```
  666               JJJJJJJ U       U LL                   888       11
 6   6                 JJ   U       U LL                  8   8     1 1
6                      JJ   U       U LL                  8   8       1
6                      JJ   U       U LL                   888        1
6 6666                 JJ   U       U LL                  8   8       1
66    6            J   JJ   U       U LL                 8     8      1
6     6            J   JJ    U     U  LL                 8     8      1
 66666              JJJJ      UUUU    LLLLLLLL            88888     11111
```

JSYS LORE: TENEX vs TOPS-20 DIFFERENCES

This file is primarily intended to be a compendium of and repository for as many Tenex/Tops-20 differences as possible, in order to facilitate conversion of programs and promote the writing of software that will run on either system. Secondarily it will serve as a place to document various information which the DEC documentation may overlook, mutiliate, or just plain get wrong.

CANONICAL LOCATION: [SRI-NIC]<KLH>JSYS.TXT
Current maintainer: KLH at SRI-NIC

Please report any additions or corrections to the maintainer(s). Changes to local copies of this file are okay, as long as these changes are reported so that they can be incorporated into the master file. Since all site-dependent information will simply be included with a note identifying the site it is applicable on, there should be no problem with conflicting implementations or interpretations of JSYSes; that is what this file is intended to document!

TO DO:

Furnish suggested routines for 10X/T20 compatibility??
Flesh out machine-language section (ADJBP, etc.)
Find something about 10X history; where is more recent doc?
Try to track down 10X evolution paths (check various sites)

[MT: General comments
- Do you have the latest edition (December 1976) of the BBN JSYS manual?
- I'll try and send documentation on VTS and Internet/TCP stuff. I'm not sure the VTS stuff belongs since it is currently only at MIT, though Foonex will have it one of these days.]

Known TENEX sites:

Name	Machine, OS, EXEC	Comments
BBN-TENEXB	1.34.56 1.54.66	
BBN-TENEXC	1.34.57 1.54.66	
BBN-TENEXE	1.34.55 1.54.66	
DARCOM-KA	KA 1.34.55 1.53.98	
I4-TENEX	1.34.8 ?	
MOFFETT-ARC	dead	
MOFFETT-SUBNET	timeout	
NOSC-SECURE2	timeout	
OFFICE-1	1.34.40 1.53.72	
OFFICE-2	1.34.40 1.53.72	
OFFICE-3	August-1.0.10 1.53.72	
OFFICE-7	1.34.40 1.53.72	
PARC-MAXC	1.34.31 1.54.12	
USC-ECL	1.33.55 1.54.30	
USC-ISI	1.34.52 1.54.28	
CCA-TENEX	1.32.66 1.53F	
SUMEX-AIM	1.31.53 1.54.32	
SCI-ICS	F2 refusing	
SRI-CSL	F2 1.34.40 1.53.98	
SRI-NIC	F3 August-1.0.10 1.53.72	
SCRC*	F?	

* Not on net

ALPHABETICAL-JSYS-PAGE:

This page lists the JSYSes in alphanumeric order, with notes as to their operation on TENEX (10X). Some V4 calls have "%" appended but no others do. The DEC Monitor Call manual is understood to be the source for any information not commented on; in other words, the information here should allow you to hack TENEX with only a TOPS-20 manual. It is NOT intended to let you hack T20 with a TENEX manual, but should be able to assure you whether your 10X stuff will run on T20 or not.

```
T20: The T20 manual is assumed to be that for TOPS-20AN V4.
        Where substantial differences exist between V3 and V4 I have
        indicated them as T20.3 and T20.4, but don't use this file
        just to list V3/V4 diffs -- that is already in the V4 manual.
10X: The "10X doc" referred to is the Sept 1, 1973 revision of the BBN
        TENEX JSYS manual.  I have included updates from the May 27, 1976
        revision as "76:".  One other source which I found on-line as
        [SRI-KL]DOC:JSYS.DOC contributed some more stuff which is included
        as "A:".  Information marked "ISI:" is from the file
        [ISIE]DOC:TENEX-TOPS20.JSYS-DIFFERENCES as of 15-May-81.
        "TCP:" is from the 20-Jun-80 version of "TCP JSYS Calling Sequences"
        by Postel and Plummer, canonical filename unknown.
```

```
------------- Format of entries --------------
JSYS:   <val>   <code> <ret-code>       <description>

Val:    JSYS number in octal

Codes:  ?  Unknown, no information
        !  New T20 call, not in 10X
        -  10X call, not in T20 doc or in monitor (may be NIM'd in MONSYM tho)
        -? 10X call, not in T20 doc but may work (i.e. MONSYM doesn't NIM it)
        =  just about the same
        == exactly the same
        * Some differences
        ** BIG differences!

Ret-code: <P><n><I> [ / <20X code if different> ]
                <P> = "P" Privileged JSYS ("p" if only for some forms)
                        normally means WHEEL/OPER cap needed.
                        "N" - Netwizard, "M" - Maintenance
                <n> = 1,2,3,4,5 - highest return addr
                <I> = PSI possible, one or more of following:
                        X = illegal instruction
                        Q = quota exceeded or disk full
                        D = data error
                        E = EOF
                        (A=AOV,F=FOV,P=POV,R=IMR,W=IMW,M=MSE,N=NXP,I=IFT)
```

e.g. "1X/p2" means that on 10X the call always returns +1 but can get ILINT on errors. On T20 the call can return +2, does not cause ILINT's, and some functions require WHEEL/OPER privileges.

Note: ILINT = Illegal Instruction Interrupt, channel B15.

```
--------------------------
ABORT:  747     ?  [TCP] Abort connection (Internet/TCP)

[TCP:

Accepts AC1: Flags,,JCN or Pointer-to-Connection-Block
Returns +1:     error, code in AC1
        +2:     OK, connection deleted
```

```
    Flags:

        JCNSupplied:   (see TCP-JSYS-DOC)

The local end of the connection is forgotten.  An attempt to notify
the remote end is made by sending a RST packet.  Should this not be
delivered, the other end will discover its half open connection the
next time it attempts to use it.

]
--------------------------
ACCES:  552      ! -/p1X Specifies access to a directory

--------------------------
ADBRK:  570      ! -/1X  Controls address breaks

Not on 2020's
--------------------------
ADVIZ:  315      - 2/-   Set up for link advice
ADVIS:

ADVIS was the name for this call on old T20 versions.  Its function is
now done by new TLINK bits.  The following is the 10X doc for this
call.

Accepts AC1:LH: B0 clear ADVIZ link
                B1 set to ADVIZ line given in right half
                B2 set to receive advice from designated line in RH
            RH: terminal line designator
Returns +1: failure, error # in AC1
        +2: success
Errors: ADVX1   advice refused
        ADVX2   not used
        ADVX3   B1 and B2 of AC1 both on in call
        ADVX4   an attempt to accept advice with ADVIZ already in progress
        (TLINK errors can also occur)

An ADVIZ link is established by having the advisee execute a "receive
advice" (B2) designating the intended advisor, and then the advisor
executing a "send advice" (B1) designating the advisee line.  An ADVIZ
link causes the advisor's characters to be input to the advisee's job.
ADVIZ provides terminal output using the TLINK mechanism.  While ADVIZ
is in progress, a ^C character typed by the advisor cuases the ADVIZ
link to be broken, but leaves the output link in effect.  A ^C will be
sent to the advisee job when the advisor types a ^Y.
--------------------------
AIC:    131      = 1X    Activates software interrupt channels

T20: new err FRKHX8
--------------------------
ALLOC:  520      ! -/P2  Allocates a device

--------------------------
ARCF:   247      ! -/p1X Archive/virtual disk operations

New V4 call (? but why is # 247 then?  Because it was developed at
BBN perhaps?)
--------------------------
ASND:    70      = 2     Assigns a device

T20: new err DSMX1 "file(s) not closed"
--------------------------
ASNDC:  262      - 2/-   Assign Display Console (not in T20)

Apparently a BBN hack, see 10X doc p. 4-36
--------------------------
```

ASNDP: 260 - 2/- Assign Display Process (not in T20)

Apparently a BBN hack, see 10X doc p. 4-36

ASNIQ: 756 ? Unknown (Internet version of ASNSQ?)

ASNSQ: 752 == N2 Assigns ARPANET special message queue

ASPTY: 360 - 2/- [A] Assign PTY

[A:

ASsign Pseudo-TtY (PTY) line to this job.

```
Accepts AC1:     0 - Check if system full, return +1 if so.
                -1 - Ignore system full check
Returns +1: Unsuccessful, no PTY assigned
        +2: Successful, PTY designator (of form 400000+n) in AC1
Errors: none
```

This assigns an arbitrary free pseudo teletype to the job if one is free. The terminal designator returned in AC1 will provide the job with a way of testing all of the states of the pseudo teletype and the means of moving characters to and from it. The system full check determines if drum space is above some minimum, a fork is available, a job is available and an OFN is available.

]

ATACH: 116 * p2 Attachs a terminal to a job

AC1 changes - the 10X flag bits in LH(AC2) have been moved to LH(AC1)

```
  B0 AT%CCJ (10X: - ) (T20: generates a ^C interrupt to the lowest
        process in the job that is enabled for a ^C interrupt
        if the job is currently attached to another terminal.  If
        bit is off, the job simply continues running when attached.)
        This flag used to be in AC2,B0 in 10X.
  B1 AT%NAT (10X: - ) (T20: do not attach unless remote job has controlling
        terminal, in which case the remote job is detached)
        [ISI: This flag used to be in AC2,B1 in 10X.]
  B2 AT%TRM (10X: - ) (T20: attach the given job to the terminal specified
        in AC4)

AC2:    T20: user number under which the job to be attached is logged in.
        10X:
          B0 - same as T20 AT%CCJ
          B1 - [ISI: proxy ATACH; job # in RH(AC1),TTY# in RH(AC4))]
          B2 - [ISI: remote DTACH]

AC3:    T20: pointer to ASCIZ password string in the caller's address space.
        10X: same as T20 [ISI: - ]

AC4:    T20: number of the terminal to be attached to the specified job,
                if AT%TRM is set.
        10X: - [ISI: see AC2,B1]

Errors: same
```

[76: (reproduced verbatim, see if you can parse it)

If the job executing the attached JSYS has WHEEL or OPERATOR capabilities all functions are legal. If the job executing the JSYS is attaching a terminal which it has assigned to a job which it owns that is a legal function. Otherwise the controlling terminal can be attached to a job only if that job doesn't already have a control terminal and the directory and password given are correct.]

```
[76: ATACH will fail unless executed by a fork in the top group of the
job.]
--------------------------
ATGRP:  332      - 2/-   [76] Get pie-slice group for account

[76:  (not in '73 doc)

Accepts AC1: account number in bits 3-35 if bits 0-2  =  5,
                else  a  string  pointer  to  the  new account
                string in the address space of caller.  If the
                normal  terminating null byte is not seen, the
                string is terminated after the 8th  full  word
                is processed.
Returns +1: Unsuccessful, error # in 1
        +2: Successful, SIXBIT group name in 2
Errors:
        ACCTX1    No user/account data file
        PIEX1     Not a pie-slice scheduler system
        ATGPX1    Invalid account or no group for this account
]
--------------------------
ATI:    137      = 1X    Assigns a terminal code to an interrupt channel

Form of call is exactly the same, but there are some differences
in the terminal codes allowed; see PSI-CHANNELS.

--------------------------
ATNVT:  274      = 2     Creates ARPANET Network Virtual Terminal Connection
ATPTY:

In 10X this is known as ATPTY, but call is the same.

AC1:
  B0    -        10X: [76: use NVT specified in AC3] (T20: - )
  B1    -        10X: [76: assign the terminal (don't deassign)] (T20: - )
  B2  AN%NTP     10X: -
                 T20: Use New-Telnet protocol (off: old telnet)
  [MT: under VTS there is a bit that means use SUPDUP]

AC3:    T20: -
        10X: [76: (only if B0 of AC1 set) terminal designator]

[76:    Unless a special capability is enabled, B1 of AC1 must be set.
        The following are meaningful combinations of B0 and B1.

        00 select and unused and unassigned  NVT  and  attach  the
                network connections.  The NVT remains unassigned.
        01 select and unused and unassigned NVT, assign it (asnd))
                and attach to the network connections.
        10 verify the assignment of the  specified  NVT,  deassign
                and attach to the network connections.
        11 verify the assignment of the specified NVT, attach to
                network connections.
]

Error code values & meanings are identical, but:
10X: error code symbols are ATPXnn
T20: error code symbols are ATNXnn, e.g. ATPX5 = ATNX5
--------------------------
BIN:     50      == 1XDE Performs byte input

--------------------------
BKJFN:   42      = 2     Backs up pointer by one byte

T20: new err TTYX01 "line is not active"
--------------------------
```

```
BOOT:    562      ! -/PIX Do functions required for loading front-end software
V4 mods
--------------------------
BOUT:     51      = 1XQD  Performs byte output

T20: new err IOX11, IOX33 "TTY input buffer full" (?!)
--------------------------
BPT:     304      -? 1/-  Breakpoint

10X doc says "implementation incomplete - identical to HALTF"
--------------------------
CACCT:     4      = 2     Changes account designator
[76:
10X also accepts bits in AC2:
        B0 -    Do account change, but not a pie-slice group change
        B1 -    Don't print an entry on the logging TTY
        These bits require WHL/OPR privileges enabled, otherwise
        they are ignored completely.

10X: err PIEX1 "not a pie-slice scheduler system"
]
T20: new err VACCX0, VACCX1
--------------------------
CADSK:  770       - 1/-   [A] Convert phys disk addr to virtual addr
[A:
Accepts AC1:    Physical disk address
Returns +1:     always, with Virtual disk address in AC1
]
--------------------------
CFGRP:  325       - 2/-   [76] Create fork group

[76:    Not in '73 doc

Accepts AC1: function bits ,, fork handle
Returns +1: unsuccessful, error # in 1
        +2: successful.

Function Bits:

 All zero: Create a new fork group which includes  the  fork
           in  1  (as  the  top  fork)  and all existing and
           to-be-created  inferior  forks.  The  effect  of
           subsequent  CNDIR JSYS's executed by forks in the
           group is limited to the fork group.
   B0:     For the fork in 1,  return  login  and  connected
           directories,  job number and controlling terminal
           in 1 through 4 respectively. (NOTE that this  is
           the "fork" specific equivalent of the GJINF JSYS)
   B1:     Return in 2  the  top  fork  in  the  group  that
           includes the fork in 1.

Errors:
        FRKHX1:    illegal fork handle
        FRKHX2:    cannot be used to manipulate a superior fork

NOTES:
        A process and its inferiors can be designated as a "fork
group".  All processes in a job are in a single group until some
process in the job executes a CFGRP JSYS.
        A fork can use the Universal fork handle, -6, to specify the
top fork in its group.

]
--------------------------
CFIBF:  100       = 1X    Clears the input buffer (TTYs only)
```

```
Accepts AC1: (10X: file des) (T20: src des)

10X: err DESX6 "string pointer not legal" (not in T20)
T20: new err TTYX01 "Line is not active"
--------------------------
CFOBF:  101     = 1X    Clears the output buffer (TTYs only)

Accepts AC1: (10X: file des) (T20: dest des)

10X: err DESX6 "string pointer not legal" (not in T20)
T20: new err TTYX01 "Line is not active"
--------------------------
CFORK:  152     * 2     Creates an inferior process

Different actions for AC1: B0 CR%MAP:
        10X: inferior gets same address space, but pages are NOT
                shared; changes made by one proc won't be seen by other.
        T20: pages are shared; changes by one proc are seen by other.

10X doesn't hack execute-only processes.

10X: error CFRKX2 - illegal to start a fork unless bit 0 is set with
        CFORK. (I think this means you can't have CR%ST on and CR%MAP
        off in the same CFORK call)  T20 allows this (obviously ok to
        start running in the ACs)
T20: new err FRKHX8
--------------------------
CGRP:   327     - P2/-  [76] Change pie-slice group

[76: (not in '73 doc)
        Change pie-slice group, but not account.

Accepts AC1: SIXBIT group name
Returns +1: unsuccessful, error # in 1
        +2: successful
Errors:
        PIEX1:     Not a pie-slice scheduler system
        CGRPX1:    No such group name
        WHELX1:    WHEEL or OPER special cap not enabled
]
--------------------------
CHANL:  746     ?  [TCP] Set connection channels (Internet/TCP)

[TCP:

Accepts AC1:    Flags,,JCN or Pointer-to-Connection-Block
        AC2:    Six 6-bit bytes (channel numbers)
Returns +1:       failure, error code in 1
        +2:      OK.  This fork will receive TCP PSIs.

      Flags:
        JCNSupplied    (See TCP-JSYS-DOC)

        Each of the 6-bit bytes may be 77 (octal) if no PSIs are desired
for the corresponding event.

      Bits  0- 5:   INTRP channel
      Bits  6-11:   RECV buffer done
      Bits 12-17:   SEND buffer done
      Bits 18-23:   Error
      Bits 24-29:   State change (open or close)
      Bits 30-35:   EOL acknowledged. (Not implemented)

      Note:   PSIs  for the above  may be dropped  or be VERY  tardy  on
              heavily  loaded  systems.  Some defensive  programming  is
              required  to guard against these problems.  See TCPTST.MAC
```

which checks the buffer rings both when a "done" interrupt
occurs and periodically.

]

CHFDB: 64   * p1X   Changes a File Descriptor Block

See "FDB:" entry for changes to FDB.

```
New bit in AC1:
  B0 CF%NUD - (T20: don't wait for disk to be updated) (10X: - )
```

T20: new err DESX7

CHKAC: 521   ! -/2   Checks access to a file

```
[ISI: claims to have a 10X version with the following format:
        E       .CKAAC  Code of desired access to file
        E+1     .CKAPD  Pointer of file being accessed in the
                        left half and number of the directory
                        containing the file in the right half.
        E+2     .CKACL  number of connected directory in left
                        half and logged in directory number in right half.
        E+3     .CKAEC  Enabled capabilities of user whose access
                        to the file is being checked.
]
```

CIS: 141   == 1   Clears the interrupt system

CLOSE: 743   ? [TCP] Close connection (Internet/TCP)

```
[TCP:
Accepts AC1:    Flags,,JCN-or-Pointer
Returns +1:      failure, code in AC1
        +2:      OK, connection fully closed.

      Flags:

         JCNSupplied:   On if RH of 1 has a JCN. Off if RH has
                     Pointer-to-Connection Blk.

         Wait:          Wait for close to happen in both directions.
```

Note that if the wait bit is not set, the JCN will remain valid, allowing more RECVs to be done. ABORT may be used to release the handle when it has been discovered that the connection has terminated.

]

CLOSF: 22   * 2   Closes a file

```
AC1 bit changes:
  B6  CZ%ABT    (T20: abort output operation) (10X: - )
  B7  CZ%NUD    (T20: don't update directory) (10X: - )
```

T20: New errs CLSX3, CLSX4, ENQX20, IOX11
Note that CLSX3: "file still mapped", i.e.
cannot close if still mapped in!!
10X: [76: If a page of the file is PMAPed the CLOSF will succeed, but
file will not be closed; it will be marked to be closed as soon
as the page is unmapped.]

CLZFF: 34   * 1/1X Closes the process' files

```
AC1 bit changes:
  B5 CZ%ARJ (T20: wait until close possible, then close file and release JFN)
            (10X: release JFN, even if file pages mapped into fork)
  B6 CZ%ABT (T20: abort output operation) (10X: - )
  B7 CZ%NUD (T20: don't update directory) (10X: - )

10X: No errs possible, always returns +1
T20: Generates ILINT for FRKHX1, FRKHX2, FRKHX3, IOX11

;Don't use AC1(B5,B6,B7)
--------------------------
CNDIR:   44      - 2/-   Connects job to a directory (replaced by ACCES)

10X calling sequence -

AC1: B0 Check the password but do not connect to the directory
     B1 [ISI: check if job is able to connect to directory but do not
              connect to the directory. ]
        [76: "Proxy" connect; change the login and connected dirs of the
             fork group to the directory in RH(1).  A password must be
             supplied.]
     B2 [ISI: connect the job specified in AC3 to the directory ]
        [76: Connect the fork in RH(4) as specified by the other bits.]
     B18-B35  number of directory being connected to job

AC2: pointer to password string. This password is of the directory
         being connected to and is not needed if the directory either
         does not have a password or is the job's logged-in directory

AC3: [ISI: number of job to be connected to directory. This argument is
         needed only if a job other than the current job is being
         connected to the directory. ]
AC4: [76: see B2]

NOTE:  previous file openings remain valid after a CNDIR.

Returns +1: failed, error # in AC1
        +2: success, string ptr in AC2 NOT updated!
Errors: CNDIX1 - incorrect password
        CNDIX3 - invalid directory #
        CNDIX4 - logged in  (????? -KLH)
        CNDIX5 - not logged in ( One of these is probably wrong! --KLH)
--------------------------
CNTSZ:   605     - 1/-   [A] Count size of job

[A:
        Counts size of job, in number of forks and size of balance
        set.  The latter is the job size considered by the Tenex
        scheduler, and is intended to reflect the current memory
        demands of the job.

Accepts AC1:     A Tenex job number.
Returns +1:      Always, with data in 2:
                  LH - Number of forks for the job.
                  RH - Number of pages in the balance set
                         assigned to the job.
]
--------------------------
COMND:   544     ! -/1X  Parses a command

V4 mods
[ISI:


COMND    544        parses one field of a command input by user
```

Unsuccessful call:
  10X: - not applicable
  T20: "When a field cannot be parsed, B1(CM%NOP) is set in AC1 and one of the following error codes is returned in AC2." (Refer to "Monitor Calls Reference Manual")

.CMDIR - Parse a directory name
  10X: On a successful return, AC2 contains appropriate bits in the left half and the directory number in the right half. The bits are ST%DIR, ST%ANA, or ST%RLM.
  T20: On a successful return, AC2 contains the 36-bit directory number.
  The new syntax of a directory name is:
          STR:<DIRECTORY>
  However, if the structure name is omitted, then the current connected structure is assumed. If the directory name is omitted, then the current connected directory is assumed. It is not possible to perform recognition on the structure name. When specifying a directory name, the brackets cannot be omitted. Both flavors of brackets ("<", ">" and [", "]") are allowed. The bits which used to be returned in the left half of AC2 (ST%DIR, ST%ANA, or ST%RLM) are no longer returned by the COMND JSYS.
]

CPRTF: 33   - 2/-   Changes protection of a file

10X doc claims this is "NOT IMPLEMENTED YET" (like CRJOB).
Doesn't exist in T20. Has it ever existed?

CPUTL: 444   - 1/-   Get CPU utilization for fork/job

[A: Possibly a SRI call?

Return average recent CPU utilization for a process or job.

Accepts AC1:   Fork handle or job number.
Returns +1: always, with:
                Utilization fraction in 2 (floating point).
                Ac's 3 and 4 may be used to return other values at some future date.

A process's "Average CPU Utilization" is a floating-point fraction in the range 0 to 1, denoting the portion of system "Sold Time" that has been charged to that process. (Sold Time is all CPU time actually charged to processes, i.e. excluding idle time and scheduler)overhead). This fraction is maintained as an exponentially-decaying average with a time constant of approximately 2.5 minutes.

If the argument is a multiple fork handle or a job number, the result returned is the sum of the Average CPU Utilization of all the forks referenced.

Note: It is generally not the case that the sum of the CPU utilization for all processes in the system will be exactly 1.0. This is because the Average CPU Utilization is computed at discrete intervals rather than continuously, and at different times for different processes.

]

CRDIR: 240   * p2/p1X   Creates, changes, or deletes a directory
V4 mods

This stuff isn't guaranteed to be complete. This JSYS is rather important, so if you use it you should probably sit down with the actual doc. 10X doc is pp 10-2 to 10-5.

AC1:
10X: Pointer to directory name string
T20: Pointer to ASCIZ string containing the structure and directory name

AC2 bit changes:
B0 CD%LEN (T20: set length of arg blk to .CDLEN value (NOT IMPLEM YET))
(10X: set dir name from param blk (NOT IMPLEM YET))
B2 CD%LIQ (T20: set limit for working storage on disk, from E+2)
(10X: set limit for maximum file storage on disk, from E+2)
B5 CD%LOQ (T20: set limit for permanent storage on disk, from E+5)
(10X: set special resource info, from E+5)
B13 CD%SDQ (T20: set subdir quota from arg blk) (10X: -)
B14 CD%CUG (T20: set user groups assignable by this dir) (10X: -)
B15 CD%DAC (T20: set default acct from arg blk) (10X: -)
B16 - (T20: -) (10X: delete directory)
B17 CD%DEL (T20: delete directory)
(10X: - [76: AC3 contains device designator])

AC3:
10X: Device designator [76: if B17 of AC2 is on]
T20: Pointer to ASCIZ string containing the password of the directory This pointer is required when a nonprivileged user is changing parameters for his directory.

AC4:
T20: -
10X: - [76: string pointer to old password, for changing password when WHL/OPR not enabled]

Returns: T20: +1 always (ILINT if error), dir # in AC1
10X: same as T20
[76: +1 for error (# in AC1), +2 for success (dir # in AC1)]

[76: CRDIR may be used to change the password of a directory if not enabled by supplying a string pointer to the old password in AC4, providing that the old password exists and is not Null.]

The T20 CRDIR JSYS has been extended to accept a full structure/directory name string in AC 1. If no structure name is specified, then the current connected structure is assumed. If the .SFCRD monitor flag is set (SMON) then non-privileged users can change the following entries with CRDIR:

.CDDAC Default account
.CDFPT Default file protection
.CDDPT Directory protection
.CDRET Default retention count
.CDPSW Password (the old password must be pointed to by AC3 to change this value)

It should be noted that the directory number supplied as an argument to the CRDIR JSYS is currently only used to simulate TOPS-10 PPN's consistently across disk reloads. It is expected that this number will ultimately disappear from the CRDIR argument block.

The CRDIR argument block has been expanded to accommodate the new feature of permitting inferior directories to be created.

Extra in T20:
0 .CDLEN LH: flag bits, RH: length of arg block. (10X: -)
15 .CDSDQ maximum number directories that can be created inferior to this directory

```
        16 .CDCUG pointer to user group list
        17 .CDDAC pointer to default account for this user

T20: new errs CRDIX7-9, CRDI10-24
--------------------------
CRJOB:    2      * p2    Creates a new job
V4 mods
```

EA: (CRJOB isn't doc'd in '76 doc and description in '73 doc was unimplemented. This was found lying around other 10X doc.

Just about all functions of T20 CRJOB exist in 10X except for the CJ%CAM, CJ%SLO bits and the arg block entries .CFCPU, .CJCAM, and .CJSLO. They are shuffled around somewhat, though.

```
AC1:
 B0   CJ%LOG    T20 == 10X: log in the new job
 B1   CJ%NAM    T20 == 10X: use name and password from arg block
 B2             10X: On - use account from arg block for the login.
                     Off - use account of caller.
                     Note: B9 overrides B2. The acct must be one which
                        is available to the user being logged in.
 B3             10X: this bit is = T20 B5 CJ%FIL.
 B2-3  CJ%ACT   T20: a 2-bit field:
                        0 .CJUCA Use current acct of caller
                        1 .CJUAA Use acct from arg block
                        2 .CJUDA Use default acct of new user
 B4   CJ%ETF    T20 == 10X: On - put an EXEC above specified file.
                            Off - specified file is top fork of job.
 B5   CJ%FIL    10X: this bit is = T20 B6 CJ%ACS
                T20: On - run the file specified in arg block.
                     Off - just put an EXEC in new job.
                Action if 10X B3,B4 both on differs from T20 CJ%ETF+CJ%FIL.
                10X: run specified file as inferior of EXEC
                T20: mumblage with PRARGS, possibly similar result.
 B6   CJ%ACS    10X: Disown the job.
                T20: Load the ACs from arg blk, if non-EXEC to be run.
 B7   CJ%OWN    10X: this bit = T20 B8 CJ%WTA.
                T20: maintain ownership of job. (negation of 10X B6?)
 B8   CJ%WTA    10X: this bit = T20 B9 CJ%NPW.
                T20: don't start job until it's ATACHed to a terminal.
 B9   CJ%NPW    10X: Use default acct for user being logged in.
                T20: don't check password for new job.
 B10 CJ%NUD     T20 == 10X: don't update login date.
 B11 CJ%SPJ     T20 == 10X: do SPJFN in new job using arg from arg block.
 B12 CJ%CAP     T20 == 10X: set new job's RH caps to current
 B13 CJ%CAM     (10X:-)T20: AND the new job's caps with arg block.
 B14 CJ%SLO     (10X:-)T20: send IPCF message when job logged out.
 B15-16 unused
 B17 CJ%DSN     T20 == 10X: release ownership of job.
 B18-35 must be zero

AC2: address of argument block

Returns +1: unsuccessful, error number in 1
        +2: successful, created job number in 1
Errors: same
```

The argument block is as follows:

```
Word     Sym      Description
0        .CJNAM   10X:    LH - Address of ASCIZ login name to use.
                          RH - Address of ASCIZ password.
                  T20: byte ptr to login name string.
1        .CJPSW   10X: 5B2+n for numeric acct, else  0,,addr of string acct.
                  T20: byte ptr to password string.
```

```
2       .CJACT  10X:    LH - Offset for SFRKV for fork to be run.
                        RH - Addr of ASCIZ file name to be run.
                T20: 5B2+n, or byte ptr to account string.
3       .CJFIL  10X: same as T20 5 .CJTTY
                T20: byte ptr to filename to be run.
4       .CJSFV  10X: CPU limit for new job (0 = none) NOT IMPLEMENTED YET.
                T20: Offset for SFRKV
5       .CJTTY  10X: same as T20 6 .CJTIM
                T20: terminal designator for new job's controlling TTY
6       .CJTIM  10X: same as T20 7 .CJACS
                T20: connect-time limit (0 = none) NOT IMPLEMENTED YET.
7       .CJACS  10X: flags for EXEC AC1 (similar to T20 10 .CJEXF)
                 but flags not described; may or may not be = to T20 flags.
                T20: addr of 16-wd AC block to load.
10      .CJEXF  10X: same as T20 11 .CJPRI
                T20: flag bits for EXEC PRARG block.
11      .CJPRI  10X: -
                T20: primary JFN's for inferiors of new job
12      .CFCPU  10X: -   (shouldn't it be .CJCPU??)
                T20: Runtime limit for job (0 = none).  Apparently works.
13      .CJCAM  10X: -
                T20: Capability mask if CJ%CAM set
14      .CJSLO  10X: -
                T20: IPCF PID #

[ISI: Changes in "owned job" LGOUT notification:
  T20: interrupt with terminal code 33
  10X: interrupt with terminal code 31 ]
]
-------------------------
CRLNM:  502     ! -/p2  Defines or deletes a logical name

-------------------------
CSYNO:   72     - 2/-   Create device name synonym

10X doc says "NOT IMPLEMENTED YET" so who knows?

Creates a new name synonym by which the known device can be
referenced.  The synonym is private to this job and synonyms are
searched first.  Old name and new name may be used interchangeably.

Accepts AC1:    string ptr to known device name, or device designator
        AC2:    string ptr to new device name
Returns +1: failure, error # in AC1
        +2: success, updated string ptrs in AC1, AC2
Errors: DESX1, ASNDX3,
        CSYNX1 - synonym already in use
-------------------------
CVHST:  276     = 2     Converts ARPANET host number to primary name

Accepts AC1:    destination designator
        AC2:    host number
Returns +1: failure (no string for that number)
        +2: success, host name written (is ptr updated?  Doc doesn't say)

T20: err code for +1 failure return is CVHST1 (10X merely implies it)
-------------------------
CVSKT:  275     == 2    Converts ARPANET local socket to absolute form

Accepts AC1:    JFN
Returns +1: failure, error # in AC1
        +2: success, absolute local socket number in AC2
Errors: 600730 CVSKX1   invalid JFN
        600731 CVSKX2   local socket invalid in this context
                        (can't decode JFN name field reasonably)
[T20.4: claims CVHST1 is possible error #.  This is most likely wrong.]
```

```
-------------------------
DBGIM:  677      ? (number conflict, which is right? 677 from MIDAS)
-------------------------
DBGIM:  766      - PN2/- [A] Debug IMP
[A:
        DeBuG IMp jsys.  Copy's current audit buffer into users core

Accepts AC1:     TENEX destination designator
        AC2:     Word count (minimum, transfer may be a few words longer)
        AC3:     non-zero to reinit imp debugging audit buffer pointers
Returns +1:      Unsuccessful, capability error
        +2:      Successful, audit buffer copied negative number of
                         words past word count transfered in 3
Errors:
        all SOUT errors are possible

        WHEEL or NETWIZARD capability is necessary
]
-------------------------
DEBRK:  136      * 1/1X  Dismisses current software interrupt

If no breaks are in progress:
        10X: returns +1
        T20: causes an illegal instuction interrupt (error DBRKX1)
                 (which could be prevented with an ERJMP)

-------------------------
DELCH:  625      - 4/-   [A] Deletes a character

[A:
        To delete character on display terminal.  Current display
        terminals are the IMLAC, TEC, and DATAMEDIA.

Accepts AC1:     a TENEX destination designator
Returns +1:      unsuccessful, destination not a terminal
        +2:      terminal is display, but line character count is 0
        +3:      terminal is display, one character position has been deleted
        +4:      non-display terminal, no action has been taken
Comment:
        This JSYS determines the terminal type by reading the
        terminal type word that is read and set with the GTTYP/STTYP
        JSYS's.
]
-------------------------
DELDF:   67      * p1/p1X        Expunges deleted files

T20
  B0  DD%DTF (T20: delete ;T files)              (10x: - )
  B1  DD%DNF (T20: delete nonopened nonexistent files) (10X: - )
  B2  DD%RST (T20: rebuild symbol table)         (10X: - )
  B3  DD%CHK (T20: check directory consistency) (10X: - )
[ISI:,A:
        All zero - Expunge deleted files
  B12 - (T20: - ) (10X: expunge nonexistent files (FB%NXF and FB%NEX))
  B13 - (T20: - ) (10X: expunge deleted files (FB%DEL))
  B15 - (T20: - ) (10X: expunge scratch files [ISI: and ;T of other jobs])
  B16 - (T20: - ) (10X: expunge temp files [ISI: of this job])
  B17 - (T20: - ) (10X: - [ISI: on device specified by AC2])
]
  B18-B35 - (T20: - reserved for future use and must be zero)
          (10X: directory number)

AC2  (T20: directory number) (10X: not applicable)

T20: Old style DELDF JSYS's with an 18 bit directory
        number  in  the  right half of AC1 will always cause an
        illegal instruction trap.

10X: Always returns +1, no errors; no-op if call illegal in any way.
T20: Returns +1 unless ILINT with several error conditions:
        DELDX1, DELDX2, DELFX2, DELFX4-8
--------------------------
DELF:    26      * p2    Deletes files
V4 mods

AC1 bit changes:
  T20:  B0 DF%NRJ - do not release JFN
        B1 DF%EXP - expunge file
        B2 DF%FGT - expunge file, but leave file address assigned
        B3 DF%DIR - delete & expunge a directory file
        B4 DF%ARC - allow deletion of file with archive status
        B5 DF%CNO - delete/expunge contents but save name & FDB
  10X: only B0 DF%NRJ is implemented.
[ISI:,76: claims that rather than B0,
  10X:  LH(AC1) non-zero means "do not release JFN" ]

T20: new errs DESX7, DESX9, DELFX[2-9], DELF10, DLFX10, DLFX12, WHELX1

;Set AC1(B0) if JFN release is not desired
;Don't use AC1(B1,B2,B3)
--------------------------
DELNF:  317      * 2     Retains specified number of generations of a file
V4 mods

AC1 diffs:
        B4 DF%ARC (T20: same as DELF) (10X: - )
        B5 DF%CNO (T20: same as DELF) (10X: - )

Content change to returning ACs:
  AC2/  (T20: positive count of files deleted)
        (10X: negative count of files deleted)

T20: errs DESX1, DESX3, DESX4, DESX7, DELFX1
10X: Doc doesn't explain what possible errs are!
--------------------------
DEQ:    514      ! -/p2  Removes request from resource queue

--------------------------
DEVST:  121      * 2     Translates a device designator to a string

10X: +1 return leaves error # in AC2 (according to doc)
T20: +1 return leaves error # in AC1

10X: err DESX5 "not open"
T20: err IOX11
--------------------------
DFIN:   234      = 2     Inputs double-precision floating point number

Note: this produces a KA-format double-prec. number.
(T20.4: doesn't say anything about that)
Warning: Although the T20 doc didn't say so, DFIN
        probably has the same EOL lossage as NIN and FLIN (see).
--------------------------
DFOUT:  235      = 2     Outputs double-precision floating point number

No changes to format word!
T20: new err IOX11
--------------------------
DIAG:   530      ! -/PM2 Reserves or releases hardware channels
V4 mods
[ISI: (why this entry? does 10X really have? --KLH)
Format of the device address word:
```

10X	T20	description
0-2	0-2	(10X: 0) (T20: address type)
3-8	3-9	device code
9-23	10-23	0
24-29	24-29	unit
30-35	30-35	subunit

]

DIBE: 212   * 1X   Dismisses until input buffer is empty

If designator not associated with a terminal,
10X: returns on EOF or when file is closed
T20: returns immediately

10X: err DESX6
T20: new err TTYX01

DIC: 133   = 1X   Deactivates software interrupt channels

T20: new err FRKHX8

DIR: 130   = 1X   Disables software interrupt system

T20: new err FRKHX3 (10X doc is probably wrong in not having this too)
plus FRKHX8

DIRST: 41   * 2X/2 Translates a directory number to a string

AC2 changes:
10X: directory number
T20: user or directory number

> There is no change to the calling sequence of the DIRST JSYS. However, the string returned is different. If the number specified is a user number, then the returned string contains just the user name with no punctuation (this is what is always returned in Release 1). If the number is a directory number, then the returned string is in the standard structure/directory name format:
>
> STR:<DIRECTORY>
>
> The structure name and directory name will never be omitted regardless of whether they are the connected structure or the connected directory.

Note different error mechanism!
Return +1:
T20: AC1/ error code (no PSI lossage)
10X: AC1/ unchanged (errors cause ILINT!)

T20: new errs DELFX6, DIRX1, DIRX2, DIRX3, STRX01, IOX11

;Save contents of AC1 if later use is required

DISMS: 167   == 1   Dismisses the process

DOBE: 104   * 1X   Dismisses until output buffer is empty

Accepts AC1: (10X: file des) (T20: dest des)

10X: err DESX6
T20: new err TTYX01

DSKAS: 244   * P2X   Assigns disk addresses

```
This call is quite similar (upward compatible), T20 has extended it
a bit.

AC1
T20: Additional flags in B2-B5 (B0 and B1 exist in 10X)
  B2 DA%CNV convert the specified address according to the setting of B3
  B3 DA%HWA specified addr is a hardware addr (off = software addr)
  B4 DA%INI initialize a private copy of the bit table
  B5 DA%WRT write the private copy of the bit table
        to a new bit table file

AC2
 (10X: - ) (T20: device designator of structure. If DA%CNV
                is on this argument is not required)

-------------------------
DSKCV:  774     - 1/-   [A] Convert hardware disk addr to virtual or v.v.
[A:
        Coverts hardware to virtual or virtual to hardware disk address

Accepts AC1:    Disk address (virtual or hardware)
Returns +1:     Always,  with disk address of opposite type
                        (hardware or virtual) in AC1.
]
-------------------------
DSKOP:  242     * P1X   Specifies disk transfers in hardware terms

This call has quite a few changes; best to look up the relevant
doc if you really plan to use this JSYS.

Content changes to calling ACs:
  AC1/ address (T20 and 10X have different formats)
  AC2/ flags,,word count (T20: B11 & B12 are new flags)
  AC4/  10X: not applicable
        T20:  device designator of the structure. This word is used if
                B2-B10 is -1.
-------------------------
DSMNT:  123     -? 2/-  Dismounts a device

Accepts AC1: dev des
Returns +1: failure, err # in AC1
        +2: success, directory is updated if necessary
Errors: DEVX1, DEVX3,
        DSMX1   Cannot dismount (e.g. files open on device)
        I/O errors can also occur
-------------------------
DTACH:  115     == 1    Detaches a terminal from a job

[76: DTACH is a no-op unless executed by a fork in the top group of
the job.  See CFGRP.]
-------------------------
DTI:    140     * 1X    Deassigns a terminal code

If terminal code was never assigned to process:
        10X: ILINT with DTIX1 error
        T20: no-op
-------------------------
DUMPI:   65     * 2     Reads data in unbuffered data mode

T20: New bit B0(DM%NWT) added in AC2 to specify buffered mode.
        ("do not wait for completion of requested operation")
T20: New errs DUMPX5, DUMPX6
-------------------------
DUMPO:   66     * 2     Writes data in unbuffered data mode

T20: New bit B0(DM%NWT) added in AC2 to specify buffered mode.
        ("do not wait for completion of requested operation")
```

("do not wait for completion of requested operation")

T20: New errs DUMPX5, DUMPX6, IOX11

DVCHR: 117 * 1X Retrieves device characteristics

AC2 bit changes:
  B7 - (10X: DV%MDV device is mountable)
       (T20: -)
AC3 RH value -2: (T20: means device allocator owns device) (10X: -)

B9-17 DV%TYP changes:

Num	10X	T20		Num	10X	T20	
0	DSK	DSK		10	CDR	CDR	card reader
1	DRM	-	drum	11	CDP	FE	frontend
2	MTAn	MTAn		12	TTY	TTY	
3	DTAn	-	dectape	13	TTP	PTY	
4	PTR	-		14	TTR	-	
5	PTP	-		15	NIL	NUL	
6	DSP	-	display	16	NET	NET	
7	LPT	LPT		17	PLT	-	plotter
				20	-	-	
				21	-	CDP	card punch
				22	-	DCN	DECnet active
				23	-	SRV	DECnet passive

EFACT: 5 = P2 Makes an entry in the FACT file
        T20: No-op unless monitor flag SF%FAC is set, since FACT file
                is obsoleted by USAGE file.
        T20: Requires WHEEL/OPER cap
        10X: Requires LOG cap

EIR: 126 = 1X Enables software interrupt system

T20: new err FRKHX3 (10X doc is probably wrong in not having this too)
        plus FRKHX8

ENQ: 513 ! -/p2 Places request in resource queue

[ISI: (why this entry? 10X really have it? --KLH)

.ENQLN (first word of argument block)
  10X: "number of requested locks in the left half and length
          of argument block in the right half."
  T20:   "Length of the header and the number of requested locks in the
          left half, and length of argument block in the right half."
.ENQLV
  B2 - (10X: -) (T20: allow ownership of the lock to be nested to
                  any level)
  B3 - (10X: -) (T20: allow a long-term lock on this resource)
.ENQUC
  (10X: the address of an ASCIZ string)
  (T20:   a byte pointer to a string of any size byte. Byte size
          is specified by pointer.

Argument Block - T20:extra word added - "address of a resource mask block"
]

ENQC: 515 ! -/p2 Obtains status of resource queue

EPCAP: 151 == 1X Enables process capabilities

ERSTR: 11 * 3/3X Converts error number to string

Note: Page 3-5 of 10X doc has interesting stuff about 10X error message handling (data file format, etc)

T20: if error # is specified in RH(AC2), LH should contain .FHSLF.

```
RH(AC3) T20: must be 0
        10X: B18 off - expand parameter typeout commands
                 on  - don't
             B19 off - use 5 words in PSB of designated fork
                 on  - use 5 words from ACs 4-10
10X: AC4-AC10 may optionally be used.
T20: -
```

T20: Gets ILINT on errors DESX1, FRKHX1, IOX11. Fuck.
10X: Apparently generates no interrupts nor returns error codes; just returns to +2 for any errors other than "undefined error number".

ESOUT: 313 == 1XQD Outputs an error string

Neither 10X nor T20 doc details possible errors, but obviously similar to PSOUT.

EXEC: 777 =? 1 [A] Enter mini-EXEC
Same as T20 MDDT?
[A:

Enter mini-EXEC. Equivalent (almost) to a quit from a top level EXEC.

Returns +1: Always
Errors: none
^P is not enabled, and the jsys returns only if the "^" to the mini-exec is given. WHEEL or OPERATOR capability is required.
]

FDFRE: 213 - 1X/- File directory free space

```
Accepts AC1: dev des (only DSK legal)
        AC2: directory number
Returns +1: always, with number in AC2 representing amount of free space
            left. For disk dirs, this amt is returned as a number
            of words.
Errors: ILINT on DESX1,DESX2,
        FDFRX1  not a multiple dir device (not DSK)
        FDFRX2  no such dir number
```

FFFFP: 31 * 1X Finds first free page in file
V4 mods

AC1 LH: (T20: starting page number) (10X: 0)

FFORK: 154 = 1X Freezes processes

T20: new err FRKHX3

FFUFP: 211 = 2 Finds first used page in file

T20: new err DESX7

FLHST: 277 * PN1/PN1X Flushes an ARPANET host

Flushes NCP tables of info for that host, and sends host-host RST command to that host.

Accepts AC1: host number
Returns +1: always (unless T20 and not enabled)

```
If process does not have WHL/OPR/NWZ cap enabled,
10X: no-op
T20: ILINT (error code not given in doc)
--------------------------
FLIN:   232      * 2      Inputs floating-point number

T20: If terminating char was a CR followed by a LF, the LF is also
        input.  Potential screw.
10X: not applicable since 10X EOL is single char (sigh)
--------------------------
FLOUT:  233      = 2      Outputs floating-point number

No changes to format word!
T20: new err IOX11
--------------------------
GACCT:  546      ! -/p1X Gets current account designator

--------------------------
GACTF:   37      = 3      Gets account designator of file

[76: claims that 10X AC2 is not updated, i.e. it points to 1st char
of returned string, rather than at end as '73 10X doc and T20 doc imply.]

T20: new errs DESX7, GACTX3
--------------------------
GACTJ:  333      - 2/-   [76] Get account for job

[76:  (not in '73 doc)

Accepts AC1: Address of 8 word block for string account
        AC2: job # (-1 for self)
Returns +1: Unsuccessful, error # in 1
        +2: Successful, account designator in 1
Errors:
        GCTJX1 Invalid job #
        GCTJX2    Job doesn't exist
]
--------------------------
GCVEC:  300      = 1X     Gets entry vector of compatibility package

T20: if package was disabled, AC2 contains -1 on return; if
        package not available, AC2 contains 0.
        Not sure if 10X does same thing or not.
--------------------------
GDACC:  331      - 2/-   [76] Get default account for user

[76: (not in '73 doc)

Accepts AC1: Address of an 8 word block for a string
               account
        AC2: user directory # (-1 for self)
Returns +1: unsuccessful, error # in 1
        +2: Successful, account designator in 1
Errors:
        GDACX1    User name not in account matrix
        GDACX2    No default for this user
        ACCTX1    No user/account data file
]
--------------------------
GDSKC:  214      * 1X     Gets disk count

AC1:
        10X: only 777777 for DSK: allowed
        T20: must be designator for a structure. If DSK: given,
                connected structure is assumed.
T20: err DEVX1
```

```
Neither doc says how error is handled, but implication is ILINT.
--------------------------
GDSTS:  145      = 1X    Gets device's status

The GDSTS call works in exactly the same way.  However the values
returned (AC2 dev status bits, AC3 dev values) may be different
for some devices.

NET is same.
MTA AC3:
        10X: word count of last xfer completed, negative if last xfer
                attempt failed.
                [76: above value is in LH!!]
        T20: byte count of xfer(s?) in LH, 0 in RH.
LPT AC3:
        T20: last value of page count register (0 if no p.c.r.)

--------------------------
GDVEC:  542      ! -/1X  Gets entry vector of RMS

--------------------------
GET:    200      * 1X    Gets a save file
V4 mods

T20: AC1 new T20 bits:
  B19 GT%ADR use the memory address limits given in AC2
  B20 GT%PRL preload pages being mapped
  B21 GT%NOV do not overlay existing pages
  B22 GT%FL2 read additional flag bits specified in AC3 (reserved for
        future development)

AC2 (10X - )  (T20 - See B19 AC1.  Lowest page number in left half,
                                    Highest page number in right half)

T20: new errs OPNX2, GETX3 "illegal to overlay existing pages" (when
        GT%NOV is set)
--------------------------
GETAB:   10      * 2     Gets a word from a monitor table

T20: If -1 is given as index, returned value is negative length of
        specified table.
10X: no mention [76: same as T20]

T20: new err GTABX3 "GETAB privileges required"!  Process must have
        the SC%GTB GETAB capability (although not necessary to enable it).
10X: - [76: has GTABX3 error too]
--------------------------
GETER:   12      * 1     Returns the last error in a process

10X: Also returns 5 PSB parameters in AC4-AC10
T20: -

10X: no errors possible? (wonder what happens with bad fork handle)
T20: error code LSTRX1 "process has not encountered any errors".
        Doesn't explain if this is returned as "the" error number
        or if this causes an ILINT (won't surprise me)
--------------------------
GETJI:  507      ! -/2   Gets specified job information
V4 adds

[ISI:   (why this entry?  10X really have it? --KLH)

word 3 .JIDNO    10X: Job's directory number
                 T20:  Job's connected directory number

word 17 .JILNO   T20: job's logged in directory number (10X: - )
word 20 .JISRM  T20: pointer to job's session remark (10X: - )
word 21 .JILLN  T20: job's last login date and time (10X: - )

AC2 on return:
  10X: -
  T20:  AC2 is "updated on a successful return and cannot be used
        for the returned data."
]
--------------------------
GETNM:  177      == 1    Returns the program name currently being used

--------------------------
GETOK%: 574      ! -/1X  Requests access to a protected resource
V4 addition
--------------------------
GEVEC:  205      == 1X   Gets entry vector

--------------------------
GFACC:  335      - 2/-   [76] Determine access to directory or file

[76:  This JSYS appears in '76 doc.
Accepts AC1: LH: flags
             B0 Accept file protection in 2,  return  file
                   access in 2.
             B1 Accept directory number in 3, do  proxy  ;
                   gfacc    (requires    WHEEL    or    OPERATOR
                   capability enabled).
             RH: Directory  to  which   access   is   being
                   checked.

        AC2: File protection (500000 in LH) if B0 AC1 on.
        AC3: Directory number of user whose access is being
               checked if B1 AC1 on.

Returns +1: unsuccessful, error number in 1
        +2: success
              AC1: Access available to directory in B30-35
                        B30: Can reference directory
                        B31: Can open files subject to file protection
                        B32: Can connect without password
                        B33: Can create new files in directory
                        B34, B35: Unused
              AC2: If B0 on in call AC1, access available to
                   file whose protection was given in AC2
                        B30: Read
                        B31: Write
                        B32: Execute
                        B33: Append
                        B34: Access per page table
                        B35: Unused
                   Zero is returned if  directory  access  is
                   such   that  files  cannot  be  referenced
                   regardless of file protection.

If B1 is off, this JSYS determines the access that the executing fork
has to a specified directory and (optionally) to files with a given
protection in that directory.  Owner access is returned if the fork is
connected to the specified directory, and all access (77) is returned
if the executing fork has WHEEL or OPERATOR capability enabled.

If B1 is on, the access checked is that of an unprivileged process
logged in under (and connected to) the directory specified in AC3.
this option is intended for use by privileged programs which must
perform access checking interpretively.

GFACC ERROR MNEMONICS:
```

```
GFACX1:    No such directory (AC1 or AC3)
CHKAX2:    Illegal protection word (at present, not 500000000000+prot)
WHELX1:    WHEEL, OPERATOR or MAINTENANCE capability not enabled
]
--------------------------
GFRKH:  164     * 2/2X  Gets process handle

AC2: relative handle (T20: 400001-400777) (10X: 400001-400030)
        i.e. 10X doesn't allow as many relative handles.

10X: returns +1 for err, +2 if win
T20: claims ditto, but also says gets ILINT on errors???!!!
--------------------------
GFRKS:  166     * 2/2X  Gets process structure

Content changes to calling ACs:
  AC2/  (T20: bits only (currently B0 GF%GFH, B1 GF%GFS))
        (10X: same bits,,table location)
  AC3/  (T20: -<max table length>,,table location)
        (10X: - )

Return: (T20: +1 if error, +2 if win, but claims ILINT???!!!)
        (10X: ? doc says +1 only, but also says "NOT IMPLEMENTED YET"!)
        [76: says +1 if err, +2 if win]

Table entry:
        T20: 3 words always, 3rd is -1 if GF%GPS is off
        10X: doc sez 2 words if GF%PGS is off.

;Always set up all three ACs
;Don't use AC2(B0) if the starting point is a superior fork of the
; executing fork

--------------------------
GFUST:  550     ! -/1X  Returns author and last writer name strings

--------------------------
GIVOK%: 576     ! -/P1X Grants access to a protected resource
V4 addition
--------------------------
GJINF:   13     = 1     Gets current job information

Minor diff in AC1:
  10X: directory number under which job was logged in
  T20: user number under which job is running

AC2: T20 returns a full 36 bit directory num which represents the
        connected structure and directory pair.
--------------------------
GNJFN:   17     * 2     Gets the next JFN

Successful return:
  AC1 extra flag setting:
    B13 GN%STR (T20: structure changed) (10X: -)
        10X: [76: GN%EXT not meaningful if GN%NAM or GN%DIR are set.
                likewise GN%NAM not meaningful if GN%DIR is set. ]

        T20: Error if previous file still open (close it before the call!)
        10X: GNJFN closes the previous file [76: maybe not??]
Different error interpretation of GNJFX1:
        T20: no more files in this specification
        10X: cannot close the file
PSI lossage:
        T20: errors take +1 return (GNJFX1 means all done)
                (Q: is err code returned in AC1??)
```

```
10X: errors cause ILINT, apparently +1 taken only when all done
--------------------------
GPJFN:  206      == 1X   Gets the primary JFNs

T20: returns -1 in AC2 if no SPJFN has been done.  Not clear if 10X
        does the same thing.
--------------------------
GPLD:   337      - p2/-  [76] Get pie-slice group load average

[76:  This call appears in '76 manual.

Accepts AC1: a pie slice group handle, which is either -1, indicating
                the caller's own current group, or an explicit group
                index, for which wheel or operator capability is required.
Returns +1: error (# in AC1?)
        +2: successful, with the group load average in AC2
                in floating point format.
Errors:
        GPLDX1  explicit group index supplied and out of range
        CAPX1   explicit group index supplied and caller not  wheel
              or operator.
]
--------------------------
GPSGN:  334      - 2/-   [76] Get pie-slice group name for job

[76:  (not in '73 doc)

Accepts AC1: TENEX destination designator
        AC2: job # (-1 for self)
Returns +1: unsuccessful, error # in 1
        +2: successful, updated string pointer in 1 (if pertinent)
Errors:
        PIEX1     Not a pie-slice scheduler system
        GCTJX1    Invalid job #
        GCTJX2    Job doesn't exist
        ACCTX1    No user/account data file
]
--------------------------
GTABS:  105      -? 1X/- Get tab settings for file

[MT: not in VTS version of FOONEX]
Accepts AC1: file des
Returns +1: always, with tab settings in AC2, AC3, AC4
Errors: ILINT for DESX1, DESX3, DESX5, DESX6, DEVX2

These ACs are interpreted as a 107-bit string, with B0 of AC2 ignored,
and each remaining bit indicating the presence of a tab in the
corresponding column.
        B1 of AC2 for col. 1
        B0 of AC3 for col. 36
        B0 of AC4 for col. 72
Returns tab every 8 places if designator associated with non-terminal.

--------------------------
GTAD:   227      * 1     Gets current date and time

Identical except for timeword format.  See the TIMEWORD entry.
--------------------------
GTBLT:  634      - 2/-   [A] Get system tables with BLT

[MT: SRI addition, in MIT T20 and FOONEX, maybe others?]

Get systab tables with BLT.  BLT's a table listed in SYSTAB into user's
virtual core.

Accepts AC1: LH Starting index or -1 for entire table
```

```
            RH Table number
        AC2: LH Number of entries to move
             RH Buffer address in user space
Returns +1: failure, error # in AC1
        +2: success, table copied
Errors:
        GTABX1 Illegal table number
        GTABX2 Illegal index
        GTABX3 GETAB capability required (as for plain GETAB!)
-------------------------
GTDAL:  305     * 1X    Gets disk allocation of a directory

Content changes to calling ACs:
  AC1/ (T20: positive 36 bit directory number, or -1 for connected directory)
       (10X: positive directory number, or 0 for connected directory)

Returns AC1: (T20: working disk storage limit) (10X: allocated limit)
        AC2: # of pages currently used
        AC3: (T20: permanent disk storage limit) (10X: - )

T20: ILINT on errors DIRX1, DELFX6
10X: No errors given in doc

[ISI: Always use -1 for connected directory]    ; will this work on 10X?

-------------------------
GTDIR:  241     * P1X/p1X       Gets information of directory entry
```

The calling sequence (plus errs) is the same; the argument (parameter) block is slightly different. On T20 it has been expanded to accommodate the new feature of permitting inferior directories to be created. See the CRDIR entry for a description of the changes.

```
T20: Checks .CDLEN (0) to find length of arg block
10X: Assumes length 15 (octal)

T20: If directory number in AC1 is zero, GTDIR returns default settings
        for these parameters: .CDLIQ .CDLOQ .CDFPT .CDDPT .CDRET .CDSDQ
        .CDDNE .CDDFE
10X: ?

10X: Fails unless WHL/OPR enabled
T20: Allows use without WHL/OPR if caller:
        1) is connected to an immediately superior directory
        2) has owner access to directory
    Will never return password string unless WHL/OPR enabled.
-------------------------
GTFDB:   63     * 1X    Gets a File Descriptor Block
```

This JSYS is the same, but see "FDB:" entry for changes to FDB format.

```
T20: LH(AC3) must be zero  [ISI note] (well, naturally...)
T20: new err DESX7
-------------------------
GTHST:  273     = 2     Get ARPANET hostname information (new)
```

This is a new call introduced for long-leader NCP operation. Not all sites may have it, but apparently those which do all implement it in the same way.

```
Accepts AC1: function code
        AC2,3,4: function-specific args
Returns +1: failure, error # in AC1
        +2: success, function-specific data returned in AC's

Code            Function
```

```
0 .GTHSZ          Gets general table data
                  args: none
                  rets:   AC2:    -<number of host names>,,0
                          AC3:    -<Length of HSTSTS table>,,0
                          AC4:    local host number (32-bit Internet fmt)
1 .GTHIX          Gets data for index.  If name is a nickname, HS%NCK is on
                  in the status word.
                  args:   AC1: destination byte ptr
                          AC3: index into name table (returned by GETAB)
                  rets:   AC2: updated byte ptr
                          AC3: host number
                          AC4: host status
2 .GTHNS  Gets primary name for host number.
                  args:   AC2: dest byte ptr
                          AC3: host number
                  rets:   AC2: updated byte ptr
                          AC3: host number
                          AC4: host status
3 .GTHSN  Gets number for name.  If name is nickname, HS%NCK will be on in
                  the status word.
                  args:   AC2: source byte ptr
                  rets:   AC2: updated byte ptr
                          AC3: host number
                          AC4: host status
4 .GTHHN  Get current status of host.
                  args:   AC3: host number
                  rets:   AC3: host number
                          AC4: host status
5 .GTHHI  Get number/status for HSTSTS index.
                  args:   AC3: index into HSTSTS (returned by GETAB)
                  rets:   AC3: host number
                          AC4: host status
```

Flags in host status word:

Bits	Symbol	Meaning
1B0	HS%UP	Host is up
1B1	HS%VAL	Valid status
7B4	HS%DAY	Day when up if currently down
37B9	HS%HR	Hour
17B13	HS%MIN	5 minute interval
17B17	HS%RSN	Reason
1B18	HS%SRV	Host is server
1B19	HS%USR	Host is user
1B20	HS%NCK	Nickname
77B26	HS%STY	System type mask
1B27	HS%NEW	RAS, RAR, RAP, etc
1B26	.HS10X	TENEX
2B26	.HSITS	ITS
3B26	.HSDEC	TOPS-10
4B26	.HSTIP	TIP
5B26	.HSMTP	MTIP
6B26	.HSELF	ELF
7B26	.HSANT	ANTS
10B26	.HSMLT	MULTICS
11B26	.HST20	TOPS-20
12B26	.HSUNX	UNIX

GTHST% ERROR MNEMONICS:

ARGX02:	Invalid function
GTHSX1:	Unknown host number
GTHSX2:	No number for that host name
GTHSX3:	No string for that host number
GTJIX1:	Invalid index

GTJFN: 20 * 2 Gets a JFN

This is one of the most complex and obscure calls. The 10X version is said to be riddled with bugs, and neither 10X nor T20 documents it very well at all.

T20 doc mumbles "All I/O errors can occur. These errors cause software interrupts or process terminations, and only a single (+1) return is given" -- what the fuck does this mean, does the call really ever produce an ILINT in spite of the existence of an error return????

T20: Long-form GTJFN allows extended arg block (words 11-16, .GJF2-.GJATR)
10X: .GJACT can be either string or numeric.
Doc implies .GJPRO is numeric only.
T20 doesn't mention numeric possibility for either.
Numeric arg is given as 5B2+N.

AC1 bit changes:
B6 GJ%NS (T20: use 1st specification of multiple logical name definition)
(10X: retype file name when complete)
B12 GJ%OFG (T20: will not recognize an output file after <escape>,
but instead builds file name as specified up to <escape>)
(10X: will recognize an output file after <escape>)
B14 GJ%PHY (T20: Ignore logical names, use specified physical device)
(10X: - [76: Scratch (;S)])
B15 GJ%XTN (T20: for long call GTJFN, addtional words at E+11)
(10X: -)

Version-number values are exactly the same (-1, -2, etc).
Returned flags in LH are the same except:
B15 (T20: -) (10X: - [76: ;S given])
B17 GJ%INV (10X: -) (T20: Invisible files were not considered)

T20: new wildcard character %, not in 10X. [ISI: says % not allowed in device field.]

Errors:
GJFX24 10X: "no new files"
T20: "file not found"
GJFX25 10X: "non-null name used with non-directory device" (T20: -)
GJFX26 10X: "non-null ext used with non-dir dev" (T20: -)
T20: new errs GJFX37-49, IOX11, DESX9, STRX09
Confirming chars:
10X: TAB LF FF CR ESC EOL SP , @ _
T20: LF FF ^Z CR ! " # & ' ( ) + , / = @ SP ESC
Terminating chars:
10X: unclear. Conf chars plus "=" is all doc will say.
T20: all the T20 confirming chars plus ESC

T20: if terminating char is also a confirming char, a confirming message will NOT be typed to user, nor will user be required to confirm the string obtained, regardless of setting of GJ%MSG and GJ%CFM.
10X: appears to insist on confirmation anyway. (Boooo!)

Characters allowed in filename identifiers (dev, dir, name, ext):
10X: 40-137 except .:;<>* and _,=@ and SP (according to doc)
T20: there is NO desciption of this!
The doc is pretty bad here. Empirical observation suggests:
10X: 41-137 except :;<> and @&_? (& is really weird)
T20: A-Z, 0-9, $ - _ (yes just three non-alphanumerics!)
All non-allowed chars must be quoted with ctl-V.

[ISI:
<Carriage-return> termination:
T20: next character will be read (because a <Line-feed> is expected)
and it will be seen as the terminator (regardless of what it is)

```
  10X:  no further input is required (or done).

Recognition of default extension:
  T20:  If a default is specified, will not recognize any file
        extension except the default after <escape>.
  10X:  Will recognize a unique file extension regardless of the
        specified default.
]
```

```
GTNCP:  272      = 2      Get NCP information (new)
```

This is a new call introduced for long-leader NCP operation. Not all sites may have it, but apparently those which do all implement it in the same way.

```
Accepts AC1: function code
        AC2,3,4: function-specific args
Returns +1: failure, error # in AC1
        +2: success, function-specific data returned in AC's

Code    Function
 0 .GTNSZ  Gets (negative) # of NCP connections
                args: none
                rets:   AC2: -<number of NCP connections>,,0
                        AC3: -<number of NVTs>,,<line number of first NVT>
 1 .GTNIX  Get connection number status
                args:   AC2: connection number
                        AC3: 30 bit address of storage block
                        AC4: -<length of block>,,<index of 1st item to get>
                rets: data in block (see format)
The following are exactly like .GTNIX above except for the particular
        type of argument furnished.
 2 .GTNNI  Get status of NVT (input)    AC2: NVT line number (input connection)
 3 .GTNNO  Get status of NVT (output)   AC2: NVT line # (output connection)
 4 .GTNJF  Get status of JFN            AC2: JFN
```

Format of returned data block:

Word	Symbol	Contents
0	.NCIDX	NCP connection index
1	.NCFHS	Foreign host
2	.NCLSK	Local socket
3	.NCFSK	Foreign socket
4	.NCFSM	State of connection
5	.NCLNK	Link
6	.NCNVT	NVT, -1 if none
7	.NCSIZ	Byte size of connection
10	.NCMSG	MSG allocation
11	.NCBAL	Bit allocation
12	.NCDAL	Desired allocation
13	.NCBTC	Bits transferred
14	.NCBPB	Bytes per buffer
15	.NCCLK	Time-out countdown
16	.NCSTS	Connection status

```
Errors: ARGX02: Invalid function
        GTJIX1: Invalid index
        GTNCX1: Invalid network JFN
        GTNCX2: Invalid or inactive NVT
```

```
GTRPI:  172      == 1X    Get trap information
```

```
GTRPW:  171      * 1X     Gets trap words

10X: no errors documented
T20: ILINT on errs FRKHX1,2,3
```

```
Content changes to returning ACs:
  AC1/  Bits B14-17 are the same (TSW%RD TSW%WT TSW%EX TSW%MN)
        (T20:   B0 PF%USR is complement of B17 TSW%MN
                B5 PF%WRT is same as B15 TSW%WT
                B1-B4,B6-B13 are not documented)
        (10X: B0-B13 are ghastly bits described in 10X doc, page 5-25,26)
  AC2/  (T20: last monitor call with an error)
        (10X: write data)
```

On 10X, a trap on a read or execute ref will trap with the PC
pointing to the guilty instruction, but if it was a WRITE reference,
the PC will already have been incremented or otherwise changed.
In order to proceed, the "write data" word must be stored into
the address so as to complete the interrupted instruction.
[MT: on F5 and possibly other F machines, address will be indeterminate
(may or may not be +1).]

I believe that T20 does the right thing, i.e. traps with PC pointing
to guilty instruction no matter what it is, so it will be re-tried
when the process is continued. For this reason no "write data" value
needs to be returned, and T20 uses AC2 for some random information.

```
-------------------------
GTSIG:  730     - 2     [BBN76] Get signal ID
```

[76: experimental BBN call, p. 11-7]

```
-------------------------
GTSTS:   24     = 1     Gets a file's status

AC2 bits returned:
  B5  - (T20: -)(10X: ok to access as specified by page table)
  B16 - (T20: -)(10X: ok to change byte size)
  B14 - (T20: -)(10X: - [76: open thawed])
  B11 GS%AST T20: the JFN is parse-only (GJ%OFG was set in GTJFN call)
             10X: a * was typed in one of the filename fields
  B17 GS%FRK T20: This is a restricted JFN
             10X: file is restricted to some fork (open for restricted acc)
  B18 GS%PLN T20: 1 = line numbers are passed to program during input
                  0 = line numbers are stripped
             10X: -
-------------------------
GTTYP:  303     = 1X    Gets the terminal type number (TTYs only)
```

10X: Doc doesn't say, but assume it gets ILINT on DESX1
T20: new err TTYX01

```
-------------------------
HALTF:  170     = 1     Halts the current process

If HALTF executed at top level process:
        10X: If WHL/OPR not enabled, process is replaced by EXEC and started
                at initialization entry.
        T20: If WHL/OPR not enabled, job is logged out.
        Both: if WHL/OPR enabled, control passes to mini-exec.
-------------------------
HANDS:  700     - 2/-   [A] Get multiple monitor tables

[A:     Possibly a SUMEX/IMSSS call?

        Returns multiple monitor tables to the user address space.

Accepts AC1: -length of argument table,,address of argument table
                entries in the argument table are
                of the form <table number>,,<place to store table>
Returns +1: error if no such table, or table not implemented
        +2: success, all tables transferred completely
Errors:
    HNDRNG  table number out of range
    HNDEMP  table empty or not implemented
]
-------------------------
HFORK:  162     * 1X    Halts a process

10X: err FRKHX5 "fork already halted"  (T20: no-op)
T20: new err HFRHX1 "illegal to halt self with HFORK" (10X: can halt self)
-------------------------
HPTIM:  501     ! -/2   Returns values of high precision clocks

-------------------------
HSYS:   307     * PM2   Halts the system

10X: Putting 0 in AC1 will cancel pending shutdown.
T20: doc doesn't mention this.

10X: [76: also accepts
    AC3: A 4 bit number the meanings of which are defined by
        BBN Report #1822, page 3-14 "Table of Host
        going down messages".
        The only ones which have meanings are:
            5 Preventive Maintenance
            6 Hardware
            7 Software Maintenance
            8 Emergency Restart
]
10X: return sequence not documented, nor are errors!
T20: Returns +1 if failure (err code in AC1), +2 if success.
    errs CAPX2, TIMEX1, TIMEX2

-------------------------
IDCNV:  223     * 2     Inputs date and time conversion

T20: extra flag (AC4: B3 IC%JUD) - interpret # in RH(AC2) as being
    in Julian day format (Jan 1 is day 1).
-------------------------
IDTIM:  221     == 2    Inputs date and time

-------------------------
IDTNC:  231     * 2     Inputs date/time without converting

T20: extra flag (AC4: B3 IC%JUD) - a number in Julian day format was input
-------------------------
IIC:    132     = 1X    Initiates software interrupts on specified channels

T20: new err FRKHX2 "illegal to manipulate a superior process"
    (10X doc may be wrong in not having this too,
    but possibly 10X WANTS to allow ints from inferior)
    plus FRKHX8
-------------------------
IIT:    630     - 1X    [A] Initiate interrupt with timing delay
[A:
    Initiate delayed pseudo-interupt on specified channels in a
    fork.  This is like the IIC JSYS with the delay timing added.

Accepts AC1:    a fork handle
    AC2:        a 36-bit word, b0 for channel 0, b1 for
                    channel 1, etc.
    AC3:        milliseconds to wait to initiate interupt

Returns +1:     always

    There is no promise of other than gross accuracy for the
timing of this JSYS.  It is subject to the vagaries of the scheduler
and will not be accurate within milliseconds.
```

```
--------------------------
INIDR:  124     -? 2    Initializes device directory

Accepts AC1: dev des
Returns +1: failure, err # in AC1
        +2: success, directory initialized
Errors: DEVX1, DEVX3,
        INIDX1  Device busy
        I/O errors can also occur
--------------------------
INLNM:  503      ! -/2   Lists job's logical names

--------------------------
JFNS:    30      * 1X    Translates a JFN to a string

Additional Input:
  AC2 (10X: can only be file handle) (T20: can also be pointer to string)
  AC4 (10X: -)(T20: pointer to string containing prefix of file
        specification attribute)

AC3 additional flag settings for T20:
  B26 JSXPTR - AC2 contains pointer to string to be returned
  B27 JS%ATR - return file specification attributes if appropriate
  B28 JS%AT1 - return the specific specification attribute whose prefix is
                indicated by the string pointed to in AC4
  B29 JS%OFL - return the "OFFLINE" attribute

10X: has err DESX5 "not open" (??) (not in T20)
T20: new err IOX11
--------------------------
JOBTM:  316      - 1/-   Get job runtime

This JSYS seems to be almost the same as RUNTM with a -5 argument.
Accepts no args
Returns +1: AC1 contains runtime in milliseconds of current job.
--------------------------
KFORK:  153      = 1X    Kills a process

T20: Doc notes that KFORK will not release a handle for proc that
has already been killed by another proc; RFRKH must be used to
release handle.  Not sure if 10X is same here.

T20: new err FRKHX3
--------------------------
LGOUT:    3      = 2     Kills a job
        T20: Add'l errs LOUTX3, LOUTX4, LOUTX5
        T20: Can require WHL/OPR,LOG caps
        10X:  "      "    LOG caps

[76: a LGOUT with AC1 = -1 (logout of self) will fail unless fork
executing it is in the top fork group of job.  See CFGRP.]
--------------------------
LITES:  215      -? PM1X/-      Displays data in console lights

Accepts AC1: 36 bit word to be displayed in the MI lights on CPU
                console.
Returns: +1: success
Errors: ILINT on WHELX1 (WHL/OPR/MNT cap needed)
--------------------------
LNMST:  504      ! -/2   Converts a logical name to a string

--------------------------
LOGIN:    1      = 2     Logs in a job

T20: add'l errs LGINX6 and LGINX3
```

```
-------------------------
LPINI:  547     ! -/P1X Loads VFU or translation RAM

-------------------------
MDDT%:  777     = -/P1X Enter Monitor DDT
MDDT:

In T20 and 10X, require WHL/OPR cap enabled.
-------------------------
METER%: 766     ! -/1X  Returns EBOX/MBOX clock values
V4 addition
-------------------------
MOUNT:  122     -? 2/-  Mounts a device

Mountable devices such as DECtapes must be MOUNTed before being
accessed.
Accepts AC1: dev des  (if B3 on = read directory; off = don't read)
Returns +1: failure, error # in AC1
        +2: success

If device is already mounted, it is first dismounted.
If B3 of AC1 is on, a DECtape or magtape is assumed to be non-directory.

Errors: DEVX1,2,
        MNTX1   Illegal directory fmt and dir read specified
        MNTX2   Failed to mount (e.g. device off-line)
        MNTX3   Device type not mountable
        DSMX1   Failed to dismount (see DSMNT)
I/O errors can also occur.
-------------------------
MRECV:  511     ! -/p2  Receives an IPCF message
V4 mods

[ISI: (this seems to imply that 10X has some flavor of IPCF??? --KLH)
Packet descriptor block:
word 4  .IPCFD  10X: LH - connected dir of sender, RH - logged in dir of sender
                T20: 36 bit user number of sender
words 6-10 (.IPCSD, .IPCAS, .IPCLL) don't exist on 10X)
]
-------------------------
MRPAC:  772     ? 1     Returns access of resident monitor

[A: (listed in MONSYM)

Accepts AC1:    Page address
Returns +1:     Always, with access information in AC2

                B2  read access allowed
                B3  write access allowed
                B4  execute access allowed
                B5  page exists
                B10 private
]
-------------------------
MSEND:  510     ! -/p2  Sends an IPCF message
V4 adds
-------------------------
MSFRK:  312     * P1X   Starts a process in monitor mode

Calling sequence is the same, but the context setup is a bit
different:
        FBGN:   MOVSI 1,UMODF   ; A fake user-mode PC
                MOVEM 1,FPC     ; Simulate a JSYS call
                (10X: JSYS MENTR)       ; Establish usual JSYS context
                (T20: MCENTR)
```

```
T20: errs FRKHX1,2,3,CAPX1
10X: Doc doesnt describe errs, but refers to SFORK description.
-------------------------
MSTR:   555     ! -/p1X Performs structure-dependent functions
V4 mods
-------------------------
MTALN:  774     ! -/P1X Associates magnetic tape drive with logical unit numbe

-------------------------
MTOPR:   77     * 1X    Performs device-dependent functions
V4 mods

The 10X doc for MTOPR is really bad and incomplete.  AC3 is not
specifically shown as being used, but some functions do use it as
a function arg (like T20).  The only devices
documented are MTA, DTA and NET.  All other MTOPR operations on
10X can be considered non-existent or undocumented.
For MTA: All 10X functions supported by T20.  These are .MOxxx where
        xxx is: 1 REW, 3 EOF, 6 FWR, 7 BKR, 10 EOT, 11 RUL, 13 ERS,
                16 FWF, 17 BKF
For DTA: (Dectape -- don't think T20 supports this)
        operations are: 1 rewind, 11 Rewind & flap, 30 Use block # in AC3
                for next DUMPI/O.
For NET: All 10X functions supported by T20.  Functions are
        20 .MOACP, 21 .MOSND, 22 .MOSIN, 24 .MOAIN (yes, arg is = 10X)
For TTY: none
        [MT: T20 operations will be added to FOONEX]

[ ISI STUFF:

LPT function has extra flag in code 37 (read status of line printer)
  that indicates front end has been reloaded

CDR function has extra flag in software status word that indicates if
  front end has been reloaded

PTY function (code 24-definition for software interrupt channel
  number for Input/Output reversed from 10X)
  10X: "the channel number used for output is one greater than
           the input channel number"
  T20:     "the channel number used for input from the PTY is one
           greater than the channel number used for output to the PTY"

TTY function, codes 30-36 are extra. They have to do with page
  width and length and terminal line status
] ;ISI

T20: new errs IOX5, MTOX1-20, TTYX01
-------------------------
MTU%:   600     ! -/P1X Performs various functions for MT: devices
V4 addition
-------------------------
MUTIL:  512     ! -/p2  Performs IPCF control functions
V4 mods
-------------------------
NIN:    225     * 2     Inputs an integer number

T20: If terminating char was a CR followed by a LF, the LF is also
        input.  Potential screw.
10X: not applicable since 10X EOL is single char (sigh)
-------------------------
NODE:   567     ! -/p1X Performs DECnet network functions
V4 addition
-------------------------
NOUT:   224     = 2     Outputs an integer number
```

```
RH(AC3) NO%RDX - (10X: 2-16) (T20: 2-36)

T20: new err IOX11
--------------------------
ODCNV:  222      * 1X    Outputs date and time conversion

T20: extra flag (AC4: B3 IC%JUD) - apply Julian day format on input
       and mark output result in AC4 with same bit.
--------------------------
ODTIM:  220      * 1X    Outputs date and time

10X: Default of -1 gets 336021,,0        ; add time-zone at end
T20: Default of -1 gets 336001,,0        ; don't add time-zone at end
--------------------------
ODTNC:  230      * 1X    Outputs date/time without converting

T20: extra flag (AC4: B3 IC%JUD) to apply Julian day format on output
--------------------------
OPEN:   742      ?  [TCP] Open connection  (Internet/TCP)

[TCP:
Accepts AC1:         Flags,,Pointer-to-Connection-Block
        AC2:         Persistence in seconds
        AC3:         Retransmission parameters
Returns +1:       failure, code in AC1
        +2:       OK, useable handle (a JCN, Job Connection Number) in 1

      Flags:

         ForceSync:  On to force synchronization without any data having
                     been sent.

         Wait:       Don't return until connection is opened.

         Persistent: Keep trying by sending SYN packets periodically.
]
--------------------------
OPENF:   21      * 2     Opens a file

AC2 bit changes:
  B6-B9 OF%MOD  (10X: only values of 0 or 17 valid)
                (T20: in addition to values of 0 and 17
             10 indicates image mode
             13 indicates image binary mode
             14 indicates binary mode)
  B10-17 (10X: device dependent) (T20: - )
  B18 OF%HER    (10X: - ) (T20: halt on I/O device or data error)
  B21 OF%EX     (10X: allow execute access) (T20: reserved )
  B23   (T20: - ) (10X: access as specified by page table of file)
  B24   (T20: - ) (10X: protected entry only (NOT IMPLEMENTED YET))
  B29 OF%RTD    (10X: - ) (T20: enforce restricted access)
  B30 OF%PLN    (10X: - ) (T20: disable line number checking)
  B31 OF%DUD    (10X: - ) (T20: suppress the system updating of modified
                               pages in memory to thawed files on disk)
  B32 OF%OFL    (10X: - ) (T20: open the device even if it's off line)
  B33 OF%FDT    (10X: - ) (T20: force update of .FBREF, .FBCNT)
  B34 OF%RAR    (10X: - ) (T20: wait if the file is offline)

[76: 10X:  if an existing file is opened for output only  (write
       but  not  read),  the EOF will get set to 0 and any existing
       data will be overwritten.  (very painful!) ]

10X: network errs OPNX20,21,22
T20: new err OPNX17,18,23,25,26 DESX7, TTYX01

[ISI:
```

Under T20:

1) Opening TTY for byte size 8, automatically sets binary mode
[KLH: does this mean it works by changing the mode word and
doing nothing else (yuck!) or just happens to do it as a
side effect?? Still yuck.]

2) OPNX25 = "device is write-locked"
(10X: OPNX25 = "directory access does not permit opening files")

]

OPRFN: 326 - P2/- [76] Perform 10X operator functions

[MT: added to MIT T20]
[76: (not in '73 doc)

OPRFN (OPeRator FuNctions) allows programs in user code to perform many of the functions which have been done via MDDT.

Accepts AC1: Sixbit function name (see SWPMON for
up-to-date list)
AC2-4: function dependent arguments
Returns +1: Unsuccessful, error code in 1
+2: Successful, function completed

Each function requires a specific capability and calls a short routine in SWPMON to do its task.

NOTE TO FIELD SITES: BBN will never define a sixbit name starting with the letter "X" in this table. This allows field sites to add subfunctions without conflicting with future BBN additions.

Functions implemented in OPRFN:

SIXBIT NAME	CAPABILITIES REQUIRED	ARGUMENTS IN AC2, AC3
KFACT	W or OP	Fltg Pt number from 0 thru 1.0
NETON	W or OP or M	zero or -1
JTRPON	W or OP	zero or -1
MKPGSA	W or OP or M	first page # in 2, last page # in 3
MKPGSU	W or OP or M	first page # in 2, last page # in 3
DDTRCL	W or OP	none
DDTFSH	W or OP	none
DBUGSW	W or OP	Zero or 1 or 2
DCHKSW	W or OP	zero or 1

The functions KFACT, NETON, JTRPON, DBUGSW and DCHKSW simply check and then store their argument in the named monitor control cell.

DDTRCL and DDTFSH call those named routines.

MKPGSA and MKPGSU call those routines for the pages supplied.

Errors:
OPRFX1 function requested is not defined
OPRFX2 illegal parameters to an OPRFN function

]

PBIN: 73 = 1XDE Inputs the next byte

10X: old err DESX3 (not on T20)

PBOUT: 74 = 1XDQ Outputs the next byte

T20: new err IOX11

PEEK: 311 * PM2 Obtains monitor data

```
T20: Information from multiple pages may be acquired in a single PEEK
10X: Each PEEK is restricted to accessing ONE page per call
        (specifically, data transfer cannot cross a monitor page boundary)

10X: err PEEKX1 "xfer cannot cross mon page boundary"
--------------------------
PLOCK:  561     ! -/PM1X        Locks physical pages
V4 addition
--------------------------
PMAP:    56     * 1X    Maps pages
V4 mods

AC3 bit changes:
  B0 PM%CNT (T20: RH(AC3) contains number of pages to be operated on)
              (10X: - )
  B5 PM%PLD (T20: preload the page, dont wait til ref'd) (10X: - )
  B8   -      (T20: - ) (10X: trap-to-user on any access)
  B11 PM%ABT (T20: Unmap a page and throw changed contents away)
              (10X: - )

Additional T20 changes:
1) Fork-to-file mapping (Case II) releases the page from the fork map.
        10X retains the fork page.
2) File-to-fork mapping (Case I) is illegal if the file was not openned
   with read access allowed.
3) A file with pages still mapped into a fork cannot be closed.
        [76: 10X: CLOSF will succeed, but won't actually close file.]
4) T20.4 allows Case V: unmapping (AC1 -1) pages from a file.

10X: For file-to-fork mapping, if file was opened with "access as spec'd
        in page table" then the access granted is the AND of that requested
        and that in the page table of the file.
10X: A mem ref to a page hacking trap-to-user set will:
        1) Clear the trap-to-user bit and continue the ref.
        2) Request a PSI on channel 21.
        Trap-to-user is noticed by the monitor before other access
        restrictions.  Illegal read & write are next, then copy-on-write.

10X: err DESX2, DESX4
T20: new errs PMAPX3, PMAPX4, PMAPX5, PMAPX6, PMAPX7, FRKHX7,
        LNGFX1, IOX11, ARGX06
--------------------------
PMCTL:  560     ! -/P1X Controls physical memory

--------------------------
PPNST:  557     ! -/1X  Translates project-programmer number to string

--------------------------
PRARG:  545     ! -/1X  Reads/sets process argument block

--------------------------
PSIN:    75     - ?     Inputs a string

NOTE:  this call is not shown anywhere in the 10X doc, nor does it
appear in the T20 doc.  However it appears to exist (or did exist),
being to SIN as PSOUT is to SOUT.  Perhaps more info will turn up.
[MT: must be west coast hack, never in BBN 10X]
--------------------------
PSOUT:   76     = 1XDQ  Outputs a string

T20: new err IOX11
--------------------------
PSTI:   362     - 2/-   [A] PTY simulate TTY input
[A:
        Pseudo-tty Simulate tty input function; insert character into
        pseudo-tty input buffer.

Accepts AC1: PTY designator
        AC2:    character
Returns +1: Unsuccessful, error if error number in 1
                else input buffer full
        +2: Successful, character inserted
Errors:
        illegal PTY line designator
]
--------------------------
PSTIN:  611     - 1/-   String input from TTY, with editing

EA:     Apparently an IMSSS call?

        String input from the teletype, with editing provided.

        SHORT FORM CALL (simple line input):

Accepts AC1:    Tenex string descriptor
                        (locates beginning of buffer)
        AC2:    Maximum number of bytes to input.
        AC3:    Bits 0-8:     Zero for short call.
                Bits 18-26:  Optional additional character
                                to terminate string.
                Bits 27-35:  Optional additional character
                                to terminate string.
Returns +1: Always, with an updated string pointer in 1,
                remaining buffer room in 2.

        1 is left in the standard Tenex convention:

        LDB AC, 1  will fetch the terminating character;
        DPB AC, 1  will clobber the terminator (as with a null);
        IDPB AC, 1 will continue the string.
```

PSTIN accepts a line of input from the teletype. The line is terminated by RETURN, ALT-MODE, any control-character other than TAB or LINEFEED, or either of the special terminators given in the right half of AC3.

Note: A RETURN is echoed as both RETURN and LINEFEED, but only RETURN is placed in the buffer.

```
Various editing funtions are provided by the system during typein:
   ^X           Prints  ##  and goes to next line;
                 erases everything (start over).
  RUBOUT        Erases one character.  If nothing is left,
                 same as ^X.
   ^A           Same as RUBOUT.
   ^W           Erases one word, where a word is delimited
                 by space and/or TAB and/or LINEFEED.  Erases
                 spaces or tabs back to last word if necessary.
                 If nothing left, same as ^X.
   ^R           Retype.  Prints  ##, goes to next line, and
                 prints out everything it has so far.
  LINEFEED      Line continuation.  Echoes as carriage-return
                 line-feed, but does not terminate the input.
                 May be erased with RUBOUT or ^W.  It is
                 suggested that programs interpret LINEFEED
                 as synonymous with space.
```

Disposition of erasure:

On an imlac (or TEC), the action of RUBOUT and ^W will be to actually erase the appropriate characters from the screen. (This will not work correctly if a tab is erased).

On a teletype, RUBOUT echoes the erased characters inside of square brackets. Thus, for the first erasure, RUBOUT prints [and then the character being erased. Subsequent RUBOUT's only echo the character. When a new input character is typed, a] is printed prior to echoing the character.

On a teletype, ^W causes the system to print __ (two underlines and a space). Another ^W does the same thing. The space following the underlines is to emphasize that the spaces preceeding the erased word are still present in the buffer.

On any device, erasing a LINEFEED will be indicated by the printing of ^^ .

Several other features are supported by PSTIN, in the hopes of providing the features needed for various implementations.

LONG FORM CALL:

```
Accepts AC1:    Tenex string descriptor
                          (locates beginning of buffer)
        AC2:    Maximum number of bytes to input
        AC3:    Bit 0 = 1:  Indicates a special character
                                 table has been provided.
                Bit 1 = 1:  Indicates that a erasure message
                                 has been provided.
                Bit 2 = 1:  Indicates that characters are
                                 already in the buffer and
                                 should be skipped on startup.
                Bits 18-26: Optional additional character
                                 to terminate input.
                Bits 27-35: Optional additional character
                                 to terminate input.

        AC4:    LH        Number of characters to skip on
                           startup (if B2 of 3 is set).
                RH        Address of special character table
                           (if B0 of 3 is set).
        AC5:    Tenex string descriptor for string to print
                          following ^X or ^R  (basically for
                          retype of prompt character).

Returns +1: Always, with updated string pointer in 1,
                remaining buffer room in 2.
```

Special features available in long call:

SKIPPING CHARACTERS ON STARTUP:

The string descriptor in 1 locates the beginning of the buffer. However, the situation may arise in which the user types part of a line and causes termination of the PSTIN without really completing what he has to type; the program can then re-enter the PSTIN with the same buffer (now containing the partial input) and instruct PSTIN to skip the characters already seen. Input processing continues as if the first portion of the bufferhad just been typed; that part may be erased, retyped, etc.

STRING POINTER IN AC5:

This string will be printed at the beginning of the new line following the ^X editing action. Similarly, when ^R is used, this string will be printed at the front of the line before the input is retyped.

SPECIAL CHARACTER TABLE:

This provides a facility to alter the editing character assignments, and also provides afacility for arbitrary character translation during the PSTIN. The table has 200(8) entries, one per character. Each entry is 9 bits long, so they are packed 4 to a word.

Byte format:

```
  400 bit on:   Special editing action.  Rest of byte is
                coded with the function, as follows:
                  0    ignore              [null]
                  1    erase character     [RUBOUT]
                  2    erase word          [ ^W ]
                  3    erase all           [ ^X ]
                  4    retype              [ ^R ]
                  5    continue line       [LINEFEED]
                         (echo CRLF; character passed to program)
```

The characters for functions 0 thru 4 are not put into the buffer.

400 bit off: The low-order 7 bits specify the character to be echoed and placed in the buffer. This may or may not correspond with the character typed. A suggested use would be to force upper case translation, though any mapping is possible.

200 bit on: Character is a terminator. It is translated as just described, and placed in the buffer, and then the PSTIN is terminated.

If the user does not supply a special character table, the system will use a default table giving the editing functions and terminators described above.

]

PSTO: 363   - 2/-   [A] Simulate PTY output

[A:

Pseudo-tty Simulate Tty Output function; retrieves character from pseudo-tty output buffer.

```
Accepts AC1:    PTY line designator
        AC2:    0 => on empty output buffer, take error return
                -1 => on empty output buffer, block until
                        not empty and then return char,
                      or
                        PTY process blocks for input and then
                        take error return
Returns +1: Unsuccessful, error if error number in 1,
                else output buffer empty
        +2: Successful, char in 2
Errors:
        Illegal PTY designator
```

This provides for moving characters from the output buffer of the designated pseudo teletype. If the output buffer is empty, the action depends upon the blocking option. If AC2 is greater than or equal to 0, then a return to +1 is made immediately. If AC2 is less than 0, then the process will be blocked until a character appears in the output buffer, in which case the return is to +2; or until the pseudo teletype process blocks for input, when the return will be

```
to +1 with AC2 set to -1.
]
--------------------------
PUPI:   441      ?  Unknown [PARC: Xerox PUP protocol hack]
--------------------------
PUPNM:  443      ?  Unknown [PARC: Xerox PUP protocol hack]
--------------------------
PUPO:   442      ?  Unknown [PARC: Xerox PUP protocol hack]
--------------------------
PUPSK:  444      ?  Unknown [PARC: Xerox PUP protocol hack]
--------------------------
RCDIR:  553      ! -/1X  Translates string to directory number

--------------------------
RCM:    134      == 1X   Reads the channel word mask

--------------------------
RCUSR:  554      ! -/1X  Translates string to user number

--------------------------
RCVIM:  751      * 2     Retrieves message from ARPANET special message queue
V4 mods

AC1 flag bits:
        B0 10X: ?
           T20: If set, user will receive 96-bit leader.  0 = 32-bit
        B1 10X: ?
           T20: If set, user will receive data in high-order 32 bits of
                        each word of message.  If 0, data is in all
                        36 bits of each word.

--------------------------
RCVIN:  755      ?  Unknown (Internet version of RCVIM?)

--------------------------
RCVOK%: 575      ! P1X   Retrieves access request from GETOK queue
V4 addition
--------------------------
RDDIR:   32      -? 2/-  Read Device Directory

Accepts AC1: dev des (only DECtapes currently allowed)
        AC2: address of area in which the directory is to be returned
Returns +1: failure, err # in AC1
        +2: success, directory has been copied to specified area in
                caller's address space
Errors: RDDIX1  Cannot read directory for this device (not DECtape,
                        not mounted, etc.)
--------------------------
RDSDP:  267      - 2/-   Read Status of Display Process

Apparently a BBN hack, see 10X doc p. 4-42
--------------------------
RDTTY:  523      ! -/2   Reads data from primary input designator

--------------------------
RDTXT:  505      ! -/?   Read Text (obsoleted by RDTTY, TEXTI)
--------------------------
RECV:   741      ?  [TCP] Receive data (Internet/TCP)

[TCP:
Accepts AC1:          Flags,,JCN or Pointer-to-Connection-Block
        AC2:          0,,Pointer-to-Data-Ring
Returns +1:         failure, error code in 1
        +2:        OK, JCN in 1

      Flags:
         JCNSupplied:   (see TCP-JSYS-DOC)
         Wait:          (ditto)
]
--------------------------
RELD:    71     = 2     Releases a device

T20: new err DEVX6 "job has open JFN on device"
--------------------------
RELDC:  263     - 1/-   Release Display Console

Apparently a BBN hack, see 10X doc p. 4-37
--------------------------
RELDP:  261     - 1/-   Release Display Process

Apparently a BBN hack, see 10X doc p. 4-36
--------------------------
RELIQ:  757     ?  Unknown (Internet version of RELSQ?)

--------------------------
RELSQ:  753     == 1    Deassigns ARPANET special message queue

--------------------------
REPTY:  361     - 2/-   [A] Release a PTY

[A:
        RElease Pseudo-TtY (PTY) line.

Accepts AC1: PTY designator to release, -1 to release all
                PTY's owned by this job.
Returns +1: Unsuccessful, error in 1
        +2: Successful, PTY (or all of job's PTYs) released
Errors:
                illegal PTY designator

        This releases a pseudo teletype from the job which assigned
it. The job may not perform further operations on that pseudo
terminal. If AC2 is equal to -1, all pseudo teletypes held by this
job are released.
]
--------------------------
RESET:  147     * 1     Resets/initializes the current process

T20: does not affect the setting of CCOC words, nor tab stops
10X: Sets the CCOC words to: send CR, LF, ^G; simulate EOL, VTAB;
     simulate or send FF, TAB (per B1, B2 of mode word); indicate
     all others with ^X
     Sets tab stops every 8 columns
--------------------------
RFACS:  161     = 1X    Reads process' ACs

T20: new err FRKHX8
--------------------------
RFBSZ:   45     * 1X/2  Reads files's byte size

Return sequence change:
  T20:  Ret +1 on error; Ret +2 if successful
  10X:  Ill inst PSI on error; Ret +1 if successful

Err codes are same, though.
;Suggested coding:
;       RFBSZ (or SFBSZ)        ;Do JSYS
;        JUMP 16,error  (JUMP 17,error)
--------------------------
RFCOC:  112     = 1X    Reads file's control character output

10X: err DESX6
T20:
```

```
T20: new err TTYX01
-------------------------
RFMOD:  107     = 1X    Reads a file's mode

Accepts AC1: (10X: file des) (T20: src des)

10X: errs DESX1,3,5,6,DEVX2 (not in T20??)
T20: only err is TTYX01 (??)  Doc doesn't say, but assume this generates
        ILINT just as 10X errors do.
-------------------------
RFORK:  155     = 1X    Resumes a process

T20: doc notes that RFORK is a no-op for processes which were
        frozen indirectly.  True for 10X too?

T20: new err FRKHX3
-------------------------
RFPOS:  111     = 1X    Reads terminal's position

Accepts AC1: (10X: file des) (T20: dev des)
10X: doc says line number (LH of returned AC2) is set to zero only
        when ^L typed and upon LOGIN.  True??

10X: err DESX6
T20: new err TTYX01
-------------------------
RFPTR:   43     == 2    Reads file's pointer position

-------------------------
RFRKH:  165     * 1X/2  Releases a process handle

T20: allows use of -1 to specify "all relative handles" (like RESET)

Return sequence change:
  T20:  Ret +1 on error; Ret +2 if successful
  10X:  Ill inst PSI on error; Ret +1 if successul

T20: new errs FRKHX2,3
;Suggested coding:
;       RFRKH
;        JUMP 16,error  (JUMP 17,error)

-------------------------
RFSTS:  156     * 1X    Reads a process' status
V4 mods

T20: B0 RF%LNG in AC1 invokes "long form" call, which 10X doesn't have.
T20: new err FRKHX2 (cannot get status of superior)

T20: new values returned in AC1,B1-B17:
  5 .RFSLP - the process is dismissed for a specified amount of time.
  6 .RFTRP - the process is dismissed because it attempted to
       execute a call on which an intercept has been set by its superior.
  7 .RFABK - the process is dismissed because it encountered an
       instruction on which an address break was set.
-------------------------
RFTAD:  533     ! -/1X  Reads file's time and dates
V4 mods
-------------------------
RIN:     54     == 1XDE Performs random input

-------------------------
RIR:    144     == 1X   Reads software interrupt table addresses

-------------------------
RIRCM:  143     == 1X   Reads inferior reserved channel mask
```

```
--------------------------
RLJFN:   23      = 2     Releases JFNs

T20: -1 releases JFNs of process and inferiors
10X:  ditto [76: in this job, not just proc & infs!]
        [76: will not release a JFN if file is still mapped]

T20: new err OPNX1 [76: not new, 10X also has]
--------------------------
RLSIG:  731      - 2/-   [BBN76] Release signal ID

[76: experimental BBN call, p. 11-8]
--------------------------
RMAP:    61      * 1X    Obtains a handle on a page

AC2 bit changes:
  B8  - (T20: - ) (10X: trap-to-user on any access)

10X: If argument to RMAP specifies a page in a fork (not a file)
and that page is private, then the page will be moved to
the job's PMF and will then be shred between the fork and
the file.  The identified returned in this case will contain
the JFN of the job's PMF and the page number in that file that
was assigned.  (Use RPACS to just find out if a page exists,
without changing its state.)

;Don't use AC2(B8)
;Under T20, there is no notion of a Private-Memory-File to provide
; automatic linkage of core memory to a disk file under RMAP; find
; alternative if so used.
--------------------------
RNAMF:   35      =? 2    Renames a file

I suspect funnyness here.  The 10X doc sounds just like the T20
doc in saying that JFN-1 (old) is released whereas JFN-2 (new) isn't.
HOWEVER, in actual practice it seems that the 20X doc is lying,
and JFN-1 is NOT in fact released!

10X: has err DESX2 (not in T20)
T20: new errs DESX7, RNAMX13
--------------------------
ROUT:    55      = 1XDQ  Performs random output

T20: new err IOX11
--------------------------
RPACS:   57      * 1X    Reads a page's accessibility

AC2 bit changes (on return):
  B8  - (T20: - ) (10X: trap-to-user)
  B26 - (T20: - ) (10X: trap-to-user in first pointer)

10X: errs DESX2, DESX7
T20: new errs DESX8, FRKHX2

;Under TOPS-20, trap-to-user feature unavailable
--------------------------
RPCAP:  150      = 1X    Reads process capabilities

T20: new err FRKHX3
--------------------------
RSCAN:  500      ! -/2   Accepts a new string or uses the last string as input

--------------------------
RTCHR:  640      ?  Unknown [MT: VTS]
```

```
RTFRK:  322      * 2/1X  Returns the handle of JSYS-trapped fork
```

10X: this call is not in '73 doc. Following is from '76 doc.

RTFRK's actions are the same, but return sequence isn't.

```
10X: returns     +1: failure, error # in AC1
                 +2: success, returns
                 AC1: 0 if no fork trapped, else
                  <relative fork handle>,,<JSYS #>
                                  where 0 < JSYS# < 512.
T20: returns +1: success, with
                 AC1: relative process handle, or 0 if no fork trapped
                 AC2: instruction causing trap, or 0 if "
T20: error causes ILINT.
```

10X and T20 both have only the one error FRKHX6.

```
RTIW:   173      == 1X   Reads terminal interrupt word
```

```
RTMOD:  636      ?  Unknown [MT: VTS]
```

```
RUNTM:   15      = 1X    Returns runtime of process or job
```

T20: AC2 (divisor) always 1000. Is this true for 10X too?

```
RWM:    135      * 1X    Reads waiting channel interrupt word mask

AC2:
        10X: Bn means level n interrupt in progress
        T20: Bn means level n interrupt in progress out of USER code
          B18+n   "      "  "      "      "     "     "   " MONITOR "
```

[ISI claims 10X RWM doesn't touch AC2?]

```
RWSET:  176      == 1    Releases the working set
```

```
SACTF:   62      = 2     Sets account designator of file
```

```
10X: Account must be numeric if LOGIN dir only takes numeric accounts,
        regardless of directory JFN is in, unless process has WHL/OPR cap.
10X: err DESX2
T20: new errs VACCX0, VACCX1, VACCX2
```

```
SAVE:   202      = 1X    Saves a file as nonsharable
```

```
10X: doc says SAVE notes whether proc is 10X or 10/50 format by its
        entry vector word (LH = JRST for 10/50) and appropriately formats
        last word(s) of SAVE file.  Does T20 do this too?  Probably.
```

T20: new errs SAVX1, IOX11, FRKHX8

```
SCSLV:  744      ?  Unknown (Internet/TCP?)
```

```
SCTTY:  324      * 2/1X  Changes controlling terminal
```

10X: this call not in '73 doc, stuff here is from '76 doc.

10X: A terminal assigned to a job can be designated as the source of
terminal PSI's for a fork and its inferiors. Each process
in a job has one source of terminal PSI's but different processes may

```
have different sources.  ATI, DTI and STIW effect the fork controlling
terminal (with the exception that -5 passed to STIW in AC1 effects the
job controlling TTY).

Accepts AC1:    T20: function code,,process handle
                10X: function bits,,process handle
        AC2:    T20 == 10X

T20: returns +1 always, errors cause ILINT
10X: returns    +1: failure, error # in AC1
                +2: success

10X: all T20 functions are also in 10X, i.e. .SCRET, .SCSET, and .SCRST.
        Function bits are defined just as for the TFORK call:
                10X function bit = 1B<T20 function code>

10X err CAPX2 replaced by T20 err SCTX4.
-------------------------
SCVEC:  301     = 1X    Sets entry vector of compatibility package

T20: if AC2 is -1, UUO simulation is disabled.
10X: may not have this feature (doc doesn't mention it)
T20: new err FRKHX8
-------------------------
SDSTS:  146     * 1X    Sets device's status

AC2: (10X: new status bits) (T20: mask indicating bits to be changed)
AC3: (10X: device dependent) (T20: - )

Not sure if the different AC2 wording means anything; probably the
        call works in the same way.
T20: new err DESX9
-------------------------
SDVEC:  543     ! -/1X  Sets entry vector of RMS

-------------------------
SEND:   740     ?  [TCP] Send data (Internet/TCP)

[TCP:
Accepts AC1:        Flags,,JCN or Pointer-to-Connection-Block
        AC2:        0,,Pointer-to-Data-Ring
        AC3:        TimeOut in Seconds (0 for infinite)
        AC4:        Retransmission parameters
Returns +1:       failure, error code in 1
        +2:      OK, JCN in 1

     Flags:
        JCNSupplied:    (see TCP-JSYS-DOC)
        Wait:           (ditto)
]
-------------------------
SETER:  336     * 1X    Sets the last error in a process

10X: this call does not exist in '73 doc.
AC1 diffs:
        B0 (T20: - )(10X: [76: if on, use ACs 4-10 to set PSB err params.])

T20: returns +1 unless error (ILINT)
10X: [76: returns +1 if error, +2 if success]

T20: new errs FRKHX3, FRKHX8
-------------------------
SETJB:  541     ! -/1X  Sets job parameters
V4 mods
[ISI:  (not sure why they mention this; possibly a 101B vs V3 diff? --KLH)
Extra function .SJSRM - set remark for current job session.
```

AC3 contains a pointer to the session remark, which is updated on a successful return.
]

SETNM: 210 = 1 Sets program name

AC1 can contain:

```
        T20: only a SIXBIT program name
        10X: a SIXBIT program name,
             or [76: a JFN - the first 6 chars of the file name are used]
             or [76: 1,,0 for "insist SETNM" in which case
                        AC2: JFN or SIXBIT program name]
```

10X: [76:

The name specified by AC1 is put in the table JOBNM2. If the name specified by AC1 is already associated with a statistical slot then this fork is associated with that statistics slot. If the name specified by AC1 is not already associated with a statistics slot then this fork is associated with the .OTHER statistic slot, unless the "insist SETNM" form of the call was used. The "insist" form will create a new statistics slot for the name if none exists, unless there is no room for any more slots, in which case .OTHER is used after all.
]

T20: SETNM only sets the "program name", and no stats are kept for that. The equivalent to 10X SETNM's name is SETSN's "subsystem name"; T20 keeps stats for the subsystem name in exactly the same way that 10X keeps stats for the SETNM name (in SNAMES, STIMES, and SPFLTS).
The T20 and 10X EXECs set names similarly.

SETNT: 603 - 1/- [A] Sets network on or off
[A:
Accepts AC1: 0 turns network off, -1 turns network on
Returns +1: always
]

SETPV: 773 ? Unknown

SETSN: 506 ! -/2 Sets system name for a process

SEVEC: 204 = 1X Sets entry vector

T20: new err FRKHX8

SFACS: 160 = 1X Sets process' ACs

T20: new err FRKHX8

SFBSZ: 46 * 1X/2 Sets file's byte size

Return sequence change:
T20: Ret +1 on error; Ret +2 if successful
10X: Ill inst PSI on error; Ret +1 if successful

T20: new err DESX8

```
;Suggested coding:
;       SFBSZ            ;Do JSYS
;        JUMP 16,error   (JUMP 17,error)
```

SFCOC: 113 = 1 Sets file's control character output

10X: err DESX6
T20: new err TTYX01

```
-------------------------
SFMOD:  110      * 1X    Sets a file's mode

10X: err DESX6
T20: new err TTYX01 (doc doesn't say, but assume errors generate
        ILINT as for 10X)

10X: affects
        B18-23, wakeup control
        B24-25, echo mode
        B28-29, terminal data mode
        B31     convert lower case to upper case on input
T20: affects
        B0 TT%OSP         output suppression control
        B18-23 TT%WAK     wakeup control
        B24 TT%ECO        echoes on
        B28-29 TT%DAM     data mode

[ISI:
AC2 bit changes:
  B0   TT%OSP     (T20: output suppression control (SFMOD)) (10X: - )
  B25 - echo mode (T20: (SFMOD)) (10X: (STPAR))
  B34 - (T20: output page mode (STPAR))
        (10X: repeat last character (read only bit))

;Suggested coding:
;       MOVE 2,[bits]
;       SFMOD                     ;Do both SFMOD and STPAR if playing
;       STPAR                     ; with AC2(B25)

]

-------------------------
SFORK:  157      * 1X    Starts a process

AC1 new flag:
  B0 SF%CON      (10X: - ) (T20: continue process, ignore AC2)

10X: err FRKHX4 "fork already running" (not in T20)
T20: new err FRKHX5 "process has not been started" (SF%CON error)
        plus FRKHX8
-------------------------
SFPOS:  526      ! -/1   Sets terminal's position

-------------------------
SFPTR:   27      = 2     Sets file's pointer position

T20 doc doesn't make it clear (as does 10X doc) that when ptr is
set past EOF, a read will cause EOF lossage but a write will store
the data and update the file length.

T20: New errs DESX2, DESX8
-------------------------
SFRKV:  201      * 1X    Starts process using its entry vector
V4 mod

If process has 10/50 entry vector (JRST in LH) then AC2 RH = 0 means
use .JBSA=120 contents, = 1 means use .JBREN=124 contents, but:
        T20: also add AC2 LH to contents, as offset.  Only 0 and 1
                legal, and only for RH = 0.
        10X: -

T20: new err FRKHX8
-------------------------
SFTAD:  534      ! -/p1X Sets file's time and dates
V4 mods
```

```
SFUST:  551      ! -/p1X Sets author and last writer name strings

--------------------------
SIBE:   102      * 2X/2  Skips if input buffer is empty

Accepts AC1: (10X: file des) (T20: src des)
Returns +1: Input buffer not empty, number of bytes in input buffer
                returned in AC2
        +2: 10X: buffer empty, no ACs changed
            T20: if AC2 = 0, buffer empty
                 else AC2 = error code

10X: Generates ILINT for errors (add'l error DESX6)
T20: Returns +2 for errors, code in AC2 (add'l error TTYX01)
--------------------------
SIBF:   364      - 2/-   [A] Skip if input buffer full
[A:
Accepts AC1: TTY designator
Returns +1: Input buffer not full, number of chars in buffer in AC2.
        +2: Input buffer full.
Errors:
                Illegal TTY designator
]
--------------------------
SIGNL:  733      - 2/-   [BBN76] Generate signal

[76: experimental BBN call, p. 11-8]
--------------------------
SIN:     52      = 1XDE/1XDE     Performs string input

[KLH: I think T20 halts quietly on EOF with truncated count,
whereas 10X blows itself out of the water with an EOF int.  Not clear
under what conditions (if any) a T20 SIN will cause an EOF interrupt.]

T20: new errs IOX7, IOX8
--------------------------
SINM:   571      ?  Reads data from block-mode terminals
Not in V4 manual??
--------------------------
SINR:   531      ! 1XDE  Performs record input

--------------------------
SIR:    125      = 1X    Sets software interrupt table addresses

T20: new err FRKHX8
--------------------------
SIRCM:  142      * 1X    Sets inferior reserved channel mask

New arg
  AC3: (10X: - ) (T20: deferred terminal interrupt word)

Note that on T20, ERJMP/ERCAL will prevent the superior from
        seeing an interrupt that it would otherwise see on 10X.
T20: new err FRKHX8
--------------------------
SIZEF:   36      == 2    Gets the size of a file

--------------------------
SJPRI:  245      * 2/1X  Sets job's priority
V4 mods

The 10X doc seems to be messed up about the return sequence; it
lists two possibilities but both are marked "+1:"!!!  Assuming
the second is supposed to be "+2:", we have:
```

```
Return sequence change:
  T20: Ill inst PSI on error; Ret +1 if successful
  10X: Ret +1 on error; Ret +2 if successful

T20: new err SJPRX1 "job is not logged in"

[ISI:
;Suggested coding:
;       SJPRI
;        JRST [ SKIPE TOPS20     ;Are we TENEX?
;               JRST .+1         ;No, ok
;               handle.error]
]

--------------------------
SKED%:  577      ! p1X   Performs services relating to the class scheduler
V4 addition
--------------------------
SKPIR:  127      == 2X   Tests the state of the software interrupt system

--------------------------
SKUSR:  606      - 1/-   [A] Set job % of CPU for a user
[A:

Accepts AC1:      -1 self
                  >0 Job number (must have wheel or oper capability)
        AC2:      % CPU in range .01 to 100. If greater than 0,
                          must have wheel or oper capability.
Returns +1:       always
]
--------------------------
SMAP:   767      ?  Unknown

[MT: mungs KL-extended addressing]
--------------------------
SMON:     6      * P1X   Sets monitor flags
V4 mods
Content changes to calling ACs:
  AC1/  (T20: function code) (10X: bit mask)
  AC2/  (T20: function value) (10X: bit value mask)

10X doc doesn't document any of the bits except B0 .SFFAC - fact file enabled.

        T20: New err SMONX2 (invalid function)
                Requires WHL/OPR cap
        10X: Requires LOG cap
;TOPS20 test required
--------------------------
SNDIM:  750      * 2     Sends a message to ARPANET special message queue
V4 mods
AC1 new flags:
  B0 -  If set, msg contains 96-bit leader, else 32-bit.
  B1 -  If set, data is in hig-order 32 bits of each word of msg,
        else data is in all 36 bits of each word.
--------------------------
SNDIN:  754      ?  Unknown (Internet version of SNDIM?)

--------------------------
SNOOP:  516      ! -/2   Performs system analysis

[ISI:  (why this entry?  101B vs V3 diff? --KLH)
SNOOP   516     performs system performance analysis

Extra function added:

        code    symbol          meaning
```

```
          7       .SNPAD  Obtain a monitor symbol.

                          AC2: 36-bit value of symbol that is to be
                          looked up in the monitor's symbol table.

                          AC3: radix-50 program name if a local value is
                          desired. If AC3 is 0, the entire symbol table is
                          searched.

                          On return, AC2 contains the first radix-50 monitor
                          symbol that is closest to and has a value less
                          than the specified value, and AC3 contains the
                          difference between the value of the symbol returned
                          and the specified value.
]

--------------------------
SOBE:   103      * 2X/2   Skips if output buffer is empty

Accepts AC1: (10X: file des) (T20: dest des)
Returns +1: output buffer not empty, # bytes remaining returned in AC2
        +2: 10X: output buffer empty, no ACs changed
            T20: AC2: 0 = buffer empty
                 else AC2: error number

10X: Generates ILINT for errors (add'l error DESX6)
T20: Returns +2 for errors, code in AC2 (add'l error TTYX01)
--------------------------
SOBF:   175      * 2X/2   Skips if output buffer is full

Accepts AC1: file designator
Returns +1: output buffer not full, AC2 has # bytes remaining in buffer
            T20: if AC2: 0, error occurred.
        +2: output buffer full, AC2 has # bytes remaining in buffer

10X: Generates ILINT on errors
T20: Returns +1 on errors, with AC2: 0.  New err TTYX01
--------------------------
SOUT:    53      = 1XDQ   Performs string output

T20: new errs IOX7, IOX8, IOX11
--------------------------
SOUTM:  572      ?  Writes data to block mode terminals
Not in V4 manual???
--------------------------
SOUTR:  532      ! 1XDQ   Performs record output

--------------------------
SPACS:   60      * 1X     Sets a page's accessibility

AC2 bit change:
  B8  - (T20: - ) (10X: trap-to-user (PSI channel 21) on any access)

T20: new err DESX8, FRKHX8
--------------------------
SPJFN:  207      = 1X     Sets the primary JFNs

T20: new err DESX3
--------------------------
SPLFK:  314      = 2      Splices a process structure

10X: err symbols SPLFKn correspond to T20 symbols SPLFXn.
--------------------------
SPOOL:  517      ! -/P2   Defines and initializes input spooling
```

SPRIW: 243   =? -/P1X   Sets the priority word

The 10X doc only says this is "under development" but the T20 doc seems to describe a call so simple that it can't be much different in 10X.

SSAVE: 203   = 1X   Saves a file as sharable

10X: SSAVE does not close/release its JFN (unlike SAVE).
T20: SSAVE DOES close/release JFN!!

T20: new errs SSAVX3,4 IOX11

STABS: 106   -? 1X/- Set tabs for file

[MT: not in FOONEX w/VTS]
Acts like NOP if designator associated with non-terminal.

```
Accepts AC1: file des
        AC2, AC3, AC4 in same format as for GTABS (see)
Returns +1: always
Errors: ILINT on DESX1, DESX3, DESX5, DESX6, DEVX2
```

STAD: 226   * P1X   Sets system date and time

Identical except for timeword format. See the TIMEWORD entry.

STAT: 745   ? [TCP] Get status of connection (Internet/TCP)

```
[TCP:
Accepts AC1:          Flags,,JCN or Pointer-to-Connection-Block
        AC2:          -N,,Offset into TCB
        AC3:          -M,,Address in user's space
Returns +1:        failure, error code in 1
        +2:        OK.  Min(M, N) words have been transferred from the TCB
      to the caller's  space.   The  TCB  offset  identifies  where  the
      transfer starts and the Address in user space identifies the start
      of the destination area.

      Flags:

         JCNSupplied    (see TCP-JSYS-DOC)
         Returns statistics:    This flag causes the TCP to dump words
                    from the statistics  area rather  than  a  specific
                    TCB.  Thus,  the JCN is irrelevant.  The Source and
                    Destination  ACs are updated as if a TCB were being
                    dumped.

]
```

STCHR: 641   ? Unknown [MT: VTS]

STCMP: 540   ! -/1   Compares two strings

STDEV: 120   == 2   Translates string to device designator

STDIR: 40   -? 3/- Translates string to directory number

T20: This call replaced by RCUSR and RDUSR.

10X calling sequence -
  AC1: If positive, the entire string is taken literally for an
       exact match. If negative, recognition is attempted on

```
        the string.
        If B17 (1,,0) is 1, use specified device; otherwise use
        default device. [76: only DSK allowed, so bit is useless]
  AC2: pointer to string to be translated
  AC3: device designator if B17 of AC1 is on.
Returns +1: no match
        +2: ambiguous
        +3: unique match, dir number returned in RH of AC1.
        LH flags:
                B0 - name can be used only for dir connection (no logins)
                B1 - user can use alphanumeric acct (off = number only)
                B2 - repeat LOGIN msg on every login (off = only if
                        login msg newer than date/time of last login)
        Appends remainer of string (if any) to original string if recog
        was invoked, and updated string ptr is returned in AC2.
--------------------------
STI:    114     = p1X   Simulates terminal input

T20: new errs DESX2, WHELX1, TTYX01
--------------------------
STIW:   174     = 1X    Sets terminal interrupt word

T20: STIW call requires the process to have SC%CTC capability enabled
to disable the code for CTRL/C interrupts or to give -5 as an argument.
T20: new err FRKHX8
--------------------------
STMOD:  637     ?  Unknown [MT: VTS]

--------------------------
STO:    246     ! -/1X  Simulates terminal output

[MT: in 10X as well]
--------------------------
STPAR:  217     * 1X/1  Sets terminal parameters

10X: no errors documented
T20: new errs DESX1,3,5,DEVX2,TTYX01

10X: can hack all T20 bits except
        B25 TT%ECM (10X SFMOD sets this)
        B31 TT%LIC (10X SFMOD sets this) [76: STPAR can set this too!]
        B34 TT%PGM (10X: nothing hacks this)

;Suggested coding:
;       MOVE 2,[bits]
;       SFMOD                   ;Do both SFMOD and STPAR if playing
;       STPAR                   ; with incompatible bits.
--------------------------
STPDP:  265     - 1/-   Stop Display Process

Apparently a BBN hack, see 10X doc p. 4-40
--------------------------
STPPN:  556     ! -/1X  Translates string to project-programmer number

--------------------------
STRDP:  264     - 2/-   Start Display Process

Apparently a BBN hack, see 10X doc p. 4-37
--------------------------
STSDP:  266     - 2/-   Set Status of Display Process

Apparently a BBN hack, see 10X doc p. 4-41
--------------------------
STSTS:   25     * 2     Sets a file's status
V4 mod
Doc here is confusing with respect to actions of bits in AC2.  All
```

```
agree that only bits 9,13,17 (GS%ERR, GS%HLT, GS%FRK) can be changed.
T20.3 claims that STSTS can SET those bits (presumably to 1)
T20.4 claims that you can only CLEAR those bits (by furnishing 0 values)
        -- furnishing a 1 value won't change the bit.
10X says nothing, but implies that it acts like T20.3 doc, i.e. the 3
        bits are set to the value (0 or 1) furnished by AC2.

--------------------------
STTYP:  302     * 1X    Sets the terminal type number (TTYs only)

T20: Also sets mode word (STPAR) bits B1-17 from internal table
10X: Doc says only sets bits B1-3??

T20: new errs DESX2, STYPX1 "invalid terminal type", TTYX01
--------------------------
SWJFN:   47     = 1X    Swaps two JFNs

T20: new err SWJFX1 "illegal to swap same JFN" (can't those cretins
        simply no-op it???)
--------------------------
SWTCH:  320     -? 1/-  Returns data switch setting

Accepts: none
Returns +1: always, with 36-bit data switch setting in AC1
Errors: none
--------------------------
SWTRP%: 573     ! -/1X  Traps for arithmetic underflow or overflow conditions
V4 addition
--------------------------
SYERR:  527     ! -/P1X Writes data to the system error file

--------------------------
SYSGT:   16     == 1    Returns information for a system table

--------------------------
TBADD:  536     ! -/1X  Adds entry to command table

--------------------------
TBDEL:  535     ! -/1X  Deletes entry from command table

--------------------------
TBLUK:  537     ! -/1X  Looks up entry in command table

[ISI:   (why this entry?  101B vs V3 diff?  --KLH)

Command table changes:

words 1 through n (left half) -
10X:    address of ASCIZ in LH
T20:    address of an argument in LH. This argument contains optional
        bits pertinent to the string followed by the ASCIZ string itself.

The argument has two formats in T20:
  format 1 - When B0-B6 are all off and B7 is on, the string actually
        begins in the next word of the argument and the remainder of
        this word contains data bits relevant to the string.

        currently defined:
                B34 - do not recognize this string

  format 2 - When any bits (B0-B6) are on or if B7 is off, the string
        begins in that word.
]
--------------------------
TEXTI:  524     ! -/2   Reads input from a terminal or a file
V4 mods
```

```
---------------------------
TFORK:  321      * 2/1X  Sets and removes monitor call intercepts

10X: doesn't exist in '73 doc, so 10X info here is from '76.

AC1:    T20: function code,,process handle
        10X: function bits,,process handle
AC2:    T20: PSI chan #,,size in bits of bit table
        10X: PSI chan #,,address of bit table
AC3:    T20: address of bit table
        10X: -

10X: function bits are scanned from left to right (B0 to B6).  Only
        the function corresponding to the most significant bit is
        performed.  Error if no bits or an undefined bit is specified.
        The bits correspond simply to the T20 function codes:
                10X function bit = 1B<T20 function code>

        Codes 0-6 (.TFSET, .TFRAL, .TFRTP, .TFSPS, .TFRPS, .TFTST, .TFRES)
        work exactly the same.

T20: new codes 7 .TFUUO, 10 .TFSJU, 11 .TFRUU to trap TOPS-10 UUOs.
10X: traps of 10/50 UUOs are not supported.

T20: returns +1 always (errors get ILINT)
10X: returns +1 on error (# in AC1), +2 on success

T20: new errs FRKHX8, TFRKX3
---------------------------
THIBR:  770      ! -/2   Blocks the current job

---------------------------
TIME:    14      = 1     Returns time system has been up

T20: AC2 (divisor) always 1000.  Is this true for 10X too?
---------------------------
TIMER:  522      ! -/2   Sets time limit for a job

---------------------------
TLINK:  216      * p2    Controls terminal linking

[Random note: on some T20's, links pass along ALL output exactly
as sent to original terminal, including binary-mode output.  This
really messes people up when they link to someone using EMACS or
NLS!!! ]

New bits in AC1:
  B6 TL%STA (10X: - ) (T20: set object's accept-advice bit to TL%AAD)
  B7 TL%AAD (10X: - ) (T20: desired state of object's accept-advice bit)

T20: new errs TLNKX2, TLNKX3, TTYX01
---------------------------
TMON:     7      * 2/1X  Tests monitor flags
V4 mods

10X doc doesn't document any of the bits except B0 .SFFAC - fact file enabled.

[ISI CLAIMS:
Content changes to calling ACs:
  AC1/  (T20: function code) (10X: bit mask)
  AC2/  (T20: function value) (10X: bit value mask)
]
[10X DOC SEZ:
        TMON only accepts one argument:
                AC1: Mask word of flags to be tested
        Returns +1: all flags tested are off
```

```
                 +2: any of the flags tested are on
]
        T20: Always returns +1, causes ILINT with error
                TMONX1 for invalid function.

;TOPS20 test required
------------------------
TTMSG:  775     ! 1X    Sends a message to a terminal

[A:
        Send all TeleTypes a MesSaGe via echo buffers.  Outputs a small
        message to all or one TTY without hanging on buffer full (instead,
        chars get lost.)

Accepts AC1:    TTY line number (NOT TTY designator)
                or -1 for all ttys
Returns +1:     Always. Message sent to TTY(s)
Errors: none

        Requires WHEEL or OPERATOR capabilities.  Pseudo-tty's and all lines
        in BINARY mode with REFUSE set are skipped.
]
------------------------
TVPIC:  600     - 1/-   [A] Take TV picture

[A:
Accepts AC1:    Starting memory address
        AC2:    TV mode
                Bit 33 = 0 for camera 1
                         1 for camera 2
                Bit 34 = 0 for single density
                         1 for double density
                Bit 35 = 0 for non-stereo
                         1 for stereo
Returns +1: always
]
------------------------
TWAKE:  771     ! -/2   Wakes a specified job

------------------------
UFPGS:  525     ! -/2   Updates file's pages
V4 mods
------------------------
USAGE:  564     ! -/P1X Writes entries into the accounting data file
V4 mods
------------------------
USRIO:  310     = PM2X  Places program in user I/O mode

10X: err CAPX1 (WHL/OPR needed)
T20: err CAPX2 (WHL/OPR/MNT needed)
------------------------
UTEST:  563     ! -/P1X Test monitor routines

------------------------
UTFRK:  323     * 1X    Resumes a process suspended because of a monitor call

10X: this call not in '73 doc.  Stuff here is from '76 doc.

AC1 new bit:
  B0 UT%TRP     (10X: - )(T20: cause failure return for suspended proc)

T20: new err FRKHX8
------------------------
VACCT:  566     ! -/1X  Validates an account (T20)
------------------------
VACCT:  330     - 2/-   [76] Verify user and account pair (10X)
```

```
[76:  (not in '73 doc)

Accepts AC1: 10X: -1 for self, or user directory #
             T20: -1 for current user, or user #, or directory #
        AC2: T20: byte pointer to account string
             10X:  ditto, or account # in B3-35 if B0-2 = 5.
Returns +1: T20: always, ILINT if error
            10X: failure, error # in 1
        +2: 10X: successful, account is ok for given user

10X: VACCT always skips if AC1 = -1.
10X: VACCT is a NOP for enabled WHEEL's or OPERATORs.

Errors:
        10X:    VACX1     No such user
                VACX2     Invalid name/account pair
        T20: VACCX0-2,MONX02,DELFX6,DIRX1,DIRX3,STRX01,OPNX9,OPNX16
]
--------------------------
VTSOP:  635      ?  Unknown [MT: VTS]

--------------------------
WAIT:   306      == 1    Dismisses process until interrupt occurs

--------------------------
WATDP:  270      - 1/-   Wait for Display Process to Stop

Apparently a BBN hack, see 10X doc p. 4-42
--------------------------
WFORK:  163      = 1X    Waits for processes to terminate

T20: can use -4 as AC1 fork handle to specify any inferior fork
10X: doc says "NOT IMPLEMENTED YET"
--------------------------
WTFOR:  732      - 2/-   [BBN76] Wait for signal

[76: experimental BBN call, p. 11-8]
--------------------------
WILD%:  565      ! -/1X  Compares wild and non-wild strings
V4 addition
--------------------------
XRIR:   601      ?  Unknown

[MT: for KL-extended addressing]
--------------------------
XSIR:   602      ?  Unknown

[MT: ditto]
```

```
------------------------
FDB:                        ; File Descriptor Block format

        The new format of the FDB is:
[Note: WO=Wheel/Operator, OR=anyone with Ownership Rights, WA=anyone
 who has write access to file.  Where there is a difference in the
 10X/T20 access allowed, it is expressed as <10X access>/<T20 access>.
]
 0 .FBHDR        Header Word
                 T20: B29-35 FB%LEN Length of this file's FDB (10X: - )
 1 .FBCTL  Flag word differences:
        B0  FB%TMP or/or+wo
        B1  FB%PRM wo/or+wo
        B2  FB%NEX ==
        B3  FB%DEL ==
        B4  FB%NXF wo/-
        B5  FB%LNG ==
        B6  FB%SHT (10X: compressed page table (no access))(T20: rsvd for DEC)
[76:    B7  FB%DIR (T20: file is directory)
            FDBENV (10X: environment dump file)
        B8  FB%NOD (T20: backup system shouldn't save file)
            FDBSUB (10X: subroutine file)
        B9  FB%BAT (T20: file may have bad pages)
            FDBUND (10X: undeletable file)       ] ;76
        B10-13 (10X: - ) (T20: FB%DIR, FB%NOD, FB%BAT, FB%SDR)
        B14-17 (T20: file class field.  1 = .FBRMS = file is RMS file.)
        B17 FDBEPH (10X: file is ephemeral (wo+or))
        B18-35  (10X: location of file NAME block) (T20: ? )
 2 .FBEXL  Link to next FDB in extension chain
                 10X: <loc of EXTENSION block>,,<pointer to other extensions>
 3 .FBADR  Disk address of index block
                 10X: B5-13 class, B14-35 address.
 4 .FBPRT  File protection (==)
 5 .FBCRE  10X: Creation date/time of version 1 (wo)
           T20: Date and time of last write, updated by monitor when any
                 program writes to file (wo) (10X equiv is .FBWRT)
 6 .FBAUT  T20: Pointer to author name string (R/W with G/SFUST only (or))
           10X: FDBUSE (wo) LH: dir number of last writer
                        RH: use count (+1 for each indirect ptr & saved env)
                 [ISI:   RH: directory number of author ]
 7 .FBGEN  T20: <generation num>,,<if directory, has internal # of dir>
           10X: FDBVER <version num>,,<internal ptr to next version>
10 .FBACT  Account information (R/W with GACTF/SACTF only (or))
11 .FBBYV  File I/O information
        B0-5   FB%RET or/or+wo
        B6-11  FB%BSZ ==
        B12-13 unused
        B14-17 FB%MOD (T20: data mode of last open (wa+or+op))(10X: - )
        B18-35 FB%PGC -/wo
12 .FBSIZ  Number of bytes in the file (==)
13 .FBCRV  Date and time of file creation.  wo/wa+or+wo
14 .FBWRT  Date and time of last user write to file. wo/wa+or+wo
                 T20: modified ONLY by user, not monitor.
                 10X: modified by monitor; this word is == to T20 .FBCRE.
15 .FBREF  Date and time of last non-write reference wo/wa+or+wo
16 .FBCNT  Count of writes, reads. (==)

;;; The following 5 words are used by backup systems.  The 10X format
;;; is described here, since it is too different from the T20 system.
[76:
17 .FBBK0
        B0 Used by DUMPER.
        B1 FDBARC Archive requested.
```

```
          B2 FDBNAR Do not archive this file.
          B3 FDBDMP File dumped, not marked as such yet.
          B4 FDBMRK File archived, not marked as such yet.
          B5 FDBADL Do not delete after archiving.
          B6 FDBAAR File has been archived.
          RH: most recent dump tape number
20 .FBBK1  <first archive tape number>,,<second archive tape number>
21 .FBBK2  Date and time of most recent dump
22 .FBBK3  Date and time of first archive
23 .FBBK4  Date and time of second archive
(T20: new syms .FBBBT and .FBNET for 22 and 23)

24 .FBUSW  User settable word (==)

25 .FBGNL  Link to next generation in chain     (10X: ? )
26 .FBNAM  Pointer to file name block           (10X: ? )
27 .FBEXT  Pointer to extension name block      (10X: ? )
30 .FBLWR  Pointer to last writer name string   (10X: - )
                  (R/W with GFUST/SFUST only)
[10X:   25 .FBLEN  length of FDB]
[T20.3: 31 .FBLEN  length of FDB]
[T20.4: 37 .FBLEN  max length of FDB]
```

[ISI claims .FBLEN for 10X is 30; It's possible the 10X doc didn't bother explaining values 25, 26, 27 because they were illegal for either CHFDB/GTFDB.]

NET: ; Network filename formats etc.

Network filename: NET:[LS[#].[FH-FS][;T]

If LS is omitted:

```
        10X: LS = JFN*8
        T20: LS = JFN*2 (plus 1 if a sending socket)
```

T20: new err on OPENF: OPNX20 "local imp not up or NCP not enabled" Other network errors are identical.

For CLOSF, AC1 B1 = CO%WCL, hang until connection completely closed. Exists on both 10X and T20; just describing here since symbol isn't shown in normal CLOSF writeup.

GDSTS state values:
T20 doc forgets to mention value 0, "DEAD".

TABLES: ; SYSGT-accessible tables

10X doc page 3-9 to 3-13

Name	Index		Contents
JOBDIR	job #	*	T20: - 10X: LH: connected directory number RH: logged in dir # (0 if not logged in)
JOBTTY	job #	*	LH: Controlling TTY #, -1 if none (detached) RH: (10X: top fork) (T20: reserved for DEC)
JOBRT	job #	=	CPU time used by job (neg if no such job)
TTYJOB	line #	=	LH: (10X: 400000+<job#>) (T20: <job#>) otherwise same
TICKPS	0	=	# of clock ticks per sec (divisor for JOBRT)
NCPGS	0	=	# pages of phys user core available in sys

Next 3 tables parallel

DEVNAM	d	=	SIXBIT device name, including unit #
DEVCHR	d	=	dev characteristics wd as per DVCHR JSYS except B5 DV%AV not meaningful

```
DEVUNT   d          =       LH: job # device assigned to (-1 if none)
                                    (T20.4: -2 if resvd for dev allocator)
                            RH: unit # (-1 if dev has no units)

DSKERR   0-n        =       0: # recoverable disk errs
                            1-n: varies depending on type of disk
DRMERR   0-n        =       0: # recoverable drum errs
                            1-n: varies depending on type of disk

SYSVER   0-n        =?      ASCII string identifying system name & version
                            (T20: also date)
VERNUM   0          *       (T20: - )(10X: word with sys version # as number)
PGSTAT   0-2        *       T20: -
                            10X: pager trap info for this process
                                    0: count of pager traps
                                    1: count of page faults
                                    2: time spent in trap routines
SYSTAT   0-n        *       Table is similar up to entry 27:
                            27: (10X: timeword of pending sys shutdown)
                                (T20: sched over head time (#2 in high prec))
                            30: (10X: timeword sys scheduled back up)
                                (T20: idle time (#0 in high prec))
                            31: (10X: - )(T20: lost time (#1 in high prec))
                            32: (10X: - )(T20: user time in high prec)
                            10X doc shows no entries beyond 30.
                    Note: 10X load ave integrals are in floating pt format!
QTIMES   0-n        =       Accumulated runtime of jobs on the n+1 sched queues
                            (10X: 5 sched queues)
JOBNAM   job #      =       LH: unused (T20: reserved)
                            RH: index into system program tables for prog
                                    being used by this job
Next tables parallel, indexed by "spx" garnered above.
SNAMES   spx        =       SIXBIT name of prog, or 0 if unused entry
STIMES   spx        =       total runtime of prog (or 0 if unused)
SPFLTS   spx        =       total # page faults of prog (or 0 if unused)
SWAKES   spx        *       (10X: totals wrt TTY use ("see code")) (T20: - )
SBLKTM   spx        *       (10X:   "     "   "   "    "    "      (T20: - )
SSIZE    spx        !       (T20: Time integral of working set size (10X: - )
SNBLKS   spx        !       (T20: Number of samples in SSIZE value) (10X: - )

ENTFLG   0                  (10X: non-zero if logins permitted) (T20: - )
DBUGSW   0-1        =?      0: value of DBUGSW, 1: value of DCHKSW
LOGDES   0-1        =       0: designator for logging info
                            1: des for job 0 and error info
SYMTAB   0-n        =       SIXBIT table names of all GETAB entries
PTYPAR   0          !       T20: <# PTYs>,,<TTY # of 1st PTY>
DWNTIM   0-1        !       T20: 0: datime for shutdown, 1: datime for back up
BDLTD    0          !       T20: Date/time system was generated
APRID    0          !       T20: processor serial #
JBONT    job #      !       T20: owning job for CRJOB-created jobs
JOBPNM   job #      !       T20: SIXBIT name of program running in this job
HQLAV    ?          !       T20: high queue load averages
LQLAV    ?          !       T20: low queue load averages
NSWPGS   ?          !       T20: default swapping pages
PTYPAR   0          !       T20: <# PTYs in sys>,,<# of 1st PTY>

ARPANET tables:
NETRDY   0-1/7      *       10X: 0-1 same as T20
                            T20: 0-7, has 6 extra entries
IMPHRT   N/36       =       Host ready table (this may be obsoleted by 32-bit
                                    host numbers)
                            T20.4: - (!)
HOSTN    0-n        *       T20.4: - (!)
                                    10X     T20.3
                    server host     B0      =
                    user host       B1      =
                nickname        B3      B2
                host type       B6-8    B3-8
                host number     B9-17   =
                index in HSTNAM RH      =
HSTNAM  0-n     ==      ASCIZ hostname strings, indexed by HOSTN values
                        T20.4: - (!)
LHOSTN  0-1     *       T20.4: - (!)
                        0: Local host #
                        1: 10X: -<# of PTYs>,,<# of 1st PTY>
                           T20.3:<# of NVTs>,,<# of 1st NVT>

        ; Parallel tables indexed by internal connection number (cx)
        ; from IMPLT1
NETLSK  cx      ==      Local skt #
NETFSK  cx      ==      Foreign skt #
NETAWD  cx      *       ; I think the doc is fucked here, too.
                                        10X     T20.3   T20.4
                        Link            B0-8    B9-17   B0-8
                        Fgn hst         B9-17   B0-8    -
                        "Internal uses" B18-26  -       -
                        timeout cntdwn  -       B18-23  B18-23
                        undefined       -       B24-26? B9-17
                        Index "ix"      B27-35  B25-35  B24-35

NETBAL  cx      ==      # bits allocated to connection
NETSTS  cx      ?       10X doc doesn't explain format (T20.3 does)
                        T20.4: - (!)
NETBUF  cx      *       10X: line number for terminal connections
                        T20: <bytes per buffer>,,<buffer loc-1>
NETBTC  cx      ==      # bits sent/rcvd since connection created
NETHST  cx? ix? !       10X: -
                        T20.4: -1 if no foreign host, else == IMPLT5
                        Very unclear what actually indexes this table!

        ; Parallel tables indexed by NETAWD internal index (ix)
IMPLT1  ix      ?       10X: "internal uses"
                        T20:    B0-17 is index "cx" (-1 if control link)
                                B18-19 type: 0 rcv, 1 delete, 2 send, 3 free
                                B20-27 host number
                                B28-35 link number
IMPLT2  ix      ?       10X: "internal uses"
                        T20:    B0-9 flags
                                B10-17 byte size of buffer
                                B18-35 address of input buffer
IMPLT3  ix      ?       10X: "internal uses"
                        T20: <addr of output buffer>,,<msg saved for re-trans>
IMPLT4  ix      *       10X: B0-19 internal uses, B20-35 msg alloc
                        T20: LH addr of current buff, RH msg alloc in bits
IMPLT5  ix      !       10X: -
                        T20.4:  B4-11  network # (Arpanet = 12)
                                B12-27 IMP #
                                B28-35 Host #

[ISI: The LSTDRN entry is also being eliminated.]
(I wonder what that 10X table is? --KLH)

-------------------------
TIMEWORD:       DATA 000        ; Date/time word format


TIMEWORD: Both 10X and T20 represent the date and time as a 36-bit
quantity, but their RHs differ:
        LH: <# days since Nov 17, 1858 (day 0) GMT>
        RH: 10X: <# seconds since midnight (0) GMT>
            T20: <fraction of day since midnight (0) GMT>
```

The 10X time value is easy to understand, but the T20 value is somewhat obscure. It appears that T20 has set things up so that the timeword is an integer fraction; one day is equal to 2**18 units, thus they squeeze the maximum precision out of the RH and ensure that a simple AOS loop at the right time interval will automatically clear the RH and increment the LH when midnight comes around. I'm not sure whether it was worth it.

This is a routine for converting a T20 timeword to a 10X timeword. The code is based on that in the T20 monitor.

```
;; CVTIM - Given T20 timeword in A, returns 10X timeword in A

cvtim:  push p,b
        movei b,(a)             ; Get time-of-day fraction
        push p,c
        muli b,24.*3600.        ; Multiply day-fraction by # secs in a day
        div b,[1,,]             ; then divide by 1.0 (i.e. 2**18)
        cail c,400000           ; If remainder is .5 sec or more
         aoj b,                 ;  then round up to nearest sec.
        pop p,c
        cail b,24.*3600.        ; If resulting # seconds greater than 1 day,
         jrst [ setz b,         ;   reset to midnight
                add a,[1,,0]    ;   of next day.
                jrst .+1]
        hrri a,(b)              ; Put timeword back together.
        pop p,b
        popj p,
```

```
TTYDOC:                        ; General TTY info
General 10X doc re TTYs is in pgs 4-12 to 4-18
Review SFMOD/STPAR bits

Terminal (JFN) mode word             (one per terminal, not one per JFN!)
0       TT%OSP  T20:    SFMOD   output suppress control (0=allow,1=ignore)
                10X:    sys     output active
19      TT%IGN  T20:    SFMOD   Ignore the other TT%WAK (B18-23) bits
                10X:    -
24      TT%ECO  T20:    SFMOD   echo on (10X: see below)
25      TT%ECM  T20:    STPAR   echo mode (0=deferred, 1=immediate)
                10X:    SFMOD   bits 24-25 are taken together:
                                        00 - no echo
                                        01 - immediate echo only
                                        10 - immediate or deferred
                                        11 - immediate and deferred
30      TT%UOC  T20:    STPAR   indicate uppercase by 'X
                10X:    STPAR   indicate lowercase by %X
31      TT%LIC  T20:    STPAR   convert lowercase to upper on input
                10X:    SFMOD     "          "          "
34      TT%PGM  T20:    STPAR   output page (XON/XOFF) mode (0=off, 1=enable)
                10X:    BKJFN   repeat last character
```

CCOC words are same.
Character wakeup classes:

Char	10X	T20
ESC	F+C	F+C+P+A
DEL	P	F+C+P+A

Lowercase input conversion:
10X: if B3 TT%LCA is off, ASCII 175 and 176 are converted to ESC
T20: if B31 TT%LIC is on, ASCII 175-176 are converted to ESC

[NOTE: I suspect the 10X doc may be wrong, the T20 operation is more reasonable.]

```
--------------------------
PSI-CHANNELS:                ; List of PSI bits

8                   (T20: reserved for DEC) (10X: not used)
12       .ICQTA     (T20: panic, disk full or quota exceeded)
                    (10X: file condition 3, unassigned)
13                  (10X: file condition 4, unassigned)
                    (T20: reserved for DEC)
14                  (10X: time of day (NOT IMPLEMENTED YET))
                    (T20: reserved for DEC)
18                  (10X: panic, illegal memory execute)
                    (T20: reserved for DEC)
20       .ICMSE     (10X: panic, machine size exceeded (NOT IMPL YET))
                    (T20: panic, system resources exhausted)
21                  (10X: trap to user ("see SPACS"))
                    (T20: reserved for DEC)
23                  (10X: not used (cannot be used for terminal chars))
                    (T20: assignable to user program)

About terminal interrupt codes:
         10X == T20 except for two (three?) additional codes on T20:
31.      .TICTI     (T20: Typein ) (10X: - [ISI: int on LGOUT by "owned job"
                                       (CRJOB-created job)])
32.      .TICTO     (T20: Typeout) (10X: - )
[ISI:
33.      ?          (T20: int on LGOUT by "owned job") (10X: - )
]

--------------------------
CAPABILITIES:                ; List of capabilities (sigh)

B5       SC%SDV     (T20: process can control special devices)
                    (10X: ditto, but "NOT IMPLEMENTED YET")
B6       SC%SCT     (T20: process can change source of terminal
                       interrupts for other processes)
                    (10X: - [76: same as T20])
B9       SC%SUP     (T20: process can manipulate its superior)
                    (10X: process can do map operations on superior)
         Bits given to inferior which process cannot change for
         itself: 10X: B14-17
                    T20: B17 SC%FRZ
B22      SC%IPC     (T20: user has IPCF privs)       (10X: - )
B23      SC%ENQ     (T20: user has ENQ/DEQ privs)    (10X: - )
B24      SC%NWZ     (T20: user has ARPANET wizard privs) (10X: - )
B25      SC%NAS     (T20: user has absolute ARPANET skt privs) (10X: - )
--------------------------
SAVEDUMP:                    ; Save/Dump format diffs
```

Binary files on 10X have the extension "SAV"; on T20 this is "EXE". This applies to both sharable and non-sharable formats. The non-sharable format is identical on both 10X and T20, but the sharable format is not transferable between 10X and T20 systems.

10X: The T20 manual does not document the 10/50 non-sharable format variant, although it probably does support it. This is identical to the standard non-sharable format except that instead of an entry-vector word, there is a one or two-word block with:

```
         JRST <start>
         JRST <reenter>
```

For a 10/50 format SAVE file (i.e. a JRST seen), entry vector relative position 0 means use JOBSA (loc 120), and 1 means use JOBREN (loc 124); all other relative positions are illegal.

Note: files actually saved by a TOPS-10 monitor have the contents of location 41 shifted to location 122. After GETing such a file on

TENEX, it may be necessary for the user to move this word to location 41.

Sharable SAVE files:
10X and T20 are MUCH different here. There is not much point in comparing them since they are too different. Consequently the 10X format will simply be described completely here:

10X SSAVE files contain sequences of pages. The first two pages (0 and 1) are used for the page table; the remaining pages (2, 3, ...) are data pages. Usually all of the control information will fit into page 0, so page 1 will not exist (a "hole") in the file.

Word (page 0)

0:	1000,,<N = number of data pages>	
1-N:	<map info words>	
N+1:	<len>,,<addr>	; Entry vector word

Map info word:

B0-8	Access bits, left 9 bits of AC3 in PMAP
B9-17	Fork page, AC2 of PMAP
B18-35	File page, AC1 of PMAP

DEVICES: ; LPT, MTA, PLT, PTR, PTP, CDR

LPT control code differences:
(p. 4-3 of 10X doc)

Code	Name	Function
0	null	(10X: prints nothing) (T20: -)
37	EOL	(10X: same as CR, LF) (T20: -)
177	escape	(10X: quotes next char) (T20: -)

Also, all of the functions which skip a line (12,13,14,20,21,22,23,24) are different in that
10X: stays in same column
T20: returns to first column

No status bits are documented for the 10X LPT.
Sigh.

MTA:
The MTA status bits are almost exactly the same in 10X and T20.

B27-28 MT%DEN	T20: value 0 is "system default" (probably 1600) 10X: value 0 is 800 bpi All other values are same.
B32 MT%NSH	T20: Selected mode or density not supported 10X: -
B33-35	(T20: -) (10X: 1 (thats what doc sez))

The 10X doc says "Magtape I/O is currently available in TENEX only via the DUMPI/O JSYS's".

It would be nice to know what sort of magtape formats are available under 10X which correspond to T20's dump/indus-compat/ANSI/SIXBIT modes. In particular, if there isn't a way to get indus-compatible (8-bit byte) mode instead of dump mode, 10X is an incredible screw....

PLT/PTR/PTP: (Plotter, paper tape reader/pucnh)
These are only defined for 10X. What documentation there is, is on pages 4-6 and 4-7 of the 10X doc.

CDR: (card reader)
There is no 10X doc for such a device.

TTY:
10X defines no status bits.

T20 defines B35 GD%PAR - TTY will tolerate (?!) a parity bit.

CONVENTIONS: ; Designators etc.

10X and T20 "designators" are identical for all known purposes.

```
Source/Destination Designators                      (abbr: src/dest des)
   File Designators - subset of src/dest des        (abbr: file des)
      Device Designators - subset of file des       (abbr: dev des)
```

10X "Fork Handle" is T20 "process handle". All values are exactly the same, but T20 permits more relative handles (400001-400777) than does 10X (400001-400020).

10X: [76: -6 is "Universal fork handle" (?). A fork can use it to specify the top fork in its group (See CFGRP)]

Timeword values are different. See the TIMEWORD entry.

File names are very similar, but these exceptions:

- * 10X "ext" is T20 "typ" (nomenclature difference)
- * 10X "ver" is T20 "gen" (")
- * Separator char for ext/typ and ver/gen:
  10X: ";" T20: "." (main difference!)
- * T20 uses the device field to specify logical names and "structures" as well as devices.
- * T20 (V3 and up) has subdirectories. The format however is still compatible, since the directory path is specified within "<>" and just looks like a single directory name. Programs just have to allow periods in directory specifications.
- * T20 (V4 up) has "node names" -- the delimiter is "::" (two chars) and the node name must precede the device name.

See the GTJFN entry for a list of chars allowed in filenames.

Editing characters during GTJFN, possibly other input:

	10X	T20
erase one char	^A	DEL
erase to punct		^W
erase field	^W	
abort & retry	^X	^U
retype spec	^R	^R

Archive/virtual disk system:
V4 T20 has this. 10X doesn't.

File access differences:

10X and T20 have the same triad of Owner, Group, World.
The 6 bits of each have the same meanings except for:

```
   02 FP%DIR    T20: directory listing access
                10X: access as specified in page table of file
```

10X has only 36 groups since each group is one bit of a word. I think T20 has a "group list" and consequently more.

One possible difference: it appears that in 10X for group access to work, you have to be CONNECTED to the directory in question, otherwise you are just a random "other", even if you share a group with the directory. Someone verify this.

There is probably some other difference to do with connection giving you ownership access, but the doc is no help.

The 10X doc does not explain anything about directory access as opposed to file access.

File closing: (must add T20 stuff)

10X: The CLOSF

10X: The CLOSF JSYS will close a file but will not release the JFN if a page of that file is mapped.

10X: RLJFN will release a JFN only if the file is closed and no pages are mapped.

10X: CLZFF can be told to release the JFN which has been closed even if a page of that file is mapped.

10X: If a file was closed but the JFN was not released because a page was mapped the JFN will be released as soon as the page is unmapped either by becoming private or by doing an explicit PMAP into the same page.

Execute-only files/processes:

V4 T20 has this stuff. 10X doesn't have any of it as far as I know, although the doc has plenty of references to execute-access bits.

Error handling:

T20 has a way to get around all those obnoxious ILINT's. JSYS instructions may be followed by one of two no-op instructions which the monitor will interpret on errors:

```
        JUMP 16,addr    (ERJMP addr)    works like a JUMPA addr
        JUMP 17,addr    (ERCAL addr)    works like a PUSHJ 17,addr
```

These work for either case of error return: illegal instruction PSI or Return +1. ERJMP and ERCAL are OPDEFed in MONSYM, the TOPS-20 equivalent of STENEX.

MT: note that ERJMP can be simulated at user level by smart PSI handler; might include code to do so

KLH: the T20 V4 manual says (p. 1-2) that ERJMP/ERCAL will now work after machine instructions for the conditions (1) Illegal instruction, (2) Illegal mem read, and (3) Illegal mem write. It also says that "If an ERJMP or ERCAL is taken on an error from a JSYS, any AC's that would normally hve contained an error code may be unreliable. Using the GETER JSYS is the sure way to find the error code in such a case". Holy shit!!!!!

STARTUP:

[ISI:

If it is necessary for a subsystem to know what operating system it is running under, so it may choose values, courses of action, etc., the following determining code should appear at initialization.

```
        MOVE AC,[112,,11]       ;Entry 112 in table 11
        CALLI AC,41             ;GETAB
         MOVEI AC,30000         ; assume TENEX on fail
```

On a successful return (+2), AC will contain:

```
        10000   if TOPS10
        20000   if ITS
        30000   if TENEX
        40000   if TOPS20
```

]

Remaining 10X doc refs:

pg 3-5	Error-string format scheme
pg 3-11	SYSTAT table contents
pp 4-6,7	Devices PLT, PTR, PTP
pp 4-36 to 42	BBN display-process hacks
pp 5-24 to 26	Pager traps & bits

pg 6-13   Fork status table
pg 10-6   DSKOP call
pp 10-2 to 5   CRDIR, GTDIR format

TCP-JSYS-DOC:
[TCP:

TCP JSYS CALLING SEQUENCES

Bill Plummer 24 April 1978
Jon Postel   22 August 1979
Bill Plummer 20 June 1980

In the following, a "JCN" may be thought of much as a JFN is for files. A "Connection Block" (referred to below) is a 3-word block:

Word-0:   16-bit Local Port
Word-1:   8-bit Foreign Network and 24-bit Foreign Host
Word-2:   16-bit Foreign Port

These values are right justified in the 36 bit word.

All JSYS's take flags in the left half of AC 1.
Not all JSYS's look at all of the flags. Flag bits are:

Bit-0:   RH has JCN rather than pointer to connection block
Bit-1:   Wait for the JSYS to complete.
Bit-5:   ForceSync -- cause SYN to be sent when OPEN executed.
Bit-6:   Persist -- keep resending SYN packet
Bit-7:   Return statistics (STAT call only)

Some JSYSs take a "Retransmission Parameters" word.

This controls the retransmission function. The right half is the initial retransmission interval which is to be used. If the right half is 0, the initial interval will be computed based on the measured round trip time. The left half of the parameters control word has two 9-bit quantities. In computing the next retransmission interval from the previous one, the TCP multiplies by the number in the leftmost 9 bits and then divides by the number in the next 9-bit byte.

Common backoff functions are:

SRI PR demo: Numerator=1, Denominator=1, Initial Interval=3.
(3 seconds constant retransmission interval with no backoff)

BBN (vanilla): Numerator=3, Denominator=2, Initial interval=0.
(Used in "average" conditions involving congested gateways and few dropped packets. 150% backoff from best guess initial interval).

BBN (old): Same as above but 200% backoff.
Quickly hits the 1 minute maximum interval and turns into slow, constant period retransmission).

TCP Data Buffer Ring Format (SEND, RECV):

Word-0:   Flags,,unused (typically ptr to next buffer header)
Word-1:   0,,Address of data buffer
Word-2:   Word/Byte count for this buffer

Flags:
Done: Cleared when TCP receives this buffer. Set when TCP has finished with it.
Error: Buffer has an error associated with it.
EOL: Send an end-of-letter with this buffer. Or, end-of-letter received with this buffer.

WordMode: Buffer is formatted as 36-bit bytes. Off if buffer has four 8-bit bytes per word. [Not Implemented Yet]

Flag Bit Assignments:

Bit 0:	Error
Bit 1:	Local
Bit 2:	Permanent
Bits 3-7:	Error Number
Bits 8-11:	Unused
Bit 12:	Done
Bits 13-15:	Unused
Bit 16:	EOL
Bit 17:	Word Mode

The error numbers are listed under "Error Returns".

Error Returns

When a JSYS does an error return (returns +1 instead of +2), AC1 contains an error code. This code is an 8-bit number composed as follows:

Flag Bit Assignments:

Bit 28:	Error
Bit 29:	Local
Bit 30:	Permanent
Bits 31-35:	Error Number

The error numbers are:

0	Unknown Error
1	Argument Error in JSYS (no access, bad JCN, etc.)
3	Connection Not Open
4	Temporarily Out of Resources
6	Connection Already Exists
7	Connection Error or Rejected (No such TCB either here or there.)
9	Transmission Timeout
12	Connection Closed or Closing (Closed remotely.)
15	Bad Buffer Argument
17	Bad Argument to CHANL
20	Funny pointer to STAT (wraps around memory, etc)
21	Bad Transfer Size to STAT
29	Cannot change security level (SCSLV)
31	TCP Not Available

The Error bit indicates if an error occured, for example error number 12 might not have the Error bit set in response to a CLOSE call. The Local bit indicates if the situation is local to this host or is due to the remote host. The Permanent Bit indicates if the situation is permanent, or temporary.

] ;TCP

This page will become a section about machine-language dependencies introduced by non-KA processors. For example, ADJBP, ADJSP, DMOVX, and so forth. Include code to simulate.

[MT: there is F2 ucode available to simulate all the KL opcodes, although not all sites have it yet.]

```
! This page is a TECO (EMACS) macro to generate the numerical
  listing on the next page.
!
[1[2[3[4[5[6[7
! First find range of old numerical listing if any, so we can delete it
  when done with new listing !
        zj 0u6 0u7
        -:sNUMERICAL-JSYS-PAGE:"L -s
```

```
 .u6 zu7 '

! Now gobble all JSYS defs into a q-reg !
        0j sALPHABETICAL-JSYS-PAGE: .u4 s
 .u5    ! Set search bounds !
        0u3 <q4,q5:fb
--------------------------
 ; .u1 1l q1,.@x3 (.-2)u4>              ! Found a JSYS entry, gobble line !

! Now stick defs at end of buffer, and crunch them !
        zj 14.i i
        .u4 g3 q4j              ! Stick on page at end of buffer !
        < :s
; -1l iDEFJS  .u1 s: -1d i, <(.-q1)-7; i >      ! Set up name !
        <(1a-11.)*(1a-40.)"N 0;' 1d>            ! Flush whitespace to num !
 1r .u2 .u1 s
        q1j <(q2-(q1+2))"G 0;' i0 %2>   ! Flesh numbers out with leading 0s!
        q2j i   ; 1a-11."E 1d ' ! Make comments nice !
        1l >

! Now sort them numerically !
        q4,zfsboundaries                        ! Set buffer bounds !
```

```
ws,$<1a-40."N 0;' 1c>$ w \ 1l   ! Sort buffer numerically !
        0,zfsboundaries                     ! Reset to whole buffer !

! Now tidy up the listing with various comments and suchlike.
  This part could be infinitely hairy, probably. !
        q4j ! ---- This section checks the JSYS numbers ---- !
        fsibaseu3 8fsibase       ! Save input radix, set to octal !
        0u1 0u2
        [..E 8u..E       ! Set radix for "\" to octal !
        < q2u1           ! Bump current val to prev val !
         :sDEFJS; s, <1a-40."N 0;'1c> \u2       ! Get number for next JSYS !
                1+q1-q2"E !<!>'                 ! If +1, all's well, get next
                q1-q2"E 0l i;; Duplicate:
                        q2\ 15.i 12.i 1l !<!> '
                1+q1-q2"N 0l i;; Missing:   (q1+1)\
                        2+q1-q2"N i- (q2-1)\ '  15.i 12.i 1l !<!>'
        >
        q3fsibase ]..E           ! Restore radix stuff !

        q4j ! ---- This section adds various comments ---- !
        i
;;; NUMERICAL- iJSYS-PAGE:
;; This page is a table of JSYS definitions in numerical order.
;; It could be extracted and used as part of a program, with a DEFJS
;; macro defined as appropriate for the application.

;; DEFSYM       ERJMP,<JUMP 16,0>
;; DEFSYM       ERCAL,<JUMP 17,0>
;; DEFSYM       JSYS,<104000,,0>


! Okay, we seem to have done all right, so flush old listing !
        q6,q7k
]7]6]5]4]3]2]1
```

```
;;; NUMERICAL-JSYS-PAGE:
;; This page is a table of JSYS definitions in numerical order.
;; It could be extracted and used as part of a program, with a DEFJS
;; macro defined as appropriate for the application.

;; DEFSYM       ERJMP,<JUMP 16,0>
;; DEFSYM       ERCAL,<JUMP 17,0>
;; DEFSYM       JSYS,<104000,,0>

DEFJS LOGIN, 001          ;= 2    Logs in a job
DEFJS CRJOB, 002          ;* p2   Creates a new job
DEFJS LGOUT, 003          ;= 2    Kills a job
DEFJS CACCT, 004          ;= 2    Changes account designator
DEFJS EFACT, 005          ;= P2   Makes an entry in the FACT file
DEFJS SMON,  006          ;* P1X  Sets monitor flags
DEFJS TMON,  007          ;* 2/1X Tests monitor flags
DEFJS GETAB, 010          ;* 2    Gets a word from a monitor table
DEFJS ERSTR, 011          ;* 3/3X Converts error number to string
DEFJS GETER, 012          ;* 1    Returns the last error in a process
DEFJS GJINF, 013          ;= 1    Gets current job information
DEFJS TIME,  014          ;= 1    Returns time system has been up
DEFJS RUNTM, 015          ;= 1X   Returns runtime of process or job
DEFJS SYSGT, 016          ;== 1   Returns information for a system table
DEFJS GNJFN, 017          ;* 2    Gets the next JFN
DEFJS GTJFN, 020          ;* 2    Gets a JFN
DEFJS OPENF, 021          ;* 2    Opens a file
DEFJS CLOSF, 022          ;* 2    Closes a file
DEFJS RLJFN, 023          ;= 2    Releases JFNs
DEFJS GTSTS, 024          ;= 1    Gets a file's status
DEFJS STSTS, 025          ;* 2    Sets a file's status
DEFJS DELF,  026          ;* p2   Deletes files
DEFJS SFPTR, 027          ;= 2    Sets file's pointer position
DEFJS JFNS,  030          ;* 1X   Translates a JFN to a string
DEFJS FFFFP, 031          ;* 1X   Finds first free page in file
DEFJS RDDIR, 032          ;-? 2/- Read Device Directory
DEFJS CPRTF, 033          ;- 2/-  Changes protection of a file
DEFJS CLZFF, 034          ;* 1/1X Closes the process' files
DEFJS RNAMF, 035          ;=? 2   Renames a file
DEFJS SIZEF, 036          ;== 2   Gets the size of a file
DEFJS GACTF, 037          ;= 3    Gets account designator of file
DEFJS STDIR, 040          ;-? 3/- Translates string to directory number
DEFJS DIRST, 041          ;* 2X/2 Translates a directory number to a string
DEFJS BKJFN, 042          ;= 2    Backs up pointer by one byte
DEFJS RFPTR, 043          ;== 2   Reads file's pointer position
DEFJS CNDIR, 044          ;- 2/-  Connects job to a directory (replaced by ACCES)
DEFJS RFBSZ, 045          ;* 1X/2 Reads files's byte size
DEFJS SFBSZ, 046          ;* 1X/2 Sets file's byte size
DEFJS SWJFN, 047          ;= 1X   Swaps two JFNs
DEFJS BIN,   050          ;== 1XDE        Performs byte input
DEFJS BOUT,  051          ;= 1XQD Performs byte output
DEFJS SIN,   052          ;= 1XDE/1XDE    Performs string input
DEFJS SOUT,  053          ;= 1XDQ Performs string output
DEFJS RIN,   054          ;== 1XDE        Performs random input
DEFJS ROUT,  055          ;= 1XDQ Performs random output
DEFJS PMAP,  056          ;* 1X   Maps pages
DEFJS RPACS, 057          ;* 1X   Reads a page's accessibility
DEFJS SPACS, 060          ;* 1X   Sets a page's accessibility
DEFJS RMAP,  061          ;* 1X   Obtains a handle on a page
DEFJS SACTF, 062          ;= 2    Sets account designator of file
DEFJS GTFDB, 063          ;* 1X   Gets a File Descriptor Block
DEFJS CHFDB, 064          ;* p1X  Changes a File Descriptor Block
DEFJS DUMPI, 065          ;* 2    Reads data in unbuffered data mode
DEFJS DUMPO, 066          ;* 2    Writes data in unbuffered data mode
DEFJS DELDF, 067          ;* p1/p1X       Expunges deleted files
DEFJS ASND,  070          ;= 2    Assigns a device
DEFJS RELD,  071          ;= 2    Releases a device
DEFJS CSYNO, 072          ;- 2/-  Create device name synonym
DEFJS PBIN,  073          ;= 1XDE Inputs the next byte
DEFJS PBOUT, 074          ;= 1XDQ Outputs the next byte
DEFJS PSIN,  075          ;- ?    Inputs a string
DEFJS PSOUT, 076          ;= 1XDQ Outputs a string
DEFJS MTOPR, 077          ;* 1X   Performs device-dependent functions
DEFJS CFIBF, 100          ;= 1X   Clears the input buffer (TTYs only)
DEFJS CFOBF, 101          ;= 1X   Clears the output buffer (TTYs only)
DEFJS SIBE,  102          ;* 2X/2 Skips if input buffer is empty
DEFJS SOBE,  103          ;* 2X/2 Skips if output buffer is empty
DEFJS DOBE,  104          ;* 1X   Dismisses until output buffer is empty
DEFJS GTABS, 105          ;-? 1X/-        Get tab settings for file
DEFJS STABS, 106          ;-? 1X/-        Set tabs for file
DEFJS RFMOD, 107          ;= 1X   Reads a file's mode
DEFJS SFMOD, 110          ;* 1X   Sets a file's mode
DEFJS RFPOS, 111          ;= 1X   Reads terminal's position
DEFJS RFCOC, 112          ;= 1X   Reads file's control character output
DEFJS SFCOC, 113          ;= 1    Sets file's control character output
DEFJS STI,   114          ;= p1X  Simulates terminal input
DEFJS DTACH, 115          ;== 1   Detaches a terminal from a job
DEFJS ATACH, 116          ;* p2   Attachs a terminal to a job
DEFJS DVCHR, 117          ;* 1X   Retrieves device characteristics
DEFJS STDEV, 120          ;== 2   Translates string to device designator
DEFJS DEVST, 121          ;* 2    Translates a device designator to a string
DEFJS MOUNT, 122          ;-? 2/- Mounts a device
DEFJS DSMNT, 123          ;-? 2/- Dismounts a device
DEFJS INIDR, 124          ;-? 2   Initializes device directory
DEFJS SIR,   125          ;= 1X   Sets software interrupt table addresses
DEFJS EIR,   126          ;= 1X   Enables software interrupt system
DEFJS SKPIR, 127          ;== 2X  Tests the state of the software interrupt syste
DEFJS DIR,   130          ;= 1X   Disables software interrupt system
DEFJS AIC,   131          ;= 1X   Activates software interrupt channels
DEFJS IIC,   132          ;= 1X   Initiates software interrupts on specified char
DEFJS DIC,   133          ;= 1X   Deactivates software interrupt channels
DEFJS RCM,   134          ;== 1X  Reads the channel word mask
DEFJS RWM,   135          ;* 1X   Reads waiting channel interrupt word mask
DEFJS DEBRK, 136          ;* 1/1X Dismisses current software interrupt
DEFJS ATI,   137          ;= 1X   Assigns a terminal code to an interrupt channel
DEFJS DTI,   140          ;* 1X   Deassigns a terminal code
DEFJS CIS,   141          ;== 1   Clears the interrupt system
DEFJS SIRCM, 142          ;* 1X   Sets inferior reserved channel mask
DEFJS RIRCM, 143          ;== 1X  Reads inferior reserved channel mask
DEFJS RIR,   144          ;== 1X  Reads software interrupt table addresses
DEFJS GDSTS, 145          ;= 1X   Gets device's status
DEFJS SDSTS, 146          ;* 1X   Sets device's status
DEFJS RESET, 147          ;* 1    Resets/initializes the current process
DEFJS RPCAP, 150          ;= 1X   Reads process capabilities
DEFJS EPCAP, 151          ;== 1X  Enables process capabilities
DEFJS CFORK, 152          ;* 2    Creates an inferior process
DEFJS KFORK, 153          ;= 1X   Kills a process
DEFJS FFORK, 154          ;= 1X   Freezes processes
DEFJS RFORK, 155          ;= 1X   Resumes a process
DEFJS RFSTS, 156          ;* 1X   Reads a process' status
DEFJS SFORK, 157          ;* 1X   Starts a process
DEFJS SFACS, 160          ;= 1X   Sets process' ACs
DEFJS RFACS, 161          ;= 1X   Reads process' ACs
DEFJS HFORK, 162          ;* 1X   Halts a process
DEFJS WFORK, 163          ;= 1X   Waits for processes to terminate
DEFJS GFRKH, 164          ;* 2/2X Gets process handle
DEFJS RFRKH, 165          ;* 1X/2 Releases a process handle
DEFJS GFRKS, 166          ;* 2/2X  Gets process structure
DEFJS DISMS, 167          ;== 1   Dismisses the process
```

```
        DEFJS HALTF, 170        ;= 1    Halts the current process
        DEFJS GTRPW, 171        ;* 1X   Gets trap words
        DEFJS GTRPI, 172        ;== 1X  Get trap information
        DEFJS RTIW,  173        ;== 1X  Reads terminal interrupt word
        DEFJS STIW,  174        ;= 1X   Sets terminal interrupt word
        DEFJS SOBF,  175        ;* 2X/2 Skips if output buffer is full
        DEFJS RWSET, 176        ;== 1   Releases the working set
        DEFJS GETNM, 177        ;== 1   Returns the program name currently being used
        DEFJS GET,   200        ;* 1X   Gets a save file
        DEFJS SFRKV, 201        ;* 1X   Starts process using its entry vector
        DEFJS SAVE,  202        ;= 1X   Saves a file as nonsharable
        DEFJS SSAVE, 203        ;= 1X   Saves a file as sharable
        DEFJS SEVEC, 204        ;= 1X   Sets entry vector
        DEFJS GEVEC, 205        ;== 1X  Gets entry vector
        DEFJS GPJFN, 206        ;== 1X  Gets the primary JFNs
        DEFJS SPJFN, 207        ;= 1X   Sets the primary JFNs
        DEFJS SETNM, 210        ;= 1    Sets program name
        DEFJS FFUFP, 211        ;= 2    Finds first used page in file
        DEFJS DIBE,  212        ;* 1X   Dismisses until input buffer is empty
        DEFJS FDFRE, 213        ;- 1X/- File directory free space
        DEFJS GDSKC, 214        ;* 1X   Gets disk count
        DEFJS LITES, 215        ;-? PM1X/-      Displays data in console lights
        DEFJS TLINK, 216        ;* p2   Controls terminal linking
        DEFJS STPAR, 217        ;* 1X/1 Sets terminal parameters
        DEFJS ODTIM, 220        ;* 1X   Outputs date and time
        DEFJS IDTIM, 221        ;== 2   Inputs date and time
        DEFJS ODCNV, 222        ;* 1X   Outputs date and time conversion
        DEFJS IDCNV, 223        ;* 2    Inputs date and time conversion
        DEFJS NOUT,  224        ;= 2    Outputs an integer number
        DEFJS NIN,   225        ;* 2    Inputs an integer number
        DEFJS STAD,  226        ;* P1X  Sets system date and time
        DEFJS GTAD,  227        ;* 1    Gets current date and time
        DEFJS ODTNC, 230        ;* 1X   Outputs date/time without converting
        DEFJS IDTNC, 231        ;* 2    Inputs date/time without converting
        DEFJS FLIN,  232        ;* 2    Inputs floating-point number
        DEFJS FLOUT, 233        ;= 2    Outputs floating-point number
        DEFJS DFIN,  234        ;= 2    Inputs double-precision floating point number
        DEFJS DFOUT, 235        ;= 2    Outputs double-precision floating point number
        ;; Missing:  236-237
        DEFJS CRDIR, 240        ;* p2/p1X       Creates, changes, or deletes a director
        DEFJS GTDIR, 241        ;* P1X/p1X      Gets information of directory entry
        DEFJS DSKOP, 242        ;* P1X  Specifies disk transfers in hardware terms
        DEFJS SPRIW, 243        ;=? -/P1X       Sets the priority word
        DEFJS DSKAS, 244        ;* P2X  Assigns disk addresses
        DEFJS SJPRI, 245        ;* 2/1X Sets job's priority
        DEFJS STO,   246        ;! -/1X Simulates terminal output
        DEFJS ARCF,  247        ;! -/p1X        Archive/virtual disk operations
        ;; Missing:  250-257
        DEFJS ASNDP, 260        ;- 2/-  Assign Display Process (not in T20)
        DEFJS RELDP, 261        ;- 1/-  Release Display Process
        DEFJS ASNDC, 262        ;- 2/-  Assign Display Console (not in T20)
        DEFJS RELDC, 263        ;- 1/-  Release Display Console
        DEFJS STRDP, 264        ;- 2/-  Start Display Process
        DEFJS STPDP, 265        ;- 1/-  Stop Display Process
        DEFJS STSDP, 266        ;- 2/-  Set Status of Display Process
        DEFJS RDSDP, 267        ;- 2/-  Read Status of Display Process
        DEFJS WATDP, 270        ;- 1/-  Wait for Display Process to Stop
        ;; Missing:  271
        DEFJS GTNCP, 272        ;= 2    Get NCP information (new)
        DEFJS GTHST, 273        ;= 2    Get ARPANET hostname information (new)
        DEFJS ATNVT, 274        ;= 2    Creates ARPANET Network Virtual Terminal Connec
        DEFJS CVSKT, 275        ;== 2   Converts ARPANET local socket to absolute form
        DEFJS CVHST, 276        ;= 2    Converts ARPANET host number to primary name
        DEFJS FLHST, 277        ;* PN1/PN1X     Flushes an ARPANET host
        DEFJS GCVEC, 300        ;= 1X   Gets entry vector of compatibility package
        DEFJS SCVEC, 301        ;= 1X   Sets entry vector of compatibility package
```

```
	DEFJS STTYP, 302		;* 1X   Sets the terminal type number (TTYs only)
	DEFJS GTTYP, 303		;= 1X   Gets the terminal type number (TTYs only)
	DEFJS BPT,   304		;-? 1/- Breakpoint
	DEFJS GTDAL, 305		;* 1X   Gets disk allocation of a directory
	DEFJS WAIT,  306		;== 1   Dismisses process until interrupt occurs
	DEFJS HSYS,  307		;* PM2  Halts the system
	DEFJS USRIO, 310		;= PM2X Places program in user I/O mode
	DEFJS PEEK,  311		;* PM2  Obtains monitor data
	DEFJS MSFRK, 312		;* P1X  Starts a process in monitor mode
	DEFJS ESOUT, 313		;== 1XQD        Outputs an error string
	DEFJS SPLFK, 314		;= 2    Splices a process structure
	DEFJS ADVIZ, 315		;- 2/-  Set up for link advice
	DEFJS JOBTM, 316		;- 1/-  Get job runtime
	DEFJS DELNF, 317		;* 2    Retains specified number of generations of a f
	DEFJS SWTCH, 320		;-? 1/- Returns data switch setting
	DEFJS TFORK, 321		;* 2/1X Sets and removes monitor call intercepts
	DEFJS RTFRK, 322		;* 2/1X Returns the handle of JSYS-trapped fork
	DEFJS UTFRK, 323		;* 1X   Resumes a process suspended because of a monito
	DEFJS SCTTY, 324		;* 2/1X Changes controlling terminal
	DEFJS CFGRP, 325		;- 2/-  [76] Create fork group
	DEFJS OPRFN, 326		;- P2/- [76] Perform 10X operator functions
	DEFJS CGRP,  327		;- P2/- [76] Change pie-slice group
	DEFJS VACCT, 330		;- 2/-  [76] Verify user and account pair (10X)
	DEFJS GDACC, 331		;- 2/-  [76] Get default account for user
	DEFJS ATGRP, 332		;- 2/-  [76] Get pie-slice group for account
	DEFJS GACTJ, 333		;- 2/-  [76] Get account for job
	DEFJS GPSGN, 334		;- 2/-  [76] Get pie-slice group name for job
	DEFJS GFACC, 335		;- 2/-  [76] Determine access to directory or file
	DEFJS SETER, 336		;* 1X   Sets the last error in a process
	DEFJS GPLD,  337		;- p2/- [76] Get pie-slice group load average
	;; Missing: 340-357
	DEFJS ASPTY, 360		;- 2/-  [A] Assign PTY
	DEFJS REPTY, 361		;- 2/-  [A] Release a PTY
	DEFJS PSTI,  362		;- 2/-  [A] PTY simulate TTY input
	DEFJS PSTO,  363		;- 2/-  [A] Simulate PTY output
	DEFJS SIBF,  364		;- 2/-  [A] Skip if input buffer full
	;; Missing: 365-440
	DEFJS PUPI,  441		;?  Unknown [PARC: Xerox PUP protocol hack]
	DEFJS PUPO,  442		;?  Unknown [PARC: Xerox PUP protocol hack]
	DEFJS PUPNM, 443		;?  Unknown [PARC: Xerox PUP protocol hack]
	DEFJS CPUTL, 444		;- 1/-  Get CPU utilization for fork/job
	;; Duplicate: 444
	DEFJS PUPSK, 444		;?  Unknown [PARC: Xerox PUP protocol hack]
	;; Missing: 445-477
	DEFJS RSCAN, 500		;! -/2  Accepts a new string or uses the last string as
	DEFJS HPTIM, 501		;! -/2  Returns values of high precision clocks
	DEFJS CRLNM, 502		;! -/p2 Defines or deletes a logical name
	DEFJS INLNM, 503		;! -/2  Lists job's logical names
	DEFJS LNMST, 504		;! -/2  Converts a logical name to a string
	DEFJS RDTXT, 505		;! -/?  Read Text (obsoleted by RDTTY, TEXTI)
	DEFJS SETSN, 506		;! -/2  Sets system name for a process
	DEFJS GETJI, 507		;! -/2  Gets specified job information
	DEFJS MSEND, 510		;! -/p2 Sends an IPCF message
	DEFJS MRECV, 511		;! -/p2 Receives an IPCF message
	DEFJS MUTIL, 512		;! -/p2 Performs IPCF control functions
	DEFJS ENQ,   513		;! -/p2 Places request in resource queue
	DEFJS DEQ,   514		;! -/p2 Removes request from resource queue
	DEFJS ENQC,  515		;! -/p2 Obtains status of resource queue
	DEFJS SNOOP, 516		;! -/2  Performs system analysis
	DEFJS SPOOL, 517		;! -/P2  Defines and initializes input spooling
	DEFJS ALLOC, 520		;! -/P2 Allocates a device
	DEFJS CHKAC, 521		;! -/2  Checks access to a file
	DEFJS TIMER, 522		;! -/2  Sets time limit for a job
	DEFJS RDTTY, 523		;! -/2  Reads data from primary input designator
	DEFJS TEXTI, 524		;! -/2  Reads input from a terminal or a file
	DEFJS UFPGS, 525		;! -/2  Updates file's pages
```

```
DEFJS SFPOS, 526       ;! -/1  Sets terminal's position
DEFJS SYERR, 527       ;! -/P1X        Writes data to the system error file
DEFJS DIAG,  530       ;! -/PM2        Reserves or releases hardware channels
DEFJS SINR,  531       ;! 1XDE Performs record input
DEFJS SOUTR, 532       ;! 1XDQ Performs record output
DEFJS RFTAD, 533       ;! -/1X Reads file's time and dates
DEFJS SFTAD, 534       ;! -/p1X        Sets file's time and dates
DEFJS TBDEL, 535       ;! -/1X Deletes entry from command table
DEFJS TBADD, 536       ;! -/1X Adds entry to command table
DEFJS TBLUK, 537       ;! -/1X Looks up entry in command table
DEFJS STCMP, 540       ;! -/1  Compares two strings
DEFJS SETJB, 541       ;! -/1X Sets job parameters
DEFJS GDVEC, 542       ;! -/1X Gets entry vector of RMS
DEFJS SDVEC, 543       ;! -/1X Sets entry vector of RMS
DEFJS COMND, 544       ;! -/1X Parses a command
DEFJS PRARG, 545       ;! -/1X Reads/sets process argument block
DEFJS GACCT, 546       ;! -/p1X        Gets current account designator
DEFJS LPINI, 547       ;! -/P1X        Loads VFU or translation RAM
DEFJS GFUST, 550       ;! -/1X Returns author and last writer name strings
DEFJS SFUST, 551       ;! -/p1X        Sets author and last writer name string
DEFJS ACCES, 552       ;! -/p1X        Specifies access to a directory
DEFJS RCDIR, 553       ;! -/1X Translates string to directory number
DEFJS RCUSR, 554       ;! -/1X Translates string to user number
DEFJS MSTR,  555       ;! -/p1X        Performs structure-dependent functions
DEFJS STPPN, 556       ;! -/1X Translates string to project-programmer number
DEFJS PPNST, 557       ;! -/1X Translates project-programmer number to string
DEFJS PMCTL, 560       ;! -/P1X        Controls physical memory
DEFJS PLOCK, 561       ;! -/PM1X       Locks physical pages
DEFJS BOOT,  562       ;! -/P1X        Do functions required for loading front
DEFJS UTEST, 563       ;! -/P1X        Test monitor routines
DEFJS USAGE, 564       ;! -/P1X        Writes entries into the accounting data
DEFJS WILD%, 565       ;! -/1X Compares wild and non-wild strings
DEFJS VACCT, 566       ;! -/1X Validates an account (T20)
DEFJS NODE,  567       ;! -/p1X        Performs DECnet network functions
DEFJS ADBRK, 570       ;! -/1X Controls address breaks
DEFJS SINM,  571       ;?  Reads data from block-mode terminals
DEFJS SOUTM, 572       ;?  Writes data to block mode terminals
DEFJS SWTRP%,573       ;! -/1X Traps for arithmetic underflow or overflow cond
DEFJS GETOK%,574       ;! -/1X Requests access to a protected resource
DEFJS RCVOK%,575       ;! P1X  Retrieves access request from GETOK queue
DEFJS GIVOK%,576       ;! -/P1X        Grants access to a protected resource
DEFJS SKED%, 577       ;! p1X  Performs services relating to the class schedul
DEFJS MTU%,  600       ;! -/P1X        Performs various functions for MT: devi
;; Duplicate: 600
DEFJS TVPIC, 600       ;- 1/- [A] Take TV picture
DEFJS XRIR,  601       ;?  Unknown
DEFJS XSIR,  602       ;?  Unknown
DEFJS SETNT, 603       ;- 1/-  [A] Sets network on or off
;; Missing:  604
DEFJS CNTSZ, 605       ;- 1/-  [A] Count size of job
DEFJS SKUSR, 606       ;- 1/-  [A] Set job % of CPU for a user
;; Missing:  607-610
DEFJS PSTIN, 611       ;- 1/-  String input from TTY, with editing
;; Missing:  612-624
DEFJS DELCH, 625       ;- 4/-  [A] Deletes a character
;; Missing:  626-627
DEFJS IIT,   630       ;- 1X   [A] Initiate interrupt with timing delay
;; Missing:  631-633
DEFJS GTBLT, 634       ;- 2/-  [A] Get system tables with BLT
DEFJS VTSOP, 635       ;?  Unknown [MT: VTS]
DEFJS RTMOD, 636       ;?  Unknown [MT: VTS]
DEFJS STMOD, 637       ;?  Unknown [MT: VTS]
DEFJS RTCHR, 640       ;?  Unknown [MT: VTS]
DEFJS STCHR, 641       ;?  Unknown [MT: VTS]
;; Missing:  642-676
DEFJS DBGIM, 677       ;? (number conflict, which is right? 677 from MIDAS)
DEFJS HANDS, 700        ;- 2/-  [A] Get multiple monitor tables
;; Missing:  701-727
DEFJS GTSIG, 730        ;- 2    [BBN76] Get signal ID
DEFJS RLSIG, 731        ;- 2/-  [BBN76] Release signal ID
DEFJS WTFOR, 732        ;- 2/-  [BBN76] Wait for signal
DEFJS SIGNL, 733        ;- 2/-  [BBN76] Generate signal
;; Missing:  734-737
DEFJS SEND,  740        ;?  [TCP] Send data (Internet/TCP)
DEFJS RECV,  741        ;?  [TCP] Receive data (Internet/TCP)
DEFJS OPEN,  742        ;?  [TCP] Open connection  (Internet/TCP)
DEFJS CLOSE, 743        ;?  [TCP] Close connection (Internet/TCP)
DEFJS SCSLV, 744        ;?  Unknown (Internet/TCP?)
DEFJS STAT,  745        ;?  [TCP] Get status of connection (Internet/TCP)
DEFJS CHANL, 746        ;?  [TCP] Set connection channels (Internet/TCP)
DEFJS ABORT, 747        ;?  [TCP] Abort connection (Internet/TCP)
DEFJS SNDIM, 750        ;* 2    Sends a message to ARPANET special message queu
DEFJS RCVIM, 751        ;* 2    Retrieves message from ARPANET special message
DEFJS ASNSQ, 752        ;== N2  Assigns ARPANET special message queue
DEFJS RELSQ, 753        ;== 1   Deassigns ARPANET special message queue
DEFJS SNDIN, 754        ;?  Unknown (Internet version of SNDIM?)
DEFJS RCVIN, 755        ;?  Unknown (Internet version of RCVIM?)
DEFJS ASNIQ, 756        ;?  Unknown (Internet version of ASNSQ?)
DEFJS RELIQ, 757        ;?  Unknown (Internet version of RELSQ?)
;; Missing:  760-765
DEFJS DBGIM, 766        ;- PN2/-        [A] Debug IMP
;; Duplicate: 766
DEFJS METER%,766        ;! -/1X Returns EBOX/MBOX clock values
DEFJS SMAP,  767        ;?  Unknown
DEFJS CADSK, 770        ;- 1/-  [A] Convert phys disk addr to virtual addr
;; Duplicate: 770
DEFJS THIBR, 770        ;! -/2  Blocks the current job
DEFJS TWAKE, 771        ;! -/2  Wakes a specified job
DEFJS MRPAC, 772        ;? 1    Returns access of resident monitor
DEFJS SETPV, 773        ;?  Unknown
DEFJS DSKCV, 774        ;- 1/-  [A] Convert hardware disk addr to virtual or v
;; Duplicate: 774
DEFJS MTALN, 774        ;! -/P1X        Associates magnetic tape drive with log
DEFJS TTMSG, 775        ;! 1X   Sends a message to a terminal
;; Missing:  776
DEFJS EXEC,  777        ;=? 1   [A] Enter mini-EXEC
;; Duplicate: 777
DEFJS MDDT%, 777        ;= -/P1X        Enter Monitor DDT
```