

[illegible][illegible]

05/31/72

01:49:17

```

*****
*****
**
** PDP-9 MINI TIME-SHARING SYSTEM **
** RESIDENT PROGRAM **
** DTSS:RES **
** MTSS:B01 **
**
*****
*****

```

100	.TITLE	PDP-9 MINI TIME-SHARING SYSTEM RESIDENT EXECUTIVE PROGRAM
110	.NAME	RES--B01
120	.ABS	
130	.PMC	ON
140	.INSRT	DEFINS
100	.IFUND	DEFINS

```
5720      ,LIST    ON
5730      ,END
150
160
170
180
190
200      *
210      *      THE PDP-9 MINI TIME-SHARING SYSTEM RESIDENT PROGRAM CONTAINS A NUMBER OF
220      *      SOMEWHAT DISJOINT ITEMS, MAINLY IT CONTAINS:
230      *
240      *      1) RESIDENT STORAGE:
250      *          A) TEMPORARY STORAGE USED BY THE RESIDENT PROGRAM ITSELF
260      *          B) A SET OF RESIDENT PARAMETERS FOR EACH TELETYPE
270      *          C) A TELETYPE I/O BUFFER FOR EACH TELETYPE
280      *          D) STORAGE TO RECORD ALLOCATION OF VARIOUS RESOURCES
290      *          E) FLAGS GIVING INFORMATION ON THE STATE OF THE SYSTEM
300      *          F) SOFTWARE-IMAGE FLAGS FOR CERTAIN HARDWARE DEVICES (E.G. PAPER TAPE PUNCH)
310      *          G) CATALOG INFORMATION TO RETRIEVE THE SWAPPER OVERLAY
320      *
330      *      2) ROUTINES TO HANDLE TELETYPE INPUT AND OUTPUT
340      *
350      *      3) ROUTINES TO SERVICE OTHER PROGRAM INTERRUPTS, INCLUDING THE CLOCK AND UNWANTED INTERRUPTS
360      *
370      *      4) A PHYSICAL DISK HANDLER
380      *
390      *      5) A ROUTINE TO SERVICE CAL INSTRUCTION ERRORS AND TO PREVENT CAL
400      *          (INDIRECT) FROM CRASHING THE SYSTEM.
410      *
420      *      ALSO IN THIS LISTING IS A COMPLETE LISTING OF THE CONTENTS OF USER JOB TABLES
430      *      AND PHANTOM JOB TABLES.
440      *
```

## INITIALIZE LOCATIONS 0-37

```

450      ,STIL INITIALIZE LOCATIONS 0-37
460      ,HEAD R
470      *
480      * LOCATIONS 0-37 CONTAIN <JMP ,> UNTIL SOMETHING ELSE IS PUT THERE.
490      * THIS IS PURELY AS A DEBUGGING AID TO TRAP MISCELLANEOUS TRANSFERS.
500      *
      000000 510      ,LOC 0
520      ,DET SAVE,CFF
530      ,DUP 1,40
      000000 600000 540      JMP .
550      ,DET RESTORE
560      *
570      * NORMALLY WHEN A USER'S PROGRAM IS RUNNING, LOCATIONS 0 & 10-17 CONTAIN
580      * WHAT HE THINKS THEY DO. THIS IS TO PERMIT HIS INDIRECT REFERENCES
590      * THROUGH HIS ALTO-INDEX REGISTERS
600      * AND LOCATION C TO WORK PROPERLY.
610      *
620      * LOCATIONS 1-7 & 20-37 ARE THEREFORE AVAILABLE FOR SYSTEM USE.
630      *
640      *
650      * A PROGRAM INTERRUPT HAS OCCURRED -- GO SERVICE IT
660      *
      000001 670      ,LOC 1
      000001 600165 680      JMP PISVC
```

R

## RESOURCE ALLOCATION AND TEMPORARY STORAGE

```

690      ,STITL  RESOURCE ALLOCATION AND TEMPORARY STORAGE
700      *
710      *   TEMPORARY VARIABLES STORAGE AND CONTROL LINE ALLOCATION RECORD
720      *
000002 000000 730      3TM21  ,DSA
000003 000000 740      3TM22  ,DSA
000004 000000 750      310TM  ,DSA
000005 000000 760      3AC     ,DSA
000006 000000 770      CNTRL   ,DSA
780      *
790      *
800      *   CURRENTLY CAL'S ARE NOT PERMITTED AT ALL, THERE IS A SPECIAL IOT INSTRUCTION
810      *   PROVISION FOR SYSTEM SERVICES, ALL CAL'S AUTOMATIC ERROR MESSAGE PRINTOUT
820      *   (PROVIDED BY THE USER'S MEMORY PROTECT OVERLAY),
830      *
840      *   EVENTUALLY USER CAL'S SHOULD BE HANDLED MUCH LIKE USR P! --
850      *   BASICALLY DO AN XCT OF USER LOCATION 21, SYSTEM CAL'S ARE
860      *   AN ERROR ANYWAY -- THE SYSTEM CURRENTLY DOES NOT USE ANY CAL'S.
870      *
000020 880      ,LOC    20
000020 000020 890      20      LOCATION 20 MUST ALWAYS CONTAIN AN ADDRESS FIELD OF 20 AS CAL,X PROTECTION
000021 200020 900      LAC     20      LOAD THE USER'S PC
000022 040000 910      DAC     0      AND SAVE IT FOR THE ERROR MESSAGE ROUTINE
000023 760020 920      LAW     20      RELOAD THE CAL,X PROTECTION
000024 040020 930      DAC     20      AND SET IT
000025 600342 940      JMP     ERRCAL  GO PRINT THE ERROR MESSAGE
950      *
960      *   TEMPORARY VARIABLE STORAGE
970      *
000026 000000 980      ,310   ,DSA
000027 000000 990      ,311   ,DSA
1000     *
1010     *   LOCATIONS 30-37 ARE THE DATA CHANNEL CELLS (30-31 FOR DECTAPE AND
1020     *   36-37 FOR THE DISK -- 32-35 ARE CURRENTLY UNUSED), CONSEQUENTLY
1030     *   30-31 & 36-37 MUST BE KEPT FREE, BUT 32-35 ARE AVAILABLE FOR
1040     *   SYSTEM USE.
1050     *
000030 1060     ,LOC    30
000030 776031 1070     -2000+.+1  DECTAPE WORD COUNT SO AS NOT TO DISTURB A DECTAPE READ-IN
1080     *
1090     *   TABLE OF AVAILABLE RESOURCES WHICH CAN ONLY BE ASSIGNED TO ONE
1100     *   USER AT A TIME, EACH RESOURCE'S ENTRY IS ZERO IF THE RESOURCE IS
1110     *   CURRENTLY UNASSIGNED, AND CONTAINS A POINTER TO THE USER'S RESIDENT
1120     *   PARAMETERS IF IT IS ASSIGNED.
1130     *
000032 1140     ,LOC    32
000032 000000 1150     RDT0    ,DSA  DECTAPE HANDLER
000033 000000 1160     RDT1    ,DSA  OTHER DECTAPE HANDLER
000034 000000 1170     RACS    ,DSA  ACCUMULATOR SWITCHES
1180     *
1190     *   RCORE CONTAINS THE STATUS OF NON-PROTECTED CORE, IT CONTAINS
1200     *   THE USER NUMBER OF THE CURRENTLY ACTIVE JOB, IF THERE IS ONE.

```

R

## RESOURCE ALLOCATION AND TEMPORARY STORAGE

```

1210 * ELSE IT IS ZERO, IF RCORE IS NON-ZERO, IT IS ASSUMED THERE
1220 * IS IN CORE A MEMORY PROTECTION OVERLAY TO GO WITH THE ACTIVE
1230 * JOB, IN THIS CASE THE EXEC WILL FEEL FREE TO JUMP TO THE OVERLAY
1240 * AREA TO ACCESS ROUTINES (E.G. TO CHECK THE NEED TO GENERATE A
1250 * USER PROGRAM INTERRUPT,)
000035 000000 1260 *
1270 RCORE ,DSA NON-PROTECTED CORE STATUS -- 0= NO ACTIVE USER; ELSE CONTAINS USER NUMBER
1280 *
1290 *
1300 * RESIDENT CATALOG -- ALL OVERLAY FILES ARE CATALOGED HERE TO MINIMIZE SWAPPER USAGE.
1310 *
000040 1320 ,LOC 40
000040 000000 1330 SWPS ,DSA PHYSICAL DISK ADDRESS OF THE OVERLAY
000041 000777 1340 SOVSTRT-1 OVERLAY FILE CORE ADDRESS -1
000042 777100 1350 -SOVLEN OVERLAY FILE (TWO'S COMPLEMENT) WORD COUNT
000043 000002 1360 SDKRD DISK READ COMMAND
1370 *
000044 000000 1380 CSWP ,DSA SWAPPER PHYSICAL DISK ADDRESS
000045 000000 1390 CMP1 ,DSA MEMORY PROTECTION #1 PHYSICAL DISK ADDRESS
000046 000000 1400 CMP2 ,DSA MEMORY PROTECTION #2 PHYSICAL DISK ADDRESS
000047 000000 1410 CSPL ,DSA SPECIAL IOT HANDLER PHYSICAL DISK ADDRESS
1420 *
1430 *
1440 * TEMPORARY VARIABLE STORAGE
1450 *
000050 000000 1460 3TM20 ,DSA
000051 000000 1470 3TEM0 ,DSA
000052 000000 1480 3TEM1 ,DSA
000053 000000 1490 3TEM2 ,DSA
000054 000000 1500 3TEM3 ,DSA
000055 000000 1510 3TEM4 ,DSA
000056 000000 1520 3TEM5 ,DSA
000057 000000 1530 3TEM6 ,DSA

```

## R TELETYPE BUFFERS AND CONSTANTS

```

1540      .STITL  TELETYPE BUFFERS AND CONSTANTS
1550      *
1560      *
1570      *   KEYBOARD BUFFERS MUST BE OF A CERTAIN MINIMUM SIZE (SEE DEFINITIONS).
1580      *   HOWEVER, THE LARGER THEY CAN BE, THE BETTER, TELEPRINTER I/O ROADBLOCK
1590      *   OCCURS WHEN THE BUFFER IS FULL OF OUTPUT AND IS NOT RELIEVED UNTIL THERE ARE
1600      *   ONLY ENOUGH CHARACTERS REMAINING IN THE BUFFER TO COVER THE WORST CASE
1610      *   REMAINING TIME UNTIL THE PROGRAM COULD GET BACK INTO CORE TO PUT MORE
1620      *   OUTPUT IN THE BUFFER.  THUS THE TELEPRINTER IS KEPT CONTINUALLY BUSY PRINTING
1630      *   AS LONG AS THE PROGRAM HAS OUTPUT TO PRINT.
1640      *
1650      *   *****WARNING*****: CTNAM (& L1NAM & L2NAM) SERVE MANY FUNCTIONS.
1660      *   THEY ARE OFTEN REFERRED TO AS $SYSTEM4.  THEY SERVE AS THE NAME
1670      *   OF THE USER'S CORE-IMAGE FILE ON THE DISK, WITH A (1) ADDED TO
1680      *   THEM THEY SERVE THE SAME PURPOSE FOR THEIR OWN PHANTOM PROGRAMS.
1690      *   THEY ARE POINTERS NOT ONLY TO THE END OF THEIR TELETYPE BUFFERS
1700      *   +1, BUT ALSO TO THE START OF THE RESIDENT PARAMETER LIST-1.
1710      *   WITH ONE OF THEM IN THE AC A <JMS $IO,IN> INSTRUCTION WILL SET
1720      *   UP THAT USER'S I/O PARAMETERS IN THE TEMPORARY VARIABLES
1730      *   USED FOR EXECUTIVE RE-ENTRANCE.
1740      *
1750      *   I WOULDN'T BE SURPRISED IF THERE ARE ALSO OTHER FUNCTIONS I AM FORGETTING.
1760      *
1770      *
1780      *
1790      *
1800      *
1810      *
1820      *
1830      *
1840      *
1850      *
1860      *
1870      *
1880      *
1890      *
1900      *
1910      *
1920      *
1930      *
1940      *
1950      *
1960      *
1970      *
1980      *
1990      *
2000      *
2010      *
2020      *
2030      *
2040      *
2050      *

```

000060 1760 CTBFR ,BLOCK SKBLEN

000076 000000 1770 LQLOK ,DSA

000077 777770 1780 ,SKBNUM

000100 000060 1790 CTBIN CTBFR

000101 000060 1800 CTBFR

000102 000000 1810 CTFLG ,DSA

1820

1830

1840

1850

1860

1870

1880

1890

1900

000103 000060 1910 CTBFR

000104 000076 1920 CTNAM CTBIN-2

000105 700406 1930 TLR

000106 741000 1940 SKP

1950

000107 1960 L1BFR ,BLOCK SKBLEN

000125 000000 1970 L1LOK ,DSA

000126 777770 1980 ,SKBNUM

000127 000107 1990 L1BIN L1BFR

000130 000107 2000 L1BFR

000131 100000 2010 L1FLG 100000

000132 000107 2020 L1BFR

000133 000125 2030 L1NAM L1BIN-2

000134 704006 2040 TLRSLT1

000135 704006 2050 TLRSLT1

CONSOLE TELETYPE MONITOR REQUEST  
NUMBER OF PARAMETERS FOLLOWING  
BIT 0 = COUNT ALREADY IN; BITS 5-17 = ACTIVE ADDRESS  
BIT 0 = COUNT ALREADY OUT; BITS 5-17 = ACTIVE ADDRESS  
SOFTWARE TELETYPE I/O FLAG. BIT:  
0! OUTPUT-IN-PROGRESS FLAG  
1! TELEPRINTER I/O ROADBLOCKED FLAG  
2! KEYBOARD I/O ROADBLOCKED FLAG  
3! KEYBOARD FLAG  
4! TELEPRINTER FLAG  
5! PI INTERRUPT PENDING  
6! KEYBOARD CHARACTER ECHOED FLAG (0=YES 1=NEEDS AN ECHO)  
7! I/O BUFFER TYPE ( 0=INPUT 1=OUTPUT)  
10-17! KEYBOARD BUFFER  
(CONSTANT) START OF CONTROL TELETYPE BUFFER  
(CONSTANT) END OF CONSOLE TELETYPE BUFFER +1 -- ALSO SERVES AS USER IDENTITY  
(CONSTANT) PRINT INSTRUCTION FOR CONSOLE TELETYPE  
CONSOLE TELETYPE WILL NOT NEED TO ECHO ANY CHARACTERS BY SOFTWARE

PARAMETER LIST SAME AS FOR CT ABOVE

START IN KEYBOARD I/O ROADBLOCK CONDITION

R

## TELETYPE BUFFERS AND CONSTANTS

	000136	2060		
		2070	L2BFR	,BLOCK SKBLEN
000154	000000	2080	L2LOK	,QSA
000155	777770	2090		=SKBNUM
000156	000136	2100	L2BIN	L2BFR
000157	000136	2110		L2BFR
000160	100000	2120	L2FLG	100000
000161	000136	2130		L2BFR
000162	000154	2140	L2NAM	L2BIN-2
000163	704026	2150		TLSLT2
000164	704026	2160		TLSLT2



R

## PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```

2170      ,STITL PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE
2180      *
2190      *
2200      * WHEN A PROGRAM INTERRUPT OCCURS, CONTROL IS ALWAYS TRANSFERRED TO HERE.
2210      * SAVE REGISTERS AC, 10, AND 11 ON INTERRUPTS. ROUTINES USING MQ OR SC MUST SAVE THEIR OWN.
2220      *
2230      * PISVC
000165      2230      ...
000165 040005 2240      DAC      3AC
000166 200010 2250      LAC      10
000167 040026 2260      DAC      .310
000170 200011 2270      LAC      11
000171 040027 2280      DAC      .311
2290      *
2300      * WHEN EXIT FROM SERVICING THIS PROGRAM INTERRUPT FINALLY OCCURS,
2310      * IT WILL BE WITH THE SEQUENCE:
2320      * DBK
2330      * JMP <USER CORE>
2340      * UNLESS THE INTERRUPT WAS CAUSED BY THE USER TRYING TO DO A DBR
2350      * INSTRUCTION. IF HE HAS DONE A LEGAL DBR INSTRUCTION THE EXIT
2360      * SEQUENCE WILL HAVE THE DBK REPLACED BY A DBR, SO THAT THE RESTORE
2370      * EFFECT WILL OCCUR WHEN THE USER EXPECTS IT TO.
2380      * IN EITHER CASE THE STATE OF THE MACHINE WILL BE RESORED BY
2390      * EXECUTIVE SOFTWARE IMMEDIATELY PRIOR TO EXIT.
2400      *
2410      * IF THE USER TRIED TO DO A DBR INSTRUCTION, THE MEMORY PROTECT
2420      * ROUTINES WILL GIVE AN ERROR MESSAGE IF IT CAN HURT THE SYSTEM,
2430      * (CURRENTLY THEY REQUIRE A DBR TO BE FOLLOWED BY A JMP (INDIRECT)
2440      * THROUGH A WORD WITH THE MEMORY PROTECT BIT ON.)
2450      *
2460      * THE EFFECT OF ALL OF THIS IS TO PERMIT A REASONABLE AMOUNT OF
2470      * TRANSPARENCY TO BE PRESERVED FOR USER PROGRAMS RUNNING WITH
2480      * THE PROGRAM INTERRUPT SYSTEM ON, WHILE NOT ALLOWING AN UNWANTED
2490      * RESTORE FUNCTION TO BE LEFT HANGING AROUND, IF ONE WERE LEFT,
2500      * A NOT-VERY-CLEVER USER COULD CRASH THE SYSTEM.
2510      *
000172 200665 2520      LAC      DBK
000173 040303 2530      DAC      PIOUT      SET THE STANDARD EXIT -- PROGRAM WILL RESTORE THE LINK
2540      *
2550      * NOW FIND OUT WHAT CAUSED THE INTERRUPT
2560      * IF THE INTERRUPT WAS CAUSED BY A HARDWARE INTERRUPT FLAG SETTING,
2570      * CLEAR IT AND TRANSFER TO A ROUTINE TO SET THE CORRESPONDING SOFTWARE
2580      * FLAG AND TEST FOR A USER PROGRAM INTERRUPT.
2590      *
000174 701713 2600      MPSKIMPCV+10
000175 741000 2610      SKP
000176 601000 2620      JMP      SMPST      MEMORY PROTECTION VIOLATION
000177 700001 2630      CLSF
000200 741000 2640      SKP
000201 600317 2650      JMP      CLK      SERVICE A CLOCK INTERRUPT
000202 700403 2660      TSF!TCF
000203 741000 2670      SKP
000204 600347 2680      JMP      CTOUT      CONSOLE TELEPRINTER

```

R

## PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

000205	700313	2690		KSF1KRB	
000206	741000	2700		SKP	
000207	600346	2710		JMP	CTKBD CONSOLE KEYBOARD
000210	704113	2720		KSFLT1:KRBLT1	
000211	741000	2730		SKP	
000212	600351	2740		JMP	L1KBD LT#1 KEYBOARD
000213	704133	2750		KSFLT2:KRBLT2	
000214	741000	2760		SKP	
000215	600354	2770		JMP	L2KBD LT#2 KEYBOARD
000216	704003	2780		TSFLT1:TCFLT1	
000217	741000	2790		SKP	
000220	600352	2800		JMP	L1OUT LT#1 TELEPRINTER
000221	704023	2810		TSFLT2:TCFLT2	
000222	741000	2820		SKP	
000223	600355	2830		JMP	L2OUT LT#2 TELEPRINTER
000224	700203	2840		PSF1PCF	
000225	600231	2850		JMP	.+4
000226	100246	2860		JMS	FLAG SERVICE THE PAPER TAPE PUNCH INTERRUPT
000227	000000	2870	PFLAG	,DSA	SOFTWARE PAPER TAPE PUNCH FLAG
000230	000000	2880	RPTP	,DSA	PAPER TAPE PUNCH ALLOCATION WORD
000231	700103	2890		RSF1PCF	
000232	600236	2900		JMP	.+4
000233	100246	2910		JMS	FLAG SERVICE THE PAPER TAPE READER INTERRUPT
000234	000000	2920	RFLAG	,DSA	SOFTWARE PAPER TAPE READER FLAG
000235	000000	2930	RPTR	,DSA	PAPER TAPE READER ALLOCATION WORD
000236	704405	2940		SPB1CPB	
000237	600243	2950		JMP	.+4
000240	100246	2960		JMS	FLAG SERVICE GRAPHICS II PUSHBUTTONS INTERRUPT
000241	000000	2970	PBFLAG	,DSA	SOFTWARE PUSHBUTTONS FLAG
000242	000000	2980	RSCO	,DSA	GRAPHICS II ALLOCATION WORD
000243	707001	2990		DSSF	
000244	600267	3000		JMP	PISV2 IT IS AN UNKNOWN INTERRUPT
000245	600253	3010		JMP	DKSVC SERVICE A DISK INTERRUPT
		3020	*		
		3030	*		
		3040	*		RECORD THE HARDWARE FLAG IN THE CORRESPONDING SOFTWARE FLAG.
		3050	*		THE SKIP CHAIN WILL HAVE ALREADY CLEARED THE HARDWARE FLAG.
		3060	*		THEN TEST FOR A USER PROGRAM INTERRUPT.
000246		3070		ENTER	FLAG SAVE THE TYPE OF INTERRUPT HERE
				,PMC	SAVE.ON
000246	740040		FLAG	XX	
				,PMC	RESTORE
000247	460246	3080		INX	FLAG,X SET THE SOFTWARE FLAG
000250	440246	3090		INX	FLAG BUMP THE TABLE POINTER
000251	220246	3100		LAC	FLAG,X LOAD THE ALLOCATION WORD
000252	600411	3110		JMP	PITST SEE WHETHER OR NOT TO GENERATE A USER PROGRAM INTERRUPT

R

## PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```

3120      ,EJECT
3130      *
3140      *   AN INTERRUPT FROM A USER DISK OPERATION HAS BEEN RECEIVED.
3150      *   SAVE ALL OF HIS INFORMATION, AND CLEAR THE DISK FOR POSSIBLE SYSTEM
3160      *   DISK USE.
3170      *
000253 140266 3180  DKSVC  DZM      DKLOK      CLEAR THE USER-USING-DISK FLAG
000254 707272 3190      DSR$+10
000255 041765 3200      DAC      DSTAT      SAVE THE USER'S DISK STATUS REGISTER
000256 441761 3210      INX      DFLAG      SET THE USER'S DISK FLAG
000257 707023 3220      DSCC:DRA  DISABLE POSSIBLE DISK FREEZE THROWN IN ON GENERAL PRINCIPLES
000260 041762 3230      DAC      DAP0      SAVE THE USER'S DISK REGISTER APO
000261 707242 3240      DSCD      CLEAR THE STATUS REGISTER AND DISK FLAG
3250      *
3260      *   WE HAVE HAD THE DISK LOCKED OUT FOR A WHILE, NOW. SO SEE IF A CLOCK
3270      *   INTERRUPT OCCURRED IN THE MEANTIME.
3280      *
000262 200316 3290      LAC      CLKLOK      LOAD THE CLOCK INTERRUPT RECORD
000263 741200 3300      SNA              SKIP IF THERE HAS BEEN ONE
000264 600270 3310      RET      PIDON      IF NONE, EXIT NORMALLY
000265 600327 3320      JMP      CLKST      IF THERE HAS BEEN ONE, DELIVER IT NOW
3330
000266 000000 3340      DKLOK  ,DSA      MINUS (SYSTEM-USING-DISK); PLUS (USER-USING-DISK); OR ZERO (DISK FREE)
3350      *
3360      *
3370      *   AN UNKNOWN INTERRUPT OCCURRED -- CLEAR REMAINING FLAGS AND IGNORE IT
3380      *
000267      3390      PISV2  ...
000267 703302 3400      CAP      GET A CLEAN START
3410      *
3420      *
3430      *   WE ARE DONE SERVICING THE LATEST PROGRAM INTERRUPT. NOW FIX THINGS UP
3440      *   AND RETURN TO THE USER.
3450      *
3460      *   RESTORE THE REGISTERS ON EXIT
3470      *
000270 200026 3480      PIDON  LAC      .310
000271 040010 3490      DAC      10
000272 200027 3500      LAC      .311
000273 040011 3510      DAC      11
3520      *
3530      *   SET UP THE ADDRESS, LINK AND MEMORY PROTECT FOR THE RETURN
3540      *
000274 200000 3550      PIDN2  LAC      0
000275 500663 3560      AND      ADR$S
000276 340664 3570      TAD      JMP
000277 040304 3580      DAC      PIGO
000300 100305 3590      JMS      3REST
000301 701742 3600      MPEV
000302 700042 3610      ION
000303 740040 3620      PIOJT  XX
000304 740040 3630      PIGO   XX

```

SET UP RETURN (DIRECT)  
RESTORE LINK, USER LOCATION 0, AC, AND MEMORY PROTECT  
RETURN HERE IF MEMORY PROTECT BIT (BIT 2) WAS ON  
RETURN HERE IF MEMORY PROTECT BIT (BIT 2) WAS OFF  
DBK (UNLESS MEM PRO DBR INTERRUPT -- THEN DBR)  
RETURN (DIRECT)

R

## PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```

3640 *
3650 * SUBROUTINE TO RESTORE THE USER'S LINK, AC, AND LOCATION 0.
3660 * THE SUBROUTINE RETURNS TO THE ENABLE USER MODE (MPEU) INSTRUCTION
3670 * ONLY IF USER MODE WAS ENABLED WHEN THE INTERRUPT OCCURRED.
3680 * OTHERWISE THAT INSTRUCTION IS SKIPPED, THIS SUBROUTINE IS
3690 * NECESSARY TO CORRECTLY ACCOMPLISH THE EXIT BY THE SEQUENCE:
3700 * DBK
3710 * JMP <USER CORE>
3720 *
000305 3730 3REST ENTER
,PMC SAVE,ON
000305 740040 XX
,PMC RESTORE
000306 200000 3740 LAC 0 LOAD THE STATE OF THE MACHINE WHEN INTERRUPTED
000307 742010 3750 RTL MOVE THE MEMORY PROTECT BIT TO AC(0)
000310 740120 3760 SMA,RR RESTORE THE CORRECT LINK FOR RETURN
000311 440305 3770 INX 3REST SET THE RETURN TO +2 FOR NO MEMORY PROTECTION ON
000312 201713 3780 LAC .0 LOAD THE USER'S LOCATION 0
000313 040000 3790 DAC 0 RESTORE THE USER'S LOCATION ZERO BEFORE RETURN TO HIM
000314 200005 3800 LAC 3AC RESTORE THE USER'S AC
000315 620305 3810 JMP 3REST,X

```

R

## DEVICE INTERRUPT SERVICE ROUTINES

## ,STITL DEVICE INTERRUPT SERVICE ROUTINES

```

3820
3830
3840 *
3850 *
3860 *
3870 *
3880 *
3890 *
3900 *
3910 *
3920 *
3930 *
3940 *
3950 *
000316 000000 3960 CLKLOK ,DSA
000317 440316 3970 CLK INX CLKLOK RECORD WE HAVE RECEIVED ANOTHER CLOCK INTERRUPT
000320 740000 3980 NOP GENERAL PRINCIPLES -- IN CASE IT EVER DOES SKIP
000321 700044 3990 CLON NEED THIS TO CLEAR THE FLAG, EVEN THOUGH THE CLOCK IS NEVER TURNED OFF
000322 777730 4000 LAW -CLKMAX
000323 040007 4010 DAC 7 IN ANY CASE, RESET THE CLOCK
000324 200266 4020 LAC DKLOK
000325 740200 4030 SZA
000326 600270 4040 RET PIDON SKIP IF THE SOFTWARE FLAG SAYS THE DISK IS FREE
ELSE EXIT -- CAN'T DO ANYTHING ELSE HERE UNTIL THE DISK IS FREE
4050
000327 121003 4060 CLKST JMS SRDBLK,X SEE WHO IS THE NEXT NON-ROADBLOCKED USER
000330 200055 4070 LAC 3TEM4 LOAD HIS USER NUMBER
000331 341774 4080 TAD TYPE CONVERT TO A FILENAME
000332 541772 4090 SAD NAME SKIP IF HE IS NOT ALREADY RUNNING
000333 600270 4100 RET PIDON SAME USER, SO JUST RESTART HIM
000334 761003 4110 LAW SSWCLK THERE IS ANOTHER FREE USER, SO REQUEST A CLOCK SWAP
4120 *
4130 *
4140 *
000335 040666 4150 SWAP DAC DO SET THE RETURN ADDRESS
000336 200044 4160 SWAP1 LAC CSWP
000337 040040 4170 DAC SWPS SET THE SWAPPER'S PHYSICAL DISK ADDRESS
000340 760037 4180 LAW SWPS-1 SET A POINTER TO THE CATALOG DATA FOR THE SWAPPER
000341 600667 4190 JMP DO+1 GET THE SWAPPER
4200 *
4210 *
4220 *
000342 760014 4230 ERRCAL LAW 12.
000343 041706 4240 DAC UTEM2 PASS THE ERROR MESSAGE NUMBER TO THE ERROR ROUTINE
000344 761004 4250 LAW SSWERR
000345 600335 4260 JMP SWAP CALL THE ERROR MESSAGE PROGRAM

```

R

## KEYBOARD INPUT ROUTINES

## ,STILL KEYBOARD INPUT ROUTINES

```

4270
4280
4290 *
4300 * THE FOLLOWING THREE LINES OF CODE FOR EACH TELETYPE
4310 * WHICH ARE UNIQUE TO THEIR RESPECTIVE TELETYPES. THESE INSTRUCTIONS
4320 * PERMIT ALL INPUT AND OUTPUT TO BE CARRIED ON BY COMMON ROUTINES.
4330 *
4340 * ON INPUT, A JMS KBDIN PROVIDES THE INPUT ROUTINE WITH A POINTER TO
4350 * THE TELETYPE'S OWN RESIDENT STORAGE, KBDIN RETRIEVES THIS BY AN
4360 * XCT KBDIN,X. KBDIN DOES NOT RETURN TO HERE -- THE JMS IS MERELY
4370 * TO PROVIDE A POINTER TO THE CALLING TELETYPE.
4380 *
4390 * ON OUTPUT KBDOT IS ENTERED WITH THE TELETYPE'S UNIQUE POINTER TO
4400 * ITS RESIDENT STORAGE ALREADY IN THE ACCUMULATOR,
4410 *
4420 * NOTE THAT <--KBD> IS THE ENTRANCE FOR AN INTERRUPT FROM A TELETYPE
4430 * KEYBOARD, WHILE <--OUT> IS THE ENTRANCE FOR AN INTERRUPT FROM A
4440 * TELETYPE TELEPRINTER.
4450 *
4460
4470 *
4480 * CONSOLE TELETYPE
4490 *
000346 4500 CTKBD ... SERVICE INTERRUPTS FROM CONSOLE KEYBOARD
000346 100357 4510 JMS KBDIN PROCESS THE INPUT
000347 4520 CTOUT ... SERVICE INTERRUPTS FROM CONSOLE TELEPRINTER
000347 760076 4530 LAW CTBIN-2 IDENTIFY YOURSELF
000350 600441 4540 JMP KBDOT PRINT THE NEXT CHARACTER, IF ANY
4550 *
4560 * TELETYPE #1
4570 *
000351 4580 L1KBD ...
000351 100357 4590 JMS KBDIN PROCESS THE INPUT
000352 760125 4600 LAW L1BIN-2 USED AS SYSTEM JOB NUMBER
000353 600441 4610 JMP KBDOT
4620 *
4630 * TELETYPE #2
4640 *
000354 4650 L2KBD ...
000354 100357 4660 JMS KBDIN
000355 760154 4670 LAW L2BIN-2
000356 600441 4680 JMP KBDOT
4690

```

```

R
ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES
,STIL ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES
4700
4710 *
4720 *
4730 * KBDIN HANDLES INPUT FROM ANY KEYBOARD.
4740 *
000357 4750 KBDIN ENTER
,PMC SAVE,ON
000357 740040 XX
,PMC RESTORE
4760 *
4770 * INITIALIZE THE KEYBOARD INPUT ROUTINE
4780 *
000360 040002 4790 DAC 3TM21 SAVE THE CHARACTER
000361 420357 4800 XCT KBDIN,X LOAD THE CALLER'S IDENTITY
000362 100526 4810 JMS IO,IN SET UP THE REENTRANT TEMPS
000363 200002 4820 LAC 3TM21 RESTORE THE CHARACTER
4830 *
4840 * CHECK FOR SPECIAL CHARACTERS
4850 *
4860 * A NULL (BREAK OR CONTROL SHIFT 'P') WILL STOP THE USER'S CURRENT JOB
4870 * CANCEL ANY INCOMPLETE I/O, AND RESTART THE MONITOR.
4880 *
4890 * IF THE DELETE OPTION IS ON, A CONTROL 'X' WILL KILL THE REST OF THE
4900 * CURRENT OUTPUT BUFFER WITHOUT AFFECTING THE INPUT STATUS.
4910 *
000364 500654 4920 AND BL7 KEEP ONLY THE LAST 7 BITS -- NECESSARY TO CHECK FOR A NULL
000365 741200 4930 SNA CHECK FOR NULL (= CONTROL SHIFT P) ON INPUT
000366 600425 4940 JMP KBD0 IF NULL, CANCEL JOB AND RESTART MONITOR
4950 *
4960 * BEGIN TO PROCESS NORMAL INPUT, CLEAR THE FOLLOWING BITS:
4970 * BIT2: KEYBOARD I/O ROADBLOCK FLAG
4980 * BIT3: SOFTWARE KEYBOARD FLAG
4990 * BIT6: CHARACTER ECHO FLAG ( 0 = CHARACTER HAS BEEN ECHOED )
5000 * BITS 10-17: ONE CHARACTER KEYBOARD BUFFER
5010 *
5020 * THEN SET THE FOLLOWING BITS:
5030 * BIT3: SOFTWARE KEYBOARD FLAG
5040 * BIT6: CHARACTER ECHO FLAG ( 1 = CHARACTER HAS NOT YET BEEN ECHOED )
5050 * BITS 10-17: SAVE THE CHARACTER JUST TYPED
5060 *
000367 200053 5070 LAC 3TEM2 LOAD THE TELETYPE SOFTWARE FLAGS
000370 500720 5080 AND (633400) CLEAR KEYBOARD & KEYBOARD I/O ROADBLOCK FLAGS, & KEYBOARD SOFTWARE BUFFER
000371 240647 5090 XOR BIT36 SET THE KEYBOARD AND CHARACTER-NOT-ECHOED FLAGS
000372 240002 5100 XOR 3TM21 PUT THE LATEST CHARACTER IN THE SOFTWARE KEYBOARD BUFFER
000373 040053 5110 DAC 3TEM2 SAVE THE SOFTWARE FLAGS
5120 *
5130 * CHECK FOR AN INPUT BUFFER, IF NOT, THE CHARACTER CANNOT YET BE PACKED
5140 *
000374 500652 5150 AND BIT7 RECOVER THE BUFFER TYPE
000375 740200 5160 SZA SKIP IF IT IS AN INPUT BUFFER
000376 600401 5170 JMP KBD05 ELSE GO DIRECTLY TO THE OUTPUT-IN-PROGRESS TEST
000377 100561 5180 JMS PUTIN INPUT BUFFER -- TRY TO PACK THE CHARACTER

```

R			ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES		
000400	400056	5190	XCT	3TEM5	ECHO A GARBAGE CHARACTER IF IT IS FULL
		5200	*		
		5210	*		CHECK FOR OUTPUT-IN-PROGRESS. IF SO, THE INPUT CHARACTER CANNOT BE
		5220	*		ECHOED UNTIL THE NEXT TELEPRINTER INTERRUPT.
		5230	*		
	000401	5240	KBD05	...	
000401	200053	5250	LAC	3TEM2	RELOAD THE SOFTWARE FLAGS
000402	741100	5260	SPA		SKIP IF NO OUTPUT IS IN PROGRESS
000403	600417	5270	JMP	KBD4	ELSE DON'T ECHO IT
		5280	*		
		5290	*		OUTPUT IS NOT IN PROGRESS.
		5300	*		TURN ON BIT 0: THE OUTPUT-IN-PROGRESS FLAG (IF THE TERMINAL REQUIRES A SOFTWARE ECHO)
		5310	*		TURN OFF BIT 6: THE CHARACTER-NOT-ECHOED FLAG ( 0=OFF )
		5320	*		
		5330	*		ECHO THE CHARACTER IF THIS IS A TERMINAL REQUIRING A SOFTWARE ECHO
		5340	*		
000404	240651	5350	XOR	BIT6	TURN OFF THE CHARACTER-NOT-ECHOED FLAG
000405	400057	5360	XCT	3TEM6	ECHO THE CHARACTER IF SOFTWARE ECHO TERMINAL; ELSE SKIP
000406	240646	5370	XOR	BIT0	SET THE OUTPUT-IN-PROGRESS FLAG (FOR SOFTWARE ECHO TERMINALS)
	000407	5380	KBD2	...	
000407	040053	5390	DAC	3TEM2	SAVE THE UPDATED TELETYPE FLAGS
		5400	*		
		5410	*		RESIDENT EXECUTIVE ROUTINE TO GENERATE A SIMULATED PROGRAM
		5420	*		INTERRUPT FOR THE USER IF APPROPRIATE. IF THE USER IS CURRENTLY
		5430	*		RUNNING, TRANSFER TO HIS MEMORY PROTECTION OVERLAY TO SEE ABOUT
		5440	*		GIVING HIM THE SIMULATED INTERRUPT. OTHERWISE JUST SET THE
		5450	*		PROGRAM INTERRUPT BIT ON IN HIS I/O FLAGS WORD, AND THE SWAPPER
		5460	*		WILL SEE ABOUT GENERATING THE SIMULATED INTERRUPT (IF NECESSARY)
		5470	*		THE NEXT TIME HE IS SWAPPED IN.
		5480	*		
000410	200055	5490	KBD9	LAC	3TEM4
000411	541771	5500	PITST	SAD	NUMBR
000412	600421	5510		JMP	KBD7
000413	200053	5520	KBD5	LAC	3TEM2
000414	500660	5530		AND	CB5
000415	240650	5540		XOR	BIT5
000416	040053	5550	KBD41	DAC	3TEM2
000417	100540	5560	KBD4	JMS	IO,OT
000420	600270	5570		RET	PIDON
		5580			
000421	200035	5590	KBD7	LAC	RCORE
000422	741200	5600		SNA	
000423	600413	5610		JMP	KBD5
000424	601001	5620		JMP	SPINT
					NO
					YES



R

ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```

5630      ,EJECT
5640      *
5650      *   THE USER TYPED A NULL, STOP HIS CURRENT JOB, KILL ALL INCOMPLETE
5660      *   I/O, AND SET UP A MONITOR REQUEST FOR HIM.
5670      *
000425 100553 5680      KBD0  JMS      NEWBR      NULL KILLS ALL OLD I/O
000426 100540 5690      JMS      IO.OT      CLEAN UP THIS USER
000427 200055 5700      LAC      3TEM4      LOAD IDENTITY OF ONE WHO PRINTED A NULL
000430 060055 5710      DAC      3TEM4,X    AND SET IT AS A MONITOR REQUEST
000431 541771 5720      SAD      NUMBR      SEE IF HE IS THE CURRENT USER
000432 741000 5730      SKP
000433 600270 5740      RET      PIDON      NO -- EXIT
000434 200266 5750      LAC      DKLOK
000435 740200 5760      SZA
000436 600270 5770      RET      PIDON      SEE IF THE DISK IS AVAILABLE TO GET THE SWAPPER
000437 761002 5780      MTR1  LAM      SSWMTR   NO -- EXIT
000440 600335 5790      JMP      SWAP      YES -- GET THE MONITOR

```

R

## ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```

5800      ,EJECT
5810      *
5820      *
5830      *   KBDOT HANDLES OUTPUT TO ANY TELETYPE FROM ITS ROTARY LINE BUFFER.
5840      *   SUFFICIENT OUTPUT CLEARS THE TELEPRINTER I/O ROADBLOCKED CONDITION, EMPTYING THE
5850      *   BUFFER CLEARS THE OUTPUT-IN-PROGRESS CONDITION.
5860      *   THE OUTPUT BUFFER IS FILLED BY THE MEMORY PROTECTION ROUTINES, THEY ALSO
5870      *   SET THE OUTPUT-IN-PROGRESS FLAG.
5880      *
5890      *   INITIALIZE....CHECK FOR OUTPUT IN PROGRESS....IF NONE, ASSUME THE INTERRUPT
5900      *   WAS GENERATED BY THE LAST CHARACTER OF THE PRECEEDING MESSAGE AND EXIT IMMEDIATELY.
000441    5910      *
000441    5920      *   KBDOT
000441 100526 5920      JMS      IO,IN
000442 200053 5930      LAC      3TEM2      LOAD THE TELETYPE SOFTWARE FLAGS
000443 740100 5940      SMA      SKIP IF OUTPUT IS IN PROGRESS
000444 600410 5950      JMP      KBD9      ELSE EXIT THROUGH THE SIMULATED PI ROUTINE
5960      *
5970      *   SEE WHETHER OR NOT TO ECHO A CHARACTER
5980      *   ECHO IT IF NECESSARY, AND EXIT, ELSE CONTINUE.
5990      *
000445 500651 6000      AND      BIT6      RECOVER THE CHARACTER-NOT-ECHOED FLAG
000446 741200 6010      SNA      SKIP IF THERE IS UN-ECHOED INPUT
000447 600454 6020      JMP      KBD01     NO ECHO NEEDED -- PROCEED TO NORMAL OUTPUT
000450 240053 6030      XOR      3TEM2     TURN OFF THE CHARACTER-NOT-ECHOED FLAG
000451 400057 6040      XCT      3TEM6     ECHO THE CHARACTER FOR SOFTWARE ECHO TERMINALS; ELSE SKIP
000452 600416 6050      JMP      KBD41     SOFTWARE ECHO TERMINALS EXIT HERE
000453 040053 6060      DAC      3TEM2     HARDWARE ECHO TERMINALS GET HERE
6070      *
6080      *   CHECK TO SEE WHETHER THE BUFFER IS AN INPUT BUFFER OR AN OUTPUT
6090      *   BUFFER CURRENTLY. IF IT IS AN OUTPUT BUFFER, CONTINUE WITH
6100      *   OUTPUT, IF IT IS AN INPUT BUFFER, BRANCH TO THE OUTPUT DONE ROUTINE.
000454    6110      *
000454 200053 6120      *   KBD01
000455 500652 6130      LAC      3TEM2     LOAD THE SOFTWARE FLAGS
000456 740200 6140      AND      BIT7      RECOVER THE BUFFER TYPE
000457 600463 6150      SZA      SKIP IF THE I/O BUFFER IS AN INPUT BUFFER
000460 200053 6160      JMP      KBD02     ELSE CONTINUE WITH NORMAL OUTPUT
000461 500656 6170      LAC      3TEM2     LOAD THE SOFTWARE FLAGS
000462 600416 6180      AND      CB0      CLEAR THE OUTPUT IN PROGRESS FLAG
6190      *
6200      *   CHECK FOR I/O ROADBLOCK REMOVAL -- REMOVE WHEN BARELY ENOUGH
6210      *   CHARACTERS ARE LEFT TO COVER OTHER USERS' MAXIMUM CPU TIMES
6220      *
000463 777777 6230      *   KBD02
000464 340052 6240      LAW      -1
000465 500663 6250      TAD      3TEM1
000466 740001 6260      AND      ADRSS
000467 040002 6270      CMA
000470 200051 6280      DAC      3TM21      SET THE (TWO'S COMPLEMENT) START-OF-OUTPUT ADDRESS
000471 500663 6290      LAC      3TEM0      LOAD THE END-OF-OUTPUT ADDRESS
000472 340002 6300      AND      ADRSS
000473 741100 6310      TAD      3TM21      SUBTRACT THE START
      SPA      SKIP IF O.K. (AC = AMOUNT OF OUTPUT STILL TO GO)

```

R			ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES		
000474	340721	6320	TAD	(SKBLEN)	ELSE MAKE AC = AMOUNT OF OUTPUT TO GO
000475	340722	6330	TAD	(-MINBUFF)	SUBTRACT THE MINIMUM NEEDED TO SUSTAIN THE I/O ROADBLOCK
000476	740300	6340	SMA:SZ		SKIP IF THE I/O ROADBLOCK NEEDS CLEARING
000477	600503	6350	JMP	KBD8	ELSE CARRY ON
		6360	*		
		6370	*		REMOVE THE TELEPRINTER I/O ROADBLOCK
		6380	*		
000500	200053	6390	LAC	3TEM2	LOAD THE SOFTWARE FLAGS
000501	500657	6400	AND	CB1	CLEAR THE TELEPRINTER I/O ROADBLOCK FLAG
000502	040053	6410	DAC	3TEM2	SAVE THE UPDATED SOFTWARE FLAGS
		6420	*		
		6430	*		NOW PRINT THE CHARACTER IF THERE IS ONE
		6440	*		
000503	100614	6450	KBD8	JMS	FGET OUTPUT ONGOING -- GET THE NEXT CHARACTER
000504	600507	6460	JMP	KBD6	OUTPUT-IN-PROGRESS IS DONE WHEN AN EMPTY BUFFER IS FOUND.
000505	400056	6470	XCT	3TEM5	PRINT THE CHARACTER
000506	600410	6480	JMP	KBD9	EXIT THROUGH THE PI SIMULATION ROUTINE
		6490	*		
		6500	*		THERE IS NO CHARACTER...CLEAR THE OUTPUT FLAGS...LOAD THE LAST
		6510	*		INPUT CHARACTER IF THE KEYBOARD FLAG IS SET.
		6520	*		
000507	200053	6530	KBD6	LAC	3TEM2 LOAD THE TELETYPE SOFTWARE FLAGS
000510	500723	6540	AND	(375777)	CLEAR BITS 0 (OUTPUT-IN-PROGRESS) AND 7 (0=INPUT BUFFER)
000511	040002	6550	DAC	3TEM21	SAVE THE LAST INPUT CHARACTER
000512	040002	6560	DAC	3TEM21	PASS TO THE PACKING ROUTINE
000513	040053	6570	DAC	3TEM2	SAVE THE FLAGS
000514	500652	6580	AND	BIT7	RECOVER THE BUFFER TYPE
000515	741200	6590	SNA		SKIP IF IT WAS AN OUTPUT BUFFER
000516	600410	6600	JMP	KBD9	ELSE EXIT
000517	240053	6610	XOR	3TEM2	MAKE IT AN INPUT BUFFER
000520	040053	6620	DAC	3TEM2	SAVE THE CORRECTED SOFTWARE FLAGS
000521	500647	6630	AND	BIT36	GET THE SOFTWARE KEYBOARD FLAG (BIT 6 IS ALREADY CLEARED)
000522	741200	6640	SNA		SKIP IF IT IS SET
000523	600410	6650	JMP	KBD9	ELSE DONE NOW -- EXIT THROUGH THE SIMULATED PI ROUTINE
000524	200053	6660	LAC	3TEM2	IF SET, LOAD THE INPUT CHARACTER
000525	600407	6670	JMP	KBD2	GO BACK THE CHARACTER AND SIMULATE A USER PROGRAM INTERRUPT

R

## ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```

6680      .EJECT
6690      *
6700      *
6710      *
6720      *   DUE TO THE LACK OF INDEX REGISTER CAPABILITY ON THE PDP-9, IT IS NECESSARY TO
6730      *   COPY A TELETYPE'S PARAMETERS INTO A SET OF TEMPORARY VARIABLES WHENEVER
6740      *   WE GET AN INTERRUPT THAT WILL REQUIRE US TO USE THEM. WHEN WE ARE DONE
6750      *   SERVICING THAT INTERRUPT IT IS NECESSARY TO COPY THEM BACK OUT TO PRESERVE
6760      *   ANY CHANGES WE MAY HAVE MADE.
6770      *
6780      *   ROUTINE TO PRESERVE RE-ENTRANT VARIABLES ON ENTRANCE
6790      *
000526      IO,IN  ENTER      TRANSFER PARAMETERS TO TEMPS ON ENTRANCE
                ,PMC  SAVE,ON
                XX
                ,PMC  RESTORE
000527      040010  6800      DAC      10      AC = PTR TO 1 BEFORE FIRST PARAMETER
000530      040004  6810      DAC      310TM
000531      760047  6820      LAX      3TM20-1
000532      040011  6830      DAC      11
000533      220010  6840      IO,1  LAC      10,X
000534      060011  6850      DAC      11,X      TRANSFER THE PARAMETER
000535      440050  6860      ISZ      3TM20
000536      600533  6870      JMP      .-3      DO THE NEXT ONE
000537      620526  6880      RET      IO,IN,X      DONE, EXIT
                *
                *   ROUTINE TO PRESERVE RE-ENTRANT VARIABLES ON EXIT
                *
000540      6920      IO,OT  ENTER      TRANSFER TEMPS TO PARAMETERS ON EXIT
                ,PMC  SAVE,ON
                XX
                ,PMC  RESTORE
000541      200004  6930      LAC      310TM
000542      040011  6940      DAC      11
000543      220011  6950      LAC      11,X
000544      040050  6960      DAC      3TM20      2'S COMPLEMENT OF PARAMETER COUNT
000545      440050  6970      INX      3TM20      CORRECT THE TRANSFER COUNT
000546      760050  6980      LAX      3TEM0-1
000547      040010  6990      DAC      10
000550      200540  7000      LAC      IO,OT
000551      040526  7010      DAC      IO,IN
000552      600533  7020      JMP      IO,1

```

R

## I/O BUFFER HANDLING ROUTINES

```

7030      ,STIHL I/O BUFFER HANDLING ROUTINES
7040      *
7050      * THESE ROUTINES ARE TO HANDLE EIGHT-BIT ASCII INPUT TO AND OUTPUT FROM
7060      * ANY TELETYPE'S ROTARY INPUT/OUTPUT BUFFER. BUFFER FORMAT IS TWO EIGHT-
7070      * BIT CHARACTERS PER WORD. CHARACTER #1 IN BITS 2-9 AND CHARACTER #2 IN
7080      * BITS 10-17.
7090      * 3TEM3 HOLDS THE (CONSTANT) ADDRESS OF THE START OF THE BUFFER.
7100      * 3TEM4 HOLDS THE (CONSTANT) MAXIMUM+1 ADDRESS OF THE ROTARY BUFFER.
7110      * 3TEM0 IS THE BUFFER INPUT POINTER, BIT 0 = 0 IF THE NEXT CHARACTER
7120      * IS TO BE THE FIRST CHARACTER STORED IN THAT WORD. BIT 0 = 1 IF THE NEXT
7130      * CHARACTER IS TO BE THE SECOND CHARACTER STORED IN THAT WORD.
7140      * 3TEM1 IS THE BUFFER OUTPUT POINTER, BIT 0 = 0 IF THE NEXT CHARACTER
7150      * IS TO BE THE FIRST CHARACTER SUPPLIED FROM THAT WORD, BIT 0 = 1 IF THE
7160      * CHARACTER IS TO BE THE SECOND CHARACTER SUPPLIED FROM THAT WORD.
7170      *
7180      * NEWBR IS A SUBROUTINE TO RE-INITIALIZE THE BUFFER AND TO ZERO ALL SOFTWARE TELETYPE FLAGS.
7190      *
7200      *
000553 7210 NEWBR ENTER ROUTINE TO CLEAR THE I/O BUFFER
          ,PMC SAVE,ON
          XX
          ,PMC RESTORE
000554 200054 7220 LAC 3TEM3 LOAD THE START OF THE BUFFER
000555 040051 7230 DAC 3TEM0 RESET THE IN-POINTER
000556 040052 7240 DAC 3TEM1 RESET THE OUT-POINTER
000557 140053 7250 DZM 3TEM2 KILL ALL SOFTWARE TELETYPE FLAGS
000560 620553 7260 RET NEWBR,X
          *
          * PUTIN PACKS THE EIGHT-BIT ASCII CHARACTER FROM 3TM21 INTO THE PROPER ROTARY LINE BUFFER.
          * IT RETURNS +2 IF SUCCESSFUL, +1 ON OVERFLOW.
          *
000561 7310 PUTIN ENTER
          ,PMC SAVE,ON
          XX
          ,PMC RESTORE
000562 200002 7320 LAC 3TM21 LOAD THE ALLEGED CHARACTER
000563 500655 7330 AND BL8 MASK TO JUST ASCII TO PROTECT OURSELVES
000564 040002 7340 DAC 3TM21 RESTORE THE CHARACTER
000565 200051 7350 LAC 3TEM0 LOAD THE ACTIVE ADDRESS
000566 100635 7360 JMS NXPTR ADVANCE THE POINTER TO THE NEXT CHARACTER LOCATION
000567 540052 7370 SAD 3TEM1 SKIP IF NO OVERFLOW
000570 620561 7380 RET PUTIN,X RETURN +1 FOR OVERFLOW
000571 440561 7390 INX PUTIN RETURN +2 FOR SUCCESSFUL
000572 040051 7400 DAC 3TEM0 SAVE THE UPDATED POINTER
000573 740100 7410 SMA SKIP IF IT IS THE FIRST CHARACTER IN THIS WORD
000574 600610 7420 JMP PUT2
          *
          *
000575 7440 PUT1 ... PUT FIRST CHARACTER IN BITS 2-9
000575 220003 7450 LAC 3TM22,X LOAD THE BUFFER WORD
000576 500655 7460 AND BL8 CLEAR ROOM FOR THIS CHARACTER
000577 060003 7470 DAC 3TM22,X
000600 200002 7480 LAC 3TM21 LOAD THE CHARACTER -- BIT 0 MUST BE ZERO

```

R		I/O BUFFER HANDLING ROUTINES	
000601	742010 7490	RTL	POSITION THE CHARACTER
000602	742010 7500	RTL	
000603	742010 7510	RTL	
000604	742010 7520	RTL	
000605	260003 7530	XOR 3TM22,X	PACK THE CHARACTER
000606	060003 7540	DAC 3TM22,X	STORE THE CHARACTER
000607	620561 7550	RET PUTIN,X	
	7560		
000610	7570	PUT2	PUT SECOND CHARACTER INTO BITS 11-17
000610	220003 7580	LAC 3TM22,X	LOAD THE STORAGE WORD
000611	500662 7590	AND CBL8	CLEAR ROOM FOR THIS CHARACTER
000612	240002 7600	XOR 3TM21	AND INSERT THE NEW CHARACTER
000613	600606 7610	JMP PUT4	EXIT
	7620	*	
	7630	*	FGET REMOVES THE FIRST REMAINING CHARACTER FROM THE I/O BUFFER.
	7640	*	IT RETURNS IT AS 7-BIT ASCII IN THE AC.
	7650		
000614	7660	FGET	ENTER
		,PMC	SAVE,ON
000614	740040	XX	
		,PMC	RESTORE
000615	200052 7670	LAC 3TEM1	LOAD THE ACTIVE ADDRESS
000616	540051 7680	SAD 3TEM0	
000617	620614 7690	RET FGET,X	RETURN +1 FOR NO CHARACTER
000620	440614 7700	INX FGET	RETURN +2 FOR SUCCESSFUL
000621	100635 7710	JMS NXPTR	ADVANCE THE POINTER TO THE NEXT CHARACTER LOCATION
000622	040052 7720	DAC 3TEM1	AND SAVE THE NEW POINTER
000623	740010 7730	RAL	
000624	220003 7740	LAC 3TM22,X	LOAD THE CHARACTER
000625	744400 7750	SNL;CLL	
000626	600633 7760	JMP FGET2	
000627	742020 7770	RTR	
000630	742020 7780	RTR	
000631	742020 7790	RTR	
000632	742020 7800	RTR	
000633	500655 7810	FGET2 AND BL8	MASK TO EIGHT-BIT ASCII
000634	620614 7820	RET FGET,X	
	7830	*	
	7840	*	NXPTR INCREMENTS THE POINTER PASSED IN THE AC, AND RETURNS THE RESULT IN
	7850	*	THE AC. THE LINK WILL ALWAYS BE FLIPPED, THE ADDRESS INCREMENTED OR WRAPPED
	7860	*	ONLY AS APPROPRIATE.
	7870	*	
000635	7880	NXPTR	ENTER
		,PMC	SAVE,ON
000635	740040	XX	
		,PMC	RESTORE
000636	040003 7890	DAC 3TM22	SAVE THE OLD POINTER -- THE ROUTINES STILL NEED IT
000637	240646 7900	XOR BIT0	FLIP BIT 0
000640	741100 7910	SPA	
000641	620635 7920	RET NXPTR,X	NO NEED TO INCREMENT THE ADDRESS
000642	340653 7930	TAD BIT17	ELSE ADVANCE THE POINTER
000643	540055 7940	SAD 3TEM4	END OF THE BUFFER??

R

I/O BUFFER HANDLING ROUTINES

000644	200054	7950	LAC	3TEM3	YES, SO WRAP THE POINTER
000645	620635	7960	RET	NXPTR,X	EXIT
		7970	*		
		7980	*		
		7990	*		
				RESIDENT CONSTANTS ARE USED INSTEAD OF LITERALS TO PERMIT THEM TO BE ACCESSED BY OVERLAYS.	
000646	400000	8000	BIT0	400000	
000647	044000	8010	BIT36	044000	
000650	010000	8020	BIT5	010000	
000651	004000	8030	BIT6	004000	
000652	002000	8040	BIT7	002000	
000653	000001	8050	BIT17	000001	
000654	000177	8060	BL7	177	LAST SEVEN BITS
000655	000377	8070	BL8	000377	LAST EIGHT BITS
000656	377777	8080	CB0	377777	MASK TO CLEAR BIT 0
000657	577777	8090	CB1	577777	MASK TO CLEAR BIT 1
000660	767777	8100	CB5	767777	MASK TO CLEAR BIT 5
000661	775777	8110	CB7	775777	
000662	777400	8120	CB18	777400	
000663	017777	8130	ADRS5	17777	MASK TO CLEAR LAST EIGHT BITS
000664	600000	8140	JMP	JMP	MASK TO RETAIN JUST THE ADDRESS BITS
000665	703304	8150	DBK	DBK	CONSTANT

R

## RESIDENT DISK ROUTINES

```

      8160      ,STITLE RESIDENT DISK ROUTINES
      8170      *
      8180      * THE RESIDENT DISK ROUTINE IS STRICTLY A MINIMUM SIZE PHYSICAL
      8190      * DISK HANDLER.
      8200      *
      8210      * CALLING SEQUENCE: LAW PNTR
      8220      * JMS DO
      8230      *
      8240      * PNTR+1: PHYSICAL DISK ADDRESS ( = BLOCK NUMBER * 400 )
      8250      * PNTR+2: CORE ADDRESS -1
      8260      * PNTR+3: TWO'S COMPLEMENT WORD COUNT
      8270      * PNTR+4: DISK READ (3) OR DISK WRITE (5)
      8280      *
      000666 8290 DO ENTER
      000666 740040 ,PMC SAVE,ON
      000667 040010 8300 XX
      000670 707074 8310 ,PMC RESTORE
      000671 707212 8320 DAC 10 SET THE PARAMETER POINTER
      000672 751101 8330 DLAW+10 BE SURE WE ARE SET FOR DISK ZERO
      000673 600671 8340 DLK+10 FIND OUT WHETHER OR NOT THE DISK IS IN USE
      000674 040266 8350 SPA,CLA,CMA SKIP IF IT IS FREE
      000675 220010 8360 JMP -2 ELSE WAIT FOR IT
      000676 707024 8370 DQ2 DAC DKLOK FLAG THE SYSTEM IS USING THE DISK
      000677 220010 8380 DQ3 LAC 10,X GET THE ADDRESS
      000700 040037 8390 DLAL PLACE IT INTO THE ADDRESS REGISTER
      000701 220010 8400 LAC 10,X LOAD THE STARTING ADDRESS
      000702 040036 8410 DAC SDKCA SET THE CORE ADDRESS POINTER
      000703 220010 8420 LAC 10,X SET THE DISK WORD COUNT
      000704 707047 8430 DKOVR DSCFIDSFIDSCN ISSUE THE READ COMMAND
      000705 707001 8440 DSSF WAIT FOR THE OPERATION TO COMPLETE
      000706 600705 8450 JMP -1
      8460      *
      8470      * CHECK THE OPERATION AND RETURN TO THE APPROPRIATE PLACE
      8480      *
      000707 707272 8490 DKDON DSRS+10 CLEAR THE AC AND GET THE STATUS OF THE OPERATION
      000710 707242 8500 DSCD CLEAR THE FLAGS
      000711 741100 8510 SPA SEE IF OK
      000712 740040 8520 HLT
      000713 620666 8530 RET DO,X
      8540      *
      8550      *
      8560      * OVERLAY COMMON CONTROL AND COMMUNICATION WORDS
      8570      *
      000714 000000 8580 DC0 ,DSA
      000715 000000 8590 OC1 ,DSA
      000716 000000 8600 OC2 ,DSA
      000717 000000 8610 OC3 ,DSA
      000720 633400 8620 ,LIT COLLECT LITERALS TEMPORARILY BEFORE USER TABLE
      000721 000016
      000722 777770

```



RES--B01 05/31/72 01104108 PDP-9 MINI TIME-SHARING SYSTEM RESIDENT EXECUTIVE PROGRAM

PAGE 24

R

RESIDENT DISK ROUTINES

000723 375777

R

FORMAT OF THE USER JOB TABLE

		8630		.STITLE	FORMAT OF THE USER JOB TABLE
001700		8640		.LOC	\$JTSTRT
001700	000000	8650	FRDA	.DSA	DEVICE ADDRESS OF THE USER'S CURRENTLY OPEN FILE
001701	000000	8660	FRCA	.DSA	CORE ADDRESS OF THE USER'S CURRENTLY OPEN FILE
001702	000000	8670	FRLEN	.DSA	LENGTH OF THE USER'S CURRENTLY OPEN FILE
001703	000000	8680	FRSTA	.DSA	START ADDRESS OF THE USER'S CURRENTLY OPEN FILE
		8690			
001704	000000	8700	UTEM0	.DSA	USED FOR PASSING PARAMETERS TO OR BETWEEN EXECUTIVE PROGRAMS
001705	000000	8710	UTEM1	.DSA	
001706	000000	8720	UTEM2	.DSA	
001707	000000	8730	UTEM3	.DSA	
001710	000000	8740	UTEM4	.DSA	USED FOR PASSING PARAMETERS TO OR BETWEEN USER OR PHANTOM PROGRAMS
001711	000000	8750	UTEM5	.DSA	
001712	000000	8760	UTEM6	.DSA	
		8770			
001713		8780	.0	.BLOCK	40 USER IMAGE OF FIRST 40 LOCATIONS
		8790			
001753	000000	8800	AC	.DSA	USER AC SAVED
001754	000000	8810	HQ	.DSA	USER HQ SAVED
001755	000000	8820	SC	.DSA	USER STEP COUNTER SAVED
001756	000000	8830	ACS	.DSA	SOFTWARE ACCUMULATOR SWITCHES REGISTER
		8840			
001757	000000	8850	CLOCK	.DSA	STORE USER'S CLOCK HERE
001760	000000	8860	IORS	.DSA	STORE USER'S PI STATUS
001761	000000	8870	DFLAG	.DSA	SOFTWARE IMAGE OF THE DISK FLAG
001762	000000	8880	DAPO	.DSA	SOFTWARE IMAGE OF DISK REGISTER APO
001763	000000	8890	DAP1	.DSA	SOFTWARE IMAGE OF DISK REGISTER AP1
001764	000000	8900	DFN	.DSA	SOFTWARE IMAGE OF DISK FUNCTION REGISTER
001765	000000	8910	DSTAT	.DSA	SOFTWARE IMAGE OF DISK STATUS REGISTER
		8920			
001766	000000	8930	UCORE	.DSA	PHYSICAL DEVICE ADDRESS OF THE USER PROGRAM CORE IMAGE
001767	000000	8940	UDISK	.DSA	PHYSICAL DEVICE ADDRESS OF THE PHYSICAL USER PHYSICAL DEVICE, FILE
001770	000000	8950	VALID	.DSA	STORE THE USER'S VALIDATION BITS HERE
001771	000000	8960	NUMBR	.DSA	TELETYPE NUMBER == POINTER TO RESIDENT TELETYPE PARAMETERS (US0, US1 OR US2)
		8970			
001772	000000	8980	NAME	.DSA	NUMBER+TYPE
001773	000000	8990	OVER	.DSA	NAME OF USER'S CURRENT OVERLAY PROGRAM
001774	000000	9000	TYPE	.DSA	USER PROGRAM = 0; PHANTOM PROGRAM = -1
001775	000000	9010	PURNM	.DSA	NAME FOR PURE CODE PORTION OF USER'S CURRENT PHANTOM PROGRAM (= 0 IF NONE)
001776	000000	9020	RSTRT	.DSA	RESTART PC (CONTROLS ADDRESS, LINK, MEMORY PROTECT)
		9030			
001777		9040		.END	1

TRANSFER ADDRESS 600001

R

### CROSS REFERENCE TABLE

Address	Label	Value	Value	Value	Value	Value	Value	Value	Value
1713	.0	4510							
26	.310	3400							
27	.311	3410							
4464	.DT	570							
6460	.TP	550							
2023	10SAVE	1870	1880						
2024	11SAVE	1880	1920						
5	3AC	3370							
105	3REST	3870	3880						
51	3TEM0	3530	3540						
52	3TEM1	3540	3550						
53	3TEM2	3550	3560						
54	3TEM3	3560	3570						
55	3TEM4	3570	3580						
56	3TEM5	3580	3590						
57	3TEM6	3590	3600						
50	3TM20	3520	3530						
2	3TM21	3350							
3	3TM22	3360							
14000	7K	1030							
16000	8K	1020	910	1010	2640	2650			
1753	AC	4520	4530						
1756	ACS	4550	4560						
2015	ACSAVE	1810	1820						
2022	ACSW	1860	1870						
651	ADRSS	4100	4110						
300	AT	2950							
300	ATSGN	2900							
422030	BAS	420							
2151	BCNTRL	2330	2340						
634	BIT0	3970	3980						
641	BIT17	4020	4030						
635	BIT36	3980	3990						
636	BIT5	3990	4000						
637	BIT6	4000	4010						
640	BIT7	4010	4020						
642	BL7	4030	4040						
643	BL8	4040	4050						
2000	BOUNDA	970	960	980	990	1000	1630	5040	5080
377	BRK	5550							
2170	BUFFER	2490	2550						
1000	BUFLN	2500	2550						
644	CB0	4050	4060						
645	CB1	4060	4070						
646	CB5	4070	4080						
647	CB7	4080	4090						
650	CLB	4090	4100						
6	CHRMX	3180	3200						
2	CHRPX	3130	3200						
516	CLKLOK	3960	3290	3970					
50	CLKMAX	2840	3180	4000					
60	CLKSPD	3160	3170						

## R

## CROSS REFERENCE TABLE

1757	CLOCK	4560	4570		
45	CMP1	3490	3500		
46	CMP2	3500	3510		
6	CNTRL	3380	3390		
2053	COMFLG	2200	2210		
2150	COMST0	2270	2280		
16000	CORMAX	910	980		
47	CSPL	3510	3520		
44	CSWP	3480	3490		
60	CTBFR	3600	3630	3640	
100	CTBIN	3640	3650	3670	4250
2000	CTEMP0	1630			
2001	CTEMP1	1640			
2002	CTEMP2	1650			
2003	CTEMP3	1660			
2004	CTEMP4	1670			
2005	CTEMP5	1680			
2006	CTEMP6	1690			
2007	CTEMP7	1700			
2010	CTEMP8	1710			
2011	CTEMP9	1720			
102	CTFLG	3650	3660		
104	CTNAM	3660			
2043	D PC	2120	2130		
2154	D BCA	2370	2380		
2153	D BDA	2360	2370		
2163	D FDA	2440	2450		
2042	D LOC	2110	2120		
2022	D AGSW	1860			
2156	D BALY	2390	2400		
2155	D BLEN	2380	2390		
2161	D BMAX	2420	2430		
2157	D BMIN	2400	2410		
2162	D BPTR	2430	2440		
2167	D FMAX	2480	2490		
2165	D FMIN	2460	2470		
2046	D MASK	2150	2160		
2164	D MFDA	2450	2460		
2036	DADRSW	2070	2080		
1762	DAP0	4590	4600		
1763	DAP1	4600	4610		
653	DBK	4120	4130		
24	DBKNUM	2220	2270		
2054	DBKTAB	2210	2270		
2035	DBSTOR	2050	2060		
422027	DDT	410			
12000	DDTST	5000			
2037	DDUMSW	2080	2090		
1761	DFLAG	4580	4590		
1764	DFN	4610	4620		
2151	DFTYPE	2340	2350		
2045	DWICOR	2140	2150		

R

## CROSS REFERENCE TABLE

2050	DINDIR	2170	2180
100	DK0	4270	
127	DK1	4310	
156	DK2	4350	
37	DKCA	2750	8390
675	DKDON	4170	4180
16000	DKLEN	2650	2660
34	DKLENB	2660	
266	DKLOK	3830	3840
672	DKOVR	4160	4170
2	DKRD	2760	1360
36	DKWC	2740	8410
4	DKWRT	2770	
2041	DLIMIT	2100	2110
2044	DLOCOR	2130	2140
2160	DMBMIN	2410	2420
2166	DMFMIN	2470	2480
654	DO	4130	4140
662	DQ2	4140	4150
663	DQ3	4150	4160
2152	DOFTYP	2350	2360
2032	DPACSW	1980	
2040	DPATSW	2090	2100
2051	DPCHSK	2180	2190
2052	DREGBR	2190	2200
2035	DREGSW	2060	2070
2047	DRELOC	2160	2170
1765	DSTAT	4620	4630
446400	DT.	560	
2000	DTEMP0	1630	
2001	DTEMP1	1640	
2002	DTEMP2	1650	
2003	DTEMP3	1660	
2004	DTEMP4	1670	
2005	DTEMP5	1680	
2006	DTEMP6	1690	
2007	DTEMP7	1700	
2010	DTEMP8	1710	
2011	DTEMP9	1720	
275	EQUAL	2910	
382	ERRCAL	4230	940
602	FGET	3950	3960
1701	FRCA	4410	4420
1700	FRDA	4400	4410
1702	FRLN	4420	4430
1703	FRSTA	4430	4440
2	FUDGE	3190	3200
276	GREAT	2930	
1700	IMPLEN	990	
3170	IMPSTR	2550	
422020	INT	320	
513	IO.IN	3910	3920

### CROSS REFERENCE TABLE

525	IO.DT	3920	3930																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
-----	-------	------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

R

## CROSS REFERENCE TABLE

574646	OFF	2730					
575600	ON	2720					
1773	OVER	4680	4690				
700	OVLEN	940	1350				
1000	OVSTRT	930	920	940	4750	4880	4960 1340
2033	P10SAV	1990	2000				
2034	P11SAV	2000	2050				
2025	PACSAV	1930	1940				
2032	PACSW	1980	1990				
241	PBFLAG	3810	3820				
2017	PCSAVE	1830	1840				
227	PFLAG	3770	3780				
77	PH0	4260	4270				
126	PH1	4300	4310				
155	PH2	4340	4350				
1	PHANTO	2780					
2150	PHFLAG	2280	2330				
1700	PHLEN	2640					
2025	PHSTOR	1920	1930				
274	PIDN2	3850	3860				
270	PIDON	3840	3850				
1001	PINT	4890	4900	5620			
303	PIOUT	3860	3870				
602026	PLDR	400					
2026	PMQSAV	1940	1950				
602025	PMTR	380					
2027	PPCSAV	1950	1960				
606064	PPT	520					
2031	PSCSAV	1970	1980				
2030	PSTSAV	1960	1970				
606460	PYP	510					
606462	PTR	500					
12100	PURLEN	1010					
1775	PURNM	4700	4710				
3700	PURSTR	2560	990	1010	2560		
546	PUTIN	3940	3950				
1713	R .0	8780	3780				
1753	R AC	8800					
666	R DQ	8290	4150	4190	8530		
1754	R MQ	8810					
1755	R SC	8820					
5	R 3AC	760	2240	3800			
1756	R ACS	8830					
654	R BL7	8060	4920				
655	R BL8	8070	7330	7460	7810		
656	R CB0	8080	6170				
657	R CB1	8090	6400				
660	R CB5	8100	5530				
661	R CB7	8110					
317	R CLK	3970	2650				
665	R DBK	8150	2520				
1764	R DFN	8900					

## R

## CROSS REFERENCE TABLE

674	R	D02	8350						
675	R	D03	8360						
664	R	JMP	8140	3570					
714	R	OC0	8580						
715	R	OC1	8590						
716	R	OC2	8600						
717	R	OC3	8610						
26	R	.310	980	2260	3480				
27	R	.311	990	2280	3500				
646	R	BIT0	8000	5370	7900				
650	R	BIT5	8020	5540					
651	R	BIT6	8030	5350	6000				
652	R	BIT7	8040	5150	6130	6580			
662	R	CBL8	8120	7590					
45	R	CMP1	1390						
46	R	CMP2	1400						
47	R	CSPL	1410						
44	R	CSWP	1380	4160					
1762	R	DAP0	8880	3230					
1763	R	DAP1	8890						
614	R	FGET	7660	6450	7690	7700	7820		
246	R	FLAG	3070	2860	2910	2960	3080	3090	3100
1701	R	FRCA	8660						
1700	R	FRDA	8650						
533	R	IO.1	6840	7020					
1760	R	IORS	8860						
425	R	KBD0	5680	4940					
407	R	KBD2	5380	6670					
417	R	KBD4	5560	5270					
413	R	KBD5	5520	5610					
507	R	KBD6	6530	6460					
421	R	KBD7	5590	5510					
503	R	KBD8	6450	6350					
410	R	KBD9	5490	5950	6480	6600	6650		
437	R	MTR1	5780						
1772	R	NAME	8980	4090					
1773	R	OVER	8990						
304	R	PIGO	3630	3580					
575	R	PUT1	7440						
610	R	PUT2	7570	7420					
606	R	PUT4	7540	7610					
34	R	RACS	1170						
32	R	RDT0	1150						
33	R	RDT1	1160						
230	R	RPTP	2880						
235	R	RPTR	2930						
242	R	RSCQ	2980						
335	R	SWAP	4150	4260	5790				
40	R	SWPS	1330	4170	4180				
1774	R	TYPE	9000	4080					
4	R	R310TM	750	6810	6930				
305	R	R3REST	3730	3590	3770	3810			



## R

### CROSS REFERENCE TABLE

51	R3TEM0	1470	6280	7230	7350	7400	7680							
52	R3TEM1	1480	6240	7240	7370	7670	7720							
53	R3TEM2	1490	5070	5110	5250	5390	5520	5550	5930	6030	6060	6120	6160	6390
			6410	6530	6570	6610	6620	6660	7250					
54	R3TEM3	1500	7220	7950										
55	R3TEM4	1510	4070	5490	5700	5710	7940							
56	R3TEM5	1520	5190	6470										
57	R3TEM6	1530	5360	6040										
50	R3TM20	1460	6820	6860	6960	6970								
2	R3TM21	730	4790	4820	5100	6270	6300	6550	6560	7320	7340	7480	7600	
3	R3TM22	740	7450	7470	7530	7540	7580	7740	7890					
34	RACS	3440												
663	RADDRS	8130	3560	6250	6290									
653	RBIT17	8050	7930											
647	RBIT36	8010	5090	6630										
327	RCLKST	4060	3320											
1757	RCLQCK	8850												
6	RCNT	3390												
6	RCNTRL	770												
35	RCORE	3450												
60	RCTBFR	1760	1790	1800	1910									
100	RCTBIN	1790	1920	4530										
102	RCTFLG	1810												
346	RCTKBD	4500	2710											
104	RCTNAM	1920												
347	RCTOUT	4520	2680											
1003	RDBLK	4910	4920	4060										
1761	RDFLAG	8870	3210											
707	RDKDON	8490												
266	RDKLOK	3340	3180	4020	5750	8350								
704	RDKOVR	8430												
253	RDKSVC	3180	3010											
1765	RDSTAT	8910	3200											
32	RDT0	3420												
33	RDT1	3430												
422021	RES	330												
40	RESCAT	3470	3480											
1000	RESLEN	920												
633	RPGT2	7810	7760											
234	RFLAG	3790	3800											
1702	RFRLEN	8670												
1703	RFRSTA	8680												
526	RIO,IN	6790	4810	5920	6880	7010								
540	RIO,OT	6920	5560	5690	7000									
454	RKBD01	6110	6020											
401	RKBD05	5240	5170											
416	RKBD41	5550	6050	6180										
357	RKBDIN	4750	4510	4590	4660	4800								
463	RKBD02	6230	6150											
441	RKBDOT	5910	4540	4610	4680									
76	RL0LOK	1770												
107	RL1BFR	1960	1990	2000	2020									

## R

## CROSS REFERENCE TABLE

127	RL1BIN	1990	2030	4600					
131	RL1FLG	2010							
351	RL1KBD	4580	2740						
125	RL1LOK	1970							
133	RL1NAM	2030							
352	RL1OUT	4600	2800						
136	RL2BFR	2070	2100	2110	2130				
156	RL2BIN	2100	2140	4670					
160	RL2FLG	2120							
354	RL2KBD	4650	2770						
154	RL2LOK	2080							
162	RL2NAM	2140							
355	RL2OUT	4670	2830						
553	RNEWBR	7210	5680	7260					
1771	RNUMBR	8960	5500	5720					
635	RNXPTR	7880	7360	7710	7920	7960			
241	RPBFLG	2970							
227	RPFLAG	2870							
234	RPIDN2	3550							
270	RPIDON	3480	3310	4040	4100	5570	5740	5770	
303	RPIDOUT	3620	2530						
267	RPISV2	3390	3000						
165	RPISVC	2230	680						
411	RPITST	5500	3110						
230	RPTP	3780	3790						
235	RPTR	3800	3810						
1775	RPURNM	9010							
561	RPUTIN	7310	5180	7380	7390	7550			
35	RRCORE	1270	5590						
234	RRFLAG	2920							
1776	RRSTRY	9020							
242	RSCQ	3820	3830						
1776	RSTRY	4710							
336	RSWAP1	4160							
340	RSWAP3	4180							
1766	RUCORE	8930							
1767	RUDISK	8940							
1704	RUTEM0	8700							
1705	RUTEM1	8710							
1706	RUTEM2	8720	4240						
1707	RUTEM3	8730							
1710	RUTEM4	8740							
1711	RUTEM5	8750							
1712	RUTEM6	8760							
1770	RVALID	8950							
1755	SC	4540	4550						
640000	SCRSTR	2670							
2021	SCSAVE	1850	1860						
243	SHARP	2890							
377	SPCOD	5410							
422122	SPL	430							
1090	SPLST	4960							

R

## CROSS REFERENCE TABLE

777400	SPMSK	5390	
2020	STSAVE	1840	1850
335	SWAP	3880	3890
336	SWAP1	3890	3900
340	SWAP3	3900	3910
1000	SWCAT	4750	4760
1003	SWCLK	4780	4790 4110
1004	SWERR	4790	4800 4250
1007	SWMP1	4820	4830
1010	SWMP2	4830	4840
1002	SWMTR	4770	4780 5780
1011	SWOPR	4840	
422022	SWP	340	
1001	SWPPR	4760	4770
40	SWPS	3460	3470
1005	SWSP1	4800	4810
1006	SWSP1	4810	4820
1300	SYSBAS	2800	2810
41300	SYSDA	2810	
1777	SYSMAX	2820	
100	TABLEN	2630	2640
2000	TEMP0	1630	1640
2001	TEMP1	1640	1650
2012	TEMP10	1730	1740
2013	TEMP11	1740	1750
2014	TEMP12	1750	1800
2002	TEMP2	1650	1660
2003	TEMP3	1660	1670
2004	TEMP4	1670	1680
2005	TEMP5	1680	1690
2006	TEMP6	1690	1700
2007	TEMP7	1700	1710
2010	TEMP8	1710	1720
2011	TEMP9	1720	1730
646000	TR	540	
376	TRCOFF	5540	
375	TRCON	5530	
2000	TTEMP0	1630	
2001	TTEMP1	1640	
2002	TTEMP2	1650	
2003	TTEMP3	1660	
2004	TTEMP4	1670	
2005	TTEMP5	1680	
2006	TTEMP6	1690	
2007	TTEMP7	1700	
2010	TTEMP8	1710	
2011	TTEMP9	1720	
6	TTYCLK	3170	3180
3	TTYNUM	3140	
10	TTYSPD	3150	3170
1774	TYPE	4690	4700
1766	UCORE	4630	4640

## R

## CROSS REFERENCE TABLE

1767	UDISK	4640	4650	
336	UPARR	2940		
76	US0	4250	4260	4280
125	US1	4290	4300	4320
154	US2	4330	4340	4360
0	USER	2790		
3	USERS	2850	3200	
14000	USLEN	980	2640	
2015	USTORE	1800	1810	
75	UT0	4280		
124	UT1	4320		
153	UT2	4360		
1704	UTEM0	4440	4450	
1705	UTEM1	4450	4460	
1706	UTEM2	4460	4470	
1707	UTEM3	4470	4480	
1710	UTEM4	4480	4490	
1711	UTEM5	4490	4500	
1712	UTEM6	4500	4510	
1770	VALID	4650	4660	

**R**

## UNDEFINED SYMBOLS

[illegible]

**R**

## MACRO CROSS REFERENCE TABLE

[illegible]