Unattended Operations Software System Design Progress Report

L. Lim
Radio Frequency and Microwave Subsystems Section

A small MBASIC™ program called ORGNIZE has been developed to assist the software designer in the organization of MDS PL/M programming modules. It provides a logical way to distribute the various programming procedures in the proper module. The program also provides a consolidated listing of any procedure call and a summary of all procedures used by any procedure call.

The DSS 13 Unattended Station Controller has been in operation since December 1, 1977. The station controller design is based on an Intel 8080 microprocessor. The main program which controls the subsystem is a large computer program written in PL/M, a high level programming language. The program contains 87 procedures and was developed on the UNIVAC 1108, a maxicomputer. The station controller program was then cross-compiled from the U1108 to the station controller. Because of the inaccessibility of the U1108 and the excessive cost of cross-compilation of large PL/M programs, a Microcomputer Development Systems (MDS) has been acquired. The MDS has a resident PL/M compiler and other software development tools. However, the sizes of the memory storage and the auxiliary storage are obviously less than the U1108. Therefore, if the conversion of the U1108 PL/M program to the MDS PL/M system is to be possible, one must reorganize the large program into a number of smaller programming modules, compile these modules separately, then link and collect these modules on the MDS to form the final executable program.

A small MBASIC™* program, called ORGNIZE, has been developed to assist the software designer in the organization of MDS PL/M programming modules. It provides a logical way to distribute the various programming procedures in the proper module. The program also provides a consolidated listing of any procedure call and a summary of all procedures used by any procedure call.

The MBASIC™ program ORGNIZE requires a file which contains all the procedure names, sorted in lexicographic order, used by the various modules. Procedure names which are used by the module are surrounded by parentheses (i.e., external procedure); otherwise parentheses are omitted for public procedure within the modules and source files of each module. These files are available from the MDS. The program will prompt the user to input which procedure call is to be analyzed.

*A trademark of the California Institute of Technology.

The program ORGNIZE was designed to analyze three PL/M syntaxes: (1) definition of procedure, (2) end of procedure, and (3) the calling of procedures. Procedure definition takes the following form:

Procedure name: PROCEDURE;.

End of procedure takes the following form:

END procedure name;

Procedure call takes the following form:

CALL procedure name;

The basic function of the program ORGNIZE is to trace through all procedure calls generated by the user's input, i.e., the procedure call which the user wanted to be analyzed. The program first determines the file location of the User's request from the sorted file, PROCEDURE TABLE. The located PL/M source file is then opened for input. The requested procedure is then analyzed for procedure calls. These calls are then stacked onto a queue to be used as the next procedure to be analyzed. The program will terminate when the queue is empty. The file PROCTBL is shown in Fig. 1, and the listing of the program ORGNIZE is shown in Fig. 2.

The DSS 13 Unattended Station Controller PL/M program contains four basic procedures: INITIATE, CONFIGURE, CALIBRATE, and OPERATE. Figure 3 shows the listing of all procedure calls by each of the above four procedures. Figure 4 shows a summary of these called procedures in tabular form.

From this summary, the MDS version of the PL/M program will be reorganized to have a program module which contains the following procedures: INITIATE, CONFIGURE, CALIBRATE, OPERATE, ALLCAL, ANALYZECAL, CHANNELCFG, DOWNLINK1CAL, DOWNLINK2CAL, IOCHECK, MISSIONOP, OPERATORIN, PORT, RCVALD1, RCVCALD2, RNGDELAYBERTEST, SDAACQD1, SDAACQD2, SDACALD1, SDACALD2, STARTCAL, STARTOPERATE, and ULCAL. All other procedures should then be grouped accordingly (e.g., all input and output functions: SEND, STARIN, STARO, etc).

```
ALLCAL.................5A6DHPROCEDURE STACK=0006H,VARCCM
ANALYZECAL.............5B6AHPROCEDURE STACK=0008H,VARCCM
AP.....................3361HPROCEDURE ADDRESS PUBLIC STACK=0002H,VARMAIN,(VARIO)
APPEND.................2127HPROCEDURE EXTERNAL(15) STACK=0000H,(VARIO),(VARW1),VARW2
BUILDCFG...............543FHPROCEDURE EXTERNAL(64) STACK=0000H,(VARCCM),VARW2
BUILDERRMSG............3AF6HPROCEDURE STACK=0004H,VARIO
CALIBRATE..............5BC0HPROCEDURE EXTERNAL(8) STACK=0000I,(VARMAIN),VARCCM
CALLCI.................22CDHPROCEDURE BYTE EXTERNAL(16) STACK=0000H,(VARIO),VARW2
CALLCO.................22D5HPROCEDURE PUBLIC(71) STACK=0000H,VARMON,(VARIO),(VARW
CALLRI.................22D1HPROCEDURE BYTE EXTERNAL(68) STACK=0000H,VARW2
CHANNELCFG.............5611HPROCEDURE STACK=0002H,VARCCM
CKSUM..................2642HPROCEDURE EXTERNAL(42) STACK=0000H,(VARW1),VARW2
COMPCHECK..............325CHPROCEDURE EXTERNAL(79) STACK=0000H,(VARMON),VARW1,(VARW2)
CONCAT.................21A5HPROCEDURE EXTERNAL(60) STACK=0000H,(VARCCM),(VARMON),VARW2
COND1..................4ADBHPROCEDURE EXTERNAL(66) STACK=0000H,(VARMON),VARW1
COND2..................4A2AHPROCEDURE STACK=0004H,VARW1
CONFIGURE..............56C6HPROCEDURE EXTERNAL(7) STACK=0000H,(VARMAIN),VARCCM
CONIN..................22FEHPROCEDURE EXTERNAL(65) STACK=0000H,(VARMON),VARIO
CONOUT.................2618HPROCEDURE EXTERNAL(59) STACK=0000H,(VARCCM),(VARMON),VARIO,(VARW1),(VARW2)
CONX...................2672HPROCEDURE EXTERNAL(74) STACK=0000H,(VARMON),VARW2
DEVICE.................395AHPROCEDURE STACK=0002H,VARIO
DOWNLINK1CAL...........5A59HPROCEDURE STACK=0004H,VARCCM
DOWNLINK2CAL...........5A63HPROCEDURE STACK=0004H,VARCCM
EQUAL..................26ACHPROCEDURE BYTE EXTERNAL(73) STACK=0000H,(VARMON),(VARIO),VARW2
EQUATE..................   PROCEDURE BYTE PUBLIC ,VARW2,(VARMAIN),(VARCCM),(VARIO),(VARMON),(VARW1)
GETCHR.................22ECHPROCEDURE BYTE STACK=0002H,VARIO
GETSTS.................22DDHPROCEDURE BYTE EXTERNAL(80) STACK=0000H,(VARMON),VARIO,(VARW2)
HEXTOASCII.............41A5HPROCEDURE EXTERNAL(63) STACK=0000H,(VARMON),(VARIO),(VARW1),VARW2
INITIATE...............52D9HPROCEDURE EXTERNAL(6) STACK=0000H,(VARMAIN),VARW2
IOCHECK................31FEHPROCEDURE STACK=0002H,VARW1,(VARW2)
KEY....................40F7HPROCEDURE STACK=0004H,VARMON
LIST...................4801HPROCEDURE STACK=0004H,VARMON
LISTOUT................2569HPROCEDURE EXTERNAL(72) STACK=0000H,(VARMON),VARIO
MESAGE.................33C9HPROCEDURE STACK=0002H,VARIO
MISSIONOP..............5D91HPROCEDURE PUBLIC STACK=0004H,VARW2
MONITOR................4FE9HPROCEDURE STACK=0008H,VARMON
NOCCINPUT..............4EA0HPROCEDURE STACK=0004H,VARMON
NULL...................2059HPROCEDURE PUBLIC STACK=0000H,VARW2
OPADD..................2A44HPROCEDURE PUBLIC STACK=0000H,VARMAIN,(VARW1)
OPERATE................5DFCHPROCEDURE EXTERNAL(9) STACK=0000H,(VARMAIN),VARW2
OPERATORIN.............5D19HPROCEDURE PUBLIC STACK=0002H,VARW2
PORT...................264CHPROCEDURE EXTERNAL(40) STACK=0000H,(VARW1),VARW2
PROMPTCALDISPLAY.......3ED5HPROCEDURE STACK=0002H,VARCCM,(VARMON)
PROMPTCFGDISPLAY.......3C7EHPROCEDURE PUBLIC STACK=0002H,VARCCM,(VARMON)
PROMPTERR1MSG..........3B29HPROCEDURE EXTERNAL(68) STACK=0000H,(VARMON),VARW1
PROMPTERR2MSG..........3B5EHPROCEDURE EXTERNAL(69) STACK=0000H,(VARMON),VARIO
PROMPTERR3MSG..........3B8BHPROCEDURE STACK=000AH,VARIO
PROMPTERR4MSG..........3BDCHPROCEDURE STACK=0002H,VARW1
PROMPTERR5MSG..........3BF8HPROCEDURE EXTERNAL(67) STACK=0000H,(VARMON),VARW1
PROMPTHALTMSG..........3C1FHPROCEDURE STACK=0002H,VARW1
PROMPTOPRDISPLAY.......3F28HPROCEDURE EXTERNAL(62) STACK=0000H,(VARMON),VARW1
RCVCALD1...............593EHPROCEDURE STACK=0002H,VARCCM
RCVCALD2...............59B7HPROCEDURE STACK=0002H,VARCCM
RNGDELAYBERTEST........5AA2HPROCEDURE STACK=0002H,VARCCM
SDAACQD1...............5992HPROCEDURE STACK=0002H,VARCCM
SDAACQD2...............5A0BHPROCEDURE STACK=0002H,VARCCM
SDACALD1...............5968HPROCEDURE STACK=0002H,VARCCM
SDACALD2...............59E1HPROCEDURE STACK=0002H,VARCCM
SEGMENT................2151HPROCEDURE PUBLIC STACK=0006H,VARW2
SEGMSG.................42CFHPROCEDURE STACK=0002H,VARMON
SEGOPMSG...............4388HPROCEDURE STACK=0004H,VARMON
SEND...................4700HPROCEDURE EXTERNAL(63) STACK=0000H,(VARCCM),(VARMON),VARIO,(VARW1),(VARW2)
SENDAVAIL..............47C5HPROCEDURE EXTERNAL(4) STACK=0000H,(VARMAIN),VARMON
SENDSTATUS.............4E3CHPROCEDURE STACK=0006H,VARMON
SIMULATE...............4BCFHPROCEDURE STACK=0006H,VARMON
STACK..................51BFHPROCEDURE EXTERNAL(62) STACK=0000H,(VARCCM),VARMON,(VARW2)
STARI..................21EEHPROCEDURE STACK=0002H,VARMON
STARIN.................2202HPROCEDURE STACK=0004H,VARMON
STARINIT...............22C3HPROCEDURE EXTERNAL(1) STACK=0000H,(VARMAIN),VARW2
STARO..................21F8HPROCEDURE EXTERNAL(59) STACK=0000H,(VARMON),VARW2
STAROUT................226AHPROCEDURE STACK=0002H,VARMON,(VARIO)
STARTCAL...............5794HPROCEDURE STACK=0006H,VARCCM
STARTOPERATE...........5BEBHPROCEDURE PUBLIC STACK=000AH,VARW2
STATUSSENDLIST.........48ACHPROCEDURE STACK=0002H,VARW1
TABLEINIT..............2656HPROCEDURE EXTERNAL(2) STACK=0000H,(VARMAIN),VARW1
TABLESEARCH............265AHPROCEDURE ADDRESS EXTERNAL(81) STACK=000,(VARMON),VARW1
TERMINATE..............5E57HPROCEDURE EXTERNAL(10) STACK=0000H,VARMAIN
TIMEX..................26A8HPROCEDURE EXTERNAL(47) STACK=0000H,(VARW1),VARW2
TOD....................239DHPROCEDURE EXTERNAL(78) STACK=0000H,(VARMON),VARIO
TTY1...................2565HPROCEDURE BYTE STACK=0002H,VARIO
TTYO...................255DHPROCEDURE STACK=0002H,VARIO
ULCAL..................5A30HPROCEDURE STACK=0002H,VARCCM
WAIT...................2009HPROCEDURE EXTERNAL(5) STACK=0000H,(VARMAIN),(VARMON),VARW2
XCONOUT................2500HPROCEDURE STACK=0006H,VARIO
XENABLE................4583HPROCEDURE EXTERNAL(22) STACK=0000H,(VARIO),VARW1
XKEY...................40F3HPROCEDURE EXTERNAL(0) STACK=0000H,(VARMAIN),VARMON
XXCON..................2486HPROCEDURE EXTERNAL(77) STACK=0000H,(VARMON),VARIO
```

Fig. 1. Sorted procedure table

```
PROGRAM: *ORGNIZE

100 INPUT USING 'ENTER PROCEDURE NAME:#':P$
110 OPEN 'MBAOPS*MBAOPS.PROCTBL',INPUT,1\'*QQ',OUTPUT,2
120 AT ENDFILE(1) GO TO 150
130 STRING PROCTB(100):132,NUMB:16,DUMMY:132
140 INPUT FROM 1 USING '(R)':PROCTB
150 STRING PROCTB(NTB):132,PROC:132,RECD:132,STACK(NS):30 WHERE NTB=ITEMS,&
          NS=1,NE=0
160 CLOSE 1
165 STRING EXIST(1):30 WHERE NE=1&

170 FOR I=1 UNTIL DONE WHERE DONE=0,PROC=P$,THERE=0
171 THERE=0\THERE=(PROC=EXIST(PP)) FOR PP=1 UNTIL THERE OR PP>NE IF I#1
172 GO TO 230 IF THERE
175    GO TO 230 IF LEFT(PROC,3)='ROM' !SKIP ALL ROMS ROUTINES
180   GOSUB 1000 !SEARCH PROCTB TO FIND P$
190   IF FIND=0 THEN GO TO 250 WHERE DONE=1
195    RECD=PROCTB((F+L)/2)
200   GOSUB 2000 !BRING IN THE P$ LISTING
210     GOSUB 3000 ! WRITE OUT THE LISTING ONTO *QQ
220     GOSUB 4000 ! FIND ALL CALLS IN THIS LISTING AND APPEND IT TO STACK
230     IF I<=NS THEN PROC=STACK(I) ELSE&
           DONE=1  ! WHERE STACK IS EMPTY
235 PRINT 'I=':I;'NS=':NS;'NE=':NE IF I MOD 10=0
240     NEXT I
245 CLOSE 1, 2
250 END 'DONE'&
&

1000 !BINARY SEARCH PROCEDURE
1010 NUMB=PROC,NUMB=RJUST(NUMB),PP=LENGTH(NUMB),NUMB=NUMB+RPT('.',16-PP),&
      F=1,L=NTB
1020 IF F>L THEN FIND=0 ELSE&
        IF NUMB=LEFT(PROCTB((F+L)/2),16) THEN FIND=1 ELSE&
          IF NUMB<LEFT(PROCTB((F+L)/2),16) THEN&
            GOSUB 1020 WHERE L=IP((F+L)/2)-1 ELSE&
            GOSUB 1020 WHERE F=IP((F+L)/2)+1
1030 RETURN&

2000 ! FIND MODULE AND BRING IT IN
2010 REAL POS(COMAS)=0 WHERE COMAS=COUNT(RECD,',')
2020 POSCMA=0\POS(K)=INDEX(RECD,',',POSCMA),POSCMA=POS(K)+1 FOR K=1 TO COMAS
2030 FLAG=0\FLAG=(SUBSTR(RECD,POS(K),2)#',(') AND (SUBSTR(RECD,POS(K),1)=',')&
        FOR K=1 UNTIL FLAG OR K>COMAS\K=K-1
2040 IF K>=COMAS THEN FL$=SUBSTR(RECD,POS(K)+1) ELSE&
          FL$=SUBSTR(RECD,POS(K)+1,POS(K+1)-POS(K)-1)
2045 FL$=SUBSTR(FL$,4)
2046 THERE=0\THERE=(PROC=EXIST(PP)) FOR PP=1 UNTIL THERE OR PP>NE
2047 GO TO 2180 IF THERE
2050 OPEN FL$,INPUT,1
2060 AT ENDFILE(1) GO TO 2097
2070 FOR K=1 UNTIL FIND WHERE FIND=0
2080  INPUT FROM 1 USING '(R)':DUMMY
```

Fig. 2. Program listing

```
*ORGNIZE


2085 IF INDEX(DUMMY,'PROCEDURE') AND INDEX(DUMMY,'$') THEN&
       GO TO 2085 WHERE POS1=INDEX(DUMMY,'$'),DUMMY=LEFT(DUMMY,POS1-1)+&
            SUBSTR(DUMMY,POS1+1)
2090  FIND=(INDEX(' '+DUMMY,' '+PROC) AND INDEX(DUMMY,'PROCEDURE') AND&
           INDEX(DUMMY,'EXTERNAL')=0)
2095 GO TO 2100
2097  FIND=1
2100  NEXT K
2110 STRING LISTN(N):132:/DUMMY/ WHERE N=1
2115 AT ENDFILE(1) GO TO 2160
2120 FOR K=1 UNTIL FEND WHERE FEND=0
2130   INPUT FROM 1 USING '(R)':DUMMY
2135 IF INDEX(DUMMY,'END ') AND INDEX(DUMMY,'$') THEN&
       GO TO 2135 WHERE POS1=INDEX(DUMMY,'$'),DUMMY=LEFT(DUMMY,POS1-1)+&
            SUBSTR(DUMMY,POS1+1)
2140   IF INDEX(' '+DUMMY,' END ') AND INDEX(DUMMY,' '+PROC) THEN&
          STRING LISTN(N):132:/LISTN,DUMMY /WHERE FEND=1 ,N=N+1ELSE&
          STRING LISTN(N):132:/LISTN,DUMMY/ WHERE N=N+1
2150      GO TO 2170
2160     FEND=1
2170     NEXT K
2175 CLOSE 1
2180 RETURN&
&

3000 ! WRITE IT OUT IF NOT ALREADY THERE
3010 NOK=0\NOK=(EXIST(K)=PROC) FOR K=1 UNTIL NOK=1 OR K>NE
3020 IF NOK#0 THEN GO TO 3050
3030 WRITE ON 2 USING '(R)':CHAR(13)\LISTN\CHAR(13)
3040 IF EXIST(1)=NULL THEN&
          STRING EXIST(NE):30:/P$/ ELSE&
          STRING EXIST(NE):30:/EXIST,PROC/ WHERE NE=NE+1
3050 RETURN&

4000 ! FIND ALL CALLS IN LISTN AND APPEND IT ON STACK
4010 FOR K=1 TO N
4020    POS1=INDEX(LISTN(K),' CALL ')
4030   IF POS1=0 THEN GO TO 4100
4040   DUM$=SUBSTR(LISTN(K),POS1+5),DUM$=LJUST(DUM$)
4050   DUM$=DUM$+' ',POS1,FLAG=0\A$=SUBSTR(DUM$,J,1),FLAG=(INDEX('; (',A$)),&
         POS1=J FOR J=1 UNTIL FLAG OR J>=LENGTH(DUM$)\DUM$=LEFT(DUM$,POS1-1),
    NUM$=LJUST(RJUST(DUM$))
4070   IF INDEX(DUM$,'$') THEN&
          GO TO 4070 WHERE POS1=INDEX(DUM$,'$'),DUM$=LEFT(DUM$,POS1-1)+&
            SUBSTR(DUM$,POS1+1)
4080      IF STACK(1)=NULL THEN&
           STRING STACK(NS):30:/DUM$/ WHERE NS=1 ELSE&
           STRING STACK(NS):30:/STACK,DUM$/ WHERE NS=NS+1
4100   NEXT K
4200 RETURN&
```

Fig. 2 (contd)

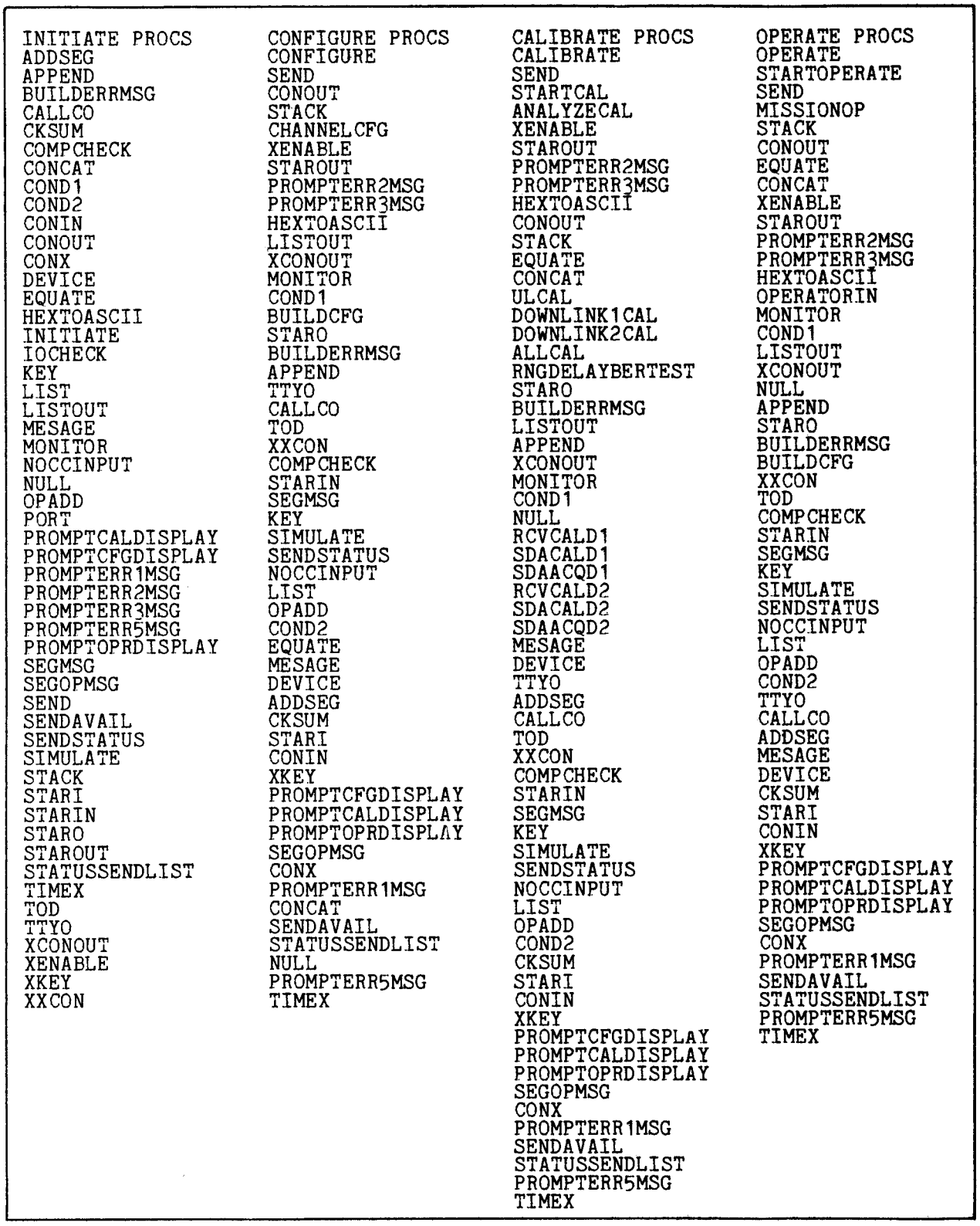

Fig. 3. Typical output

PROCEDURES	INITIATE	CONFIGURE	CALIBRATE	OPERATE
ADDSEG	X	X	X	X
ALLCAL			X	
ANALYZECAL			X	
APPEND	X	X	X	X
BUILDCFG		X		X
BUILDERRMSG	X	X	X	X
CALIBRATE			X	
CALLCO	X	X	X	X
CHANNELCFG		X		
CKSUM	X	X	X	X
COMPCHECK	X	X	X	X
CONCAT	X	X	X	X
COND1	X	X	X	X
COND2	X	X	X	X
CONFIGURE		X		
CONIN	X	X	X	X
CONOUT	X	X	X	X
CONX	X	X	X	X
DEVICE	X	X	X	X
DOWNLINK1CAL			X	
DOWNLINK2CAL			X	
EQUATE	X	X	X	X
HEXTOASCII	X	X	X	X
INITIATE	X			
IOCHECK	X			
KEY	X	X	X	X
LIST	X	X	X	X
LISTOUT	X	X	X	X
MESAGE	X	X	X	X
MISSIONOP				X
MONITOR	X	X	X	X
NOCCINPUT	X	X	X	X
NULL	X	X	X	X
OPADD	X	X	X	X
OPERATE				X
OPERATORIN				X
PORT	X			
PROMPTCALDISPLAY	X	X	X	X
PROMPTCFGDISPLAY	X	X	X	X
PROMPTERR1MSG	X	X	X	X
PROMPTERR2MSG	X	X	X	X
PROMPTERR3MSG	X	X	X	X
PROMPTERR5MSG	X	X	X	X
PROMPTOPRDISPLAY	X	X	X	X
RCVCALD1			X	
RCVCALD2			X	
RNGDELAYBERTEST			X	
SDAACQD1			X	
SDAACQD2			X	
SDACALD1			X	
SDACALD2			X	
SEGMSG	X	X	X	X
SEGOPMSG	X	X	X	X
SEND	X	X	X	X
SENDAVAIL	X	X	X	X
SENDSTATUS	X	X	X	X
SIMULATE	X	X	X	X
STACK	X	X	X	X
STARI	X	X	X	X
STARIN	X	X	X	X
STARO	X	X	X	X
STAROUT	X	X	X	X
STARTCAL			X	
STARTOPERATE				X
STATUSSENDLIST	X	X	X	X
TIMEX	X	X	X	X
TOD	X	X	X	X
TTYO	X	X	X	X
ULCAL			X	
XCONOUT	X	X	X	X
XENABLE	X	X	X	X
XKEY	X	X	X	X
XXCON	X	X	X	X

Fig. 4. Summary table