs that read a terminal and test for end-of-file can terminate appropriately when hung up on.

When one or more characters are written, they are transmitted to the terminal as soon as previously-written characters have finished typing. Input characters are echoed by putting them in the output queue as they arrive. If a process produces characters more rapidly than they can be typed, it will be suspended when its output queue exceeds some limit. When the queue has drained down to some threshold, the program is resumed.

Several *ioctl*(2) system calls apply to terminal files. The primary calls use the following structure, defined in <**termio.h**>:

```
#define  NCC       8
struct   termio {
         unsigned  short  c_iflag;     /* input modes */
         unsigned  short  c_oflag;     /* output modes */
         unsigned  short  c_cflag;     /* control modes */
         unsigned  short  c_lflag;     /* local modes */
         char             c_line;      /* line discipline */
         unsigned  char   c_cc[NCC];   /* control chars */
};
```

The special control characters are defined by the array *c_cc*. The relative positions and initial values for each function are as follows:

0	INTR	DEL
1	QUIT	FS
2	ERASE	#
3	KILL	@
4	EOF	EOT
5	EOL	NUL

6 reserved
7 reserved

The *c_iflag* field describes the basic terminal input control:

IGNBRK	0000001	Ignore break condition.
BRKINT	0000002	Signal interrupt on break.
IGNPAR	0000004	Ignore characters with parity errors.
PARMRK	0000010	Mark parity errors.
INPCK	0000020	Enable input parity check.
ISTRIP	0000040	Strip character.
INLCR	0000100	Map NL to CR on input.
IGNCR	0000200	Ignore CR.
ICRNL	0000400	Map CR to NL on input.
IUCLC	0001000	Map upper-case to lower-case on input.
IXON	0002000	Enable start/stop output control.
IXANY	0004000	Enable any character to restart output.
IXOFF	0010000	Enable start/stop input control.

See NOTES below for Plexus additions to this list.

If IGNBRK is set, the break condition (a character framing error with data all zeros) is ignored, that is, not put on the input queue and therefore not read by any process. Otherwise if BRKINT is set, the break condition will generate an interrupt signal and flush both the input and output queues. If IGNPAR is set, characters with other framing and parity errors are ignored.

If PARMRK is set, a character with a framing or parity error which is not ignored is read as the three character sequence: 0377, 0, X, where X is the data of the character received in error. To avoid ambiguity in this case, if ISTRIP is not set, a valid character of 0377 is read as 0377, 0377. If PARMRK is not set, a framing or parity error which is not ignored is read as the character NUL (0).

If INPCK is set, input parity checking is enabled. If INPCK is not set, input parity checking is disabled. This allows output parity generation without input parity errors.

If ISTRIP is set, valid input characters are first stripped to 7-bits, otherwise all 8-bits are processed.

If INLCR is set, a received NL character is translated into a CR character. If IGNCR is set, a received CR character is ignored (not read). Otherwise if ICRNL is set, a received CR character is translated into a NL character.

If IUCLC is set, a received upper-case alphabetic character is translated into the corresponding lower-case character.

If IXON is set, start/stop output control is enabled. A received STOP character will suspend output and a received START character will restart output. All start/stop characters are ignored and not read. If IXANY is set, any input character will restart output that has been suspended. Note that some terminals experience difficulty with IXANY.

If IXOFF is set, the system will transmit START/STOP characters when the input queue is nearly empty/full.

The initial input control value is all bits clear.

The *c_oflag* field specifies the system treatment of output:

OPOST	0000001	Postprocess output.
OLCUC	0000002	Map lower case to upper on output.
ONLCR	0000004	Map NL to CR-NL on output.
OCRNL	0000010	Map CR to NL on output.
ONOCR	0000020	No CR output at column 0.
ONLRET	0000040	NL performs CR function.
OFILL	0000100	Use fill characters for delay.
OFDEL	0000200	Fill is DEL, else NUL.
NLDLY	0000400	Select new-line delays:
NL0	0	
NL1	0000400	
CRDLY	0003000	Select carriage-return delays:
CR0	0	
CR1	0001000	
CR2	0002000	
CR3	0003000	
TABDLY	0014000	Select horizontal-tab delays:
TAB0	0	
TAB1	0004000	
TAB2	0010000	
TAB3	0014000	Expand tabs to spaces.
BSDLY	0020000	Select backspace delays:
BS0	0	
BS1	0020000	
VTDLY	0040000	Select vertical-tab delays:
VT0	0	
VT1	0040000	
FFDLY	0100000	Select form-feed delays:
FF0	0	
FF1	0100000	

If OPOST is set, output characters are post-processed as indicated by the remaining flags, otherwise characters are transmitted without change.

If OLCUC is set, a lower-case alphabetic character is transmitted as the corresponding upper-case character. This function is often used in conjunction with IUCLC.

If ONLCR is set, the NL character is transmitted as the CR-NL character pair. If OCRNL is set, the CR character is transmitted as the NL character. If ONOCR is set, no CR character is transmitted when at column 0 (first position). If ONLRET is set, the NL character is assumed to do the carriage-return function; the column pointer will be set to 0 and the delays specified for CR will be used. Otherwise the NL character is assumed to do just the line-feed function; the column pointer will remain unchanged. The column pointer is also set to 0 if the CR character is actually transmitted.

The delay bits specify how long transmission stops to allow for mechanical or other movement when certain characters are sent to the terminal. In all cases a value of 0 indicates no delay. If OFILL is set, fill characters will be transmitted for delay instead of a timed delay. This is useful for high baud rate terminals which need only a minimal delay. If OFDEL is set, the fill character is DEL, otherwise NUL.

If a form-feed or vertical-tab delay is specified, it lasts for about 2 seconds.

New-line delay lasts about 0.10 seconds. If ONLRET is set, the carriage-return delays are used instead of the new-line delays. If OFILL is set, two fill characters will be transmitted.

Carriage-return delay type 1 is dependent on the current column position, type 2 is about 0.10 seconds, and type 3 is about 0.15 seconds. If OFILL is set, delay type 1 transmits two fill characters, and type 2 four fill characters.

Horizontal-tab delay type 1 is dependent on the current column position. Type 2 is about 0.10 seconds. Type 3 specifies that tabs are to be expanded into spaces. If OFILL is set, two fill characters will be transmitted for any delay.

Backspace delay lasts about 0.05 seconds. If OFILL is set, one fill character will be transmitted.

The actual delays depend on line speed and system load.

The initial output control value is all bits clear.

The *c_cflag* field describes the hardware control of the terminal:

CBAUD	0000017	Baud rate:
B0	0	Hang up
B50	0000001	50 baud
B75	0000002	75 baud
B110	0000003	110 baud
B134	0000004	134.5 baud
B150	0000005	150 baud
B200	0000006	200 baud

B300	0000007	300 baud
B600	0000010	600 baud
B1200	0000011	1200 baud
B1800	0000012	1800 baud
B2400	0000013	2400 baud
B4800	0000014	4800 baud
B9600	0000015	9600 baud
EXTA	0000016	External A (19200 baud)
EXTB	0000017	External B
CSIZE	0000060	Character size:
CS5	0	5 bits
CS6	0000020	6 bits
CS7	0000040	7 bits
CS8	0000060	8 bits
CSTOPB	0000100	Send two stop bits, else one.
CREAD	0000200	Enable receiver.
PARENB	0000400	Parity enable.
PARODD	0001000	Odd parity, else even.
HUPCL	0002000	Hang up on last close.
CLOCAL	0004000	Local line, else dial-up.

The CBAUD bits specify the baud rate. The zero baud rate, B0, is used to hang up the connection. If B0 is specified, the data-terminal-ready signal will not be asserted. Normally, this will disconnect the line. For any particular hardware, impossible speed changes are ignored.

The baud rate for EXTB is determined from switch settings in the hardware. See the *Plexus User's Manual* for details.

The CSIZE bits specify the character size in bits for both transmission and reception. This size does not include the parity bit, if any. If CSTOPB is set, two stop bits are used, otherwise one stop bit. For example, at 110 baud, two stops bits are required.

If PARENB is set, parity generation and detection is enabled and a parity bit is added to each character. If parity is enabled, the PARODD flag specifies odd parity if set, otherwise even parity is used.

If CREAD is set, the receiver is enabled. Otherwise no characters will be received.

If HUPCL is set, the line will be disconnected when the last process with the line open closes it or terminates. That is, the data-terminal-ready signal will not be asserted.

If CLOCAL is set, the line is assumed to be a local, direct connection with no modem control. Otherwise modem control is assumed.

The initial hardware control value after open is B300, CS8, CREAD, HUPCL.

The *c_lflag* field of the argument structure is used by the line discipline to control terminal functions. The basic line discipline (0) provides the following:

ISIG	0000001	Enable signals.
ICANON	0000002	Canonical input (erase and kill processing).
XCASE	0000004	Canonical upper/lower presentation.
ECHO	0000010	Enable echo.
ECHOE	0000020	Echo erase character as BS-SP-BS.
ECHOK	0000040	Echo NL after kill character.
ECHONL	0000100	Echo NL.
NOFLSH	0000200	Disable flush after interrupt or quit.

If ISIG is set, each input character is checked against the special control characters INTR and QUIT. If an input character matches one of these control characters, the function associated with that character is performed. If ISIG is not set, no checking is done. Thus these special input functions are possible only if ISIG is set. These functions may be disabled individually by changing the value of the control character to an unlikely or impossible value (e.g. 0377).

If ICANON is set, canonical processing is enabled. This enables the erase and kill edit functions, and the assembly of input characters into lines delimited by NL, EOF, and EOL. If ICANON is not set, *read*(2) requests are satisfied directly from the input queue. A *read* will not be satisfied until at least MIN characters have been received or the timeout value TIME has expired. This allows fast bursts of input to be read efficiently while still allowing single character input. The MIN and TIME values are stored in the position for the EOF and EOL characters respectively. The time value represents tenths of seconds; values for TIME range from 2 to 255. If TIME has the value 0 or 1, no timeout occurs.

If XCASE is set, and if ICANON is set, an upper-case letter is accepted on input by preceding it with a \ character, and is output preceded by a \ character. In this mode, the following escape sequences are generated on output and accepted on input:

for:	*use*:
\	
|	\!
~	\^
{	\(
}	\)
\	\\

For example, **A** is input as **\a**, **\n** as **\\n**, and **\N** as **\\\n**.

If ECHO is set, characters are echoed as received.

When ICANON is set, the following echo functions are possible. If ECHO and ECHOE are set, the erase character is echoed as ASCII BS SP BS, which will clear the last character from a CRT screen. If ECHOE is set and ECHO is not set, the erase character is echoed as ASCII SP BS. If ECHOK is set, the NL character will be echoed after the kill character to emphasize that the line will be deleted. Note that an escape character preceding the erase or kill character removes any special function. If ECHONL is set, the NL character will be echoed even if ECHO is not set. This is useful for terminals set to local echo (so-called half duplex). Unless escaped, the EOF character is not echoed. Because EOT is the default EOF character, this prevents terminals that respond to EOT from hanging up.

If NOFLSH is set, the normal flush of the input and output queues associated with the quit and interrupt characters will not be done. When NOFLSH is set, a *del* (0177) or a ^| will cause a signal to be sent to the process. This process will be terminated. The character has already been placed in the raw queue and will be read with the next **read**.

The initial line-discipline control value is all bits clear.

The primary *ioctl*(2) system calls have the form:

```
ioctl (fildes, command, arg)
struct termio *arg;
```

The commands using this form are:

TCGETA — Get the parameters associated with the terminal and store in the *termio* structure referenced by **arg**.

TCSETA — Set the parameters associated with the terminal from the structure referenced by **arg**. The change is immediate.

TCSETAW — Wait for the output to drain before setting the new parameters. This form should be used when changing parameters that will affect output.

TCSETAF — Wait for the output to drain, then flush the input queue and set the new parameters.

Additional *ioctl*(2) calls have the form:

```
ioctl (fildes, command, arg)
int arg;
```

The commands using this form are:

TCSBRK	Wait for the output to drain. If *arg* is 0, then send a break (zero bits for 0.25 seconds).
TCXONC	Start/stop control. If *arg* is 0, suspend output; if 1, restart suspended output.
TCFLSH	If *arg* is 0, flush the input queue; if 1, flush the output queue; if 2, flush both queues.

FILES

/dev/tty
/dev/tty*
/dev/console

NOTES

Plexus defines two extra flags in the *c_iflag* field of the <**termio.h**> structure. They are:

```
     15    14
  ------------------------------------------------------
 |    |    |    |    |                ...              |
 |_|_|_|_|_|_|___|___|_________________________________|
     |       |
     |       |____     ICTS
     |
     |__________       ICTSLO
```

Where

ICTS	specifies that Clear to Send (CTS) is to be used as flow control. Note that XON and XOFF processing is not disabled and the ICP must be configured to accept the CTS signal.
ICTSLO	inverts the action of the CTS activity. For use with devices in which the polarity of the CTS signal is inverted.

The ICPs that correspond to *tty* device addresses are as follows:

/dev/tty0 - **/dev/tty7**	ICP 0
/dev/tty8 - **/dev/tty**15	ICP 1
/dev/tty16 - **/dev/tty**23	ICP 2
/dev/tty24 - **/dev/tty**31	ICP 3
/dev/tty32 - **/dev/tty**39	ICP 4

The ACPs that correspond to *tty* device addresses are as follows:

/dev/tty0 - **/dev/tty**15	ACP 0
/dev/tty16 - **/dev/tty**31	ACP 1
/dev/tty32 - **/dev/tty**47	ACP 2
/dev/tty48 - **/dev/tty**63	ACP 3
/dev/tty64 - **/dev/tty**79	ACP 4

See the the User's Guide for your system for ICP/ACP system configuration.

SEE ALSO

stty(1), ioctl(2), icp(4).

NAME

intro – introduction to system maintenance programs

DESCRIPTION

This section outlines certain procedures that will be of interest to those charged with the task of system maintenance. These are the standalone programs and a brief discussion of recovery from system crash.

BUGS

No manual can take the place of good, solid experience.

NAME

cat – concatenate and print files

SYNOPSIS

cat [**–u**] [**–s**] file ...

DESCRIPTION

Cat reads each *file* in sequence and writes it on the standard output. Thus:

cat file

prints the file, and:

cat file1 file2 >file3

concatenates the first two files and places the result on the third.

If no input file is given, or if the argument – is encountered, *cat* reads from the standard input file. Output is buffered in 512-byte blocks unless the **–u** option is specified. The **–s** option makes *cat* silent about non-existent files. No input file may be the same as the output file unless it is a special file.

NOTES

Plexus provides a standalone version of *cat* in addition to the one that runs under Sys5.

SEE ALSO

cp(1), pr(1).

NAME

crash – what to do when the system crashes

DESCRIPTION

This entry gives at least a few clues about how to proceed if the system crashes. It can not pretend to be complete.

How to bring it back up . If the reason for the crash is not evident (see below for guidance on "evident") you may want to try to dump the system if you feel up to debugging. At the moment a dump can be taken only on magtape. With a tape mounted and ready, stop the machine, load the address and start. This should write a copy of all of core on the tape with an EOF mark. Be sure the ring is in, the tape is ready, and the tape is clean and new.

In restarting after a crash, always bring up the system single-user, as modified for your particular installation. Then perform an *fsck (1M)* on all file systems which could have been in use at the time of the crash. If any serious file system problems are found, they should be repaired. When you are satisfied with the health of your disks, check and set the date if necessary, then come up multi-user.

To even boot the UNIX system at all, three files (and the directories leading to them) must be intact. First, the initialization program **/etc/init** must be present and executable. If it is not, the CPU will loop in user mode. For *init* to work correctly, **/dev/console** and **/bin/sh** must be present. If either does not exist, the symptom is best described as thrashing. *Init* will go into a *fork/exec* loop trying to create a shell with proper standard input and output.

If you cannot get the system to boot, a runnable system must be obtained from a backup medium. The root file system may then be doctored as a mounted file system as described below. If there are any problems with the root file system, it is probably prudent to go to a backup system to avoid working on a mounted file system.

Repairing disks . The first rule to keep in mind is that an addled disk should be treated gently; it should not be mounted unless necessary, and if it is very valuable yet in quite bad shape, perhaps it should be copied before trying surgery on it. This is an area where experience and informed courage count for much.

Fsck(1M) is adept at diagnosing and repairing file system problems. It first identifies all of the files that contain bad (out of range) blocks or blocks that appear in more than one file. Any such files are then identified by name and *fsck(1m)* requests permission to remove them from the file system. Files with bad blocks should be removed. In the case of duplicate blocks, all of the files except the most recently modified should be removed. The contents of the survivor should be checked after the file system is repaired to ensure that it contains the proper data. (Note that running *fsck(1M)* with the **-n** option will cause it to report all problems without attempting any repair.)

Fsck(1M) will also report on incorrect link counts and will request permission to adjust any that are erroneous. In addition, it will reconnect any files or directories that are allocated but have no file system references to a "lost+found" directory. Finally, if the free list is bad (out of range, missing, or duplicate blocks) *fsck(1M)* will, with the operators concurrence, construct a new one.

Why did it crash ? The UNIX system types a message on the console typewriter when it voluntarily crashes. Here is the current list of such messages, with enough information to provide a hope at least of the remedy. The message has the form "panic: ...", possibly accompanied by other information. Left unstated in all cases is the possibility that hardware or software error produced the message in some unexpected way.

blkdev
: The *getblk* routine was called with a nonexistent major device as argument. Definitely hardware or software error.

devtab
: Null device table entry for the major device used as argument to *getblk* . Definitely hardware or software error.

iinit
: An I/O error reading the super-block for the root file system during initialization.

no fs
: A device has disappeared from the mounted-device table. Definitely hardware or software error.

no imt
: Like "no fs", but produced elsewhere.

no clock
: During initialization, neither the line nor programmable clock was found to exist.

I/O error in swap
: An unrecoverable I/O error during a swap. Really should not be a panic, but it is hard to fix.

out of swap space
: A program needs to be swapped out, and there is no more swap space. It has to be increased. This really should not be a panic, but there is no easy fix.

trap
: An unexpected trap has occurred within the system. This is accompanied by three numbers: a "ka6", which is the contents of the segmentation register for the area in which the system's stack is kept; "aps", which is the location where the hardware stored the program status word during the trap; and a "trap

type" which encodes which trap occurred.

If you wish to examine the stack after such a trap, either dump the system, or use the console switches to examine core. The required address mapping is described below.

Interpreting dumps . All file system problems should be taken care of before attempting to look at dumps. The dump should be read into the file **/usr/tmp/core** ; *cp (1)* will do. At this point, you should execute *ps −el −c /usr/tmp/core* and *who* to print the process table and the users who were on at the time of the crash.

You should dump *(adb (1))* the first 30 bytes of **/usr/tmp/core** . Starting at location 4, the registers R0, R1, R2, R3, R4, R5, SP and KDSA6 are stored. If the dump had to be restarted, R0 will not be correct. Next, take the value of KA6 (location 22(8) in the dump) multiplied by 100(8) and dump 2000(8) bytes starting from there. This is the per-process data associated with the process running at the time of the crash. Relabel the addresses 140000 to 141776. R5 is C's frame or display pointer. Stored at (R5) is the old R5 pointing to the previous stack frame. At (R5)+2 is the saved PC of the calling procedure. Trace this calling chain until you obtain an R5 value of 141756, which is where the user's R5 is stored. If the chain is broken, you have to look for a plausible R5, PC pair and continue from there. Each PC should be looked up in the system's name list using *adb (1)* and its : command, to get a reverse calling order. In most cases this procedure will give an idea of what is wrong. A more complete discussion of system debugging is impossible.

SEE ALSO

adb(1), cp(1), fsck(1M).

NAME

dconfig – configure logical disks

SYNOPSIS

/etc/dconfig - for use under UNIX

dconfig - for running program from release tape only

/stand/dconfig - for standalone use (UNIX not running) only

DESCRIPTION

Dconfig allows you to change the Sys5 default logical disk address assignments and the default UNIX device mapping. It also can be used to verify the logical disk configuration, change the system nodename for **uucp** and **uname**, or change the primary bootname.

Dconfig has both regular (**/etc/dconfig**) and standalone (**/stand/dconfig**) versions. Plexus release tapes also contain a copy of **dconfig**. The arguments to **/etc/dconfig** (the regular version) differ from those for the standalone and tape versions. **/etc/dconfig** expects the special files defined in the **/dev** directory as arguments, while the standalone version and the release tape version both use built-in special filenames as described in your the user's manual for your system.

Dconfig prompts for responses, and gives the current values for each parameter in brackets. A <return> leaves the values the same; a <return> in response to a yes or no question defaults to "no". Unlike most Sys5 programs, **dconfig** expects response in terms of 512-byte sectors, rather than 1024 byte blocks.

If **dconfig** for any reason (e.g., permissions) cannot access the disk you type, it continues to give the "Disk?" prompt. For more complete information and examples, see the chapter on standalone programs in your user's manual.

NOTES

This is a Plexus command. It is not part of stock SYSTEM V.

Dconfig should not be run on disks containing a raw file system which starts at block 0 of the physical disk, as it will ruin the data in the raw file system.

Dconfig cannot use the first two blocks on a disk in a file system other than the first logical one. That is, if you have two disks, the file system size declarations for */dev/dsk/0s0* and */dev/dsk/0s1* must start at sector 0; 0s2-0s15 must not use sectors 0 and 1. On the second disk, the file system size declarations for */dev/dsk/1s0 (/dev/dsk/0s16)* and */dev/dsk/1s1 (/dev/dsk/0s17)* must start at sector 0; 1s2-1s15 (0s18-0s31) must not use sectors 0 and 1.

/etc/dconfig should be used only to examine, and not change, data.

SEE ALSO
uname(1).

NAME

dd – convert and copy a file

SYNOPSIS

dd [option=value] ...

DESCRIPTION

Dd copies the specified input file to the specified output with possible conversions. The standard input and output are used by default. The input and output block size may be specified to take advantage of raw physical I/O.

option	*values*
if=*file*	input file name; standard input is default
of=*file*	output file name; standard output is default
ibs=*n*	input block size *n* bytes (default 1024)
obs=*n*	output block size (default 1024)
bs=*n*	set both input and output block size, superseding *ibs* and *obs* ; also, if no conversion is specified, it is particularly efficient since no in-core copy need be done
cbs=*n*	conversion buffer size
skip=*n*	skip *n* input blocks before starting copy
seek=*n*	seek *n* blocks from beginning of output file before copying
count=*n*	copy only *n* input blocks
conv=**ascii**	convert EBCDIC to ASCII
ebcdic	convert ASCII to EBCDIC
ibm	slightly different map of ASCII to EBCDIC
lcase	map alphabetics to lower case
ucase	map alphabetics to upper case
swab	swap every pair of bytes
noerror	do not stop processing on an error
sync	pad every input block to *ibs*
flip	invert bits for P/35 compatibility.
...,...	several comma-separated conversions

Where sizes are specified, a number of bytes is expected. A number may end with **k** , **b** , or **w** to specify multiplication by 1024, 512, or 2, respectively; a pair of numbers may be separated by **x** to indicate a product.

Cbs is used only if *ascii* or *ebcdic* conversion is specified. In the former case *cbs* characters are placed into the conversion buffer, converted to ASCII, and trailing blanks trimmed and new-line added before sending the line to the output. In the latter case ASCII characters are read into the conversion buffer, converted to EBCDIC, and blanks added to make up an output block of size *cbs* .

After completion, *dd* reports the number of whole and partial input and output blocks.

EXAMPLE

This command will read an EBCDIC tape blocked ten 80-byte EBCDIC card images per block into the ASCII file **x** :

```
dd if=/dev/rmt/0m of=x ibs=800 cbs=80 conv=ascii,lcase
```

Note the use of raw magtape. *Dd* is especially suited to I/O on the raw physical devices because it allows reading and writing in arbitrary block sizes.

SEE ALSO

cp(1).

DIAGNOSTICS

f+p blocks in(out) *numbers of full and partial blocks read(written)*

NOTES

Plexus provides a standalone version of dd in addition to the one that runs under Sys5.

BUGS

The ASCII/EBCDIC conversion tables are taken from the 256-character standard in the CACM Nov, 1968. The *ibm* conversion, while less blessed as a standard, corresponds better to certain IBM print train conventions. There is no universal solution.

New-lines are inserted only on conversion to ASCII; padding is done only on conversion to EBCDIC. These should be separate options.

NAME

dformat - disk formatter

SYNOPSIS

dformat - for running the program from a release tape only

/stand/dformat - for standalone use (no UNIX) only

DESCRIPTION

Dformat is the Sys5 disk formatting program. With this utility you can format the disk and spare bad sectors, list the bad sectors at the console, or read the disk for bad spots on the disk media. This utility is explained in detail in the *Plexus User's Manual.*

Dformat prompts for the parameters it needs. For examples, see the *Plexus User's Manual.*

NOTES

This is a Plexus command. It is not part of standard SYSTEM V.

SEE ALSO

Plexus User's Manual

NAME

du – summarize disk usage

SYNOPSIS

du [**–ars**] [names]

DESCRIPTION

Du gives the number of blocks contained in all files and (recursively) directories within each directory and file specified by the *names* argument. The block count includes the indirect blocks of the file. If *names* is missing, . is used. Blocks are 1024 bytes long.

The optional argument **–s** causes only the grand total (for each of the specified *names)* to be given. The optional argument **–a** causes an entry to be generated for each file. Absence of either causes an entry to be generated for each directory only.

Du is normally silent about directories that cannot be read, files that cannot be opened, etc. The **–r** option will cause *du* to generate messages in such instances.

A file with two or more links is counted only once.

NOTES

Plexus provides a standalone version of *du* in addition to the one that runs under Sys5.

BUGS

If the **–a** option is not used, non-directories given as arguments are not listed.
If there are too many distinct linked files, *du* will count the excess files more than once.
Files with holes in them will get an incorrect block count.

NAME

fbackup - make a fast tape backup of a file system

SYNOPSIS

fbackup - for running the program from a release tape only

/stand/fbackup - for standalone (no UNIX) use only

DESCRIPTION

The standalone program **fbackup** makes a fast (intermittently streaming) copy of data on disk to tape, or data on tape to disk. It is usually used to make a copy of a file system. **Fbackup** is faster than **dump** and writes in a format that is understood by **dd** (i.e., it is a byte-by-byte copy), so you should use **fbackup** rather than **dump** if you need the speed.

Fbackup prompts for its arguments. It can copy between an EMSP disk and a 9-track tape, or between an IMSC disk and a 9-track or cartridge tape. It does not support copies between an EMSP disk and cartridge tape. **Fbackup** writes to 9-track tape in block sizes of 16K bytes per record.

To use **fbackup** to backup a logical file system, you will need to know the sector number where the file system starts and the length of the file system in 512-byte disk sectors. Use **dconfig**(8) to find these numbers.

NOTES

This is a Plexus program.

SEE ALSO

Plexus User's Manual

NAME

fsck, dfsck – file system consistency check and interactive repair

SYNOPSIS

/etc/fsck [**–y**] [**–n**] [**–s**X] [**–S**X] [**–t** file] [**–q**] [**–D**] [**–f**] [file-systems]

/etc/dfsck [options1] filsys1 ... – [options2] filsys2 ...

DESCRIPTION

Fsck

Fsck audits and interactively repairs inconsistent conditions for UNIX system files. If the file system is consistent then the number of files, number of blocks used, and number of blocks free are reported. If the file system is inconsistent the operator is prompted for concurrence before each correction is attempted. It should be noted that most corrective actions will result in some loss of data. The amount and severity of data lost may be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond **yes** or **no**. If the operator does not have write permission *fsck* will default to a **–n action.**

Fsck has more consistency checks than its predecessors *check* , *dcheck* , *fcheck* , and *icheck* combined.

The following options are interpreted by *fsck* .

–y Assume a yes response to all questions asked by *fsck* .

–n Assume a no response to all questions asked by *fsck* ; do not open the file system for writing.

–s *X* Ignore the actual free list and (unconditionally) reconstruct a new one by rewriting the super-block of the file system. The file system should be unmounted while this is done; if this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately afterwards. This precaution is necessary so that the old, bad, in-core copy of the superblock will not continue to be used, or written on the file system.

The **–s** *X* option allows for creating an optimal free-list organization. The following forms of *X* are supported for the following devices:

–s3 (RP03)
–s4 (RP04, RP05, RP06)
–sBlocks-per-cylinder:Blocks-to-skip (for anything else)

If *X* is not given, the values used when the file system was created are used. If these values were not specified, then the value *400:7* is used.

–S *X* Conditionally reconstruct the free list. This option is like **–s** *X* above except that the free list is rebuilt only if there were no discrepancies discovered in the file system. Using **–S** will force a no response to all questions asked by *fsck* . This option is useful for forcing free list reorganization on uncontaminated file systems.

–t If *fsck* cannot obtain enough memory to keep its tables, it uses a scratch file. If the **–t** option is specified, the file named in the next argument is used as the scratch file, if needed. Without the **–t flag,** *fsck* will prompt the operator for the name of the scratch file. The file chosen should not be on the file system being checked, and if it is not a special file or did not already exist, it is removed when *fsck* completes.

–q Quiet *fsck.* Do not print size-check messages in Phase 1. Unreferenced **fifos** will silently be removed. If *fsck* requires it, counts in the superblock will be automatically fixed and the free list salvaged.

–D Directories are checked for bad blocks. Useful after system crashes.

–f Fast check. Check block and sizes (Phase 1) and check the free list (Phase 5). The free list will be reconstructed (Phase 6) if it is necessary.

If no *file-systems* are specified, *fsck* will read a list of default file systems from the file **/etc/checklist** .

Inconsistencies checked are as follows:

- Blocks claimed by more than one i-node or the free list.
- Blocks claimed by an i-node or the free list outside the range of the file system.
- Incorrect link counts.
- Size checks:
  Incorrect number of blocks.
  Directory size not 16-byte aligned.
- Bad i-node format.
- Blocks not accounted for anywhere.
- Directory checks:
  File pointing to unallocated i-node.
  I-node number out of range.
- Super Block checks:
  More than 65536 i-nodes.
  More blocks for i-nodes than there are in the file system.
- Bad free block list format.
- Total free block and/or free i-node count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the **lost+found** directory, if the files are nonempty. The user will be notified if the file or directory is empty or not. If it is empty, *fsck* will silently remove them. *Fsck* will force the reconnection of nonempty directories. The name assigned is the i-node number. The only restriction is that the directory **lost+found** must preexist in the root of the file system being checked and must have empty slots in which entries can be made. This is accomplished by making **lost+found** , copying a number of files to the directory, and then removing them (before *fsck* is executed).

Checking the raw device is almost always faster and should be used with everything but the *root* file system.

Dfsck

Dfsck allows two file system checks on two different drives simultaneously. *options1* and *options2* are used to pass options to *fsck* for the two sets of file systems. A – is the separator between the file system groups.

The *dfsck* program permits an operator to interact with two *fsck (1M)* programs at once. To do this, *dfsck* prints the file system name for each message to the operator. When answering a question from *dfsck* , the operator must prefix the response with a **1** or a **2** (indicating that the answer refers to the first or second file system group).

Do not use *dfsck* to check the *root* file system.

FILES

/etc/checklist	contains default list of file systems to check.
/etc/checkall	optimizing *dfsck* shell file.

SEE ALSO

checkall(1M), clri(1M), ncheck(1M), crash(8).
checklist(4), fs(4) in the *Sys5 UNIX Programmer's Reference Manual.*
Setting Up the Sys5 UNIX in the *Sys5 UNIX Administrator Guide* .

BUGS

I-node numbers for . and .. in each directory should be checked for validity.

DIAGNOSTICS

The diagnostics produced by *fsck* are intended to be self-explanatory.

NAME

fsdb – file system debugger

SYNOPSIS

/etc/fsdb special [–]

DESCRIPTION

Fsdb can be used to patch up a damaged file system after a crash. It has conversions to translate block and i-numbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an i-node. These greatly simplify the process of correcting control block entries or descending the file system tree.

Fsdb contains several error-checking routines to verify i-node and block addresses. These can be disabled if necessary by invoking *fsdb* with the optional – argument or by the use of the **O** symbol. (*Fsdb* reads the i-size and f-size entries from the superblock of the file system as the basis for these checks.)

Numbers are considered decimal by default. Octal numbers must be prefixed with a zero. During any assignment operation, numbers are checked for a possible truncation error due to a size mismatch between source and destination.

Fsdb reads a block at a time and will therefore work with raw as well as block I/O. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block.

The symbols recognized by *fsdb* are:

#	absolute address
i	convert from i-number to i-node address
b	convert to block address
d	directory slot offset
+ , –	address arithmetic
q	quit
> , <	save, restore an address
=	numerical assignment
=+	incremental assignment
=–	decremental assignment
="	character string assignment
O	error checking flip flop
p	general print facilities
f	file print facility
B	byte mode
W	word mode
D	double word mode

!	escape to shell

The print facilities generate a formatted output in various styles. The current address is normalized to an appropriate boundary before printing begins. It advances with the printing and is left at the address of the last item printed. The output can be terminated at any time by typing the delete character. If a number follows the **p** symbol, that many entries are printed. A check is made to detect block boundary overflows since logically sequential blocks are generally not physically sequential. If a count of zero is used, all entries to the end of the current block are printed. The print options available are:

i	print as i-nodes
d	print as directories
o	print as octal words
e	print as decimal words
c	print as characters
b	print as octal bytes

The **f** symbol is used to print data blocks associated with the current i-node. If followed by a number, that block of the file is printed. (Blocks are numbered from zero.) The desired print option letter follows the block number, if present, or the **f** symbol. This print facility works for small as well as large files. It checks for special devices and that the block pointers used to find the data are not zero.

Dots, tabs, and spaces may be used as function delimiters but are not necessary. A line with just a new-line character will increment the current address by the size of the data type last printed. That is, the address is set to the next byte, word, double word, directory entry or i-node, allowing the user to step through a region of a file system. Information is printed in a format appropriate to the data type. Bytes, words and double words are displayed with the octal address followed by the value in octal and decimal. A **.B** or **.D** is appended to the address for byte and double word values, respectively. Directories are printed as a directory slot offset followed by the decimal i-number and the character representation of the entry name. I-nodes are printed with labeled fields describing each element.

The following mnemonics are used for i-node examination and refer to the current working i-node:

md	mode
ln	link count
uid	user ID number
gid	group ID number
sz	file size

a#	data block numbers (0 – 12)
at	access time
mt	modification time
maj	major device number
min	minor device number

EXAMPLES

386i	prints i-number 386 in an i-node format. This now becomes the current working i-node.
ln=4	changes the link count for the working i-node to 4.
ln=+1	increments the link count by 1.
fc	prints, in ASCII, block zero of the file associated with the working i-node.
2i.fd	prints the first 32 directory entries for the root i-node of this file system.
d5i.fc	changes the current i-node to that associated with the 5th directory entry (numbered from zero) found from the above command. The first logical block of the file is then printed in ASCII.
512B.p0o	prints the superblock of this file system in octal.
2i.a0b.d7=3	changes the i-number for the seventh directory slot in the root directory to 3. This example also shows how several operations can be combined on one command line.
d7.nm="name"	changes the name field in the directory slot to the given string. Quotes are optional when used with **nm** if the first character is alphabetic.
a2b.p0d	prints the third block of the current i-node as directory entries.

SEE ALSO

fsck(1M), dir(4), fs(4).

NAME

help – ask for help

SYNOPSIS

help [args]

DESCRIPTION

Help finds information to explain a message from a command or explain the use of a command. Zero or more arguments may be supplied. If no arguments are given, *help* will prompt for one.

The arguments may be either message numbers (which normally appear in parentheses following messages) or command names, of one of the following types:

type 1	Begins with non-numerics, ends in numerics. The non-numeric prefix is usually an abbreviation for the program or set of routines which produced the message (e.g., **ge6**, for message 6 from the *get* command).
type 2	Does not contain numerics (as a command, such as **get**)
type 3	Is all numeric (e.g., **212**)

The response of the program will be the explanatory information related to the argument, if there is any.

When all else fails, try "help stuck".

FILES

/usr/lib/help	directory containing files of message text.
/usr/lib/help/helploc	file containing locations of help files not in **/usr/lib/help**.

DIAGNOSTICS

Use *help (1)* for explanations.

NAME

ls – list contents of directories

SYNOPSIS

ls [**–logtasdrucif**] names

DESCRIPTION

For each directory named, *ls* lists the contents of that directory; for each file named, *ls* repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents. There are several options:

–l List in long format, giving mode, number of links, owner, group, size in bytes, and time of last modification for each file (see below). If the file is a special file, the size field will contain the major and minor device numbers, rather than a size.

–o The same as **–l** , except that the group is not printed.

–g The same as **–l** , except that the owner is not printed.

–t Sort by time of last modification (latest first) instead of by name.

–a List all entries; in the absence of this option, entries whose names begin with a period (.) are *not* listed.

–s Give size in 1024-byte blocks (including indirect blocks) for each entry.

–d If argument is a directory, list only its name; often used with **–l** to get the status of a directory.

–r Reverse the order of sort to get reverse alphabetic or oldest first, as appropriate.

–u Use time of last access instead of last modification for sorting (with the **–t** option) and/or printing (with the **–l** option).

–c Use time of last modification of the inode (mode, etc.) instead of last modification of the file for sorting (**–t**) and/or printing (**–l**).

–i For each file, print the i-number in the first column of the report.

–f Force each argument to be interpreted as a directory and list the name found in each slot. This option turns off **–l** , **–t** , **–s** , and **–r** , and turns on **–a** ; the order is the order in which entries appear in the directory.

The mode printed under the **–l** option consists of 10 characters that are interpreted as follows:

The first character is:

- **d** if the entry is a directory;
- **b** if the entry is a block special file;
- **c** if the entry is a character special file;
- **p** if the entry is a fifo (a.k.a. "named pipe") special file;
- – if the entry is an ordinary file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of others in the user-group of the file; and the last to all others. Within each set, the three characters indicate permission to read, to write, and to execute the file as a program, respectively. For a directory, "execute" permission is interpreted to mean permission to search the directory for a specified file.

The permissions are indicated as follows:

- **r** if the file is readable;
- **w** if the file is writable;
- **x** if the file is executable;
- – if the indicated permission is *not* granted.

The group-execute permission character is given as **s** if the file has set-group-ID mode; likewise, the user-execute permission character is given as **s** if the file has set-user-ID mode. The last character of the mode (normally **x** or **–**) is **t** if the 1000 (octal) bit of the mode is on; see *chmod (1)* for the meaning of this mode. The indications of set-ID and 1000 bit of the mode are capitalized if the corresponding execute permission is *not* set.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

FILES

/etc/passwd	to get user IDs for **ls –l** and **ls –o** .
/etc/group	to get group IDs for **ls –l** and **ls –g** .

NOTES

Plexus provides a standalone version of *ls* in addition to the one that runs under Sys5.

SEE ALSO

chmod(1), find(1).

BUGS

The "-*g*" and "-*o*" options are incompatible.

NAME

mkfs – construct a file system

SYNOPSIS

/etc/mkfs special blocks[:inodes] [gap blocks/cyl]
/etc/mkfs special proto [gap blocks/cyl]

DESCRIPTION

Mkfs constructs a file system by writing on the special file according to the directions found in the remainder of the command line. The command waits 10 seconds before starting to construct the file system. If the second argument is given as a string of digits, *mkfs* builds a file system with a single empty directory on it. The size of the file system is the value of *blocks* interpreted as a decimal number. The boot program is left uninitialized. If the optional number of inodes is not given, the default is the number of *logical* blocks divided by 4.

If the second argument is a file name that can be opened, *mkfs* assumes it to be a prototype file *proto*, and will take its directions from that file. The prototype file contains tokens separated by spaces or new-lines. The first token is the name of a file to be copied onto block zero as the bootstrap program. The second token is a number specifying the size of the created file system in *physical* disk blocks. Typically it will be the number of blocks on the device, perhaps diminished by space for swapping. The next token is the number of i-nodes in the file system. The maximum number of i-nodes configurable is 65500. The next set of tokens comprise the specification for the root file. File specifications consist of tokens giving the mode, the user ID, the group ID, and the initial contents of the file. The syntax of the contents field depends on the mode.

The mode token for a file is a 6-character string. The first character specifies the type of the file. (The characters **–bcd** specify regular, block special, character special and directory files respectively.) The second character of the type is either **u** or – to specify set-user-id mode or not. The third is **g** or – for the set-group-id mode. The rest of the mode is a three digit octal number giving the owner, group, and other read, write, execute permissions (see *chmod*(1)).

Two decimal number tokens come after the mode; they specify the user and group ID's of the owner of the file.

If the file is a regular file, the next token is a pathname whence the contents and size are copied. If the file is a block or character special file, two decimal number tokens follow which give the major and minor device numbers. If the file is a directory, *mkfs* makes the entries . and .. and then reads a list of names and (recursively) files specifications for the entries in the directory. The scan is terminated with the token **$**.

A sample prototype specification follows:

```
/stand/diskboot
4872 110
d—777 3 1
usr     d—777 3 1
        sh      ——755 3 1 /bin/sh
        ken     d—755 6 1
                $
        b0      b—644 3 1 0 0
        c0      c—644 3 1 0 0
        $
$
```

In both command syntaxes, the rotational *gap* and the number of *blocks/cyl* can be specified. The *default* will be used if the supplied *gap* and *blocks/cyl* are considered illegal values or if a short argument count occurs. Your User's Manual lists the default values for your system.

The best gap factor should be calculated as:

gap = (sectors per track / 2) + number of heads

If you are using a Xylogics disk (P/60 and P/75 only) and do not enter the gap size and blocks/cyl, a warning will advise you of this.

At any time during the program you can <DEL> to cancel the program and start over.

A new flag value has been added to help **mkfs** to do its job quickly and quietly. A **-q** before the device name prevents it from sleeping, or from printing any warnings or statistics.

SEE ALSO

chmod(1), dir(4), fs(4).

BUGS

If a prototype is used, it is not possible to initialize a file larger than 64K bytes, nor is there a way to specify links.

NAME

od – octal dump

SYNOPSIS

od [**–bcdox**] [file] [[+]offset[.][**b**]]

DESCRIPTION

Od dumps *file* in one or more formats as selected by the first argument. If the first argument is missing, **–o** is default. The meanings of the format options are:

–b Interpret bytes in octal.

–c Interpret bytes in ASCII. Certain non-graphic characters appear as C escapes: null=**\0**, backspace=**\b**, form-feed=**\f**, new-line=**\n**, return=**\r**, tab=**\t**; others appear as 3-digit octal numbers.

–d Interpret words in decimal.

–o Interpret words in octal.

–x Interpret words in hex.

The *file* argument specifies which file is to be dumped. If no file argument is specified, the standard input is used.

The offset argument specifies the offset in the file where dumping is to commence. This argument is normally interpreted as octal bytes. If . is appended, the offset is interpreted in decimal. If **b** is appended, the offset is interpreted in blocks of 512 bytes. If the file argument is omitted, the offset argument must be preceded by + .

Dumping continues until end-of-file.

NOTES

Plexus provides a standalone version of *od* in addition to the one that runs under Sys5.

SEE ALSO

adb(1).

NAME

restor – incremental file system restore

SYNOPSIS

restor *key* [*arguments*]

DESCRIPTION

Restor is used to read magnetic tapes dumped with the *dump* command. A *dump* followed by a *mkfs* and a *restor* is used to change the size of a file system.

In the standalone version of this program, a final +*n* argument advances the tape *n* files before executing the *restor*. To space forward *n* files in the online version type

```
/usr/plx/tape srcheof n
```

before typing the *restor* command.

The *key* specifies what is to be done. *Key* is one of the characters **rRxt**, optionally combined with **f**.

f Use the first *argument* as the name of the tape instead of the default.

r or R The tape is read and loaded into the file system specified in *argument*. If the key is **R**, *restor* asks which tape of a multi-volume set to start on. This allows *restor* to be interrupted and then restarted (an *fsck* must be done before the restart). The **r** option should only be used to restore a complete dump tape onto a clear file system, or to restore an incremental dump tape onto a file system so created. Thus:

```
/etc/mkfs /dev/dsk/0s1 18000
restor r /dev/dsk/0s1
```

is a typical sequence to restore a complete dump. Another *restor* can be done to get an incremental dump in on top of this.

x Each file on the tape named by an *argument* is extracted. The file name has all "mount" prefixes removed; for example, if **/usr** is a mounted file system, **/usr/bin/lpr** is named **/bin/lpr** on the tape. The extracted file is placed in a file with a numeric name supplied by *restor* (actually the inode number). In order to keep the amount of tape read to a minimum, the following procedure is recommended:

1. Mount volume 1 of the set of dump tapes.
2. Type the *restor* command.

3. *Restor* will announce whether or not it found the files, give the numeric name that it will assign to the file, and rewind the tape.

4. It then asks you to "mount the desired tape volume". Type the number of the volume. On a multi-volume dump the recommended procedure is to mount the last through the first volumes, in that order. *Restor* checks to see if any of the requested files are on the mounted tape (or a later tape–thus the reverse order) and doesn't read through the tape if no files are. If you are working with a single-volume dump or if the number of files being restored is large, respond to the query with **1** and *restor* will read the tapes in sequential order.

t Print the date the tape was written and the date the file system was dumped from.

FILES

/dev/rpt/0m	(cartrige tape - rewind)
/dev/rpt/0mn	(cartrige tape - no rewind)
/dev/rrm/0m	(9-track tape - rewind)
/dev/rrm/0mn	(9-track tape - no rewind)
rst*	

NOTES

This command has a standalone version.

SEE ALSO

dump(1M), dumpdir(1M), fsck(1M), mkfs(1M).

DIAGNOSTICS

There are various diagnostics involved with reading the tape and writing the disk. There are also diagnostics if the i-list or the free list of the file system is not large enough to hold the dump.

If the dump extends over more than one tape, it may ask you to change tapes. Reply with a new-line when the next tape has been mounted.

BUGS

There is redundant information on the tape that could be used in case of tape reading problems. Unfortunately, *restor* doesn't use it. The **x** option of the standalone version does not work.

The Sys5 version of *restor* cannot read multiple volume dumps made with the Sys3 version of *dump*. If you have multiple volume dumps of a Sys3 file system, use the standalone *restor* on your old Sys3 release tape to load the dump onto your new Sys5 file system. Then use the Sys5 version of */etc/dump* to make a new backup.