Course: 28-29 Aug 79

John Swanson

NOS/VE
INTERNALS OVERVIEW

CURRENT STATUS

DOCUMENTATION

* NOS/180 RELEASE 1 DR ARH2711

* NOS/180 ERS COMMAND INTERFACE REV.4 NOV78

* NOS/180 ERS PROGRAM INTERFACE REV.4 DEC78

* NOS/180 DESIGN SPECIFICATION REV.3 MAR79
  1 - DESIGN DIRECTION
  2 - DESIGN ANALYSIS

* INTRODUCTION TO CYBER 180. A.D.& C. A.WILSON.

* NOS/180 CONVENTIONS. ARH SHELDON FEWER.

* MACHINE INDEPENDENT G.D.S. A.D.& C.

COURSE OVERVIEW

DAY1

INTRODUCTION

HARDWARE OVERVIEW

SOFTWARE FEATURES
OVERVIEW

SYSTEM STRUCTURE
& TABLES

JOB FLOW

PROGRAM MANAGEMENT

CPU MONITOR

RESOURCE MANAGEMENT

COURSE OVERVIEW

DAY2

OBJECT CODE

NOS/VE FILE SYSTEM

DUAL-STATE

MAINTENANCE SERVICES

OPERATOR COMMUNICATION

CY180 HARDWARE OVERVIEW

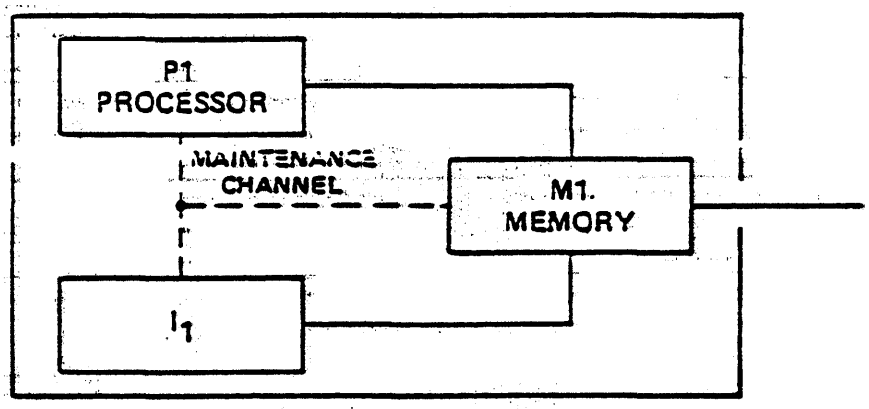
HARDWARE OVERVIEW

* SYSTEMS S1,S2,S3 & THETA

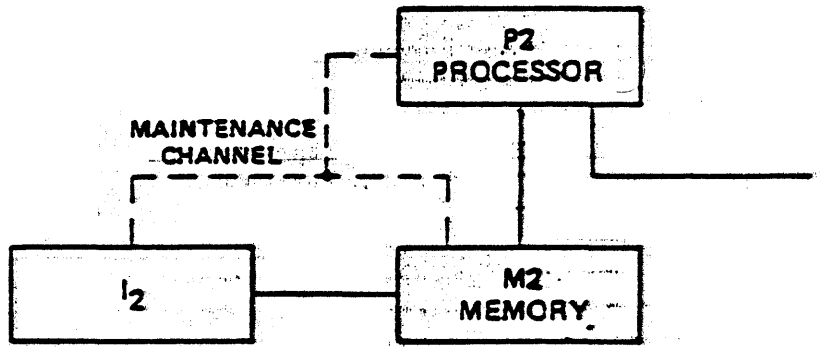
* CY180 CONFIGURATION

* COMPONENTS
  - PROCESSOR
  - MEMORY
  - IOU
  - PERIPHERALS

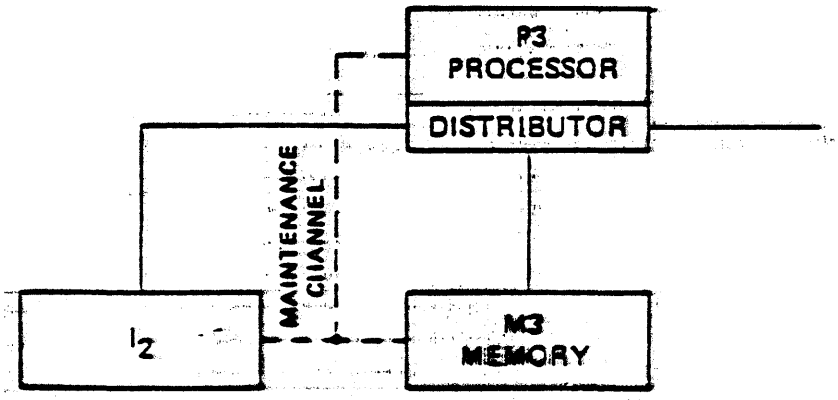
DIAGRAM - S1, S2, AND S3



S1 SYSTEM

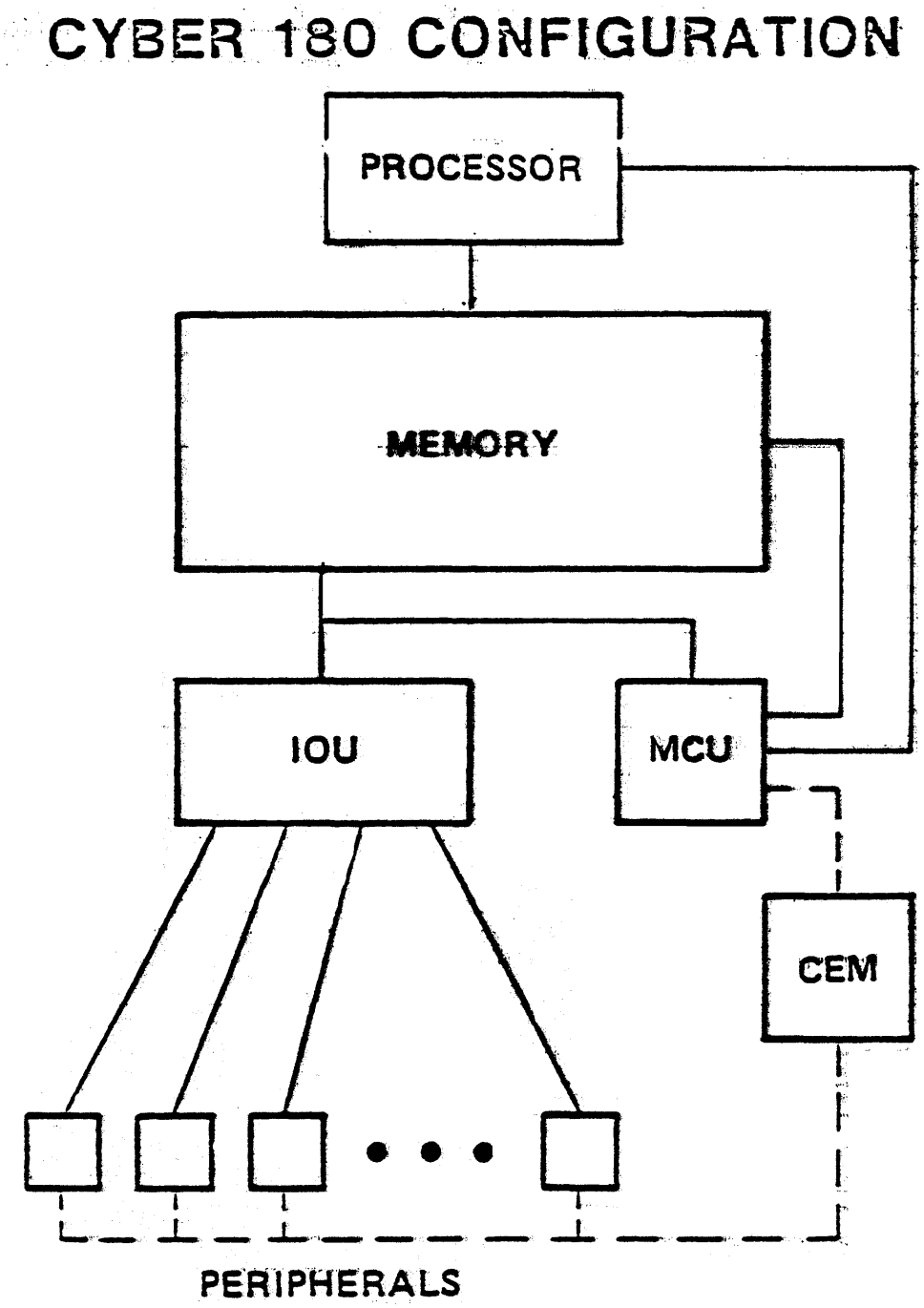


S2 SYSTEM



S3 SYSTEM

DIAGRAM - CYBER 180 CONFIGURATION



PROCESSOR

P1 P2 P3 Theta

- 1, 3, 9, 34 TIMES A CYBER 172
- REGISTER OPERATION:
  - -- 16 48 BIT A REGISTERS
  - -- 16 64 BIT X REGISTERS
- 128 TWO-REGISTER INSTRUCTIONS
  - -- 16 SCIENTIFIC
  - -- 18 BDP
  - -- 76 GENERAL
  - -- 18 SYSTEM
  - -- 16 AND 32 BIT FORMATS
  - -- OPTIONAL VECTOR INSTRUCTIONS (THETA)
- EXECUTES CYBER 170 INSTRUCTION SET UNDER MICRO PROGRAM CONTROL
- SECURITY STATE SWITCHING WITH 15 LEVELS
- PROCESSOR-CONTROLLED 16K – 32K CACHE MEMORY
- DUAL PROCESSOR OPTION

MEMORY

- 64 BIT WORD WITH SECDED
- BYTE ADDRESSABLE
- VERY LARGE REAL MEMORIES
  - -- 1MB TO 64MB CONFIGURATIONS
  - -- PHYSICALLY ORGANIZED BY PAGES
- VIRTUAL MEMORY CAPABILITY
  - -- LOGICALLY ORGANIZED INTO SEGMENTS
  - -- SHARABLE, SECURE
  - -- 4096 SEGMENTS OF 2 BILLION BYTES EACH

INPUT OUTPUT UNIT

- 5 TO 20 PERIPHERAL PROCESSORS
  - -- 170 UPWARD COMPATIBLE INSTRUCTION SET
  - -- 16 BIT MEMORY, INSTRUCTIONS, PATHS
- 8 TO 24 CHANNELS
  - -- CYBER 170 COMPATIBLE
  - -- 16 BIT PARALLEL HIGH SPEED CHANNEL (5-7 MB)
- SUPPORTS MAINTENANCE SYSTEMS
  - -- DEDICATED MAINTENANCE CONTROL UNIT
  - -- SYSTEM CONSOLE - LOCAL/REMOTE
  - -- CONFIGURATION MONITOR
  - -- TWO PORT MULTIPLEXER

DIAGRAM - IOU

PERIPHERALS

THREE CLASSES

- LOW PERFORMANCE -- COMMUNICATION LINES

  UNIT RECORD EQUIPMENT

  PRINTER
  CARD READER
  CRT TERMINALS

- INTERMEDIATE PERFORMANCE -- 6000 CHANNEL

  MAGNETIC TAPE
  SERIAL HEAD DISK
  MASS STORAGE SYSTEM

- HIGH PERFORMANCE -- 50 M BIT CHANNEL

  PARALLEL HEAD DISK
  SECONDARY STORAGE

  CCD, BUBBLE OR EBAM

NOS/VE SOFTWARE OVERVIEW

VIRTUAL ENVIRONMENT (DUAL-STATE)

* VIRTUAL ENVIRONMENT IS THE SHARED USE OF A CY180 MAINFRAME BY
  - NOS/VE & NOS/170
  - NOS/VE & NOS-BE/170

* VIRTUAL ENVIRONMENT PROVIDES A MIGRATION AID FROM CY170 TO CY180

* VIRTUAL ENVIRONMENT PROVIDES CY180 PRODUCT FEATURES PRIOR TO DEVELOPING NATIVE CY180 EQUIVALENTS. E.G.
  - JOB ENTRY & DISPOSAL
  - OPERATOR COMMUNICATION
  - NETWORKING

* SYSTEM RESOURCES ARE PARTITIONED BETWEEN CY170 & CY180 VIRTUAL ENVIRONMENTS

NOS/VE PROVIDES THE USER INTERFACE TO CY180 HARDWARE

* A JOB IS THE UNIT IN WHICH WORK IS PRESENTED TO NOS/VE

* JOB PROVIDES THE ENVIRONMENT FOR
  - ACCESS CONTROL
  - RESOURCE ALLOCATION
  - LOGGING AND ACCOUNTING FOR SYSTEM USE
  - RECOVERY
  - SCHEDULING

* A TASK IS THE BASIC EXECUTION ELEMENT KNOWN TO NOS/VE
  - COMPONENT OF A JOB
  - SEPERATE ADDRESS SPACE

* NOS/VE SERVICES ARE REQUESTED THROUGH
  - COMMAND INTERFACE
  - PROGRAM INTERFACE

NOS/VE MANAGES ALL SYSTEM RESOURCES

* VIRTUAL MEMORY MANAGEMENT
  - SECURITY FEATURES OF THE SEGMENTED ADDRESS SPACE
  - SEGMENT MANAGEMENT INTERFACE

* REAL MEMORY MANAGEMENT
  - PAGE FAULT PROCESSING
  - PAGE FRAME SUPERVISION

* CPU SCHEDULING & DISPATCHING

* DEVICE MANAGEMENT
  - RMS SPACE ALLOCATION INTERFACE
  - NON-PREEMPTIVE DEVICE SCHEDULING INTERFACE
  - PHYSICAL DEVICE ACCESS CONTROL

* FILE SYSTEM
  - PERMANENT AND TEMPORARY FILES
  - FILE ACCESS CONTROL
  - FILE ACCESS METHODS

NOS/VE PROVIDES AN ACCESS CONTROL STRATEGY

* USER IDENTIFICATION AND VALIDATION
  - USER NAMES
  - VALIDATION LIMITS
  - FAMILY, ACCOUNT AND PROJECT HIERARCHY

* FILE SYSTEM SECURITY
  - FILE OWNER DEFINES ACCESS RIGHTS
  - ACCESS RIGHTS ARE ATTRIBUTES OF BOTH FILE & USER

* SEGMENT ACCESS CONTROL
  - ACCESS RIGHTS ARE ATTRIBUTES OF THE SEGMENT AND OF THE REQUESTOR
  - ENFORCED BY CY180 HARDWARE

[illegible] files have segment attributes
[illegible]

NOS/VE PROVIDES A STANDARD APPROACH TO ERROR HANDLING

* CONSISTENT ERROR REPORTING
  - AT THE PROGRAM INTERFACE
  - AT THE COMMAND INTERFACE

* CONDITIONS - IMMEDIATE TRANSFER OF CONTROL WHEN AN EXCEPTION CONDITION OCCURS
  - CONDITION MAY BE HARDWARE, SYSTEM, LANGUAGE OR USER TYPE
  - TASK ENVIRONMENT TAKES DEFAULT ACTION IF USER PROCESSING NOT SELECTED

* STATUS - THE DETECTING PROCEDURE RETURNS ALL ERROR INFORMATION IN A STATUS VARIABLE
  - THE CALLER MAKES THE DECISION ON ERROR HANDLING
  - ALL INTERFACES MUST SUPPLY A STATUS VARIABLE

* EXTERNAL MONITORING OF ERRORS
  - MAINTENANCE CONTROL UNIT
  - CONFIGURATION AND ENVIRONMENT MONITOR

NOS/VE IS DESIGNED TO SUPPORT FUTURE GROWTH

* CYBIL IMPLEMENTATION LANGUAGE

* PROCEDURE INTERFACE IS UNIFORM
  - LOCAL PROCEDURES
  - TASKS
  - ERROR HANDLING

* INTERFACE RULES & CONVENTIONS

* VIRTUAL ADDRESS SPACE SIZE

* SECURITY AND CODE-SHARING FEATURES OF CY180 SEGMENTED ADDRESS SPACE

* JOB AND TASK STRUCTURE

* SYNCHRONOUS AND ASYNCHRONOUS TASK EXECUTION FACILITY

* SIMPLE REPLACEMENT OF SYSTEM MODULES

* DE-EMPHASIS OF PPUS

NOS/VE CONCEPTS

* BASIC ELEMENTS
  - JOBS
  - PROGRAMS/TASKS
  - FILES
  - SEGMENTS

* NOS/VE ENVIRONMENTS

* CY180 PROTECTION MECHANISM
  - ADDRESS SPACE SEGMENTATION
  - SEGMENT PROTECTION ATTRIBUTES
  - SEGMENT ASSIGNMENT

JOBS

* A JOB IS THE BASIC UNIT OF WORK PRESENTED TO NOS/VE
  - USER ORIENTED SCL STATEMENTS & VARIABLES TO DIRECT THE COMPUTING PROCESS
  - SYSTEM ORIENTED STRUCTURES TO IDENTIFY, MONITOR AND ACCOUNT FOR THE COMPUTING SESSION

* THE JOB IS THE BASIS FOR ACCESS CONTROL
  - A JOB RUNS FOR A SINGLE USER VALIDATED AT JOB INITIATION TIME
  - USER VALIDATION DEFINES JOB LIMITS

* THE JOB IS THE BASIC UNIT FOR
  - RESOURCE ALLOCATION
  - SCHEDULING
  - ACTIVITY LOGGING & ACCOUNTING
  - RECOVERY

* EVERY JOB CONSISTS OF MULTIPLE TASKS FOR
  - PROTECTION
  - SERIALIZATION
  - MULTI-PROGRAMMING

PROGRAMS/TASKS

* A PROGRAM IS A SET OF OBJECT AND/OR LOAD MODULES ORGANIZED TO PERFORM A SPECIFIC FUNCTION

* A TASK IS THE EXECUTION OF A PROGRAM
  - TASKS ARE DYNAMICALLY CREATED & DESTROYED
  - ACTIVE TASKS ARE PROTECTED FROM ONE ANOTHER AND COMPETE FOR CPU WITHIN A SCHEDULING SCHEME
  - A JOB MAY INVOKE MULTIPLE ASYNCHRONOUS TASKS
  - TASKS CAN COMMUNICATE TO SYNCHRONIZE THEIR ACTIVITIES

* EACH TASK HAS
  - AN EXECUTION CONTROL BLOCK DEFINING CURRENT EXECUTION STATUS (INCLUDES EXCHANGE PACKAGE)
  - A SEGMENT DESCRIPTOR TABLE DEFINING THE TASK ADDRESS SPACE
  - A JOB LOCAL QUEUE FOR INTER-TASK COMMUNICATION

FILES

* FILES ARE LOGICAL CONTAINERS FOR DATA
  - PERMANENT
  - TEMPORARY (DISCARDED AT JOB TERMINATION)

* A FILE IS OWNED BY A SINGLE USER WHO DEFINES ACCESS RIGHTS

* ACCESS TO A FILE IS PROVIDED BY BASIC ACCESS METHODS
  - RECORD ACCESS
  - PHYSICAL ACCESS
  - SEGMENT ACCESS

* BASIC FILE ORGANIZATIONS ARE
  - SEQUENTIAL
  - BYTE ADDRESSABLE

SEGMENTS

* VIRTUAL MEMORY IS ORGANIZED AS A SET OF SEGMENTS
  - TEMPORARY (DISCARDED AT JOB TERMINATION)
  - PERMANENT

* A TEMPORARY SEGMENT MAY BE ALLOCATED FOLLOWING A USER REQUEST FOR STORAGE

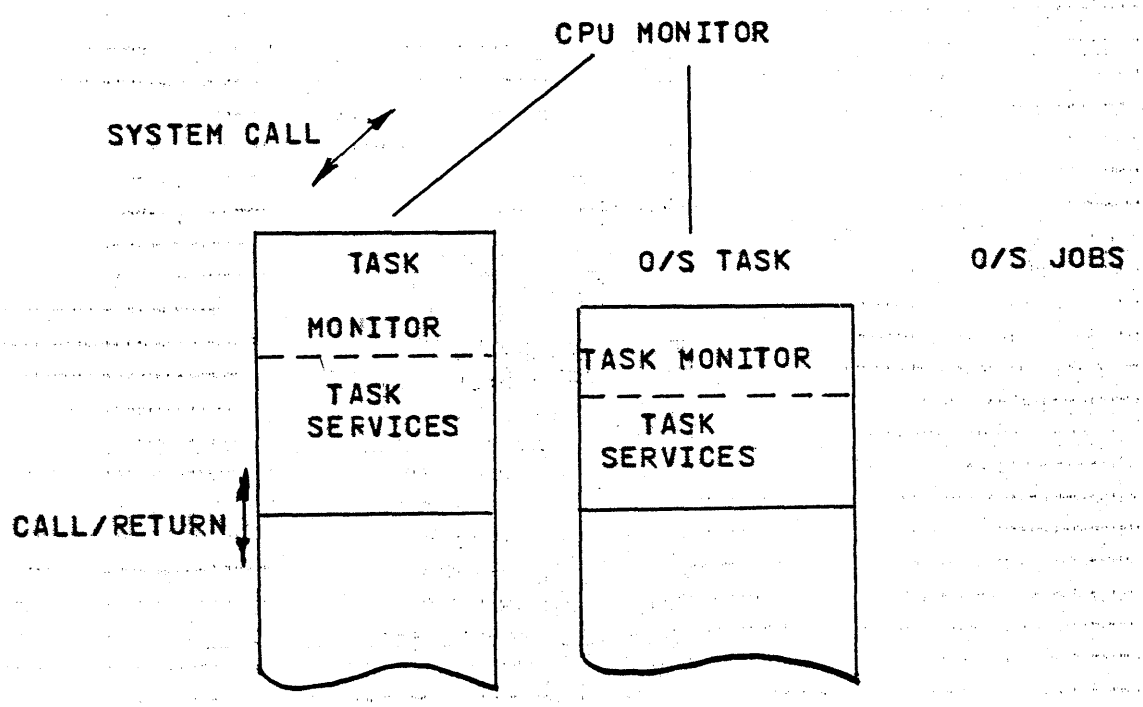
* A FILE SEGMENT (PERMANENT OR TEMPORARY) MAY BE ASSOCIATED WITH THE TASK BY OPENING WITH SEGMENT LEVEL ACCESS

* ALL ACTIVE SEGMENTS HAVE A SEGMENT NUMBER WHICH IS PART OF THE PROCESS VIRTUAL ADDRESS

RING	SEG. NUMBER	BYTE NUMBER

* ALL ACTIVE SEGMENTS ARE BACKED BY MASS STORAGE EQUIVALENT TO THE CURRENT SEGMENT SIZE (DETERMINED BY THE LARGEST BYTE ADDRESS)

NOS/VE STRUCTURE



* NOS/VE SYSTEM ENVIRONMENT INCLUDES
  - CPU MONITOR
  - TASK SERVICES/TASK MONITOR
  - OPERATING SYSTEM TASKS
  - OPERATING SYSTEM JOBS

CPU MONITOR ENVIRONMENT

* CPU MONITOR IS THE INTERFACE WITH CY180 HARDWARE
  - INTERRUPT HANDLING
  - I/O REQUEST QUEUEING
  - DEVICE DRIVER INTERFACE
  - PAGE MANAGEMENT
  - BASIC CPU SCHEDULING
  - CPU DISPATCHING
  - INTER-TASK COMMUNICATION (SIGNALS)
  - CHANGING TASK STATUS

* TASK INTERFACE TO CPU MONITOR IS VIA TASK SERVICES AND BY HARDWARE INTERRUPTS

TASK SERVICES

* TASK SERVICES PROVIDE THE REQUESTORS VIEW OF NOS/VE SERVICES
  - FILE MANAGEMENT (E.G. OPEN A FILE)
  - ACCESS METHODS (E.G. WRITE TO A TERMINAL)
  - PROGRAM MANAGEMENT (E.G. EXECUTE A PROGRAM)
  - SEGMENT MANAGEMENT (E.G. ALLOCATE A SEGMENT)
  - RESOURCE ALLOCATION (E.G. REQUEST A TAPE)

* TASK SERVICES INTERFACE TO CPU MONITOR BY THE HARDWARE EXCHANGE (SYSTEM CALL) MECHANISM

  opposed to call mechanism

* USER MODULES INTERFACE TO TASK SERVICES BY THE CALL/RETURN MECHANISM

* TASK SERVICES COMPRISE TASK MONITOR
  - EXTENSION OF CPU MONITOR INTO TASK'S ADDRESS SPACE
  - INTERFACE TO SYSTEM TABLES. E.G. SEGMENT TABLE, EXCHANGE PACKAGE, SIGNALS
  - INTERFACE TO CPU MONITOR

  AND A PACKAGE OF SERVICES DEFINED FOR EACH JOB THE SET IS PART OF THE "JOB TEMPLATE" FOR THAT JOB TYPE
  - USER JOB
  - SYSTEM JOB
  - ON-LINE DIAGNOSTIC JOB

OPERATING SYSTEM JOBS & TASKS

* TO MAXIMIZE INTEGRITY OF CPU MONITOR, WHERE POSSIBLE NOS/VE FUNCTIONS ARE PROVIDED BY SYSTEM JOBS AND TASKS

* O/S SERVICES WHICH MAY BE PROVIDED ASYNCHRONOUSLY ARE IMPLEMENTED AS O/S TASKS. E.G.

  - JOB MANAGEMENT

  - JOB SCHEDULING

* O/S TASKS PROVIDE OPERATING SYSTEM SERVICES BUT EXECUTE IN USER MODE FULLY PROTECTED BY THE SEGMENT STRUCTURE

* O/S SERVICES WHICH REQUIRE ACCESS CONTROL AND OTHER JOB ATTRIBUTES ARE PROVIDED AS O/S JOBS. E.G.

  - OPERATOR COMMUNICATIONS

  - MAINTENANCE JOB

PROTECTION WITHIN TASK ADDRESS SPACE

* NOS/VE TASK STRUCTURE COMBINES USER, RUN-TIME AND O/S MODULES WITHIN THE SAME ADDRESS SPACE SO THERE MUST BE PROTECTION BETWEEN THEM

* PROTECTION IS PROVIDED BY SEGMENTING THE ADDRESS SPACE AND EXPLOITING THE CY180 RING HARDWARE
  - SEGMENT ACCESS ATTRIBUTES
  - SEGMENT KEYS AND LOCKS
  - RING BRACKETS FOR READ, WRITE AND EXECUTE ACCESS

* ACCESS TO A SEGMENT IS ONLY PERMITTED WHEN ALL THREE PROTECTION TESTS ARE PASSED
  - REQUESTED ACCESS MODE IS PERMITTED TO BOTH SEGMENT AND REQUESTOR
  - REQUESTOR HAS CORRECT KEY TO A LOCKED SEGMENT
  - SEGMENT HAS BEEN ASSIGNED RING BRACKETS WITHIN RANGE OF THE REQUESTOR

SEGMENT USAGE AND RING ASSIGNMENT/ENFORCEMENT

* VIRTUAL MEMORY IS ALLOCATED IN SEGMENTS, EACH TASK ADDRESS SPACE IS A SET OF VIRTUAL MEMORY SEGMENTS

* BY CAREFUL ASSIGNMENT OF ATTRIBUTES TO SEGMENTS A PROTECTION FRAMEWORK MAY BE DEFINED FOR THE TASK

* ACCESS CONTROL ATTRIBUTES LIMIT THE TYPE OF REFERENCES TO THE ADDRESS SPACE SPANNED BY EACH SEGMENT
  - READ: PERMITS REFERENCE ONLY
  - WRITE: PERMITS ASSIGNMENT ONLY
  - EXECUTE: PERMITS EXECUTION OF INSTRUCTIONS
  - BINDING

* LOCK ATTRIBUTE LIMITS ACCESS TO THE ADDRESS SPACE SPANNED BY A SEGMENT TO:
  - SEGMENTS HAVING THE MASTER KEY
  - SEGMENTS HAVING THE MATCHING KEY

* READ RING LIMIT GIVES THE HIGHEST NUMBERED RING FROM WHICH THE SEGMENT MAY BE READ

* WRITE RING LIMIT GIVES THE HIGHEST NUMBERED RING FROM WHICH THE SEGMENT MAY BE WRITTEN

* EXECUTE RING BRACKET GIVES THE RANGE OF RINGS IN WHICH THE SEGMENT MAY EXECUTE

* CALL RING LIMIT GIVES THE HIGHEST NUMBERED RING FROM WHICH THE SEGMENT MAY BE CALLED

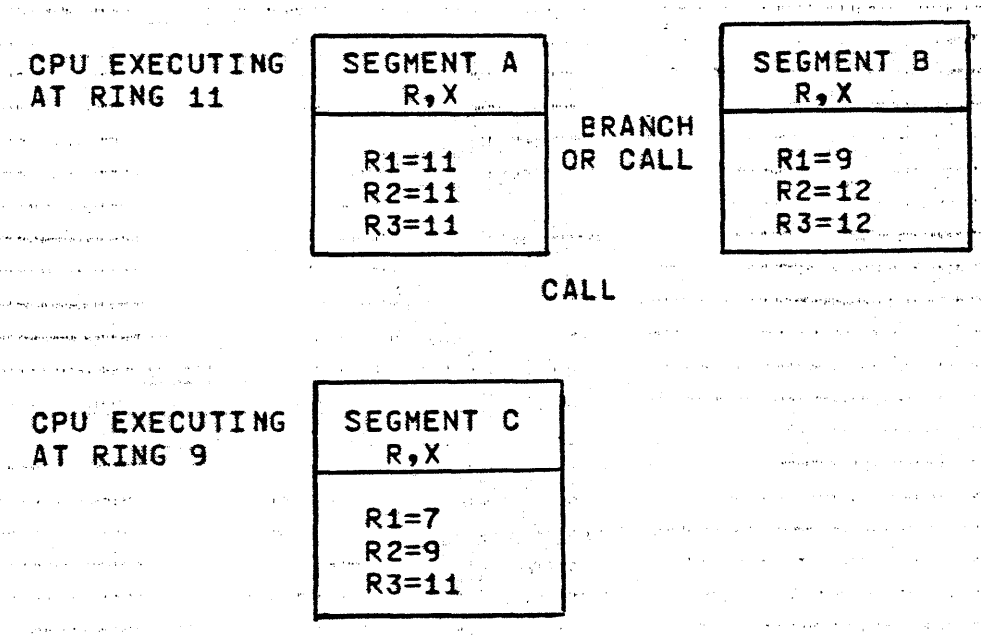
EXAMPLE - INTER-SEGMENT TRANSFERS

* MAY BRANCH IF CPU IS CURRENTLY EXECUTING WITHIN THE EXECUTE BRACKET OF THE TARGET SEGMENT

  EXECUTE ACCESS IF:- R1 .LE. P.RING .LE. R2

* MAY CALL IF CPU IS CURRENTLY EXECUTING AT A HIGHER NUMBERED RING, BUT WITHIN THE CALL RING LIMIT

  CALL ACCESS IF:- R2 .LT. P.RING
  AND P.RING .LE. R3



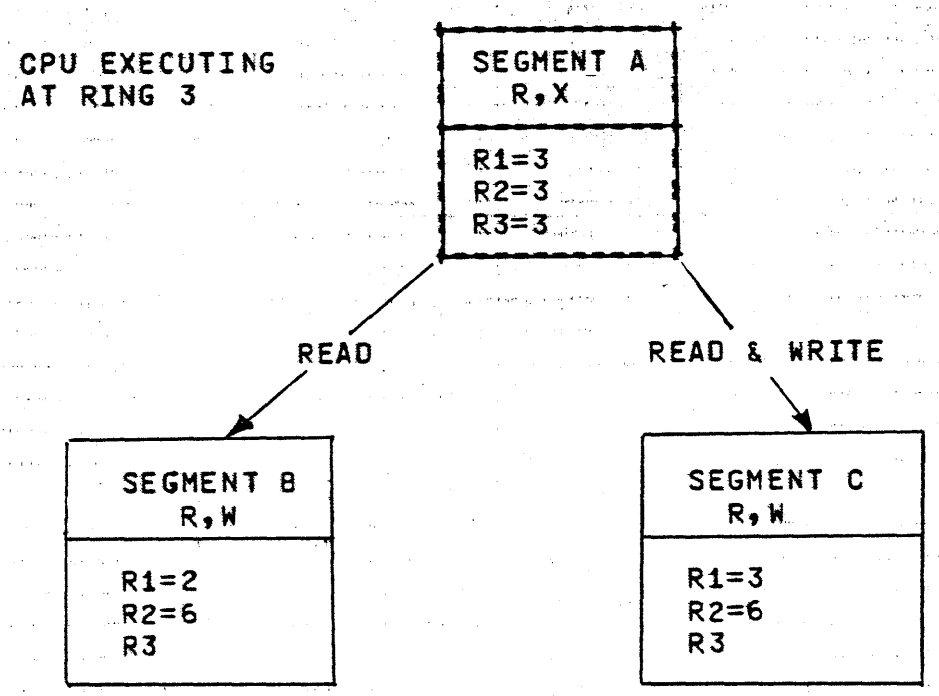
EXAMPLE - INTER-SEGMENT READ/WRITE

* MAY READ A SEGMENT IF THE CPU IS CURRENTLY EXECUTING WITHIN IT'S READ BRACKET

  READ ACCESS IF:- 1 .LE. P.RING .LE. R2

* MAY WRITE A SEGMENT IF THE CPU IS CURRENTLY EXECUTING WITHIN IT'S WRITE BRACKET

  WRITE ACCESS IF:- 1 .LE. P.RING .LE. R1



EXAMPLE OF SEGMENT ASSIGNMENT FOR USER MODULES

OBJECT TEXT FROM COMPILER	SEGMENT ASSIGNMENT BY LOADER
IDENT RECORD MODULE MAIN	SEGMENT N (R,X)
DESCRIPTIVE HEADER INFORMATION	CODE SECTION MAIN CODE SECTION SUBRN
	SEGMENT N+1 (B)
	BINDING SECTION MAIN BINDING SECTION SUBRN
CODE (READ,EXECUTE) • INSTRUCTIONS	SEGMENT N+2 (R,W)
BINDING (BINDING) • BASE ADDRESS OF EACH SECTION • PROCEDURE DESCRIPTIONS	STATIC DATA MAIN ANY NAMED COMMON
WORKING STORE (READ,WRITE) •COMMON BLOCKS	SEGMENT N+3 (R)
	LITERALS MAIN LITERALS SUBRN
STATIC (READ,WRITE) • MODIFIABLE DATA	SEGMENT N+4 (R,W)
LITERAL (READ) • CONSTANT DATA	UNIVERSAL HEAP
IDENT RECORD MODULE SUBRN.	SEGMENT N+5 (R,W) RUN-TIME STACK
SECTION DEFN: CODE SECTION DEFN: BINDING	

TASK STRUCTURE

* TASK ADDRESS SPACE IS DEFINED BY THE SEGMENT DESCRIPTOR TABLE UNIQUE TO THE TASK

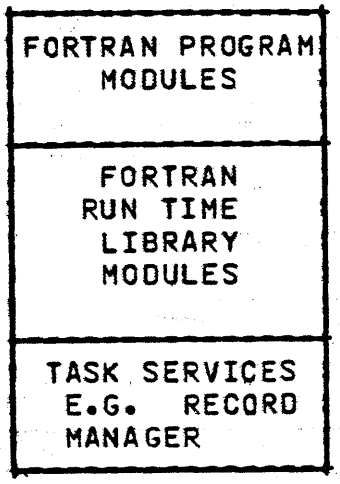
VL	XP	RP	WP	R1	R2	ASID	GL	KEY/LOCK
PROTECTION						LOCATION	PROTECTION	

* EACH SEGMENT HAS AN ENTRY IN THE SDT

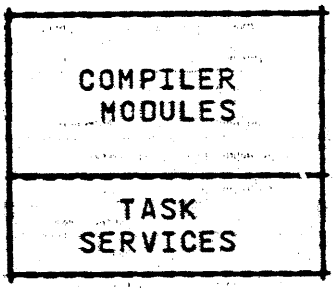
* THE SET OF SEGMENTS COMPRISING THE TASK MAY INCLUDE
  - USER SUPPLIED MODULES (CODE & DATA SEGMENTS)
  - DYNAMICALLY ALLOCATED TEMPORARY SEGMENTS (WORKING STORAGE, HEAP ETC.)
  - A BINDING SEGMENT
  - USER DEFINED LIBRARY SEGMENTS
  - PROTECTED RUN-TIME LIBRARY SEGMENTS
  - OPERATING SYSTEM LIBRARY SEGMENTS
  - TASK SERVICES CODE AND DATA SEGMENTS
  - TASK MONITOR CODE AND DATA SEGMENTS

EXAMPLES OF DIFFERENT TASKS

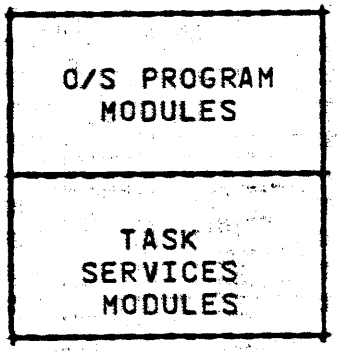
A USER TASK



A PRODUCT SET TASK



AN O/S TASK



JOB FLOW

BATCH JOB FLOW - OVERVIEW

* RECOGNIZES A VALID JOB SOURCE
* INVOKES INPUT SPOOLER

REMOTE HOST FACILITY

* VALIDATES THE JOB
* COPIES JOB DECK TO FILE
* INVOKES JOB SCHEDULER

JOB SCHEDULER

* EVALUATES CURRENT SYSTEM
* EVALUATE CURRENT JOB STATUS
* IF APPROPRIATE INVOKES JOB INITIATOR

JOB INITIATOR

* LINKS JOB INTO EXECUTION QUEUE
* INVOKES JOB MONITOR

JOB MONITOR

* INTERPRETS SCL EXECUTING EACH COMMAND
* AT "END OF JOB" LINKS JOB OUTPUT FILES INTO OUTPUT QUEUE
* INVOKES OUTPUT SPOOLER
* INVOKES JOB TERMINATOR

JOB TERMINATOR

* COMPLETES JOB ACCOUNTING
* REMOVES JOB FROM SYSTEM

* OUTPUT SPOOLER PRINTS JOB OUTPUT

INTERACTIVE JOB FLOW - OVERVIEW

* RECOGNIZES A VALID INTERACTIVE TERMINAL
* INVOKES JOB SCHEDULER

TERMINAL JOB STARTER

↓

* EVALUATES CURRENT SYSTEM
* NOTIFIES THE USER IF SYSTEM BUSY
* INVOKES JOB INITIATOR

JOB SCHEDULER

↓

* LINKS JOB INTC EXECUTION QUEUE
* INVOKES JOB MONITOR
* CONNECTS TERMINAL TO JOB MONITOR

JOB INITIATOR

↓

* INTERPRETS TERMINAL INPUT EXECUTING EACH COMMAND
* AT "LCGOUT" INVOKES JOB TERMINATOR

JOB MONITOR

↓

* COMPLETES JOB ACCOUNTING
* REMOVES JOB FROM SYSTEM

JOB TERMINATOR

SYSTEM ACCESS

* USER MUST BE CORRECTLY IDENTIFIED FOR ACCESS
  - VALID USER NAME
  - CORRECT PASSWORD
  - SATISFY ADDITIONAL INSTALLATION DEFINED SECURITY CHECKS

* USER IS VALIDATED FOR ACCESS AND RESOURCE USE WITHIN AN ADMINISTRATIVE HIERARCHY
  - FAMILY
  - ACCOUNT
  - PROJECT
  - MEMBER

* USER DEFAULTS AND LIMITS DEFINED BY HIS STATUS IN THE FAMILY/ACCOUNT/PROJECT HIERARCHY AND WITHIN THE CURRENT SYSTEM STATE

* EXAMPLES OF POSSIBLE SYSTEM STATES ARE
  - MAINTENANCE MODE, EXTERNAL CUSTOMERS ONLY, NO OPERATOR, PRIME TIME, TOP SECRET

* EXAMPLES OF DEFAULTS AND LIMITS ARE:
  - FORCED PROLOG AND/OR EPILOG EXECUTION
  - LIMITS ON ACCESS PATHS INTO THE SYSTEM
  - MAXIMUM NUMBER OF MESSAGES, SCL STATEMENTS, TAPES
  - MAXIMUM MASS STORAGE OR CENTRAL MEMORY PER JOB
  - MAXIMUM NUMBER OF SRUS ALLOWED

JOB MANAGEMENT - OVERVIEW

JOB MANAGEMENT TASKS

- JOB INITIATOR
- JOB SCHEDULER
- JOB MONITOR
- JOB TERMINATOR

JOB MANAGEMENT DATA STRUCTURES

KNOWN JOB LIST
JOB NAME JOB SEQUENCE NUMBER JOB STATUS SCHEDULING PRIORITY JOB CLASS LINK INTO KJL THREAD LINK TO JCB

JOB CONTROL BLOCK
JOB IDENTIFICATION USER/ACCOUNT ID. JOB PRIORITY RESOURCE LIMITS RESOURCE ACCUMULATORS LINK TO KJL ENTRY

JOB SCHEDULER

* JOB SCHEDULER EXECUTES AS A SYSTEM TASK

* JOB SCHEDULER DETERMINES:
  - ORDER IN WHICH JOBS IN THE INPUT QUEUE SHOULD BE INITIATED
  - WHEN A JOB SHOULD BE SWAPPED INTO OR OUT OF MEMORY

* SOME EXAMPLES OF SCHEDULING CRITERIA ARE:
  - CURRENT PRIORITY WITHIN JOB CLASS
  - JOB RESOURCE REQUIREMENTS
  - JOB CLASS AND STATUS
  - CURRENT SYSTEM RESOURCE AVAILABILITY

* JOB SCHEDULER MONITORS THE AVAILABLE MIX OF QUEUED AND INITIATED JOBS AND PRIORITIZES THEM BASED ON CURRENT SYSTEM USAGE

* THE KJL ENTRY FOR A JOB IS LINKED INTO A THREAD WHICH REPRESENTS ONE OF THE FOLLOWING STATES
  - "DEFERRED" WAITING FOR A TIME INTERVAL TO ELAPSE
  - "QUEUED" WAITING TO BE INITIATED
  - "INITIATED" ACTIVE, INACTIVE OR SWAPPED OUT BUT AVAILABLE FOR EXECUTION
  - "TERMINATED" COMPLETED BUT OUTPUT FILES QUEUED FOR DISPOSITION

* JOB SCHEDULER IS INVOKED:

  - PERIODICALLY BY CPU MONITOR

* AT EVERY CYCLE JOB SCHEDULER UPDATES THE TABLES WHICH DESCRIBE:

  - STATUS OF THE SYSTEM

    . NO. OF JOBS IN MEMORY
    . NO. OF JOBS SWAPPED OUT
    . RESOURCES AVAILABLE

  - STATUS OF EACH JOB

    . SCHEDULING PRIORITY
    . SWAP OUT STATUS
    . RESOURCES ASSIGNED AND REQUIRED
    . TOTAL WORKING SET

* AT EVERY CYCLE JOB SCHEDULER INCREMENTS (AGES) THE PRIORITY OF "QUEUED" JOBS

  - PRIORITY MAY AGE TO MAXIMUM FOR THE JOB CLASS

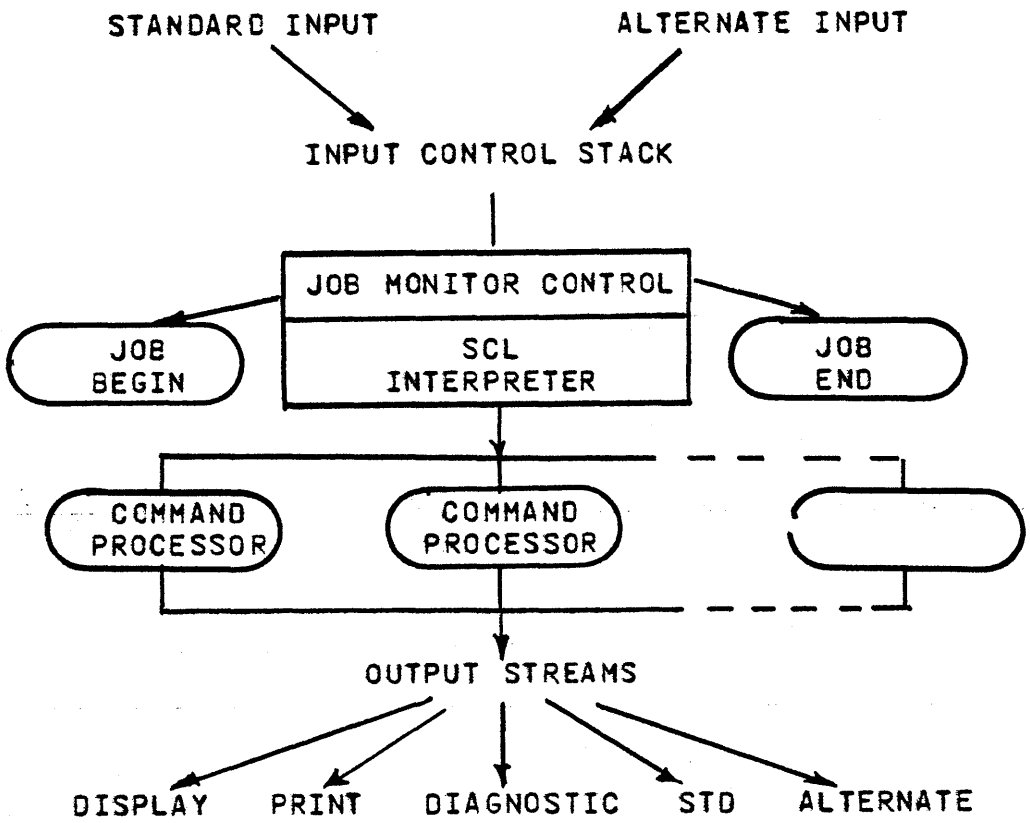
  - HIGHEST PRIORITY JOB IS SELECTED FOR INITIATION WHEN
    . NO. OF ACTIVE JOBS IN THE SAME CLASS IS BELOW THE INSTALLATION THRESHOLD
    . RESOURCE REQUIREMENTS CAN BE SATISFIED

JOB INITIATOR

* JOB INITIATOR EXECUTES AS A SYSTEM TASK

* JOB INITIATOR BUILDS THE ENVIRONMENT TO BEGIN JOB EXECUTION
  - OBTAINS HIGHEST PRIORITY JOB IDENTIFIED FOR INITIATION BY JOB SCHEDULER
  - ACQUIRES A JOB SEGMENT AND INITIALIZES IT WITH THE JOB TEMPLATE
    - TASK SERVICES CODE SEGMENTS
    - TASK MONITOR CODE SEGMENTS
    - INITIAL JOB CONTROL BLOCK
    - INITIAL EXECUTION CONTROL BLOCK
    - PROGRAM DESCRIPTION
  - ACQUIRES THE STANDARD INPUT FILE FOR THE JOB FROM SYSTEM INPUT QUEUE MANAGER
  - RELINKS THE KJL ENTRY INTO THE "INITIATED" QUEUE
  - INVOKES THE JOB MONITOR TASK

JOB MONITOR - OVERVIEW



* JOB MONITOR EXECUTES AS TASK OF THE USER JOB

* JOB MONITOR IS THE STARTING POINT FOR EXECUTION WITHIN A NEWLY INITIATED JOB

  - JOB MONITOR CONTROL IS THE INITIAL ENTRY POINT AND CONTROLS THE SEQUENCE OF EXECUTION

  - JOB BEGIN COMPLETES VALIDATION, COMPLETES THE JCB AND FORCES EXECUTION OF THE USER PROLOG

  - SCL INTERPRETER EXECUTES THE JOB COMMAND STREAM INVCKING COMMAND PROCESSOR TASKS IF REQUIRED

  - JOB END CLOSES AND RELEASES FILES AND EXECUTES THE USER EPILOG

JOB END

* JOB END EXECUTES AS A SYSTEM TASK

* JOB END RETURNS RESOURCES AND CLEANS UP AFTER JOB COMPLETION

  - FORMATS AN ACCOUNTING RECORD FROM THE DATA ACCUMULATED IN THE JOB CONTROL BLOCK AND LOGS IT IN THE ACCOUNTING FILE

  - RELEASES ALL LOCAL JOB FILES EXCEPT THE DAYFILE AND STANDARD OUTPUT FILES

    . "DISPLAY" OUTPUT GENERATED BY DISPLAY COMMANDS
    . "PRINT" OUTPUT GENERATED BY PRINT COMMANDS
    . "DIAGNOSTIC" ERROR CONDITIONS
    . "STANDARD" ECHO OF STANDARD INPUT FILE
    . "ALTERNATE" ECHO OF ALTERNATE INPUT FILE

  - REQUESTS QUEUED FILE MANAGER TO LINK OUTPUT FILES INTO THE KNOWN OUTPUT LIST. OUTPUT TYPES ARE:

    . "LINK" REMOTE STATION VIA NETWORK
    . "BATCH" LOCAL UNIT RECORD EQUIPMENT

  - RELEASES RESOURCES ASSIGNED TO THE JOB

    . PERIPHERAL EQUIPMENT
    . JOB ADDRESS SPACE (SEGMENTS ALLOCATED TO THE JOB)
    . JOB CONTROL BLOCK

* JOB TERMINATOR

  - RELINKS THE KJL ENTRY INTO "TERMINATED" THREAD

  - RELEASES JOB SEGMENT

  - INVOKES JOB SCHEDULER

SYSTEM ACCOUNTING

* NOS/VE LOGS SYSTEM USE BY EACH JOB FOR BILLING

* THE SET OF CHARGEABLE ACTIVITIES AND THE FUNCTION TO COMPUTE A BILLING UNIT IS DEFINED BY THE INSTALLATION

* EXAMPLES OF CHARGEABLE PARAMETERS ARE:

  - CPU TIME, REAL MEMORY AND MASS STORAGE USED

* THE USER NAME IS VALIDATED FOR CHARGING TO ONE OR MORE ACCOUNTS/PROJECTS. THE JOB MAY SELECT THE ACCOUNT/PROJECT TO BE CHARGED

* AS EACH TASK COMPRISING THE JOB EXECUTES RESOURCE USED DATA ACCUMULATES IN THE JOB ACCOUNTING RECORD

* AT JOB TERMINATION THE ACCUMULATED CHARGES ARE FORMATTED AND RECORDED IN THE SYSTEM ACCOUNTING LOG FOR INPUT TO THE BILLING PROGRAM

PROGRAM MANAGEMENT

PROGRAM MANAGEMENT FUNCTIONS

* PROVIDES COMMAND LEVEL SERVICES
  - GENERATE OBJECT LIBRARIES
  - SET LOADER DEFAULTS
  - EXECUTE A PROGRAM

* PROVIDES PROGRAM LEVEL SERVICES
  - EXECUTE A PROGRAM
  - COMMUNICATE WITH OTHER EXECUTING PRCGRAMS
  - ESTABLISH EXCEPTION CONDITION HANDLERS
  - SERVICE GENERAL REQUESTS (E.G. TIME, DATE)
  - LOAD OBJECT MODULES

* RECOGNIZES & HANDLES EXCEPTION CONDITIONS WHICH OCCUR DURING PROGRAM EXECUTION. (E.G. ARITHMETIC ERRORS, TASK TERMINATION REQUEST)

PROGRAM COMMUNICATION

* COMMUNICATION BETWEEN TASKS WITHIN A JOB
  "JOB LOCAL QUEUES"

  - LOCAL QUEUES HAVE NAMES DEFINED WITHIN THE JOB

  - ACCESS TO A QUEUE IS ONLY LIMITED BY RING LEVEL

  - MESSAGE MAY BE A SEGMENT OFFSET/BYTE STRING OR ENTIRE SEGMENT DESCRIPTOR(S)

* COMMUNICATION BETWEEN JOBS (ACCESS CONTROL MUST BE ENFORCED)

  - SIGNALS

  - SHARING A FILE SEGMENT

* A SIGNAL IS A SHORT MESSAGE DIRECTED TO A SPECIFIC TASK VIA A MONITOR REQUEST

* A LARGE AMOUNT OF DATA IS PASSED BY SHARING SEGMENT LEVEL ACCESS TO THE SAME PERMANENT FILE

  - FILE SYSTEM HANDLES ACCESS CONTROL

  - SIGNALS ARE USED TO PASS CONTROL INFORMATION

PROGRAM MANAGEMENT - COMPONENTS

* PROCESSORS FOR COMMANDS:
  - LIBRARY
  - SET_PROGRAM_OPTIONS
  - EXECUTE
  - "NAME CALL"

* CREATE_OBJECT_LIBRARY UTILITY

* PROGRAM EXECUTION COMPONENTS OF TASK SERVICES
  - TASK INITIATOR
  - LOADER
  - TASK TERMINATOR
  - LOCAL QUEUE PROCESSOR
  - CONDITION PROCESSOR
  - PROGRAM DEBUGGER

PROGRAM EXECUTION ENVIRONMENT

TASK SERVICES

- TASK INITIATOR
- TASK TERMINATOR
- OBJECT PROGRAM LOADER
- LOCAL QUEUE PROCESSOR
- TRAP HANDLER FOR USER CONDITIONS
- PROGRAM DEBUGGER
- EXCEPTION CONDITION PROCESSOR

USER OBJECT MODULES

- USER EXCEPTION CONDITION PROCESSOR

SYSTEM MONITOR

- TASK INITIATION
- TASK TERMINATION

PROGRAM EXECUTION - FLOW

* PROGRAM OR COMMAND REQUESTS PROGRAM EXECUTION
* CALLS TASK INITIATOR

* BUILDS TABLES FOR THE NEW TASK

* EXCHANGES TO SYSTEM MONITOR TO REQUEST TASK INITIATION

* LINKS NEW TASK INTO CPU DISPATCH LIST

* CPU IS DISPATCHED TO NEW TASK

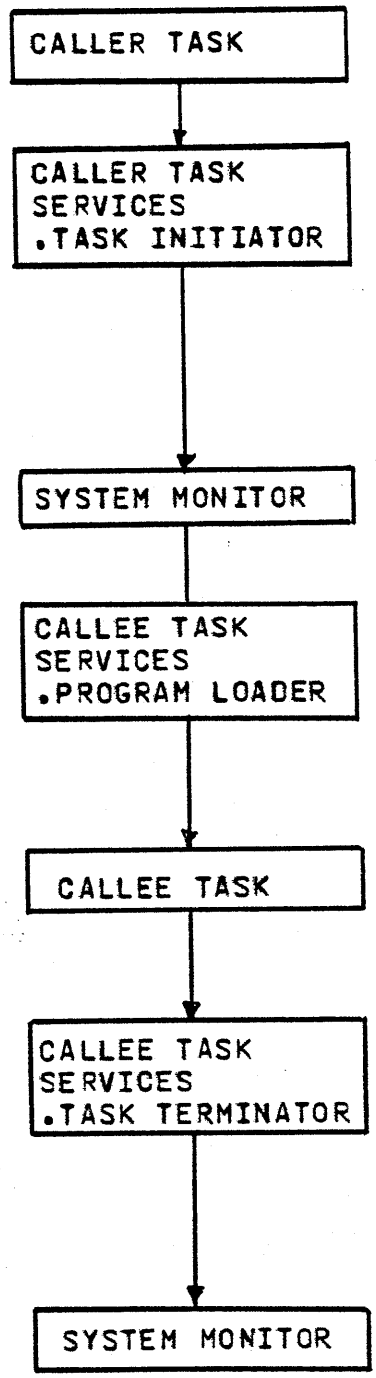
* LOADER LOADS OBJECT MODULES
* LOADER PASSES CONTROL TO INITIAL ENTRY POINT

* NEW TASK EXECUTES ASYNCHRONOUSLY TO CALLER TASK

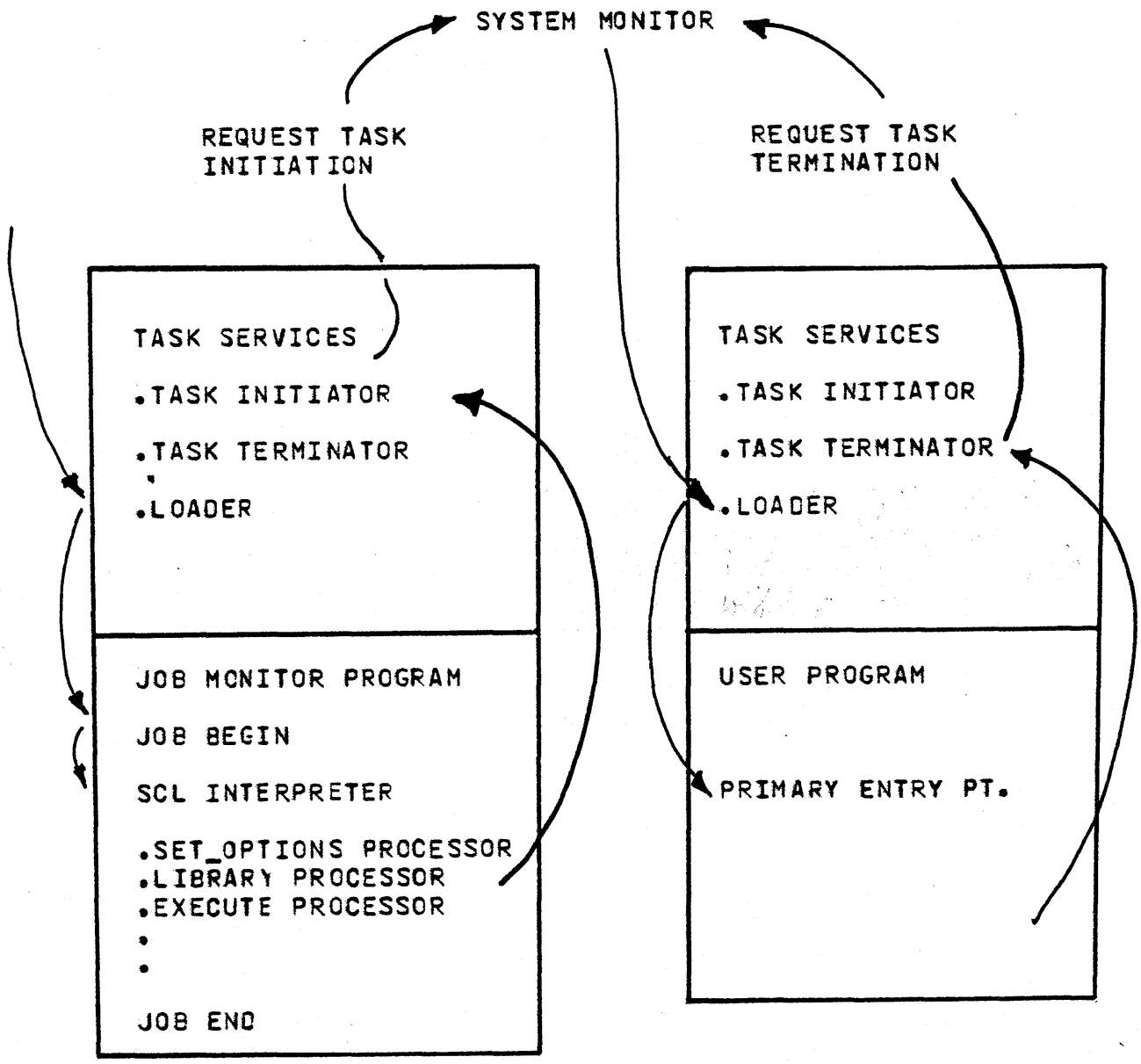
* NEW TASK CALLS EXIT INTERFACE

* CLEANS UP TASK
* EXCHANGES TO SYSTEM MONITOR TO REQUEST TASK TERMINATION

* REMOVE TASK ENTRIES FROM DISPATCH LIST
* INFORMS CALLER THAT CALLEE HAS TERMINATED



PROGRAM EXECUTION - EXAMPLE



CALLER TASK
(E.G. JOB MONITOR)

CALLEE TASK
(E.G. USER PROGRAM)

CPU MANAGEMENT

CPU MONITOR - OVERVIEW

* CPU MONITOR IS THE PART OF NOS/VE MOST DIRECTLY RELATED TO THE CY180 HARDWARE ENVIRONMENT
  - HANDLES ALL EXCHANGE INTERRUPTS
  - HANDLES ALL TRAPS OCCURRING IN MONITOR PROCESS STATE
  - PROCESSES PAGE FAULTS
  - DISPATCHES THE CPU
  - QUEUES I/O REQUESTS
  - MANAGES INTER-TASK COMMUNICATION
  - CHANGES TASK STATUS
  - EXCHANGES CPU BETWEEN DUAL-STATES

* CPU MONITOR OPERATIONAL CHARACTERISTICS
  - MOST PRIVILEGED PIECE OF CODE
  - ENTERED VIA AN EXCHANGE TO MONITOR PROCESS STATE
  - CPU MONITOR CODE & DATA ARE NON-PAGEABLE, "WIRED DOWN" IN REAL MEMORY

* WHEN EXECUTING CPU MONITOR
  - INTERRUPTS MAY BE DISABLED
  - TRAPS ARE ENABLED FOR EXCHANGE REQUEST, EXTERNAL INTERRUPT, SYSTEM INTERVAL TIMER, AND POWER FAIL
  - ALL OTHER CONDITIONS CAUSE CPU HALT

INTERRUPT AND TRAP HARDWARE REVIEW

* VARIOUS CONDITIONS DETECTED BY CY180 HARDWARE REQUIRE THE CURRENT PROCESS TO BE INTERRUPTED WHILE THE CONDITION IS RESOLVED
  - HARDWARE FAULTS (E.G. POWER WARNING, PROCESSOR DETECTED MALFUNCTION)
  - HARDWARE DETECTED SOFTWARE FAULTS (E.G. ACCESS VIOLATION, ARITHMETIC ERRORS)
  - SYSTEM CONDITIONS (E.G. PAGE FAULT, SYSTEM INTERVAL TIMER, EXCHANGE REQUEST, KEYPOINT)

* THE CURRENT STATE OF THE CPU DETERMINES HOW A CONDITION IS HANDLED
  - EXCHANGE TO MONITOR PROCESS STATE, CPU MONITOR HANDLES THE CONDITION
  - CALL THE TRAP SERVICE ROUTINE, HANDLES TRAP IN JOB PROCESS STATE (CPU MONITOR HANDLES TRAPS IN MONITOR PROCESS STATE)
  - STACK CONDITION AND CONTINUE
  - CPU HALT

* THE ACTION TAKEN DEPENDS ON
  - MONITOR OR JOB PROCESS STATE
  - INTERRUPT CONDITIONS MASKED
  - INTERRUPTS ENABLED/DISABLED
  - TRAPS ENABLED/DISABLED
  - MONITOR OR USER CONDITION

MONITOR CONDITION REGISTER

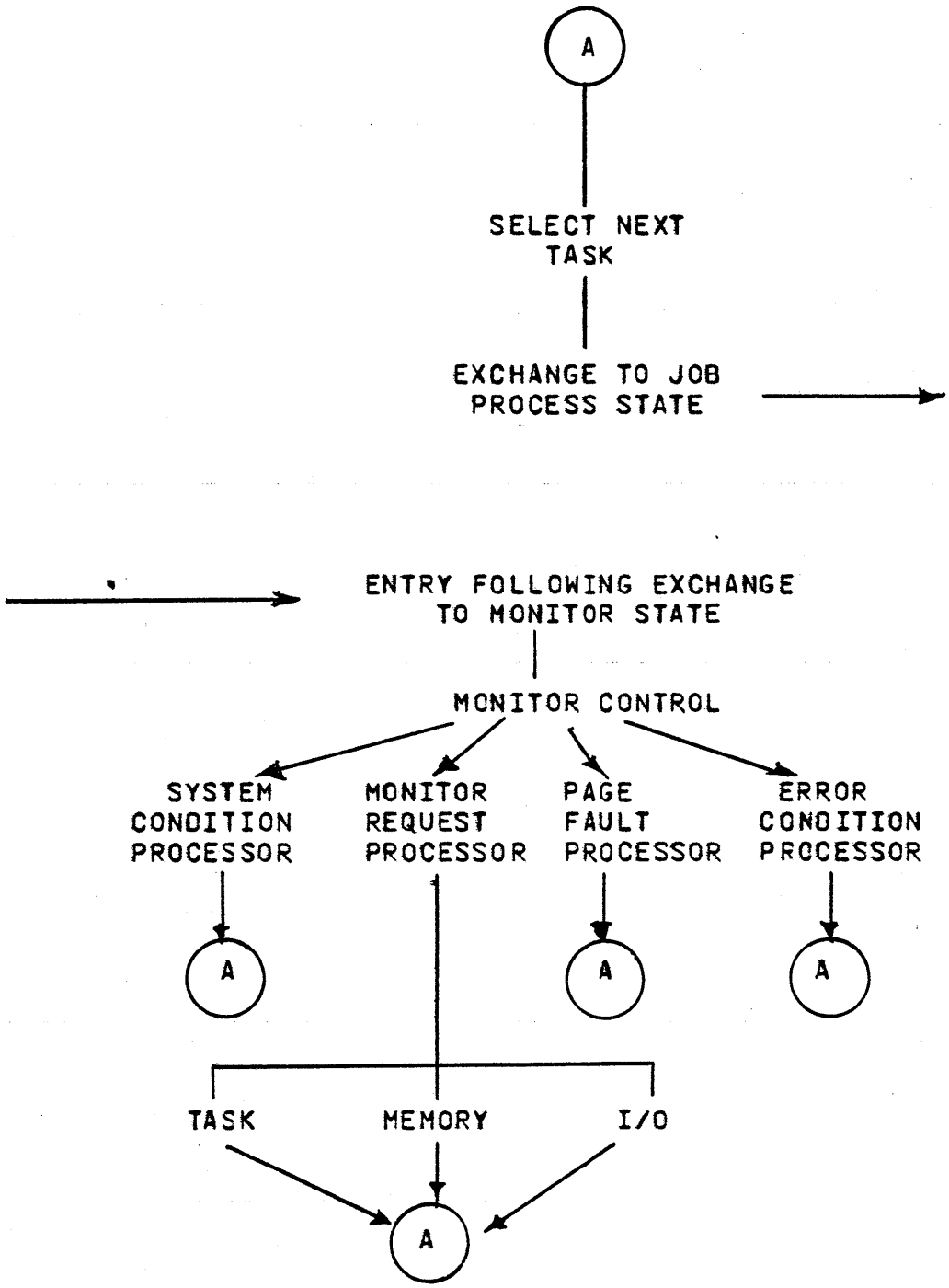
BIT NUMBER AND DEFINITION			TRAPS ENABLED		TRAPS DISABLED		
			TRAP ENABLE F/F SET AND TRAP ENABLE DELAY F/F CLEAR AND MASK BIT SET		TRAP ENABLE F/F CLEAR OR TRAP ENABLE DELAY F/F SET AND MASK BIT SET		MASK BIT CLEAR
			JOB MODE	MONITOR MODE	JOB MODE	MONITOR MODE	JOB OR MONITOR MODE
0	Processor Detected Malfunction	Mon	EXCH	TRAP	EXCH	HALT	HALT
1	Memory Detected Malfunction	Mon	EXCH	TRAP	EXCH	HALT	HALT
2	Power Warning	Sys	EXCH	TRAP	EXCH	STACK	STACK
3	Instruction Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
4	Address Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
5	Exchange Request	Sys	EXCH	TRAP	EXCH	STACK	STACK
6	Access Violation	Mon	EXCH	TRAP	EXCH	HALT	HALT
7	Environment Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
8	External Interrupt	Sys	EXCH	TRAP	EXCH	STACK	STACK
9	Page Table Search Without Find	Mon	EXCH	TRAP	EXCH	HALT	HALT
10	System Call	Status - This bit is a flag only and does not cause any hardware action.					
11	System Interval Timer	Sys	EXCH	TRAP	EXCH	STACK	STACK
12	Invalid Segment	Mon	EXCH	TRAP	EXCH	HALT	HALT
13	Outward Call/Inward Return	Mon	EXCH	TRAP	EXCH	HALT	HALT
14	Soft Error Log	Sys	EXCH	TRAP	EXCH	STACK	STACK
15	Trap Exception	Status - This bit is a flag only and does not cause any hardware action.					

combined in bit 0 bit 1 access

CPU MONITOR - COMPONENTS

* CPU MONITOR IS IMPLEMENTED BY
  - MAIN CONTROL LOOP/CPU DISPATCH
  - SYSTEM CONDITIONS PROCESSORS
  - MONITOR REQUEST PROCESSORS
  - PAGE FAULT HANDLER
  - ERROR CONDITION PROCESSORS

CPU MONITOR - MAIN CONTROL LOOP



PAGE FAULT HANDLER

* ALLOCATES REAL MEMORY PAGE FRAMES FOLLOWING
  - ASSIGN PAGE FRAME REQUEST
  - SEARCH-WITHOUT-FIND INTERRUPT

* PAGE MANAGER TRIES TO MAINTAIN A POOL OF FREE PAGE FRAMES TO SATISFY PAGE FAULTS
  - MONITORS USE OF ASSIGNED PAGE FRAMES
  - REASSIGNS PAGE FRAMES TO THE FREE POOL BASED ON FREQUENCY OF REFERENCE

PAGE FRAME TABLE (ONE ENTRY FOR EACH PAGE FRAME)

- PAGE STATUS
- USAGE STATISTICS
- LINK INTO A QUEUE

* EACH PAGE FRAME IS LINKED INTO A QUEUE
  - FREE (EMPTY PAGE FRAME)
  - AVAILABLE (PREVIOUSLY USED)
  - AVAILABLE MODIFIED (PREVIOUSLY MODIFIED)
  - SHARED (CONTAINS PAGE OF A SHARED SEGMENT)
  - JOB WORKING SET (ACTIVE PAGE OF A JOB)
  - WIRED (PAGE LOCKED DOWN IN THE FRAME)

CPU MONITOR - MONITOR REQUEST PROCESSORS

* PROCESSES REQUESTS FROM THE TASKS FOR BASIC SERVICES
  - MEMORY MANAGEMENT
  - TASK MANAGEMENT
  - SIGNALS
  - DEVICE ACCESS

* THESE FUNCTIONS ARE IN THE CPU MONITOR ENVIRONMENT FOR
  - SERIALIZATION
  - ACCESS TO SYSTEM TABLES
  - PERFORMANCE
  - ISOLATION

* MONITOR FUNCTIONS ARE REQUESTED BY SYSTEM CALL PASSING A REQUEST BLOCK CONTAINING THE REQUEST PARAMETERS. MONITOR CONTROL ROUTES THE REQUEST TO THE APPROPRIATE PROCESSOR
  - TASK REQUEST PROCESSOR
  - MEMORY REQUEST PROCESSOR
  - I/O REQUEST PROCESSOR

MONITOR REQUEST PROCESSORS

* TASK REQUEST EXAMPLES

  - INITIATE-TASK: ALLOCATE A PTL ENTRY, LINK IT TO THE TASK EXECUTION CONTROL BLOCK AND TO THE READY QUEUE

  - TERMINATE-TASK: REMOVE TASK FROM THE READY QUEUE DELINK AND FREE THE PTL ENTRY

  - DELAY: DELAY EXECUTION OF A TASK FOR THE SPECIFIED TIME PERIOD

  - WAIT: DELAY EXECUTION OF A TASK WAITING FOR A SIGNAL

  - CYCLE: SWITCH EXECUTION TO THE NEXT READY TASK

  - SEND-SIGNAL: SEND A SIGNAL TO A SPECIFIED TASK OR PROCESSOR

  - GET-SIGNAL: GET ANY SIGNALS QUEUED WHILE THE TASK WAS SWAPPED OUT OR ITS BUFFER WAS FULL

* MEMORY REQUEST EXAMPLES

  - ASSIGN-PAGE: ASSIGN PAGE FRAMES TO A SEGMENT

  - RELEASE-PAGE: RELEASE PAGE FRAMES FROM A SEGMENT

  - LOCK-PAGE: LOCK PAGE FRAMES TO PREVENT PAGING

  - UNLOCK-PAGE: UNLOCK SPECIFIED PAGE FRAMES

  - SWAP-OUT: WRITE OUT ALL NON-SHARED PAGES COMPRISING THE JOB WORKING SET

  - SWAP-IN: RE-ESTABLISH JOB WORKING SET BY ASSIGNING PAGE FRAMES AND READING PAGES FROM THE SWAP AREA

* I/O REQUESTS ARE PASSED TO THE IOU BASED DEVICE DRIVERS

CPU MONITOR - SYSTEM CONDITIONS PROCESSORS

HANDLES EXCEPTION CONDITIONS WHICH HAVE INTERRUPTED THE CURRENTLY EXECUTING TASK BUT DO NOT PRECLUDE FURTHER PROCESSING

* EXCHANGE INTERRUPT - A 170 PPU HAS ISSUED A MEJ INSTRUCTION TO EXCHANGE CONTROL TO THE 170 MODE CPU MONITOR

* POWER WARNING - POWER TO THE MAINFRAME IS FAILING CPU MONITOR ATTEMPTS TO SELECTIVELY DUMP REAL MEMORY FOR RECOVERY

* EXTERNAL INTERRUPT - IOU HAS SIGNALLED I/O COMPLETION OR ANOTHER PROCESSOR WISHES TO COMMUNICATE

* SYSTEM INTERVAL TIMER INTERRUPT - SYSTEM REAL TIME CLOCK. USED FOR

  - TASK EXECUTION QUANTA

  - DELAY TIME FOR TASKS

  - TIME TO UPDATE TASK SCHEDULER STATUS

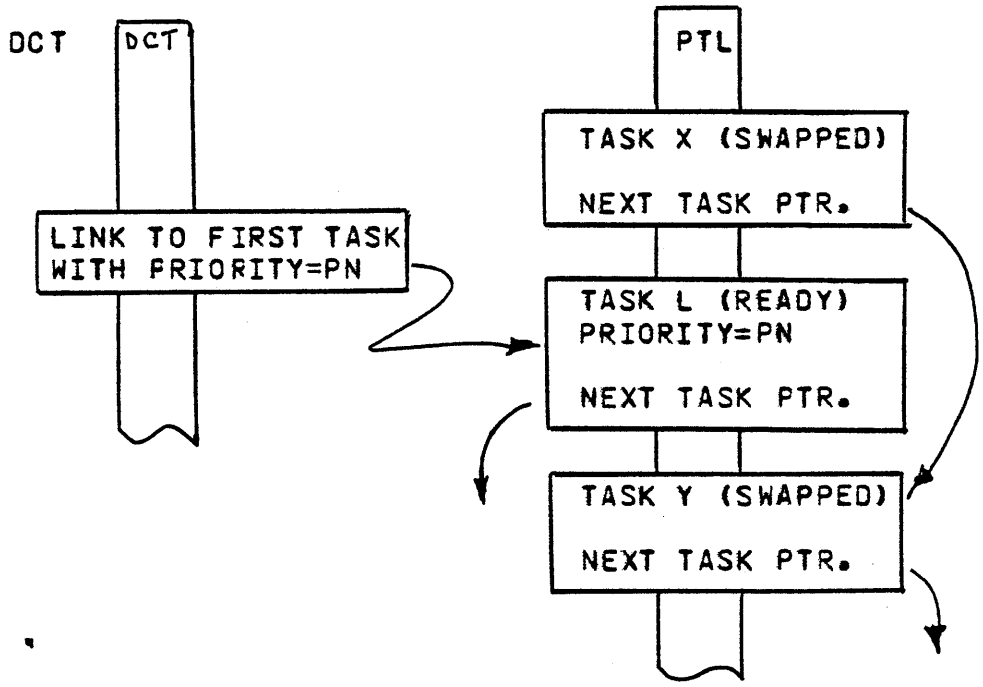
  - UPDATE USAGE STATISTICS

CPU MONITOR - ERROR CONDITION PROCESSORS

* HANDLES ERROR CONDITIONS WHICH HAVE INTERRUPTED THE CURRENTLY EXECUTING TASK AND MAY PRECLUDE FURTHER PROCESSING
  - PROCESSOR OR MEMORY DETECTED MALFUNCTION
  - INSTRUCTION SPECIFICATION ERROR
  - ADDRESS SPECIFICATION ERROR
  - ACCESS VIOLATION
  - ENVIRONMENT SPECIFICATION ERROR
  - INVALID SEGMENT
  - OUTWARD CALL/INWARD RETURN
  - TRAP EXCEPTION
  - ALL TRAPS IN MONITOR MODE

* ERROR CONDITION PROCESSOR RECEIVES CONTROL WHEN THE EXCHANGE WAS A RESULT OF ANY OF THE CONDITIONS
  - PROCESSES THE CONDITION
  - RETURNS CONTROL TO CPU MONITOR TO DISPATCH THE NEXT TASK

CPU MONITOR - CPU DISPATCHING



PRIMARY TASK LIST
(1 ENTRY PER TASK)

- TASK SEQUENCE NUMBER
- TASK STATUS
- PRIORITY
  - JOB IDENTIFIER
  - LINK TO EXECUTION CONTROL
- LINK TO NEXT TASK IN QUEUE

DISPATCH CONTROL TABLE
(1 ENTRY FOR EACH CPU
PRIORITY LEVEL)

- LINK TO FIRST TASK IN T
  READY QUEUE WITH THAT
  PRIORITY

* CPU IS DISPATCHED TO HIGHEST PRIORITY TASK IN THE "READY" QUEUE

RESOURCE MANAGEMENT

RESOURCE MANAGEMENT

* RESOURCES WHICH ARE MANAGED BY NOS/VE INCLUDE
  - PREEMPTIVE RESOURCES WHICH CAN BE RELINQUISHED TO OTHER JOBS (E.G. REAL MEMORY, CPUS, I/O CHANNELS, MASS STORAGE)
  - NON-PREEMPTIVE RESOURCES (E.G. MAGNETIC TAPE DRIVES, VIRTUAL MEMORY)

* DEVICE MANAGEMENT FUNCTIONS INCLUDE
  - CONTROLLING ACCESS TO PHYSICAL DEVICES
  - ASSOCIATING A FILE WITH A DEVICE
  - MOUNTING/DISMOUNTING REMOVABLE MEDIA
  - LABELLING DISK AND TAPE DEVICES
  - SET MANAGEMENT
  - CONFIGURATION MANAGEMENT

* SEGMENT MANAGEMENT FUNCTIONS INCLUDE
  - SEGMENT MANAGEMENT SERVICES FOR A TASK
  - SEGMENT MANAGEMENT FOR NOS/VE JOB MANAGEMENT,
  - PROGRAM MANAGEMENT, I/O AND MAINTENANCE SERVICES
  - ASSIGNING THE ACTIVE SEGMENT IDENTIFIER (ASID) TO A SEGMENT

DEVICE MANAGEMENT - OVERVIEW

* RECONCILES USER DEMANDS FOR SYSTEM RESOURCES WITH THEIR AVAILABILITY
  - INTERFACE TO GIVE ADVANCE REQUIREMENTS FOR NON-PREEMPTIVE RESOURCES ("RESOURCE" COMMAND/REQUEST)
  - INTERFACE TO ASSOCIATE A FILE WITH A PARTICULAR DEVICE PRIOR TO THE FIRST ACCESS ("REQUEST" COMMAND/REQUEST)
  - INTERFACE TO BREAK FILE & DEVICE ASSOCIATION ("RETURN","UNLOAD" COMMAND/REQUESTS)

* SCHEDULES DEVICES AMOUNG JOBS TO PREVENT DEADLOCK

* ENABLES SHARING OF DISK VOLUMES BETWEEN MAINFRAMES

* MAINTAINS AND MODIFIES THE CONFIGURATION TABLES BUILT AT DEADSTART TIME
  - DYNAMIC RE-CONFIGURATION
  - GROUPING RMS VOLUMES INTO LOGICAL STORAGE UNITS

RMS MANAGEMENT

* MASS STORAGE VOLUMES ARE GROUPED INTO LOGICAL SETS
  - SET NAME
  - VOLUME NAMES
  - MASTER VOLUME HAS SET DESCRIPTOR AND DIRECTORY TO MASTER CATALOGS RESIDING WITHIN THE SET
  - SETS MAY BE ON-LINE OR REMOVABLE (NOT RELEASE 1)

* EACH FAMILY IS ASSOCIATED WITH AT LEAST ONE SET. NOS/VE MANAGES FAMILY SETS. E.G.
  - SPACE ASSIGNMENT
  - MOUNTING/DISMOUNTING

* AUXILIARY SETS (NOT DEFINED TO A FAMILY) CAN BE ACCESSED BY VALIDATED USERS ("REQUEST" COMMAND/REQUEST)

CONFIGURATION MANAGEMENT

* SYSTEM CONFIGURATION IS DEFINED USING "CONFIGURATION DEFINITION LANGUAGE" (CDL)
  - FOR PROCESSING AT DEADSTART TIME
  - FOR PROCESSING DURING SYSTEM OPERATION FOLLOWING OPERATOR REQUEST FOR CDL PROCESSOR

* CONFIGURATION DISPLAYS PROVIDE DATA ON HARDWARE ELEMENTS IDENTIFIED BY "SYSTEM ELEMENT NUMBER" (SEN)
  - CONNECTION PATHS
  - ELEMENT STATE (ON/OFF/MAINTENANCE)
  - ERROR HISTORY
  - CURRENT MAINTENANCE ACTIVITY
  - MEDIA DATA (MOUNTED/SETNAME/VSN/ETC.)
  - JOB DATA (USER NAME/LAST ACCESS/ETC.)

* PROVIDES AN INTERFACE FOR MAINTENANCE SOFTWARE
  - SYSTEM MAINTENANCE INTERLOCK CHECK
  - PLACE AN ELEMENT IN MAINTENANCE STATE
  - "ASSIGN" AND "RETURN" AN ELEMENT TO/FROM A TEST/DIAGNOSTIC JOB
  - REINSTATE AN ELEMENT FROM MAINTENANCE STATE

SEGMENT MANAGEMENT - OVERVIEW

* SEGMENT DESCRIPTORS ARE THE ROOT OF NOS/VE SYSTEM SECURITY AND MUST BE PROTECTED FROM UNAUTHORIZED ACCESS. SEGMENT MANAGER THUS HAS TWO INTERFACES

* TRANSIENT SEGMENT MANAGER IS THE USER PROGRAM INTERFACE. AVAILABLE FUNCTIONS ARE
  - ALLOCATE A TRANSIENT SEGMENT
  - PROVIDE STATUS OF SEGMENT ATTRIBUTES
  - MODIFY SEGMENT ATTRIBUTES
  - DELETE A SEGMENT FROM TASKS ADDRESS SPACE

* SYSTEM SEGMENT MANAGER IS THE SYSTEM ROUTINE INTERFACE AND PROVIDES MORE PRIVILEGED ACCESS
  - ALLOCATE A TRANSIENT SEGMENT
  - ALLOCATE/DEALLOCATE FILE SEGMENTS (FOR BAM)
  - ALLOCATE BINDING SEGMENTS (FOR LOADER)
  - CAN MAKE MORE DETAILED CHANGES TO SEGMENT ATTRIBUTES
  - OBTAIN A COPY OF SEGMENT DESCRIPTOR TABLE ENTRY (FOR LOCAL QUEUE MANAGER)
  - ADD AN EXISTING SEGMENT DESCRIPTOR TO A TASKS ADDRESS SPACE TO PASS A SEGMENT (FOR LOCAL QUEUE MANAGER)

* BOTH TRANSIENT SEGMENT MANAGER AND SYSTEM SEGMENT MANAGER RESIDE IN TASK SERVICES, THEY SHARE BOTH TABLES AND CODE

SEGMENT MANAGER - DATA STRUCTURES

HARDWARE ORIENTED DATA STRUCTURES

VL	XP	RP	WP	R1	R2	ASID	GL	KEY/LOCK

SEGMENT DESCRIPTOR

* SEGMENT DESCRIPTOR TABLE

  - TABLE UNIQUE TO EACH NOS/VE TASK

  - TABLE LOCATION AND LENGTH ARE IN THE TASK EXCHANGE PACKAGE

  - EACH SEGMENT ALLOCATED TO THE TASK HAS AN ENTRY

  IN THE SDT

  - SDT ENTRIES ARE BUILT BY SEGMENT MANAGER WHEN THE SEGMENT IS ALLOCATED

  - SDT ATTRIBUTES MAY BE CHANGED DYNAMICALLY

RING	SEG. NUMBER	BYTE NUMBER

* PROCESS VIRTUAL ADDRESS

  - CREATED BY SEGMENT MANAGER WHEN A SEGMENT IS ALLOCATED

  - CONTAINS REQUESTOR'S RING NUMBER AND NEW SEGMENT NUMBER (BYTE NUMBER=0)

  - PVA IS PASSED BACK TO THE REQUESTOR AND USED FOR ALL SUBSEQUENT REFERENCES TO THE SEGMENT

EXAMPLE - ALLOCATE A TEMPORARY SEGMENT

* CALL TRANSIENT SEGMENT MANAGER OR SYSTEM SEGMENT MANAGER
  - FIND AN EMPTY SLOT IN THE SDT, THE INDEX BECOMES THE SEGMENT NUMBER
  - CONSTRUCT THE SEGMENT DESCRIPTOR
    - ACCESS PROTECTION (READ, WRITE, EXECUTE)
    - RING BRACKET R1 & R2 SET TO CALLERS RING NUMBER BY TSM (SSM ALLOWS READ, WRITE, EXECUTE AND CALL RING BRACKETS TO BE DEFINED)
    - SET GLOBAL KEY/LOCK TO ZERO (SSM ALLOWS KEY/LOCK TO BE SPECIFIED)
    - GENERATE A NEW ASID
  - ASSIGN SEGMENT TO A DEVICE SUITABLE FOR PAGING (SPACE IS NOT ALLOCATED UNTIL REQUIRED)
  - MAKE UP PVA AND RETURN TO CALLER

OBJECT CODE

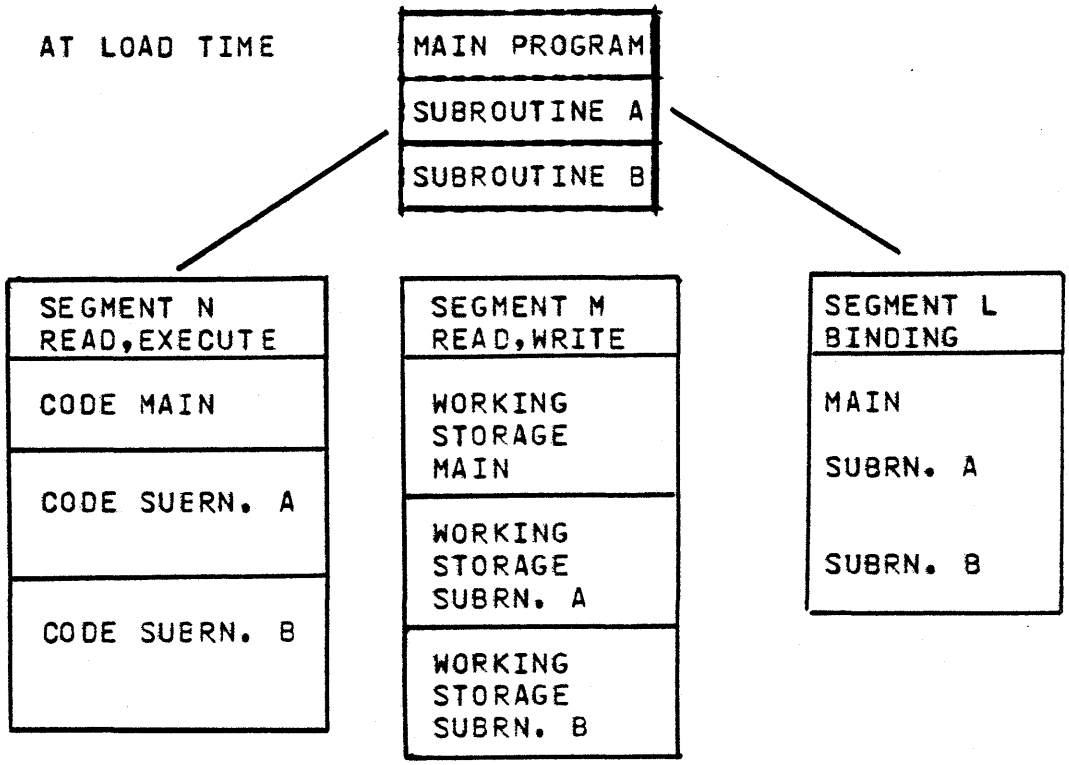
OBJECT CODE - INTRODUCTION

* PROGRAMS EXECUTING UNDER NOS/VE ARE ORGANIZED INTO SECTIONS HAVING COMMON CHARACTERISTICS

AT COMPILE TIME

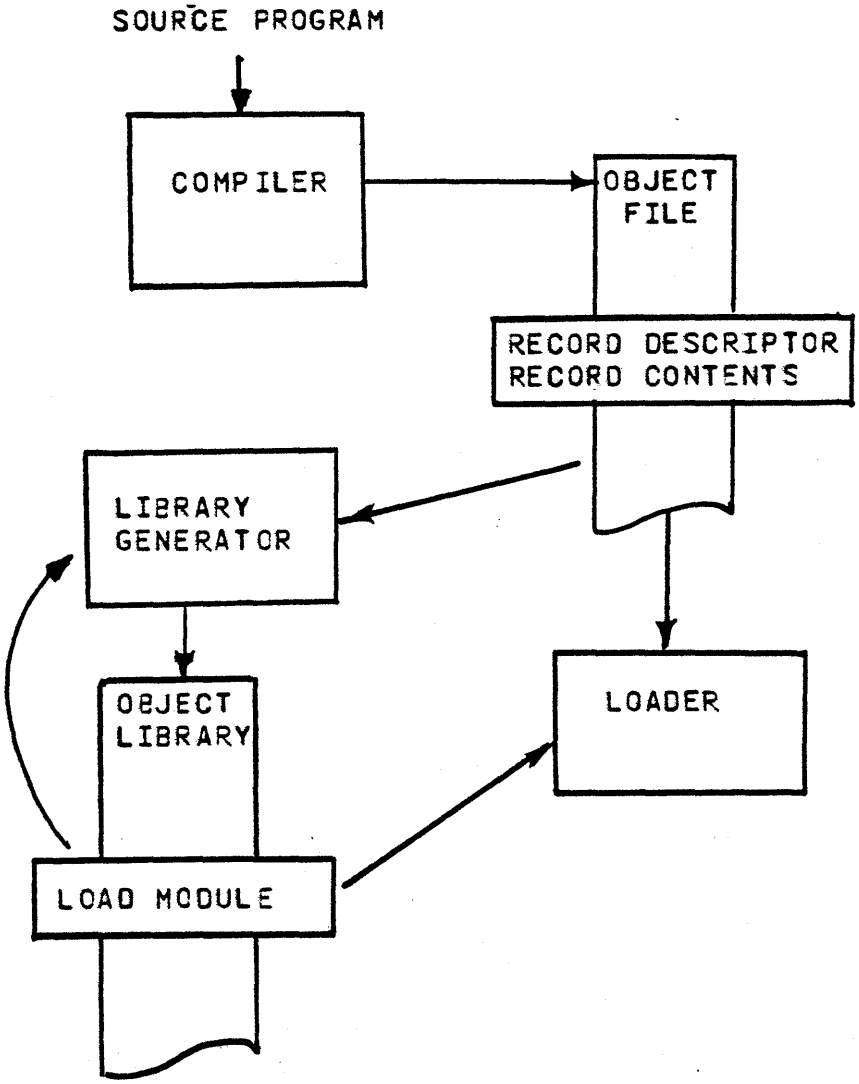
SECTION	PROTECTION	CHARACTERISTICS
CODE	READ EXECUTE	SHARABLE CODE AND CONSTANT DATA
WORKING STORAGE	READ WRITE	NON-SHARABLE, MODIFIABLE DATA
BINDING	READ BINDING	NON-SHARABLE POINTERS

AT LOAD TIME



OBJECT CODE - FORMATS

* OBJECT CODE FOR ANY PROGRAM MODULE MAY BE RETAINED AS EITHER
  - OBJECT MODULES
  - LOAD MODULES



OBJECT MODULE - INTERNAL FORMAT

* EACH OBJECT MODULE IS A SET OF RECORDS ON THE OBJECT FILE

* THE OBJECT RECORD DESCRIPTOR CONTAINS
  - ITEM TYPE
  - RECORD LENGTH

* ITEM TYPES

  IDR: IDENTIFICATION OF MODULE AND ATTRIBUTES

  LIB: LIBRARIES FROM WHICH TO SATISFY EXTERNAL REFERENCES

  SDC: LENGTH AND ATTRIBUTES OF EACH SECTION, CODE, WORKING STORAGE, BINDING AND ALL COMMON BLOCKS

  TEX: TEXT TO BE PLACED IN EACH SECTION

  RPL: TEXT TO BE REPETITIVELY PLACED IN EACH SECTION

  BIT: INSERTS BIT-LEVEL DATA INTO A SECTION

  EPT: DEFINES AN ADDRESS IN A SECTION AS AN ENTRY POINT

  RIF: IDENTIFIES ADDRESSES WHICH MUST BE RELOCATED BY THE LIBRARY GENERATOR WHEN BINDING MODULES TOGETHER

  ADR: ALLOWS PVAS TO BE BUILT AT LOAD TIME (WHEN RING,SEGMENT NUMBER AND OFFSET ARE KNOWN)

  XRL: LIST OF EXTERNAL REFERENCES TO BE SATISFIED

  BTI: A LOCATION IN THE BINDING SECTION

  TRA: TERMINATES THE OBJECT MODULE AND GIVES THE PRIMARY ENTRY POINT

EXAMPLE - OBJECT MODULE

LOCAL FILE LGO R1=11, R2=11, R3=11

USER COMMAND STREAM (VALIDATED FOR RING 11)

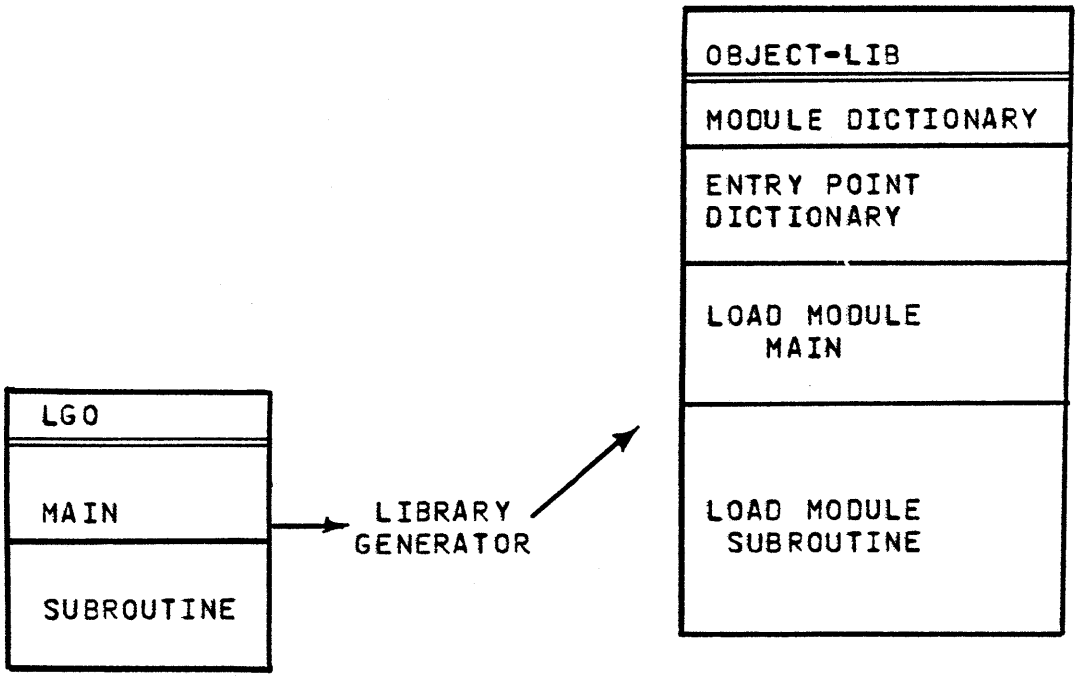
•
•
•
FTN,I=MAIN,B=LGO
FTN,I=SUB,B=LGO
LGO
•
•
•

IDR	• NAME • TIME & DATE CREATED • ETC,	OBJECT MODULE FOR MAIN
LIB	• FTNLIB	
SDC	CODE SECTION	
SDC	BINDING SECTION	
SDC	WORKING STORAGE SECTION	
SDC	COMMON BLOCKS	
	TEX, RPL, BIT, REL, ADR, XRL, EPT, BIN RECORDS FOR CODE, BINDING AND WORKING STORAGE SECTIONS	
TRA	• STARTING ADDRESS • END OF MODULE	
IDR		OBJECT MODULE FOR SUB
LIB	• FTNLIB	
SDC	• CODE	
SDC	• BINDING	
SDC	• WORKING STORAGE	
SDC	• COMMON BLOCKS	
	TEX, RPL, BIT, REL, ADR, XRL, EPT, BIN RECORDS FOR CODE BINDING AND WORKING STORAGE SECTIONS	
TRA		

LIBRARY GENERATION

* A LIBRARY IS AN ORGANIZED SET OF LOAD MODULES
  - MAY BE ACCESSED AS A LIBRARY SEGMENT WITHIN A TASK'S ADDRESS SPACE
  - MODULES ARE SHARABLE BETWEEN TASKS
  - MODULES ARE DIRECTLY EXECUTABLE (THE WORKING STORAGE AND BINDING SECTIONS ARE ALLOCATED AND INITIALIZED WHEN THE MODULE IS LOADED)

* A LIBRARY IS CREATED BY OBJECT LIBRARY GENERATOR




```
CREATE_OBJECT_LIBRARY
  ADD OBJECT_FILE = LGO
  GENERATE OBJECT_LIB
  END
DEFINE OBJECT_LIB
```

LOAD MODULE - INTERNAL FORMAT

MODULE HEADER
CODE ELEMENT .INSTRUCTIONS .CONSTANTS
LINKAGE ELEMENT .EXTERNAL NAMES AND LINKS
WORKING STORAGE ELEMENT .INITIALIZATION FOR ALL WORKING STORAGE AND COMMON BLOCKS
ENTRY POINT DEFINITIONS .NAMES OF ALL EXTERNALLY ACCESSIBLE ADDRESSES
INFORMATION ELEMENT .HEADER .COMPONENT IDENTIFICATION (NAMES OF ALL OBJECT MODULES COMPRISING THE LOAD MODULE) .RELOCATION INFORMATION (USED BY LIBRARY GENERATOR TO BIND OBJECT OR LOAD MODULES TOGETHER) .BINDING SECTION TEMPLATE (IDENTIFIES THE CONTENTS OF THE BINDING SECTION FOR THE LOAD MODULE)

NOS/VE FILE SYSTEM

NOS/VE FILE SYSTEM

* NOS/VE FILE SYSTEM SUPPORTS BOTH TEMPORARY AND PERMANENT FILES
  - TEMPORARY FILES ARE DISCARDED AT JOB OR TASK TERMINATION
  - PERMANENT FILES ARE REGISTERED IN A NOS/VE CATALOG AND ARE SAVED FOR SUBSEQUENT ACCESS

* NOS/VE FILES MAY BE ACCESSED VIA AN ACCESS METHOD

BASIC ACCESS METHOD

RECORD MANAGER FILE MANAGER	USER INTERFACE
BUFFER MANAGER FILE SERVICES	SERVICES

* BASIC ACCESS METHOD PROVIDES A DEVICE INDEPENDENT INTERFACE TO THE I/O SUB-SYSTEM

* BASIC ACCESS METHOD TRANSLATES THE USERS LOGICAL I/O REQUESTS INTO REQUESTS TO PHYSICAL I/C MANAGER
  - MAINTAINS THE FILE DESCRIPTION
  - OPENS AND CLOSES FILES
  - PROVIDES ACCESS TO THE RECORD STRUCTURE OF A FILE VIA GET/PUT
  - PROVIDES UNSTRUCTURED ACCESS TO A FILE VIA PHYSICAL & SEGMENT ACCESS
  - MANAGES BUFFERING FOR RECORD ORIENTED FILE ACCESS
  - PROCESSES FILE LABELS, FILE ERRORS ETC.

RECORD ACCESS

* RECORD ACCESS
  - LOGICAL RECORDS ARE TRANSFERRED BETWEEN USER STORAGE AND SYSTEM BUFFERS BY BUFFER MANAGER
  - DATA IS TRANSFERRED BETWEEN SYSTEM BUFFERS AND THE DEVICE BY PPU BASED DEVICE DRIVER

* BOTH TEMPORARY AND PERMANENT FILES MAY BE OPENED FOR RECORD ACCESS

* THE FILE MAY RESIDE ON ANY NOS/VE DEVICE (DISC, TAPE, TERMINAL ETC.)

* DATA ON THE FILE IS ORGANIZED INTO RECORDS AND BLOCKS (CONTROL INFORMATION IS INVISIBLE TO THE ACCESSOR)

* DATA MOVEMENT IS TRIGGERED BY PROGRAM REQUEST

* WHEN I/O IS PERFORMED TO A RANDOM BYTE ADDRESS, SUFFICIENT SPACE ON DISK IS ALLOCATED TO THE FILE TO COVER THE RANGE BETWEEN CURRENT EOI AND NEW EOI

PHYSICAL ACCESS

* PHYSICAL ACCESS
  - DATA TRANSFER IS IN PHYSICAL UNITS (DISC MTUS, TAPE BLOCKS)
  - USER CONTROLS NUMBER AND SIZE OF BUFFERS
  - DATA IS TRANSFERRED BETWEEN THE USER BUFFER AND THE DEVICE BY PPU BASED DEVICE DRIVER

* BOTH TEMPORARY & PERMANENT FILES MAY BE OPENED FOR PHYSICAL ACCESS

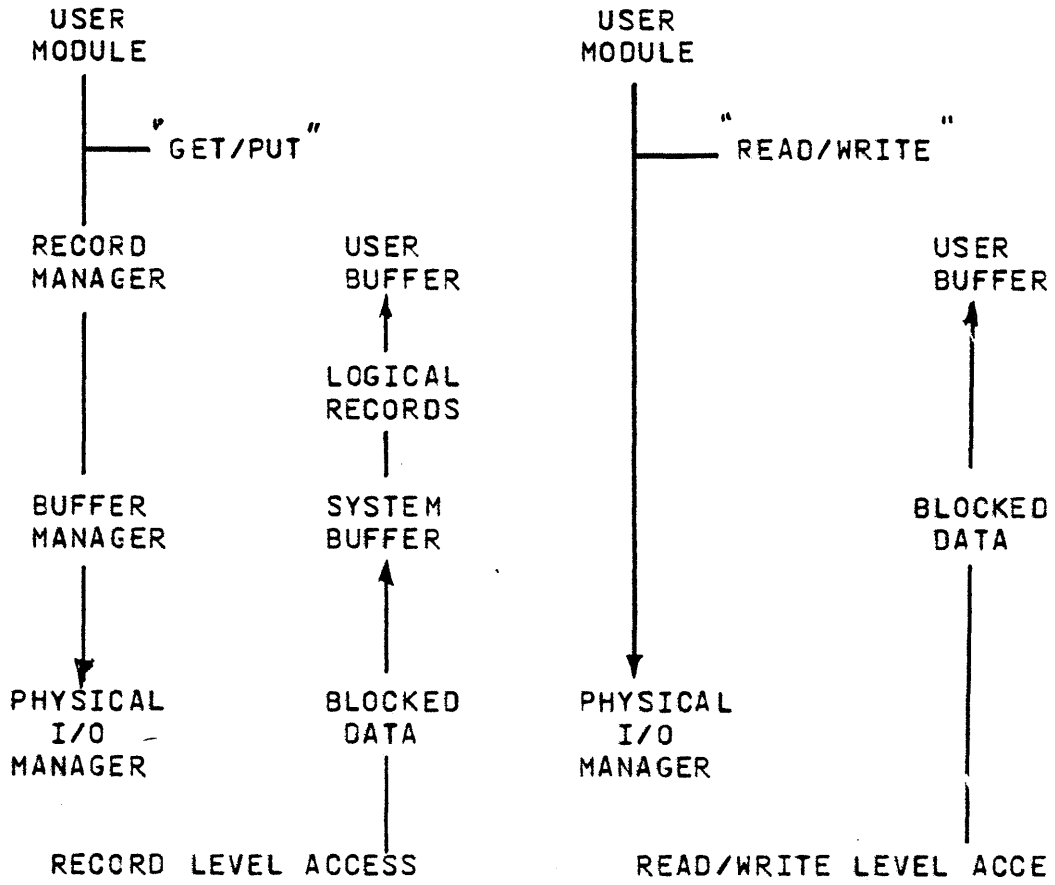
* THE FILE MAY RESIDE ON DISK OR TAPE

* FILE CONTENT IS DEFINED SOLELY BY THE USER. NO STRUCTURE IS IMPOSED BY NOS/VE

* DATA MOVEMENT IS TRIGGERED BY PROGRAM REQUEST

* WHEN I/O IS PERFORMED TO A RANDOM BYTE ADDRESS, SUFFICIENT SPACE ON DISK IS ALLOCATED TO THE FILE TO COVER THE RANGE BETWEEN CURRENT EOI AND NEW EOI

RECORD MANAGER & BUFFER MANAGER



* RECORD MANAGER PROCESSES USER REQUESTS FOR DATA TRANSFER BETWEEN AN ACTIVE FILE AND USER WORKING STORAGE

* BUFFER MANAGER INTERFACES TO PHYSICAL I/O MANAGER TO SERVICE REQUESTS FOR DATA TRANSFER BETWEEN A FILE AND A SET OF BUFFERS

SEGMENT ACCESS

* SEGMENT ACCESS
  - A VIRTUAL MEMORY SEGMENT IS ASSOCIATED WITH THE FILE
  - THE FILE APPEARS TO THE USER AS PART OF HIS ADDRESS SPACE. LOAD, STORE & COPY INSTRUCTIONS ACCESS THE DATA.
  - PAGE FAULT HANDLER ENSURES THAT REFERENCED PAGES ARE AVAILABLE IN REAL MEMORY

* BOTH TEMPORARY AND PERMANENT FILES MAY BE OPENED FOR SEGMENT LEVEL ACCESS

* FILES WHICH ARE TO BE ACCESSED AS VIRTUAL MEMORY SEGMENTS MUST RESIDE ON MASS STORAGE DEVICES

* REAL MEMORY IS ALLOCATED TO A SEGMENT FILE IN PAGES

* AS DATA WITHIN THE SEGMENT IS READ PAGE MANAGER INTERFACES TO PHYSICAL I/O MANAGER TO READ THE REFERENCED PAGE FROM THE FILE INTO REAL MEMORY

* WHEN ADDRESSES WITHIN THE SEGMENT BEYOND THE FILE ARE REFERENCED
  - SPACE IS ALLOCATED TO THE FILE TO COVER THE BYTE ADDRESS RANGE BETWEEN THE FILE EOI AND THE PAGE INCLUDING THE REFERENCED ADDRESS

PREPARATION FOR FILE ACCESS

* PRIOR TO OPENING A FILE THE USER MAY

  - CHANGE DEFAULT FILE ATTRIBUTES
    ("FILE" COMMAND/REQUEST)

  - REQUEST THE FILE BE ASSIGNED TO A PARTICULAR
    DEVICE
    ("REQUEST" COMMAND/REQUEST)

  - REQUEST A PERMANENT FILE BE MADE LOCAL
    ("GET" OR "ATTACH" COMMAND/REQUEST)

* THE FILE MUST BE OPENED BEFORE ANY REFERENCE

  "OPEN"

  - IS GIVEN A LOCAL FILE NAME

  - IS GIVEN AN ACCESS LEVEL (SEGMENT, RECORD,
    PHYSICAL)

  - CHANGES FILE ATTRIBUTES AS REQUESTED

  - GENERATES A FILE IDENTIFIER

FILE MANAGER & FILE SERVICES

* FILE MANAGER PROCESSES USER REQUESTS FOR FILE ACCESS, STATUS AND CONTROL
  - PROCESSES FILE INITIATION & TERMINATION (OPEN & CLOSE)
  - UPDATES CURRENT FILE DESCRIPTION (FILE)
  - PROCESSES FILE LABELS
  - PROCESSES FILE POSITIONING REQUESTS (SKIP, REWIND)
  - HANDLES REMOVABLE VOLUME SWITCHING
  - HANDLES STATUS CHECKS FOR USER I/O REQUESTS SUBMITTED WITH "NO WAIT" OPTION

* FILE SERVICES COMPONENT CONTAINS UTILITY PROCEDURES USED BY LOGICAL I/O MANAGER
  - VOLUME SWAPPING
  - FILE POSITIONING
  - ERROR PROCESSING

PHYSICAL I/O PROCESSING

PHYSICAL I/O MANAGER

- DEVICE QUEUE MANAGERS
- DEVICE ALLOCATORS
- DEVICE DRIVERS

* PHYSICAL I/O MANAGER HANDLES ALL PHASES OF DATA TRANSFER BETWEEN THE REQUESTOR'S BUFFERS AND THE EXTERNAL DEVICE
  - TRANSLATES LOGICAL FILE ADDRESS TO PHYSICAL DEVICE ADDRESS
  - ALLOCATES DEVICES TO FILES
  - ALLOCATES SPACE ON A DEVICE
  - MANAGES QUEUES OF REQUESTS FOR DEVICES
  - PROVIDES INTERFACE TO DEVICE CONTROL HARDWARE FOR ALL EXTERNAL DEVICES (E.G. DISC, TAPE, UNIT RECORD)
  - PROVIDES INTERFACE TO COMMUNICATIONS AND CHANNEL LINK SOFTWARE

I/O PROCESSING - COMMUNICATION

I/O REQUEST PACKET

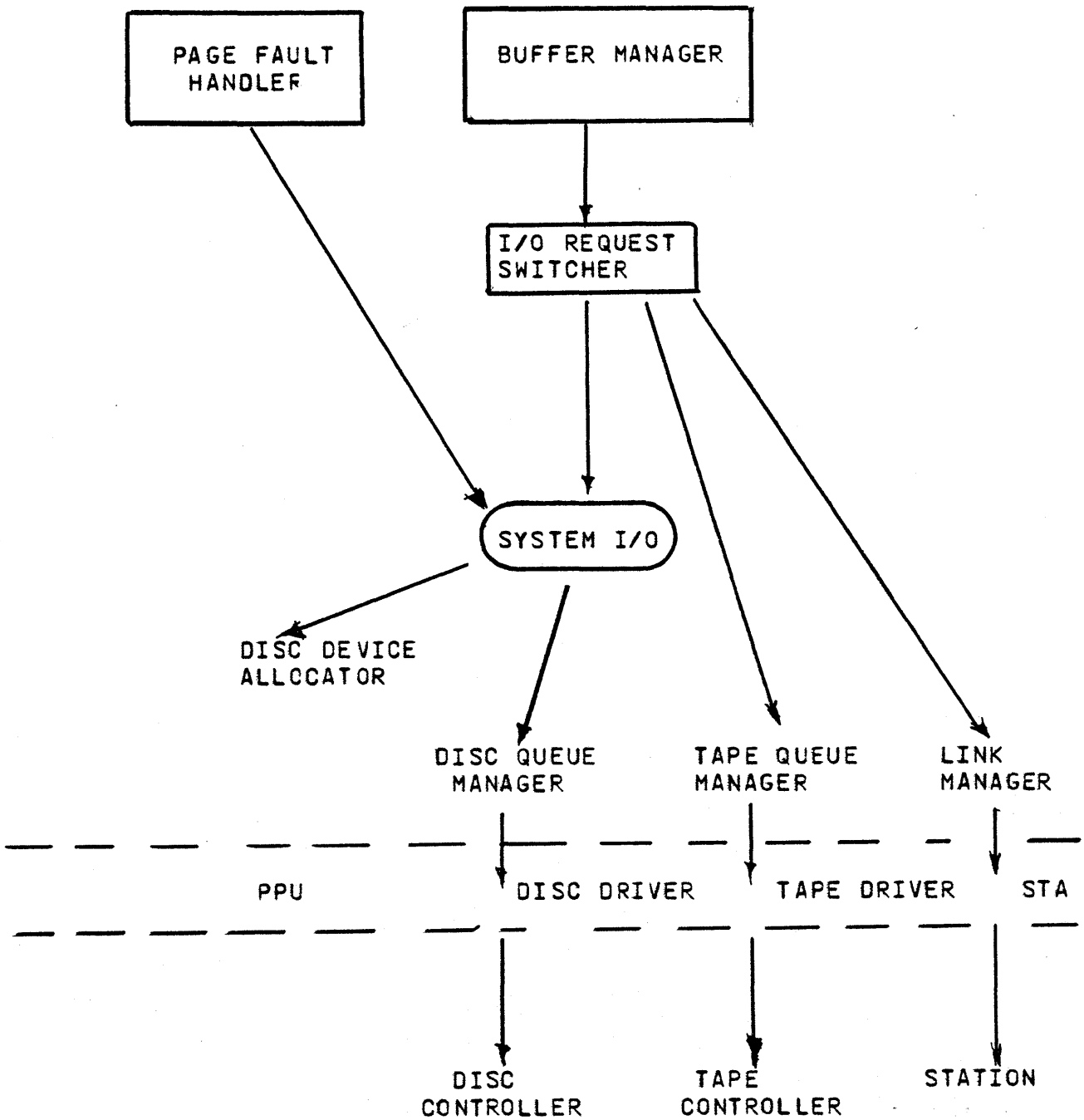
- FUNCTION CODE
- STATUS
- BUFFER ADDRESSES
- NO. OF BYTES TO BE ACCESSED
- FILE/DEVICE TO BE ACCESSED
- FILE/DEVICE POSITION
- LINK TO REQUEST INITIATOR
- LINK TO RESPONSE DESTINATION
- LINK INTO REQUEST QUEUE

* IORP IS BUILT BY BUFFER MANAGER TO DESCRIBE THE I/O REQUEST
  - QUEUING DEVICE FOR I/O REQUESTS
  - ENABLES ASYNCHRONOUS PROCESSING OF I/O REQUESTS
  - ROUTES I/O COMPLETION SIGNAL TO THE REQUESTOR

* DRIVER QUEUE TABLE IS USED TO QUEUE I/O REQUESTS FOR SERVICING BY EACH DEVICE DRIVER. DRIVERS ARE PPU RESIDENT AND POLL THEIR QUEUE FOR A WAITING IORP

* HAVING COMPLETED A REQUEST THE DRIVER CAUSES AN EXTERNAL INTERRUPT TO CAUSE QUEUE MANAGER TO ROUTE A SIGNAL TO THE REQUESTOR TASK

PHYSICAL I/O PROCESSING - OVERVIEW



PHYSICAL I/O MANAGER - COMPONENTS

* I/O REQUEST SWITCHER ROUTES LOGICAL I/O REQUESTS TO APPROPRIATE QUEUE MANAGER

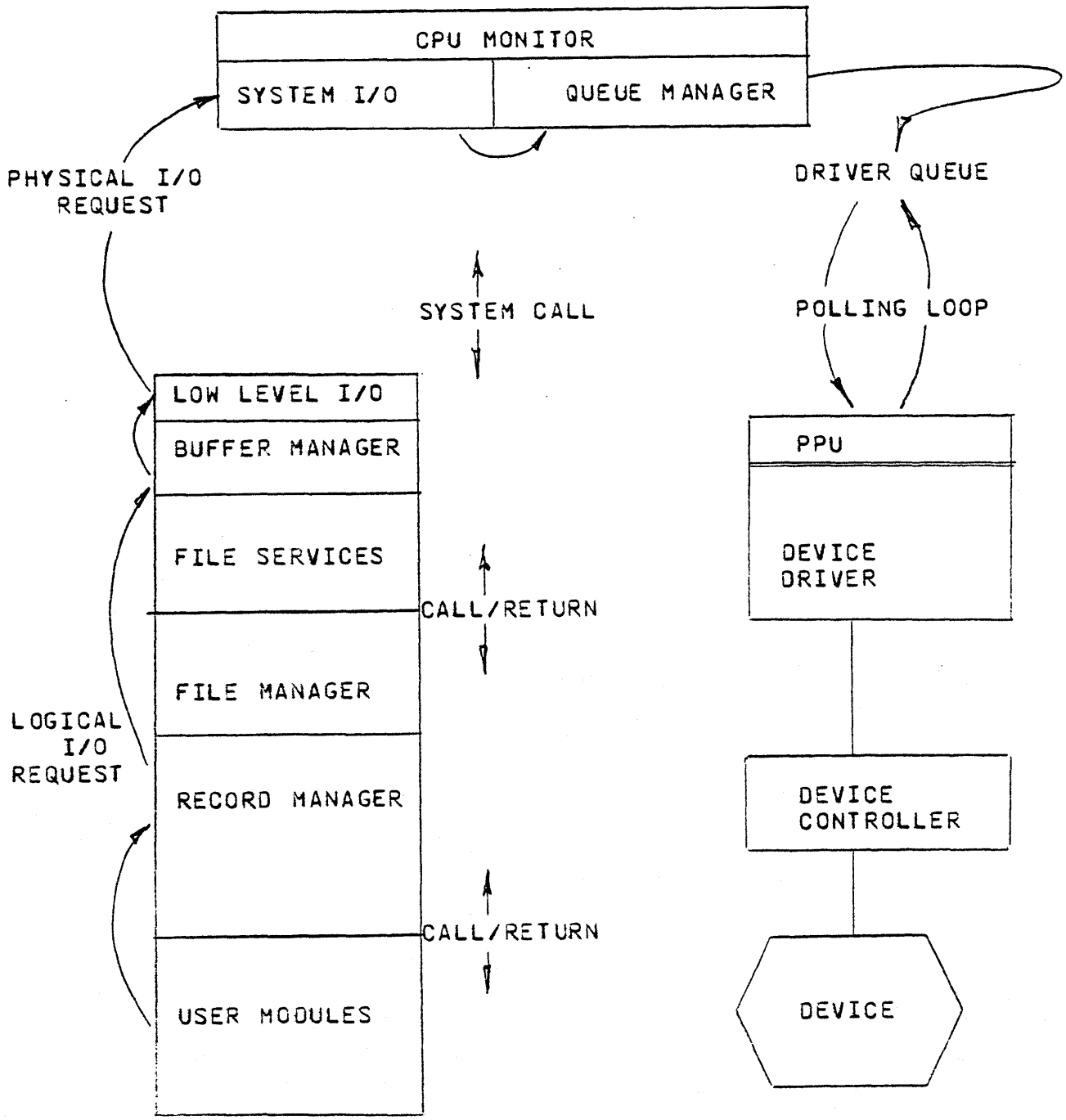
* SYSTEM I/O MAPS LOGICAL FILE POINTERS INTO PHYSICAL MASS STORAGE ADDRESSES

* DEVICE ALLOCATORS PROCESS REQUESTS BY SYSTEM I/O TO ALLOCATE AND DEALLOCATE SPACE ON MASS STORAGE DEVICES

* EACH CLASS OF DEVICE HAS A QUEUE MANAGER INTERFACE TO THE PPU-BASED DEVICE DRIVER
  - QUEUES I/O REQUESTS FOR EACH UNIT
  - SUBMITS THE NEXT REQUEST TO THE DRIVER
  - OPTIMIZES THE ORDER OF REQUESTS SUBMITTED BASED ON CHARACTERISTICS AND UTILIZATION OF THE DEVICE
  - PROCESSES EXTERNAL INTERRUPTS FROM THE PPU BASED DRIVER
  - RETURNS I/O STATUS TO THE REQUESTOR
  - INTERFACES TO TASK SCHEDULER FOLLOWING I/O COMPLETION
  - LOCKS/UNLOCKS CENTRAL MEMORY BUFFER PAGES IN REAL MEMORY

* EACH TYPE OF DEVICE CONTROLLER HAS A CORRESPONDING DRIVER WHICH IS RESIDENT IN ONE OR MORE PPUS
  - PROVIDES INTERFACE TO THE PHYSICAL DEVICE CONTROLLER
  - PICKS UP IORPS FROM THE CORRESPONDING DEVICE TABLE IN CENTRAL MEMORY

I/O REQUEST PROCESSING - EXAMPLE
RECORD ACCESS



RECORD TRANSFER FROM USER MODULE TO A DEVICE

PERMANENT FILE MANAGEMENT - OVERVIEW

* AT RELEASE 1 PERMANENT FILES MAY ONLY RESIDE ON RMS DEVICES. LATER RELEASES WILL SUPPORT PERMANENT FILES ON MAGNETIC TAPE AND MSS MEDIA

Mass Storage [illegible]

* PERMANENT FILE MANAGEMENT FUNCTIONS ARE
  - MANAGING THE REGISTRATION OF FILES IN A CATALOG
  - ESTABLISHING JOB ACCESS TO A PERMANENT FILE
  - ACCESS CONTROL MECHANISMS FOR LIMITING FILE ACCESS TO PERMITTED USERS ACCORDING TO AUTHORIZED ACCESS MODES
  - MANAGING THE MIGRATION OF FILES TO/FROM DIFFERENT STORAGE MEDIA
  - BACKUP AND RECOVERY OF PERMANENT FILES

* CONTROL OF ACCESS TO PERMANENT FILES IS A MAJOR ELEMENT OF NOS/VE SECURITY AND PROTECTION
  - USER/FAMILY/ACCOUNT/PROJECT QUALIFICATION
  - ACCESS MODES - READ/WRITE/APPEND/EXECUTE/NONE
  - PASSWORD
  - SECURITY LEVEL
  - RING BRACKETS

PERMANENT FILE CATALOG

* A CATALOG IS A SYSTEM FILE CONTAINING INFORMATION LINKING LOGICAL NAMES OF ELEMENTS (E.G. PERMANENT FILES, DEVICES) WITH THEIR DESCRIPTION AND ACCESS CONTROL LISTS
  - DESCRIPTOR HAS IDENTIFICATION & LOCATION OF THE ELEMENT
  - ACCESS CONTROL LIST HAS USER NAMES AND THEIR PERMITTED ACCESS STYLE
  - ACCESS CONTROL LIST IS MAINTAINED BY THE ELEMENT OWNER

* ALL PERMANENT FILES ARE REGISTERED IN A CATALOG

* EACH USER IS ASSOCIATED WITH A MASTER CATALOG AND MAY CREATE SUB-CATALOGS

* EACH PERMANENT FILE HAS A NAME UP TO 31 CHARACTERS WHICH IS UNIQUE WITHIN THE MASTER CATALOG

* MULTIPLE VERSIONS OF A FILE MAY BE REGISTERED AS CYCLES UNDER A SINGLE NAME
  - EACH CYCLE IS A UNIQUE FILE
  - ACCESS CONTROL LIST IS COMMON FOR ALL CYCLES

* USERS AND THEIR PERMANENT FILES ARE LOGICALLY GROUPED AS FAMILIES
  - LOGICAL FAMILY NAME IS USED TO ROUTE INFORMATION WITHIN A COMPLEX OF MAINFRAMES

ACCESS CONTROL

* EACH FERMANENT FILE HAS AN ACCESS CONTROL LIST. EACH ENTRY SPECIFIES ACCESS IN TERMS CF FAMILY, ACCOUNT, PROJECT & USER NAMES E.G.

F A P U USER U IN FAMILY F, ACCOUNT A & PROJECT P MAY ACCESS THE FILE

F A P - ANYONE IN FAMILY F, ACCOUNT A & PROJECT P MAY ACCESS THE FILE

F - - U USER U IN FAMILY F REGARDLESS OF ACCOUNT OR PROJECT MAY ACCESS THE FILE

* EACH ENTRY SPECIFIES THE USAGE MODE PERMITTED THE ASSOCIATED IDENTIFICATION. ONE OR MCRE OF

  - "READ" MAY READ THE FILE
  - "WRITE" MAY WRITE STARTING AT THE BEGINNING OF THE FILE OR MAY ESTABLISH MODIFY OR APPENC USAGE
  - "MODIFY" MAY REPLACE, DELETE OR INSERT INFORMATI
  - "EXECUTE" MAY EXECUTE THE FILE
  - "NONE" PROHIBITED ACCESS TO THE FILE

* OWNER MAY REQUIRE A PASSWORD TO BE GIVEN WHENEVER THE FILE IS ACCESSED

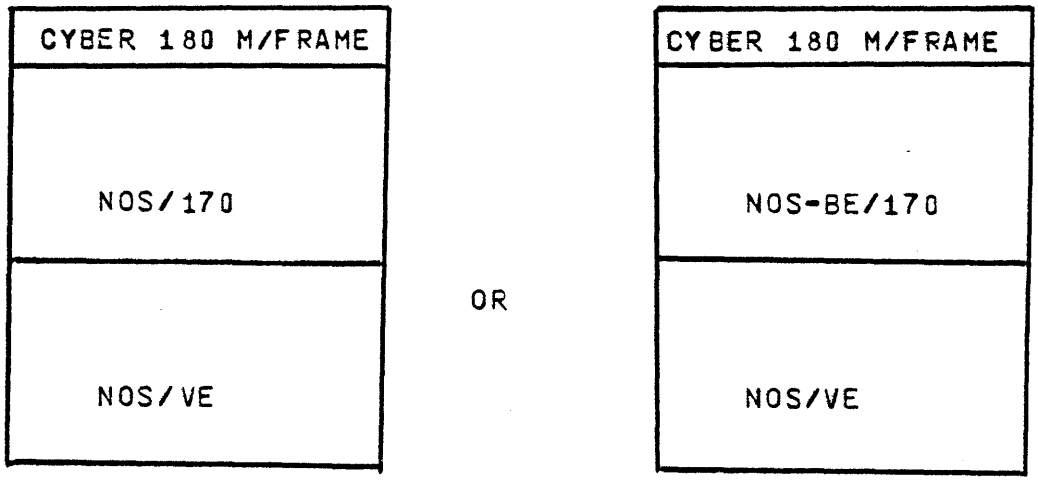
* USER'S SECURITY LEVEL, ESTABLISHED AT LOGIN, MUST BE .GE. SECURITY LEVEL OF THE FILE

* RING BRACKETS OF THE REQUESTOR MUST MATCH THOSE OF THE FILE FOR THE DESIRED USAGE

VIRTUAL ENVIRONMENT (DUAL-STATE)

DEADSTART/RECOVERY

DUAL-STATE



* MIGRATION AID FOR USER CONVERSION TO NOS/VE

* TO PROVIDE CY170 PRODUCT FEATURES PRIOR TO DEVELOPING CY180 COUNTERPARTS
  - LOCAL BATCH JOB ENTRY AND DISPOSAL
  - OPERATOR COMMUNICATION
  - NETWORK SUPPORT

VIRTUAL ENVIRONMENT PARTITIONING

* THE SYSTEM RESOURCES ARE PARTITIONED BETWEEN CY170 & CY180 LOGICAL MACHINES

* CPU IS PARTITIONED USING THE VMID FIELD IN THE EXCHANGE PACKAGE. DETERMINES HOW THE CPU WILL
  - FETCH AND INTERPRET INSTRUCTIONS
  - INTERPRET THE REGISTER FILE
  - INTERPRET INTERRUPTS

* CPU ACCESS TO CENTRAL MEMORY
  - CY170 ADDRESSES MAP INTO REAL MEMORY ADDRESSES 0-N
  - CY180 ADDRESSES MAP INTO (N+1) - (MEMORY SIZE-1)

* PPU ACCESS TO CENTRAL MEMORY
  - PPUS ARE ASSIGNED TO EITHER 170 SYSTEM OR NOS/VE
  - IOU BOUNDS REGISTER LIMITS WRITE ACCESS TO CM

* CHANNELS ARE SOFTWARE PARTITIONED TO ACCESS ONLY CY170 OR CY180 PERIPHERAL DEVICES. (EXCEPT MAINTENANCE CHANNEL)

* DUAL-CHANNEL CONTROLLERS MAY BE SHARED BETWEEN CY170 & CY180
  - DISKS WOULD BE SOFTWARE PARTITIONED FOR CY170 OR CY180 FORMAT I/O (I.E. 64X60BIT SECTORS OR 256X64BIT SECTORS)

DUAL-STATE - OVERVIEW

* USER VIEW IS OF TWO LOGICALLY DISTINCT MACHINES IN A MULTI-MAINFRAME ARRANGEMENT
  - JOBS AND FILES BELONG TO CY170 OR CY180
  - AT DEADSTART TIME; MEMORY, PPS, CHANNELS AND PERIPHERAL DEVICES ARE PARTITIONED BETWEEN CY170 AND CY180 LOGICAL MACHINES
  - INTERFACE BETWEEN CY170 AND CY180 IS THROUGH OPERATING SYSTEM SERVICES

* NOS/VE RHF FACILITY RELEASE 1 WILL SUPPORT ACCESS BETWEEN CY180 & CY170 STATE VIA CENTRAL MEMORY

* LATER RELEASES WILL SUPPORT CONNECTION BETWEEN MULTI-MAINFRAMES VIA CHANNEL COUPLERS OR COMMUNICATION LINES

* DUAL-STATE FUNCTIONS ARE NOT COMPLETELY SYMMETRIC AT RELEASE 1 BUT WILL ALLOW
  - JOB SUBMISSION TO ANOTHER LOGICAL MACHINE
  - THE CONTROL AND MONITORING OF A JOB EXECUTING IN ANOTHER LOGICAL MACHINE
  - FILE COPYING AND ROUTING BETWEEN LOGICAL MACHINES
  - INTER-JOB MESSAGE COMMUNICATION

* AT RELEASE 1 SOME "FRONT-END" FUNCTIONS MAY BE DEFINED ON CY170 LOGICAL MACHINE TO SERVE BOTH
  - INPUT/OUTPUT SPOOLING
  - OPERATOR COMMUNICATION
  - NETWORK COMMUNICATION

NOS/VE REMOTE HOST FACILITY

* PROVIDES A COMMUNICATION PATH FOR FILE TRANSFER
  - NOS/VE & NOS/170
  - NOS/VE & NOS/BE

* SUPPORTS COMMUNICATION BETWEEN LOGICAL MACHINES (DUAL-STATE)

* USERS MUST BE VALIDATED FOR ACCESS TO THE REMOTE HOST FACILITY
  - NOS/VE USES FAMILY NAME FOR MAINFRAME ID.
  - REQUESTS TO ACCESS PERMANENT FILES VIA THE RHF INCLUDE USER VALIDATION (LINK_USER COMMAND)

* FILE SIZE LIMITATIONS WILL BE ASSOCIATED WITH EACH LINKED FAMILY TO RESTRICT TRANSFERS VIA THE RHF

NOS/VE REMOTE HOST FACILITY - COMPONENTS

* NOS/VE REMOTE HOST FACILITY JOB
  - COMMUNICATES WITH THE LINKED SYSTEM
  - RECEIVES INPUT JOBS AND SENDS OUTPUT FILES

* LINKED COMMUNICATION SERVICES
  - USER INTERFACE FOR PERMANENT FILE HANDLING VIA THE LINK
    (GET, SAVE, REPLACE, PURGE, PERMIT & CATLIST)

* LINKED FILE CONVERSION
  - LINK FILES ARE INTERCHANGE FORMAT
  - QUEUE FILES AND PERMANENT FILES ARE CONVERTED BEFORE AND AFTER TRANSFER

NOS/VE REMOTE HOST FACILITY - COMPONENTS

* NOS/VE REMOTE HOST FACILITY JOB

- COMMUNICATES WITH THE LINKED SYSTEM

- RECEIVES INPUT JOBS AND SENDS OUTPUT FILES

* LINKED COMMUNICATION SERVICES

- USER INTERFACE FOR PERMANENT FILE HANDLING VIA THE LINK
  (GET, SAVE, REPLACE, PURGE, PERMIT & CATLIST)

* LINKED FILE CONVERSION

- LINK FILES ARE INTERCHANGE FORMAT

- QUEUE FILES AND PERMANENT FILES ARE CONVERTED BEFORE AND AFTER TRANSFER

DEADSTART/RECOVERY

* DEADSTART FUNCTION ACTIVATES NOS/VE TO THE STATE IN WHICH IT IS READY TO EXECUTE USER WORKLOADS

* INCLUDES RECOVERY OR INITIALIZATION OF
  - PERMANENT FILE BASES
  - SYSTEM LOG FILES
  - I/O QUEUES
  - HARDWARE CONFIGURATION INFORMATION
  - USER/SYSTEM JOBS AND THEIR TRANSIENT FILES

* DEADSTART/RECOVERY FUNCTION SUPPORTS BOTH DUAL-STATE AND STAND-ALONE CY180 OPERATION

* DEADSTART/RECOVERY LEVELS SUPPORTED

1 INSTALLATION (NO RECOVERY)

2 RECOVER PERMANENT FILE BASE

3 LEVEL 2 PLUS I/O QUEUES

4 LEVEL 3 PLUS SWAPPED JOBS

5 LEVEL 4 PLUS EXECUTING JOBS

6 RECOVER FROM A SYSTEM CHECKPOINT FILE

DEADSTART/RECOVERY - STAND ALONE MODE

COMMON TEST & INITIALIZATION
MCU DEADSTART MONITOR
CONFIGURATION RECORDS
BASIC O/S DEADSTART JOB (CM IMAGE)
SYSTEM RMS CONTROL WARE
SYSTEM RMS DRIVER
OTHER SYSTEM DRIVERS & CONTROLWARE
BALANCE OF NOS/VE (LOAD MODULES)
PRODUCT SET LIBRARIES

CYBER 180 DEADSTART FILE FORMAT

DEADSTART/RECOVERY - COMPONENTS

* COMMON TEST AND INITIALIZATION (CTI)
  - COMMON TO ALL SYSTEMS DEADSTARTED ON CY180/170
  - LOADS FIRMWARE
  - DUMPS CM, PPU MEMORY, CONTROLLER MEMORY AND AND CONTROL STORE TO DISC
  - VERIFIES MAINFRAME HARDWARE
  - PROVIDES OPERATOR COMMUNICATION FOR CTI ONLY
  - SELECTS AND DETERMINES MAINFRAME ATTRIBUTES (E.G. CM SIZE, CPU ON/OFF, PAGE SIZE)
  - INITIATES OPERATING SYSTEM LOAD FROM THE SELECTED DEVICE

* CTI TERMINATES WHEN THE HARDWARE IS IN A KNOWN STATE AND THE NOS/VE MCU DEADSTART MONITOR IS EXECUTING IN THE MAINTENANCE CONTROL PPU

DEADSTART/RECOVERY - MINIMUM SYSTEM LOAD

* MINIMUM SYSTEM LOAD ESTABLISHES THE ENVIRONMENT FOR NOS/VE DEADSTART JOB
  - ESTABLISHES THE SOURCE FOR SYSTEM LOAD (LOCAL TAPE, LOCAL DISC, CY170 STATE)
  - ESTABLISHES THE CONFIGURATION
  - LOADS THE CM IMAGE OF THE BASIC NOS/VE SYSTEM
  - BUILDS BASIC NOS/VE TABLES E.G. SYSTEM PAGE TABLE, STATUS TABLE)
  - LOADS THE NOS/VE DEVICE DRIVERS & DOWN-LOADS THE CONTROLWARE
  - MCU HALTS THE CPU(S), ESTABLISHES PROCESS STATE REGISTERS AND EXCHANGES THE CPU(S) TO NOS/VE MONITOR STATE
  - MCU REVERTS TO MAINTENANCE MONITOR ROLE

DEADSTART/RECOVERY - DEADSTART JOB

* THE DEADSTART JOB IS THE FIRST NOS/VE JOB TO BE ACTIVATED FOLLOWING SYSTEM INITIALIZATION
  - LOADS REMAINING DRIVERS AND CONTROLWARE
  - CHECKS THE SYSTEM RMS SET FOR VALID LABELS AND BUILDS RMS FLAW TABLES
  - RECOVERS OR INITIALIZES THE PERMANENT FILE BASE AND I/O QUEUE FILES
  - BUILDS SYSTEM LIBRARIES ON THE SYSTEM RMS SET
  - ESTABLISHES SYSTEM JOBS
  - RECOVERS USER JOBS (IF SELECTED)
  - LOGS THE REASON FOR DEADSTART IN THE SYSTEM LOG

DEADSTART/RECOVERY - DUAL-STATE INITIALIZATION

* DUAL-STATE INITIALIZATION ENABLES THE NOS/VE DEADSTART PACKAGE TO BE ESTABLISHED FROM A CY170 ENVIRONMENT

* THE OPERATOR INITIATES A JOB IN THE CY170
  - ATTACHES THE CM, PPS, CHANNELS AND PERIPHERAL EQUIPMENT NEEDED TO ESTABLISH THE NOS/VE OPERATING ENVIRONMENT
  - WHEN RESTARTING NOS/VE FROM THE CY170 ENVIRONMENT MAY DUMP CM, PP AND NON-SHARED CONTROLLER MEMORIES
  - LOADS THE NOS/VE MCU DEADSTART MONITOR INTO THE MCU FROM CY170 SYSTEM LIBRARY
  - LOADS NOS/VE FROM A NOS/170 PERMANENT FILE VIA THE CY170 JOB
  - WHEN THE BASIC OPERATING SYSTEM IS RUNNING THE REMAINING NOS/VE MODULES ARE TRANSFERRED TO THE NOS/VE SYSTEM DISC

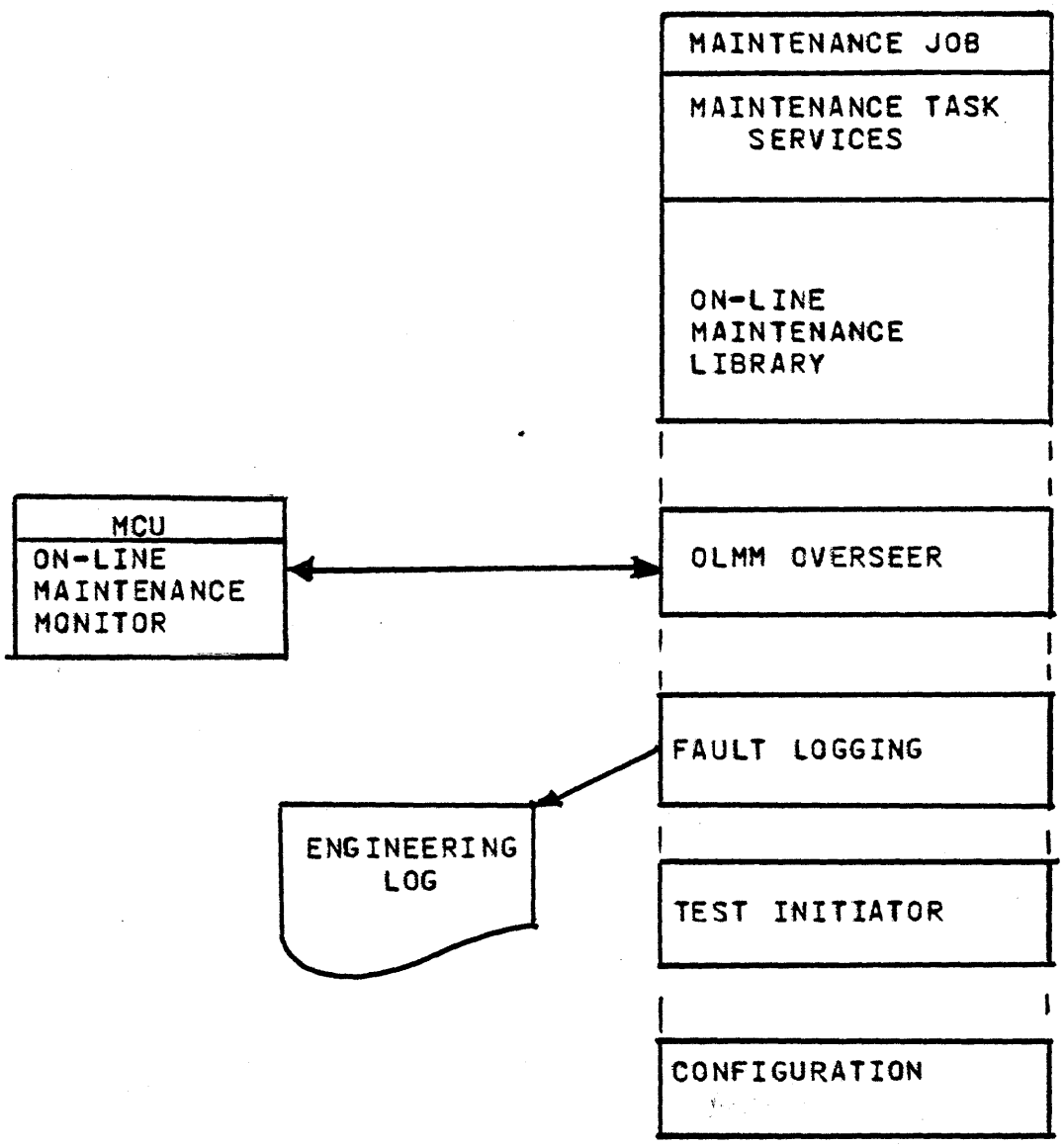
MAINTENANCE SERVICES

MAINTENANCE SERVICES

* NOS/VE MAINTENANCE SERVICES SUPPORTS 180 STATE
  - HARDWARE TEST/DIAGNOSIS
  - REMEDIAL MAINTENANCE
  - ON-LINE PREVENTIVE MAINTENANCE OF NON-CRITICAL COMPONENTS

* NOS/VE MAINTENANCE SERVICES
  - DETECTS AND REACTS TO ALL HARDWARE REPORTED ERRORS WHILE MAINTAINING SYSTEM INTEGRITY
  - RECONFIGURES THE SYSTEM TO DEFER MAINTENANCE UNTIL A REPAIR PERIOD
  - CUSTOMER WORK IS CONTINUED WHILE DEGRADING TO MINIMUM CONFIGURATION IF NECESSARY
  - RECOVERABLE ERRORS ARE TRANSPARENT TO THE USER
  - MAINTAINS THE ENGINEERING LOG

MAINTENANCE SERVICES - OVERVIEW



NOS/VE MAINTENANCE JOB

* THE MAINTENANCE JOB IS A SYSTEM JOB WHICH IS ALWAYS ACTIVE IN THE NOS/VE SYSTEM

* MAINTENANCE JOB COMPONENTS HANDLE ALL ASPECTS OF ON-LINE MAINTENANCE AND FAULT REPORTING

  - ON-LINE MAINTENANCE MONITOR OVERSEER TASK INTERFACES WITH THE MCU

  - LOGGING TASK RECORDS FAULTS DETECTED BY CPU MONITOR, PHYSICAL I/O AND MCU

  - HANDLES DYNAMIC RECONFIGURATON

  - SUBMITS TEST/DIAGNOSTIC JOBS TO NOS/VE

  - ON-LINE MAINTENANCE LIBRARY CONTAINS TEST & DIAGNOSTIC PROCEDURES

  - MAINTENANCE TASK SERVICES PROVIDES A SPECIAL TASK SERVICES ENVIRONMENT. E.G.

    . MAINTENANCE CHANNEL I/O
    . PPU LOADING & DUMPING
    . FIRMWARE LOADING/DUMPING
    . ENGINEERING LOG ENTRY

ON-LINE MAINTENANCE MONITOR

* ON-LINE MAINTENANCE MONITOR RESIDES IN A PPU, THE MAINTENANCE CONTROL UNIT (MCU)

* OLMM MONITORS THE STATUS OF CPUS, PPUS, CM IOU AND CEM VIA THE MAINTENANCE CHANNEL
  - RECORDS RECOVERED ERRORS AND REPORTS WHEN THRESHOLD REACHED
  - PASSES FAULT DATA TO THE MAINTENANCE JOB
  - ACTIVATES SYSTEM RECOVERY VIA CTI OR DUAL-STATE
  - PRESERVES CRITICAL FAULT DATA FOR OFF-LINE DIAGNOSTICS
  - COMMUNICATES WITH SYSTEM OPERATOR

* OLMM OVERSEER TASK OF THE NOS/VE MAINTENANCE JOB HANDLES ALL COMMUNICATION WITH THE MCU

OPERATOR COMMUNICATION

OPERATOR COMMUNICATION

* SUPPORTS COMMUNICATION BETWEEN
  - NOS/VE AND SYSTEM OPERATORS
  - USER JOBS AND SYSTEM OPERATORS

* NOS/VE OPERATOR CONSOLES MAY BE
  - THE STANDARD SYSTEM CONSOLE (752 TERMINAL) CONNECTED THROUGH THE TWO PORT MUX (SUPPORTED AT RELEASE 2)
  - CC545 170 CONSOLE MAY BE USED VIA DUAL-STATE AT RELEASE 1
  - ANY INTERACTIVE TERMINAL (CONNECTED VIA CY170 AT RELEASE 1)

* AN OPERATOR CONSOLE IS A TERMINAL "LOGGED-IN" WITH SYSTEM OPERATOR PRIVILEGES
  - OPERATOR COMMANDS & DISPLAYS ARE PROCESSED BY NOS/VE INTERACTIVE JOBS HAVING SYSTEM OPERATOR PRIVILEGES GRANTED BY THE NOS/VE USER VALIDATION
  - THE INSTALLATION MAY DISTRIBUTE ACCESS PRIVILEGES BETWEEN USERS
    - ON-LINE EQUIPMENT CONFIGURATION COMMANDS
    - TAPE DRIVE ASSIGNMENT COMMANDS
    - JOB QUEUE ACCESS AND CONTROL
    - STATUS AND CONTROL OF SYSTEM JOBS

NOS/VE OPERATOR FACILITY - FUNCTION

* NOS/VE OPERATOR FACILITY PROVIDES THE LINK BETWEEN THE SYSTEM OPERATOR AND NOS/VE
  - STATUS AND CONTROL OF HARDWARE COMPONENTS
  - STATUS AND CONTROL OF NOS/VE USER JOBS AND THEIR RESOURCE ALLOCATION
  - STATUS AND CONTROL FOR THE OPERATING SYSTEM, SYSTEM JOBS AND SPECIAL APPLICATIONS
  - ALLOWS NOS/VE TO REQUEST OPERATOR ASSISTANCE FOR TAPE MOUNTS
  - PROVIDES VISIBLE INFORMATION ON SYSTEM OPERATION, CURRENT PARAMETER VALUES ETC.
  - REPORTS HARDWARE AND SOFTWARE PROBLEMS
  - ALLOWS OPERATOR-JOB AND OPERATOR-TERMINAL COMMUNICATION
  - SUPPORTS ON-LINE SYSTEM DEBUGGING
  - SUPPORTS ON-LINE DIAGNOSTIC INITIATION AND CONTROL

* NOS/VE DEADSTART WILL COMMUNICATE WITH THE OPERATOR THROUGH THE CY170 UTILITY WHICH LOADS THE BASIC NOS/VE SYSTEM

NOS/VE OPERATOR FACILITY

* CONSOLE INTERFACE
  - INTERACTIVE TERMINALS USED AS SYSTEM OPERATOR CONSOLES REQUIRE NO SPECIAL INTERFACE
  - CC545 CY170 CONSOLE K OR L DISPLAY INTERFACES TO A NOS/VE INTERACTIVE JOB
  - THE STANDARD SYSTEM CONSOLE (752) IS DRIVEN BY MCU AND PHYSICAL I/O COMPONENT TO INTERFACE TO A NOS/VE INTERACTIVE JOB

* SYSTEM OPERATOR JOBS
  - HANDLED AS NORMAL NOS/VE INTERACTIVE JOBS
  - SYSTEM OPERATOR ACCESS PRIVILEGE ALLOWS USE OF RESTRICTED SCL COMMANDS

* SYSTEM INTERFACE PROCEDURES
  - SET OF TASK SERVICES & MONITOR FUNCTIONS WHICH CAN OBTAIN DATA & CONTROL JOBS, SYSTEM TABLES ETC.

NCS/VE OPERATOR FACILITY - COMMANC/DISPLAY

* OPERATOR COMMAND AND DISPLAY PROCESSORS ARE ACTIVATED BY COMMANDS
  - COMMANDS AND PROCEDURES MAY BE PRIVILEGED BY INSTALLATION OPTION
  - COMMANDS WHICH REQUIRE DISPLAYS WILL INTERFACE TO DISPLAY BUILDING PROCEDURES

* DISPLAY BUILDING PROCEDURES COLLECT AND STRUCTURE INFORMATICN FOR THE DISPLAY GENERATOR

* DISPLAY GENERATOR
  - ALL COMMAND & DISPLAY REQUESTS ARE LOGGED IN THE JOB & SYSTEM LOGS
  - FORMATS THE DISPLAY STRUCTURE BASED ON THE CONSOLE CHARACTERISTICS AND THE A DISPLAY TEMPLATE