

00010

XLIST

01970

00030	
00040	EXTERNAL CLOSE1,DEVLST,ERROR,JBADR
00050	EXTERNAL JOB,JOBFF,RELA1,USRJDA,WAIT1,CLDS,CLEN
00060	EXTERNAL PJOBN,CPOPJ1,PIOMOD,CPOPJ,CPOPJ2
00070	EXTERNAL SCNOFF,SCNON

00080					
00090					,DEC 06 00 EX COM L PT PRE 01 ADVBFE
00100					,H. R. MORSE 64-12-26
00110					,CALLING SEQUENCE
00120					, PUSHJ PDP,ADVBFE
00130					, EXIT1 RETURN IF NEXT BUFFER IS EMPTY
00140					, EXIT2 RETURN IF NEXT BUFFER IS FULL
00150					,CLEARS THE USE BIT (IOUSE:=0) OF THE BUFFER POINTED TO BY THE
00160					,OUTPUT BUFFER ADDRESS (DEVOAD) OF THE CURRENT DEVICE DATA BLOCK
00170					,AND ADVANCES THE BUFFER ADDRESS TO THE NEXT BUFFER IN THE RING.
00180					,UPON RETURN, SKIPS IF THE NEXT BUFFER IS FULL.
00190					,MONITOR INTERFACE
00200					, STORAGE: 6
00210					, SYMBOLS SET/USED:
00220					, ACCUMULATORS: DEVDAT U
00230					, PDP S/U
00240					, TAC S/U
00250					, DEVICE DATA BLOCK: DEVOAD S/U
00260					, JOB BUFFER HEADER: IOUSE S/U
00270					INTERN ADVBFE
00280					ADVBFE: ADRCHK DEVOAD(DEVDAT)
000000	260140	000000			PUSHJ PDP,ADRCK
000001	550046	000010			HRRZ TAC,DEVOAD(DEVDAT)
000002	515040	400000	00290		HRLZI TAC,IOUSE; IOUSE:=0
000003	413066	000010	00300		ANDCAB TAC,@DEVOAD(DEVDAT)
000004	542046	000010	00310		HRRM TAC,DEVOAD(DEVDAT); DEVOAD 18-35:=NEXT BUFFER ADDRESS
000005	335026	000010	00320		SKIPGE @DEVOAD(DEVDAT); IS IOUSE=0?
000006	350003	000000	00330		AOS (PDP); EXIT2. BUFFER IS FULL
000007	263140	000000	00340		POPJ PDP,; EXIT1. BUFFER IS EMPTY

```

00350
00360 ,DEC 06 00 EX COM L PT PRE 01 ADVBFF
00370 ,H. R. MORSE 64-12-26
00380 ,CALLING SEQUENCE
00390 , PUSHJ PDP,ADVBFF
00400 , EXIT1 RETURN IF NEXT BUFFER IS FULL
00410 , EXIT2 RETURN IF NEXT BUFFER IS EMPTY
00420 ,SETS THE USE BIT (IOUSE:=1) OF THE BUFFER POINTED TO BY THE
00430 ,INPUT BUFFER ADDRESS (DEVIAD) OF THE CURRENT DEVICE DATA BLOCK
00440 ,AND ADVANCES THE BUFFER ADDRESS TO THE NEXT BUFFER IN THE RING.
00450 ,UPON RETURN, SKIPS IF THE NEXT BUFFER IS EMPTY.
00460 ,MONITOR INTERFACE
00470 , STORAGE: 6
00480 , SYMBOLS SET/USED:
00490 , ACCUMULATORS: DEVDAT U
00500 , PDP S/U
00510 , TAC S/U
00520 , DEVICE DATA BLOCK: DEVIAD S/U
00530 , JOB BUFFER HEADER: IOUSE S/U
00540 , INTERN ADVBFF
00550 ADVBFF: ADRCHK DEVIAD(DEVDAT)
000010 260140 000000' PUSHJ PDP,ADRCK
000011 550046 000007 HRRZ TAC,DEVIAD(DEVDAT)
000012 515040 400000 00560 HRLZI TAC,IOUSE; IOUSE:=1
000013 437066 000007 00570 IORB TAC,@DEVIAD(DEVDAT)
000014 542046 000007 00580 HRRM TAC,DEVIAD(DEVDAT); DEVIAD:=NEXT BUFFER ADDRESS
000015 331026 000007 00590 SKIPL @DEVIAD(DEVDAT); IOUSE=1?
000016 350003 000000 00600 AOS (PDP); EXIT2. NEXT BUFFER IS EMPTY
000017 263140 000000 00610 POPJ PDP,; EXIT1. NEXT BUFFER IS FULL

```

```
00620  
00630 ,DEC 06 00 EX COM L PT PRE 01 ASCIA  
00640 ,C. FRAZIER AND W. SEGAL 64-12-26  
00650 ,CALLING SEQUENCE  
00660 , PUSHJ PDP,ASCIA  
00670 , EXIT ALWAYS RETURNS HERE  
00680 ,ADDS THE ASCII INCREMENT (RIGHT JUSTIFIED) IN AC TAC1 TO THE FIVE  
00690 ,DIGIT ASCII NUMBER (LEFT JUSTIFIED) IN AC TAC AND LEAVES THE  
00700 ,RESULT (LEFT JUSTIFIED) IN AC TAC. TAC1 IS RESTORED.  
00710 ,MONITOR INTERFACE  
00720 , STORAGE: 21  
00730 , SYMBOLS SET/USED:  
00740 , ACCUMULATORS: PDP U  
00750 , TAC S/U  
00760 , TAC1 S/U
```

```
00770  
00780 ; INTERN ASCIA  
00790 ;ASCIA: PUSH PDP,TAC1; SAVE TAC1 = ASCII INCREMENT (RIGHT JUST  
00800 ; AND TAC1,K2; CLEAR MOST SIGNIFICANT 3 BITS OF  
00810 ;, EACH CHARACTER  
00820 ; LSH TAC,-1; TAC=FIVE DIGIT ASCII NO.(RIGHT JUST.)  
00830 ; IOR TAC,K4; SET SECOND AND THIRD MOST SIGNIFICANT  
00840 ;, BITS OF EACH CHARACTER TO ONES.  
00850 ; TLZN TAC,400000  
00860 ; ADD TAC,K1  
00870 ; ADD TAC1,TAC  
00880 ; AND TAC1,K3  
00890 ; MOVE TAC,K4  
00900 ; AND TAC,TAC1  
00910 ; ASH TAC,-3  
00920 ; SUBM TAC1,TAC  
00930 ; IOR TAC,K4  
00940 ; LSH TAC,1  
00950 ; TRO TAC,1  
00960 ; POP PDP,TAC1; RESTORE TAC1  
00970 ; POPJ PDP,; RETURN  
00980 ;K1: OCT 215064321506; CHARACTER MASK: 1000110  
00990 ;K2: OCT 036170743617; CHARACTER MASK: 0001111  
01000 ;K3: OCT 176773757677; CHARACTER MASK: 0111111  
01010 ;K4: OCT 140603014060; CHARACTER MASK: 0110000
```

```

01020
01030          INTERNAL ASSIN,ASSASG
01040
000020  260140 000041' 01050  ASSIN:  PUSHJ PDP,DEVSRG      ;SEARCH FOR DEVICE
000021  263140 000000 01060          POPJ PDP,          ;DEVICE NOT FOUND
000022  201100 200000 01070          MOVEI TAC1, ASSPRG  ;DEVICE FOUND, FLAG AS ASSIGNED BY PROG
01080
01090          ;ASSIGN DEVICE IF UNASSIGNED
01100          ;CALL:  MOVE ITEM, JOB NUMBER
01110          ;
01120          ;
01130          ;
01140          ;
01150          ;
01160
000023  700200 000000 01170  ASSASG:  CONO APR,CLDS      ;DISABLE CLOCK
000024  135040 000000 01180          LDB TAC,PJOBN      ;GET JOB NUMBER IN DEV DATA BLOCK
000025  316040 000004 01190          CAMN TAC,ITEM      ;IS IT ALREADY ASSIGNED TO THIS JOB
000026  254000 000035' 01200          JRST ASSAS1      ;YES
000027  201040 600000 01210          MOVEI TAC, ASSPRG+ASSCON  ;NO, IS IT ASSIGNED TO ANOTHE
000030  700600 000000 01220          CONO PI, SCNOFF  ;TURN SCANNER OFF
000031  612046 000004 01230          TDNE TAC, DEVMOD(DEVDAT) ;ARE EITHER ASSIGNED BITS SET
000032  254000 000036' 01240          JRST ASSAS2      ;YES
000033  436106 000004 01250          IORM TAC1, DEVMOD(DEVDAT) ;NO, SET ONE OF THEM
000034  137200 000024' 01260          DPB ITEM,PJOBN      ;AND STORE JOB NUMBER
000035  350003 000000 01270  ASSAS1:  AOS (PDP)
000036  700600 000000 01280  ASSAS2:  CONO PI, SCNON      ;TURN SCANNER BACK ON
000037  700200 000000 01290          CONO APR,CLEN      ;RENABLE CLOCK
000040  263140 000000 01300          POPJ PDP,

```

```

01310
01320 ;ROUTINE TO SEARCH FOR A DEVICE
01330 ;CALL:   HRR ITEM, JOB NUMBER
01340 ;       MOVE TAC, [SIXBIT .DEVICE NAME.]
01350 ;       PUSHJ PDP, DEVSRC
01360 ;       NOT FOUND
01370 ;       FOUND
01380
01390         INTERNAL DEVLG, DEVSRC, DEVPHY
01400         EXTERNAL SYSTAP, DEVOPR
01410
000041 260140 000044' 01420  DEVSRC:  PUSHJ PDP, DEVLG      ;SEARCH LOGICAL NAMES FIRST
000042 254000 000056' 01430          JRST DEVPHY          ;NOT FOUND, SEARCH PHYSICAL NAMES
000043 254000 000000 01440          JRST CPOPJ1         ;FOUND
01450
01460 ;SEARCH LOGICAL NAMES
01470
000044 554300 000000 01480  DEVLG:   HLRZ DEVDAT, DEVLST  ;BEGINNING OF DDB CHAIN
000045 322040 000000 01490          JUMPE TAC, CPOPJ      ;0 CANNOT BE A LOGICAL NAME
000046 312046 000005 01500  DEVLPO:  CAME TAC, DEVLOG(DEV DAT) ;COMAPRE WITH LOGICAL NAME
000047 254000 000053' 01510          JRST DEV0          ;NO MATCH
000050 135100 000034' 01520          LDB TAC1, PJOBN      ;DOES THE LOGICAL NAME BELONG TO THIS J
000051 316100 000004 01530          CAMN TAC1, ITEM
000052 254000 000043' 01540          JRST CPOPJ1         ;YES
000053 554306 000003 01550  DEV0:   HLRZ DEVDAT, DEVSER(DEV DAT) ;NO, KEEP LOOKING
000054 326300 000046' 01560          JUMPN DEVDAT, DEVLPO
000055 263140 000000 01570          POPJ PDP,           ;FINISHED AND NOT FOUND
01580
01590 ;SEARCH PHYSICAL NAMES
01600
000056 322040 000045' 01610  DEVPHY:  JUMPE TAC, CPOPJ      ;ZERO CANNOT BE A LOGICAL NAME
000057 316040 000370' 01620          CAMN TAC, [SIXBIT /OPR/] ;IS IT "OPR"?
000060 200040 000000 01630          MOVE TAC, DEVOPR      ;YES, CHANGE TO OPERATORS TTY
000061 316040 000371' 01640          CAMN TAC, [SIXBIT /SYS/] ;IS IT "SYS"?
000062 200040 000000 01650          MOVE TAC, SYSTAP     ;YES, CHANGE TO TO SYSTEM TAPE NAME
000063 554300 000044' 01660          HLRZ DEVDAT, DEVLST  ;SEARCH DEVICE DATA BLOCKS
000064 316046 000000 01670  DEVLPO1: CAMN TAC, DEVNAM(DEV DAT)
000065 254000 000052' 01680          JRST CPOPJ1         ;FOUND
000066 554306 000003 01690          HLRZ DEVDAT, DEVSER(DEV DAT)
000067 326300 000064' 01700          JUMPN DEVDAT, DEVLPO1
000070 263140 000000 01710          POPJ PDP, ;NOT FOUND

```

```

01720
01730 ,DEC 06 00 EX COM L PT PRE 01 BUFCLC
01740 ,H. R. MORSE 64-12-26
01750 ,CALLING SEQUENCE
01760 , PUSHJ PDP,BUFCLC
01770 , EXIT RETURNS HERE IF MEMORY NOT EXCEEDED
01780 , SETS UP AN N BUFFER RING FOLLOWING THE USERS PROGRAM, WHERE N
01790 , IS IN THE ADDRESS FIELD OF AC UUO.
01800 , THE BUFFER RING FORMAT IS AS FOLLOWS:
01810 , LOCATION LH CONTENTS RH
01820 , C(JOBFF) + 1 BUFFER C(JOBFF) + 1
01830 , + 0(BUFFER SIZE+2) SIZE + 1(BUFFER SIZE+2)
01840 , C(JOBFF) + 1 BUFFER C(JOBFF) + 1
01850 , + 1(BUFFER SIZE+2) SIZE + 2(BUFFER SIZE+2)
01860 , .
01870 , .
01880 , .
01890 , C(JOBFF) + 1 BUFFER C(JOBFF) + 1
01900 , + (N-2)(BUFFER SIZE+2) SIZE +(N-1)(BUFFER SIZE+2)
01910 , C(JOBFF) + 1 BUFFER C(JOBFF) + 1
01920 , + (N-1)(BUFFER SIZE+2) SIZE
01930 , THEN SET BUFPNT:=IOUSE,C(JOBFF) + 1
01940 , AND JOBFF:=C(JOBFF) + 1 + N(BUFFER SIZE + 2)
01950 , BUFWRD IS RESTORED.
01960 , MONITOR INTERFACE
01970 , STORAGE: 20
01980 , ROUTINES CALLED: ADRCK
01990 , SYMBOLS SET/USED:
02000 , ACCUMULATORS: BUFPNT S/U PROG U
02010 , BUFWRD S/U TAC S/U
02020 , DEVDAT U TAC1 S/U
02030 , PDP S/U UUO U
02040 , DEVICE DATA BLOCK: DEVCHR U
02050 , JOB AREA: JOBFF S/U
02060 , JOB BUFFER HEADER: IOUSE U

```

			02070		
			02080		
000071	261140	000013	02090	BUFCLC:	INTERN BUFCLC
000072	135040	000372'	02100		PUSH PDP,BUFWRD; SAVE BUFWRD ON STACK
000073	550507	000000	02110		LDB TAC,[POINT 12,DEVCHR(DEV DAT),35];TAC:=BUFFER SIZE
000074	271500	000001	02120		HRRZ BUFPNT,JOBFF(PROG);BUFPNT:=FIRST FREE LOCATION + 1
000075	550540	000012	02130		ADDI BUFPNT,1
000076	505500	000007	02140		HRRZ BUFWRD,BUFPNT
000077	504540	000001	02150		HRLI BUFPNT,PROG
000100	271040	000002	02160		HRL BUFWRD,TAC; BUFWRD:=BUFFER SIZE,FIRST FREE LOC + 1
000101	550100	000014	02170		ADDI TAC,2; TAC:=BUFFER SIZE + 2
000102	270540	000001	02180	BUFC1:	HRRZ TAC1,UUO; TAC1:=N=ADDRESS FIELD OF AC UUO
			02190		ADD BUFWRD,TAC; BUFWRD:=C(BUFWRD) + C(TAC)
000103	260140	000010'			ADRCHK BUFPNT; CHECK BUFFER FOR MEMORY BOUND
000104	550040	000012		PUSHJ PDP,ADRCK	
000105	202560	000012	02200		HRRZ TAC,BUFPNT
000106	540500	000013	02210		MOVEM BUFWRD,@BUFPNT; BUFFER HEADER+1:=C(BUFWRD)
000107	367100	000102'	02220		HRR BUFPNT,BUFWRD; BUFPNT 18-35:=C(BUFWRD 18-35)
000110	540547	000073'	02230		SOJG TAC1,BUFC1; N:=N-1. IS N>0?
000111	271540	000001	02240		HRR BUFWRD,JOBFF(PROG)
000112	260140	000103'	02250		ADDI BUFWRD,1
000113	551052	777776	02260		PUSHJ PDP,ADRCK ;CHECK LAST ADR. OF HEADER
000114	274500	000001	02270		HRRZI TAC,-2(BUFPNT)
000115	202560	000012	02280		SUB BUFPNT,TAC
000116	270500	000001	02290		MOVEM BUFWRD,@BUFPNT;LINK LAST BUFFER TO FIRST BUFFER
000117	542507	000110'	02300		ADD BUFPNT,TAC
000120	540500	000013	02310		HRRM BUFPNT,JOBFF(PROG);JOBFF:=C(JOBFF)+1+N(BUFFER SIZE+2)
			02320		HRR BUFPNT,BUFWRD; BUFPNT:=IOUSE,ADDRESS OF FIRST BUFFER
			02330		IN RING.
000121	505500	400000	02340		HRLI BUFPNT,IOUSE
000122	262140	000013	02350		POP PDP,BUFWRD; RESTORE BUFWRD FROM STACK.
000123	263140	000000			POPJ PDP,; RETURN

```

02360
02370 ,DEC 06 00 EX COM L PT PRE 01 BUFCLR
02380 ,H. R. MORSE 64-12-26
02390 ,CALLING SEQUENCE
02400 , PUSHJ PDP,BUFCLR
02410 , EXIT RETURNS HERE IF MEMORY NOT EXCEEDED
02420 , CLEARS THE WORD COUNT AND DATA AREA OF THE BUFFER WHOSE ADDRESS
02430 , IS IN TAC 18-35.
02440 ,MONITOR INTERFACE
02450 , STORAGE: 13
02460 , ROUTINES CALLED: ADRCK
02470 , SYMBOLS SET/USED:
02480 , ACCUMULATORS: PDP U TAC S/U
02490 , PROG U TAC1 S/U
02500 INTERN BUFCLR
000124 505040 000007 02510 BUFCLR: HRLI TAC,PROG
02520 ADRCHK TAC; PROCEED IF CURRENT BUFFER ADDRESS
000125 260140 000112' 02530 ,
000126 550040 000001 02540 HLRZ TAC1,@TAC; TAC1 18-35=SIZE
02550 TRZ TAC1,400000
000127 554120 000001 02560 ADD TAC1,TAC; TAC1:=CURRENT BUFFER ADDRESS + SIZE
000130 620100 400000 02570 ADRCHK TAC1; PROCEED IF CURRENT BUFFER ADDRESS
000131 270100 000001
000132 260140 000125' 02580 ,
000133 550040 000002 02590 MOVEI TAC,@TAC; TAC:=CURRENT BUFFER ADDRESS+1,
02600 , CURRENT BUFFER ADDRESS+1
000134 201060 000001 02610 HRL TAC,TAC
000135 504040 000001 02620 AOBJN TAC,+.1
000136 253040 000137' 02630 CLEARM 0(TAC); WORD COUNT:=0
000137 402001 000000 02640 AOS TAC TAC:=CURRENT BUFFER ADDRESS+1,
000140 350000 000001 02650 , CURRENT BUFFER ADDRESS+2
000141 251060 000002 02660 BLT TAC,@TAC1; CLEAR BUFFER
000142 263140 000000 02670 POPJ PDP,; RETURN

```

02680

```
02700  
02710 ,CALLING SEQUENCE  
02720 ,     PUSHJ PDP,CKS12  
02730 ,     EXIT                                ALWAYS RETURNS HERE  
02740 ,CALCULATES FOLDED 12 BIT CHECKSUMS OF THE DATA WORDS IN THE  
02750 ,BUFFER WHOSE ADDRESS IS IN AC TAC1. TWO ALGORITHMS ARE USED.  
02760 ,ON RETURN, THE LEFT HALF OF AC TAC CONTAINS A CHECKSUM OBTAINED  
02770 ,BY ACCUMULATING, IN ONES COMPLEMENT, THE DATA WORDS AND FOLDING IT.  
02780 ,THE LEFT HALF OF AC DAT CONTAINS A CHECKSUM OBTAINED BY ACCUMULATING  
02790 ,IN TWOS COMPLEMENT, THE DATA WORDS AND FOLDING IT. AC TAC1  
02800 ,CONTAINS A 1.  
02810 ,MONITOR INTERFACE  
02820 ,     STORAGE: 20  
02830 ,     ROUTINES CALLED: FOLD  
02840 ,     SYMBOLS SET/USED:  
02850 ,     ACCUMULATORS:  DAT  S/U          SUB  S  
02860 ,                     PDP  U           TAC  S/U  
02870 ,                     PROG U           TAC1 S/U
```

```

02880
02890
000143 270100 000007 02900
000144 350000 000002 02910
000145 550042 000000 02920
000146 213000 000001 02930
000147 350000 000002 02940
000150 504100 000001 02950
000151 402000 000001 02960
000152 270042 000000 02970
000153 253100 000152 02980
000154 246040 777750 02990
000155 242100 777764 03000
000156 270040 000002 03010
000157 246040 777764 03020
000160 242100 777750 03030
000161 270040 000002 03040
000162 622040 770000 03050
000163 350000 000001 03060
000164 517000 000001 03070
000165 201100 000001 03080
000166 263140 000000 03090

```

```

INTERN CKS12
CKS12:  ADD TAC1,PROG;      TAC1:=-WORD COUNT,ADDRESS OF FIRST DATA
        AOS TAC1
        HRRZ TAC,0(TAC1)
        MOVNS TAC
        AOS TAC1
        HRL TAC1,TAC
        CLEARM TAC;
CKS12A: ADD TAC,0(TAC1);   INITIALIZE TWOS COMPLEMENT SUM
        AOBJN TAC1,CKS12A; TWOS COMPLEMENT ADD
                                DONE?
FOLD:  LSHC TAC,-30
        LSH TAC1,-14
        ADD TAC,TAC1
        LSHC TAC,-14
        LSH TAC1,-30
        ADD TAC,TAC1
        TRZE TAC,770000
        AOS TAC
        HRLZS TAC
        MOVEI TAC1,1      ;TAC1:=1
        POPJ PDP,

```

```

03100
03110 ,DEC 06 00 EX COM L PT PRE 01 CLRBYT
03120 ,H. R. MORSE 64-12-26
03130 ,CALLING SEQUENCE
03140 , PUSHJ PDP,CLRBYT
03150 , EXIT ALWAYS RETURNS HERE
03160 ,CALLED WITH A BYTE POINTER IN AC TAC, IT CLEARS THE REST OF THE
03170 ,WORD POINTED TO BY THE BYTE POINTER.
03180 ,MONITOR INTERFACE
03190 , STORAGE: 7
03200 , SYMBOLS SET/USED:
03210 , ACCUMULATORS: PDP U
03220 , TAC U
03230 , TAC1 S/U
03240 INTERN CLRBYT
03250 CLRBYT: LDB TAC1,[POINT 6,TAC,5]; TAC1:=P
03260 DPB TAC1,[POINT 12,TAC,11]; TAC 0-5:=0,TAC 6-12:=P
03270 SETZM TAC1
03280 DPB TAC1,TAC; CLEAR BITS 36-P THROUGH 35
03290 POPJ PDP,; RETURN

000167 135100 000373' 03250
000170 137100 000374' 03260
000171 402000 000002 03270
000172 137100 000001 03280
000173 263140 000000 03290

```

```

03300
03310 ;ROUTINE TO RELEASE ALL DEVICES ASSIGNED TO JOB
03320
03330 INTERNAL IORELS
03340 EXTERNAL PUUOAC,RELEA3
03350
03360
000174 201040 000000 03370 IORELS: MOVEI TAC,RELEA3 ;RELEASE ALL IO DEVICES(DONT CLOSE)
03380
03390 ;ROUTINE TO DO IO FOR ALL DEVICES ASSIGNED TO JOB
03400 ;CALL MOVEI TAC,ADR. OF IO SUB.
03410 ; PUSHJ PDP,IOALL
03420
03430 INTERNAL IOALL
03440
000175 261140 000001 03450 IOALL: PUSH PDP,TAC ;SAVE ADR. FO SUB.
000176 515240 777760 03460 HRLZI DAT, -20
000177 200305 000000 03470 IOALL1: MOVE DEVDAT, USRJDA(DAT)
000200 200006 000002 03480 MOVE IOS,DEVIOS(DEVDAT)
000201 201600 000000 03490 MOVEI UO,0
000202 137240 000000 03500 DPB DAT,PUUOAC
000203 332000 000006 03510 SKIPE DEVDAT
000204 260163 000000 03520 PUSHJ PDP,@(PDP) ;CALL THE SUB.
000205 253240 000177 03530 AOBJN DAT, IOALL1
000206 262140 000001 03540 POP PDP,TAC
000207 263140 000000 03550 POPJ PDP,
03560
03570 ;KILL ALL DEVICES(RELEASE WITHOUT WAITING FOR DEVICE INACTIVE)
03580
03590 INTERNAL IOKILL
03600 EXTERNAL RELEA5
03610
000210 201040 000000 03620 IOKILL: MOVEI TAC,RELEA5
000211 254000 000175 03630 JRST IOALL

```

```

03640
03650 ,DEC 06 00 EX COM L PT PRE 01 IOSET
03660 ,H. R. MORSE 64-12-26
03670 ,CALLING SEQUENCE
03680 , PUSHJ PDP,IOSET
03690 , EXIT ALWAYS RETURNS HERE
03700 ,THIS PROGRAM IS CALLED FROM AN INTERRUPT SERVICE ROUTINE.
03710 ,IT PUTS THE ADDRESS OF THE DATA AREA OF THE JOB (C(JBTADR18-35))
03720 ,CONNECTED TO THE DEVICE SPECIFIED BY AC DEVDAT IN AC PROG AND
03730 ,PUTS THE ITEM POINTER (C(DEVCTR)) IN AC ITEM.
03740 ,MONITOR INTERFACE
03750 , STORAGE: 5
03760 , SYMBOLS SET/USED:
03770 , ACCUMULATORS: DEVDAT U PDP U
03780 , ITEM S PROG S/U
03790 , DEVICE DATA BLOCK: DEVCHR U
03800 , DEVCTR U
03810 , SYSTEM ADDRESSES: JBTA DR U
03820 INTERN IOSET
03830
000212 135340 000050' 03840 IOSET: LDB PROG,PJOBN
000213 200206 000011 03850 MOVE ITEM,DEVCTR(DEVDAT); ITEM:=ITEM POINTER=C(DEVCTR)
000214 550347 000000 03860 HRRZ PROG,JBTADR(PROG); PROG:=C(JBTADR 18-35)
000215 263140 000000 03870 POPJ PDP,; RETURN

```

```

03880
03890 ,6-CSS-DEC-IOSETC-PL-PRE1
03900 ,H.R. MORSE 17-11-64
03910
03920 ,CALLING SEQUENCE
03930 , PUSHJ PDP,IOSETC
03940 , EXIT ALWAYS RETURNS HERE
03950
03960 ,SETS JBFPTR18-35:=C(TAC1 18-35)
03970 , JBFCTR:=C(ITEM)*[WORD LENGTH/BYTE SIZE]
03980 ,WHERE WORD LENGTH:=36 DECIMAL
03990 , BYTE SIZE:=C(JBFPTR6-11)
04000 , [X]:= INTEGER PART OF X
04010
04020 ,MONITOR INTERFACE
04030 , STORAGE: 11
04040 , ROUTINES CALLED: ITMCT1
04050 , SYMBOLS SET/USED:
04060 , ACCUMULATORS: JBUF S/U TAC S/U
04070 , ITEM S/U TAC1 S/U
04080 , PDP U
04090
04100 INTERN IOSETC
04110
04120
04130 IOSETC: ADDI JBUF,1 ;JBFPTR12-18:=0
04140 HRLZI TAC,7777 ;JBFPTR18-35:=C(TAC1 18-35)+1
04150 ANDM TAC,@JBUF
04160 HRRM TAC1,@JBUF
04170 AOS @JBUF
04180 LDB TAC1,[POINT 6,@JBUF,11] ;TAC1:=BYTE SIZE
04190 PUSHJ PDP,ITMCT1 ;JBFCTR:=C(ITEM)*[36/BYTE SIZE]
04200 ADDI JBUF,1
04210 MOVEM ITEM,@JBUF
04220 POPJ PDP, ;EXIT
000216 271240 000001
000217 515040 007777
000220 406060 000005
000221 542120 000005
000222 350020 000005
000223 135100 000375'
000224 260140 000233'
000225 271240 000001
000226 202220 000005
000227 263140 000000

```

```
04230
04240 ,DEC 06 00 EX COM L PT PRE 01 ITMSET, ITMCNT, ITMCT1
04250 ,H.R. MORSE 64-12-26
04260
04270 ,CALLING SEQUENCE
04280 ,     PUSHJ PDP,ITMSET
04290 ,     EXIT             ALWAYS RETURNS HERE
04300 ,SETS AC ITEM:=(BUFFER SIZE-1)*[WORD LENGTH/BYTE SIZE]
04310 ,WHERE BUFFER SIZE:=BITS 1-17 OF THE BUFFER HEADER WORD POINTED TO
04320 ,     BY C(DEVADR)
04330 ,     WORD LENGTH:=36 DECIMAL
04340 ,     BYTE SIZE:=INTEGER PART OF X.
04350
04360 ,CALLING SEQUENCE
04370 ,     PUSHJ PDP,ITMCNT
04380 ,     EXIT             ALWAYS RETURNS HERE
04390 ,SETS AC ITEM:=C(ITEM)*[WORD LENGHT/BYTE SIZE]
04400
04410 ,CALLING SEQUENCE
04420 ,     PUSHJ PDP,ITMCT1
04430 ,     EXIT             ALWAYS RETURNS HERE
04440 ,SETS AC ITEM:=C(ITEM)*[WORD LENGHT/C(TAC1)]
04450
04460 ,MONITOR INTERFACE
04470 ,     STORAGE: 7
04480 ,     SYMBOLS SET/USED:
04490 ,     ACCUMULATORS:  DEVDAT U  TAC S/U
04500 ,                     ITEM S/U  TAC1 S/U
04510 ,                     PDP U
04520 ,     DEVICE DATA BLOCK: DEVADR U
04530 ,                     DEVPTR U
```

			04540
			04550
000230	135200	000376'	04560
000231	275200	000001	04570
000232	135100	000377'	04580
000233	201040	000044	04590
000234	230040	000002	04600
000235	220200	000001	04610
000236	263140	000000	04620

```
INTERN ITMSET,ITMCNT,ITMCT1
ITMSET:  LDB ITEM,[POINT 17,@DEVADR(DEVDAT),17];ITEM:=BUFFER SIZE-1
          SUBI ITEM,1
ITMCNT:  LDB TAC1,[POINT 6,DEVPTR(DEVDAT),11];TAC1:=BYTE SIZE
ITMCT1:  MOVEI TAC,44          ;ITEM:=C(ITEM)*[WORD LENGTH/C(TAC1)]
          IDIV TAC,TAC1
          IMUL ITEM,TAC
          POPJ PDP,
```

			04630	
			04640	;ROUTINE TO SET DEVICE STATUS WORD FROM UUU
			04650	
			04660	INTERNAL SETIOS
			04670	EXTERNAL ILLMOD
			04680	
000237	135100	000400	04690	SETIOS: LDB TAC1,[POINT 4,UUU,35] ;GET DEVICE DATA MODE
000240	201040	000001	04700	MOVEI TAC,1 ;AND CHECK FOR LEGALITY
000241	242042	000000	04710	LSH TAC,(TAC1)
000242	616046	000004	04720	TDNN TAC,DEVMOD(DEV DAT)
000243	254000	000000	04730	JRST ILLMOD ;ILLEGAL MODE
000244	620600	010000	04740	TRZ UUU,IOACT ;LET USER SET ALL BITS EXCEPT IOACT
000245	542606	000002	04750	HRRM UUU,DEVIOS(DEV DAT)
000246	263140	000000	04760	POPJ PDP,

```
04770
04780 ,DEC 06 00 EX COM L PT PRE 01 NEWBUF, BPNSET
04790 ,H. R. MORSE 64-12-26
04800 ,CALLING SEQUENCE
04810 , PUSHJ PDP,NEWBUF
04820 , EXIT ALWAYS RETURNS HERE
04830 ,CLEARS THE BUFFER CURRENTLY POINTED TO BY THE INPUT BUFFER
04840 ,ADDRESS (DEVDR) OF THE CURRENT DEVICE. SETS UP THE BYTE
04850 ,POINTER (DEVPTR), AND THE ITEM COUNT (DEVCTR) AND RETURNS.
04860 ,CALLING SEQUENCE
04870 , PUSHJ PDP,BPNSET
04880 , EXIT ALWAYS RETURNS HERE
04890 ,SETS UP THE BYTE POINTER (DEVPTR) AND THE ITEM COUNT (DEVCTR)
04900 ,OF THE CURRENT DEVICE AND RETURNS.
04910 ,MONITOR INTERFACE
04920 , STORAGE: 10
04930 , ROUTINES CALLED: BUFCLR, ITMSET
04940 , SYMBOLS SET/USED:
04950 , ACCUMULATORS: DEVDAT U PDP U
04960 , ITEM U TAC S/U
04970 , DEVICE DATA BLOCK: DEVADR U
04980 , DEVCTR S
04990 , DEVPTR S/U
```

			05000
			05010
			05020
000247	550046	000007	05030
000250	260140	000124	05040
000251	540046	000007	05050
000252	515040	007737	05060
000253	404046	000010	05070
000254	540046	000007	05080
000255	350000	000001	05090
000256	202046	000010	05100
000257	260140	000230	05110
000260	202206	000011	05120
000261	263140	000000	05130

```
INTERN BPNSET,NEWBUF  
NEWBUF: HRRZ TAC,DEVADR(DEVDAT); TAC:=INPUT BUFFER HEADER ADDRESS  
        PUSHJ PDP,BUFCLR; CLEAR INPUT BUFFER.  
BPNSET: HRR TAC,DEVADR(DEVDAT)  
        HRLZI TAC,7737  
        AND TAC,DEVPTR(DEVDAT); DEVPTR 0-5:=0, DEVPTR 12:=0  
        HRR TAC,DEVADR(DEVDAT); DEVPTR 18-35:=C(DEVADR 18-35) + 1  
        AOS TAC  
        MOVEM TAC,DEVPTR(DEVDAT)  
        PUSHJ PDP,ITMSET; ITEM:=(BUFFER SIZE-1)*[36/BYTE SIZE]  
        MOVEM ITEM,DEVCTR(DEVDAT); DEVCTR:=ITEM COUNT  
        POPJ PDP,; RETURN
```

```

05140
05150 ,DEC 06 00 EX COM L PT PRE 02 SETBYT
05160 ,H. R. MORSE 64-12-26
05170 ,CALLING SEQUENCE
05180 , PUSHJ PDP,SETBYT
05190 , EXIT ALWAYS RETURNS HERE
05200 ,SETS TAC 0-5:=0
05210 , TAC 6-11:=S
05220 , TAC 12-13:=0
05230 , TAC 14-17:=PROG
05240 ,WHERE S=36 IF DATA MODE (IOS 32-25) IS BINARY (B)
05250 , S=8 IF DATA MODE IS IMAGE (I)
05260 , S=7 IF DATA MODE IS ASCII PACKED (A)
05270 , ASCII LINE (AL)
05280 , ASCII SEQUENCED (AS)
05290 , ASCII SEQUENCED LINE (ASL)
05300 , OR ALTERNATE MODE BREAK (AM)
05310 ,MONITOR INTERFACE
05320 , STORAGE: 12
05330 , SYMBOLS SET/USED:
05340 , ACCUMULATORS: PROG U TAC S
05350 , PDP U TAC1 S/U
05360 , DEVICE DATA BLOCK: DEVIOS U
05370 INTERN SETBYT
05380 SETBYT: MOVE TAC1,DEVIOS(DEVDAT); FETCH DATA MODE, BITS 32-35 0
05390 ANDI TAC1,17
05400 CAIE TAC1,IB ;DATA MODE IMAGE BINARY?
05410 CAIN TAC1,B; IS DATA MODE=BINARY?
05420 HRLI TAC,4400+PROG; SET BYTE SIZE TO 36.
05430 CAIN TAC1,I; IS DATA MODE=IMAGE?
05440 JRST SETBY1
05450 TRNE TAC1,14; IS DATA MODE=ASCII PACKED,ASCII LINE,
05460 JRST .+2; ASCII SEQUENCED,OR ASCII SEQUENCED LINE
05470 HRLI TAC,700+PROG; SET BYTE SIZE TO 7.
05480 POPJ PDP,; RETURN
05490
05500 SETBY1: HLLZ TAC,DEVMOD(DEVDAT)
05510 TLZ TAC,770077
05520 TLO TAC,PROG
05530 POPJ PDP,
000262 200106 000002 05380
000263 405100 000017 05390
000264 302100 000013 05400
000265 306100 000014 05410
000266 505040 004407 05420
000267 306100 000010 05430
000270 254000 000275' 05440
000271 602100 000014 05450
000272 254000 000274' 05460
000273 505040 000707 05470
000274 263140 000000 05480
05490
000275 510046 000004 05500
000276 621040 770077 05510
000277 661040 000007 05520
000300 263140 000000 05530

```

```

05540
05550 ,DEC 06 00 EX COM L PT PRE 01 STORE ITEM
05560 ,H.R. MORSE          64-12-26
05570 ,CALLING SEQUENCE
05580 ,      PUSHJ PDP,STODAT
05590 ,      EXIT1          CHECKSUM ERROR
05600 ,      EXIT2          BLOCK FULL OR BLOCK COMPLETE
05610 ,      EXIT3          DATA STORED CORRECTLY
05620 ,CALLED FROM AN INPUT SERVICE ROUTINE WITH A DATA ITEM IN AC DAT.
05630 ,STORES THE DATA ITEM IN THE BUFFER, CHECKING TO SEE IF IT WERE
05640 ,THE FIRST ITEM ON THE BUFFER AND SETTING UP THE POINTER AND
05650 ,WORD COUNT APPROPRIATELY CHECKING THE MODE TO SEE IF ANY SPECIAL
05660 ,PROCESSING NEED BE DONE.  FOR EXAMPLE, GENERATION OF SEQUENCE
05670 ,NUMBERS PRECEDING EACH LINE IN SEQUENCE MODE, OR THE TERMINATION
05680 ,OF A BUFFER ON CERTAIN CHARACTERS IN OTHER MODES, OR IF THE BUFFER
05690 ,IS FULL.  THERE ARE THREE RETURNS FROM THIS ROUTINE: THE FIRST
05700 ,RETURN OCCURS ON AN ERROR CONDITION, THE SECOND RETURN OCCURS
05710 ,ON A BLOCK FULL CONDITION OR BLOCK COMPLETE CONDITION, THE THIRD
05720 ,RETURN OCCURS ON THE DATA STORED CORRECTLY CONDITION.  THIS
05730 ,ROUTINE ALSO DOES SOME CHECKING ON INPUT OF BINARY RECORD,
05740 ,PAPER TAPE OR CARDS.
05750 ,CALLING SEQUENCE
05760 ,      PUSHJ PDP,STOSQD
05770 ,      XXXX          ALWAYS SKIPS
05780 ,      EXIT          ALWAYS RETURNS HERE
05790 ,STORES THE WORD COUNT:=C(DEVPTR 18-35) -C(DEVIAD 18-35) -- 1
05800 ,IN THE BUFFER.
05810 ,MONITOR INTERFACE
05820 ,      STORAGE: 69
05830 ,      ROUTINES CALLED: ASCIA, CKS12, ITMSET, NEWBUF
05840 ,      SYMBOLS SET/USED:
05850 ,      ACCUMULATORS:          DAT    U    PDP    U
05860 ,                              DEVDAT U    TAC S/U
05870 ,                              IOS     S/U          TAC1 S/U
05880 ,                              ITEM   S/U
05890 ,      DEVICE DATA BLOCK:   DEVCTR U    DEVPTR S/
05900 ,                              DEVIAD U    DEVSEQ S/
05910 ,      IO STATUS WORD: A    U          I    U
05920 ,                              AS     U          IOBKTL S
05930 ,                              ASL    U          IODTER S
05940 ,                              B      U          IOFST  S/
05950 ,      JOB BUFFER AREA:     WORD COUNT S

```

			05960
			05970
			05980
000301	603000	000004	05990
			06000
000302	260140	000247	06010
			06020
000303	135100	000000	06030
000304	306100	000014	06040
000305	254000	000344	06050
000306	621000	000004	06060
000307	136246	000010	06070
000310	302100	000000	06080
000311	306100	000010	06090
000312	254000	000324	06100
000313	306100	000014	06110
000314	254000	000324	06120
000315	307240	000214	06130
000316	305240	000212	06140
000317	254000	000321	06150
000320	254000	000336	06160
000321	365200	000000	06170
000322	660000	040000	06180
000323	254000	000056	06190
000324	367200	000321	06200
000325	306100	000000	06210
000326	254000	000336	06220
000327	306100	000014	06230
000330	254000	000354	06240
000331	260140	000230	06250
			06260
000332	274206	000011	06270
000333	550106	000007	06280
000334	271200	000001	06290
000335	344100	000342	06300

```

INTERN STODAT, STOSQD
STODAT:  TLNE IOS, IOFST;      WILL THE NEXT ITEM BE THE FIRST ITEM
,        OF A BUFFER?
,        PUSHJ PDP, NEWBUF;    SET UP A NEW BUFFER. ITEM:=(BUFFER
,        SIZE - 1)*[36/BYTE SIZE]
,        LDB TAC1, PIOMOD;      TAC1:=DATA MODE
,        CAIN TAC1, B;          MODE=BINARY?
,        JRST STOBIN
,        TLZ IOS, IOFST
STO1:    DPBI DAT, DEVPTR(DEV DAT); STORE DATA IN BUFFER.
,        CAIE TAC1, A;          MODE=ASCII, IMAGE, OR BINARY?
,        CAIN TAC1, I
,        JRST STOAIB
,        CAIN TAC1, B;          CAIN TAC1, I B
,        JRST STOAIB
,        CAIG DAT, 214          ;LINE FEED, FORM FEED, OR VERTICAL TAB?
,        CAIGE DAT, 212
,        JRST .+2 ;NO
,        JRST STOSQD          ;YES
,        SOJGE ITEM, CPOPJ2;    ITEM:=C(ITEM)-1. IS C(ITEM)>0?
STOE1:   TRO IOS, IOBKTL      ;IOBKTL:=1
,        JRST CPOPJ
STOAIB:  SOJG ITEM, CPOPJ2;    ITEM:=C(ITEM)-1. IS C(ITEM)>0?
,        CAIN TAC1, A;          MODE=ASCII?
,        JRST STOSQD
,        CAIN TAC1, B;          MODE=BINARY?
,        JRST STOBND
STOI:    PUSHJ PDP, ITMSET;    ITEM:=(BUFFER SIZE-1)*[36/BYTE SIZE]
,        - C(DEVCTR)
,        SUB ITEM, DEVCTR(DEV DAT)
,        HRRZ TAC1, DEVIAD(DEV DAT) ;STORE ITEM COUNT INSTEAD OF
,        ADDI ITEM, 1          ;IN FIRST WORD OF BUFFER
,        AOJA TAC1, STOSQE

```

*STOSQD: TLZE IOS, IOFST
PUSHJ PDP, NEWBUF*

MOVE

CAIE TAC1, I B

			06310
000336	200106	000007	06320
000337	350000	000002	06330
000340	550206	000010	06340
			06350
000341	274200	000002	06360
000342	542220	000002	06370
000343	254000	000065'	06380
000344	627000	000004	06390
000345	254000	000307'	06400
000346	550040	000005	06410
000347	313040	000004	06420
000350	254000	000322'	06430
000351	200200	000001	06440
000352	202266	000010	06450
000353	254000	000324'	06460
000354	550106	000007	06470
000355	260140	000143'	06480
000356	270106	000007	06490
000357	510120	000002	06500
000360	316040	000002	06510
000361	254000	000343'	06520
000362	660000	100000	06530
000363	254000	000323'	06540
			06550

```

STOSQ: MOVE TAC1,DEVIAD(DEV DAT); TAC1:=ADDRESS OF WORD COUNT
      AOS TAC1
      HRRZ ITEM,DEVPTR(DEV DAT); ITEM:=C(DEV PTR 18-35) --
      C(DEVIAD 18-35) -1
      ,
STOSQE: SUB ITEM,TAC1
      HRRM ITEM,@TAC1 ;WORD COUNT TO FIRST WORD IN BUFFER
      JRST CPOPJ1; EXIT2. BLOCK COMPLETE
STOBIN: TLZN IOS,IOFST; WILL THE NEXT ITEM BE THE FIRST ITEM
      JRST ST01; OF A BUFFER? IOFST:=0
      HRRZ TAC,DAT
      CAMLE TAC,ITEM; IS WORD COUNT <OR= (BUFFER SIZE-1)*
      JRST ST0E1; [36/BYTE SIZE]?
      MOVE ITEM,TAC; ITEM:=WORD COUNT
      MOVEM DAT,@DEVPTR(DEV DAT); STORE WORD COUNT IN BUFFER
      JRST CPOPJ2; EXIT3. DATA STORED CORRECTLY.
STOBND: HRRZ TAC1,DEVIAD(DEV DAT)
      PUSHJ PDP,CKS12; COMPUTE CHECKSUM
      ADD TAC1, DEVIAD(DEV DAT)
      HLLZ TAC1,@TAC1; DATA CHECKSUM=COMPUTED CHECKSUM?
      CAMN TAC,TAC1
      JRST CPOPJ1; EXIT2. BLOCK COMPLETE
      TRO IOS,IODTER; IODTER:=1
      JRST CPOPJ; EXIT1. CHECKSUM ERROR

```

		06560		
		06570	,DEC 06 00 EX COM L PT PRE 01 UINTQ	
		06580	,H.R. MORSE 64-12-26	
		06590	,CALLING SEQUENCE	
		06600	, PUSHJ PDP, UINTQ	
		06610	, EXIT ALWAYS RETURNS HERE	
		06620	,TAC:=CONTENTS OF WORD FOLLOWING UO CALL	
		06630	,FROM USERS AREA.	
		06640	,MONITOR INTERFACE	
		06650	, STORAGE: 4	
		06660	, SYMBOLS SET/USED:	
		06670	, ACCUMULATORS: PDP U TAC S/U	
		06680	, PROG U	
		06690	INTERN UINTQ	
000364	200043	777777	06700 UINTQ: MOVE TAC, -1(PDP)	;TAC:=ADDRESS FOLLOWING UO C
000365	505040	000007	06710 HRLI TAC, PROG	;RELOCATE ADDRESS
000366	200060	000001	06720 MOVE TAC, @TAC	;TAC:=CONTENTS OF WORD FOLLOW
			06730	;UO CALL FROM USERS AREA
000367	263140	000000	06740 POPJ PDP,	;RETURN
			06750	
000370	576062	000000	06760	END,
000371	637163	000000		
000372	001406	000001		
000373	360600	000001		
000374	301400	000001		
000375	300620	000005		
000376	222126	000007		
000377	300606	000010		
000400	000400	000014		

THERE ARE NO ERRORS

PROGRAM BREAK IS 000401

A	000000	INT
AC1	000015	INT
AC2	000016	INT
AC3	000017	INT
ADRCK	000132'	EXT
ADVBFE	000000'	INT
ADVBFF	000010'	INT
AL	000001	INT
ASSAS1	000035'	
ASSAS2	000036'	
ASSASG	000023'	INT
ASSCON	400000	INT
ASSIN	000020'	INT
ASSPRG	200000	INT
B	000014	INT
BPNSSET	000251'	INT
BUFC1	000102'	
BUFCLC	000071'	INT
BUFCLR	000124'	INT
BUFPNT	000012	INT
BUFWRD	000013	INT
CKS12	000143'	INT
CKS12A	000152'	
CLDS	000023'	EXT
CLEN	000037'	EXT
CLOSB	002000	INT
CLOSE1	000000	EXT
CLRBYT	000167'	INT
CLSIN	000002	INT
CLSOUT	000001	INT
CPOPJ	000363'	EXT
CPOPJ1	000361'	EXT
CPOPJ2	000353'	EXT
D	000017	INT
DAT	000005	INT
DCL	000001	INT
DCW	020000	INT
DDI	000007	INT
DDO	000006	INT
DDTMEM	000037	INT
DDTSYM	000036	INT
DEN	000004	INT
DEV0	000053'	
DEVADR	000007	INT
DEVBUF	000006	INT
DEVCHR	000001	INT
DEVCTR	000011	INT
DEVDAT	000006	INT
DEVIAD	000007	INT
DEVIOS	000002	INT
DEVLG	000044'	INT
DEVLOG	000005	INT
DEVLP0	000046'	
DEVLP1	000064'	
DEVLST	000063'	EXT

DEVMOD	000004	INT
DEVNAM	000000	INT
DEVOAD	000010	INT
DEVOPR	000060'	EXT
DEVPHY	000056'	INT
DEVPTR	000010	INT
DEVSER	000003	INT
DEVSRC	000041'	INT
DGF	000012	INT
DIN	000003	INT
DLK	000005	INT
DOU	000002	INT
DR	000016	INT
DRL	000000	INT
DSI	000011	INT
DSO	000010	INT
DTW	040000	INT
DVAVAL	000040	INT
DVCDR	100000	INT
DVDIR	000004	INT
DVDIRI	400000	INT
DVIN	000002	INT
DVLPT	040000	INT
DVMTA	000020	INT
DVOUT	000001	INT
DVTTY	000010	INT
ENTRB	020000	INT
ERROR	000000	EXT
FOLD	000154'	
I	000010	INT
IB	000013	INT
IBUFB	200000	INT
ILLMOD	000243'	EXT
INITB	400000	INT
INPB	010000	INT
IO	000020	INT
IOACT	010000	INT
IOALL	000175'	INT
IOALL1	000177'	
IOREG	000002	INT
IOBKTL	040000	INT
IOCON	000040	INT
IODEND	020000	INT
IODERR	200000	INT
IODISC	400000	INT
IODONE	400000	INT
IODTER	100000	INT
IOEND	000040	INT
IOFST	000004	INT
IOIMPM	400000	INT
IOKILL	000210'	INT
IONRCK	000100	INT
IOREL	000100	INT
IORELS	000174'	INT
IORET	000020	INT

IOS	000000	INT
IOSET	000212'	INT
IOSETC	000216'	INT
IOSTRT	000010	INT
IOUSE	400000	INT
IOW	000001	INT
IOWC	000020	INT
IOWS	400000	INT
ITEM	000004	INT
ITMCNT	000232'	INT
ITMCT1	000233'	INT
ITMSET	000230'	INT
JBFA DR	000000	INT
JBFC TR	000002	INT
JBFP TR	000001	INT
JBTADR	000214'	EXT
JBUF	000005	INT
JDAT	000011	INT
JERR	002000	INT
JIOW	100000	INT
JNA	004000	INT
JOB	000000	EXT
JORFF	000117'	EXT
LOOKB	040000	INT
MTW	010000	INT
NEWBUF	000247'	INT
OBUFB	100000	INT
OUTPB	004000	INT
PDP	000003	INT
PIC HN	000100	INT
PIOMOD	000303'	EXT
PJOB N	000212'	EXT
PROG	000007	INT
PUUOAC	000202'	EXT
RELEA1	000000	EXT
RELEA3	000174'	EXT
RELEA5	000210'	EXT
RUN	200000	INT
RUNABL	204000	INT
SCNOFF	000030'	EXT
SCNON	000036'	EXT
SETBY1	000275'	
SETBYT	000262'	INT
SETIOS	000237'	INT
ST01	000307'	
STOAIB	000324'	
STOBIN	000344'	
STOBND	000354'	
STODAT	000301'	INT
STOE1	000322'	
STOI	000331'	
STOSQD	000336'	INT
STOSQE	000342'	
SYSTAP	000062'	EXT
TAC	000001	INT

IOCSS - IO COMMON SYSTEM SUBROUTINES
SYMBOL TABLE

PAGE 33

TAC1	000002	INT
TEM	000010	INT
TTYATC	020000	INT
TTYUSE	010000	INT
UINTQ	000364'	INT
USRJDA	000177'	EXT
USRMOD	010000	INT
UUO	000014	INT
WAIT1	000000	EXT

END OF ASSEMBLY