

**MULTIPROGRAMMING
SYSTEM MANUAL**

PDP-6

MULTIPROGRAMMING TIMESHARING SYSTEM
MANUAL INDEX

CHAPTER I - INTRODUCTION TO THE PDP-6 MULTI-PROGRAMMING SYSTEM

DESIGN OBJECTIVES	7
CONSOLE USE OF THE SYSTEM FOR PROGRAMMING	
CONSOLE USE OF THE COMMON USER SERVICE PROGRAMS	
CONSOLE USE OF THE COMMON USER SERVICE PROGRAMS FOR DATA CONVERSION	
UNATTENDED STACK OR BATCH PROCESSING	
OPEN-ENDED SYSTEM WHICH CAN SERVE SPECIAL CONSOLES	
OPEN-ENDED SYSTEM WHICH CAN SERVE REAL TIME PROCESSES	
COMMON IO SUB-PROGRAMS	
COMMON SYSTEM SUB-PROGRAMS	
TIME SHARING SYSTEM STAGES	8
SYSTEM I - MULTI-PROGRAMMING TIME SHARING	
SYSTEM II - DRUM TIME SHARING	
SYSTEM III - DRUM-DISC TIME SHARING	
SYSTEM IV - MULTIPROCESSOR TIME SHARING	
HARDWARE FOR TIME SHARING	9
EXECUTIVE-USER MODE HARDWARE	
LARGE STORAGE	
DATA REPRESENTATION	
PRIORITY INTERRUPT	
TABLE OF MULTIPROGRAMMING SYSTEM HARDWARE	10
REQUIRED HARDWARE	
OPTIONAL HARDWARE	
OPTIONAL HARDWARE WITH SOFTWARE SUPPORT	
SOFTWARE MONITOR COMPONENTS	11
CONSOLES	
EXECUTIVE OR MONITOR STORAGE (EXECUTIVE MODE)	
MULTI-PROGRAMMING EXECUTIVE	
IO DEVICE SERVICE MODULES	
JOBS STORAGE (USER MODE)	
COMMON USER SERVICE PROGRAMS (CUSP)	
USER PROGRAMS (UP)	
IO DATA FILES ORGANIZATION	
SYSTEM LIBRARY FILE	
DIRECTORIES	
SYSTEMS	
CUSPS	
RELOCATABLE FILES	

CHAPTER II - CONSOLE USE OF SYSTEM

INTRODUCTION	16
CONSOLE STATES	16

MONITOR LEVEL CONTROL
COMMON USER SERVICE PROGRAM LEVEL CONTROL
USER PROGRAM LEVEL CONTROL

MONITOR COMMANDS	17
IJOB - INITIALIZE JOB COMMAND	
PJOB - PRINT JOB NUMBER COMMAND	
KJOB - KILL JOB COMMAND	
CORE - ASSIGN CORE COMMAND - COREN	
GET - A PROGRAM COMMAND - GET DEVICE FILE	
SAVE - JOB COMMAND - SAVE DEVICE FILE	
START - JOB - START LOC	
CONT - CONTINUE JOB COMMAND	
DDT - START USER DDT COMMAND	
STARTM - START MONITOR MODE COMMAND	
CONTM - CONTINUE MINITOR MODE COMMAND	
DETATCH - CONSOLE FROM JOB COMMAND	
ATTACH - CONSOLE TO JOB COMMAND	
ASSIGN - DEVICE TO JOB COMMAND	
DEASSIGN - DEVICE FROM JOB COMMAND	

GENERAL USE OF COMMON USER SERVICE PROGRAMS (CUSP)	26
CUSP COMMAND FORMATS	

MONITOR/USER CONSOLE ERROR MESSAGES	23
-------------------------------------	----

CHAPTER III - JOB AREA USE OF SYSTEM

GENERAL	28
---------	----

SYSTEM PARAMETERS IN JOB AREA	28
-------------------------------	----

JOBAC
JOBDAC
JOBUUO
JOBPC
JOBPDP
JOBREL
JOBEXT
JOBSAV
JOBIDA
JOBDDT
JOBSYM
JOBISA
JOBFF
JOBTEM
JOBINF
JOBICDP
JOBTRP
JOBUSY

SYSTEM PUSH DOWN LIST	31
-----------------------	----

PROGRAM SYSTEM INDEPENDENCE	31
-----------------------------	----

LOADING USER PROGRAMS	31
-----------------------	----

CHAPTER IV - USER INPUT OUTPUT COMMANDS

GENERAL	33
RING BUFFERS	34
BUFFER STRUCTURE	34
RESET COMMAND	35
ASSIGN COMMAND	35
INITIALIZATION	36
INIT COMMAND	
DATA MODES	
INBUF COMMAND	
OUTBUF COMMAND	
DATA CONTROL MODES	
NON DUMP INPUT OUTPUT COMMANDS	39
INPUT D COMMAND	
INPUT D,ADR COMMAND	
OUTPUT D COMMAND	
OUTPUT D,ADR COMMAND	
DUMP INPUT OUTPUT COMMANDS	41
GENERAL	
OUTPUT COMMAND	
INPUT D,ADR COMMAND	
INPUT D COMMAND	
COMMAND CONTROL LIST	
STATUS EXAMINATION COMMANDS	40
STATUS COMMAND	
STATZ COMMAND	
STATO COMMAND	
TABLE OF IO DEVICE CHARACTERISTICS	
TELETYPEWRITERS	43
PAPER TAPE READER	43
PAPER TAPE PUNCH	43
LINE PRINTER	43
CARD READER	44
DECTAPE	45
DECTAPE FORMAT	
LOOKUP COMMAND	
ENTER COMMAND	
MISCELLANEOUS COMMANDS	
NON DUMP OUTPUT COMMANDS	
NON DUMP INPUT COMMANDS	
DUMP INPUT,OUTPUT COMMAND	

MAGNETIC TAPE	47
MAGNETIC TAPE FORMAT	
DIRECTORY FORMAT	
PHYSICAL RECORDS FORMAT	
ENTER COMMAND - ENTER D,E	
LOOKUP COMMAND - LOOKUP D,E	
DATA TRANSMISSION COMMANDS DUMP MODE	
MTAPE MOVEMENT COMMAND - MTAPE D,FCN	
CLOSE COMMAND - CLOSE,D	
GET DUMP WORD COUNT COMMAND - GETND,E	
DISPLAY	49
COMMAND LIST OPERATION	
THE DISPLAY REGISTERS	
DISPLAY COMMANDS	
CHAPTER V - MISCELLANEOUS MONITOR FUNCTIONS	51
CHAPTER VI - THE LOADER	
GENERAL	52
COMMANDS	52
COMMAND FORMAT	52
EXAMPLE	52
ERROR MESSAGES	53
DEFINITIONS	53
LINKING LOADER FORMAT	54
LOADER MAP OUTPUT FORMAT	56
THE RADIX 50 REPRESENTATION OF SYMBOLS	57
RESTRICTIONS FOR ONE PASS ASSEMBLY	57
RESTRICTIONS FOR GLOBAL REQUESTS	58
RESTRICTIONS FOR COMMON	58
CHAPTER VII - THE EDITOR	
GENERAL	59
USAGE	59
COMMANDS	59

ERROR MESSAGES	60
RESTRICTIONS	61
CHAPTER VIII - THE PERIPHERAL INTERCHANGE PROGRAM (PIP)	
GENERAL	62
COMMAND FORM	62
ERROR MESSAGES	63
EXAMPLES	64
CHAPTER IX - THE BATCH CONTROL PROGRAM (BCP)	
67	
APPENDICES	
CUSPS COMMAND SUMMARY	
MONITOR COMPONENTS SUMMARIES	
CORE ALLOCATION OF MONITOR	71
TABLE OF DEVICE PROGRAM CHARACTERISTICS	73
TABLE OF PHYSICAL DEVICE CHARACTERISTICS	72
SYSTEMS LIBRARY FILE CONTENTS	74
OPERATING THE MULTIPROGRAMMING MONITOR	75
LOADING AND STARTING	
MONITOR ERROR HALTS	
ERROR RECOVERY PROCEDURE	
OPDEF LISTING	77
JOBDEF LISTING	78
SYSPAR LISTING	83
FIGURES	
MULTI-PROGRAMMING SYSTEM DIAGRAM	
MULTI-PROGRAMMING SYSTEM STORAGE (SIMPLIFIED)	
JOB AREA STORAGE (SIMPLIFIED)	

CONSOLE STATE DIAGRAM

COMMON USER SERVICE PROGRAM (CUSP)

JOB AREA STORAGE

SYSTEM DIAGRAM

I/O SERVICE BLOCK DIAGRAM (SIMPLIFIED)

I/O SERVICE BLOCK DIAGRAM

CHAPTER I

INTRODUCTION TO THE PDP-6 MULTIPROGRAMMING TIME SHARING SYSTEM

THE PROGRAMMED DATA PROCESSOR 6 TIME SHARING SYSTEMS ARE DESIGNED TO PROVIDE INTEGRATED SOFTWARE AND HARDWARE FOR THE SIMULTANEOUS USE OF A SINGLE SYSTEM BY MANY USERS AT CONSOLES. A CONSOLE CONSISTS OF A TELETYPEWRITER WHICH IS USED TO COMMUNICATE WITH THE SYSTEM. A USER MAY ALSO HAVE REQUESTED FACILITIES OR PROGRAMS. USER'S FILES OR DATA ARE KEPT SEPARATELY ON A DECTAPE FOR EACH USER, A SEGMENT OF THE DRUM OR DISC. THESE FILES ARE REQUIRED WHEN PERMANENT STORAGE OF RESULTS IS DESIRED.

A USER MAY HAVE ONE OR MORE ACTIVE JOBS. A JOB IS CONSOLE INITIATED AND RUNS IN AN ASSIGNED AND PROTECTED MEMORY AREA.

DESIGN OBJECTIVES

THE BASIC SYSTEM OBJECTIVES ARE TO PROVIDE A SIMPLE, MODULAR, TIME SHARING SYSTEM.

CONSOLE USE OF THE SYSTEM FOR PROGRAMMING

A USER MAY CREATE AND EDIT A PROGRAM (OR TEXT), TRANSLATE THE PROGRAM TO MACHINE LANGUAGE (USING FORTRAN, MACRO, OR SOME OTHER) IN THE FORM OF RELOCATABLE LINKING BINARY FOR THE LINKING LOADER. LOAD AND RUN THE PROGRAM, OR IF NECESSARY DIRECTLY COMMUNICATE WITH THE PROGRAM THROUGH A DEBUGGING LANGUAGE.

CONSOLE USE OF THE COMMON USER SERVICE PROGRAMS

THE ABILITY OF A TIME SHARED SYSTEM IS NEARLY OPEN ENDED SINCE FACILITIES ARE IMMEDIATELY AVAILABLE TO ALL USERS WHEN A COMMON USER SERVICE PROGRAM (CUSP) IS ADDED. THIS CLASS OF PROGRAMS INCLUDES FILE MANIPULATION, EDITING, SPECIAL DESK CALCULATING, SIMULATION, AND USER PROGRAMS (UP), OF GENERAL INTEREST.

CONSOLE USE OF THE COMMON USER SERVICE PROGRAMS FOR DATA CONVERSION

THE PERIPHERAL INTERCHANGE PROGRAMS ACCOMPLISH GENERAL DATA CONVERSION FROM MEDIA TO MEDIA, EG. CARD TO TAPE, TAPE TO PRINTER, ETC.

UNATTENDED STACK OR BATCH PROCESSING OF JOBS

A STACK IS RUN AS THOUGH IT WERE CONSOLE CONTROLLED. JOBS ARE STACKED IN A CARD READER OR OTHER INPUT DEVICE AND PROCESSED IN SEQUENCE. THE CONTROL LANGUAGE (CONTROL STATEMENTS OR CARDS) IS IDENTICAL TO THAT OF A CONSOLE USER, AND THE CONTROL STATEMENTS ARE PLACED WITHIN A SPECIAL FILE.

OPEN-ENDED SYSTEM WHICH CAN SERVE SPECIAL CONSOLES

THE EXECUTIVE AND IO SUBROUTINES MAY BE MODIFIED SO THAT CONSOLES REQUIRING A SPECIAL SERVICE OR LANGUAGE MAY BE INCLUDED.

OPEN-ENDED SYSTEM WHICH CAN SERVE REAL TIME PROCESSES

THE NECESSARY INPUT/OUTPUT ROUTINES TO CONNECT THE SPECIAL DEVICE WITH THE PROGRAMMING SYSTEM WOULD BE ADDED. A USER OR JOB WOULD ISSUE SYSTEM INPUT/OUTPUT COMMANDS FOR HIS SPECIAL DEVICE IN THE

SAME MANNER AS FOR CONVENTIONAL DEVICES. IN SOME SPECIAL APPLICATIONS, IT MAY BE NECESSARY TO MODIFY THE SCHEDULER PORTION OF THE MONITOR THAT THE OCCURRENCE OF I/O DEVICE CONDITIONS WOULD CAUSE SPECIAL SERVICE FOR THE USER.

COMMON I/O SUB-PROGRAMS

THE I/O ROUTINES PROVIDE A BASIC SET OF DEVICE INDEPENDENT INPUT/OUTPUT COMMANDS WHICH MAY BE GIVEN TO ALL DEVICES. THE SYSTEM CONTROLS THE ASSIGNMENT, INITIALIZATION, AND CLOSING FOR ALL I/O DEVICES, AND THE RENAMING FOR A USER IF NECESSARY. PROGRAMS MAY BE EASILY WRITTEN WHICH COMMUNICATE WITH ANY I/O DEVICE+ ALL I/O PROGRAMS USE THE HARDWARE PRIORITY INTERRUPT SO THAT A DEVICE'S NULL TRANSMISSION DOES NOT REQUIRE THE SYSTEM TO WAIT.

COMMON SYSTEM SUB-PROGRAMS

THESE SYSTEM ROUTINES ARE CALLED FOR SPECIAL SYSTEM FUNCTIONS, SUCH AS TIME, DATE, EXTENSION OF CORE ALLOTTMENT, ETC.

TIME SHARING SYSTEM STAGES

THE PDP-6 TIME SHARING SYSTEMS ARE AVAILABLE IN FOUR CONFIGURATIONS. THESE ARE DISTINGUISHED BY VARIOUS HARDWARE-SOFTWARE CONFIGURATIONS. THESE ALLOW A DIFFERENT BACKUP STORAGE MEDIUM TO STORE PROGRAMS WHILE OTHERS ARE RUNNING, AND THIS DETERMINES THE NUMBER OF SIMULTANEOUS ACTIVE SYSTEM USERS AND THE SYSTEM RESPONSE TIME TO A USER.

SYSTEM I - MULTIPROGRAMMING TIME SHARING

THE MULTI-PROGRAMMING TIME SHARING SYSTEM USES MULTI-PROGRAMMING TO PROVIDE SIMULTANEOUS SERVICE FOR THE USER9 NO BACKUP STORAGE IS USED TO STORE PROGRAMS AND ACTIVE PROGRAMS MUST RESIDE IN CORE MEMORY. DECTAPE IS USED TO RETAIN ALL THE LIBRARY PROGRAMS, AND EACH USER'S INACTIVE PROGRAMS. MULTI-PROGRAMMING MEANS THE SIMULTANEOUS EXISTENCE OF MULTIPLE PROGRAMS IN CORE MEMORY, WITH A CONTROL MECHANISM TO SWITCH FROM PROGRAM TO PROGRAM AS THE ACTIVE PROGRAMS COME TO AN INPUT/OUTPUT WAITING CONDITION, OR COMPLETE THEIR ALLOTTED SHARE OF COMPUTATION TIME.

SYSTEM II - DRUM TIME SHARING

THE DRUM TIME SHARING SYSTEM USES THE PDP-6 ONE MILLION WORD DRUM TO PROVIDE THE BACKUP STORAGE FOR MULTIPLE USERS. BOTH MULTI-PROGRAMMING AND PROGRAM SWAPPING TECHNIQUES FOR TIME SHARING CONTROL ARE USED TO PROVIDE SERVICE TO MANY ACTIVE USERS. WITH PROGRAM SWAPPING, A PROGRAM IS BROUGHT INTO CORE MEMORY AND RUN WHEN ITS TURN FOR COMPUTATION IS AVAILABLE8 OR INPUT/OUTPUT ASSOCIATED WITH IT IS COMPLETE, AND THE PROGRAM IS NOT IN CORE. A PROGRAM IS RETURNED TO THE DRUM WHEN IT MUST WAIT FOR INPUT/OUTPUT, OR HAS FINISHED ITS SHARE OF COMPUTING, AND SPACE IN CORE IS REQUIRED. AS IN SYSTEM I, DECTAPE IS USED TO STORE USER'S INACTIVE PROGRAMS, AND A USER BEGINS BY FIRST READING HIS DECTAPE FILES UNTO THE DRUM. AT THE COMPLETION OF CONSOLE USE, THE USER'S FILES ON THE DRUM MAY BE COPIED TO A DECTAPE, AND WITH THIS MODE OF OPERATION, AUTOMATIC DAY BY DAY BACKUP IS PROVIDED.

SYSTEM III - DRUM-DISC TIME SHARING

THE DRUM-DISC TIME SHARING SYSTEM REQUIRES THE ADDITION OF A DISCFILE WITH AT LEAST FIVE MILLION WORDS OF STORAGE, AND A TYPE 516 MAGNETIC TAPE SYSTEM. THE DISCFILE IS USED AS A LONG TERM STORAGE FOR INACTIVE FILES OR PROGRAMS. WITH THIS SYSTEM A USER MAY REQUEST A DUPLICATE OF A FILE ON THE DISC TO BE PLACED

ON THE DRUM, OR HE MAY WORK WITH FILES DIRECTLY ON EITHER THE DRUM OR DISC. THE DRUM RETAINS COPIES OF PROGRAMS BEING SWAPPED. THE MAGNETIC TAPE SYSTEM IS REQUIRED FOR DAY TO DAY BACKUP IN CASE OF SYSTEM FAILURES.

SYSTEM IV - MULTI-PROCESSOR TIME SHARING

THE MULTIPROCESSOR SYSTEM GIVES ADDITIONAL COMPUTATION POWER BY ALLOWING TWO ADDITIONAL TYPE 166 ARITHMETIC PROCESSORS TO BE CONNECTED TO THE SYSTEM.

HARDWARE FOR TIME SHARING

EXECUTIVE MODE HARDWARE

WHEN UN-DEBUGGED PROGRAMS SHARE MEMORY WITH OTHER PROGRAMS, PROTECTION MUST BE PROVIDED AGAINST ILLEGAL REFERENCES TO OTHER AREAS OF MEMORY. TO PROVIDE THIS PROTECTION, PROGRAMS PREPARED AND RUN BY THE USER ARE RUN IN THE USER MODE. THEY ARE PLACED IN MEMORY, AND RELOCATED, BY THE RELOCATION REGISTER. MEMORY REFERENCES OUTSIDE THE AREA SPECIFIED BY THE MEMORY PROTECTION REGISTER ARE ILLEGAL. AN ILLEGAL MEMORY REFERENCE, AN ATTEMPT TO STOP THE MACHINE, OR AN I/O INSTRUCTION, CALL THE EXECUTIVE.

LARGE STORAGE

SINCE A NUMBER OF USERS ARE TO OCCUPY MEMORY AT ONCE, A TIME SHARING SYSTEM MUST HAVE A LARGE CORE MEMORY AND/OR THE ABILITY FOR A FAST INTERCHANGE BETWEEN CORE MEMORY AND AUXILIARY FILE STORES, TO PROVIDE MULTI-PROGRAMMING SERVICE.

A PROCESSOR IN A PDP-6 SYSTEM MAY DIRECTLY ADDRESS UP TO 262,144-36 BIT WORDS OF CORE STORAGE IN MEMORY MODULES OF VARYING SIZE AND SPEED. THE POWERFUL INSTRUCTION SET ALSO PROVIDES FOR ADDRESSING THE SAME STORAGE AS CHARACTERS, PROVIDING A STORE OF UP TO 1.6 MILLION CHARACTERS. MAGNETIC DRUMS, DISC, OR DECTAPE CAN BE CONNECTED FOR AUXILIARY FILE STORAGE AND FOR PROGRAM SWAPPING.

DATA REPRESENTATION

TO AVOID UNNECESSARY CONVERSION PROGRAMS ALL DATA IN THE PDP-6 IS REPRESENTED BY THE AMERICAN STANDARDS ASSOCIATION INFORMATION INTERCHANGE [ASCII] 7 BIT CODE. LINE PRINTER AND TELEPRINTER CODES ARE ASCII, AND OTHER CODES ARE CONVERTED TO ASCII AS THEY ENTER AND LEAVE THE SYSTEM.

PRIORITY INTERRUPT FOR IO DEVICES

THE 7 LEVEL PRIORITY INTERRUPT SYSTEM ALLOWS ALL I/O DEVICES OF A SYSTEM TO OPERATE CONCURRENTLY. EACH PRIORITY LEVEL MAY HAVE A NUMBER OF DEVICES CONNECTED TO IT WHICH REQUEST SERVICE. HIGHEST LEVEL CHANNELS GENERALLY CONNECT WITH ONLY ONE MEDIUM SPEED DEVICE FOR DATA TRANSMISSION [E.G. DISC, MAGNETIC TAPE]. LOWER SPEED DEVICES CONNECT TO LOW PRIORITY CHANNELS AND THE EXECUTIVE SYSTEM MAY POLL MULTIPLE DEVICE REQUESTS OCCURRING ON A SINGLE CHANNEL.

TABLE OF MULTIPROGRAMMING SYSTEM HARDWARE

REQUIRED HARDWARE

1-166	ARITHMETIC PROCESSOR WITH 626 TELEPRINTER
1-161 OR 163C	(16,384 MEMORY WORDS)
1-136	DATA CONTROL
1-551	DECTAPE CONTROL
1-555	DUAL DECTAPE (
1-760	PAPER TAPE READER (HARDWARE MAINTENANCE)

OPTIONAL HARDWARE

1-162	16 WORD MEMORY
-------	----------------

OPTIONAL HARDWARE WITH SOFTWARE SUPPORT

N-161 OR 163C	MEMORIES (FORMING UP TO 262,144 WORDS)
3-555	DUAL DECTAPES (1-555/2 USERS)
1-630	DATA COMMUNICATIONS SYSTEM WITH HALF OR FULL DUPLEX 8 LEVEL TELETYPEWRITERS
1-516	MAG TAPE CONTROL WITH 520,521,522 INTERFACE AND UP TO 8 TRANSPORTS.
1-346	DISPLAY
1-461	CARD READER
1-646	LINE PRINTER (300,600,1000 LPM)

SOFTWARE MONITOR COMPONENTS

THE MULTIPROGRAMMING SYSTEM DIAGRAM DEPICTS THE FLOW OF DATA AT THE USER CONSOLE TO EXTERNAL FILES, AND THE SOFTWARE-HARDWARE INTERACTION.

CONSOLES

THE USER'S CONSOLE IS HIS COMMUNICATION WITH THE SYSTEM, AND HIS FILES. ALL COMMANDS TO INITIATE, CARRY-OUT, AND TERMINATE A JOB ARE AT THE CONSOLE. EACH TIME A SYSTEM PROGRAM, OR SERVICE IS REQUESTED, THIS COMMUNICATION IS CARRIED OUT AT THE CONSOLE. THE CONSOLE IS ALSO THE MEDIA THROUGH WHICH A USER REQUESTS PERSONAL PROGRAMS, DATA FILES, ETC. CONTAINED IN THE SYSTEM TO BE MANIPULATED.

THE TELETYPE DATA IS TRANSMITTED ON A STANDARD TELEGRAPH LINE TO THE DATA COMMUNICATIONS EQUIPMENT AT THE COMPUTER. UP TO 64 LINES ENTER A SINGLE DATA COMMUNICATION DEVICE, AND THE CHARACTERS ARE PROCESSED IN SEQUENCE AS THEY ARRIVE. THE CHARACTER IS TRANSMITTED TO A USER'S OWN BUFFER AREA THROUGH THE TELETYPE IO SERVICE ROUTINE.

AS INITIAL CHARACTERS ARE RECEIVED, THE I/O SERVICE ROUTINE FIRST LOOKS TO SEE IF A BUFFER AREA HAS BEEN ASSIGNED TO THE USER STATION CAUSING THE INTERRUPT. IF NO BUFFER HAS BEEN ASSIGNED AND THERE IS ONE AVAILABLE, THE STATION IS ASSIGNED THE OPEN BUFFER BY MEANS OF A WORD POINTER. IF ALL AVAILABLE BUFFERS ARE BEING USED, THE OPERATOR OF THE STATION IS APPROPRIATELY NOTIFIED.

THE NORMAL QUANTUM OF DATA HANDLED IN THE INTERRUPT SERVICE BUFFERS IS A LINE TERMINATED BY A CARRIAGE RETURN (OR UP TO 75 CHARACTERS). THEREFORE, CHARACTERS ARE RECEIVED OR TRANSMITTED UNTIL THE CONSOLE BUFFER IS EMPTY, FULL OR TERMINATED BY AN APPROPRIATE CHARACTER. WHEN A BUFFER BECOMES EMPTY IN OUTPUT MODE, THE I/O SERVICE SEES IF THE USER PROGRAM HAS MORE DATA IT WISHES TO TRANSMIT. THIS IS DETERMINED BY THE USER PROGRAM SPECIFYING EITHER CONTINUOUS OPERATION, IN WHICH ALL FULL BUFFERS ARE SERVICED, OR NON-CONTINUOUS, IN WHICH ONLY ONE BUFFER IS SERVICED.

WHEN A BUFFER BECOMES FULL OR TERMINATED IN INPUT MODE, THE I/O SERVICE TRANSFERS THE BUFFER INTO THE BUFFER AREA OF THE USER PROGRAM THAT REQUESTED THE DATA. IF THE DATA HAS NOT BEEN REQUESTED BY ANY USER PROGRAM, IT IS TRANSFERRED TO THE COMMAND DECODER IN THE MONITOR FOR COMMAND ANALYSIS. THUS, INPUT IS EITHER TO A USER'S OWN PROGRAM, OR A COMMAND TO THE SYSTEM.

EXECUTIVE OR MONITOR STORAGE (EXECUTIVE MODE)

THE TIME SHARING MONITOR IS A COLLECTION OF PROGRAMS THAT ARE PERMANENTLY IN CORE MEMORY DURING NORMAL DATA PROCESSING AND DEBUGGING OPERATIONS AND SUPERVISE THE OPERATION OF THE USER'S PROGRAM. THE TIME SHARING MONITOR PERFORMS SEVERAL FUNCTIONS.

FIRST, IT PERMITS SEVERAL USER'S PROGRAMS TO BE LOADED INTO CORE MEMORY SIMULTANEOUSLY. THE MONITOR MAKES USE OF THE PDP-6 TIME SHARING HARDWARE TO PREVENT EACH USER'S PROGRAM FROM INTERFERING WITH OTHER USER'S PROGRAMS. EACH PROGRAM IS RUN FOR A CERTAIN LENGTH OF TIMES THEN THE MONITOR SWITCHES CONTROL TO ANOTHER PROGRAM IN A ROUND-ROBIN FASHION. SWITCHING IS FREQUENT ENOUGH SO THAT ALL PROGRAMS APPEAR TO RUN SIMULTANEOUSLY.

THE SECOND FUNCTION OF THE TIME SHARING MONITOR IS TO PROCESS INPUT/OUTPUT COMMANDS. ONLY ONE USER AT A TIME IS PERMITTED TO OPERATE EACH PARTICULAR DEVICE. THE INPUT/OUTPUT SERVICE ROUTINES PREPROCESS INPUT/OUTPUT DATA SO THAT ALL DEVICES APPEAR IDENTICAL TO THE USER'S PROGRAM THUS SIMPLIFYING INPUT/OUTPUT CODING. THE MONITOR MAKES USE OF THE PDP-6 PROGRAM INTERRUPT SYSTEM TO OVERLAP INPUT/OUTPUT OPERATIONS WITH COMPUTATION. IF A USER'S PROGRAM MUST WAIT FOR THE COMPLETION OF AN INPUT OR OUTPUT OPERATION, THE MONITOR AUTOMATICALLY SWITCHES TO ANOTHER USER'S PROGRAM.

THE RESULT OF THESE FUNCTIONS IS THAT THE PDP-6 COMPUTER ALWAYS IS PROCESSING DATA, SINCE AS SOON AS ONE PROGRAM HAS NOTHING TO DO, TIME SHARING MONITOR SWITCHES CONTROL TO ANOTHER PROGRAM. TIME SHARING PERMITS THE ECONOMICAL USE OF THE PDP-6 TO PERFORM ON-LINE DEBUGGING OR PROGRAM EDITING.

MULTI-PROGRAMMING EXECUTIVE

THE EXECUTIVE PROVIDES OVERALL CO-ORDINATION AND CONTROL OF THE TOTAL OPERATING SYSTEM. THE MULTI-PROGRAMMING SYSTEM STORAGE DIAGRAM DEPICTS THESE COMPONENTS. IT MUST:

1. ALLOCATE FACILITIES
2. HANDLE ALL COMMANDS ADDRESSED TO THE SYSTEM FROM THE USER CONSOLE
3. CO-ORDINATE I/O BETWEEN THE USER AND THE I/O SERVICE.
4. PROVIDE COMMUNICATION TO A USER'S CONSOLE.

IT SCHEDULES JOBS. ONCE A USER PROGRAM HAS BEEN GIVEN CONTROL, IT CAN BE REMOVED FROM RUN STATUS BY ANY OF THE FOLLOWING MEANS:

1. THE PROGRAM IS COMPLETED AND DOES NOT REQUIRE FURTHER PROCESSING.
2. THE PROGRAM HAS RUN FOR ITS ALLOTTED AMOUNT OF TIME AND MUST BE RE-SCHEDULED.
3. THE PROGRAM HAS ISSUED AN I/O COMMAND AND ISSUES A "WAIT" REQUEST.
4. A PROGRAM OF HIGHER PRIORITY MUST BE RUN.

AS A PROGRAM IS REMOVED FROM RUN STATUS, THE EXECUTIVE MUST TRANSFER CONTROL TO ANOTHER PROGRAM AS QUICKLY AS POSSIBLE.

A DESCRIPTION OF THE EXECUTIVE OR MONITOR COMPONENTS IS GIVEN IN AN APPENDIX.

INPUT-OUTPUT INTERRUPT SERVICE MODULES

EACH ACTIVE IO DEVICE IN THE SYSTEM HAS AN INPUT/OUTPUT SERVICE ROUTINE MODULE.

THE EFFECTIVENESS OF THE TIME SHARING SYSTEM DEPENDS LARGELY ON THE EFFICIENCY OF THE I/O INTERRUPT SERVICE. TO IMPLEMENT IT PROPERLY, TWO HARDWARE FEATURES WERE INCORPORATED INTO THE PDP-6.

1. A MULTIPLE LEVEL PRIORITY INTERRUPT SYSTEM THAT CAN BE PROGRAM ASSIGNED.
2. A FULLY BUFFERED I/O SYSTEM TO ALLOW THE CENTRAL

PROCESSOR TO SERVICE I/O AT A RATE THAT APPROACHES MEMORY SPEEDS.

THE INTERRUPT SERVICE IS COMPLETELY MODULAR TO ALLOW A SMALL SYSTEM TO CARRY ONLY THOSE ROUTINES DIRECTLY RELATED TO ITS PERIPHERAL EQUIPMENT. THUS THE MEMORY CAPACITY OF A SMALL SYSTEM IS NOT OVER-TAXED, YET EXPANSION OF THE SYSTEM IS NOT RESTRICTED. AS MORE I/O IS ADDED, THE PROPER SERVICE ROUTINES ARE ADDED INTO THE SYSTEM AND BUFFER AREAS ARE EXTENDED.

THUS SERVICE IS IMPLEMENTED IN SUCH A WAY THAT THE PROGRAMS ARE ABLE TO RECEIVE AND TRANSMIT A SUFFICIENTLY LARGE QUANTUM OF DATA IN THE ASCII FORMAT, COMMON TO ALL I/O EQUIPMENT. THEREFORE, I/O SERVICE RELIEVES THE PROGRAMMER OF THE BURDEN OF FORMAT TRANSLATIONS AND ALSO INSURES THAT WHEN A USER RECEIVES OR TRANSMITS A QUANTUM OF DATA, IT WILL BE LARGE ENOUGH TO BE USEFUL.

JOB STORAGE (USER MODE)

A SIMPLIFIED DIAGRAM OF JOB AREA STORAGE DEPICTS A JOB OR USER'S AREA OF MEMORY. THE USER AREA COMPOSES THE LARGEST BULK OF MEMORY AND IS UNDER COMPLETE CONTROL OF THE MONITOR. SINCE MANY USER PROGRAMS WILL NOT HAVE BEEN DEBUGGED, SEVERAL SAFEGUARDS ARE USED TO INSURE THERE WILL BE NO UNWANTED USER-USER OR USER-MONITOR INTERACTION, INCLUDING

1. TRYING TO HALT THE PROCESSOR
2. MAKING MEMORY REFERENCES OUTSIDE A MONITOR DEFINED AREA IN MEMORY
3. TRYING TO CONTROL OR INFLUENCE THE I/O OR PRIORITY INTERRUPT SYSTEM OUTSIDE MONITOR CONTROL.

FOR CONVENIENCE AND FLEXIBILITY, HOWEVER, THE USER MUST BE ABLE TO INPUT, OUTPUT, CONTROL, AND SELECT PERIPHERAL EQUIPMENT THROUGH THE MONITOR. THE USER MUST BE ABLE TO LOAD SEVERAL OF HIS PROGRAMS TOGETHER WITH SYMBOLS, HAVING THEM LINKED AND RELOCATED TO FIT PROPERLY INTO HIS ASSIGNED AREA.

CHAPTER III DESCRIBES THE JOB OR USER'S AREA IN DETAIL, BUT THIS AREA IS WHERE A COMMON USER SERVICE PROGRAM (CUSP) OR A USER PROGRAM (UP) IS LOADED AND RUN.

AREA IS WHERE A CUSP OR UP IS LOADED AND RUN.

THE MAJOR AREAS OF A JOB OR USER'S AREA ARE:

1. SYSTEM USAGE - TEMPORARY STORAGE IS REQUIRED WHERE THE SYSTEM IS DOING WORK ON THE BEHALF OF A USER. ALSO MOST SYSTEM DATA REGARDING A JOB IS KEPT WITH THE JOB. THIS TEMPORARY WHICH THE SYSTEM REQUIRES IS LOADED AND/OR UNLOADED AS THE JOB IS STARTED OR STOPPED.
2. PROGRAM - THE PROGRAM ITSELF REMAINS IN THE JOB AREA, INCLUDING ITS FORTRAN COMMON, SYMBOL TABLE, COPY OF DDT, LIBRARY FILES, FUNCTION, ETC.
3. I/O BUFFERS - AS A JOB DOES INPUT/OUTPUT, BUFFERS ARE DYNAMICALLY ASSIGNED IN A JOB'S OWN AREA, AND DATA ARE TRANSMITTED DIRECTLY TO THESE BUFFERS FROM THE IO DEVICE SERVICE PROGRAMS.

INPUT/OUTPUT DATA FILES ORGANIZATION

THIS PORTION OF THE SYSTEM CONTAINS THE FILES OR DATA STORAGE FOR ALL JOBS AND USERS.

A FILE IS A LINEAR STRING OF DATA WORDS OR DATA CHARACTERS. FILES HAVE FILE NAMES WHICH APPEAR IN ASSOCIATED FILE DIRECTORIES, OR FILE NAMES ARE NAMED IMPLICITLY BY THE DEVICE ON WHICH THEY APPEAR.

EACH FILE HAS A DATA MODE WHICH DENOTES THE TYPE OF DATA WITHIN THE FILE.

A FILE NAME EXTENSION IS OFTEN INCLUDED TO FURTHER DISTINGUISH TWO FILES.

FILE DIRECTORY OR LOCATION DESIGNATION

1-3 OR 1-6	CHARACTER DEVICE NAME
1-6	CHARACTER FILE NAME
1-3	CHARACTER FILE NAME EXTENSION
1 CHARACTER	(4 BITS) FILE MODE
FILE ACTIVITY OR STATUS INFORMATION	
POINTER TO DATA	

DATA

DATA STRING (CHARACTERS, WORDS, ETC.)

THE ALLOWABLE FILE NAME FORMATS FROM A CONSOLE ARE DESCRIBED IN CHAPTER II, FILE COMMAND FORMATS.

THE FILE DATA MODE SPECIFICATIONS ARE GIVEN IN CHAPTER IV.

BRIEFLY, THE FILE DATA STRUCTURE IS:

1. DEVICES ARE DENOTED PTR (PAPER TAPE READER), PTP, CDR, DTA0, ..., 7, ETC.
2. FILE NAMES (ARBITRARY)
3. FILE NAME EXTENSIONS TXT (TEXT), MAC, FOR, ETC.
4. FILE DATA MODES
 - A ASCII MODE DATA PACKED 5-7 BIT CHARACTERS, EITHER PACKED IN SMALL PHYSICAL RECORDS (IF TAPE) OR ON A SINGLE PAPER TAPE OR CARD DECK.
 - B BINARY MODE DATA 36 BIT WORDS. USUALLY PACKED IN SMALL PHYSICAL RECORDS.
 - D DUMP MODE, CONTIGUOUS BLOCK OF BINARY WORDS, GENERALLY USED TO STORE A CORE IMAGE, OR PROGRAM, IN A SINGLE FILE.
 - I IMAGE MODE, RECEIVED DIRECTLY FROM DEVICE

SYSTEM LIBRARY FILE

THIS LIBRARY CONTAINS ALL THE ACTIVE PROGRAMS WHICH ARE USED WITHIN THE SYSTEM. A TABLE OF THE LIBRARY, ITS COMPONENT

MNEMONICS, ETC. IS GIVEN IN THE APPENDIX.

CHAPTER I I
CONSOLE USE OF THE SYSTEM

INTRODUCTION

THE CONSOLE COMMANDS GIVE THE SYSTEM USER COMPLETE ACCESS TO ALL SYSTEM FUNCTIONS AND PROGRAMS, AS WELL AS THE ABILITY TO CREATE, EDIT, LOAD, RUN, AND DEBUG HIS OWN PROGRAMS. THE MONITOR MAKES NO DISTINCTION BETWEEN COMMON USER SERVICE PROGRAMS AND USER PROGRAMS. IT HANDLES BOTH KINDS OF PROGRAMS IN THE SAME WAY.

PHYSICAL DECTAPE 1 (DTA1) CONTAINS THE COMPLETE SYSTEM LIBRARY. THE SYSTEM IS OPEN ENDED SINCE A USEFUL USER PROGRAM MAY BECOME A CUSP SIMPLY BY ADDING IT TO THE SYSTEM LIBRARY TAPE. HENCEFORTH, THE TERM "PROGRAM" SHALL MEAN BOTH COMMON USER SERVICE PROGRAMS AND USER PROGRAMS.

THE PHILOSOPHY OF THE MULTIPROGRAMMING TIME SHARING SYSTEM IS THAT USERS MUST COOPERATE WITH EACH OTHER TO A CERTAIN EXTENT. ENOUGH SAFEGUARDS HAVE BEEN IMPLEMENTED TO PREVENT MISTAKES BY ONE USER FROM HARMING A FELLOW USER. THE SYSTEM DOES NOT PREVENT A USER FROM DISTURBING OTHER USERS. FOR EXAMPLE, A USER MAY RESERVE THE LINE PRINTER FOR AN INDEFINITE PERIOD WITHOUT USING IT, OR AN EARLY USER MAY BE GRANTED AN EXCESSIVE AMOUNT OF CORE PREVENTING OTHER USERS FROM USING THE SYSTEM.

CONSOLE STATES

THE CONSOLE STATE DIAGRAM SHOWS THE "STATES" OR "MODES" OF A CONSOLE. EACH LINE INDICATES A COMMAND OR ACTION WHICH WILL CHANGE THE STATE OF THE USER'S PROGRAM, OR HIS JOB. SOME STATE TO STATE TRANSITIONS ARE AUTOMATIC, WHILE OTHERS ARE CONSOLE INITIATED. A CONSOLE IS THUS EITHER CONNECTED TO THE MONITOR, OR TO A PROGRAM RUNNING IN A JOB AREA. A JOB MAY BE EITHER A CUSP (LOADER, EDITOR, TRANSLATOR, ETC) OR A USER PROGRAM (A RUNNING PROGRAM, OR THE DDT ASSOCIATED WITH A USER'S PROGRAM.)

THE MONITOR DOES NOT DIFFERENTIATE BETWEEN A CUSP AND A UP.

MONITOR LEVEL CONTROL

THERE ARE TWO MONITOR CONSOLE STATES, DORMANT, OR MONITOR COMMAND. THE DORMANT STATE EXISTS WHERE NO JOB IS PRESENT, AND THE MONITOR COMMAND STATE INDICATES A JOB IS PRESENT. FROM THIS STATE, A JOB MAY BE GOTTEN FROM A LIBRARY FILE (USING "GET") AND PROGRAMS CAN BE LOADED.

COMMON USER SERVICE PROGRAM LEVEL CONTROL

THE MAJOR STATE ASSOCIATED WITH A CUSP, IS WHEN THE CUSP IS GOTTEN FROM THE SYSTEM LIBRARY USING "GETS". THE CONSOLE THEN COMMUNICATES WITH CUSP. A USER'S PROGRAM IS LOADED USING THE LOADER.

USER PROGRAM LEVER CONTROL

TWO MAJOR STATES EXIST WITH A USER PROGRAM WHICH HAS BEEN LOADED WITH A USER DDT RELOCATABLE BINARY FILE. ONE STATE IS WHEN THE USER PROGRAM IS BEING RUN AT THE USER'S SPECIFIED STARTING ADDRESS, AND THE CONSOLE MAY BE USED BY THE JOB OR IS DISCONNECTED.

THE SECOND STATE IS ENTERED WHEN DDT IS REQUESTED. DDT STATE IS ENTERED BY FIRST ENTERING THE MONITOR COMMAND LEVEL FOLLOWED BY A REQUEST FOR DDT.

MONITOR COMMANDS

A USER MAY RUN MORE THAN ONE PROGRAM OR JOB AT A TIME. A JOB IS IDENTIFIED BY A JOB NUMBER STARTING WITH 1. ALTHOUGH EACH JOB MAY HAVE ONLY ONE PROGRAM RUNNING UNDER IT, THE USER MAY INITIATE MORE THAN ONE JOB. IN ORDER TO COMMUNICATE WITH A PROGRAM, THE CONSOLE MUST BE ATTACHED TO THE APPROPRIATE JOB. INITIALLY A CONSOLE IS IN COMMAND MODE AND IS NOT ATTACHED TO ANY JOB. WHEN THE FIRST COMMAND IS TYPED, THE CONSOLE IS AUTOMATICALLY ATTACHED TO AN UNUSED JOB AND THE JOB IS INITIALIZED (SEE IJOB COMMAND TO SEE WHAT INITIALIZED MEANS). IF THERE ARE NO UNUSED JOBS AVAILABLE THE MONITOR WILL IGNORE THE COMMAND AND TYPE OUT:

N JOB CAPACITY EXCEEDED

WHERE N IS THE JOB CAPACITY WRITING THE NULL JOB.
THIS WILL HAPPEN EACH TIME A COMMAND IS TYPED UNTIL A JOB BECOMES FREE

INITIALLY THE CONSOLE IS IN COMMAND MODE AND EACH INPUT LINE IS INTERPRETED AS A COMMAND TO THE MONITOR. SOME OF THESE COMMANDS INSTRUCT THE MONITOR TO START OR CONTINUE EXECUTION OF A PROGRAM WHICH HAS PREVIOUSLY BEEN LOADED. ALL SUCCEEDING CONSOLE INPUT IS THEN CONSIDERED AS INPUT TO THE PROGRAM UNTIL THE PROGRAM RETURNS CONTROL TO THE MONITOR BY CALLING EXIT, (CALL SIXBIT /EXIT/) THE MONITOR RESPONDS BY TYPING:

EXIT

AND THE CONSOLE IS ONCE AGAIN IN COMMAND MODE. ALL TYPEOUT WILL BE SUSPENDED IF THE USER TYPES IN A CHARACTER WHILE HIS PROGRAM IS TYPING OUT. THE NEXT CHARACTER TYPED IN IS INTERPRETED IN A SPECIAL WAY. IF IT IS <CONTROL> C, THE MONITOR WILL STOP THE JOB, TYPE OUT +C, AND RETURN THE CONSOLE TO COMMAND MODE. IF IT IS <CONTROL>O, THE MONITOR WILL CONTINUE RUNNING THE PROGRAM, TYPE OUT +O AND SUPPRESS THE PRINTING OF THE CURRENT CONTENTS OF THE CONSOLE OUTPUT BUFFER. ANY OTHER CHARACTER WILL CAUSE BOTH THE OUTPUT AND THE EXECUTION TO CONTINUE.

COMMANDS ARE COMPOSED OF ONE OR MORE FIELDS OF ALPHANUMERIC CHARACTERS SEPARATED BY A NON-ALPHANUMERIC CHARACTER AND TERMINATED BY A CARRIAGE RETURN (OR A LINE-FEED, A FORM FEED, OR A VERTICAL TAB).

LEADING SPACES AND TABULATES ARE IGNORED IN ANY FIELD. ONLY THE FIRST 6 CHARACTERS OF THE FIRST FIELD ARE USED FOR THE COMMAND NAME AND THE FIRST 6 CHARACTERS OF EVERY FIELD THEREAFTER AS ARGUMENTS. A FIELD SEPARATION CHARACTER MAY BE SELECTED FOR MNEUMONIC VALUE. BLANK LINES ARE IGNORED.

THE SYSTEM WILL RESPOND IMMEDIATELY WITH A LINEFEED AND WILL TYPE A SECOND CARRIAGE RETURN-LINE FEED, WHEN THE COMMAND IS FINISHED.

NO OTHER ACKNOWLEDGEMENT WILL BE PRINTED UNLESS THERE IS AN ERROR. IF AN UNKNOWN COMMAND IS TYPED, THE MONITOR WILL TYPE THE COMMAND NAME BACK WITH A QUESTION MARK APPENDED. IF A FIELD IS SUPPOSED TO BE ENTIRELY NUMERIC OF A PRESCRIBED RADIX AND AN ILLEGAL ALPHANUMERIC CHARACTER OCCURS, THE MONITOR WILL TYPE OUT THE ENTIRE LINE UP TO AND INCLUDING THE ERRONEOUS CHARACTER WITH A QUESTION MARK APPENDED.

AS A GENERAL RULE ALL NUMERIC ARGUMENTS ARE DECIMAL EXCEPT WHERE OCTAL IS MORE CONVENIENT TO THE USER. EXTRA ARGUMENTS TO COMMANDS WILL BE IGNORED AND NO MESSAGE PRINTED. IF A COMMAND IS TYPED WITH AN INSUFFICIENT NUMBER OF ARGUMENTS IT WILL CAUSE THE MONITOR TO RESPOND WITH:

NOT ENOUGH ARGUMENTS

AND TO IGNORE THE COMMAND.

THE FOLLOWING USER COMMANDS ARE INDICATED BY QUOTES, ". IF ERRORS MAY OCCUR WHEN THE COMMANDS ARE GIVEN, THEY ARE INDICATED. THE COMMANDS ARE:

"IJOB" INITIALIZE JOB COMMAND
INITIALIZES THE JOB TO WHICH THE CONSOLE IS ATTACHED. THIS MEANS THE JOB HAS NO CORE ASSIGNED TO IT (SEE "CORE" COMMAND), NO DEVICES ASSIGNED TO IT EXCEPT THE TELETYPE. TTY IS SET AS THE ONLY LOGICAL/PHYSICAL DEVICE EQUIVALENCE (SEE "ASSIGN" COMMAND).

"PJOB" PRINT JOB NUMBER COMMAND
PRINTS THE JOB NUMBER TO WHICH THE CONSOLE IS CURRENTLY ATTACHED.

"KJOB" KILL THE JOB COMMAND
KILLS THE JOB AND RETURNS TO THE SYSTEM ALL RESOURCES ASSIGNED TO IT. THE CONSOLE IS ALSO DETACHED (SEE "DETACH" COMMAND) AND THE JOB IS THEN AVAILABLE FOR SOME OTHER USER. THIS COMMAND SHOULD BE TYPED WHENEVER A USER IS THROUGH WITH A JOB, IN ORDER TO ALLOW SOMEONE ELSE TO USE THE JOB NUMBER.

"CORE N" GET CORE FOR JOB COMMAND
SETS THE TOTAL NUMBER OF 1024 WORD (DECIMAL) BLOCKS OF CORE TO N (DECIMAL) FOR THE JOB TO WHICH THE CONSOLE IS CURRENTLY ATTACHED. IF THE JOB ALREADY HAS CORE ASSIGNED, IT IS EXTENDED (IF POSSIBLE) OR CONTRACTED TO SATISFY THE REQUEST. IF NO CORE IS ASSIGNED, THE MONITOR SEARCHES THE UNUSED BLOCKS OF CORE AND ASSIGNS N (IF POSSIBLE). IN EITHER CASE, IF THERE IS NOT ENOUGH FREE CORE

TO SATISFY THE REQUEST, THE MONITOR WILL ASSIGN AS MUCH AS POSSIBLE AND TYPE :

K BLOCKS ASSIGNED

JOB'S WILL NOT BE SHUFFLED IN MEMORY TO TRY TO SATISFY A PARTICULAR REQUEST. ALSO ANY NEW CORE ASSIGNED WILL NOT BE CLEARED. THE COMMAND "CORE 0" WILL RETURN ALL OF A JOB'S CORE TO THE SYSTEM. IF A USER ATTEMPTS TO EXTEND HIS CORE ASSIGNMENT TO K BLOCKS AND FINDS THAT THERE IS NOT ENOUGH ROOM, HE MAY SAVE HIS CORE IMAGE (SEE "SAVE" COMMAND), RETURN ALL OF HIS CORE TO THE SYSTEM ("CORE 0"), REQUEST K BLOCKS ("CORE K"), AND RESTORE HIS CORE IMAGE (SEE "GET" COMMAND). IT IS UP TO EACH COMPUTER INSTALLATION TO DEVELOP GROUND RULES FOR USE OF "CORE".

"GET DEVICE:FILE" GET A FILE COMMAND
GETS A PREVIOUSLY SAVED PROGRAM FROM DEVICE "DEVICE" AND PLACES IT IN CORE, WHERE THE FILE NAME OF THE PROGRAM IS FILE DMP. THIS IS THE MECHANISM FOR LOADING SYSTEM PROGRAMS SUCH AS THE ASSEMBLER. HOWEVER, IT IS ALSO USEFUL FOR LOADING USER PROGRAMS WHICH HAVE BEEN PREVIOUSLY SAVED BY THE "SAVE" COMMAND. THE PROGRAM IS NOT STARTED.

ERROR MESSAGES:

NO CORE ASSIGNED
DEVICE NOT AVAILABLE
FILE NOT FOUND
NOT ENOUGH CORE ASSIGNED
DEVICE ERROR
DATA ERROR

"SAVE DEVICE:FILE" SAVE THE JOB COMMAND
SAVES THE CORE IMAGE ON DEVICE "DEVICE" AND GIVES IT A FILE NAME OF FILE.DMP. ALL DEVICES ARE RELEASED AND THE PROGRAM COUNTER IS SET TO THE PROGRAM'S STARTING ADDRESS BEFORE THE FILE IS WRITTEN.
ERROR MESSAGES:

NO CORE ASSIGNED
DEVICE NOT AVAILABLE
DIRECTORY FULL
DEVICE ERROR
DATA ERROR

"START LOC" START AT LOC COMMANDS
STARTS EXECUTION OF THE PROGRAM AT RELATIVE OCTAL LOCATION "LOC" IN THE JOB AREA OR AT THE PROGRAM STARTING ADDRESS IF "LOC" IS NOT SPECIFIED. NO CHECK IS MADE TO SEE IF A PROGRAM HAS BEEN LOADED. ALL SUCCEEDING INPUT WILL BE DIRECTED TO THE USER'S PROGRAM. THIS COMMAND WILL BE OPERATIVE EVEN AFTER A SYSTEM ERROR MESSAGE HAS BEEN PRINTED OR "EXIT" CALLED WHILE EXECUTING

THE PROGRAM.
ERROR MESSAGES:

NO CORE ASSIGNED

"CONT" CONTINUE JOB COMMAND
CONTINUOUS EXECUTION FROM WHEREVER THE PROGRAM WAS STOPPED BY
<CONTROL>C. ALL SUCCEEDING INPUT WILL BE DIRECTED TO THE USER'S
PROGRAM. THIS COMMAND WILL NOT BE OPERATIVE AFTER A MONITOR
DETECTED ERROR MESSAGE HAS BEEN PRINTED OR "EXIT" CALLED WHILE
EXECUTING THE PROGRAM.
ERROR MESSAGES:

NO CORE ASSIGNED

"DDT" START USER DDT COMMAND
STARTS EXECUTION OF USER DDT IN THE JOB AREA. ALL SUCCEEDING
INPUT WILL BE DIRECTED TO "DDT" AND THE USER'S PROGRAM. THIS
COMMAND WILL BE OPERATIVE EVEN AFTER A SYSTEM ERROR MESSAGE HAS
BEEN PRINTED OR "EXIT" CALLED.
ERROR MESSAGES:

NO CORE ASSIGNED
NO DDT LOADED

"STARTM LOC" START JOB, MONITOR, COMMAND
WORKS EXACTLY LIKE "STARTM LOC" EXCEPT THE CONSOLE REMAINS IN
MONITOR MODE. THIS COMMAND IN CONJUNCTION WITH THE "DETACH" AND
"ATTACH" COMMANDS ALLOW THE USER TO EXECUTE MORE THAN ONE
PROGRAM AT THE SAME TIME.
ERROR MESSAGES:

NO CORE ASSIGNED

"CONTM" CONTINUE JOB, MONITOR, COMMAND
WORKS LIKE "CONT" EXCEPT THE CONSOLE REMAINS IN MONITOR MODE.
ERROR MESSAGES:

NO CORE ASSIGNED
CAN'T CONTINUE

"DETACH" CONSOLE FROM JOB COMMAND
DETACHES THE CONSOLE FROM THE JOB TO WHICH IT IS ATTACHED. IF THE
JOB IS STILL RUNNING IT WILL CONTINUE TO RUN. THIS LEAVES THE
CONSOLE IN THE STATE IT WAS IN BEFORE THE FIRST COMMAND WAS TYPED.

THE NEXT COMMAND TYPED-IN WILL AUTOMATICALLY ATTACH THE CONSOLE TO AN UNUSED JOB.

"ATTACH N" CONSOLE TO JOB COMMAND
DETACHES THE CONSOLE FROM THE JOB TO WHICH IT IS CURRENTLY ATTACHED AND ATTACHES IT TO JOB N. THE LOGICAL-PHYSICAL DEVICE EQUIVALENCE IS SETUP SO THAT A PROGRAM MAY COMMUNICATE WITH THE CONSOLE BY REFERRING TO IT SIMPLY AS TTY. (SEE "ASSIGN" COMMAND). SINCE THE CONSOLE IS IMPLICITLY ATTACHED TO A JOB WHENEVER THE FIRST COMMAND IS TYPED (INCLUDING "ATTACH"), "ATTACH" SHOULD NOT BE THE FIRST COMMAND TYPED. IN FACT, THIS COMMAND SHOULD BE USED ONLY TO RECONNECT THE CONSOLE TO A JOB WHICH HAS PREVIOUSLY BEEN "DETACH"ED.
ERROR MESSAGE:

ANOTHER CONSOLE ALREADY ATTACHED

IN THIS CASE THE CONSOLE REMAINS ATTACHED TO THE ORIGINAL JOB.

"ASSIGN DEV:NAME" TO JOB COMMAND
ASSIGNS PHYSICAL DEVICE DEV TO THE JOB. NO OTHER JOB MAY USE A DEVICE ONCE IT HAS BEEN ASSIGNED.

IF DEV IS THE FIRST 3 CHARACTERS OF ONE OF THE MULTIPLE DEVICES (TTY, DTA, MTA) THE MONITOR WILL SEARCH FOR A FREE DEVICE. IN ANY CASE, THE MONITOR WILL INDICATE THE DEVICE ASSIGN BY TYPING:

DEVICE DEVN ASSIGNED

ANY NUMBER OF DEVICES MAY BE ASSIGNED BY REPEATED USES OF ASSIGN. THIS COMMAND ALLOWS A USER TO PROTECT HIS PRIVATE DECTAPES FROM BEING REFERENCED BY ANOTHER USER. IT ALSO ALLOWS A USER AT A REMOTE CONSOLE TO TEST THE AVAILABILITY OF A PARTICULAR DEVICE. THE ASSIGN COMMAND DOES NOT HAVE TO BE TYPED BEFORE A PROGRAM CAN USE THE DEVICE AND IT MAY BE TYPED AFTER THE PROGRAM HAS BEGUN TO USE THE DEVICE.
ERROR MESSAGE:

DEVICE NOT AVAILABLE

IF THE SECOND ARGUMENT IS PRESENT, THE MONITOR WILL NAME "NAME" EQUIVALENT TO DEVICE "DEV" WHERE "NAME" MAY BE ANY ALPHA-NUMERIC STRING (FIRST SIX CHARACTERS USED). PHYSICAL DEVICES NAMES ARE ALL THREE LETTERS LONG SUFFIXED BY A NUMBER IF THE SYSTEM HAS MORE THAN ONE OF THE DEVICES. THE POSSIBLE PHYSICAL DEVICES ARE

WHEN A JOB IS INITIALIZED AN EQUIVALENCE IS SET UP BETWEEN TTY AND TTYN WHEN TTYN IS THE PHYSICAL DEVICE NAME FOR THE CONSOLE ATTACHED TO THE JOB. THUS PROGRAMS MAY REFERENCE THE ATTACHED CONSOLE BY REFERRING TO THE LOGICAL DEVICE NAME TTY.

A TYPICAL USE FOR THIS COMMAND WOULD BE WHERE A PROGRAM IS KNOWN TO REFERENCE A SPECIFIC DEVICE WHICH A PREVIOUS "ASSIGN" COMMAND HAS REVEALED IS NOT AVAILABLE.

THIS COMMAND ALLOWS USER PROGRAMS TO BE WRITTEN WITHOUT KNOWING IN ADVANCE WHICH PHYSICAL TAPE DRIVE A TAPE IS TO BE MOUNTED. IT IS RECOMMENDED NEVER TO REFER DIRECTLY TO A MULTIPLE DEVICE BY ITS PHYSICAL NAME. INSTEAD THE FIRST 3 LETTERS OF THE DEVICE BY A LETTER (STARTING WITH "A" E.G). THUS IF A USER PROGRAM REQUIRES TWO DECTAPES THEY WOULD BE DTA A AND DTAB. USING THE APPROPRIATE "ASSIGN" COMMAND, A PROGRAM MAY ALWAYS BE RUN, AND NEED NOT WAIT UNTIL PHYSICAL DEVICES ARE FREE, OR FOR CERTAIN ASSIGNMENT CONVENTIONS FOR PHYSICAL DEVICES.

IN A SYSTEM WHERE REMOTE CONSOLES ARE USED, THE SEARCH FEATURE OF THE "ASSIGN" COMMAND ALLOWS THE USER TO ASSIGN A FREE DECTAPE OR MAGTAPE BEFORE REQUESTING THE OPERATOR TO MOUNT A PARTICULAR TAPE ON THE ASSIGNED PHYSICAL DEVICE.

THIS COMMAND IS NOT REQUIRED FOR ANY OF THE SYSTEM PROGRAMS SINCE THE USER TYPES IN THE DEVICES TO BE USED DIRECTLY TO THE PROGRAM. USER PROGRAMS WHICH ARE CANDIDATES TO BECOME SYSTEM PROGRAMS SHOULD ALSO BE WRITTEN TO ACCEPT THE DEVICE NAME FROM THE CONSOLE.

"DEASSIGN DEV" FROM JOB COMMAND
DEASSIGNS DEVICE "DEV". THIS MAY BE DONE EVEN THOUGH THE PROGRAM IS CONTINUING TO USE THE DEVICE. "DEVICE" MAY BE EITHER A PHYSICAL OR LOGICAL DEVICE. A SEARCH IS FIRST MADE FOR A LOGICAL DEVICE - FOLLOWED BY A SEARCH FOR A PHYSICAL DEVICE.

MONITOR CONSOLE ERROR MESSAGES

THE CONSOLE WILL ALSO BE RETURNED TO COMMAND MODE WHENEVER THE MONITOR DETECTS AN ERROR. TO HELP DISTINGUISH BETWEEN MONITOR DETECTED ERRORS AND PROGRAM DETECTED MESSAGES, THE MONITOR WILL TYPE:

MONITOR DETECTED ERROR
ERROR IN JOB N

BEFORE TYPING THE ERROR MESSAGE.

THE MONITOR ERROR MESSAGES:

MONITOR ERROR AT N; MONITOR CALLED FROM USER/MONITOR LOC M

WHERE N IS THE LOCATION OF THE ERROR WITHIN THE MONITOR AND M IS THE LOCATION IN THE USER PROGRAM FROM WHICH THE MONITOR WAS CALLED.

OUTPUT DEVICE, DEV, CANNOT DO INPUT AT USER LOC

WHERE DEV IS THE DEVICE REQUESTED AND M IS THE LOCATION IN THE JOB AREA OF THE ILLEGAL INPUT PROGRAMMED OPERATOR.

INPUT DEVICE, DEV, CANNOT DO OUTPUT; AT USER LOC M

WHERE DEV IS THE DEVICE REQUESTED AND M IS THE LOCATION IN THE JOB AREA OF THE ILLEGAL OUTPUT PROGRAMMED OPERATOR.

ILL INST AT USER LOC M

WHERE M IS THE LOCATION IN THE JOB AREA OF THE ILLEGAL INSTRUCTION. A ZERO OPCODE IS CONSIDERED TO BE AN ILLEGAL INSTRUCTION.

ILL DEVICE DATA MODE AT USER LOC M

WHERE M IS THE LOCATION IN THE JOB AREA OF THE INIT PROGRAMMED
OPERATOR SPECIFYING AN ILLEGAL DEVICE DATA MODE.

IO TO UNASSIGNED CHANNEL AT USER LOC M

WHERE M IS THE LOCATION IN THE JOB DATA AREA OF A PROGRAMMED OPERATOR
REFERRING TO A USER CHANNEL WHICH HAS NOT BEEN ASSIGNED BY A PREVIOUS
INIT PROGRAMMED OPERATOR.

ILLEGAL PRO OPE USED AT USER/MONITOR LOC M

WHERE M IS THE LOCATION IN THE JOB AREA OF THE ILLEGAL PROGRAMMED
OPERATOR. THE SYSTEM USES THE PROGRAMMED OPERATORS FROM 40 TO 77
LEAVING 1-37 FOR THE USER'S OWN USE. IF A PROGRAM EXECUTES A PRO-
GRAMMED OPERATOR BETWEEN 1 AND 37 AND LOCATION 41 IS NOT SET UP WITH
A JSR INSTRUCTION (SEE PDP-6 INSTRUCTION MANUAL F-65), THIS ERROR
MESSAGE WILL OCCUR. OTHERWISE THE MONITOR WILL SIMULATE THE EXECUTION
OF THE PROGRAMMED OPERATOR INCLUDING THE JSR INSTRUCTION IN
LOCATION 41 IN THE JOB AREA.

ADDRESS CHECK FOR DEVICE DEVICE; MONITOR CALLED FROM USER LOC M

WHERE DEVICE IS THE DEVICE FOR WHICH THE MONITOR ATTEMPTED TO MAKE A
REFERENCE OUTSIDE THE JOB AREA AND M IS WHERE THE MONITOR WAS CALLED.

ADDRESS CHECK FOR DEVICE, DEV DURING INTERRUPT SERVICE

WHERE DEV IS THE DEVICE FOR WHICH THE INTERRUPT I/O SERVICE ROUTINE
ATTEMPTED TO MAKE A REFERENCE OUTSIDE OF THE JOB AREA.

ILL MEM REF FROM MONITOR/USER LOC M

WHERE M IS THE LOCATION OF THE INSTRUCTION WHICH ATTEMPTED AN ILLEGAL
MEMORY REFERENCE OUTSIDE ITS PROTECTED AREA.

PC EXCEEDS MEM BOUND AT MONITOR/USER LOC M

WHERE M IS THE LOCATION WHICH THE PROGRAM ATTEMPTS TO EXECUTE.

NON EX MEM AT MONITOR/USER LOC M

WHERE M IS THE LOCATION OF AN INSTRUCTION ATTEMPTING TO REFERENCE
NON-EXISTENT MEMORY.

PDL OV AT MONITOR/USER LOC M

WHERE M IS THE LOCATION OF THE INSTRUCTION CAUSING THE PUSH DOWN
OVERFLOW.

ILL BLOCK NO BUFFER AT USER LOC M

WHERE M IS THE LOCATION IN THE USER AREA OF THE BUFFER WHICH CONTAINS
THE ILLEGAL BLOCK NUMBER (GREATER THAN 1100 OCTAL).

GENERAL USES OF COMMON USER SERVICE PROGRAMS

CUSP COMMAND FORMATS

THE GENERAL CONSOLE INPUT COMMAND STRING IS:

DEST FILE 1, DEST FILE 2 ← SOURCE FILE 1,...SOURCE FILE N

THE COMMAND STRING CONSISTS OF ONE OR MORE DESTINATION FILE DESCRIPTORS SEPARATED FROM ONE OR MORE SOURCE FILE DESCRIPTORS BY AN "←", WHERE THE ALLOWABLE COMPONENTS OF A FILE DESCRIPTOR ARE A DEVICE NAME, A FILE NAME, A FILE NAME EXTENSION, THE MODE, AND SWITCHES, WHICH APPLY FOR THE FILE DESCRIPTOR CONTAINING THE SWITCHES PLUS ALL DESCRIPTORS TO THE RIGHT, SEPARATED FROM EACH OTHER BY SPECIAL DELIMITERS WHICH IDENTIFY THEM.

FILE DELIMITERS

() ENCLOSE PROGRAM SWITCHES.
: FOLLOWS A DEVICE NAME.
, FOLLOWS A FILE DESCRIPTOR.
/ PRECEDES A MODE.
. PRECEDES A FILE NAME EXTENSION
← SEPARATES DESTINATION FILES FROM SOURCE FILES
(CR) DELIMITS A FILE DESCRIPTOR

DEVICE NAMES

LPT LINE PRINTER
CDR CARD READER
PTR PAPER TAPE READER
PTP PAPER TAPE PUNCH
TTY0...N TELETYPE
DTA0...N DECTAPE
MTA0...N MAGNETIC TAPE
DIS DISPLAY

FILE NAME EXTENSIONS (NEED NOT BE SPECIFIED)

MAC MACRO SOURCE
FOL FOLA SOURCE
FOR FORTRAN SOURCE
LIB LIBRARY
REL RELOCATABLE BINARY FILE

DMP	DUMP FILE
LST	LISTING
MAP	STORAGE MAP

FILE MODES

A	ASCII
B	CHECKSUM BINARY
S	ASCII SEQUENCED
L	ASCII LINE
D	DUMP FILE

CHAPTER III
JOB AREA USE OF SYSTEM

GENERAL

THE USER'S PROGRAM AREA IS THE PORTION OF CORE MEMORY AVAILABLE TO THE USER WHEN HIS PROGRAM IS RUNNING. THE FIRST ADDRESS OF THIS AREA IS DEFINED BY THE CONTENTS OF THE MEMORY RELOCATION REGISTER WHEN THE USER'S PROGRAM RUNS, AND THE LAST ADDRESS IS DEFINED BY THE CONTENTS OF THE MEMORY PROTECTION REGISTER. THE USER'S AREA IS SHOWN IN THE DIAGRAM, JOB AREA STORAGE

IN ORDER TO MAKE THE FOLLOWING DISCUSSION CLEARER, IT IS NECESSARY TO DEFINE SOME TERMS AND TO REVIEW BRIEFLY THE OPERATION OF THE PDP-6 TIME SHARING HARDWARE. THE USER'S PROGRAM RUNS WHILE THE PDP-6 IS IN A SPECIAL MODE CALLED "USER'S MODE". IN THIS MODE, THE CONTENTS OF A REGISTER IN THE ARITHMETIC PROCESSOR CALLED THE MEMORY RELOCATION REGISTER IS ADDED TO EACH MEMORY ADDRESS BEFORE THE ADDRESS IS SENT TO THE MEMORY SYSTEM. THE ADDRESS BEFORE THIS ADDITION TAKES PLACE IS CALLED THE RELATIVE ADDRESS, AFTER THE ADDITION THE ADDRESS IS CALLED THE ABSOLUTE ADDRESS. ALL ADDRESSES IN THE USER'S AREA ARE RELATIVE ADDRESSES.

TO TAKE ADVANTAGE OF THE FAST ACCUMULATORS, ACCUMULATOR ADDRESSES (0 THROUGH 17) ARE NOT RELOCATED. THUS, RELATIVE LOCATIONS 0 THROUGH 17 CANNOT BE REFERENCED BY THE USER'S PROGRAM. (THE TIME SHARING MONITOR SAVES THE USER'S ACCUMULATORS IN THIS AREA WHILE THE SYSTEM IS SERVICING A PROGRAMMED OPERATOR REQUEST.) THE CONTENTS OF THE MEMORY PROTECTION REGISTER ARE COMPARED WITH THE EIGHT HIGH-ORDER BITS OF EACH RELATIVE ADDRESS AS THE USER'S PROGRAM RUNS. IF THE RELATIVE ADDRESS EXCEEDS THE CONTENTS OF THE MEMORY PROTECTION REGISTER, THE MEMORY VIOLATION FLAG IS SET AND CONTROL TRAPS TO THE TIME SHARING MONITOR. SOME INSTRUCTIONS TRAP TO ABSOLUTE LOCATION 40 WHEN EXECUTED IN THE USER MODE. THESE INSTRUCTIONS, JRST INSTRUCTIONS WHICH ATTEMPT TO HALT, AND ALL UNDEFINED OPERATION CODES. THE ACTION ON TRAPS TO LOCATION 40 IS THE SAME AS THE ACTION FOR PROGRAMMED OPERATORS IN EXECUTIVE MODE, WITH THE ADDITION THAT THE USER MODE IS TURNED OFF WHEN THE TRAP OCCURS. PROGRAM INTERRUPTS ALSO TRAP TO THE ABSOLUTE LOCATIONS NORMALLY USED IN EXECUTIVE MODE AND TURN OFF THE USER MODE.

SYSTEM PARAMETERS IN JOB AREA

THE FIRST 140 (OCTAL) REGISTERS OF THE USER JOB AREA COMPRISE THE JOB DATA AREA WHICH IS RESERVED FOR STORING SPECIFIC INFORMATION CONCERNING THE JOB, SUCH AS THE STARTING ADDRESS OF THE USER'S PROGRAM AND THE IN/OUT DEVICE ASSIGNMENTS MADE FOR THIS PROGRAM. ALSO IN THIS AREA ARE REGISTERS FOR SAVING ACCUMULATORS DURING REQUESTS FOR SERVICE TO THE EXECUTIVE SYSTEM USING THE PROGRAMMED OPERATORS OR WHILE SOME OTHER USER'S PROGRAM IS RUNNING. AN UNDERSTANDING OF THE FUNCTIONS OF THE SPECIFICALLY RESERVED LOCATIONS IS REQUIRED FOR EFFICIENT USE OF THE SERVICES OF THE TIME SHARING MONITOR

THESE LOCATIONS ARE RESTORED TO THE ORIGINAL VALUES WHENEVER THE USER'S PROGRAM BECOMES INACTIVE (DUE TO WAITING FOR INPUT/OUTPUT OR TO RUNNING ANOTHER USER'S PROGRAM). THESE REGISTERS ARE LABELED "PROTECTED" IN THE DESCRIPTIONS BELOW. THE REGISTERS MNEMONIC ASSIGNMENTS ARE MADE ON THE FILE , JOBDEF, AND JOBDEF SHOULD BE EITHER ASSEMBLED WITH PROGRAMS WHICH REFERENCE THIS AREA, OR EXTERNALS SHOULD BE USED TO NAME THE REGISTERS. A LISTING OF JOBDEF, AND THE ASSIGNMENTS OF UUOS TO SYSTEM FUNCTIONS, OPDEF, IS GIVEN IN THE APPENDIX.

JOBAC

THE BLOCK OF CORE MEMORY FROM USER'S LOCATION 0 THROUGH 17 CANNOT BE REFERENCED BY THE USER'S PROGRAM BECAUSE THE ARITHMETIC PROCESSOR SUBSTITUTES ABSOLUTE REGISTERS 0 THROUGH 17. THE BLOCK OF 20 REGISTERS STARTING AT JOBAC, DEFINED AS USER'S 0, STORES THE USER'S ACCUMULATORS WHILE SERVICING EXECUTIVE REQUESTS VIA THE PROGRAMMED OPERATORS.

JOBDAC

A SECOND BLOCK OF 20 REGISTERS STORES THE ACCUMULATORS WHILE A JOB IS INACTIVE. SINCE A JOB MAY BECOME INACTIVE EITHER WHILE THE USER'S PROGRAM IS COMPUTING OR WHILE THE EXECUTIVE PROGRAM IS PROCESSING A USER'S REQUEST, THE INFORMATION SAVED MAY BELONG TO TO EITHER THE USER'S PROGRAM OR THE EXECUTIVE PROGRAM.

JORUUO

TWO REGISTERS AT USER'S 40 AND 41 ARE USED FOR PROCESSING USER'S PROGRAMMED OPERATORS (00 THROUGH 37). THE TIME SHARING MONITOR RESERVES PROGRAMMED OPERATORS 40 THROUGH 77 FOR VARIOUS PURPOSES. THE REMAINING PROGRAMMED OPERATORS ARE EXECUTED AS DESCRIBED IN THE PDP-6 PROGRAMMING HANDBOOK USING USER'S 40 AND 41 INSTEAD OF ABSOLUTE 40 AND 41.

JOBPC

THE LOCATION JOBPC SAVES THE PROGRAM COUNTER AND ARITHMETIC PROCESSOR FLAGS WHILE THE JOB IS INACTIVE.

JOBPDP

PROTECTED. THE EXECUTIVE PROGRAM MAINTAINS A PUSHDOWN LIST IN THE LAST 15 (OCTAL) LOCATIONS OF THE USER'S JOB DATA AREA. JOBPDP CONTAINS THE PUSHDOWN POINTER FOR THIS PUSHDOWN LIST.

JOBREL

PROTECTED. THE LEFT HALF OF JOBREL IS UNUSED. THE RIGHT HALF CONTAINS THE HIGHEST RELATIVE LOCATION AVAILABLE TO THE USER; IN PARTICULAR, THE CONTENTS OF THE MEMORY PROTECTION REGISTER.

JOBXJT

PROTECTED. JOBXJT CONTAINS THE PROGRAM COUNTER AND PROCESSOR FLAGS FOR RETURNING TO THE USER'S PROGRAM AFTER THE EXECUTIVE IS CALLED BY A PROGRAMMED OPERATOR.

JOBSAV

PROTECTED. JOBSAV IS A GENERAL PURPOSE TEMPORARY REGISTER FOR USE BY THE PROGRAMMED OPERATOR PROCESSOR.

JOBJDA

PROTECTED. A BLOCK OF 20 REGISTERS STARTING AT JOBJDA CONTAINS THE CORRELATION BETWEEN INPUT/OUTPUT DEVICES AND THE DEVICE CHANNEL ASSIGNMENTS REQUESTED BY THE USER'S PROGRAM BY MEANS OF THE INIT COMMAND.

JOBDDT

JOBDDT CONTAINS THE STARTING ADDRESS OF DDT, THE USER'S DEBUGGING PROGRAM. IN THE EVENT OF A SYSTEM OR USER'S ERROR (FOR EXAMPLE, A USER'S MEMORY BOUNT VIOLATION) THE EXECUTIVE SYSTEM PRINTS AN ERROR MESSAGE ON THE USER'S TELETYPE CONSOLE AND RETURNS CONTROL TO THE USER'S DEBUGGING PROGRAM.

JORSYM

JORSYM CONTAINS A POINTER TO THE SYMBOL TABLE CREATED BY THE LINKING LOADER. DDT USES THIS SYMBOL TABLE FOR PRINTING AND INTERPRETING SYMBOLIC ADDRESSES. THE LEFT HALF OF JORSYM CONTAINS A NEGATIVE COUNT OF THE LENGTH OF THE SYMBOL TABLE AND THE RIGHT HALF CONTAINS THE LOWEST REGISTER USED. SEE THE DESCRIPTION OF THE LINKING LOADER FOR THE EXACT FORMAT OF THE SYMBOL TABLE.

JOBBSA

JOBBSA CONTAINS TWO ADDRESSES. THE LEFT HALF CONTAINS THE HIGHEST REGISTER USED BY THE PROGRAM. THE SYSTEM PUSHDOWN LIST IS PLACED IN THE AREA IMMEDIATELY FOLLOWING THIS HIGHEST REGISTER. THE RIGHT HALF CONTAINS THE STARTING ADDRESS OF THE USER'S PROGRAM. INITIALLY, THIS ADDRESS IS SET TO THE LATEST STARTING ADDRESS SEEN BY THE LOADER.

JOBFF

JOBFF CONTAINS THE ADDRESS OF THE FIRST FREE LOCATION FOLLOWING THE USER'S PROGRAM. THIS REGISTER IS UPDATED BY THE INPUT/OUTPUT SERVICE ROUTINES WHENEVER AN INPUT/OUTPUT BUFFER IS CREATED. THE USER'S PROGRAM SHOULD RESTORE THE CONTENTS OF JOBFF AFTER FINISHING ALL USE OF A BUFFER IN ORDER TO RECOVER THE CORE MEMORY USED.

THE REMAINING LOCATIONS IN THE JOB DATA AREA ARE RESERVED FOR FUTURE EXPANSION AND DEVELOPMENT. PROPOSED USAGE FOR SOME OF THESE LOCATIONS ARE:

JORTEM

THREE LOCATIONS BEGINNING WITH JORTEM TO BE RESERVED FOR GENERAL PURPOSE TEMPORARY SYSTEM STORAGE.

JOBINF

THE LOCATION JOBINF WOULD CONTAIN THE JOB NUMBER ASSIGNED TO THE USER IN BITS 0 THROUGH 11.

JORBCDP

JORBCDP WOULD CONTAIN A BYTE POINTER THAT POINTS TO A COMMAND STRING. THE TIME SHARING EXECUTIVE WOULD ACCEPT A COMMAND TYPED ON THE USER'S TELETYPE AND DETERMINE WHICH SYSTEM PROGRAM MUST BE CALLED TO PROCESS THE COMMAND. THE EXECUTIVE WOULD THEN LOAD THE SYSTEM PROGRAM AND SET UP JORBCDP TO POINT TO THE COMMAND STRING. THE SYSTEM PROGRAM THEN EXAMINES THE COMMAND STRING, ONE CHARACTER AT A TIME, AND PERFORMS THE COMMAND OPERATION.

JOBTRP

A BLOCK OF 20 (OCTAL) LOCATIONS BEGINNING WITH JOBTRP WILL CONTAIN INSTRUCTIONS TO BE EXECUTED WHEN AN INPUT/OUTPUT ERROR OR UNUSUAL CONDITION OCCURS. ONE LOCATION IS ASSOCIATED WITH EACH DEVICE CHANNEL. IF THE TRAP LOCATION CONTAINS 0, THEN THE PROGRAM IS DISMISSED IF AN ATTEMPT IS MADE TO TRAP ON THAT CHANNEL. OTHERWISE

A JSR INSTRUCTION IS PRESUMED TO BE IN THE TRAP LOCATION AND CONTROL IS TRANSFERRED TO THE SPECIFIED TRAP SUBROUTINE AS IF BY A JSR REGARDLESS OF WHAT INSTRUCTION IS ACTUALLY AT THE TRAP LOCATION.

JOBUSY

JOBUSY WILL CONTAIN A POINTER TO A LIST OF SYMBOLS THAT ARE UNDEFINED AT THE END OF THE LOADING PROCESS. THE USER WILL BE ABLE TO DEFINE ALL SUCH SYMBOLS BY MEANS OF DDT. DDT WILL AUTOMATICALLY ADJUST EACH LOCATION WHERE THE NEWLY DEFINED SYMBOLS ARE CALLED FOR.

IN ADDITION TO THE NAMED LOCATIONS ABOVE, ALL LOCATIONS BETWEEN JOBFF AND 140 (OCTAL) (NOT INCLUSIVE) ARE RESERVED FOR THE SYSTEM PUSHDOWN LIST. THE SYSTEM PROGRAM LOADER ALWAYS STARTS LOADING AT USER'S LOCATION 140.

LOADING USER PROGRAMS

A COPY OF THE LOADER FROM THE SYSTEM TAPE IS READ INTO A JOB AREA BY THE "GET" COMMAND AND IS STARTED BY THE "START" COMMAND, WHICH TRANSFERS COMMAND CONTROL TO THE LOADER. THE LOADER DESCRIPTION LISTS ITS COMMANDS. THE LOADER REMAINS IN CORE UNTIL ALL FILES ARE LOADED. ONCE EXECUTION IS REQUESTED, THE LOADER CALLS THE SYSTEM FUNCTION "CALL" SIXBIT /LDRBLT/ WHICH TRANSFERS THE LOADED PROGRAM DOWN OVER THE LOADER.

THE LOADER FIRST LOADS FORTRAN COMMON AT LOCATION 140, FOLLOWED BY HIS PROGRAMS, LIBRARY FILES, DDT, ETC.

THE SYMBOL TABLE IS LOADED FROM THE TOP OF MEMORY DOWN TOWARD THE PROGRAM. THE SYMBOL TABLE FORMAT, DATA FORMAT FOR THE LOADER, ETC. ARE GIVEN IN THE LOADER DESCRIPTION.

SYSTEM PUSHDOWN LIST

IN ADDITION TO SPECIFIC-PURPOSE LOCATIONS DESCRIBED BELOW, THE TIME SHARING MONITOR MAINTAINS A PUSHDOWN LIST IN THE USER'S AREA FOR GENERAL PURPOSE TEMPORARY STORAGE. ALL THE TEMPORARY STORAGE REQUIRED FOR SERVING EACH USER IS CONTAINED WITHIN THE USER'S AREA AND THUS, THE TIME SHARING MONITOR CAN SWITCH TO SERVING ANOTHER USER AT ANY TIME WITHOUT FORGETTING WHAT IT WAS DOING FOR THE ORIGINAL USER. SOME OF THE SPECIAL PURPOSE LOCATIONS DESCRIBED BELOW ARE SAVED IN REGISTERS WITHIN THE TIME SHARING MONITOR SO THAT THE USER'S PROGRAM CANNOT MODIFY THE VALUES.

MONITOR INDEPENDENCE

AS FUTURE IDEAS AND FEATURES ARE ADDED TO THE TIME SHARING MONITOR, EVERY EFFORT WILL BE MADE TO AVOID OBSOLETING OLDER PROGRAMS; HOWEVER, SOME CHANGES WILL INEVITABLY REQUIRE USERS TO MODIFY THEIR PROGRAMS. THEREFORE, EACH PROGRAM SHOULD BE WRITTEN IN TWO SEPARATE SECTIONS. ONE SECTION IS THE COMPUTATIONAL PROGRAM THAT NEED NEVER CHANGE, REGARDLESS OF CHANGES IN THE TIME SHARING MONITOR. THE OTHER SECTION IS THE INITIALIZATION AND SYSTEM INTERFACE SECTION; THE CODING THAT COMMUNICATES DIRECTLY WITH THE TIME SHARING MONITOR. WITH ALL SYSTEM-DEPENDENT CODING ISOLATED INTO ONE SECTION OF THE PROGRAM+ UPDATING

PROGRAMS TO RUN IN FUTURE SYSTEMS WILL BE A RELATIVELY EASY CHORE.

JOBDEF AND OPDEF TAPES SHOULD BE USED, RATHER THAN ABSOLUTE ASSIGNMENTS FOR NAMING SYSTEM REGISTERS OR FUNCTIONS.

CHAPTER IV
USER INPUT OUTPUT INSTRUCTIONS

GENERAL

THE INPUT/OUTPUT SERVICE ROUTINES PROVIDE THE INPUT/OUTPUT FACILITIES FOR PROGRAMS RUNNING UNDER THE PDP-6 TIME-SHARING MONITOR SYSTEM. THE OBJECT OF THESE ROUTINES IS TO SIMPLIFY CODING FOR THE VARIOUS INPUT/OUTPUT DEVICES AS MUCH AS POSSIBLE AND YET TO PROVIDE FLEXIBILITY FOR PERFORMING THE MOST SOPHISTICATED FUNCTIONS OF INPUT/OUTPUT. FULL ADVANTAGE IS TAKEN OF THE PROGRAM INTERRUPT FACILITY OF THE PDP-6 TO OVERLAP IN/OUT OPERATIONS WITH COMPUTATION.

THE USER'S PROGRAM COMMUNICATES WITH THE I/O SERVICE ROUTINES BY MEANS OF THE PROGRAMMED OPERATORS. A PROGRAMMED OPERATOR AND THE MACHINE INSTRUCTION SET, UOO (UNUSED OPERATION CODE) ARE SYNONOMOUS. TYPICAL COMMANDS ARE TO SET UP A BUFFER AREA, TO TRANSMIT A BUFFER FULL OF DATA TO AN OUTPUT DEVICE, OR TO RELEASE A DEVICE FOR USE BY OTHER USERS.

DEVICE DEPENDENT OPERATIONS

SOME I/O DEVICES REQUIRE SPECIAL COMMANDS IN ADDITION TO THOSE DESCRIBED BELOW. FOR EXAMPLE, MAGNETIC TAPE REQUIRES A REWIND COMMAND, DECTAPE REQUIRES BLOCK NUMBER COMMANDS. THESE COMMANDS AND OTHER INDIVIDUAL CHARACTERISTICS ARE DESCRIBED SEPARATELY FOR EACH DEVICE. COMMANDS IRRELEVANT TO A PARTICULAR DEVICE ARE IGNORED. FOR EXAMPLE, A FILE DIRECTORY MANIPULATION COMMAND (REQUIRED FOR DECTAPE OPERATIONS) WHEN GIVEN TO THE LINE PRINTER IS IGNORED. THUS, TO CHANGE A PROGRAM FROM OUTPUTTING TO THE LINE PRINTER, THE USER NEED ONLY CHANGE THE DEVICE NAME IN THE INITIALIZATION COMMAND. THE MNEMONICS FOR THE FOLLOWING COMMANDS AND THEIR CODE ASSIGNMENTS ARE ON THE SYSTEM FILE, OPDEF. OPDEF IS IN THE APPENDIX.

THE COMMANDS CAN BE SUMMARIZED AS FOLLOWS:

FILE INITIALIZATION

RESET	SYSTEM RESET FOR JOB
ASSIGN	GET A DEVICE, AND RENAME IT
INIT	ATTACH FREE DEVICE TO A JOB CHANNEL
INBUF,OUTBUF	SETUP BUFFERS (OPTIONAL)
LOOKUP,ENTER	FIND OR ASSIGN FILE NAMES

FILE DATA TRANSMISSION (NON-DUMP)
INPUT, OUTPUT

FILE MANIPULATION (SPECIAL DEVICES)	
INPUT,OUTPUT	(DUMPMODE)
USETO,USETI	SET IO FILE BLOCKS
UGETF	GET FREE FILE NUMBER
MTAPE	PERFORM AN ACTION ON MAGTAPE

FILE STATUS EXAMINATION
STATUS,STATZ,STATO CHECK DEVICE STATUS

FILE TERMINATION
CLOSE TERMINATES CURRENT FILE

RELEASES
DEASSIGN
EXIT

RETURNS DEVICE TO SYSTEM, UNLESS ASSIGNED
RETURNS DEVICE TO SYSTEM

THE ABOVE COMMAND PAIRS, RESET-EXIT, ASSIGN-DEASSIGN, INIT-RELEASE,
AND LOOKUP/ENTER-CLOSE, EACH WORK AT A DEVICE, CHANNEL-DEVICE,
OR FILE LEVEL RESPECTIVELY.

RING BUFFERS

CORE MEMORY SERVES AS AN INTERMEDIATE BUFFER BETWEEN THE USER'S
PROGRAM AND THE IN/OUT DEVICE. AN I/O BUFFER CONSISTS OF A HEADER
BLOCK (THREE WORDS FOR BOOKKEEPING) AND A DATA STORAGE AREA. THE
STORAGE AREA IS SUBDIVIDED INTO ONE OR MORE INDIVIDUAL BUFFERS
ARRANGED IN A RING. TO BE SPECIFIC, ASSUME THAT THE DEVICE IS AN INPUT
DEVICE. ONCE INPUT OPERATIONS ARE UNDER WAY, THE I/O SERVICE ROUTINES
FILL A BUFFER, MAKE THE BUFFER AVAILABLE TO THE USER'S PROGRAM,
ADVANCE TO THE NEXT BUFFER, AND BEGIN TO FILL IT IF IT IS FREE. AFTER
FILLING THE LAST BUFFER IN THE STORAGE AREA, THE NEXT BUFFER TO BE
FILLED IS THE FIRST BUFFER IN THE AREA; HENCE THE NAME RING BUFFER.

THE USER'S PROGRAM FOLLOWS ALONG BEHIND, EMPTYING THE NEXT BUFFER IF
IT IS FULL, OR WAITING FOR THE NEXT BUFFER TO FILL UP IF IT IS NOT
YET FULL.

FOR OUTPUT, THE USER'S PROGRAM AND THE I/O ROUTINES EXCHANGE ROLES,
THE USER FILLING THE BUFFERS AND THE I/O ROUTINES EMPTYING THEM.

BUFFER STRUCTURE

THE BUFFER STRUCTURE CONSISTS OF TWO SEPARATE SEGMENTS, THE HEADER AND
THE STORAGE AREA. THE HEADER CONSISTS OF THE FOLLOWING THREE WORDS:

- 1) THE ADDRESS OF THE SECOND WORD OF THE BUFFER CURRENTLY
IN USE BY THE USER'S PROGRAM IN THE RIGHT HALF. BIT 0=1
MEANS NO INPUT OR OUTPUT HAS OCCURRED FOR THIS RING.
- 2) A BYTE POINTER TO THE LAST ITEM REFERENCED IN THE BUFFER.
- 3) A COUNT OF THE NUMBER OF ITEMS REMAINING IN THE BUFFER.

FOR MOST APPLICATIONS, THE USER'S PROGRAM FINDS ALL THE INFORMATION
IT REQUIRES ABOUT THE STATE OF THE BUFFER AREA IN THE HEADER. WHEN
REFERRING TO A BUFFER BY NAME, THE ADDRESS OF THE BUFFER HEADER, NOT
THE STORAGE AREA, IS IMPLIED.

THE STORAGE AREA IS SUBDIVIDED INTO A NUMBER OF INDIVIDUAL,
EQUAL-LENGTH BUFFERS. THE SIZE OF EACH BUFFER DEPENDS ON THE
REQUIREMENTS OF THE DEVICE SERVICED BY THE BUFFER. THE FIRST TWO
WORDS OF EACH BUFFER ARE RESERVED FOR SPECIAL FUNCTIONS; THE REMAINING
WORDS ARE FOR DATA STORAGE. THE FIRST TWO WORDS ARE:

- 1) RESERVED FOR A BLOCK NUMBER FROM FIXED ADDRESS DEVICES
SUCH AS DECTAPE OR DRUM.

- 2) IN THE RIGHT HALF, THE ADDRESS OF THE SECOND WORD OF THE NEXT BUFFER IN THE RING. IN THE LEFT HALF, BIT 0 IS A FLAG AS EXPLAINED BELOW; BITS 1 THROUGH 17 CONTAIN THE BUFFER SIZE IN WORDS (EXCLUDING THE TWO WORDS OF OVERHEAD).

BIT 0 OF THE SECOND WORD OF THE BUFFER, CALLED THE USE BIT, IS A FLAG THAT INDICATES WHETHER THE BUFFER CONTAINS ACTIVE DATA. THIS BIT IS SET TO 1 BY THE I/O ROUTINES WHEN THE BUFFER IS FULL ON INPUT (OR BEING EMPTIED ON OUTPUT) AND SET TO 0 WHEN THE BUFFER IS EMPTY ON OUTPUT (OR BEING FILLED ON INPUT). THE USE BIT PREVENTS THE I/O ROUTINES AND THE USER'S PROGRAM FROM INTERFERING WITH THE USER'S PROGRAM BY ATTEMPTING TO USE THE SAME BUFFER SIMULTANEOUSLY.

IN ALL DATA PROCESSING MODES, THE FIRST DATA WORD OF THE BUFFER IS RESERVED FOR A COUNT OF THE NUMBER OF DATA WORDS IN THE BUFFER (EXCLUDING ITSELF). THE LEFT HALF OF THIS WORD IS RESERVED FOR OTHER BOOKKEEPING PURPOSES DEPENDING ON THE PARTICULAR DEVICE.

RESET COMMAND

TO CLEAR ALL PREVIOUS DEVICE ASSIGNMENTS AND PREPARE THE I/O SERVICE ROUTINES FOR RECEIVING FURTHER INITIALIZATION COMMANDS, EXECUTE THE INSTRUCTION:

CALL [SIXBIT /RESET/]

THIS MUST BE THE FIRST INSTRUCTION IN EACH PROGRAM.

ASSIGN COMMAND

THE ASSIGN COMMAND IS GIVEN TO ATTACH A PHYSICAL DEVICE TO A JOB, AND AT THIS TIME RENAME THE DEVICE BY A LOGICAL NAME, AN ARBITRARY SIXBIT NAME. IF MORE THAN ONE PHYSICAL DEVICE MAY BE USED (E.G. ANY 1 OF N TELETYPES). THE NUMBER OF THE DEVICE IS SUPPLIED BY THE SYSTEM. AN EXAMPLE IS:

ASSIGN D, ADR
ERROR RETURN ; NOT AVAILABLE
SUCCESSFUL RETURN

ADR: SIXBIT /DESIRED PHYSICAL NAME/
ADR+1: SIXBIT /DESIRED LOGICAL NAME/
C(ADR) MAY CONTAIN A 3 CHARACTER DEVICE NAME WHICH THE
;SYSTEM SUPPLIES. THE FIRST FREE PHYSICAL DEVICE IS USED.
;AT COMPLETION THE C(AC-D) CONTAIN A ;SIXBIT PHYSICAL NAME

IF A PHYSICAL DEVICE IS ALREADY ATTACHED TO THE JOB, AND AN ASSIGN IS GIVEN, THE NEW NAME IS USED. IF NO RENAMING IS TO OCCUR, C(ADR+1)=0.

INITIALIZATION

INIT COMMAND

EACH DEVICE IS INITIALIZED AND BUFFER HEADERS SPECIFIED BY EXECUTING THE COMMAND:

```
-----  
INIT D, MODE  
SIXBIT /DEVICE NAME/  
XWD OBUF,IBUF  
ERROR RETURN  
NORMAL RETURN  
-----
```

INIT COMMAND PERFORMS THE FOLLOWING FUNCTIONS:

- 1) THE 4 BIT CHANNEL NUMBER D IS ASSIGNED TO THE DEVICE NAME APPEARING IN THE SIXBIT STATEMENT. HENCEFORTH, THE I/O SERVICE ROUTINES INTERPRET THE NUMBER D AS REFERRING TO THAT DEVICE.
- 2) THE DATA PROCESSING MODE IS SELECTED BY MODE (SEE NEXT SECTION).
- 3) THE BUFFER AREA HEADERS ARE SPECIFIED TO THE DEVICE SERVICE ROUTINES. EACH HEADER MUST BE THREE LOCATIONS RESERVED BY THE USER IN HIS PROGRAM. ONLY THOSE HEADERS WHICH ARE TO BE USED NEED BE SPECIFIED; THE OUTPUT HEADER NEED NOT BE SPECIFIED IF THE PROGRAM PERFORMS ONLY INPUT FROM A DEVICE) THE DEVICE IS EITHER A LOGICAL OR PHYSICAL NAME. THE FIRST SEARCH IS FOR A LOGICAL NAME, FOLLOWED BY A PHYSICAL NAME.

INIT DATA MODES

ONE ITEM OF INFORMATION SPECIFIED IN THE INIT COMMAND IS THE DATA PROCESSING MODE. THE FOLLOWING MODES ARE AVAILABLE:

OCTAL CODE	MNEMONIC	MEANING
0	A	ASCII, 7-BIT CHARACTERS, PACKED AS BY THE BYTE COMMANDS.
1	AL	ASCII LINE, SAME AS A EXCEPT THAT THE BUFFER IS TERMINATED BY A FORM-FEED, VERTICAL TAB, OR LINE-FEED.
5	AM	ALT MODE, SAME AS AL EXCEPT THAT THE LINE IS TERMINATED ALSO ON ALT-MODE.
10	i	IMAGE, A DEVICE DEPENDENT MODE, THE BUFFER IS FILLED WITH UNPROCESSED DATA EXACTLY AS SUPPLIED BY THE DEVICE
14	B	BINARY, A BLOCKED FORMAT CONSISTING OF A WORD COUNT (THE FIRST WORD OF THE BUFFER) FOLLOWED BY N 36-BIT DATA WORDS. CHECK-SUMMING IS DONE AUTOMATICALLY BY THE SERVICE ROUTINES OR BY THE DEVICE ITSELF.

N IS IN RIGHT HALF OF FIRST WORD, AND IF CHECKSUM IS COMPUTED (CARDS, PAPER TAPE) IT IS IN BITS 6-17. THE 12 BIT FOLDED CHECKSUM IS COMPUTED BY SUMMING THE 36 BIT DATA (2'S COMPLEMENT ADD), FOLLOWED BY A 1'S COMPLEMENT SUM OF BITS 0-11, 12-23, 24-35.

15, 16, 17

D DUMP, DATA IS TRANSMITTED BETWEEN ANY CONTIGUOUS BLOCK OF CORE AND THE I/O DEVICE. IN DUMP MODE, RING BUFFERS ARE NOT USED. SEE SPECIFIC DUMP DESCRIPTIONS FOR EACH DEVICE. THE CONTROL FOR THE SIZE, LOCATION IS FROM A COMMAND LIST.

INIT DATA CONTROL MODES

THE USER MAY SPECIFY CHARACTERISTICS THAT FURTHER DEFINE THE DATA TRANSFER.

BIT	NAME	ACTION
31	IOWC	IO WORD COUNT. IF A 1, COMMANDS THE SERVICE ROUTINES TO ACCEPT THE WORD COUNT IN THE FIRST DATA WORD OF THE BUFFER (AS COMPUTED BY THE USER) INSTEAD OF COMPUTING A WORD COUNT FROM THE CONTENTS OF THE BYTE POINTER IN THE BUFFER HEADER.
30	IOCON	IO CONTINUOUS MODE. IF A 1, SPECIFIES STOPPING THE DEVICE AFTER EACH BUFFER IS FILLED (INPUT ONLY).
29	IORDEL	IO READ AND DELETE A FILE IF A 1. (DECTAPE, DRUM, DISC.)
29	IORCK	IO NO READ CHECK. SUPPRESS THE ERROR CORRECTING IF A 1. I.E. NORMAL MODE IS TO RE-READ A RECORD 3 TIMES BEFORE SETTING ERROR OR TO CONTINUE TO RE-WRITE UNTIL A GOOD RECORD IS WRITTEN. (MAG TAPE, DISC, DECTAPE.)
27-28	IODENS	IO DENSITY FOR MAG TAPE. 0,2=556 BPI. 1=200 BPI. 3=800 BPI.
26	IOPAR	IO PARITY CHECKING FOR MAG TAPE. 0 FOR ODD, OR NORMAL PARITY. 1 FOR BCD OR EVEN PARITY.

INBUF COMMAND

TO CAUSE AN INPUT BUFFER AREA TO BE SET, ISSUE THE COMMAND:

 INBUF D,N

 WHERE N SPECIFIES AN N BUFFER RING. THE STORAGE IS TAKEN FROM THE FREE STORAGE FOLLOWING THE PROGRAM. THE LOCATION JOBFF (A PERMANENT GLOBAL SYMBOL DEFINED BY IN SYSTEM TAPE, JOBDEF) CONTAINS

THE ADDRESS OF THE FIRST FREE LOCATION FOLLOWING THE PROGRAM. IF THE USER WISHES TO ABANDON A BUFFER FOR ONE DEVICE AND SET UP BUFFER FOR ANOTHER DEVICE, THE PROGRAM SHOULD RESTORE THE CONTENTS OF JOBFF TO THE VALUE CONTAINED BEFORE THE FIRST BUFFER WAS SET UP. THEN THE I/O SERVICE ROUTINES CAN REUSE THE OLD BUFFER AREA IN CORE MEMORY FOR SETTING UP THE NEW BUFFER.

OUTBUF COMMAND

AN OUTPUT BUFFER AREA MAY BE SET UP BY:

OUTBUF D,N

IF NO BUFFER AREA IS SET UP WHEN THE FIRST INPUT OR OUTPUT COMMAND IS ISSUED, A ONE BUFFER RING IS SET UP AUTOMATICALLY FROM THE FREE STORAGE FOLLOWING THE PROGRAM.

A DEVICE MAY BE INITIALIZED SEPARATELY ON TWO DEVICE CHANNELS, IF ONE INIT COMMAND SPECIFIES THE OUTPUT BUFFER HEADER AND THE OTHER SPECIFIES THE INPUT BUFFER HEADER. ALL FURTHER COMMANDS MAY THEN REFER TO EITHER CHANNEL NUMBER ; THE OUTPUT COMMAND CAN USE THE CHANNEL THAT WAS INITIALIZED FOR INPUT FOR EXAMPLE.

LOOKUP, ENTER COMMANDS

THESE COMMANDS ARE GIVEN TO EACH DEVICE, THEY SPECIFY A FILE NAME, FILE NAME EXTENSION, MODE FOR A FILE. THE COMMANDS HAVE NO EFFECT FOR SINGLE FILE DEVICES, E.G. PTR, PTP, TTY, LPT, CDR, BUT ARE USED WITH DTA, MTA, DRUM, DISC, ETC. WHICH HAVE MULTIPLE FILES/DEVICE AND A DIRECTORY TO HOLD FILE NAMES. AN ENTER MUST BE GIVEN BEFORE OUTPUT, AND LOOKUP BEFORE INPUT. A LOOKUP COMMAND FORMAT EXAMPLE IS:

LOOKUP D,E ;SEARCH DIRECTORY FOR
ITEM NOT FOUND ;ENTRY SPECIFIED BY C(E).
OK RETURN ;

;SEARCHES C(E,E+1, E+2, E+3,) FOR FILE NAME, (6-SIXBIT)
;FILE NAME EXTENSION (3-SIXBIT).
;AND FILLS IN BLANK INFORMATION NEEDED
;TO MANIPULATE THE FILE

AN ENTER COMMAND FORMAT EXAMPLE IS :

ENTER D,E ;ENTER DIRECTORY INFO. FROM
FULL DIRECTORY ;C(E)
OK RETURN ;
;SEARCHES DIRECTORY AND REPLACES OLD ITEM
;WITH NEW. IF NONE IS PRESENT AN ENTRY
;IS MADE.

NON DUMP INPUT, OUTPUT

TWO PROGRAMMED OPERATORS ARE REQUIRED FOR NORMAL I/O OPERATION. THESE ARE:

INPUT D,ADR AND OUTPUT D,ADR

THE FUNCTIONS OF THESE TWO COMMANDS ARE SIMILAR, ONE SERVING FOR INPUT OPERATIONS AND THE OTHER FOR OUTPUT OPERATIONS. USUALLY, THE ADDRESS ADR IS 0 AND THE COMMANDS OPERATE AS FOLLOWS:

INPUT COMMAND

RELEASES THE CURRENT BUFFER TO THE I/O ROUTINES FOR REFILLING AND ADVANCES THE BUFFER HEADER POINTERS TO THE NEXT BUFFER. IF THE NEXT BUFFER IS NOT FULL WHEN THE INPUT COMMAND IS GIVEN, THE I/O ROUTINES DELAY UNTIL THE BUFFER IS FULL OR AN END OF DATA INDICATION IS RECEIVED. SINCE INPUT OPERATIONS NORMALLY CONTINUE UNTIL ALL BUFFERS ARE FULL, THE INPUT COMMAND SERVES TO SYNCHRONIZE THE USER AND THE INPUT DEVICE.

OUTPUT COMMAND

COMPUTES A WORD COUNT FROM THE POSITION OF THE BYTE POINTER, STORES IT IN THE RIGHT HALF OF THE FIRST DATA WORD, AND MAKES THE BUFFER AVAILABLE TO THE DEVICE FOR OUTPUTTING. THE BUFFER HEADER POINTERS ARE ADVANCED TO THE NEXT BUFFER AND THE ITEM COUNT IS SET TO THE MAXIMUM NUMBER THAT CAN FIT IN THE BUFFER. THE SERVICE ROUTINES ASSURE THAT THE BUFFER IS EMPTY BY WAITING UNTIL IT IS OUTPUT, IF NECESSARY, BEFORE RETURNING AND CLEAR THE BUFFER TO ZERO. WHEN THE INIT COMMAND IS GIVEN, THE ITEM COUNT IN EACH BUFFER HEADER IS SET TO ZERO AND THE BYTE SIZE IS SET ACCORDING TO THE DATA PROCESSING MODE. THE INBUF AND OUTBUF COMMANDS SET THE POINTERS TO THE FIRST BUFFER IN THE RING. THE REMAINING POINTERS ARE SET UP WHEN THE FIRST INPUT OR OUTPUT COMMAND IS GIVEN. IF THE ITEM COUNT IS IGNORED BY THE USER'S PROGRAM DURING OUTPUT OPERATIONS, THE OUTPUT COMMAND MUST BE GIVEN ONCE AFTER THE INIT COMMAND TO SET UP THE OUTPUT BUFFER HEADER. IF THE USER'S OUTPUT SUBROUTINE CHECKS THE ITEM COUNT BEFORE PUTTING DATA INTO THE OUTPUT BUFFER, THIS INITIAL OUTPUT COMMAND HAPPENS AUTOMATICALLY.

A TYPICAL GET ONE CHARACTER SUBROUTINE FROM AN INPUT DEVICE IS :

GET: 0
 SOSG IBUF+2 ;DECREMENT ITEM COUNT AND TEST
 INPUT D, ;IF NO DATA, CALL INPUT
 ILDDB AC,IBUF+1 ;GET NEXT CHARACTER IN AC
JRST @GET ;EXIT

A TYPICAL OUTPUT ONE CHARACTER SUBROUTINE WOULD BE:

PUT: 0
 SOSG OBUF+2 ;DECREMENT ITEM COUNT AND TEST
 OUTPUT D, ;IF NO ROOM LEFT CALL OUTPUT
 IDPB AC,OBUF+1 ;DEPOSIT CHARACTER FROM AC

IF ADR IS NON ZERO, IT IS TAKEN AS A POINTER TO THE SECOND WORD OF THE NEXT BUFFER. THE SEQUENCE OF BUFFERS IN THE RING THUS CAN BE ALTERED. THE BUFFER POINTED TO BY ADR CAN BE IN AN ENTIRELY SEPARATE RING FROM THE PREVIOUS BUFFER. ONCE A NEW BUFFER POSITION IS ESTABLISHED, FOLLOWING BUFFERS ARE TAKEN FROM THE RING STARTING AT ADR. WHEN USING THIS FEATURE IN OUTPUT OPERATIONS, THE I/O ROUTINES COMPUTE AN IMPROPER WORD COUNT UNLESS ONE OF THE FOLLOWING PRECAUTIONS ARE TAKEN. EITHER 1) IN THE INIT COMMAND HAVE BIT 31 A ONE TO PREVENT THE SERVICE ROUTINES FROM COMPUTING A WORD COUNT (USEFUL IF THE USER'S PROGRAM COMPUTES ITS OWN WORD COUNT AND PLACES IT IN THE RIGHT HALF OF THE FIRST DATA WORD OF THE BUFFER OR IF OUTPUTTING DIRECTLY FROM AN INPUT BUFFER) OR 2) SET THE BYTE POINTER OF THE BUFFER HEADER TO POINT TO THE LAST ITEM IN THE NEW BUFFER.

STATUS , STATZ, STATO, COMMANDS

CERTAIN ERRORS, SUCH AS PARITY OR CHECKSUM ERRORS, CAN OCCUR DURING DATA PROCESSING. FOR EACH 4-BIT DEVICE NUMBER, A STATUS WORD IS PROVIDED WHICH CONTAINS ERROR INDICATION BITS AND OTHER USEFUL CONDITIONS. THE STATUS WORD IS EXAMINED OR TESTED BY THE COMMANDS:

STATUS D,ADR OR STATZ D, MASK OR STATO D,MASK

THESE THREE COMMANDS ARE EXACTLY ANALOGOUS TO THE COMMANDS CONI, CONSZ, AND CONSO EXCEPT THAT D IS THE 4-BIT DEVICE NUMBER.

STATUS BITS

THE BITS OF THE STATUS WORD HAVE THE FOLLOWING MEANINGS:

BIT	MEANING
18	IOIMPM, IMPROPER MODE, THE SELECTED MODE IS UNOBTAINABLE ON THE SELECTED DEVICE.
19	IODERR, DEVICE ERROR, THE DEVICE'S SELF-CHECKING CIRCUITS INDICATE AN ERROR (PARITY FAILURE(ETC.)
20	IODTER, DATA ERROR, THE COMPUTED CHECKSUM FAILED OR INVALID DATA WAS RECEIVED.
21	IOBKTL, BLOCK TOO LARGE, A BLOCK OF DATA FROM A HIGH-SPEED DEVICE (DECTAPE, MAG TAPE, DRUM) WAS TOO LARGE TO FIT IN ONE BUFFER.
22	IODEND, DATA END ENCOUNTERED.
23	IOACT, DEVICE ACTIVE AND CURRENTLY TRANSMITTING DATA.
24	IOBOT, BEGINNING OF TAPE
25	IOTEND, END OF TAPE
26	IOPAR, WRITE EVEN PARITY

- 27,28 IODENS, DENSITY OF TAPE
- 29 IONRCK, SUPPRESS ERROR CHECKING IF =1. NORMAL MODE.0.
RE-READS RECORD 3 TIMES IN ATTEMPT TO GET CORRECT DATA.
- 29 IORDEL, READ AND DELETE FILE.
- 30 IOCON, NON-CONTINUOUS OPERATION.
- 31 IOWC, DO NOT COMPUTE A WORD COUNT ON OUTPUT.
- 32-35 DATA PROCESSING MODE.

BITS 26 THROUGH 35 READ BACK THE MODE TRANSMITTED TO THE I/O SERVICE ROUTINES BY THE MOST RECENT INIT COMMAND.

DUMP MODE INPUT-OUTPUT

THE DUMP MODE FORMAT IS A CONTIGUOUS GROUP OF ARBITRARY SIZED DATA BLOCKS WITHIN A SINGLE FILE. DUMP MODE IS USED TO STORE A COMPLETE PROGRAM IN BLOCKED CORE AREAS, AND REQUIRES NO INTERMEDIATE DATA BUFFERING. IT IS AVAILABLE FOR HIGH SPEED DEVICES (DECTAPE, MAGTAPE, DRUM, OR DISC) AND IS USED TO STORE THE OPERATING EXECUTIVE SYSTEM, THE CUSPS, AND USER PROGRAMS.

THE CONTROL OF INFORMATION IS THROUGH A COMMAND LIST. THE LOCATION OF THE COMMAND LIST IS GIVEN BY THE INPUT OR OUTPUT INSTRUCTION. COMMANDS EXIST TO TRANSFER A BLOCK, JUMP, OR STOP.

COMMAND LIST FORMAT

THE COMMANDS FORMAT VARIES FOR THE SPECIFIC DEVICE BUT ARE IN GENERAL:

```

-----
IOWD      A,B      ;PLACE "A" WORDS, STARTING AT "B" ON THE
              ;FILE.
XWD       0, A     ;GO TO A FOR THE NEXT COMMAND.
              0     ;TERMINATE THE FILE
-----

```

OUTPUT D, ADR COMMAND

DUMP MODE IS WRITTEN IN A FILE BY GIVING AN OUTPUT D,ADR COMMAND. THE CORE LOCATION "ADR" CONTAINS A DUMP FILE COMMAND LIST, WHICH SPECIFIES THE BLOCKS OF CORE THAT ARE TO BE WRITTEN IN THE FILE.

INPUT D,ADR COMMAND

INPUTS A FILE AS SPECIFIED BY COMMAND LIST AT "ADR".

INPUT D, COMMAND

INPUTS A FILE USING THE FIRST WORD OF THE FILE AS THE COMMAND LIST.

CLOSE COMMAND

WHEN FINISHING INPUT AND OUTPUT OPERATIONS FOR A FILE, THE USER
ISSUES THE COMMAND:

CLOSE D

CLOSE FINISHES OUTPUT OPERATIONS ON DEVICE D BY DUMPING THE LAST
BUFFER IF IT IS PARTIALLY FULL AND, PERHAPS, BY WRITING AN END
CHARACTER IF THE DEVICE REQUIRES ONE. FOR EITHER INPUT OR OUTPUT,
THE DEVICE IS RETURNED TO THE SAME STATE AS BEFORE THE FIRST INPUT
OR OUTPUT COMMAND.

RELEASE COMMAND

RELEAS MAY PERFORM A CLOSE OPERATION AND THEN RELEASES THE DEVICE, D,
FOR USE BY OTHER USERS UNLESS AN ASSIGN HAD BEEN GIVEN TO REQUEST THE
DEVICE. A NEW INIT COMMAND MUST BE GIVEN IF FURTHER OPERATIONS WITH THE
DEVICE ARE REQUIRED. THE COMMAND IS:

RELEASE D,

IF A DEVICE IS INITIALIZED ON MORE THAN ONE CHANNEL, ALL SUCH
CHANNELS, BOTH INPUT AND OUTPUT ARE AFFECTED BY RELEAS COMMANDS.

DEASSIGN COMMAND

THIS COMMAND RETURNS A DEVICE BACK TO THE SYSTEM. THE DEASSIGNMENT
IS ON A NAMED BASIS, AND THE SYSTEM FIRST SEARCHES TO FIND A CORRECT
LOGICAL NAME FOR A DEVICE TO DEASSIGN, FOLLOWED BY A SEARCH FOR A
PHYSICAL DEVICE. THE COMMAND IS:

DEASSIGN, ADR
ADR- 0 SIXBIT /LOGICAL OR PHYSICAL NAME/

EXIT COMMAND

TO RELEASE ALL DEVICES AND STOP EXECUTION, ISSUE THE COMMAND:

CALL [SIXBIT/EXIT/]

CONSOLE OR USER TELETYPEWRITERS

EQUIPMENT

A MODEL 33 OR 35 MAY BE USED, WITH EITHER HALF OR FULL DUPLEX COMMUNICATION WITH THE 630 DATA COMMUNICATIONS SYSTEM AND THE 626 CONSOLE TELEPRINTER (FULL DUPLEX). HORIZONTAL TAB, VERTICAL TAB, AND FORM FEED ARE SYNTHETICALLY GENERATED USING SPACES AND LINE FEEDS WHEN A MODEL 33 TELETYPE IS USED. THE SETTING FOR TABULATION IF A MODEL 35 IS USED ARE: HT = 8 SPACES, VT = 4 LINE FEEDS, AND FF = 8 LINE FEEDS (1 PAGE).

INPUT CONTROL CHARACTERS

CR SYSTEM RESPONDS BY TYPING A LINE FEED.

RUB-OUT -- DELETES PREVIOUS CHARACTER TYPED, SYSTEM RESPONDS BY TYPING\.

ALT MODE,

LINE FEED (TYPED BY USER OR SYSTEM). VERTICAL TAB, OR FORM FEED. DELIMIT A LINE, COMPLETE AN INPUT COMMAND, OR TRANSFERS CONTROL TO THE SYSTEM COMMAND DECODER.
CONTROL Z =S2 -- DATA END.

PAPER TAPE READER

A, AL MODE CONTROL

IGNORE RUBOUT, BLANK TAPE, NULL (200).

B MODE CONTROL

BLOCKED DATA WITH BLOCK WORD COUNT AND CHECKSUM IN THE FIRST WORD OF THE BLOCK. (SEE PAPER TAPE PUNCH)

MANUAL OPERATION

TURNING THE READER SWITCH OFF GIVES THIS DATA END CONDITION. AN S2 CHARACTER (32) IS ALWAYS GENERATED AS THE LAST CHARACTER.

PAPER TAPE PUNCH

SPECIAL CLOSE

INITIAL AND FINAL TAPE FEED OCCURS AT FIRST OUTPUT, AND AT CLOSE.

A,AL MODE CONTROL

200 IS ADDED TO ALL CHARACTERS). TAPE FEED IS INSERTED AFTER FORM FEED. A RUBOUT IS INSERTED AFTER A TAB.

B MODE CONTROL

PUNCHES EACH BUFFER AS A BINARY BLOCK OF DATA. THE FIRST WORD IN EACH BLOCK IS THE WORD COUNT IN THE RIGHT HALF AND THE CHECKSUM IN THE LEFT HALF, FOLLOWED BY THE DATA. TAPE FEED IS INSERTED AFTER EACH BLOCK.

LINE PRINTER

SPECIAL CLOSE

FORM FEED OCCURS PRIOR TO ACTUAL FIRST OUTPUT, AND AFTER CLOSE.

SPECIAL CONTROL CHARACTERS
SPECIAL CODES HAVE MEANING TO THE LINE PRINTER. THESE ARE:

HORIZONTAL TAB (Ø11) = 1Ø SPACES
FORM FEED (Ø14) = 1 PAGE
EOT (ØØ4) = STOPS PRINTER
DCØ (Ø2Ø) = 1 LINE
DC1 (Ø21) = 2 LINES
DC2 (Ø22) = 3 LINES
DC3 (Ø23) = 6 LINES
DC4 (Ø24) = 11 LINES
VERTICAL TAB (Ø13) = 22 LINES

CARD READER

GENERAL

EACH BUFFER CONTAINS THE CONTENTS OF ONE CARD FOR ALL MODES.

THE BINARY CARD FORMAT WILL CONTAIN A 7-9 PUNCH IN COLUMN ONE. ROWS 2 THROUGH 6 OF COLUMN 1 WILL CONTAIN THE NUMBER OF DATA WORDS ON THE CARD. ROWS 11, 12, Ø, 1 MAY BE USED FOR MODE AND TYPE INFORMATION. ZERO IN THOSE ROWS WILL MEAN A NORMAL BINARY CARD. COLUMN 2 WILL CONTAIN A FOLDED 12 BIT CHECKSUM. COLUMN 3 THROUGH 8Ø MAY CONTAIN DATA WORDS. FOR DENSELY PACKED BINARY CARDS, THIS PERMITS 26 WORDS PER CARD. FOR SEQUENCED NUMBERED CARDS, THIS PERMITS 24, IF 6 COLUMNS ARE USED FOR SEQUENCING INFORMATION. THE CHECKSUM WILL INCLUDE ONLY THE DATA WORDS.

A, AL MODE OPERATION

BLANKS ARE REPLACED BY SPACES. A CARRIAGE RETURN AND LINE FEED IS APPENDED TO EACH CARD. BUFFER CONSISTS OF 8Ø CHARACTERS, CR, LF. ALL CHARACTERS ARE TRANSFORMED TO ASCII. (SEE THE MACRO-6 MANUAL FOR A TABLE GIVING THE TRANSLATION BETWEEN HOLLERITH AND ASCII)

B MODE

BINARY READS FIRST TWO COLUMNS AS CONTROL INFORMATION. COLUMN 1 MUST CONTAIN A 7-9 PUNCH. COLUMN 2 IS A 12-BIT CHECKSUM AS DESCRIBED FOR THE PAPER TAPE PUNCH.

MANUAL CONTROL

EOF SWITCH CAUSES IODEND TO BE SET.

DECTAPE

DECTAPE FORMAT

1103 ADDRESSABLE BLOCKS OF 200 WORDS EACH WITH BLOCK 0-A LOADER, BLOCK 1-THE DIRECTORY, AND BLOCKS 2-1101-DATA.

BLOCK 1: DIRECTORY FORMAT

WORD 0, C(LH): LAST BLOCK OF TAPE USED (2 INITIALLY)
WORD 0, C(RH): LOCATION ON BLOCK 1 OF THE FIRST DIRECTORY ENTRY. (5 INITIALLY)
DIRECTORY ENTRIES: 4 WORDS PER ENTRY
WORD 0: 6 SIXBIT NAME OF FILE
WORD 1, C(LH)- 0 3 SIXBIT FILE EXTENSION
WORD 1, C(RH): FIRST BLOCK OF FILE
WORD 2, BITS 0-2: ACTIVITY MODES (UNUSED)
WORD 2, BITS 3-6: DATA MODE (A,AS,B,ETC.)
WORD 2, BITS 8-35: DATE
WORD 3, C(LH): DUMP FILE WORD COUNT
WORD 3, C(RH): DUMP FILE ADDRESS

BLOCKS 2-1101:

NON DUMP FILE DATA:

EACH BLOCK CONTAINS UP TO 177 DATA WORDS.
WORD 0, C(LH): LINK TO NEXT BLOCK OF FILE.
WORD 0, C(LH)=0: LAST BLOCK OF FILE
WORD 0, C(RH): NUMBER OF ITEMS IN BLOCK
WORDS 1-177: DATA

DUMP FILE DATA - NON CONTROL WORDS IN THE DATA:

WORD COUNT WORDS ARE PLACED IN CONTIGUOUS BLOCKS, AND DATA ARE STORED ACROSS PHYSICAL BLOCK BOUNDARIES. THE FINAL BLOCK MAY NOT BE FILLED.

DUMP FILE DATA - CONTROL WORDS IN THE DATA:

WORD 0: COMMAND WORD. NEXT COMMAND WORD IS TAKEN FROM DECTAPE FILE

LOOKUP COMMAND

SEARCHES DIRECTORY FOR FILENAME AND FILENAME EXT (OR BLANK), AND FILLS DIRECTORY INFO INTO C(E,E+1,E+2,E+3). THE BLOCK NUMBER C(E+1 RIGHT) IS PLACED IN THE INPUT BLOCK NUMBER. CLEARS ALL ERROR BITS. AN EXAMPLE IS:

LOOKUP D,E ;E=LOCATION OF 4 WORD
JRST ERROR ;DIRECTORY BLOCK, ERROR
OK RETURN ;= NOT FOUND

ENTER COMMAND

SEARCHES DIRECTORY AND REPLACES OLD ITEM WITH NEW. A NEW ENTRY IS INSERTED IF NONE PRESENT. A FREE BLOCK IS PLACED IN C(E+1 RIGHT), AND OUTPUT BLOCK NUMBER CLEARS ALL ERROR BITS. AN EXAMPLE IS:

ENTER D,E ;E=LOCATION OF 4 WORD
JRST ERROR ;DIRECTORY BLOCK, ERROR=
OK RETURN ;NO DIRECTORY SPACE.

MISCELLANEOUS COMMANDS

USETI D,E ;SETS DECTAPE INPUT BLOCK NUMBER TO E.
USETO D,E ;SETS DECTAPE OUTPUT BLOCK NUMBER TO E
UGETF D,E ;GET THE NEXT FREE BLOCK NUMBER, SET THE OUTPUT
;BLOCK NUMBER (USETO), SET BLOCK
;IN C(E RIGHT), AND INCREMENT THE NEXT FREE BLOCK.
CLOSE D,X ;X=1(CLOSE OUTPUT),X=2(CLOSE INPUT),X=3(DO NOTHING)
;CLOSE PRESENT FILE OF DECTAPE
;WRITE FINAL BUFFER IF IT WERE LINKED TO A FREE
;BLOCK TERMINATE FILE.
RELEAS D,X ;RELEASES DEVICE, THE DIRECTORY
;IS REWRITTEN IF NECESSARY

NON DUMP OUTPUT COMMANDS
THE BUFFER LINK VALUE DETERMINES BLOCK ADDRESSES FOR OUTPUT
AS FOLLOWS:

VALUE	ACTION
Ø	REPLACE LINK WITH NUMBER OF NEXT FREE BLOCK ON TAPE, WRITE AT CURRENT OUTPUT BLOCK NUMBER AND SET NEW CURRENT OUTPUT BLOCK NUMBER TO NEXT FREE BLOCK (RE-LINK).
1	LAST BLOCK OF FILE. SET LINK TO Ø AND WRITE
N	ACTUAL LINK REQUESTED BY USER. SET NEXT OUTPUT BLOCK TO N. (N=2-1100.)

USETO MAY MODIFY BLOCK NUMBER.

NON DUMP INPUT COMMANDS
SUCCESSIVE BLOCKS OF THE TAPE ARE READ INTO BUFFER(S), USING
LINK TO LOCATE NEXT BLOCK.

USETI MAY MODIFY NEXT BLOCK NUMBER.

DUMP MODE INPUT OUTPUT COMMANDS
DUMP MODE IS CODE 17. THE COMMAND LIST FOR INPUT MAY BE EITHER
ON CORE OR ON TAPE. THE COMMANDS ARE SHOWN BY EXAMPLE:

INPUT D,E ;INPUT AS DUMP FILE ACCORDING TO COMMAND LIST
AT E.
INPUT D, ;INPUT ON DEVICE, TAKE COMMANDS FROM THE FILE.
;FIRST COMMAND IS FIRST WORD OF FILE.
OUTPUT D,E ;OUTPUT DUMP FILE ACCORDING TO COMMAND
;LIST AT E.

MAGNETIC TAPE

MAGNETIC TAPE DATA FORMAT

REFLECTIVE STRIP (BEGINNING OF TAPE, BOT)
DIRECTORY ENTRY RECORD (4 WORDS)
FIRST RECORD, FIRST FILE
.
.
LAST RECORD, FIRST FILE
END OF FILE
.
.
LAST RECORD, FIRST FILE
.
.
LAST RECORD, LAST FILE
END OF FILE
END OF FILE (SIGNIFIES LOGICAL END OF TAPE, EOT)

DIRECTORY FORMAT

WORD 0: FILE NAME (6-SIXBIT CHARS)
WORD 1: C(LH)=FILE EXTENSION (3-SIXBIT)
WORD 2: BIT 3-6=MODE
WORD 3: C(LH)=WORD COUNT FOR DUMP FILE, C(RH)= DUMP ADDRESS

PHYSICAL RECORDS FORMAT

ONE PHYSICAL RECORD/DUMP FILE COMMAND IN DUMP FILES
FIRST WORD OF RECORD (128 WORDS) CONTAINS NUMBER OF ITEMS
IN RECORD.

ENTER COMMAND - ENTER D, E
SET TAPE FOR OPERATION ON A PARTICULAR FILE

ENTER D,E

ENTER CAUSES THE TAPE TO BE BACKSPACED ONE FILE OR TO THE BEGINNING
OF TAPE (BOT). THE FOUR WORD FILE HEADER STARTING AT E IS WRITTEN
ON TAPE AS A RECORD AND THE TAPE IS POSITIONED FOR WRITING DATA.

LOOKUP COMMAND - LOOKUP D,E

LOOKUP CAUSES A BACKSPACE TO THE BEGINNING OF THE NEAREST FILE
AND A FORWARD SEARCH FROM THAT POINT FOR THE FILE NAME CONTAINED
IN THE FILE HEADER AT E. THE ERROR RETURN IS TAKEN IF A DOUBLE
END OF FILE IS REACHED. IF THE SEARCH IS SUCCESSFUL, THE TAPE IS
POSITIONED FOR READING OR WRITING THE DATA IN THE FILE.

AN EXAMPLE OF LOOKUP IS:

LOOKUP D,E ;E=HEADER TO LOOKUP
JRST ERROR ;NOT FOUND ON TAPE
FOUND EXIT

DATA TRANSMISSION COMMANDS - DUMP MODE
DUMP MODE =(16). DUMP DOES NOT RESTART USER UNTIL DATA TRANSFER
IS COMPLETE.
INPUT D,E INPUT ACCORDING TO THE COMMAND LIST AT E.
OUTPUT D,E OUTPUT ACCORDING TO THE COMMAND LIST AT E.

MTAPE MOVEMENT COMMAND - MTAPE D,FCN

FCN VALUE TAPE FUNCTION

1	REWIND
11	REWIND AND UNLOAD
7	BACKSPACE RECORD
17	BACKSPACE FILE
3	WRITE END OF FILE
6	SKIP RECORD
13	WRITE BLANK TAPE (3")
16	SKIP FILE
10	SPACE TO LOGICAL END OF TAPE

CLOSE COMMAND - CLOSE,D

INPUT	(NONDUMP) SET TO NEXT BUFFER
INPUT	(DUMP) SPACE TO EOF IF DUMP
OUTPUT	DUMP CURRENT BUFFER, WRITE TWO EOFS, BACKSPACE ONE EOF.

GET DUMP WORD COUNT COMMAND - GETN D,E
PLACE WORD COUNT FOR FILE IN C(E).

DISPLAY

COMMAND LIST OPERATION

THE DISPLAY HAS A COMPLETE INSTRUCTION SET FOR DISPLAYING PICTURES AND ITS COMMANDS ARE EXECUTED INTERRUPTIVELY BY THE DISPLAY SERVICE ROUTINE. THE COMMANDS PROVIDE FOR DISPLAYING DATA TRANSMISSION OF CONTROL WORD (OR INDEX VALUES), INSTRUCTIONS TO ALLOW SUBROUTINES TO BE WRITTEN IN THE COMMAND LANGUAGE, LIGHT PEN TRACKING, ETC. AND TO DISPLAY A COMPLETE PICTURE.

A COMMAND TO DISPLAY A LIGHT PEN POSITIONAL TRACKING CROSS. EACH TIME A LIGHT PEN TRACKING INSTRUCTION IS GIVEN, THE CO-ORDINATES OF THE LIGHT PEN ARE UPDATED.

THE DISPLAY IS STARTED BY ISSUING AN OUTPUT D,ADR INSTRUCTION WITH THE ADDRESS, ADR, OF A DATA BLOCK OF DISPLAY STATUS REGISTERS. THESE DISPLAY REGISTERS COMPLETELY HOLD THE STATE OF THE USER'S DISPLAY. THE DISPLAY AND USER'S PROGRAMS (WHICH INITIATED THE USER'S DISPLAY) OPERATED CONCURRENTLY.

THE DISPLAY REGISTERS

DISLCT DISPLAY LOCATION COUNTER WHICH POINTS TO USER'S
 DISPLAY INSTRUCTION CURRENTLY BEING EXECUTED.

DISBLK CURRENT IO POINTER WORD TO BLOCK OF DATA BEING DISPLAYED.

LPPOS LIGHT PEN TRACKING CROSS POSITION, BIT 0=1 IF SEEN.

LPDS LIGHT PEN DATA STATUS WORD.

DISPD PUSH DOWN POINTER FOR LOCATION COUNTER.

DISPLAY COMMANDS

IOWD N, A ;DISPLAY N DATA WORDS BEGINNING AT C(A). LOAD

XWD C, B ;DISBLK WITH C(A).
 ;THE DATA FORMAT IS THE 346 DISPLAY WORD DATA.
 ;PLACE THE FIRST FOUR WORDS OF THE DISPLAY REGISTERS
 ;IN C(B,B+1,B+2,B+3). JUMP TO C IF LIGHT PEN IS SEEN.

XWD 0, A ;JUMP TO ADDRESS A.

Ø ;STOP

XWD 2, 0 ;UPDATE LIGHT PEN TRACKING POSITION CROSS.
 ;LPPOS BIT0 IS RESET TOO.

XWD 4, A ;TRANSFER A SOURCE WORD IN MEMORY TO A DESTINATION.
 ;C(A LEFT) = SOURCE MEMORY WORD LOCATION.
 ;C(A RIGHT) = DESTINATION MEMORY WORD LOCATION.

XWD 5, A ;TRANSFER RIGHT HALF OF WORD ONLY.

XWD 6, A ;SUBTRACT 1 FROM C(A). SKIP NEXT INSTRUCTION
 ;IF C(A) IS LESS THAN OR EQUAL TO Ø.

XWD 10, A ;SKIP IF C(A) IS GREATER THAN OR EQUAL TO Ø.

```

XWD 11, A      ;ADD ONE TO C(A).
XWD 12, A      ;PUSH DOWN JUMP TO A.  C(DISPD)+1←C(DISPD).
                ;C(DISLCT)←C(DISPD).  C(DISLCT)←A.
XWD 13, 0      ;RETURN JUMP FROM PUSHDOWN C(DISLCT)←C(DISPD).
                ;C(DISPD)←C(DISPD)-1.

```

```

LPPOS - LIGHT PEN TRACKING STATUS WORD
      BIT 0      IF 1, LIGHT PEN WAS SEEN.

```

```

      BITS 8-13   X-POSITION

```

```

      BITS 24-35  Y-POSITION

```

```

LPDS - LIGHT PEN DATA STATUS WORD
      BIT 0      IF 1, LIGHT PEN WAS SEEN

```

```

      BITS 8-17   X POSITION OF PEN.

```

```

      BITS 24-35  Y POSITION OF PEN

```

```

CLOSE COMMAND
      STOPS THE DISPLAY

```

C H A P T E R V
M I S C E L L A N E O U S M O N I T O R F U N C T I O N S

THESE FUNCTIONS ARE AVAILABLE AS EITHER "CALL UUOS" OR ARE GENERALLY PERTINENT TO OPERATION OF THE SYSTEM.

"CALL" PROGRAM OPERATORS OR UUOS.

THESE FUNCTIONS HAVE THE FORM :

CALL AC, [SIXBIT /FUNCT/]

WHERE "FUNCT" IS THE NAME OF THE MONITOR ROUTINE WHICH PERFORMS THE ACTION. THE AC, OR REGISTERS FOLLOWING THE "CALL" MAY CARRY ARGUMENTS, ETC.

CALL "DATE"

CALL AC, [SIXBIT.DATE.]

RETURNS WITH THE DATE IN ACCUMULATOR AC, IN THE FORM:
(((Y-1964) 12+(M-1))31+(D-1))
WITH A 12 BIT NUMBER.

C H A P T E R V I
T H E L O A D E R

GENERAL

1. INITIALLY RESIDENT IN USER'S AREA. CONTROLLED BY A USER STATION.
2. LOADS AND LINKS RELOCATABLE BINARY FORMAT PROGRAMS ASSEMBLED BY MACRO6 OR FOLA (1 PASS FORTRAN ASSEMBLER), PROVIDING SPACE FOR FORTRAN COMMON, SEARCHES LIBRARY TAPES, AND LOADS OTHER FILES, INDEPENDENT OF IO INPUT MEDIA.
3. PROVIDES LOADED PROGRAM TO RUN, SYMBOL TABLE IN CORE FOR DDT, AND CREATES STORAGE MAP OF PROGRAMS FOR USER.
4. SIZE: 2K (OCTAL) WITH IO BUFFERS. (STORAGE IS RECOVERABLE JUST BEFORE RUNNING)
5. EQUIPMENT: USER TELETYPE, IO DEVICE FOR FILES FROM WHICH TO LOAD, IO DEVICE FOR LOADER MAP.

COMMANDS TO LOADER FROM TELETYPE

CONTROL MODE SWITCHES (ENCLOSED IN PARENTHESES)

S-	LOAD LOCAL SYMBOLS
W	DO NOT LOAD LOCAL SYMBOLS
L-	SEARCH FILES IN LIBRARY SEARCH MODE
N-	STOP LIBRARY SEARCH MODE
A-	IGNORE ALL SUBSEQUENT STARTING ADDRESSES
M-	PRINT STORAGE MAP
R-	RESTART LOADING
I-	SET LOADER TO CURRENT STATE. THIS STATE WILL BE RETURNED TO IF R IS TYPED. (OTHER CONTROL CHARACTERS)
'-	ENTER LIBRARY SEARCH FOR THE PREVIOUS FILE
ALT MOD	GO INTO EXECUTION

COMMAND FORMAT

STORAGE MAP FILE!(SWITCHES) SOURCE FILE 1,...(SWITCHES) SOURCE FILE N

EXAMPLE

TTY: +DTA4: (S)PG1,PG2, (WL)LB1,ARTLIB, (N)TSTDAT, (M)

LOADS FROM DTA4 FILES

PG1,PG2 WITH LOCALS
LB1,ARTLIB WITHOUT LOCALS AS LIBRARY FILES
TSTDAT IS LOADED WITHOUT LOCAL SYMBOLS
DESTINATION FILE IS STORAGE MAP.
(NO DESTINATION FILE IMPLIES TTY)

ERROR MESSAGES

ILLEGAL FORMAT

INDICATES THAT THE BINARY INPUT TO THE LOADER IS NOT IN PROPER FORM, AND THEREFORE CANNOT BE LOADED.

DATA ERROR

INDICATES THAT DATA CANNOT BE READ BECAUSE OF ILLEGAL CHECKSUM, BAD PARITY, ETC.

CORE EXCEEDED

THIS MESSAGE IS TYPED WHEN THE HIGHEST NUMBERED LOCATION LOADED IS GREATER THAN OR EQUAL TO THE LOWEST NUMBERED LOCATION IN THE LOADER SYMBOL TABLE. NO FURTHER LOADING IS POSSIBLE. PROGRAM EXECUTION MAY BE POSSIBLE IF THE AREA OVERLOAPPING THE SYMBOL TABLE DOES NOT CONTAIN CONSTANTS OR INSTRUCTIONS AND DDT IS NOT USED.

ILLEGAL COMMON

THIS MESSAGE IS TYPED IF A PROGRAM TRIES TO INCREASE THE LENGTH OF COMMON. THIS ERROR CAN BE AVOIDED BY REORDERING THE BINARY INPUT TO THE LOADER SO THAT THE PROGRAM WITH THE LONGEST COMMON AREA IS LOADED FIRST.

COMMAND ERROR

THIS MESSAGE IS TYPED IF A FILE OR DEVICE NAME CONTAINS TOO MANY CHARACTERS, OR IF AN INPUT FILE MODE OTHER THAN BINARY IS SPECIFIED.

FILE 'X' NOT FOUND

THIS MESSAGE INDICATES THAT FILE 'X' IS NOT IN THE DECTAPE DIRECTORY.

ALL ERRORS WITH THE EXCEPTION OF FILE 'X' NOT FOUND, CAUSE THE REST OF THE INPUT COMMAND STRING TO BE IGNORED.

DEFINITIONS

1. GLOBAL SYMBOL - ANY SYMBOL ACCESSIBLE TO OTHER PROGRAMS.
2. GLOBAL REQUEST - REQUEST TO THE LOADER TO LINK A GLOBAL TO A PROGRAM. REQUEST MAY BE ADDITIVE OR CHAINED.
 - A. CHAINED REQUEST:
A REQUEST WHICH POINTS TO THE LAST REFERENCE IN THE PROGRAM TO THE GLOBAL SYMBOL REQUESTED. EACH REFERENCE IN THE PROGRAM POINTS TO THE PREVIOUS REFERENCE TO THE REQUESTED GLOBAL. SUCH A CHAIN IS TERMINATED BY A NON-RELOCATABLE ZERO ADDRESS IN THE PROGRAM. CHAINED GLOBALS ARE RESTRICTED

TO REFERENCES APPEARING IN THE ADDRESS PART OF A STORAGE WORD. SYMBOLIC REFERENCES TO THE AC OR INDEX FIELDS CANNOT BE CHAINED.

- B. ADDITIVE REQUEST:
A REQUEST WHICH FOR SOME REASON CANNOT BE CHAINED. THE REQUEST POINTS TO A STORAGE WORD WHICH IS TO HAVE THE REQUESTED GLOBAL ADDED IN THE WORD AFTER APPROPRIATE SHIFTING AND MASKING. EACH REFERENCE TO A GLOBAL THAT CANNOT BE CHAINED REQUIRES A SEPARATE ADDITIVE REQUEST. THE ADDITIVE REQUEST CARRIES SUFFICIENT INFORMATION SO THAT A GLOBAL VALUE CAN BE PLACED IN THE AC OR INDEX FIELD OF AN INSTRUCTION.

3. PROGRAM ORIGIN - THE LOCATION ASSIGNED BY THE LOADER TO RELOCATABLE ZERO OF A PROGRAM.
4. THE PROGRAM BREAK - THE LENGTH OF A PROGRAM, THE FIRST LOCATION NOT USED BY A PROGRAM (BEFORE RELOCATION) THE RELOCATION CONSTANT FOR THE FOLLOWING PROGRAM (AFTER RELOCATION).
5. OFFSET - THE NUMBER OF LOCATIONS TOWARD ZERO, A PROGRAM MUST BE MOVED BEFORE IT CAN BE EXECUTED.
6. RELOCATION CONSTANT - THE NUMBER ADDED TO EVERY RELOCATABLE REFERENCE WITHIN A PROGRAM. THE RELOCATION CONSTANT IS THE RELOCATED PROGRAM BREAK OF THE PREVIOUS PROGRAM.
7. INTERNAL SYMBOL - GENERATES A GLOBAL DEFINITION WHICH IS USED TO SATISFY ALL GLOBAL REQUESTS FOR THAT SYMBOL.
8. EXTERNAL SYMBOL - GENERATES A CHAINED GLOBAL REQUEST AND ONE OR MORE ADDITIVE GLOBAL REQUESTS.
9. COMMON - AREA SET ASIDE FOR COMMON STORAGE BEFORE THE BEGINNING OF A PROGRAM.
10. THE LIBRARY SEARCH FEATURE - THE LOADER WILL, ON REQUEST, SEARCH A FILE AS A LIBRARY FILE. THIS MEANS THAT A PROGRAM IS LOADED ONLY IF ONE OF ITS ENTRIES MATCHES AN UNSATISFIED GLOBAL SYMBOL REQUEST. AN ENTRY IS SPECIFIED IN THE ASSEMBLER BY USE OF THE ENTRY PSEUDO-OP. ALL FORTRAN FUNCTION AND SUBROUTINE SUB-PROGRAMS ARE COMPILED WITH THE NAME OF THE PROGRAM AS AN ENTRY. THUS A COLLECTION OF PROGRAMS INCLUDING FORTRAN COMPILED PROGRAMS CAN BE LOADED SELECTIVELY FROM A TAPE.
THE SYSTEM LIBRARY IS CURRENTLY SEARCHED ONLY ON REQUEST. THIS FILE CAN BE SEARCHED BY REQUESTING THE FILE DECLIB FROM THE TAPE CONTAINING THE SYSTEM LIBRARY (USUALLY DTA1).

LINKING LOADER INPUT FORMAT

ALL DATA TO THE LOADER IS IN BLOCKS. THE FIRST WORD OF EACH BLOCK CONTAINS AN OCTAL CODE FOR THE BLOCK TYPE IN THE LEFT HALF AND THE WORD COUNT (POSITIVE) IN THE RIGHT HALF. FOLLOWING THE HEADER WORD ARE N SUB-BLOCKS, EACH OF WHICH CONTAINS A SUB-HEADER PLUS 18 OR FEWER DATA WORDS. THESE SUB-HEADERS ARE NOT INCLUDED IN THE WORD COUNT IN THE BLOCK HEADER. THEY CONTAIN 18 TWO-BIT BYTES WHICH ARE THE RELOCATION BITS FOR DATA WORDS WITHIN THAT SUB-BLOCK. THE RELOCATION CODES ARE:

- 0 - DO NOT RELOCATE
- 1 - RELOCATE RIGHT HALF OF THE STORAGE WORD
- 2 - RELOCATE LEFT HALF OF THE STORAGE WORD
- 3 - RELOCATE BOTH HALVES

THE BLOCK FORMAT IS CONSTANT FOR INPUT FROM ALL DEVICES, IT IS THE SAME FOR PAPER TAPE, BINARY CARDS, DECTAPE, DRUMS, ETC., AND IS INDEPENDENT OF PHYSICAL DIVISIONS IN THE INPUT MEDIA.

BLOCK TYPE 1: RELOCATABLE OR ABSOLUTE PROGRAMS AND DATA.
 WORD 1 THE LOCATION OF THE FIRST DATA WORD IN THE BLOCK.
 WORD 2 A CONTIGUOUS BLOCK OF PROGRAM OR DATA WORDS.
 .
 .
 .
 WORD N (N MUST BE LESS THAN 200,000 OCTAL)

BLOCK TYPE 2: SYMBOLS
 CONSISTS OF WORD PAIRS
 1ST WORD BITS 0-3 CODE BITS
 1ST WORD BITS 4-35 RADIX 50 REPRESENTATION OF SYMBOL
 2ND WORD DATA (VALUE OR POINTER)

CODE 04: GLOBAL (INTERNAL) DEFINITION
 2ND WORD BITS 0-35 VALUE OF SYMBOL

CODE 10: LOCAL DEFINITION
 2ND WORD BITS 0-35 VALUE OF SYMBOL

CODE 60: CHAINED GLOBAL REQUESTS:
 2ND WORD BITS 0-17 = 0
 2ND WORD BITS 18-35 POINTER TO FIRST WORD OF CHAIN REQUIRING DEFINITION

CODE 60: GLOBAL SYMBOL ADDITIVE REQUEST:
 2ND WORD BIT 0 = 1.
 BIT 1 SUBTRACT VALUE BEFORE ADDITION
 BIT 2 SWAP HALVES BEFORE ADDITION
 BIT 3 ROTATE LEFT 5 BEFORE ADDITION
 BIT 9 REPLACE LH WITH RESULT IN STORAGE
 BIT 10 REPLACE RH WITH RESULT IN STORAGE
 BIT 11 REPLACE INDEX FIELD WITH RESULT IN STORAGE
 BIT 12 REPLACE AC FIELD WITH RESULT IN STORAGE
 BITS 18-35 POINTER TO WORD REQUIRING ADDITION

BLOCK TYPE 4: ENTRY BLOCK
 THIS BLOCK CONTAINS A LIST OF RADIX 50 SYMBOLS, EACH OF WHICH MAY CONTAIN A ZERO OR ONE IN THE HIGH ORDER CODE BIT. EACH REPRESENTS A SERIES OF LOGICAL 'AND' CONDITIONS. IF ALL THE GLOBALS IN ANY SERIES ARE REQUESTED, THE FOLLOWING PROGRAM IS LOADED. OTHERWISE ALL INPUT IS IGNORED UNTIL THE NEXT END BLOCK. THIS BLOCK MUST BE THE FIRST BLOCK IN A PROGRAM.

BLOCK TYPE 5: END BLOCK
 THIS IS THE LAST BLOCK IN A PROGRAM. IT CONTAINS ONE WORD WHICH IS THE PROGRAM BREAK, THAT IS, THE LOCATION OF THE FIRST FREE REGISTER ABOVE THE PROGRAM. (NOTE: THIS WORD IS RELOCATABLE). IT IS THE RELOCATION CONSTANT FOR THE FOLLOWING

PROGRAM LOADED.

BLOCK TYPE 6: NAME BLOCK
THE FIRST WORD OF THIS BLOCK IS THE PROGRAM NAME (RADIX 50).
IT MUST APPEAR BEFORE ANY TYPE 2 BLOCKS. THE SECOND WORD IF
IT APPEARS DEFINES THE LENGTH OF COMMON.

BLOCK TYPE 7: STARTING ADDRESS
THE FIRST WORD OF THIS BLOCK IS THE STARTING ADDRESS OF THE
PROGRAM. THE LAST BLOCK OF THIS TYPE ENCOUNTERED BY THE
LOADER IS USED UNLESS THE CONTROL CHARACTER (A) HAS BEEN
TYPED. THE STARTING ADDRESS FOR A RELOCATABLE PROGRAM MAY
BE RELOCATED BY MEANS OF THE RELOCATION BITS.

BLOCK TYPE 10: INTERNAL REQUEST
EACH DATA WORD IS ONE REQUEST. THE LEFT HALF IS THE POINTER
TO THE PROGRAM. THE RIGHT HALF IS THE VALUE. EITHER QUANTITY
MAY BE RELOCATABLE.

LOADER MAP OUTPUT FORMAT

IF REQUESTED, THE LOADER WILL OUTPUT A LIST OF ALL PROGRAMS LOADED.
THIS WILL BE A LISTING OF ALL DEFINED GLOBALS AND THEIR VALUES, ALL
PROGRAM NAMES WITH THEIR ORIGINS AND LENGTHS, AND AT THE END. A LIST
OF ALL UNDEFINED GLOBALS, THE FORMAT IS:

PROG NAME ORIGIN LENGTH

GLOBAL VALUE

·
·
·

GLOBAL VALUE

PROG NAME ORIGIN LENGTH

GLOBAL VALUE

·
·
·

UNDEFINED GLOBALS LIST

USER AREA CORE STORAGE ALLOCATION FORMAT

RELOCATED PROGRAMS ARE LOADED INTO MEMORY FROM THE PROGRAM ORIGIN
UPWARD. THE PROGRAM ORIGIN IS THE LOWEST ADDRESS AVAILABLE FOR
LOADING PROGRAMS. IT IS NORMALLY 140 PLUS THE LENGTH OF COMMON, UNLESS
OTHERWISE ASSIGNED.

THE SYMBOL TABLE IS LOADED FROM THE HIGHEST AVAILABLE LOCATION IN
MEMORY DOWNWARD TOWARD THE PROGRAM AREA, WHICH IS BEING LOADED UP-
WARD. EACH SYMBOL REQUIRES TWO MEMORY LOCATIONS. THE LOWER NUMBERED
LOCATION CONTAINS A SYMBOL IN RADIX 50, WITH ITS CLASSIFICATION CODE
BITS, AND THE HIGHER LOCATION CONTAINS THE VALUE FOR THE SYMBOL.

CORE SYMBOL TABLE FORMAT

THE LOADER CREATES A SYMBOL TABLE FOR ITS USE WHILE LOADING,
AND FOR THE USE OF DDT WHILE DEBUGGING.
THE SYMBOL TABLE IS ORGANIZED INTO SEGMENTS HEADED BY A WORD PAIR CON-
SISTING OF A PROGRAM NAME AND ITS ORIGIN. FOLLOWING THE PROGRAM NAME

ARE WORD PAIRS CONTAINING A RADIX 50 SYMBOL PLUS FOUR CODE BITS IN ONE WORD AND A VALUE IN THE OTHER WORD.

LOCAL SYMBOLS ARE NOT PLACED INTO THE SYMBOL TABLE BY THE LINKING LOADER UNLESS A REQUEST IS MADE BY THE USER. LOCAL SYMBOLS ARE USED BY DDT. GLOBAL SYMBOLS ARE AUTOMATICALLY PLACED IN THE SYMBOL TABLE BY THE LOADER SO THAT INTER-PROGRAM LINKING CAN TAKE PLACE.

LOCAL SYMBOLS, GLOBAL SYMBOLS, AND PROGRAM NAMES ARE DISTINGUISHED FROM EACH OTHER BY FOUR CODE BITS IN BIT POSITIONS 0-3 OF THE WORD CONTAINING THE RADIX 50 SYMBOL. THEY ARE

CODE 00	PROGRAM NAME
CODE 04	GLOBAL SYMBOL (INTERNAL)
CODE 10	LOCAL SYMBOL
CODE 20	DELETED ON OUTPUT (DDT)
CODE 40	DELETED BY DDT OR THE LINKING LOADER
CODE 60	GLOBAL REQUEST (EXTERNAL)

GLOBAL SYMBOL REQUESTS HAVE THEIR CODE BITS CHANGED FROM 60 TO 40 AS THEY ARE SATISFIED. CODE 20 IS CREATED BY DDT ONLY. WHEN LOADING IS COMPLETE, THE LOADER LEAVES A SYMBOL TABLE POINTER IN JOBSYM (LOCATION 110 IN THE USER AREA) WHICH CONTAINS THE ADDRESS OF THE LOWEST NUMBERED LOCATION IN THE SYMBOL TABLE IN THE RIGHT HALF, AND THE NEGATIVE WORD COUNT OF THE SYMBOL TABLE IN THE LEFT HALF.

THE RADIX 50 REPRESENTATION OF SYMBOLS

THIS IS A SPACE SAVING DEVICE USED TO CONDENSE A SIX CHARACTER SYMBOL INTO 32 BITS. THE SIX LETTERS OF THE SYMBOL (RIGHT JUSTIFIED) ARE TRANSLATED INTO CODE NUMBERS. THE RIGHT HAND LETTER WILL BE DESIGNATED L1, AND ITS CORRESPONDING CODE NUMBER, C1, AND ACROSS FOR THE SIX LETTERS. THE RADIX 50 REPRESENTATION OF THE SYMBOL IS CALCULATED THUS:

$$((((C6*50)+C5*50+C4)*50+C3)*50+C2)*50+C1$$
 (OCTAL NUMBERS)
THE CODE NUMBERS CORRESPONDING TO THE SYMBOL CONSTITUENTS ARE:

CODE (OCTAL)	SYMBOL CHARACTER
00	BLANK
01-12	0-9
13-44	A-Z
45	.
46	\$
47	(

RESTRICTIONS FOR ONE PASS ASSEMBLY

IN A ONE PASS ASSEMBLER, SYMBOL TAGS MAY BE REFERENCED BEFORE THEY ARE ASSIGNED A VALUE. WHEN THEY ULTIMATELY BECOME DEFINED, THE REFERENCE TO THEM HAS ALREADY BEEN GENERATED, AND THEREFORE MUST BE RESOLVED AT LOAD TIME. THIS UNRESOLVED REFERENCE MAY ACTUALLY BE A SERIES OF REFERENCES. TO AVOID CREATING TABLES OF UNRESOLVED REFERENCES, THEY ARE CHAINED TOGETHER BY THE ASSEMBLER IN SUCH A FASHION THAT THE LOADER MAY EASILY FIND AND RESOLVE ALL REFERENCES TO A PARTICULAR SYMBOL TAG. THE FIRST INSTANCE IN WHICH A REFERENCE TO A YET UNDEFINED SYMBOL TAG OCCURS CAUSES A NON-RELOCATABLE 0 TO BE GENERATED

AS THE VALUE OF THE TAG AND THE ADDRESS OF THIS REFERENCE IS KEPT IN THE SYMBOL TABLE BY THE ASSEMBLER. SUCCEEDING REFERENCES TO THE SAME TAG AND THE ADDRESS OF THE REFERENCE REPLACES THE ADDRESS OF THE PREVIOUS REFERENCE IN THE ASSEMBLER'S SYMBOL TABLE THEREBY CREATING A CHAIN OF REFERENCES TO BE RESOLVED BY THE LOADER.

RESTRICTIONS FOR GLOBAL REQUESTS

GLOBAL SYMBOL REQUESTS OF THE ADDITIVE TYPE WILL BE ACCUMULATED BY THE ASSEMBLER AND OUTPUT IN A SEPARATE BLOCK (TYPE 2) WHENEVER NECESSARY. THIS BLOCK CAN OCCUR ANYWHERE ON A BINARY TAPE WITHIN A PROGRAM SEGMENT HAVING THE SAME RELOCATION CONSTANT AS THE DATA WORDS POINTED TO BY THE ADDITIVE REQUESTS. A DATA WORD POINTED TO BY AN ADDITIVE REQUEST MUST OCCUR ON THE BINARY TAPE BEFORE THE ADDITIVE REQUEST. THE LOADER WILL CHECK BIT ZERO OF A GLOBAL REQUEST TO DETERMINE WHETHER IT IS ADDITIVE OR CHAINED. A CHAIN IS A SERIES OF REFERENCES, EACH OF WHICH POINTS TO THE PREVIOUS REFERENCE. A CHAIN IS TERMINATED BY A NON-RELOCATABLE ZERO.

RESTRICTIONS FOR COMMON

THE PROGRAM WITH THE LONGEST DECLARED COMMON AREA MUST BE LOCATED FIRST. COMMON STARTS AT LOCATION 140. PROGRAMS ARE LOADED STARTING AT THE COMMON BREAK (THE FIRST FREE LOCATION ABOVE THE COMMON AREA). THEREFORE, ONCE LOADING HAS STARTED, THE COMMON AREA CANNOT BE EXPANDED.

CHAPTER VI

THE EDITOR

GENERAL

THE DECTAPE EDITOR PROVIDES A MEANS OF CREATING, ADDING TO, OR DELETING FROM SEQUENCE-NUMBERED LINES IN FILES ON DECTAPE. THIS TEXT MAY BE INTENDED AS INPUT FOR THE FORTRAN COMPILER, THE MACRO6 ASSEMBLER, OR JUST AS A CONVENIENT MEANS FOR HANDLING TEXTUAL OR LINES OF INFORMATION.

THE CHARACTERISTICS OF THE EDITOR

SIZE: 2K OCTAL
EQUIPMENT: USER TTY, 1 DECTAPE
LOADING: RUN AS USER PROGRAM,

USAGE

THE EDITOR PROVIDES THE FACILITY FOR SELECTING A TAPE UNIT, EXAMINING THE DIRECTORY (WHICH LISTS ALL THE FILES CURRENTLY ON THE TAPES), CLEARING THE DIRECTORY, ADDING A NEW FILE TO THE DIRECTORY, OR SELECTING A FILE CURRENTLY ON THE DIRECTORY. WHEN A FILE HAS BEEN SELECTED, THE USER MAY RESEQUENCE IT, PRINT A LINE OR MANY LINES, ENTER NEW LINES, DELETE EXISTING LINES, OR REPLACE EXISTING LINES.

ONCE CONTROL IS TURNED OVER TO THE EDITOR, THE EDITOR OPERATES IN EITHER OF TWO MODES: COMMAND MODE OR TEXT MODE. THE COMMAND MODE IS THE MODE IN WHICH THE USER ISSUES INSTRUCTIONS TO THE EDITOR.

COMMANDS

SX CONTROL A (CR)	(SELECT UNIT X AND CLEAR THE DIRECTORY.)
SX,NAME CONTROL A ALTMODE	(SAME AS ABOVE BUT CREATES FILE NAME AFTER CLEARING.)
SX,NAME (CR)	(ALLOW THE USER TO ACCESS FILE NAME ON UNIT X. IF NO SUCH FILE EXISTS, THE ERROR *NCF* WILL BE GIVEN.)
SX,NAME ALTMODE	(CREATE A NEW FILE CALLED NAME ON UNIT X. IF NAME ALREADY EXISTS THE ERROR *FAU* WILL BE GIVEN.)
E (CR)	(END THE CURRENT FILE.)
IN1 (CR)	(INSERT AT LINE N1, WHERE N1 MAY BE A SEQUENCE NUMBER OR THE SPECIAL CHARACTER . (POINT). IF N1 IS AN EXISTING LINE, IT IS REPLACED.)
IN1,I (CR)	(INSERT AT LINE N1, (AS ABOVE) BUT SET THE INDEXING INCREMENT TO I.)

DN1 (DELETE LINE N1.)

DN1,N2 (CR) (DELETE LINES N1 THRU N2.)

PN1 (CR) (PRINT LINE N1.)

PN1,N2 (CR) (PRINT LINES N1 THRU N2.)

RX (CR) (RESEQUENCE THE CURRENT FILE, STARTING AT X AND STEPPING BY X.)

RX,Y (CR) (RESEQUENCE THE CURRENT FILE, STARTING AT X AND STEPPING BY Y.)

THE TEXT MODE IS ENTERED AUTOMATICALLY BY THE INSERT COMMAND. THE SEQUENCE NUMBER OF THE LINE BEING REFERENCED WILL BE PRINTED FOLLOWED BY A TAB. THE LINE BEING INSERTED IS THEN TYPED IN AND TERMINATED BY A CARRIAGE RETURN. THIS LINE MAY CONTAIN ANY CHARACTER EXCEPT RUBOUT AND ALTMODE. AFTER THE LINE BEING INSERTED HAS BEEN TERMINATED BY A CARRIAGE RETURN, THE INDEXING INCREMENT IS ADDED TO THE LINE NUMBERS AND THE INCREMENTED LINE NUMBER IS THEN PRINTED, FOLLOWED BY A TAB, AND THE EDITOR IS READY TO ACCEPT THE NEXT INSERTION. THE INDEXING INCREMENT IS INITIALLY SET TO 10.

TO CORRECT A TYPING ERROR, TYPE RUBOUT TO DELETE THE OFFENDING CHARACTER. THE EDITOR WILL RESPOND BY TYPING A \. SUCCESSIVE CHARACTERS MAY BE DELETED BY TYPING MORE RUBOUTS, UNTIL THE BEGINNING OF THE LINE HAS BEEN REACHED, AT WHICH POINT THE EDITOR WILL CEASE RESPONDING WITH \.

TO RETURN FROM THE TEXT MODE TO THE COMMAND MODE, TYPE ALTMODE.

CARRIAGE RETURN (ENDS A COMMAND AND CAUSES EXECUTION. NULL COMMANDS ARE IGNORED.

. (POINT. HAS THE VALUE OF THE LAST LINE NUMBER TYPED.)

ALTMODE (SPECIAL DELINEATOR FOR S; ALSO, WHEN USED ALONE, ADVANCES . AND PRINTS THE LINE.)

ERROR MESSAGES

- *ILC* ILLEGAL COMMAND
- *NLN* A P OR D HAS MADE REFERENCE TO A NONEXISTENT LINE.
- *ILS* THE INDEX STEP (IN INSERT) HAS CAUSED A LINE L
- *ILS* THE INDEX STEP (IN INSERT) HAS CAUSED A LINE ALREADY IN THE FILE TO BE SKIPPED OVER.
- *ILR* THE INDEX STEP HAS CAUSED AN ALREADY EXISTENT LINE TO BE REFERENCED BY INSERT.
- *UNA* A DECTAPE UNIT HAS BEEN SELECTED THAT IS NOT AVAILABLE TO THE USER.
- *DCE* NO MORE ROOM ON THE DECTAPE DIRECTORY FILE.
- *NFO* A COMMAND HAS BEEN INVOKED THAT REQUIRES AN ACTIVE FILE,

BUT NO FILE HAS BEEN SELECTED (BY S).

FAU FILE NAME ALREADY IN USE.

NCF NOT A CURRENT FILE.

RESTRICTIONS

ONLY EDITS SEQUENCED MODE FILES.

C H A P T E R V I I I
T H E P E R I P H E R A L I N T E R C H A N G E P R O G R A M
(P I P)

GENERAL

PIP1 AND PIP2 ARE PDP-6 PROGRAMS TO TRANSFER DATA FROM ONE I/O MEDIUM TO ANOTHER. PIP PERFORMS SIMPLE EDITING FUNCTIONS DURING DATA TRANSMISSION.

CHARACTERISTICS OF PIP1, PIP2

SIZE: PIP1=2K, PIP2=4K OR MORE OCTAL
EQUIPMENT: TELETYPE FOR CONTROL, OTHER DEVICES AS REQUIRED.
LOADING: RUN AS USER PROGRAM, PIP2 MAY BE READ INTO CORE GREATER THAN 4K

USAGE

COMMANDS ACKNOWLEDGED BY PIP ARE COMPRISED OF DESTINATION AND SOURCE INFORMATION. DESTINATION IS SEPARATED FROM SOURCE BY THE SPECIAL CHARACTER (+). MULTIPLE SOURCES MAY BE INCLUDED IN ONE COMMAND LINE.

PIP RESTRICTIONS: THE * COMMANDS BELOW ARE NOT AVAILABLE IN PIP1.

COMMANDS FORM

DESTINATION (SPECIAL OPERATORS)+SOURCE,SOURCE,...,SOURCE

THE DESTINATION MAY BE ONE OF THE FOLLOWING FORMS:

- 1) DEVICENAME:
- 2) DEVICENAME:FILENAME
- 3) DEVICENAME:FILENAME.EXTENSION

DEVICE NAMES ARE THE SAME FORM AS THOSE USED BY THE I/O SERVICE ROUTINES. FOR EXAMPLE, PTP IS THE PAPER TAPE PUNCH, DTA3 IS DECTAPE UNIT 3. FILE NAMES AND EXTENSIONS ARE RELEVANT ONLY FOR DEVICES THAT CONTAIN MULTIPLE FILES SUCH AS DECTAPE. FILE NAMES AND EXTENSIONS ARE CONSTRUCTED OF LETTERS AND NUMERALS. ONLY THE FIRST SIX CHARACTERS OF THE FILE NAME AND THE FIRST THREE LETTERS OF THE EXTENSION ARE USED.

SOURCES HAVE THE SAME FORM AS THE DESTINATION WITH THE EXCEPTION THAT IF THE DEVICE NAME IS OMITTED, THE LAST DEVICE NAME TYPED IS USED.

CARRIAGE RETURN TERMINATES THE COMMAND REQUEST.

RUBOUT MAY BE USED TO ERASE SUCCESSIVE CHARACTERS. FOR EACH CHARACTER ERASED, ONE REVERSE SLASH IS TYPED UNTIL ALL CHARACTERS ARE ERASED.

SPECIAL OPERATORS ARE LETTERS ENCLOSED IN PARENTHESIS().
IF ONLY ONE SPECIAL OPERATOR IS TYPED PRECEEDING SLASH

CHARACTER MAY REPLACE BOTH PARENTHESES.
IF SPECIAL OPERATORS ARE OMITTED, PIP TRANSMITS THE DATA
FROM EACH SUCCESSIVE SOURCE TO THE DESTINATION WITHOUT ALTERATION EXCEPT
THAT WHEN AN UNSEQUENCED SOURCE FILE FOLLOWS A SEQUENCED FILE,
SEQUENCE NUMBERS ARE ADDED TO THE UNSEQUENCED FILE AUTOMATICALLY.

PIP RECOGNIZES THE FOLLOWING SPECIAL OPERATORS:

SPEC. OPER.	ACTION
A	ASCII LINE MODE PROCESSING. (DO NOT BREAK UP LINES BETWEEN BUFFERS.)
B	PROCESS BINARY DATA
C*	SUPPRESS TRAILING SPACES AT THE END OF EACH LINE AND CHANGE MULTIPLE BLANKS INTO TABS. (USEFULL FOR CARD INPUT)
D*	DELETE ALL SOURCE FILE NAMES FROM THE DIRECTORY OF THE DESTINATION DEVICE. THE DESTINATION DEVICE NAME SUPERCEDES ANY OTHER DEVICE NAME THAT IS TYPED.
L*	LIST THE DIRECTORY OF THE SOURCE DEVICE. ONLY ONE SOURCE MAY BE SPECIFIED.
N	REMOVE SEQUENCE NUMBERS
R*	RENAME THE SOURCE FILE TO HAVE THE NAME OF THE DESTINATION FILE. NO SOURCE DEVICE NAME IS NECESSARY BECAUSE THE DESTINATION DEVICE IS IMPLIED. ONLY ONE SOURCE MAY BE SPECIFIED.
S	A SEQUENCE NUMBER STARTING WITH 00010 AND INCREMENTING BY 10 IS ADDED TO THE FRONT OF EACH LINE. A TAB IS INSERTED AFTER THE SEQUENCE NUMBER UNLESS THE LINE CONSISTS ENTIRELY OF FORM FEED, VERTICAL TAB, OR LINE FEED, IN WHICH CASE A CARRIAGE RETURN IS INSERTED. IF A FILE IS ALREADY SEQUENCED, THE OLD SEQUENCE NUMBERS ARE REMOVED.
T	THE SAME AS C EXCEPT DO NOT REPLACE MULTIPLE BLANKS WITH TABS.
W	REWIND MAGNETIC TAPE. SOURCE FILES ARE IGNORED.
X*	COPY AN ENTIRE DECTAPE, RECOVERING UNUSED BLOCKS. EMPTY FILES ARE NOT COPIED. DOES NOT DISTURB DATA PREVIOUSLY WRITTEN ON THE DESTINATION TAPE UNLESS FILE NAMES CONFLICT. ALTHOUGH THE SOURCE DEVICE MUST BE DECTAPE, THE DESTINATION MAY BE ANY OUTPUT DEVICE.
Z	CLEAR THE DESTINATION DECTAPE AND CREATE A FRESH DIRECTORY. SOURCES ARE IGNORED.

ERROR MESSAGES

DEVICE XXX NOT AVAILABLE

THE DEVICE NAMED XXX IS NOT AVAILABLE

EITHER BECAUSE IT DOES NOT EXIST OR
SOME OTHER USER IS USING IT.

NO FILE NAMED XXX

THE SOURCE FILE NAMED XXX WAS NOT FOUND
IN THE DIRECTORY.

DIRECTORY FULL

THE DESTINATION DEVICE HAS NO ROOM IN THE
DIRECTORY FOR ANOTHER FILE NAME. THE ONLY
RECOURSE IS TO DELETE ONE OF THE PREVIOUS
FILES ON THE DESTINATION TAPE TO MAKE ROOM
IN THE DIRECTORY.

DECTAPE INPUT REQUIRED

A DIRECTORY MANIPULATION COMMAND WAS GIVEN
FOR A DEVICE OTHER THAN DECTAPE.

EXAMPLES

LPT:+DTA1:ABC.TXT,XYZ

THE TWO FILES ON DECTAPE UNIT 1 NAMED
ABC.TXT AND XYZ ARE LISTED ON THE LINE-
PRINTER.

DTA2:SUBR/S+PTR:

A FILE NAMED SUBR IS CREATED ON DECTAPE
UNIT 2 CONTAINING THE ASCII DATA FROM THE
TAPE IN THE PAPER TAPE READER WITH SE-
QUENCE NUMBERS ADDED.

DTA1:(D)+ABC.TXT,XYZ

DELETE THE FILES NAMED ABC.TXT AND XYZ

EXAMPLES: PIP1 AND PIP2

TO COPY THREE PAPER TAPES ONTO ONE DECTAPE FILE, TYPE:

DTA1:BIGFIL+PTR:,,

THE SAME AS ABOVE, BUT WITH SEQUENCE NUMBERS, TYPE:

DTA1:BIGFIL/S+PTR:,,

TO LIST THIS FILE, TYPE:

LPT:+DTA1:BIGFIL

THE SAME AS ABOVE, BUT WITHOUT SEQUENCE NUMBERS, TYPE:

LPT:/N+DTA1:BIGFIL

TO REWIND MAG TAPE:

MTA1/W+

TO COPY BINARY DATA:

DTA:1PIP2.REL/B+PTR:

TO CREAT A FRESH DIRECTORY:

DTA1:/Z+

PIP2 EXAMPLES:

TO CLEAN UP A DECTAPE AND RECOVER UNUSED BLOCKS, MOUNT SOURCE TAPE ON UNIT (SAY) 1 AND A SCRATCH TAPE ON UNIT (SAY) 2. THEN TYPE:

DTA2:/Z+
DTA2:/X+DTA1:

RETURN YOUR OLD TAPE TO THE SCRATCH TAPE SUPPLY AND TAKE POSITION OF THE TAPE ON UNIT 2.

TO RENAME FILES, TYPE:

DTA2NEWNAM.EXT/R+OLDNAM
DTA2:AFTER/R+BEFORE

TO DELETE FILES, TYPE:

DTA2:/D+FILE1,FILE2,FILE3

TO LIST A DIRECTORY:

TTY:/L+DTA2

C H A P T E R I X
B A T C H C O N T R O L P R O G R A M
(B C P)

BCP IS A 1024 WORD (DECIMAL) PROGRAM TO CONTROL A STACK OR BATCH OF PROGRAMS WHICH HAVE BEEN SUBMITTED FOR PROCESSING.

THE USER SUBMITS A DECK OF CARDS, A SEQUENCE OF PAPER TAPES, A MAGNETIC TAPE, OR DECTAPE FILES WHICH CONTAIN THE SOURCE PROGRAM AND DATA FOR THE JOBS, WITH INTERSPURSED CONTROL CARDS. THE LANGUAGE OF THE CONTROL CARDS IS IDENTICAL TO THAT OF THE CONSOLE USER COMMANDS.

BCP RUNS AS A SEPARATE JOB, INDEPENDENT OF THE STACK, AND INTERPRETS THE CONTROL CARDS ON BEHALF OF THE STACK. BCP ALSO TRANSMITS INDIVIDUAL PROGRAM ERROR CONDITIONS BACK TO THE OPERATOR WHO ATTENDS.

APPENDIX OF MULTIPROGRAMMING EXECUTIVE COMPONENTS

THE EXECUTIVE COMPONENTS RESIDE IN SEPARATE FILES WHICH MAY BE ASSEMBLED INDEPENDENTLY USING THE SYSTEM TAPE SYSPAR. SYSPAR CONTAINS THE COMMON SYSTEM DEFINITIONS, MACROS, BIT, AND AC ASSIGNMENTS FOR ALL ROUTINES. EACH ROUTINE WHICH IS REQUIRED TO LINK WITH OTHER SYSTEM ROUTINES CONTAINS THE APPROPRIATE INTERNAL AND EXTERNAL LIST OF MNEMONICS.

DIRECT JOB CONTROL ROUTINES

THESE ROUTINES ARE DIRECTLY ASSOCIATED WITH THE USER OR A JOB. THEY HANDLE THE CONSOLE COMMUNICATIONS, SCHEDULING, ETC.

UUO HANDLER

PROCESSES THE UUOS (OR PROGRAMMED OPERATORS) THAT ORIGINATE WITH THE USER. IF THE COMMANDS ARE USER DEFINED, IT RETURNS CONTROL TO THE USER AREA, OTHERWISE IT CALLS THE APPROPRIATE SYSTEM COMMAND SECTION TO HANDLE THE FUNCTION.

CALL HANDLER

PROCESSES THE SPECIAL UUO, CALL.

COMMAND DECODER

PROCESSES THE COMMANDS WHICH ORIGINATE FROM THE USER'S CONSOLE. ALL SYSTEM CALLS FROM THE CONSOLE E.G. GET, SAVE, ASSIGN, ETC. ARE DECODED AND PROCESSED BY THE COMMAND DECODER. THE COMMAND DECODER MAY CALL OTHER SYSTEM ROUTINES, E.G. CORE, TO DO SPECIFIC FUNCTIONS.

CORE

ALLOCATES CORE STORAGE FOR ALL THE USERS. IT MAY BE CALLED BY A JOB OR BY A USER CONSOLE COMMAND AND IS A REQUEST FOR A SPECIFIC AMOUNT OF CORE STORAGE. ABSOLUTE REQUESTS ARE MADE AND CORE MAY BE EXTENDED OR WITHDRAWN.

SCHEDULER

SCHEDULES THE RUNNING OF JOBS. IT IS CALLED AT REGULAR INTERVALS TO DECIDE WHICH PROGRAM IS TO BE RUN NEXT. A RUNNING PROGRAM IS TEMPORARILY TERMINATED EACH TIME ITS ALLOTTED COMPUTATION TIME EXPIRES, OR WHEN A REQUEST FOR INPUT OR OUTPUT REQUIRES WAITING. A USER MAY HALT A JOB FROM A CONSOLE AND TEMPORARILY SUSPEND OPERATION. THE ALGORITHM FOR CONTROL IS ON A ROUND-ROBIN BASIS, WITH JOBS WHICH HAVE JUST COMPLETED INPUT OR OUTPUT RECEIVING HIGHER PRIORITY.

RUN

CONTROLS THE ACTUAL STARTING AND STOPPING OF JOBS AND ALL THE SYSTEM DATA ASSOCIATED WITH A JOB IS PROCESSED BY RUN.

ERRCON

COMMUNICATES JOB ERRORS TO A USER THROUGH HIS CONSOLE. ILLEGAL MEMORY REFERENCES, DATA MODE TRANSMISSION ERRORS, ETC. ARE TRANSMITTED TO THE USER THROUGH THIS ROUTINE.

EXECUTIVE DATA

THE SYSTEM ROUTINES ARE EITHER PURE PROCEDURES OR GO TO COMPLETION FOR A GIVEN CALL. THE DATA FOR THE SYSTEM IS KEPT IN SPECIAL AREAS.

CLOCK QUEUE DATA

CONTAINS ALL THE REQUESTS FOR CLOCK PROCESSING OR TIME

SERVICE. THE AMOUNT OF TIME REQUESTED TO TIME AND THE LOCATION TO TRANSFER ON COMPLETION OF TIMED INTERVAL.

CALL DATA

CONTAINS ALL THE TABLES FOR THE SYSTEM FUNCTIONS AVAILABLE THROUGH THE UO CALL. ENTRIES FOR THE CALL NAME, AND THE LOCATION OF THE ROUTINE ARE IN CALTAB.

JOBSTS TABLE

CONTAINS THE SYSTEM DATA ASSOCIATED WITH A PARTICULAR JOB WHILE THE JOB IS RUNNING. JOBSTS ALSO CONTAINS ENTRIES FOR EACH JOB IN THE SYSTEM. THESE ARE THE ADDRESS OF THE JOB AND ITS STATUS (IO WAIT, DORMANT, AND ACTIVITY) THAT THE SCHEDULER REQUIRES.

TTYTAB

CONTAINS THE STATUS OF ALL THE TELETYPE CONSOLES. THIS IS THE INFORMATION FOR THE ASSIGNMENT OF TELETYPE BUFFERS TO SPECIFIC JOBS, THE PROGRAMS TO WHICH A TELETYPE IS COMMUNICATING, WHETHER COMMAND DECODER OR SCHEDULER SERVICE IS REQUIRED, ETC.

COMMON JOB FUNCTIONS AND INTERNAL SYSTEM FUNCTIONS

CLOCK

HANDLES REQUESTS FROM ROUTINES FOR TIMING SERVICE. FOR EXAMPLE, DECTAPE ROUTINES REQUEST CLOCK TIMING INFORMATION IF A DECTAPE IS MADE TO FREE RUN FOR A PREDETERMINED PERIOD OF TIME. THE SCHEDULER USES THE CLOCK TO TIME JOB RUNNING INTERVALS.

IOCSS

CONTAINS ALL COMMON SYSTEM SUBROUTINES FOR SYSTEM IO CONTROL. THESE INCLUDE ROUTINES WHICH MOVE BUFFER POINTERS, ETC.

IOCONT

CONTAINS THE SPECIAL PROCESSING FOR THE IO DEVICE UOOS.

SYSCOM

CONTAINS SYSTEM UO PROCESSING ROUTINES.

SYSCSS

CONTAINS COMMON SYSTEM SUBROUTINES.

IOINI

CONTAINS A LIST OF SPECIAL SETUP COMMANDS REQUIRED FOR IO DEVICES INITIALIZATION.

SYSINI

CONTAINS BASIC SYSTEM INITIALIZATION ROUTINES. THESE RESET JOBS, CALL ROUTINES WHICH HAVE DATA ASSOCIATED WITH THEM TO BE RESET, INCLUDING CLOCK DATA, CORE DATA, TELETYPE ASSIGNMENTS, JOB ASSIGNMENTS, ETC.

CREATION AND DEBUGGING

THESE PROGRAMS ARE USED TO CREATE OTHER SYSTEMS WHICH RUN IN EXECUTIVE MODE. THEY ALSO ALLOW THE DEBUGGING PROGRAM DDT TO BE LOADED WITH A NEW SYSTEM, AND ITS SYMBOL TABLE PRESERVED.

SYSMAKER

IS A PROGRAM IN THE SYSTEM WHICH IS RUN ANY TIME A PROGRAM HAS BEEN CREATED IN USER MODE AND IS TO BE RUN IN EXECUTIVE MODE IN THE POSITIONS OCCUPIED BY THE FORMER SYSTEM. WHEN IT IS CALLED, SYMBOLS WHICH RESIDE IN THE USER AREA PROGRAM ARE FIRST MOVED TO THE UPPER AREA OF MEMORY, AND THE PROGRAM IS MOVED DOWN OVER THE OLD SYSTEM, INCLUDING SYSMAKER. IN THIS FASHION A NEW SYSTEM CAN BE LOADED AS A NORMAL TIME SHARED JOB, AND THEN BROUGHT DOWN INTO THE USER EXECUTIVE AREA.

SYSDDT

IS A DDT WHICH MAY BE USED TO DEBUG THE SYSTEM. IT ASSUMES THAT A SYMBOL TABLE WILL BE IN THE HIGHEST SYSTEM MEMORY LOCATIONS AND IO IS DONE DIRECTLY BY DDT.

APPENDIX OF MONITOR CORE ALLOCATION

EXECUTIVE COMPONENTS

EXECUTIVE COMPONENTS

-----	140
SYSINI	42
SYSCON	231
SYSCSS	25
IOCONT	476
IOCSS	337
RUNCSS	244
ERRCON	405
CLOCK	146
CLKCSS	102
IOINIT	1573
COMINI	135
COMCON	200
COMCSS	161
SAVGET	165
CORE	112
SYSMAX	36

IO CONTROL MODULES

MTPSER	546
LPT	162
PTR	156
SCN	757
PTP	256
DTSER	1251
CDR	261

DDT (MONITOR DEBUGGING)

DDT	3126

TOTAL	15227

TABLE OF PHYSICAL DEVICE CHARACTERISTICS

MNEMONIC	NAME	OPTION NUMBER	APPROX. SERVICE ROUTINE SIZE (OCTAL)
TTY, TTY0, TTY1,...TTYN	CONSOLE TELEPRINTER AND/OR HALF/FULL DUPLEX TELEPRINTERS CONNECTED TO 630 DATA COMMUNICATIONS SYSTEM	626, 630 HALF/ FULL DUPLEX 33/35 TELEPRINTERS	600+(31) X NO. OF ACTIVE TELETYPES
PTR	PAPER TAPE READER	760	220
PTP	PAPER TAPE PUNCH	761	300
LPT	LINE PRINTER	646 A,B,C	200
CDR	CARD READER	461 (200 OR 800 CPM READER)	300
DTA,DTA1, ...DTA8	DECTAPE	551. 1 TO 4-555. SHARES 1-136 WITH 516	1300+(200+2)X ACTIVE DECTAPES
MTA,MTA1, ...MTA8.	MAGNETIC TAPE	516 WITH 520, 521, OR 522 FOR 1 TO 8 TYPE 50,570 OR 729VI TRANSPORTS SHARES 7-136 WITH 551.	700+20 X ACTIVE MAGTAPES
DIS	DISPLAY	346	400

TABLE OF DEVICE PROGRAM CHARACTERISTICS

MNEMONIC	BUFFER SIZE (OCTAL)	COMMANDS*	DATA MODES	DATA CONTROL MODES***	STATUS BITS**	ENTER, LOOKUP COMMANDS
TTY,TTY0 TTY1, ...TTYN	17	INPUT, OUTPUT	AL(INPUT) A=AL(OUTPUT)			
PTR	41	INPUT	A,AL,B,I(8 BITS)		IODTER	
PTP	41	OUTPUT	A=AL,B,I(8 BITS)			
LPT	32	OUTPUT	A=AL			
CDR	34	INPUT	A=AL,B,I(12 BITS)		IODTER	
DTA, DTA1, ...DTA8	200	INPUT, OUTPUT, USETO, USETI, UGETF	A=AL=B,D.	IOROEL	IOBKTL	X
MTA MTA1, ...MTA8.	200	INPUT, OUTPUT, MTAPE, GETN	A=AL=B,D	IODENS, IOPAR, IONRCK	IOBKTL, IOBOT, IOTEND.	X
DIS		OUTPUT (SPECIAL DUMP)	D.			

* IN ADDITION TO : ASSIGN, INIT, LOOKUP, ENTER, INBUF, OUTBUF, STATUS, STATZ, STATO, CLOSE, RELEAS, DEASSIGN

** IN ADDITION TO: IOIMPM, IODEND, IOACT, IODERR, MODE BITS 26-35.

***IN ADDITION TO : IOWC, IOCON.

SYSTEMS LIBRARY FILE CONTENTS

3 MARCH 1965

PROGRAM	MNEMONIC	MODE	SIZE(OCTAL)
---------	----------	------	-------------

DIRECTORY			
SYSTEM DIRECTORY			200

EXECUTIVE			
MULTI-PROGRAMMING SYSTEMS		DECDUMP	12000

TRANSLATORS

MACRO ASSEMBLER	MACRO	DUMP	22000
FORTRAN II (MIN.)	FOR2S	DUMP	22000
FORTRAN II (MAX.)	FOR2L	DUMP	46000
FORTRAN II (MAX.)FOLA	FORFOL	DUMP	56000
FORTRAN ON LINE ASSEMBLER	FOLA	DUMP	10000

OTHER CUSPS

PERIPHERAL INTERCHANGE	PIP1	DUMP	2000
PERIPHERAL INTERCHANGE	PIP2	DUMP	4000
EDITOR	EDIT	DUMP	2000
LOADER	LOAD	DUMP	2200

RELOCATABLE LINKING LOADER FILES

DDT FOR USER	USRDDT	RELBIN	5000
FORTRAN OP SYSTEM	FORSE	RELBIN	4000

OPERATING THE MULTIPROGRAMMING MONITOR APPENDIX

LOADING THE SYSTEM

1. TURN ON COMPUTER (POWER SWITCH ON CONSOLE)
2. TURN ON ALL IO DEVICES INCLUDING SCANNER.
3. LOAD DECDUMP INTO TOP OF MEMORY
 - A. MOUNT A DECTAPE WITH DECDMP AND DECDUMP LOADER ON UNIT #1 (SEE DECTAPE LOADERS DESCRIPTION)
 - B. SET AD4RESS SWITCHES TO 0
 - C. PUSH STOP SWITCH DOWN AND RETURN
 - D. LIFT START SWITCH UP
 - E. DECDUMP WILL RESPOND WITH CR-LF AFTER LOADING.
4. LOAD TIME SHARING MONITOR FROM DECTAPE.
 - A. TYPE NL WHERE N IS THE NUMBER OF THE TAPE DRIVE ON WHICH THE MONITOR HAS BEEN MOUNTED.
 - B. DECDUMP WILL RESPOND WITH CR-LF WHEN LOADING HAS COMPLETED.
 - C. TYPE G TO BEGIN EXECUTION.
 - D. AFTER A SECOND OR TWO THE MONITOR WILL TURN ON ALL OF THE PI CHANNELS AND WILL BE READY TO ACCEPT TYPE IN. WHENEVER THE MONITOR HAS NOTHING TO DO, IT WILL RUN THE NULL JOB. THE NULL JOB COUNTS IN AC0 AND RUNS IN AC1. THEREFORE, IT CAN BE DETECTED BY A 1 IN THE PROGRAM COUNTER LIGHTS. THE COUNT IN AC0 BEGINS OVER EACH TIME THE NULL JOB BEGINS TO RUN SO THAT SETTING THE ADDRESS SWITCHES TO 0 WILL GIVE A ROUGH IDEA OF THE SYSTEM IDLE TIME.
5. THE OPERATOR INITIALIZES A JOB AND KEEPS IT WHILE THE SYSTEM IS RUNNING. HE ALSO NAMES HIS CONSOLE OPER SO THAT USER'S MAY COMMUNICATE WITH HIM. TO DO THIS HE TYPES THE FOLLOWING COMMANDS:
PJOB (WHICH WILL PRINT HIS JOB NUMBER WHICH MUST BE 1)
ASSIGN TTY:OPER (WHICH WILL GIVE THE OPERATOR'S TELETYPE TTY THE NAME OPER. HE SHOULD THEN ASSIGN TO HIMSELF ANY IO DEVICES WHICH SHOULDN'T BE USED BY USERS BY TYPING '
ASSIGN DEV
WHERE DEV IS THE PHYSICAL DEVICE NAME (DTA3 FOR EXAMPLE).
A USER MAY COMMUNICATE WITH THE OPERATOR BY TELETYPE BY TYPING THE FOLLOWING:

ATTACH 1
CORE 1
GET DTA0:PIP
START
OPER+TTY:

THE MESSAGE TERMINATED BY THE ASCII END OF MESSAGE CHARACTER
<CONTROL>Z.
NOTE: IT IS A GOOD IDEA TO BEGIN AND END WITH CARRIAGE RETURN.

<CONTROL>C
CORE 0
ATTACH N

MONITOR ERROR HALTS

THE MONITOR CONTAINS NO ERROR HALTS EXCEPT FOR "IMPOSSIBLE" ERROR CONDITIONS. IF A SO CALLED "IMPOSSIBLE" ERROR HALT OCCURS, WRITE DOWN BOTH THE CONTENTS OF THE PROGRAM COUNTER AND THE MEMORY ADDRESS. THEN TRY PRESSING CONTINUE. IF THE MACHINE HALTS AT THE SAME PLACE IMMEDIATELY, THERE IS PROBABLY A HARDWARE MALFUNCTION. THE OPERATOR SHOULD EITHER CALL IN THE MAINTENANCE MAN OR ATTEMPT THE ERROR RECOVERY PROCEDURE LISTED BELOW.

ERROR RECOVERY PROCEDURE (IN ORDER OF INCREASING SEVERITY)

1. PRESS STOP SWITCH AND HOLD DOWN.
PRESS CONTINUE SWITCH (TO MAKE SURE IN EXEC MODE)(DO NOT PRESS IO RESET)
RELEASE STOP SWITCH
SET ADDRESS SWITCHES TO 144
PRESS START
THIS WILL BEGIN BY RUNNING THE NULL JOB.
THIS WILL AFFECT AT MOST ONE JOB WHICH THE USER WILL HAVE TO RESTART.
2. PRESS STOP SWITCH
PRESS IO RESET
SET ADDRESS SWITCHES TO 140
PRESS START
THIS WILL INITIALIZE ALL IO DEVICES INCLUDING THE TELETYPES. ALL CONSOLES WILL BE IN COMMAND MODE AND EACH USER MUST ATTACH HIS CONSOLE BACK TO HIS JOB. ALL JOBS MUST BE STARTED OVER.
3. PRESS STOP SWITCH
PRESS IO RESET
SET ADDRESS SWITCHES TO 143
PRESS START
THIS WILL REINITIALIZE THE MONITOR TO THE STATE IT WAS IN JUST AFTER BEING LOADED. ALL JOBS MUST BE RELOADED.
4. PRESS STOP SWITCH
RELOAD DECDUMP
RELOAD MONITOR FROM DECTAPE.

OCCASIONALLY A SYSTEM WILL GET INTO A LOOP. THIS CAN BE DETECTED BY SEEING THAT THE COMPUTER IS NEVER LEAVING EXEC MODE AND IS NOT RUNNING THE NULL JOB (PC=1). USERS WILL REPORT THAT SOMETHING APPEARS TO BE WRONG SO THAT THE OPERATOR DOES NOT NEED TO WATCH FOR THIS CONDITION. WHEN SUCH A LOOP OCCURS, NOTE WHETHER AN INTERRUPT IS IN PROGRESS. IF NOT, PRESS STOP AND LOOK AGAIN. PRESS CONTINUE AND STOP UNTIL NO INTERRUPTS ARE IN PROGRESS. THEN WRITE DOWN THE CONTENTS OF THE PROGRAM COUNTER AND MEMORY ADDRESS. THEN FOLLOW ERROR RECOVERY PROCEDURE.

,OPDEF TAPE FOR EXEC SYSTEM IO
,DECEMBER 19,1964

OPDEF ENTER [77B8]
OPDEF LOOKUP [76B8]
OPDEF USETO [75B8]
OPDEF USETI [74B8]
OPDEF UGETF [73B8]
OPDEF MTAPE [72B8]
OPDEF RELEAS [71B8]
OPDEF CLOSE [70B8]
OPDEF OUTPUT [67B8]
OPDEF INPUT [66B8]
OPDEF OUTBUF [65B8]
OPDEF INBUF [64B8]
OPDEF STATZ [63B8]
OPDEF STATUS [62B8]
OPDEF STATO [61B8]
OPDEF INIT [41B8]
OPDEF CALL [40B8]

TITLE JOB DAT - JOB DATA AREA ASSIGNMENTS
SUBTTL MARCH 12, 1965

```
DEFINE BLK (A,B)
<JOBPRO=B
A=0
INTERNAL A
DEFINE BLK (C,D)
<C=JOBPRO
JOBPRO=JOBPRO+D
INTERNAL C>>
```

```
BLK JOBAC,20 ;SYSTEM USE AC STORAGE
BLK JOB DAC,20 ;USER'S AC STORAGE WHILE JOB IS INACTIVE
BLK JOBUUO,2 ;UUO SAVE AND TRAP LOCATIONS
BLK JOBPC,1 ;BITS 0-5=APR FLAGS, C(RH)=PC
BLK JOBPDP,1 ;PUSH DOWN POINTER FOR SYSTEM USE
BLK JOBREL,0 ;C(LH)=UNUSED, C(RH)=RELOCATION ADDRESS
BLK JOBUXT,1 ;EXIT FROM UUO AT BREAK
BLK JOBLEV,1 ;UNUSED
BLK JOBSAV,1 ;TEMPROARY STORAGE FOR UUO HANDLER
BLK JOBPOV,1 ;PUSH DOWN OVERFLOW TRAP
BLK JOBTEM,2 ;TEMPORARY SYSTEM STORAGE
BLK JOBINF,1 ;BITS 0-11=JOB NUMBER
BLK JOBJDA,20 ;JOB IO DEVICE CHANNEL ASSIGNMENTS
BLK JOBDDT,1 ;ADDRESS OF USER'S DDT
BLK JOBCDP,1 ;POINTER TO COMMAND STRHNG
BLK JOBTRP,20 ;TRAP LOCATIONS
BLK JOBSYM,1 ;SYMBOL TABLE POINTER
BLK JOBUSY,1 ;POINTER TO UNDEFINED SYMBOL TABLE
BLK JOBSA,1 ;C(RH)=STARTING ADDRESS
;C(LH)=HIGHEST LOCATION USED
BLK JOBFF,1 ;FIRST FREE LOCATION
JOBPDL=JOBPRO-1 ;FIRST LOC. -1 OF PD LIST
MJOBPD=JOBPDL-137 ;- LENGTH OF PD LIST
```

```
INTERNAL JOBPDL,JOBPRO,MJOBPD
```

END

XLIST

```

DEFINE      XP(A,B)                ;SYSTEM PARAMETER
           <A=B
           INTERNAL A
>
,ACCUMULATORS
  XP IOS,0;                IO STATUS WORD
  XP TAC,1;                TEMPORARY
  XP TAC1,2;              TEMPORARY
  XP PDP,3;                C(LH)=NUMBER OF LOCATIONS LEFT IN PD LIST
  ,                        C(RH)=PUSHDOWN POINTER
  XP SUB,10               RETURN (PC) FOR ONE LEVEL SUBROUTINES
  XP ITEM,4;              BYTE OR WORD POINTER, ITEM COUNT
  XP DAT,5;                DATA OR TEMPORARY
  XP DEVDAT,6;            C(LH)=UNUSED,
  ,                        C(RH)=ADDRESS OF DEVICE DATA BLOCK
  XP JBUF,DAT;            ADDRESS OF JOB BUFFER AREA=JBFADR
  XP PROG,7;              C(LH)=UNUSED
  ,                        C(RH)=ADDRESS OF USER'S PROGRAM AREA
  XP JDAT,11;             C(RH)=ADDRESS OF JOB DATA
,ONLY 0 TO 10 NORMALLY REQUIRED
,FOR INTERRUPT SERVICE
  XP BUFPNT,12;           C(JBFADR 18-35)
  XP BUFWRD,13;          BIT 0=IOUSE
  ,                        BITS 1-17=BUFFER SIZE
  ,                        BITS 18-35=NEXT BUFFER ADDRESS
  XP UUU,15;              LAST UUU PROCESSED
  XP JB,14; JOB NUMBER
  XP AC1,16;              UNUSED
  XP AC2,17;              UNUSED

```



```

, DEVICE DATA BLOCK NAMES
  XP DEVNAM,0;      NAME IN SIXBIT ASCII
,                  C(LH)=DEVICE MNEMONIC
,                  C(RH)=DEVICE NUMBER, LEFT JUSTIFIED
  XP DEVCHR,1;     CHARACTERISTIC
,                  BITS 0-8=JOB NUMBER
,                  ZERO VALUE IMPLIES NOT ASSIGNED
,                  BITS 9-11=PI PRIORITY CHANNEL FOR DEVICE
,                  BIT 12=UNUSED
,                  BIT 13=IORET, 0 IF DEVICE HAS A SHORT
,                  DISPATCH TABLE, 1 IF LONG
,                  BITS 14-17=JOB DEVICE CHANNEL NUMBER
,                  BITS 18-23=DEVICE NUMBER, BINARY
,                  BITS 24-35=BUFFER SIZE
  XP DEVIOS,2;     STATUS WORD, SEE BELOW
  XP DEVSER,3;     C(LH)=NEXT DEVICE DATA BLOCK
,                  C(RH)=DEVICE SERVICE DISPATCH TABLE
, DEVICE SERVICE DISPATCH TABLE ASSIGNMENTS
  XP DRL,0;        XXXDSP : RELEASE
  XP DCL,1;        XXXDSP +1: CLOSE
  XP DOU,2;        XXXDSP +2: OUTPUT
  XP DIN,3;        XXXDSP +3: INPUT;SHORT DISPATCH TABLE
  XP DEN,4;        XXXDSP +4: ENTER
  XP DLK,5;        XXXDSP +5: LOOKUP
  XP DDO,6;        XXXDSP +6: DMPO
  XP DDI,7;        XXXDSP +7: DMPI
  XP DSO,10;       XXXDSP+10: SETO
  XP DSI,11;       XXXDSP+11: SETI
  XP DGF,12;       XXXDSP+12: GETF;LONG DISPATCH TABLE
  XP DEVMOD,4      ;BITS 6-11=LEFT HALF OF IMAGE MODE BYTE POINTER
,                  ;BIT 35-J=1 IF MODE J IS LEGAL FOR THIS DEVICE
,                  ;BIT 18 DEVICE ASSIGNED BY CONSOLE COMMAND
,                  ;BIT 19 DEVICE ASSIGNED BY PROGRAM (INIT)
  XP DEVLOG,5;     LOGICAL NAME FOR JOB DEVICE
  XP DEVBUF,6;     C(LH)=OUTPUT BUFFER AREA DDRESS
,                  C(RH)=INPUT BUFFER AREA ADDRESS (JBFADR)
  XP DEVIAD,7;     C(LH)=PROG IN INDEX FIELD
,                  C(RH)=CURRENT INPUT BUFFER ADDRESS
  XP DEVOAD,10;    C(LH)=PROG IN INDEX FIELD
  XP DEVPTR,DEVOAD
,                  C(RH)=CURRENT OUTPUT BUFFER ADDRESS
  XP DEVCTR,11;
, FOR DEVICE CHARACTERISTIC WORD
  XP IORET,20;     FOR RETRIEVABLE DEVICES. SHORT DISPATCH
,                  TABLE=0, LONG=1.
, FOR DEVICE MODE WORD
  XP ASSCON,400000 ;ASSIGNED BY CONSOLE COMMAND ASSIGN
  XP ASSPRG,200000 ;ASSIGNED BY PROGRAM (INIT)
  XP ATTBIT,400000 LH TTY ATTACHED TO DEVICE DATA BLOCK

```

```

, I/O STATUS WORD ASSIGNMENTS
,DATA MODES:  BITS 32-35
    XP A,0;           ASCII
    XP AL,1;         ASCII LINE
    XP AS,2;         ASCII SEQUENCED
    XP ASL,3;        ASCII SEQUENCED LINE
    XP I,10;         IMAGE
    XP B,14;         BINARY
    XP AM,5;         ALTERNATE MODE BREAK (TTY)
    XP AC,6;         CONTROL CHARACTER BREAK (TTY)
    XP DR,16 ;DUMP IN READ CONTROL MODE
    XP D,17;         DUMP
, STATUS BITS
,RIGHT HALF (USER)
    XP IOWC,20;      DON'T COMPUTE WORD COUNT
    XP IOCON,40;     CONTINUOUS (CONT=0)
    XP IORDEL,100;   READ AND DELETE
    XP IONRCK,100;   READ WITH NO. REREAD CHECK
    XP IODEND,20000; DATA END ENCOUNTERED
    XP IODERR,200000; DEVICE ERROR
    XP IODTER,100000; DATA ERROR
    XP IOIMPM,400000; IMPROPER MODE REQUESTED
    XP IOBKTL,40000; BLOCK TOO LARGE
    XP IOACT,10000;  DEVICE ACTIVE
, LEFT HALF (SYSTEM)
    XP IOW,1;        I/O WAIT
    XP IOBEG,2;      VIRGIN DEVICE
    XP IODISC,400000; DISCONNECT REQUEST
    XP IOFST,4;      NEXT ITEM WILL BE THE FIRST ITEM OF A BUFFER
    XP IOSTRT,10;    IO READY TO START
    XP IO,20;        OUT=1, IN=0
    XP IOEND,40;     SERVICE ROUTINE HAS TRANSMITTED LAST DATA

;LEFT HALF USRJDA (JOB DEVICE ASSIGNMENTS) UUU'S FOR THIS CHANNEL SINCE LAST INIT
    XP INITB,400000; INIT
    XP IBUFB,200000; INIT WITH INPUT BUFFER SPECIFIED
    XP OBUFB,100000; INIT WITH OUTPUT BUFFER SPECIFIED
    XP LOOKB,40000;  LOOKUP
    XP ENTRB,20000;  ENTER
    XP INPB,10000;   INPUT
    XP OUTPB,4000;   OUTPUT
    XP CLOSB,2000;   CLOSE
                                ;RELEASE CLEARS THEM ALL

```

<pre> , JOB BUFFER AREA HEADER XP JBFADR,0; , , , , XP JBFPTR,1; XP JBFCTR,2; , JOB BUFFER HEADER XP IOUSE,400000; , , , </pre>	<pre> BIT 0=1 IF THIS BUFFER RING HAS NEVER BEEN REFERENCED FROM THE USER'S PROGRAM BY AN INPUT OR OUTPUT COMMAND. BITS 1-17=UNUSED BITS 18-35=CURRENT BUFFER ADDRESS BYTE POINTER TO NEXT BYTE -1 POSITIVE ITEM COUNT BIT 0=IOUSE 1 IF BUFFER IS FULL (OR BEING EMPTIED) 0 IF BUFFER IS EMPTY (OR BEING FILLED) BITS 1-17=BUFFER SIZE BITS 18-35=NEXT BUFFER ADDRESS </pre>
---	--

```

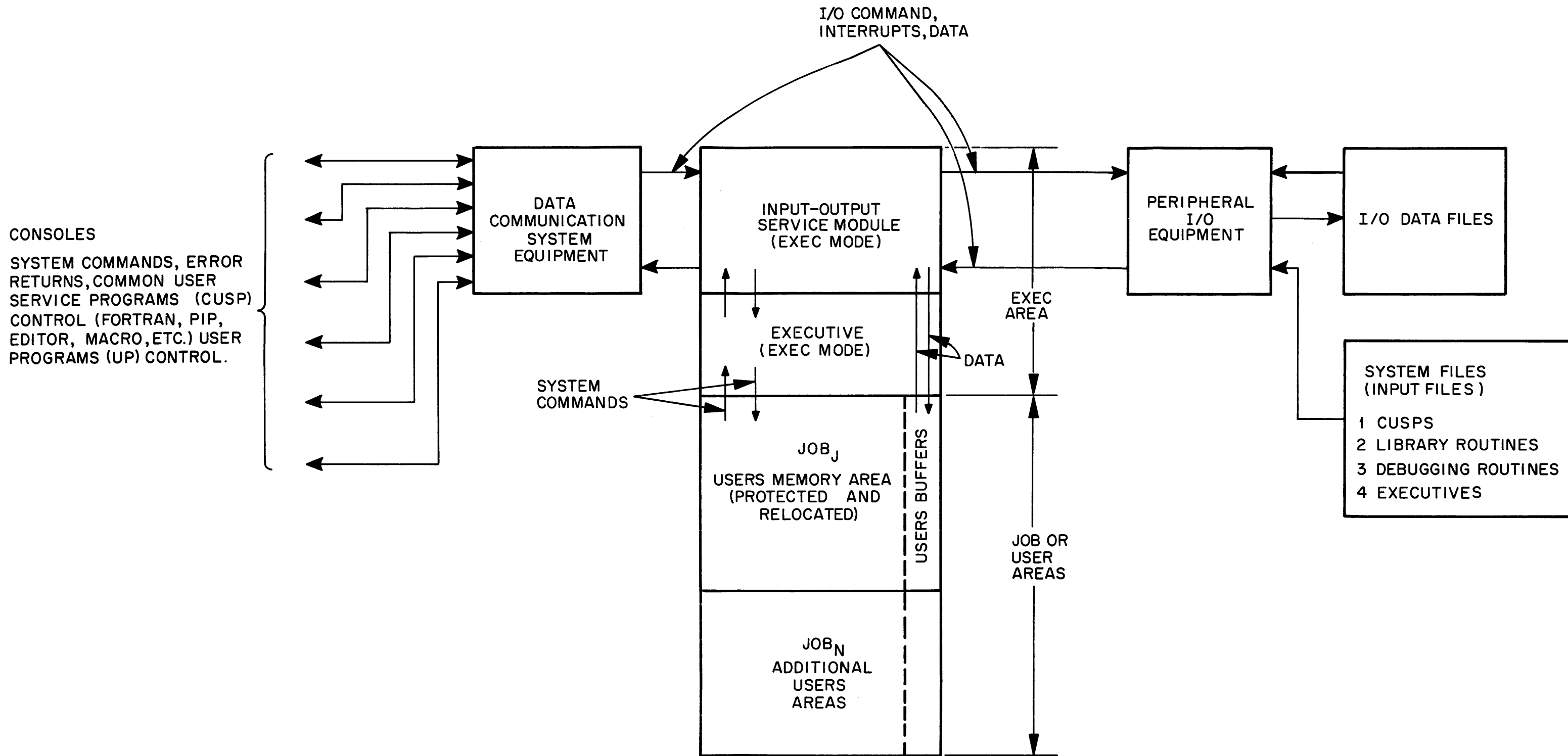
,SYSTEM PARAMETERS
  XP USRMOD,10000;      USER MODE BIT IN ARP FLAGS, PC WORD
  XP IODONE,400000;
,JOB STATUS WORD (JBTSTS)
  XP IOWS,400000;      IO WAIT SATISFIED
  XP RUN,200000;       READY TO RUN OR IS RUNNING
  XP JIOW,100000;      JOB IN IO-WAIT STATE
  XP DTW,40000;        DECTAPE CONTROL WAIT
  XP DCW,20000;        DATA CONTROL WAIT
  XP MTW,10000;        MAG TAPE CONTROL WAIT
  XP JNA,4000;         JOB NUMBER ASSIGNED
  XP JERR,2000;        JOB ERROR(ILLEGAL MEM. ETC.)

XP RUNABLE,RUN+JNA;   STATUS BIT PATTERN FOR A JOB TO STAY RUNABLE

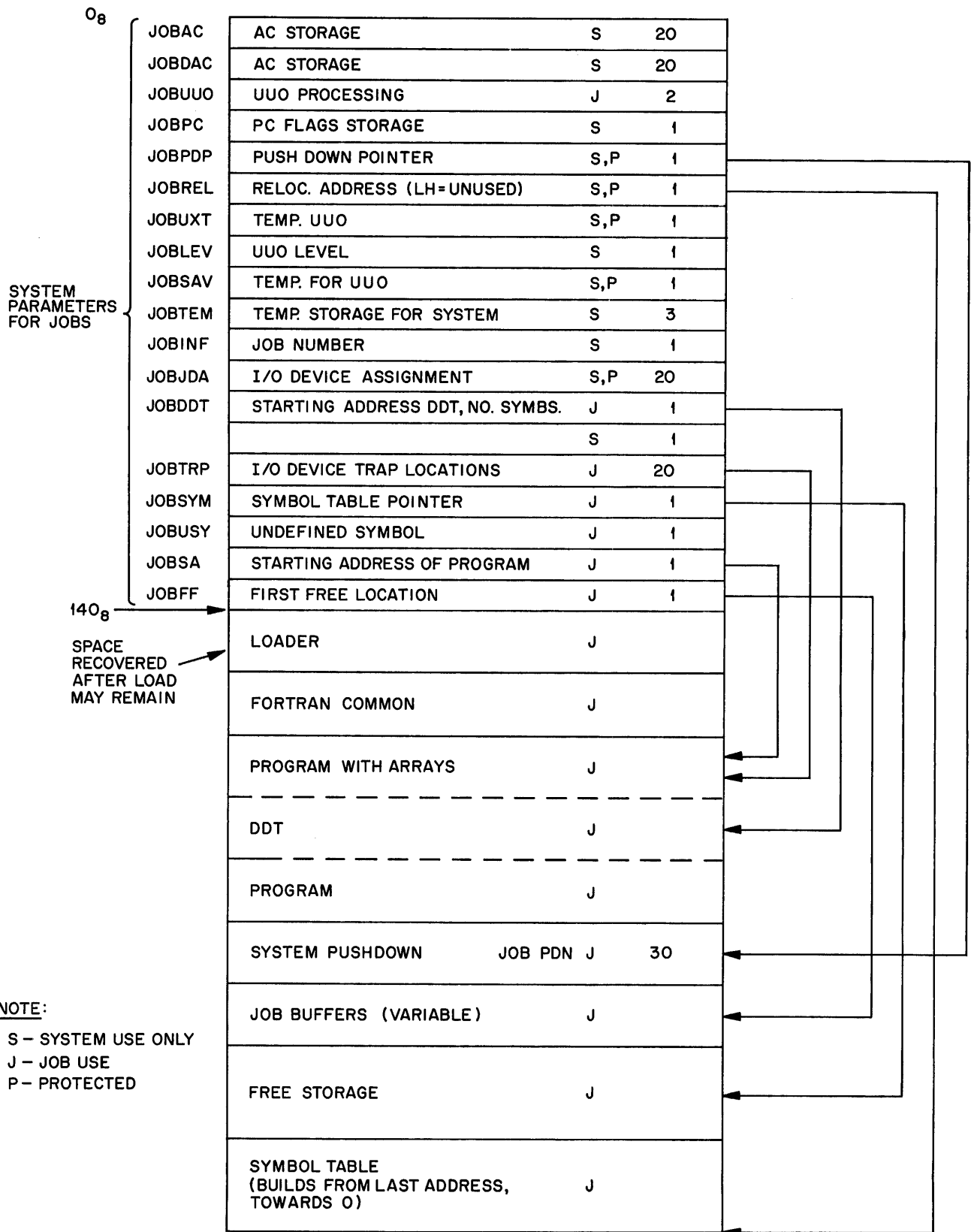
```

```
, SYSTEM MACROS
DEFINE   ADRCHK (A)
<        EXTERNAL ADRCK
DEFINE   ADRCHK (B)
        <PUSHJ PDP,ADRCK
        HRRZ TAC,B>
        PUSHJ PDP,ADRCK
        HRRZ TAC,A>
```

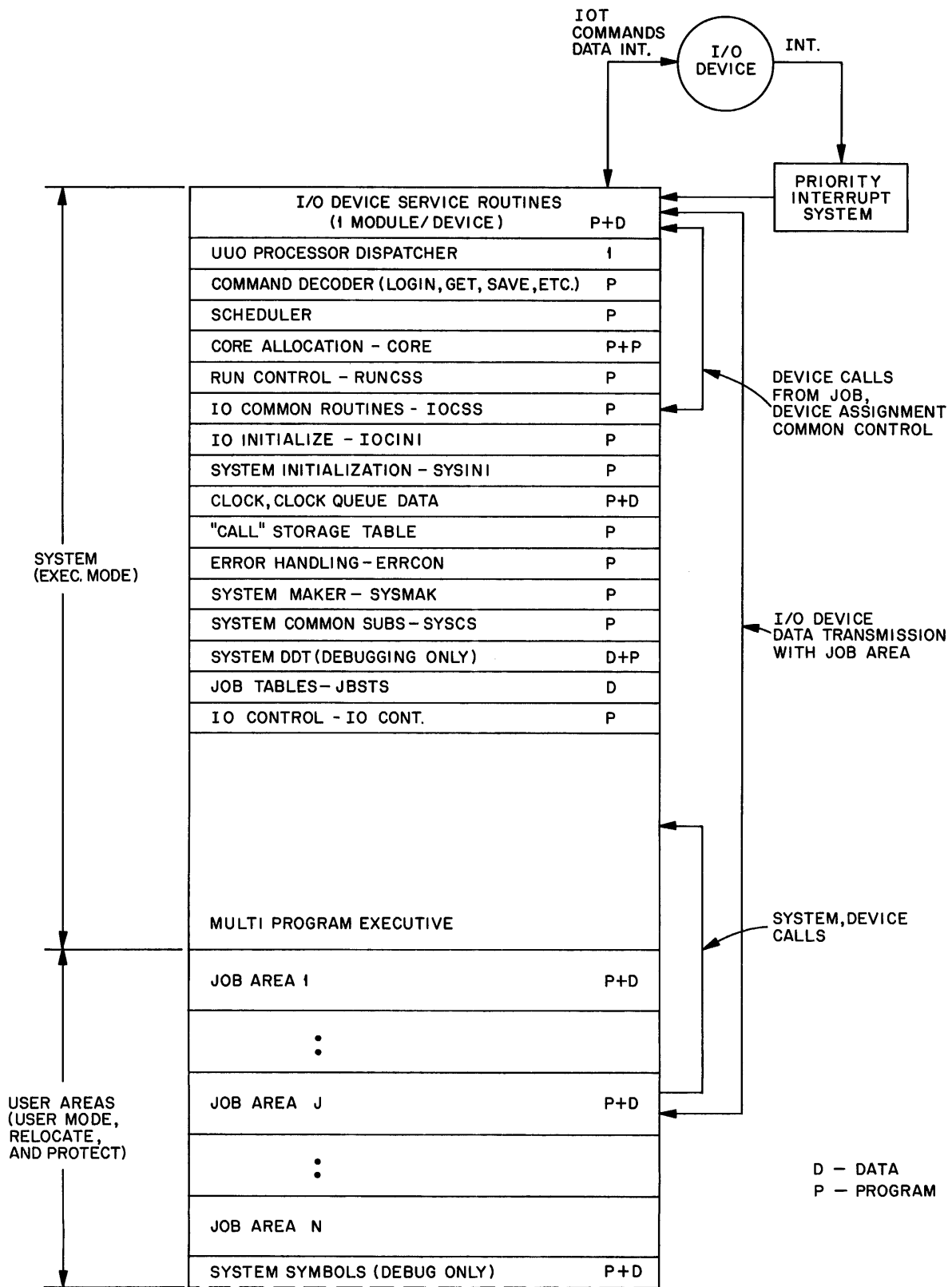
```
LIST
```



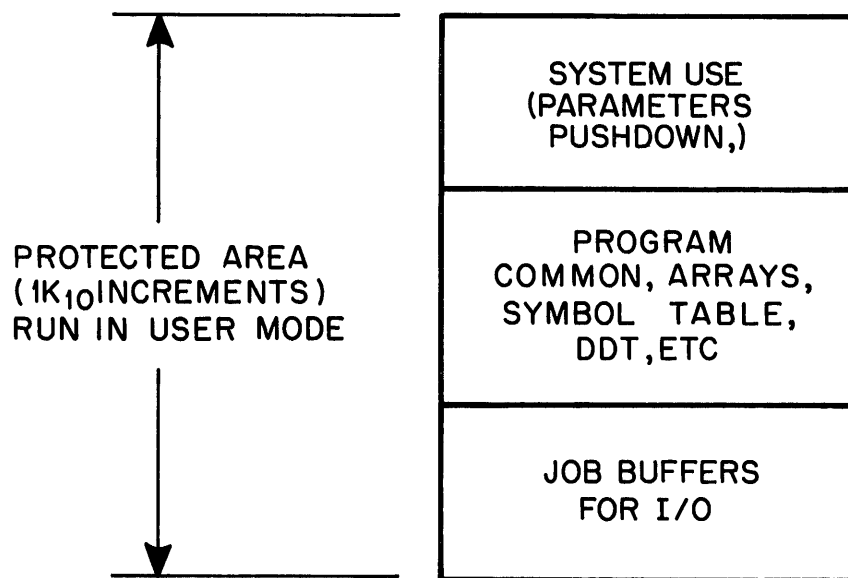
MULTIPROGRAMMING SYSTEM DIAGRAM



JOB AREA STORAGE



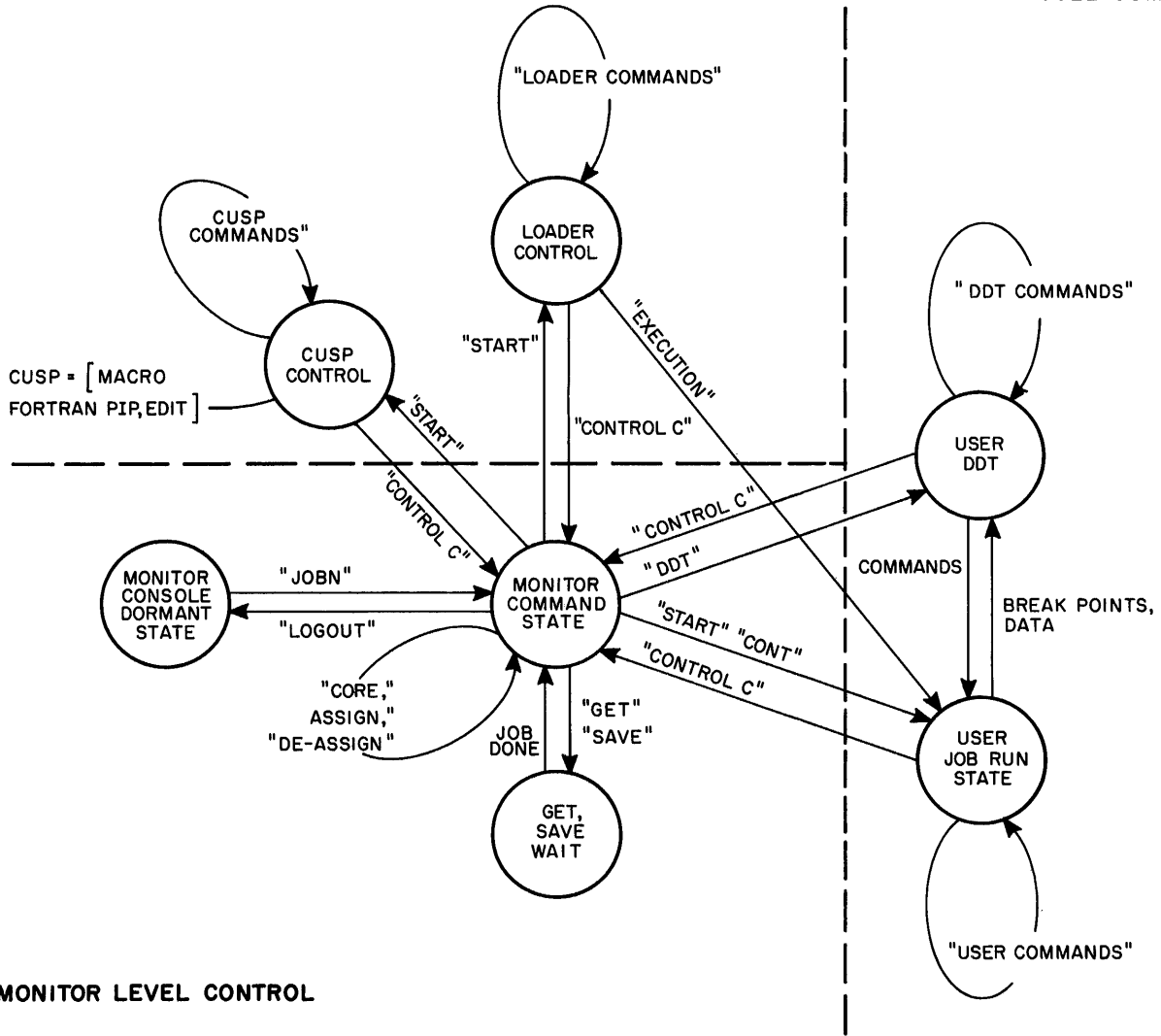
MULTIPROGRAMMING SYSTEM STORAGE



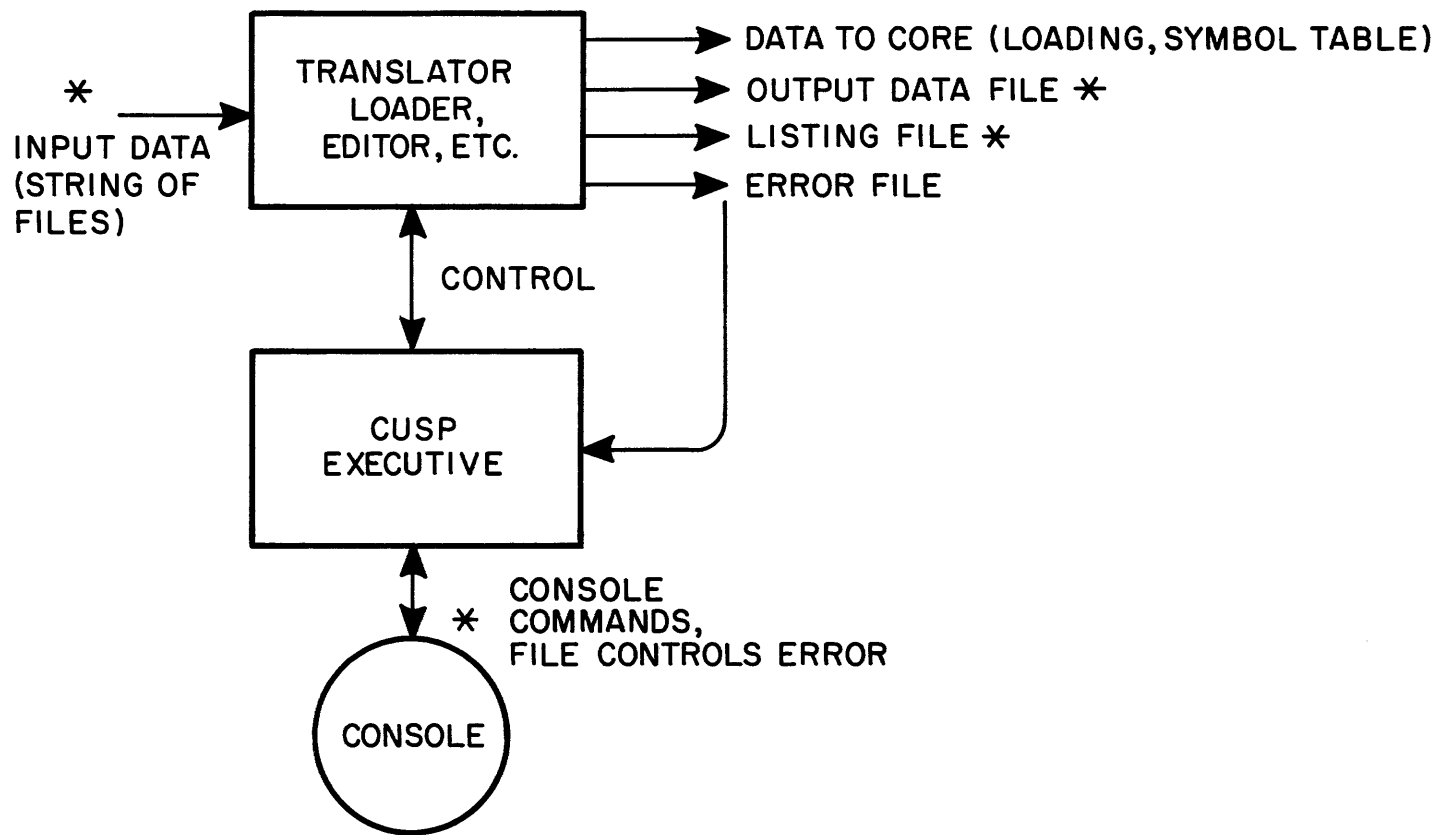
JOB AREA STORAGE (SIMPLIFIED)

**COMMON USER SERVICE PROGRAM (CUSP)
LEVER CONTROL**

**USER CONTROL JOB LEVEL
"XXX" = CONSOLE COMMAND**



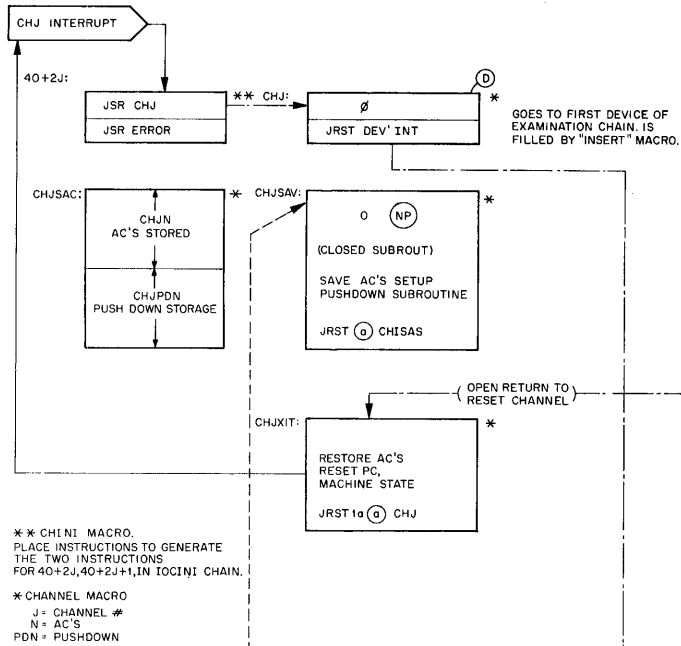
CONSOLE STATE DIAGRAM



* I/O DEVICE CHANNELS

GENERAL STRUCTURE OF COMMON USER SERVICE PROGRAM - CUSP

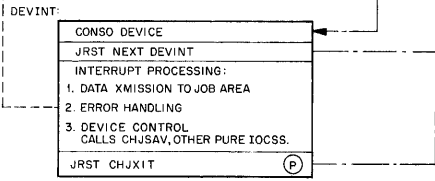
DEVICE INDEPENDENT INTERRUPT PROCESSING



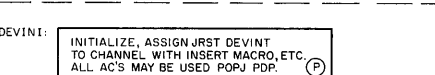
* CHINI MACRO.
PLACE INSTRUCTIONS TO GENERATE
FOR 40+2J, 40+2J+1, IN IOCINI CHAIN.

* CHANNEL MACRO
J = CHANNEL #
N = AC'S
PDN = PUSHDOWN

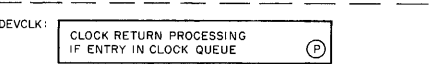
DEVICE INTERRUPT PROCESSING



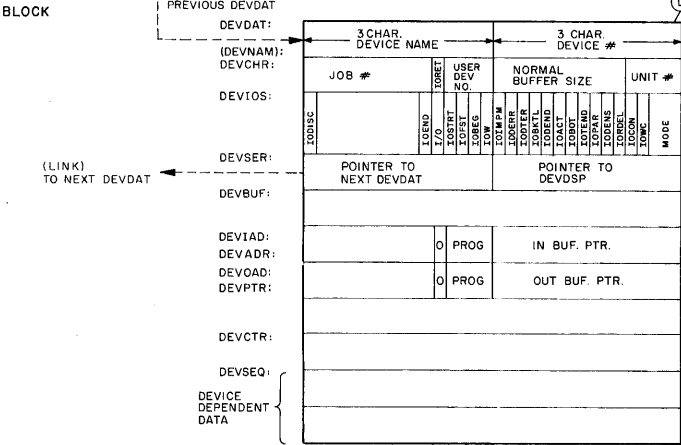
DEVICE INITIALIZATION



DEVICE CLOCK PROCESSING

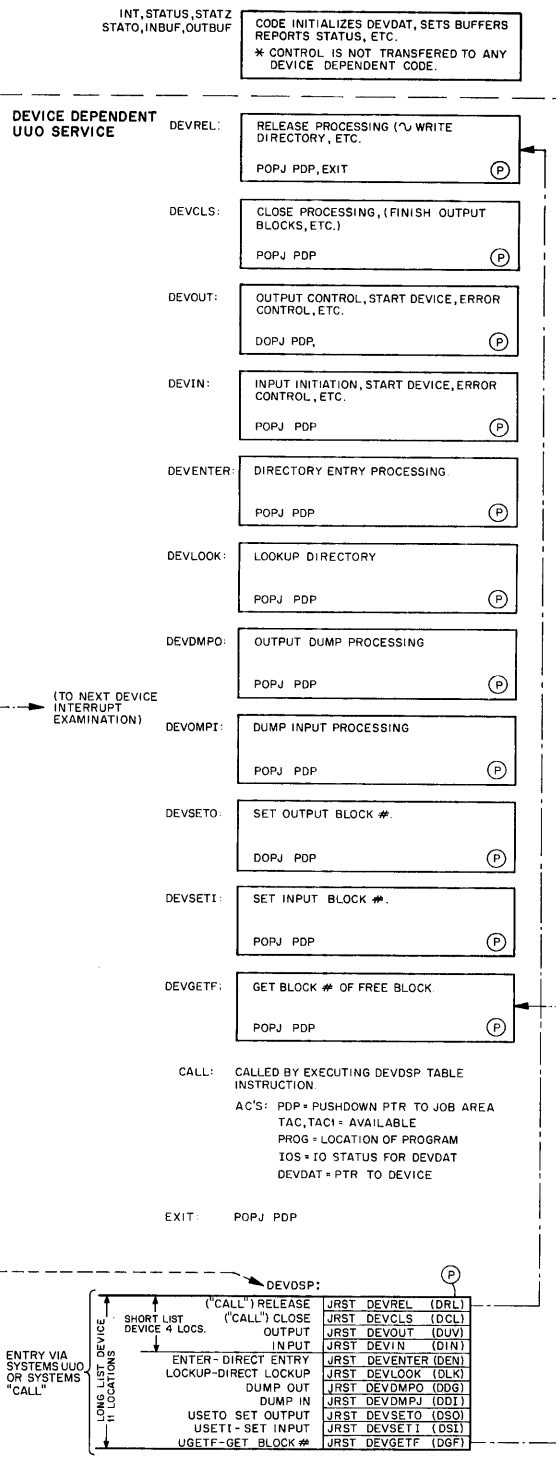


DEVICE DATA BLOCK



* LINKING OCCURS AT SYSINI BY EXECUTING IOGO CODE ON LIST OF DEVDATS TO LINK AT LOCATION LST.

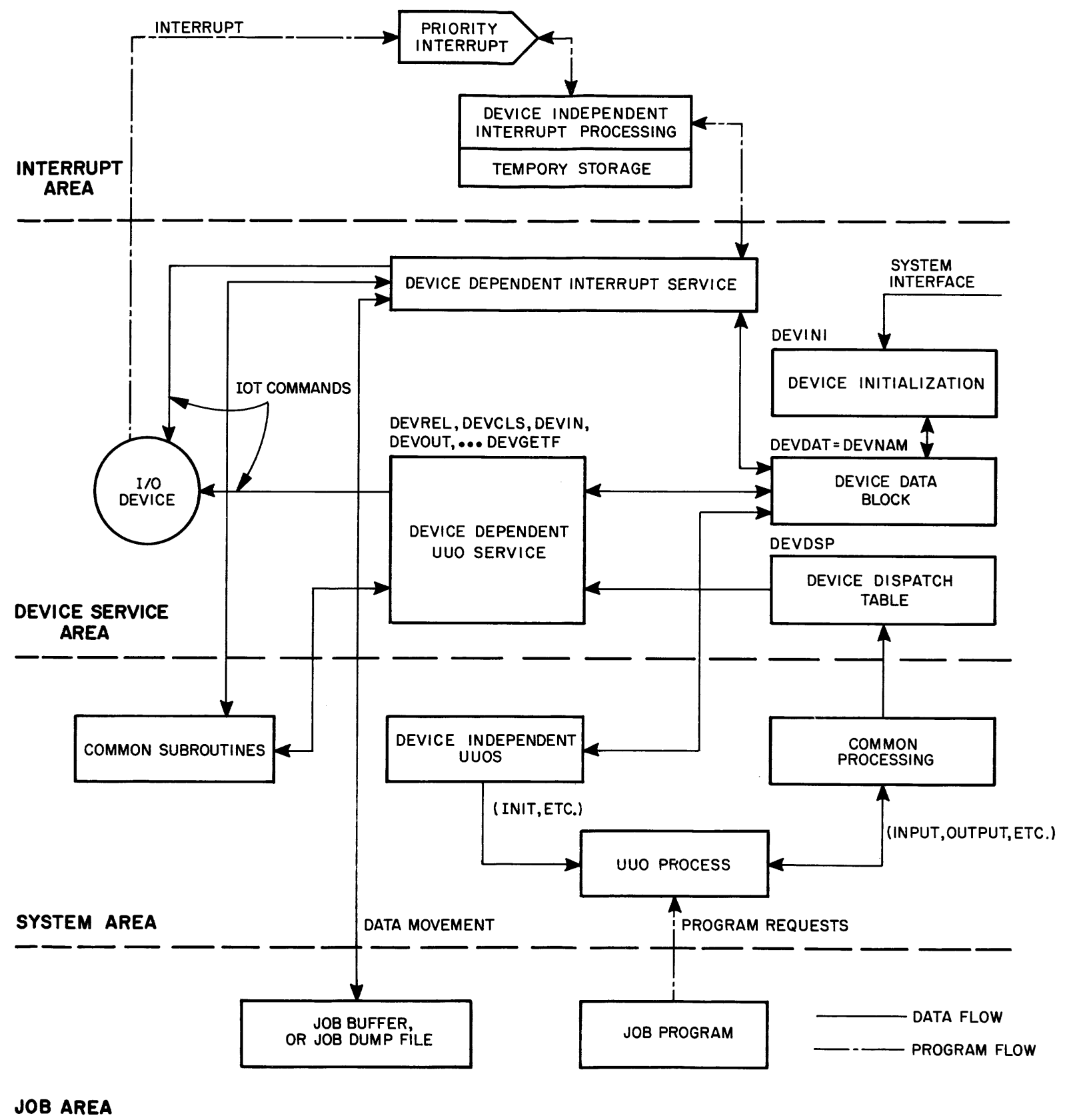
DEVICE INDEPENDENT UOO COMMANDS



DEVICE DISPATCH
ENTRY: EXECUTE INSTRUCTION IN TABLE MAY BE

1. JRST SERVICE ROUTINE
2. POPJ PDP, (NO OPERATION)
3. PUSH PDP, ERROR PROCESS

(D) = DATA
(P) = PURE/RE-ENTRANT
DEV = DEV MNEMONIC



I/O SERVICE ROUTINE BLOCK DIAGRAM SIMPLIFIED

SUMMARY OF COMMON USER SERVICE PROGRAMS					
PROGRAM	SIZE	NUMBER OF DEST. FILES	SWITCHES	COMMANDS	ERROR MESSAGES
EDITOR	1K	EDITOR Operates Upon One File (Sequenced.)	There are No Switches In The EDITOR. Special Delimiters: CR(CARRIAGE RETURN)=Ends a Command (POINT)=Has the Value of the Last line Number Typed. ALT MODE=Special Delineator For S; When Used Alone Advances and Prints the Line.	Sx Control A (CR)=Select Unit X and Clear the Directory Sx, name Control A ALT MODE=Same as Above But Creates File "NAME" Sx, Name (CR)=Open File "NAME" On Unit X Sx, Name ALT MODE=Create a New File Called "NAME" on Unit X E(CR)=CLOSE THE CURRENT FILE In1(CR)=Insert at Line n1 In1,I2(CR)=As Above, But Set Indexing Increment to I2 Dn1(CR)=Delete Line n1 Dn1,n2(CR)=Delete Line n1 thru n2 Pn1(CR)=Print Line n1 Pn1,n2(CR)=Print Lines n1 thru n2 Rx(CR)=Resequence The Current File Rx,y(CR)=Resequence, Stepping By y ALT MODE=Return From Text Mode To Command Mode, or If In Command Mode, Prints the Next Line	**ILC** ILLEGAL COMMAND **NLN** AP or D REFERENCED A NONEXISTENT LINE **ILS** The Index Step(In Insert) Has Caused a Line Already in The File To Be Skipped. **ILR** The Index Step Has Caused An Already Existing Line to be Referenced. **UNA** DecTape Unit is not Available. **DCE** DecTape Directory Full **NFD** No File Has Been Selected **FAU** File Name Already In Use **NCF** Not a Current File
PIP1	1K	1	A=ASCII Line N=Delete Sequence Numbers S=Sequence W=Rewind Mag. Tape Z=Create Fresh Directory B=Binary	General Form: Destination ← Source 1, Source 2,.... DTA1:Big Fil ← PTR:,, (Copy 3 Paper Tape Files onto DecTape) DTA1:Big Fil/S ← PTR:,, (Same as Above But Add Sequence No.) LPT: ← DTA1:Big Fil (List The File) MTA1/W ← (Rewind Mag. Tape) DTA2:/Z ← (Create a Fresh Directory)	"DEVICE name NOT AVAILABLE" "DIRECTORY FULL" "NO FILE NAMED file name" "OUTPUT DEVICE name DEVICE ERROR" "OUTPUT DEVICE name DATA ERROR" "OUTPUT DEVICE name IMPROPER MODE" "OUTPUT DEVICE name BLOCK TOO LARGE" "DECTAPE INPUT REQUIRED" "USE 2K PIP" "INPUT FILE file name DEVICE ERROR" ETC.
PIP2	2K(MINIMUM) (Will use All Available Storage)	1	A=ASCII Line B=Binary C=Card Input Surpress Trailing Spaces Imbedded Multiple Spaces To Tabs D=Delete A File L=List Directory Of Source N=Delete Sequence Numbers R=Rename Source File With Name of Dest. S=Add New Sequence Numbers W=Rewind Mag. Tape X=Copy and Clean Up DecTape Z=Create A Fresh Directory T=Same As C, But Without Tabs	General Form: Same as PIP1, but in Addition: DTA2:/X ← DTA1: (Copy and Clean Up) DTA2:After/R ← Before (Rename A File) DTA2:/D ← File 1, File 2 (Delete Two Files) LPT:/L ← DTA2: (List The Directory)	Same As PIP1

SUMMARY OF COMMON USER SERVICE PROGRAMS					
PROGRAM	SIZE	NUMBER OF DEST. FILES	SWITCHES	COMMANDS	ERROR MESSAGES
FORTRAN	11K or 23K	2	C=Card Image Input (Assumed) T=Teletype or Paper Tape Input R=Read in Rules (Paper Tape Image Assumed)	(1 Pass Compiler) Binary, Listing ← Source Ex: DTA3:BINFIL,LPT: ← DTA2:FORTIN ← PTR:(T) (No Output,Errors Only) (2 Pass Compiler) Fola Input,Listing ← Source Ex: DTA2:FOLAIN,DTA1: LIST ← DTA5:FORTIN (Card Image) DTA3:FOLAIN ← PTR: (No Listing, Paper Tape Input) DTA3:FOLAIN ← DTA1:PIMAGE(T)(Teletype Image)	"FILE name NOT IN DIRECTORY" "DEVICE name DIRECTORY FULL" "DEVICE name UNAVAILABLE"
FOLA	3K	2	NONE	General Form: Binary,Listing ← Source 1, Source 2,.... PTP:, LPT: ← PTR: DTA2: BINARY, DTA3: LIST ← DTA1: FOLAIN ← DTA3: FOLAIN (No Output, Errors Only) PTP: ← PTR: (Binary Onto Paper Tape, No Listing) ,LPT: ← PTR: (No Binary, Listing on Printer)	"COMMAND ERROR" "DEVICE name NOT AVAILABLE" "CANNOT FIND FILE name" "CANNOT ENTER FILE name"
MACRO	9K	2	S=Surpress Listing L=Reinstate Listing E=Expand Macros on Listing X=Turn Off Expansion F=Fix Symbol Table	General Form: Binary, Listing ← Source 1, Source 2,.... PTP:,LPT: ← PTR:,,,(4 Input Files) ← DTA3: Macin (No Output,Errors on User TTY) DTA1: Bin Out ← DTA2: Macin (No Listing Errors on TTY) ,LPT: ← MTA1: Assemb (No Binary) DTA2: Bin Out, LPT: ← (S)DTA6:MAC1,(L)PTR:, (No Listing of File MAC1, Two Files from PTR)	Same Messages As FOLA (See Above)
LOADER	1K	1	S=Load Local Symbols W=Do Not Load Local Symbols L=Enter Library Search Mode N=Leave Library Search Mode A=Ignore All Subsequent Starting Addresses M=Print Storage Map R=Restart Loading I=Set Loader To Current State ALT MOD KEY=Begin Execution '(SINGLE QUOTE)=Enter Library Search for Previous File	General Form: Memory Map ← (Switches) Source 1,.... LPT: ← DTA3:BIN1(S), PTR:,DTA4:USRDDT ← (S)PTR: (Map On User TTY)	"ILLEGAL FORMAT" "DATA ERROR" "CORE EXCEEDED" "ILLEGAL COMMON" "COMMAND ERROR"