MEMORANDUM
RM-5437-PR
AUGUST 1967

JOSS: ASSEMBLY LISTING OF THE SUPERVISOR

G. E. Bryan

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-5437-PR
AUGUST 1967

JOSS: ASSEMBLY LISTING OF THE SUPERVISOR

G. E. Bryan

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

Published by The RAND Corporation

PREFACE

JOSS† is a multiuser, single-server computing system that provides for the solution of numerical problems. The system consists of a central computer containing the JOSS program and a number of typewriter consoles connected to the computer via telephone lines. The central computer turns its attention rapidly from console to console in such a way that individual users appear to have exclusive use of the system.

The JOSS supervisor is the unit that exercises overall control of the system's operation: Acting as a scheduling, resource-allocating, and synchronizing device, it ensures that all data and hardware necessary for a particular action are simultaneously available; it also meters the operation of the system to provide revenue accounting information and data describing system performance and user operations. This memorandum, an exact reproduction of the assembly listing of the JOSS supervisor, is a companion piece to RM-5216-PR, JOSS: User Scheduling and Resource Allocation[1] and RM-5217-PR, JOSS: Accounting and Performance Measurement,[2] which describe in detail the operation of the supervisor.

The division of the program listing into two parts represents two assemblies: Part I is the code for the core-resident JOSS supervisor, and Part II the code for system initialization and catastrophic error recovery routines. (The latter is required only at system start-up or when serious machine errors occur; during regular JOSS operation, it is overlaid by user information.) The code is written in MACRO-6, the machine assembly language for the Digital Equipment Corporation PDP-6, on which the JOSS system is implemented. (The PDP-6 instruction set and the assembly language are described in DEC's Handbook F-65[3] and Programming Manual: MACRO-6 Assembly Language.[4])

The Index to the Program Listing enumerates the routines and data storage of the program in order of occurrence, with headings and indentations suggesting the structure of the code. Because of the limited

---

†JOSS is the trademark and service mark of The RAND Corporation for its computer program and services using that program.

ability of MACRO-6 to provide for page titling and subtitling, the heading structure of the listing itself does not always correspond to the indentations of the Index. Page numbers given in the Index refer to the pagination of this memorandum and not to those printed out by the assembler in the heading line.

This work is part of The RAND Corporation's continuing research program in computer theory and applications, particularly the implementation of man-machine systems, under U.S. Air Force Project RAND. It should be useful to those with an interest in the field of time-shared interactive terminal systems.

SUMMARY

The code listed in this memorandum provides those functions in the JOSS system that are generally described by the terms monitor, supervisor, or executive. In particular, the routines perform the following tasks:

1. *Machine interrupt handling.* These programs provide appropriate responses to hardware interrupts from the drum system, the magnetic tape unit, the master console teletype, the real-time clock, programmed operators, and machine-detected hardware errors. (Interrupts from the user consoles and the disc file are handled by the console service and the disc service routines, respectively.)

2. *Supervision, resource allocation, and scheduling.* These routines provide overall supervision of the flow of work through the JOSS system; they allocate central processor time, core space, and input/output devices to the user, and schedule the use of these resources effectively to maintain high-speed response.

3. *Monitoring.* These are various routines that maintain accounting records for allocating usage charges, that meter the overall system activity, and that record the details of individual user behavior.

4. *User log-on and log-off.* These routines provide for initial user access to the system (by recording initials, project number, and department number) and produce final charge accounting records when the user leaves the system.

5. *System initialization.* In these routines, used only at system start-up, time of day and current date are requested and recorded, and a copy of the system is saved on the magnetic drum in case system recovery is required.

6. *Machine error recovery.* These routines provide for recovery from catastrophic machine failure by attempting to reload the system from the copy saved at initialization time. To aid in failure diagnosis, a message printed at the master console indicates the immediate symptom of the failure and records the contents of certain important registers.

CONTENTS

PREFACE iii

SUMMARY v

INDEX TO THE PROGRAM LISTING ix

PART I. JOSS SUPERVISOR 1

PART II. PREAMBLE AND RECOVERY 153

REFERENCES 175

JOSS BIBLIOGRAPHY 177

INDEX TO THE PROGRAM LISTING

PART I. JOSS SUPERVISOR

Parameters

Supervisor Assembly Parameters 4
System Parameters 4
Assembly Parameters for the PI Channels 4
Register Assignments 5
Assignments for Pseudo Data Switches 6
Macro and Operation Definitions 7
Interns and Externs 8

Tables, Lists, Queues, and Counters

Supervisor Storage 9
System Storage 9
State Queue Entry Words 9
Station Status Cells 10
Core Map 10
Assignments for I/O Buffers 11
Monitor Switches 12
User State Queue Headers 13
State Action Table for Interpreter Entry 14
Log-on Scanner Storage 15
Counters for Accounting, Statistics, and Debugging 16
Header Line for Console TTY Output 18
Storage for Summary Data 19
Per User Counters 21
Field Size, Scale Factor, and Pointer for Counters 22

I/O Interrupt Routines

Data Control Interrupt Routine 24
Drum Channel Interrupt Routines 25
Write End Routine 25

Interrupt End Sequence 26
Read End Interrupt 27
Initiate Next Write 28
Drum Error Recovery 29
Checksum a User's Block 31
Initiate Transfer to Drum 32
Data for Drum Transfers 33
Initiate Transfer from Drum to Core 34
Tape Interrupt Routines 35
Mag Tape Output Start Routine 37
Tape Restart Routine 37
Console TTY Interrupt Routines 38
Start Routine for Console TTY Output 38
Input Interrupt from Console TTY 39
User Operation (UUO) Interpreter 40
Handle Machine Failure Halts--Try To Recover 41
Halt Routine 41
Attempt Recovery from the Drum 41
Processor Interrupt Routines 42
Channel 7 Interrupts 42
Detect Indirect Address Loops 44
Constants 45
Processor Error Recovery 46
Discard a Bad User Block 46
Output a GRONK Message 47
Convert Routines 49

Interpreter Entry Processing

Monitor-Interpreter Interface 50
Decode Interpreter Signals 50
Interpreter Signal Types 50
Return to Interpreter 50
Interpreter-Monitor Entry Routines 51
Computation Break 51
Transmit Line to User 51

Switch Console to User 51
Wait for Signals and Time 52
Delay User for (E) Seconds 52
Get a Buffer 53
Return a Buffer 53
Final "Off" for User 54
Send Page Heading 55
Make Up Page Heading Message 56
Disc Entry Routines 57
Disc Action Routines 57
Disc Restart Routine 57
Produce Disc Accounting Record 58
Conversion Routine--Binary to Base Ten ASCII 59
Return Blocks of Core 60
Provide More Core for User's Block 61
Imperative Request for More Core 61
Find a Free Block of Core 62
Find a Free Block of the Required Size 63
Find a Free Block Below User 63
Find a Free Block Above User 64
Move User to Free Area 65
Move Down Multiple Users 65
Move User(s) Up 66
Reset User Locations 67
Block Transfer Core 68
Set Relocation Register 68
Compact Core 69
Do One Compact 69
Find a Core Move 70

Swap Selection

Select a User for Drum Swap 71
Find a User To Bring In 71
Find a Low Priority User of the Proper Size 72
Select Multiple Users for Transfer to Drum 73

Compact Core for Swap 75
Search User States 76

Main Processing Loop Routines

Miscellaneous MPL Functions 77
Force Users Off 77
Process Users Waiting for Buffers 78
Process the Pause Queue 78
Process Disc Interrupt Signal 79
Search for GRONKed Users 79
Inter-console Synchronizing Signals 80
Get Signal Table Index 80
Reset a Signal Bit 80
Test a Signal Bit 80
Set a Signal Bit 81
Table of Inter-console Signals 81
Disc Accounting Routines 82
Midnight Disc Skulker 82
Start or Restart Skulk 82
Disc Action Complete 83
Shutdown Procedure 84

Console Signal Processing

Monitor-Distributor Interface 85
Process Distributor Signals 85
Tables for Signal Interpretation 86
Send Messages to All in the Queue 87
Off-signal Routines 88
On-signal Routines 89
Send a Queue Message 90
Convert to Two Decimal Characters 91
Compute Binary Seconds from Clock Cells 91
In-signal Routines 92
Carrier-return-signal Routines 93

Convert to Linear Range 93
Transmission-over-signal Routines 94
Switch Console to Green State 95

Accounting and Statistics Routines

Accumulate Counts on the Minute 96
Display User's Initials and Station Number 97
Display Initials for On and Off Signals 98
Convert Station Number to Octal ASCII 98
Display Statistical Distributions 99
Count Time and Gather Statistics 100
New Minute 100
New Hour 101
Tape Output Routine 102
Output Cumulative Statistics Line 103
Output Hourly Heading Line 104
Output Detail Statistics on the Minute 105
Routine To Update the Date 106
Convert Date and Time to ASCII 106
Format a Counter for Console Output 107

Main Processing Loop

Monitor Main Processing Loop 108
Master Monitor Entry Point 108
Process the Queue for Transfer to Drum 108
Select an In-core User for Interpretation 109
Monitor Idle Loop 109
Process an "On" User 110
Initialize for Interpretation 111

Log-on and Log-off Processors

JOSS Log-on Processor 112
Routine To Approve Initials 115
Scan Table for Approval of Initials 116

Routine To Approve Job Number 117
Tables of RPNs 118
Scan Table for Job Number Approval 119
Routine To Approve Department Name or Number 120
Scan Table for Department Approval 121
Department Name-Number Table 122
String Scanner 123
Character To Type Conversion Table 124
JOSS Log-off Processor 125

Service Routines

Change User State 127
Output Buffer to User 128
Get a Buffer from Available List 129
Return a Buffer to Available List 130
Send a Message to the Console TTY 131
Stuff Message in Buffer 132
Log-on and Shutdown Messages 133

Literal Table 135
Symbol Table 136

PART II. PREAMBLE AND RECOVERY

Preamble

Assembly Parameters 156
Interns and Externs 157
Dump Core on Tape 158
Routine To Input Date and Time 159
Ask about DDT and Allocate User Blocks 161
Write and Read Console TTY 162

Recovery

Recovery and Error Dump Routine 163
Clear Machine Status 164

Output Error Message and Panel Dump 165
Output Formatting Routines 166
Restore Date, Time, Etc. 167
Low Memory Contents 168
Low Memory Cells 169
Read and Write JOSS on Drum 170
JOSS Initialization 171

Symbol Table 172

PART I

JOSS SUPERVISOR

```
00040     ;                JOSS SUPERVISORY UNIT
00050
00060     ;                    .G. E. BRYAN
00070
00080
00090     ;          THIS CODE PROVIDES FOR THE JOSS SYSTEM THOSE FUNCTIONS
00100     ;       THAT ARE GENERALLY DESCRIBED BY THE TERMS EXECUTIVE, MONITOR,
00110     ;       OR SYSTEM SUPERVISOR. IN PARTICULAR, THESE ROUTINES PRO-
00120     ;       VIDE FOR HANDLING OF I/O INTERRUPTS (FROM DRUM, TAPE, CONSOLE
00130     ;       TTY, AND CENTRAL PROCESSOR), RECOVERY FROM MACHINE
00140     ;       ERROR, SCHEDULING OF PROCESSOR TIME AND CORE SPACE, MONITOR-
00150     ;       ING OF USER AND SYSTEM PERFORMANCE, AND THE LOG-ON AND LOG-
00160     ;       OFF PROCESSES.
00170
00180
00190     ;       ASSEMBLED 8/1/67 FROM TAPE 56 (A SEQUENCED COPY OF 31)
00200     ;       BINARIES ARE ON TAPE 24, CALLED SU.
00210
00220     PATCH:  BLOCK   20
00230
```

```
00250     ;        SYSTEM PARAMETERS
00260
00270     N.PP1=20;        LENGTH OF THE PUSH-POP LINKAGE LIST
00280     N.S=↑D40;        ASSEMBLED NUMBER OF STATIONS
00290     S.M=2;           NUMBER OF 32 BIT GROUPS FOR SIGNAL TABLES
00300     ;                (FP (N.S/32) =0:N.S/32;IP (N.S/32) +1)
00310     N.SG=N.S;        LENGTH OF THE SIGNAL TABLE
00320     BUFSIZ=↑D18;     BUFFER SIZE
00330     N.COR=↑D16;                      # OF CORE BLOCKS AVAILABLE TO USERS
00340     BBLOCK=↑D16;                     BEGINNING USER BLOCK
00350     DELTA=4
00360     K2=N.S+DELTA     NUMBER OF BUFFERS ASSEMBLED
00370     K1=↑D60;                           THE NUMBER OF TICKS PER SECOND.
00380     K3=↑D15;                         DISPLAY INITIALS IF FP(MIN/K3) =0
00390     SA=112
00400
00410     ;        ASSEMBLY PARAMETERS FOR THE PI CHANNELS
00420
00430     CDC=1;           1 - DATA CONTROL
00440     CDRM=2;          2 - I/O PROCESSOR AND DRUM CHANNEL
00450     CI630=3;         3 - 630 INPUT
00460     CO630=4;         4 - 630 OUTPUT
00470     CDAT=6;          INTERRUPT CHANNEL FOR TAPE
00480     CCTY=6;          6 - CONSOLE TELETYPE
00490     CAPR=7;          7 - PROCESSOR INTERRUPTS
00500
00510     CHDC=100;        THESE ARE CORRESPONDING DEFINITIONS FOR PI CONOS
00520     CHDRM=40;
00530     CH630=30
00540     CHDAT=0
00550     CHCTY=2
00560     CHAPR=1
00570     MTC=220;                         DEVICE CODES FOR MAG TAPE REGISTERS
00580     MT1=224
00590     MT2=230
00600     DR=400
00610     DP=010
00620     DMT=20;          DATA CONTROL DEVICE NUMBER FOR MAG TAPE
```

```
00640     ,        REGISTER ASSIGNMENTS
00650
00660     A=0
00670     B=1
00680     C=2
00690     D=3
00700     E=4
00710     F=5
00720     G=6
00730     H=7
00740     I=10
00750     J=11
00760     K=12;          USED BY DISTRIBUTOR
00770     L=13;          USED BY DISTRIBUTOR
00780     M=14;          USED BY DISTRIBUTOR
00790     N=15;          USED BY DISTRIBUTOR
00800     S=16;          USED FOR STATION INDEX
00810     PP=17;         PUSH-POP REGISTER
```

ASSIGNMENTS FOR PSEUDO DATA SWITCHES

```
00830     ,       PSEUDO DATA SWITCHES (IN SWITCH)
00840
00850     ,       RIGHT HALF
00860
00870     SDS=200000;      SHUTDOWN SWITCH
00880     BEEPS=400000;    BEEP SWITCH
00890     OFFS=100000;     BLAST OFF SWITCH
00900     DCLOB=40000;     DISC NOT AVAILABLE
00910     HALTS=20000;     HALT ON ERROR SWITCH
00920     OLSW=100;        OUTPUT STAT LINE (NUMBER IN LEFT HALF)
00930     PRMES=40;        ADMINISTRATIVE MESSAGE HAS PRIORITY
00940     DSTAT=20;        DISPLAY STATISTICAL DISTRIBUTIONS
00950     DOAF=10;         DISPLAY ONS AND OFFS
00960     TPQ=4;           DISABLE MAG TAPE OUTPUT
00970     DUI=2;           DISPLAY USERS INITIALS
00980     DAM=1;           DISPLAY ADMINISTRATIVE MESSAGE
00990
01000     ;       LEFT HALF
01010
01020     ;                RIGHT 6 BITS ARE STAT LINE NUMBER FOR DISPLAY
```

```
01040           DEFINE FSW  (A)
01050     <MOVE A,SWITCH
01060     >
01070           DEFINE XMT  (M,A) ; TRANSMIT M CHARACTERS FROM A
01080     <OPDEF X [M'B12]
01090     X A>
01100           DEFINE ZBLOK  (N)
01110     <REPEAT N,
01120     <Z
01130     XLIST>
01140     LIST>
01150
01160           DEFINE QCT (Q,C) ; MACRO TO COUNT QUEUES
01170     <
01180     HRRZ B,Q
01190     TSX QCTR
01200     MOVEM A,C>
01210
01220           OPDEF   INS [10B8]
01230           OPDEF   CHS [20B8]
01240           OPDEF   SEND [30B8]
01250           OPDEF   HALT [40B8]
01260           OPDEF   TALLY [AOS  0]
01270           OPDEF   NOP [JFCL]
01280           OPDEF   DONE [POPJ  PP,0]
01290           OPDEF   TSX [PUSHJ  PP,0]
01300           OPDEF   CALL [PUSHJ  PP,0]
```

```
01320          EXTERN  C20,C27,C28,T80,T80.99
01330          EXTERN  FILE,KEY,PROG
01340          EXTERN  DBUF
01350          EXTERN  DISC.D,ACTION,RESULT
01360          EXTERN  C30,C31,DISC,DISC.C,DISC.S
01370          EXTERN  SHUT
01380          EXTERN  INTENT,INTBEG,SEQ,RISIG,INITIALS,JOBNO
01390          EXTERN  PAGNO,ONTIME,COMTIM
01400          EXTERN  USIZE
01410          EXTERN  SPARE1,SPARE2,CONSOL
01420          EXTERN  SPARE3,SPARE4
01430          EXTERN  DDT,D.TIME,DUMP,DATIME,RJD,WJD
01440          EXTERN  INTENT;          LOC. IN USER AREA TO START THE PROG.
01450          EXTERN  SWITCH,FAKE
01460          EXTERN  C32
01470
01480
01490          INTERN  DATE,HR,MIN,TIME
01500          INTERN  CTD,CTDR,CTDB
01510          INTERN  CT24,CT17,CT23,CT37
01520          INTERN  CT14,CT19,CT18
01530          INTERN  CT15,CT22
01540          INTERN  CT27A,CT27B,CT46,CT47
01550          INTERN CT22A,CT18A,CT18B,TERASE,S.S,N.CB
01560          INTERN YEAR,MONTH,DAY,SEC,CT22A
01570          INTERN  S.OK
01580          INTERN  N.SON,APRR,PROP
01590          INTERN  CT11,CT13,CT16,CT28,CT31
01600          INTERN  DRMR,S.QUE,CMESS,ADATE,OCTW
01610          INTERN  T8,KILL
01620          INTERN  S.BUF,COMEBACK,N.S,SIGTBL,SG.L,SG.M,SG.LIM
01630          INTERN  USERS,MONENT,T7.9
01640          INTERN  CORE,SECONDS,CTYR
01650          INTERN  HMES,PPW,TYPE6,SOUT,SIGPR
01660          INTERN  SSIG,RSIG,TSIG
01670          INTERN  T9,T10,T11,T11.1,T11.2,T11.3,T11.4,T11.5
01680          INTERN  T11.6,T11.7,T11.8,T11.9
```

```
                                01700     ;        SYSTEM STORAGE
                                01710
000020  000000 000000           01720     PPSAV:   Z;                          MONITOR PUSH-POP REGISTER SAVE
000021  000000 000000           01730     CUI:     Z;                          CURRENT USER # IN RIGHT HALF
000022  000011 000006           01740     TYPE6:   XWD      ↑D9,6;             CONTROL WORD FOR INITIALIZATION RECORD
000023  305366 133536           01750     DATE:    ASCII    $1/17/67    $
000024  331564 020100
000025  311467 232556           01760     ATIM:    ASCII    $23:57$
000026  000000 000000           01770     YEAR:    DEC      0
000027  000000 000001           01780     MONTH:   DEC      1
000030  000000 000021           01790     DAY:     DEC      17
000031  000000 000027           01800     HR:      DEC      23
000032  000000 000071           01810     MIN:     DEC      57
000033  000000 000000           01820     SEC:     Z;                          SECONDS IN THIS MINUTE
000034  000000 000000           01830     TICK:    Z;                          TICKS IN THIS SECOND
000035  000000 000000           01840     CL9:     Z;                          REG C - PROCESSOR INTERRUPT ROUTINE
000036  000000 000000           01850     CL10:    Z;                          REG D
000037  000000 000000           01860     CL11:    Z;                          INDIRECT ADDRESS CHAIN COUNT
000040  000000 000000           01870     CL12:    Z;                          LOCATION COUNTER
000041  000000 000000           01880     CL13:    Z;                          COUNT OF INTERRUPTS AT SAME LOCATION
000042  000000 000000           01890     TIME:    Z;                          CURRENT TIME IN TICKS. (16 2/3MS=1/60 SEC)
000043  000000 000000           01900     T.CU:    Z;                          TIME USED BY CURRENT USER THIS SHOT.
000044  000000 000000           01910     SECONDS:Z;       TICKS--CONTINUOUS COUNT FOR TIMER COMPUTATION
                                01920
                                01930     SIGTBL: REPEAT   N.SG,;         DISTRIBUTOR-MONITOR  SIGNAL LIST
                                01940     <Z
                                01950     XLIST>
000045  000000 000000            Z
                                XLIST
000115  000000 000000           01970     SG.L:     Z;                         CURRENT NUMBER OF ENTRIES IN SIGTBL
000116  000000 000050           01980     SG.M:     DEC      N.SG;             MAX # ENTRIES IN SIGTBL
000117  000000 000005           01990     SG.LIM:  DEC      5;                SIGTBL ENTRIES BEFORE COMEBACK (EXCEPT CR)
                                02000
000120  777760 000120'          02010     PPW:     XWD      -N.PP1,PP1-1
                                02020     PP1:     REPEAT   N.PP1,;         PUSH-POP LINKAGE LIST
                                02030     <Z
                                02040     XLIST>
000121  000000 000000           Z
                                XLIST
                                02060     ;        STATE QUEUE ENTRY WORDS
                                02070
                                02080     S.Q:     REPEAT   N.S-1,
                                02090     <XWD 0,.+1
                                02100     XLIST>
000141  000000 000142'          XWD 0,.+1
                                XLIST
000210  000000 000000           02120              Z;                          LAST QUEUE ENTRY WORD
```

```
                         02140
                         02150      ,   STATION STATUS CELLS
                         02160
                         02170      S.S:     REPEAT  N.S,
                         02180      <BYTE (6) OF.S,0,0,1,0,0
                         02190      XLIST>
000211  230000 010000    BYTE (6) OF.S,0,0,1,0,0
                         XLIST
000261  360616 000211'   02210      S.STA:   POINT 6,S.S(S),5;        CURRENT STATE     (5 BITS)
000262  300616 000211'   02220      S.TM:    POINT 6,S.S(S),↑D11;     TIME IN SHOT      (4 BITS)
000263  220616 000211'   02230      S.COR:   POINT 6,S.S(S),↑D17;     CORE LOCATION    (5 BITS)
000264  140616 000211'   02240      S.BLOCK:POINT 6,S.S(S),↑D23;      NUMBER OF BLOCKS (2 BITS)
000265  130116 000211'   02250      S.OFR:   POINT   1,S.S(S),↑D24;   OFF SIGNAL RECEIVED
000266  120116 000211'   02260      S.INR:   POINT   1,S.S(S),↑D25;   IN SIGNAL RECEIVED
000267  110116 000211'   02270      S.DU:    POINT   1,S.S(S),↑D26;   USING DISC
000270  100116 000211'   02280      S.GK:    POINT   1,S.S(S),↑D27;     SET WHEN USER IS GRONKED
000271  070116 000211'   02290      S.SIG:   POINT   1,S.S(S),↑D28;   SET WHEN WAITING FOR SIGNALS
                         02300
                         02310      ,    CORE MAP
                         02320
                         02330      CORE:    ZBLOK N.COR; THE CORE MAP
                           REPEAT N.COR,
000272  000000 000000    Z
000312  000000 000004    02340      N.CB:    DEC     4;                MAX # OF CORE BLOCKS PER USER
000313  000000 000020    02350      N.C:     DEC     N.COR;            # OF ACTIVE CORE BLOCKS (HIGHEST # NOT TO USE)
000314  440101 000272'   02360      S.IU:    POINT   1,CORE(B),0;      USE BIT FOR BLOCK
000315  420101 000272'   02370      S.ID:    POINT   1,CORE(B),1;      IN USE BY DRUM
000316  420201 000272'   02380      S.UD:    POINT   2,CORE(B),1;      BOTH ACTIVITY BITS
000317  330601 000272'   02390      S.UR:    POINT   6,CORE(B),↑D8;    USER NUMBER FOR THE BLOCK
                         02400
                         02410      ,        SYSTEM PARAMETERS
                         02420
000320  000000 000000    02430      N.DRM:   Z;                        NUMBER OF USERS ON DRUM
000321  000000 000020    02440      DBASE:   OCT     20;               BASE TRACK FOR USER BLOCKS
000322  000000 000001    02450      N.UC:    DEC     1;                START COMPUTING IF ONLY THIS MANY BUFFERS
000323  000000 000002    02460      N.CK:    DEC     2;                QUIT COMPUTING IF THIS MANY BUFFERS FILLED
000324  000000 000050    02470      N.SON:   DEC     40;               MAX # OF USERS BEFORE QUEUE MESSAGE
000325  000000 000047    02480      S.OK:    OCT     47;               IGNORE SIGNALS FROM HIGHER STATIONS
000326  000000 000014    02490      T.MAX:   DEC     12;               MAXIMUM SHOT TIME ... 200 MS.
```

ASSIGNMENTS FOR I/O BUFFERS

```
000327  000000 000000    02510     DT.BUF:Z;                          HEADER FOR DISC AND TAPE I/O
000330  000000 000000    02520     CO.BUF: Z;                         HEADER FOR CONSOLE TTY OUTPUT
                         02530     S.BUF:  ZBLOK N.S;  BUFFER HEADERS: RIGHT = FIRST; LEFT = LAST
                           REPEAT N.S,
000331  000000 000000    Z
000401  000000 000054    02540     N.BUF:  DEC     K2;                NUMBER OF AVAILABLE BUFFERS.
000402  002064'000403'   02550     L.BUF:  XWD     LASTB,BUF;         HEAD OF AVAILABLE BUFFER LIST
                         02560
                         02570     ,       THE I/O BUFFERS
                         02580
                         02590     BUF:    REPEAT  K2-1,
                         02600      <XWD     0,.+1+BUFSIZ
                         02610      BLOCK    BUFSIZ
                         02620     XLIST>
000403  000000 000426'   XWD        0,.+1+BUFSIZ
                                 BLOCK   BUFSIZ
                         XLIST
002064  000000 000000    02640     LASTB:  XWD     0,0
                         02650             BLOCK   BUFSIZ;            THE LAST BUFFER
```

MONITOR SWITCHES

```
002107  000000 000000   02670   DMBY:    Z;        SET WHEN DRUM IS BUSY
002110  000000 000000   02680   DKBY:Z;            SET WHEN DISC IS BUSY
002111  000000 000000   02690   COMEBACK:Z;        SET TO REQUEST INTERPRETER RETURN TO MONITOR
002112  000000 000000   02700   MISC:    Z;        SWITCH USED BY XMTR, DCOMP, TRST, ADIS, RUL
002113  000000 000000   02710   CSS:     Z;                         SWITCH FOR CHS TO TOP OF LIST
002114  000000 000000   02720   CKF:     DEC   0;                   SET TO DO CHECKSUMS OF THE DRUM
```

```
                                02740
                                02750     DEFINE STAC (A)
                                02760     <IRP (A) <A'.S=A-S.QUE>>
                                02770     ,       DEFINITIONS FOR THE STATE INDEXES.
                                02780     END.S=37;       END SIGNAL FOR PARTIAL ORDERING QUEUES
                                02790
                                02800
                                02810     ,               THE ORDER OF THE TABLE REPRESENTS SWAP PRIORITIES
                                02820     S.QUE=.
002115  000000 000000           02830     TOF:    Z;                      WAITING FOR TERMINAL "OFF" PROCESSING
002116  000000 000000           02840     ON:     Z;                      WAITING FOR "ON" PROCESSOR
002117  000000 000000           02850     RC:     Z;                      CARRIAGE RETURN
002120  000000 000000           02860     RI:     Z;                      REQUEST IN (INTERRUPT BUTTON)
002121  000000 000000           02870     RIB:    Z;                      REQUEST IN--NEEDS A BUFFER
002122  000000 000000           02880     UC:     Z;                      UNCHOKE
002123  000000 000000           02890     QC:     Z;                      QUEUE FOR MORE CORE
002124  000000 000000           02900     COM:    Z;                      COMPUTE
                                02910
002125  000000 000000           02920     CU:     Z;                      CURRENT USER
                                02930
002126  000000 000000           02940     DCT:    Z;                      WAITING FOR DISC CONTINUE
002127  000000 000000           02950     DIP:    Z;                      DISC ACTION IN PROGRESS
002130  000000 000000           02960     CK:     Z;                      CHOKED ON OUTPUT
002131  000000 000000           02970     DQ:     Z;                      QUEUE FOR ACCESS TO THE DISC
002132  000000 000000           02980     GR:     Z;                      GREEN (CONSOLE INPUTTING)
002133  000000 000000           02990     DSU:    Z;                      DRAINING BUFFERS FOR SWITCH TO USER
002134  000000 000000           03000     ABG:    Z;                      AWAITING A BUFFER FOR GREEN
002135  000000 000000           03010     QP:     Z;                      PAUSE QUEUE
002136  000000 000000           03020     QDM:    Z;                      QUEUE FOR TRANSFER TO DRUM
002137  000000 000000           03030     QM:     Z;                      QUEUE FOR JOSS INITIAL SERVICE
002140  000210'000141'          03040     OF:     XWD     S.Q+N.S-1,S.Q;  STATION OFF
                                03050
                                03060             STAC <TOF,ON,RC,RI,RIB,UC,QC,COM,CU>
                                   IRP
                                03070             STAC <DCT,DIP,CK,DQ,GR,DSU,ABG,QP,QDM,QM,OF>
                                   IRP
                                03080
                                03090     ,       ALSO STATE RELATED TABLES AT T1,T10, SI1, S10, BOF10.
                                03100     ,                                    S10,S11,S12
```

```
                                03120     ,       ACTION BIT DEFINITIONS
                                03130
                                03140     CORBIT=20;      CORE BIT
                                03150     DKBIT=10;       DISC BIT
                                03160     RCBIT=1;        CARRIER RETURN BIT
                                03170     INBIT=2;        INTERRUPT BIT
                                03180     BUFBIT=4;       BUFFER REQUEST BIT
                                03190     ONBIT=40;       ON STATE BIT
                                03200
                                03210     ,       ACTION TABLE - TYPE AND ENTRY
                                03220
                                03230     T10.:
002141  000000 010151'          03240             XWD     0,EJ;           TOF
002142  000040 007462'          03250             XWD     ONBIT,BJ;       ON
002143  000001 000000           03260             XWD     RCBIT,INTENT;   RC
002144  000002 002143'          03270             XWD     INBIT,INTENT;   RI
002145  000006 002144'          03280             XWD     BUFBIT+INBIT,INTENT;  RIB
002146  000004 002145'          03290             XWD     BUFBIT,INTENT;  UC
002147  000020 002146'          03300             XWD     CORBIT,INTENT;  QC
002150  000000 002147'          03310             XWD     0,INTENT;       COM
```

LOG-ON SCANNER STORAGE

```
002151  440700 000000    03330    SCP1:   POINT   7,0;            INPUT BYTE POINTER
002152  440401 010131'   03340    SCP2:   POINT   4,CART(B);      CODE TABLE BYTE POINTER
002153  440700 000000    03350    SCP3:   POINT   7,0;            OUTPUT BYTE POINTER
002154  360600 002152'   03360    SCP4:   POINT   6,SCP2,5;       POINTER TO POINTER POSITION PART
002155  255001 000000    03370    SC9:    NOP     0,0(B);         POINTER TO EXECUTE TABLE
                         03380
```

COUNTERS FOR ACCOUNTING, STATISTICS, AND DEBUGGING

```
002156  000000 000000   03400   CT10:   Z;        MINUTE
002157  000000 000000   03410   CT10A:  Z;        HOUR
002160  000000 000000   03420   CT32:   Z;        U - USERS
002161  000000 000000   03430   CT33:   Z;        GQ - GREEN STATION QUEUE
                        03440   CT2=.
002162  000000 000000   03450   CT34:   Z;        CQ - COMPUTE QUEUE
002163  000000 000000   03460   CT35:   Z;        BQ - BUFFER, OR CHOKE QUEUE
002164  000000 000000   03470   CT28:   Z;        Q - THE QUEUE
002165  000000 000000   03480   CT13:   Z;        COM - CHARGED COMPUTE TIME
002166  000000 000000   03490   CT14:   Z;        ST - STATEMENTS INTERPRETED
002167  000000 000000   03500   CT15:   Z;        A - UNUSED
002170  000000 000000   03510   CT27:   Z;        I - IN REQUESTS
002171  000000 000000   03520   CT16:   Z;        TL - LINES TRANSMITTED TO USERS
002172  000000 000000   03530   CT37:   Z;        B - BACKSPACES RECEIVED FROM USERS
002173  000000 000000   03540   CT31:   Z;        CR - LINES RECEIVED FROM USERS
                        03550   CT2A=.
002174  000000 000000   03560   CT25:   Z;        1 - COUNTER ONE
002175  000000 000000   03570   CT26:   Z;        2 - COUNTER TWO
002176  000000 000000   03580   CT27A:  Z;        3 - COUNTER THREE
002177  000000 000000   03590   CT27B:  Z;        4 - COUNTER FOUR
002200  000000 000000   03600   CT12:   Z;        U - UNOVERLAPPED I/O TIME (DRUM)
002201  000000 000000   03610   CT11:   Z;        RP - IDLE LOOP COUNT
002202  000000 000000   03620   CT39:   Z;        IC - USERS IN CORE
002203  000000 000000   03630   CT19A:  Z;        SW - DRUM SWAPS
002204  000000 000000   03640   CT18:   Z;        S - SAVES
002205  000000 000000   03650   CT18A:  Z;        D - DELETES
002206  000000 000000   03660   CT18B:  Z;        DI - PRINT DICTIONARY
002207  000000 000000   03670   CT19:   Z;        L - LOADS FROM DISC
002210  000000 000000   03680   CT17:   Z;        SR - STATUS RESPONSES
002211  000000 000000   03690   CT23:   Z;        SE - STATUS ERRORS
002212  000000 000000   03700   CT43:   Z;        1/100 TH % OF AVAIL TIME SPENT ON USERS
002213  000000 000000   03710   CT44:   Z;        TOTAL COMPUTE TIME FOR USERS SINCE BEGIN
                        03720   CT3A=.
002214  000000 000000   03730   CT46:   Z;        INPUT INTERRUPTS
002215  000000 000000   03740   CT47:   Z;        OUTPUT INTERRUPTS
002216  000000 000000   03750   CT48A:  Z;        CELLS FOR DISPLAY OF SIZE DISTR.
002217  000000 000000   03760   CT48B:  Z;
002220  000000 000000   03770   CT48C:  Z;
```

COUNTERS FOR ACCOUNTING, STATISTICS, AND DEBUGGING

```
                                03790     CT7=.
002221  000000 000000           03800     CT41:    Z;                    T - TAPE REWRITE TRIES
002222  000000 000000           03810     CT42:    Z;                    E - TAPE ERASES
002223  000000 000000           03820     CT45:    Z;                    - - UNERASABLE TAPE
002224  000000 000000           03830     CT21:    Z;                    D - DRUM ERRORS
002225  000000 000000           03840     CT22A:   Z;                    K - DISC ERRORS
002226  000000 000000           03850     CT22:    Z;                    C - CONSOLE DETECTED PARITY ERRORS
002227  000000 000000           03860     CT24:    Z;                    S - SCANNER DETECTED PARITY ERRORS
                                03870     CT7A=.
002230  000000 000000           03880     CT50:    Z;                    COUNT CORE COMPACTS
                                03890     CT49=N.DRM
                                03900     TERROR=CT41
                                03910     TERASE=CT42
                                03920     USERS=CT32
                                03930
```

```
                                03950     ;       HEADER LINE FOR CONSOLE TTY OUTPUT
                                03960
002231  522324 052644           03970     HEAD:   ASCII   /TM UR-GQ-C-B COM-STA-A I--TL-C/;  30 CHAR
002232  266172 126606
002233  266044 041636
002234  465332 352202
002235  266024 044532
002236  266511 426606
002237  511330 344532           03980             ASCII   /R-CI-CO T-K-T-#-R-#-D T%-RP-U-IC  SW-S-L-D/; 42 CHAR
002240  416364 052132
002241  455332 426506
002242  266445 521532
002243  421012 422532
002244  512405 552532
002245  446064 020246
002246  535332 326630
002247  266100 000000
```

```
                                04000     ;       STORAGE FOR SUMMARY DATA
                                04010
                                04020     ,       BLOCKS FOR DISTRIBUTIONAL DATA
                                04030     ,       COUNTER RANGES READING THE DISPLAY FROM THE LEFT ARE:
                                04040     ,       >500,500-200,200-100,100-50,50-20,20-10,10-5,5-2,2-1,<1
                                04050
                                04060
                                04070     ,       UNITS FOR THE DISPLAYS ARE:
                                04080     ,       SESSION TIME - MINUTES
                                04090     ,       COMPUTE TIME - 10 SECONDS
                                04100     ,       PROGRAM SIZE - CELLS USED
                                04110     ,       GR-GR   TIME - 1/10 MINUTES
                                04120     ,       COMP/INTERAC - TICKS
                                04130
                                04140     SESTIM: ZBLOK 12;   CUMULATIVE SESSION TIME DISTRIBUTION
                                  REPEAT 12,
002250  000000 000000           Z
                                04150     CCTIM:  ZBLOK 12;   CUMULATIVE COMPUTE TIME
                                  REPEAT 12,
002262  000000 000000           Z
                                04160     DSIZE:  ZBLOK 12;        "      SIZE
                                  REPEAT 12,
002274  000000 000000           Z
                                04170     DTIM:   ZBLOK 12;        "    GREEN-GREEN INTERARRIVAL TIME
                                  REPEAT 12,
002306  000000 000000           Z
                                04180     DCPI:   ZBLOK 12;        "      COMPUTE TIME PER INTERACTION
                                  REPEAT 12,
002320  000000 000000           Z
                                04190
                                04200     ,       LINEAR BUCKETS
                                04210     ,       RANGE:  >=5,5-10,10-15, ... ,40-45,45-78
                                04220
                                04230     CPIL:   ZBLOK 12;   CHAR/INPUT LINE
                                  REPEAT 12,
002332  000000 000000           Z
                                04240     CPOL:   ZBLOK 12;   CHAR/OUTPUT LINE
                                  REPEAT 12,
002344  000000 000000           Z
```

```
                                04260
                                04270      ;        CHARACTER (MOSTLY) INTERRUPTS/SECOND/USER
                                04280      ,        RANGE:  >=15,15-7.5,7.5-3,3-1.5, ... ,.075-.03,<.03
                                04290
                                04300      IINT:    ZBLOK 12;   INPUT INTERRUPTS
                                  REPEAT 12,
002356  000000 000000           Z
                                04310      OINT:    ZBLOK 12;   OUTPUT INTERRUPTS
                                  REPEAT 12,
002370  000000 000000           Z
                                04320      TINT:    ZBLOK 12;   TOTAL INTERRUPTS
                                  REPEAT 12,
002402  000000 000000           Z
                                04330      T5:      ZBLOK 12;   INTERPRETATION RATE
                                REPEAT 12,
002414  000000 000000           Z
                                04340      T6:      ZBLOK 12;   BLOCKSIZE DISTRIBUTION
                                  REPEAT 12,
002426  000000 000000           Z
                                04350      T7:      ZBLOK 12;   MISC COUNTS
                                  REPEAT 12,
002440  000000 000000           Z
                                04360      T7.9=T7+9
                                04370      T8:      ZBLOK 12;   TASK TURN-AROUND TIME
                                  REPEAT 12,
002452  000000 000000           Z
                                04380      T9:      ZBLOK 12;   JOE USES T9, T10, AND T11 TO COUNT
                                  REPEAT 12,
002464  000000 000000           Z
                                04390      T10:     ZBLOK 12;   COMMAND EXECUTIONS BY VERB TYPE
                                  REPEAT 12,
002476  000000 000000           Z
                                04400      T11:     ZBLOK 12
                                  REPEAT 12,
002510  000000 000000           Z
                                04410
                                04420      T11.1=T11+1
                                04430      T11.2=T11+2
                                04440      T11.3=T11+3
                                04450      T11.4=T11+4
                                04460      T11.5=T11+5
                                04470      T11.6=T11+6
                                04480      T11.7=T11+7
                                04490      T11.8=T11+8
                                04500      T11.9=T11+9
```

```
                                 04520
                                 04530      ,       PER USER COUNTERS
                                 04540
                                 04550      ,       RIGHT=COMPUTE TICKS SINCE LAST GREEN
                                 04560      ,       LEFT=TIME OF LAST GREEN
                                 04570
                                 04580      USTAT:  ZBLOK N.S;  USER COUNTERS
                                   REPEAT N.S,
002522  000000 000000            Z
                                 04590
                                 04600      MINT:   ZBLOK N.S;  CURRENT USERS INITIALS
                                   REPEAT N.S,
002572  000000 000000            Z
                                 04610      LIST
                                 04620      SUM:    ZBLOK N.S;  USER BLOCK CHECKSUMS
                                   REPEAT N.S,
002642  000000 000000            Z
```

FIELD SIZE, SCALE FACTOR, AND POINTER FOR COUNTERS.

```
                                04640      ,        IN CTP MACRO: A=# OF COLUMNS (MUST TOTAL <=72)
                                04650      ,        B=SCALE FACTOR USED BEFORE PRINT
                                04660      ,        C=CT (OR COUNTER) #
                                04670      ,        E=0 IF COUNTER IS ZEROED EACH MINUTE
                                04680      DEFINE CTP (A,B,C,E)
                                04690      <OPDEF CTP2[BYTE (1)E(5)A(12)↑D'B]
                                04700      CTP2 CT'C>
                                04710      CTPAR:CTP 3,1,32,1; UR - USERS
                                OPDEF CTP2[BYTE (1)1(5)3(12)↑D1]
002712  430001 002160'          CTP2 CT32
                                04720      CTP 3,1,33;          GQ - GREEN USERS
                                OPDEF CTP2[BYTE (1)(5)3(12)↑D1]
002713  030001 002161'          CTP2 CT33
                                04730      CTP 2,1,34;          C - COMPUTING USERS
                                OPDEF CTP2[BYTE (1)(5)2(12)↑D1]
002714  020001 002162'          CTP2 CT34
                                04740      CTP 2,1,35;          B - OUTPUT LIMITED USERS
                                OPDEF CTP2[BYTE (1)(5)2(12)↑D1]
002715  020001 002163'          CTP2 CT35
                                04750      CTP 4,6,13;          COM- COMPUTE TIME
                                OPDEF CTP2[BYTE (1)(5)4(12)↑D6]
002716  040006 002165'          CTP2 CT13
                                04760      CTP 4,10,14;         STA - STATEMENTS INTERPRETED
                                OPDEF CTP2[BYTE (1)(5)4(12)↑D10]
002717  040012 002166'          CTP2 CT14
                                04770      CTP 2,1,15,1;        A - IGNORED PARITY ERRORS
                                OPDEF CTP2[BYTE (1)1(5)2(12)↑D1]
002720  420001 002167'          CTP2 CT15
                                04780      CTP 2,1,50;          M - CORE COMPACTS
                                OPDEF CTP2[BYTE (1)(5)2(12)↑D1]
002721  020001 002230'          CTP2 CT50
                                04790      CTP 4,1,16;          TL - LINES TRANSMITTED TO USERS
                                OPDEF CTP2[BYTE (1)(5)4(12)↑D1]
002722  040001 002171'          CTP2 CT16
                                04800      CTP 3,1,31;          CR - LINES RECEIVED FROM USERS
                                OPDEF CTP2[BYTE (1)(5)3(12)↑D1]
002723  030001 002173'          CTP2 CT31
                                04810      CTP 3,10,25;         CI - CHARACTERS INPUT
                                OPDEF CTP2[BYTE (1)(5)3(12)↑D10]
002724  030012 002174'          CTP2 CT25
                                04820      CTP 3,100,26;        CO - CHARACTERS OUTPUT
                                OPDEF CTP2[BYTE (1)(5)3(12)↑D100]
002725  030144 002175'          CTP2 CT26
                                04830      CTP 2,1,41,1;        T - TAPE ERRORS
                                OPDEF CTP2[BYTE (1)1(5)2(12)↑D1]
002726  420001 002221'          CTP2 CT41
                                04840      CTP 2,1,22A,1;       K - DISC ERRORS
                                OPDEF CTP2[BYTE (1)1(5)2(12)↑D1]
002727  420001 002225'          CTP2 CT22A
```

FIELD SIZE, SCALE FACTOR, AND POINTER FOR COUNTERS.

```
                                04860     CTP 2,1,27A;          P - LAST STATION TO TRANSMIT PARITY ERROR
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002730  020001 002176'          CTP2 CT27A
                                04870     CTP 2,1,22;           C - CONSOLE PARITY ERRORS
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002731  020001 002226'          CTP2 CT22
                                04880     CTP 2,1,27B;          B - LAST STATION TO TRANSMIT A BAD CHARACTER
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002732  020001 002177'          CTP2 CT27B
                                04890     CTP 2,1,24;           S  - SCANNER PARITY ERRORS
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002733  020001 002227'          CTP2 CT24
                                04900     CTP 2,1,21,1;         D - DRUM ERRORS
                                OPDEF CTP2[BYTE (1) 1 (5) 2 (12) ↑D1]
002734  420001 002224'          CTP2 CT21
                                04910     CTP 3,100,43,1;       1/100% COMPUTE TIME - RESET EACH HOUR (LOGS AS %)
                                OPDEF CTP2[BYTE (1) 1 (5) 3 (12) ↑D100]
002735  430144 002212'          CTP2 CT43
                                04920     CTP 3,2000,11;        ID - IDLE LOOP COUNT
                                OPDEF CTP2[BYTE (1) (5) 3 (12) ↑D2000]
002736  033720 002201'          CTP2 CT11
                                04930     CTP 2,1,12;           U - UNOVERLAPPED I/O COUNTS
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002737  020001 002200'          CTP2 CT12
                                04940     CTP 3,1,39,1;         OD - USERS IN CORE
                                OPDEF CTP2[BYTE (1) 1 (5) 3 (12) ↑D1]
002740  430001 002202'          CTP2 CT39
                                04950     CTP 4,1,19A;          SW - DRUM READS+WRITES
                                OPDEF CTP2[BYTE (1) (5) 4 (12) ↑D1]
002741  040001 002203'          CTP2 CT19A
                                04960     CTP 2,1,18;           S - DISC SAVES
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002742  020001 002204'          CTP2 CT18
                                04970     CTP 2,1,19;           L - DISC LOADS
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002743  020001 002207'          CTP2 CT19
                                04980     CTP 2,1,18A;          D - DISC DISCARDS
                                OPDEF CTP2[BYTE (1) (5) 2 (12) ↑D1]
002744  020001 002205'          CTP2 CT18A
                                04990     CTPARS=.-CTPAR
```

DATA CONTROL INTERRUPT ROUTINE

```
                          05010
002745  000000 000000     05020     DISMIS: Z;                        DATA CONTROL END ROUTINE
002746  700600 001100     05030             CONO    PI,1000+CHDC;     TURN OFF CHANNEL
002747  254520 002745'    05040             JRST    12,@DISMIS
```

DRUM CHANNEL INTERRUPT ROUTINES

```
002750  000000 000000   05060   DRMR:  Z
002751  402000 003110'  05070          SETZM   DRM14;          IF ERROR, FLAG STORED HERE
002752  701300 100060   05080          CONSZ   DP,100060;
002753  254000 003126'  05090          JRST    DEER;           MISSED DATA, NOEX MEM, OR PARITY ERROR
002754  336000 002107'  05100          SKIPN   DMBY
002755  040000 000015   05110   H15:   HALT    15;             SHOULD BE USING THE DRUM
002756  701340 000010   05120          CONSO   DP,10
002757  254000 002762'  05130          JRST    DRM3
002760  701200 000000   05140          CONO    DP,0;           KILL THE INTERRUPT
002761  254520 002750'  05150          JRST    12,@DRMR;       DISMISS END INTERRUPT FROM 167
                       05160
002762  250040 003103'  05170   DRM3:  EXCH    B,DRM9
002763  250100 003102'  05180          EXCH    C,DRM8
002764  740300 001000   05190          CONSZ   DR,1000;        CHECK FOR ERROR
002765  254000 003133'  05200          JRST    DERR
002766  201040 077777   05210          MOVEI   B,77777
002767  406040 003326'  05220          ANDM    B,DPWD1;        MASK OUT POSSIBLE HIGH CARRY BIT
002770  701040 000001   05230          DATAI   DP,B
002771  312040 003326'  05240          CAME    B,DPWD1;        CHECK FOR GOOD
002772  254000 003133'  05250          JRST    DERR
002773  740040 000001   05260          DATAI   DR,B
002774  312040 003165'  05270          CAME    B,DEDR1;        DRUM REGISTER ENDING CONTENTS
002775  254000 003133'  05280          JRST    DERR
002776  740340 000100   05290          CONSO   DR,100
002777  040000 000017   05300   H17:   HALT    17;             NO JOB DONE FLAG
003000  740200 000270   05310          CONO    DR,270;         DESELECT THE DRUM
003001  402000 002107'  05320          SETZM   DMBY
003002  336000 003330'  05330          SKIPN   DMWR;           SKIP IF WRITING
003003  254400 003043'  05340          JRST    10,DRM7;        MUST BE A READ
003004  402000 003163'  05350          SETZM   DECT;           ZERO THE ERROR COUNT
                       05360
                       05370          WRITE END ROUTINE
                       05380
003005  250700 003107'  05390          EXCH    S,DRM13
003006  200700 003327'  05400          MOVE    S,DMUSR
003007  200100 003332'  05410          MOVE    C,BLKSWRIT
003010  200040 003324'  05420          MOVE    B,DMIBK
003011  402001 000272'  05430          SETZM   CORE(B);        IDLE THE CORE JUST WRIT
003012  350000 000001   05440          AOS     B
003013  367100 003011'  05450          SOJG    C,.-2
003014  350000 000320'  05460   DR2:   AOS     N.DRM;          COUNT DRUM USERS
003015  370000 002202'  05470          SOS     CT39;           COUNT DOWN USERS IN CORE
003016  200040 003305'  05480          MOVE    B,DMNR;         GET LIST FOR OUT
003017  603040 777777   05490          TLNE    B,777777;       SKIP IF NO MORE TO GO OUT
003020  254400 003113'  05500          JRST    10,DM10;        GO FOR NEXT WRITE
003021  332000 005470'  05510          SKIPE   SS98;           SKIP IF NO COMPACT REQUEST
003022  254400 003037'  05520          JRST    10,DRM2
003023  335700 003331'  05530          SKIPGE  S,DMIN;         SKIP IF SOME TO COME IN
003024  254400 003031'  05540          JRST    10,DRM4.5
003025  250740 003111'  05550          EXCH    PP,DRMPP
003026  260740 003333'  05560          TSX     ISWAP;          INITIATE IN TRANSFER
003027  250740 003111'  05570          EXCH    PP,DRMPP
003030  254400 003032'  05580          JRST    10,DRM5
                       05590
```

```
                                05610     ;        INTERRUPT END SEQUENCE
                                05620
003031  350000 002111'          05630     DRM4.5:  AOS     COMEBACK
003032  250700 003107'          05640     DRM5:    EXCH    S,DRM13
003033  250040 003103'          05650     DRM6:    EXCH    B,DRM9
003034  250100 003102'          05660              EXCH    C,DRM8
003035  350000 002203'          05670     DRM6.5:  AOS     CT19A;          COUNT DRUM ACTIONS
003036  254120 002750'          05680              JRST    2,@DRMR
                                05690
003037  350000 005471'          05700     DRM2:    AOS     SS99;           SET REQUEST FOR COMPACT OF CORE
003040  402000 005470'          05710              SETZM   SS98
003041  350000 002107'          05720              AOS     DMBY
003042  254000 003031'          05730              JRST    DRM4.5
                                05740
```

```
                          05760     ;        READ END INTERRUPT
                          05770
003043  250700 003331'    05780     DRM7:    EXCH    S,DMIN
003044  250000 003110'    05790              EXCH    A,DRM14
003045  550040 003324'    05800              HRRZ    B,DMIBK
003046  271040 000020     05810              ADDI    B,BBLOCK
003047  135100 000264'    05820              LDB     C,S.BLOCK
003050  135000 000261'    05830              LDB     A,S.STA
003051  306000 000006     05840              CAIN    A,QC.S;          CHECK FOR EXPANDING SIZE
003052  275100 000001     05850              SUBI    C,1
003053  264000 003207'    05860              JSR     CKSUM;           CHECKSUM THE BLOCK
003054  316016 002642'    05870              CAMN    A,SUM(S)
003055  254000 003061'    05880              JRST    DRM7.5;          GOOD
003056  250000 003110'    05890              EXCH    A,DRM14
003057  250700 003331'    05900              EXCH    S,DMIN
003060  254000 003137'    05910              JRST    DMERR
003061  402000 003163'    05920     DRM7.5:  SETZM   DECT
003062  250000 003110'    05930              EXCH    A,DRM14
003063  550040 003324'    05940              HRRZ    B,DMIBK
003064  137700 000317'    05950              DPB     S,S.UR
003065  271040 000020     05960              ADDI    B,BBLOCK
003066  137040 000263'    05970              DPB     B,S.COR;         SET USER CORE LOCATION
003067  250000 003105'    05980              EXCH    A,DRM11;         SAVE REG AND GET A 2
003070  550040 003324'    05990              HRRZ    B,DMIBK
003071  135100 000264'    06000              LDB     C,S.BLOCK
003072  137000 000316'    06010              DPB     A,S.UD;          SET IN CORE AND NO DRUM USE
003073  350000 000001     06020              AOS     B
003074  367100 003072'    06030              SOJG    C,.-2
003075  250000 003105'    06040              EXCH    A,DRM11
003076  350000 002202'    06050              AOS     CT39
003077  370000 000320'    06060     DRM7.6:  SOS     N.DRM
003100  250700 003331'    06070     DRM7.7:  EXCH    S,DMIN
003101  254000 003033'    06080              JRST    DRM6
                          06090
```

DRUM CHANNEL INTERRUPT ROUTINES

```
003102  000000 000000   06110   DRM8:   Z;                      SAVE FOR REG C
003103  000000 000000   06120   DRM9:   Z;                      SAVE FOR REG B
003104  000000 000001   06130   DRM10:  OCT     1;              SAVE FOR REG A AND A ONE
003105  000000 000002   06140   DRM11:  OCT     2;              SAVE FOR REG A AND A TWO
003106  000000 000000   06150   DRM12:  Z
003107  000000 000000   06160   DRM13:  Z;                      SAVE CELL FOR S
003110  000000 000000   06170   DRM14:  Z;                      SAVE FOR A
003111  777776 003111'  06180   DRMPP:  XWD     -2,.;           LOCAL PUSH REGISTER
003112  000000 000000   06190           Z
                        06200
                        06210
                        06220
                        06230   ,       INITIATE NEXT WRITE
                        06240
                        06250   ,       B HAS PP REG FOR REQUEST LIST
                        06260
003113  250740 003111'  06270   DM10:   EXCH    PP,DRMPP
003114  262040 000016   06280           POP     B,S;            GET NEXT TO WRITE
003115  135100 000264'  06290           LDB     C,S.BLOCK
003116  250140 003106'  06300           EXCH    D,DRM12
003117  202040 003305'  06310           MOVEM   B,DMNR
003120  250000 003110'  06320           EXCH    A,DRM14
003121  260740 003234'  06330           TSX     OSWAP;          START HIM OUT
003122  250000 003110'  06340           EXCH    A,DRM14
003123  250140 003106'  06350           EXCH    D,DRM12
003124  250740 003111'  06360           EXCH    PP,DRMPP
003125  254000 003032'  06370           JRST    DRM5
```

```
                        06390     ;        DATA PROCESSOR ERRORS (PARITY, NOEX MEM, DATA MISS)
                        06400
003126  250040 003103'  06410     DEER:    EXCH    B,DRM9
003127  701240 000001   06420              CONI    DP,B
003130  506040 003110'  06430              HRLM    B,DRM14
003131  701200 000000   06440              CONO    DP,0
003132  250100 003102'  06450              EXCH    C,DRM8
                        06460
                        06470     ;        OTHER HARDWARE ERRORS--DC FLAGS AND DATA REGISTER MISCOMPARE
                        06480
003133  701240 000001   06490     DERR:    CONI    DP,B
003134  542040 003110'  06500              HRRM    B,DRM14;        SAVE FLAGS FOR DISPLAY
003135  740200 200270   06510              CONO    DR,200270;      DISMISS DRUM
003136  254400 003137'  06520              JRST    10,DMERR
                        06530
003137  350000 002224'  06540     DMERR:   AOS     CT21;           COUNT DRUM ERRORS
003140  350040 003163'  06550              AOS     B,DECT
003141  303040 000005   06560              CAILE   B,5
003142  254000 003166'  06570              JRST    FAIL
003143  350000 002107'  06580              AOS     DMBY
003144  336000 003330'  06590              SKIPN   DMWR;           SKIP IF WRITE
003145  254000 003154'  06600              JRST    DE10;           GO RECOVER READ ERROR
003146  701200 000102   06610              CONO    DP,100+CDRM
003147  701140 003325'  06620              DATAO   DP,DPWD
003150  740140 003164'  06630              DATAO   DR,DEDR
003151  740200 000260   06640              CONO    DR,260
003152  740200 000222   06650              CONO    DR,220+CDRM
003153  254000 003033'  06660              JRST    DRM6
                        06670
003154  701200 000002   06680     DE10:    CONO    DP,CDRM
003155  701140 003325'  06690              DATAO   DP,DPWD
003156  740140 003164'  06700              DATAO   DR,DEDR
003157  740200 000260   06710              CONO    DR,260
003160  740200 000232   06720              CONO    DR,230+CDRM
003161  254000 003033'  06730              JRST    DRM6
                        06740
003162  000000 000000   06750     CKER:    Z;                      SET IF A GRONKED USER
003163  000000 000000   06760     DECT:    Z;                      ERROR RETRY COUNT
003164  000000 000000   06770     DEDR:    Z;                      UNIT AND TRACK OF LAST SELECT
003165  000000 000000   06780     DEDR1:   Z;                      EXPECTED ENDING TRACK AND SECTOR
```

```
                                06800     ;        CANT RECOVER DRUM ERROR--PREPARE TO GRONK USER
                                06810
003166  250700 003331'          06820     FAIL:    EXCH    S,DMIN
003167  250000 003110'          06830              EXCH    A,DRM14
003170  202016 002572'          06840              MOVEM   A,MINT(S);       SAVE FAILING CHECKSUM
003171  350000 003162'          06850              AOS     CKER;            FLAG ERROR OCCURRENCE
003172  201000 000001           06860              MOVEI   A,1
003173  137000 000270'          06870              DPB     A,S.GK;          FLAG USER AS GRONKED
003174  250000 003110'          06880              EXCH    A,DRM14
003175  402000 003163'          06890              SETZM   DECT
003176  402000 002107'          06900              SETZM   DMBY
003177  332000 003330'          06910              SKIPE   DMWR
003200  254000 003100'          06920              JRST    DRM7.7
003201  135100 000264'          06930              LDB     C,S.BLOCK
003202  550040 003324'          06940              HRRZ    B,DMIBK
003203  402001 000272'          06950              SETZM   CORE(B);         FREE CORE
003204  350000 000001           06960              AOS     B
003205  367100 003203'          06970              SOJG    C,.-2
003206  254000 003077'          06980              JRST    DRM7.6
```

```
                                07000     ,                REPORTS ZERO CHECKSUM WHEN CKF IS ZERO
                                07010     ,                C,B CLOBBERED
                                07020     ,                S= USER #, C=BLOCKS TO SUM, B= LOCATION
003207  000000 000000           07030     CKSUM:  Z
003210  332000 002114'          07040             SKIPE   CKF
003211  254000 003214'          07050             JRST    .+3
003212  201000 000000           07060             MOVEI   A,0
003213  254120 003207'          07070             JRST    2,@CKSUM
003214  240100 000012           07080             ASH     C,↑D10
003215  275100 000001           07090             SUBI    C,1
003216  240040 000012           07100             ASH     B,↑D10
003217  542040 003231'          07110             HRRM    B,CKS7
003220  250340 003231'          07120             EXCH    7,CKS7
003221  250400 003232'          07130             EXCH    10,CKS10
003222  250440 003233'          07140             EXCH    11,CKS11
003223  201000 000000           07150             MOVEI   A,0
003224  254000 000007           07160             JRST    7
003225  250340 003231'          07170     CKS5:   EXCH    7,CKS7
003226  250400 003232'          07180             EXCH    10,CKS10
003227  250440 003233'          07190             EXCH    11,CKS11
003230  254120 003207'          07200             JRST    2,@CKSUM
                                07210
                                07220
003231  270002 000000           07230     CKS7:   ADD     A,0 (C)
003232  365100 000007           07240     CKS10:  SOJGE   C,7
003233  254000 003225'          07250     CKS11:  JRST    CKS5
                                07260
                                07270
                                07280
```

INITIATE TRANSFER TO DRUM

```
                                07300     ,        C= # OF BLOCKS TO WRITE
                                07310     ,        S= USER TO WRITE
                                07320     ,        DRUM LOCATIONS ARE FIXED BY USER - TWO TRACKS EACH,
                                07330     ,        SO THE MAX SIZE HANDLED IS 16K WORDS PER USER. B,D ARE CLOBBERED
                                07340
003234  740300 000200           07350     OSWAP:   CONSZ   DR,200
003235  040000 000014           07360     H14:     HALT    14;              DRUM SHOULD BE FREE
003236  202100 003332'          07370              MOVEM   C,BLKSWRIT
003237  135040 000263'          07380              LDB     B,S.COR
003240  264000 003207'          07390              JSR     CKSUM
003241  202016 002642'          07400              MOVEM   A,SUM(S)
003242  200100 003332'          07410              MOVE    C,BLKSWRIT
003243  240100 000006           07420              ASH     C,6
003244  202100 003165'          07430              MOVEM   C,DEDR1
003245  240100 000004           07440              ASH     C,4
003246  202100 003326'          07450              MOVEM   C,DPWD1
003247  200100 003332'          07460              MOVE    C,BLKSWRIT
003250  550140 000002           07470              HRRZ    D,C
003251  242100 000034           07480              LSH     C,↑D28
003252  212100 003325'          07490              MOVNM   C,DPWD;          WORDS TO MOVE
003253  135040 000263'          07500              LDB     B,S.COR
003254  275040 000020           07510              SUBI    B,BBLOCK
003255  202040 003324'          07520              MOVEM   B,DMIBK;         SAVE LOCATION OF BLOCK WRITTEN
003256  201100 000001           07530              MOVEI   C,1
003257  137100 000315'          07540              DPB     C,S.ID;          SET DRUM USE BIT
003260  350000 000001           07550              AOS     B
003261  367140 003257'          07560              SOJG    D,.-2
003262  135040 000263'          07570              LDB     B,S.COR
003263  240040 000012           07580              ASH     B,↑D10
003264  137040 000263'          07590              DPB     B,S.COR;         MARK USER OUT OF CORE
003265  542040 003325'          07600              HRRM    B,DPWD;          FIRST WORD TO TRANSFER
003266  272040 003326'          07610              ADDM    B,DPWD1
003267  701200 000102           07620              CONO    DP,100+CDRM;     SELECT FOR WRITE AND SET CHANNEL
003270  701140 003325'          07630              DATAO   DP,DPWD;         ACTIVATE I/O PROCESSOR
003271  550040 000016           07640              HRRZ    B,S
003272  552040 003327'          07650              HRRZM   B,DMUSR
003273  270040 000321'          07660              ADD     B,DBASE
003274  240040 000012           07670              ASH     B,↑D10;          TRACK = 2&USER+DBASE
003275  202040 003164'          07680              MOVEM   B,DEDR;          SAVE UNIT AND TRACK FOR RECOVERY
003276  740140 000001           07690              DATAO   DR,B;            SET UNIT AND TRACK
003277  740200 000260           07700              CONO    DR,260;          SELECT UNIT
003300  740200 000222           07710              CONO    DR,220+CDRM;     WRITE
003301  272040 003165'          07720              ADDM    B,DEDR1
003302  350000 003330'          07730              AOS     DMWR;            INDICATE DRUM WRITE
003303  350000 002107'          07740              AOS     DMBY
003304  263740 000000           07750              DONE
                                07760
```

DATA FOR DRUM TRANSFERS

```
003305  000000 003305'  07780     DMNR:    XWD     0,.;            LIST OF USERS TO WRITE
                        07790              ZBLOK 16
                          REPEAT 16,
003306  000000 000000   Z
003324  000000 000000   07800     DMIBK:   Z;                      INCOMING BLOCK NUMBER
003325  000000 000000   07810     DPWD:    Z;                      I/O CONTROL WORD
003326  000000 000000   07820     DPWD1:   Z;                      EXPECTED END CONTENTS OF 167
003327  000000 000000   07830     DMUSR:   Z;                      DRUM USER
003330  000000 000000   07840     DMWR:    Z;       SET WHEN WRITING
003331  000000 000000   07850     DMIN:    Z; RIGHT = USER TAGGED FOR IN, LEFT = PLACE IN QUEUE, NEG IF NO IN
003332  000000 000000   07860     BLKSWR:  Z;                      # OF BLOCKS WRITTEN
                        07870
                        07880
```

INITIATE TRANSFER FROM DRUM TO CORE

```
                                07900      ,       S HAS USER TO COME IN; DMIBK= CORE BLOCK FOR "IN"
                                07910      ,       S IS CLOBBERED
                                07920
003333  135100  000264'         07930      ISWAP:  LDB     C,S.BLOCK
003334  135040  000261'         07940              LDB     B,S.STA
003335  306040  000006          07950              CAIN    B,QC.S;          CHECK FOR SIZE INCREASE
003336  275100  000001          07960              SUBI    C,1
003337  202700  003331'         07970              MOVEM   S,DMIN
003340  202100  000016          07980              MOVEM   C,S;             SAVE CORRECT BLOCK SIZE
003341  242100  000034          07990              LSH     C,↑D28
003342  212100  003325'         08000              MOVNM   C,DPWD;          WORD COUNT
003343  550040  003324'         08010              HRRZ    B,DMIBK;         GET IN LOCATION
003344  271040  000020          08020              ADDI    B,BBLOCK
003345  240040  000012          08030              ASH     B,↑D10
003346  542040  003325'         08040              HRRM    B,DPWD;          FIRST WORD
003347  552040  003326'         08050              HRRZM   B,DPWD1
003350  550040  003324'         08060              HRRZ    B,DMIBK
003351  200100  000016          08070              MOVE    C,S
003352  240100  000012          08080              ASH     C,↑D10
003353  272100  003326'         08090              ADDM    C,DPWD1
003354  200100  000016          08100              MOVE    C,S
003355  240100  000006          08110              ASH     C,6
003356  202100  003165'         08120              MOVEM   C,DEDR1
003357  200100  000016          08130              MOVE    C,S
003360  250000  003104'         08140              EXCH    A,DRM10
003361  137000  000315'         08150              DPB     A,S.ID;          SET DRUM IN USE
003362  350000  000001          08160              AOS     B
003363  367100  003361'         08170              SOJG    C,.-2
003364  402000  003330'         08180              SETZM   DMWR
003365  350000  002107'         08190              AOS     DMBY
003366  200700  003331'         08200              MOVE    S,DMIN;          RECOVER USER #
003367  270700  000321'         08210              ADD     S,DBASE
003370  240700  000012          08220              ASH     S,↑D10;          TRACK = 2&USER+DBASE
003371  701200  000002          08230              CONO    DP,CDRM;         SAY READ AND ASSIGN CHANNEL
003372  701140  003325'         08240              DATAO   DP,DPWD;         ACTIVATE 167
003373  202700  003164'         08250              MOVEM   S,DEDR;          SAVE UNIT AND TRACK FOR RECOVERY
003374  700600  000400          08260              CONO    PI,400;          PREVENT INTERRUPT FOR A WHILE
003375  740140  000016          08270              DATAO   DR,S;            UNIT AND TRACK
003376  740200  000260          08280              CONO    DR,260;          SELECT UNIT
003377  740200  000232          08290              CONO    DR,230+CDRM;     READ
003400  250000  003104'         08300              EXCH    A,DRM10
003401  700600  000200          08310              CONO    PI,200;          ON AGAIN
003402  272700  003165'         08320              ADDM    S,DEDR1;         EXPECTED END #
003403  553000  003165'         08330              HRRZS   DEDR1;           ZERO LEFT HALF
003404  263740  000000          08340              DONE
                                08350
```

```
003405  722640 003444'  08370   TIP:    CONI    MT1,TEND4;      READ IN TAPE FLAGS
003406  722740 000004   08380           CONSO   MT1,4;          ERF INTERRUPT?
003407  254000 003432'  08390           JRST    TE20
                        08400
003410  722600 000000   08410   TEND:   CONO    MT1,0;          DISABLE THE INTERRUPTS
003411  250700 003445'  08420           EXCH    S,TEND5
003412  250040 003446'  08430           EXCH    B,TEND6
003413  250200 003447'  08440           EXCH    E,TEND7;        GIVE ME SOME ELBOW ROOM
003414  250740 003530'  08450           EXCH    PP,TPDL
003415  200040 003444'  08460           MOVE    B,TEND4;        GET THE FLAGS
003416  602040 000070   08470           TRNE    B,70;           SKIP IF NO ERROR
003417  254000 003451'  08480           JRST    TEND9;          GO FIX THE ERROR
003420  402000 003450'  08490           SETZM   TEND8;          OK, ZERO THE RETRY COUNT
003421  332000 003444'  08500           SKIPE   TEND4;          RESTART SO DONT GIVE BACK BUFFER
003422  260740 010365'  08510           TSX     MBA
003423  332000 000327'  08520           SKIPE   DT.BUF;         SKIP IF NO MORE BUFFERS
003424  260740 003475'  08530   TEND1:  TSX     CIO
003425  250700 003445'  08540   TEND2:  EXCH    S,TEND5
003426  250040 003446'  08550           EXCH    B,TEND6
003427  250200 003447'  08560           EXCH    E,TEND7
003430  250740 003530'  08570           EXCH    PP,TPDL
003431  254520 003544'  08580           JRST    12,@CTYR
                        08590
003432  336000 003450'  08600   TE20:   SKIPN   TEND8;          SKIP TO IGNORE BKSPACE OR WRITE BLANK
003433  254000 003436'  08610           JRST    TE22;           GO TO TRY TO READY TAPE
003434  722600 000000   08620   TE21:   CONO    MT1,0;          KILL THE INTERRUPT
003435  254520 003544'  08630           JRST    12,@CTYR
003436  700700 000100   08640   TE22:   CONSZ   PI,CHDC;        GET IT LATER IF DATA CONTROL IS BUSY
003437  254000 003434'  08650           JRST    TE21
003440  722740 000002   08660           CONSO   MT1,2;          SKIP IF TRANSPORT READY
003441  254000 003434'  08670           JRST    TE21;           RETURN WE WILL TRY AGAIN LATER
003442  402000 003444'  08680           SETZM   TEND4;          SET RESTART SIGNAL
003443  254000 003410'  08690           JRST    TEND;           GO DO IT
                        08700
003444  000000 000000   08710   TEND4:  Z;                      FLAG STORAGE
003445  777777 777776   08720   TEND5:  DEC     -2;             SAVE CELL FOR S AND CONTEXT FOR TAPE
003446  000000 000000   08730   TEND6:  Z
003447  000000 000000   08740   TEND7:  Z;                      SAVE CELL
003450  000000 000000   08750   TEND8:  Z;                      COUNT OF REWRITE TRIES
                        08760
```

```
003451  350000 002221'  08780    TEND9:  AOS     TERROR;          COUNT TAPE ERRORS
003452  350040 003450'  08790    TEND93: AOS     B,TEND8;         COUNT REWRITE TRIES
003453  722600 000000   08800            CONO    MT1,0;           OFF WITH THE LIGHTS
003454  260740 003470'  08810            TSX     TWAIT;           WAIT ON TAPE
003455  722200 003406   08820            CONO    MTC,CDAT+3400;   BACKSPACE TAPE
003456  260740 003470'  08830            TSX     TWAIT
003457  307040 000003   08840            CAIG    B,3;             TRY REWRITE 3 TIMES.
003460  254000 003424'  08850            JRST    TEND1
                        08860
003461  722200 005406   08870    TEND95: CONO    MTC,CDAT+5400;   WRITE BLANK TAPE
003462  260740 003470'  08880            TSX     TWAIT
003463  722700 000030   08890            CONSZ   MT1,30;          CHECK FOR NOT ERASED
003464  350000 002223'  08900            AOS     CT45;            COUNT ERASE ERRORS
003465  350000 002222'  08910            AOS     TERASE
003466  402000 003450'  08920            SETZM   TEND8
003467  254000 003424'  08930            JRST    TEND1
                        08940
003470  722740 040000   08950    TWAIT:  CONSO   MT1,40000;       WAIT FOR COMMAND OK
003471  254000 003470'  08960            JRST    .-1
003472  722740 000002   08970            CONSO   MT1,2;           AND FOR TAPE READY
003473  254000 003472'  08980            JRST    .-1
003474  263740 000000   08990            DONE
                        09000
                        09010
```

```
                        09030     ;        MAG TAPE OUTPUT START ROUTINE
                        09040
003475  332000 006103'  09050     CIO:     SKIPE   SKR;              SKIP IF SKULKER NOT USING DC
003476  254000 003524'  09060              JRST    CIO1
003477  336000 002127'  09070              SKIPN   DIP;              NO START IF DISC USING DC
003500  722700 000200   09080              CONSZ   MT1,200
003501  254000 003524'  09090              JRST    CIO1;             CANT WRITE ON FILE PROTECTED TAPE
003502  722740 000002   09100              CONSO   MT1,2;            TAPE SHOULD BE READY
003503  254000 003524'  09110              JRST    CIO1;             TAPE MUST BE READY
003504  200040 010533'  09120              MOVE    B,[BLKO DC,TCW]
003505  202040 000042   09130              MOVEM   B,40+2*CDC;       SET UP DC INTERRUPT ROUTINE
003506  200040 010534'  09140              MOVE    B,[JSR DISMIS]
003507  202040 000043   09150              MOVEM   B,41+2*CDC
003510  550040 000327'  09160              HRRZ    B,DT.BUF;         ADDRESS OF BUFFER
003511  542040 003527'  09170              HRRM    B,TCW
003512  554041 000001   09180              HLRZ    B,1(B);           GET WORD COUNT
003513  213040 000001   09190              MOVNS   B,B
003514  506040 003527'  09200              HRLM    B,TCW;            NEG COUNT TO CONTROL WORD
003515  722600 000000   09210              CONO    MT1,0;            DISABLE ALL, I DONT KNOW WHY!
003516  720200 003421   09220              CONO    DC,CDC+DMT+3400;   START DATA OUT
003517  722200 071006   09230              CONO    MTC,CDAT+71000;   START TAPE - BINARY @556 BPI
003520  700600 002100   09240              CONO    PI,2000+CHDC;     ENABLE DATA CONTROL
003521  402000 003526'  09250              SETZM   TAPES
003522  722600 000001   09260              CONO    MT1,1;            ENABLE TAPE CONTROL FREE TRAP
003523  263740 000000   09270              DONE
                        09280
003524  350000 003526'  09290     CIO1:    AOS     TAPES;            SET RESTART FLAG
003525  263740 000000   09300              DONE
                        09310
003526  000000 000000   09320     TAPES:   Z;                        SET IF TAPE RESTART IN NEEDED
003527  000000 000000   09330     TCW:     Z;                        DC CONTROL WORD FOR TAPE
003530  777776 003530'  09340     TPDL:    XWD     -2,.;             PUSH LIST DURING INTERRUPT
003531  000000 000000   09350              Z
                        09360
                        09370     ,        TAPE RESTART ROUTINE
                        09380
003532  336000 002127'  09390     TRST:    SKIPN   DIP;              NO RESTART IF DISC USING DC
003533  336000 003526'  09400              SKIPN   TAPES
003534  263740 000000   09410              DONE;                     NOR IF NONE INDICATED
003535  712300 000170   09420              CONSZ   TTY,170;          SKIP IF NOT BUSY
003536  254000 003541'  09430              JRST    TRST1
003537  201040 000007   09440              MOVEI   B,"\";              THATS A BELL
003540  712140 000001   09450              DATAO   TTY,B
003541  332000 000327'  09460     TRST1:   SKIPE   DT.BUF;           SKIP IF NOTHING TO START
003542  260740 003475'  09470              TSX     CIO;              START TAPE I/O
003543  263740 000000   09480              DONE
                        09490
```

```
                                09510     ;        CHANNEL 6 INTERRUPTS: TTY IN, TY OUT, AND MAG TAPE
                                09520
003544  000000 000000           09530     CTYR:    Z
003545  712340 000010           09540              CONSO    TTY,10;          IS IT AN OUTPUT INTERRUPT?
003546  254000 003612'          09550              JRST     ST41;            MUST BE INPUT
003547  336000 000330'          09560              SKIPN    CO.BUF;          SKIP IF BUFFER ATTACHED
003550  254000 003572'          09570              JRST     ST40
003551  250040 003577'          09580              EXCH     B,ST31;          SAVE REGISTER
003552  134040 003600'          09590              ILDB     B,ST32;          GET NEXT BYTE
003553  322040 003557'          09600              JUMPE    B,ST33;          DISMISS IF END CHARACTER
003554  712140 000001           09610              DATAO    TTY,B;           OUTPUT CHARACTER
003555  250040 003577'          09620     ST34:    EXCH     B,ST31;          RESTORE
003556  254520 003544'          09630              JRST     12,@CTYR;        RETURN AND DISMISS
003557  712200 000206           09640     ST33:    CONO     TTY,200+CCTY;    TURN OFF OUTPUT FLAG
003560  250700 003601'          09650              EXCH     S,ST35
003561  250740 003574'          09660              EXCH     PP,ST22
003562  250200 003576'          09670              EXCH     E,ST24
003563  260740 010365'          09680              TSX      MBA
003564  332000 000330'          09690              SKIPE    CO.BUF;          ARE THERE MORE BUFFERS TO SEND
003565  260740 003602'          09700              TSX      BIO
003566  250740 003574'          09710              EXCH     PP,ST22
003567  250200 003576'          09720              EXCH     E,ST24
003570  250700 003601'          09730              EXCH     S,ST35
003571  254000 003555'          09740              JRST     ST34
                                09750
003572  712200 000206           09760     ST40:    CONO     TTY,200+CCTY
003573  254520 003544'          09770              JRST     12,@CTYR
                                09780
003574  777776 003574'          09790     ST22:    XWD      -2,.
003575  000000 000000           09800     ST23:    Z;                        PDL FOR INTERRUPT ROUTINE
003576  000000 000000           09810     ST24:    Z
003577  000000 000000           09820     ST31:    Z
003600  440700 000000           09830     ST32:    POINT    7,0;             POINTER TO OUTPUT TEXT
003601  777777 777777           09840     ST35:    DEC      -1
                                09850
                                09860     ;                START ROUTINE FOR CONSOLE TTY OUTPUT
                                09870
003602  550040 000330'          09880     BIO:     HRRZ     B,CO.BUF
003603  500040 010535'          09890              HLL      B,[POINT 7,0]
003604  271040 000001           09900              ADDI     B,1;             POINTER TO TEXT
003605  202040 003600'          09910              MOVEM    B,ST32;          INITIALIZE
003606  712200 000006           09920              CONO     TTY,CCTY;        ASSIGN CHANNEL
003607  134040 003600'          09930              ILDB     B,ST32
003610  712140 000001           09940              DATAO    TTY,B;           START OUT FIRST CHARACTER
003611  263740 000000           09950              DONE
                                09960
                                09970
```

```
                                09990     ;        INPUT INTERRUPT FROM CONSOLE TTY
                                10000
003612  712340  000040          10010     ST41:    CONSO   TTY,40;          SKIP IF TTY INPUT INTERRUPT
003613  254000  003405'         10020              JRST    TIP;             MUST BE TAPE
003614  250040  003664'         10030              EXCH    B,ST46
003615  250200  003665'         10040              EXCH    E,ST47
003616  250740  003672'         10050              EXCH    PP,ST52
003617  326200  003627'         10060              JUMPN   E,ST42;          JUMP IF WE ALREADY HAVE A BUFFER
003620  260740  010343'         10070              TSX     GETBUF
003621  254000  003657'         10080              JRST    ST45-1;          ITS TOUGH IF NO BUFFERS
003622  201040  000120          10090              MOVEI   B,↑D80;          LIMIT ON NUMBER OF INPUT CHARACTERS
003623  202040  003671'         10100              MOVEM   B,ST51
003624  542200  003666'         10110              HRRM    E,ST48;          SAVE BUFFER ADDRESS
003625  500200  010535'         10120              HLL     E,[POINT 7,0];   MAKE UP POINTER
003626  350000  000004          10130              AOS     E;               TEXT IS WORD ONE
003627  712040  000001          10140     ST42:    DATAI   TTY,B;           READ THE CHARACTER
003630  136040  000004          10150              IDPB    B,E;             DEPOSIT IN BUFFER
003631  376000  003671'         10160              SOSN    ST51;            COUNT INPUT CHARACTERS
003632  254000  003635'         10170              JRST    .+3;             NOT TOO MANY FOR THE BUFFER NOW
003633  302040  000215          10180              CAIE    B,215;           SKIP IF CARRIAGE RETURN
003634  254000  003660'         10190              JRST    ST45;            NOT YET - EXIT
                                10200
003635  250140  003667'         10210              EXCH    D,ST49
003636  201040  000000          10220              MOVEI   B,0
003637  200200  003666'         10230              MOVE    E,ST48;          GET BUFFER LOCATION
003640  350000  000004          10240              AOS     E;               TO TEXT
003641  202200  003670'         10250              MOVEM   E,ST50;          SAVE IT
003642  200140  010536'         10260              MOVE    D,[POINT 7,OPMSG]
                                10270
003643  134200  003670'         10280     ST43:    ILDB    E,ST50
003644  306200  000015          10290              CAIN    E,15;            SKIP IF NOT CARRIAGE RETURN
003645  254000  003652'         10300              JRST    ST44;            CR - END OF LINE
003646  136200  000003          10310              IDPB    E,D;             PUT CHARACTER IN MESSAGE
003647  350000  000001          10320              AOS     B
003650  305040  000036          10330              CAIGE   B,↑D30;          SKIP IF ENOUGH CHAR FOR MESSAGE LINE
003651  254000  003643'         10340              JRST    ST43;            GO FOR ANOTHER
                                10350
003652  202040  004562'         10360     ST44:    MOVEM   B,L.OPM;         COUNT OF CHARACTERS IN MESSAGE
003653  250140  003667'         10370              EXCH    D,ST49
003654  550200  003666'         10380              HRRZ    E,ST48;          GET BUFFER POINTER
003655  260740  010401'         10390              TSX     MBA1;            AND GIVE IT BACK
003656  201200  000000          10400              MOVEI   E,0;             RESET BUFFER SWITCH
003657  712200  001006          10410              CONO    TTY,CCTY+1000;   OFF WITH THE FLAG!
003660  250200  003665'         10420     ST45:    EXCH    E,ST47
003661  250040  003664'         10430              EXCH    B,ST46
003662  250740  003672'         10440              EXCH    PP,ST52
003663  254520  003544'         10450              JRST    12,@CTYR
                                10460
003664  000000  000000          10470     ST46:    Z;                       SAVE REG B
003665  000000  000000          10480     ST47:    Z;                       BYTE POINTER FOR INPUT,SAVE REG E, SIGNAL
003666  440700  000000          10490     ST48:    POINT   7,0;             CONSTANT IN LEFT, BUFFER POINTER IN RIGHT
003667  000000  000000          10500     ST49:    Z;                       SAVE REG D
003670  000000  000000          10510     ST50:    Z;                       HOLDS BYTE POINTER
003671  000000  000000          10520     ST51:    Z;                       COUNT OF INPUT CHARACTERS
003672  777776  003670'         10530     ST52:    XWD     -2,ST51-1;       PDL POINTER FOR INTERRUPT ROUTINE
                                10540
```

```
                         10560     ;                 USES 00X CLASS OF PROGRAMMED OPERATORS, ASSUMES AN OUTPUT
                         10570     ;                 POINTER IN C, CLOBBERS A AND B. CHARACTERS SHOULD BE ASCII
                         10580     ;                 AND MUST BE LEFT JUSTIFIED IN THE "FROM" LOCATION.
                         10590
003673  000000 000000    10600     PROP:   Z
003674  250040 000030    10610             EXCH    B,30;             SAVE FOR RECOVERY
003675  135040 010537'   10620             LDB     B,[POINT 3,40,5]; GET OP CODE
003676  322040 003716'   10630             JUMPE   B,XMTR;           ZERO IS THE TRANSMITTER
003677  307040 000004    10640             CAIG    B,4;              TEST RANGE
003700  254001 003701'   10650             JRST    .+1(B);           GO TO PROPER ROUTINE
003701  254000 003706'   10660             JRST    PR1
003702  254000 003727'   10670             JRST    INSR;             INSERT OP
003703  254000 003732'   10680             JRST    R2;               CHANGE STATE
003704  254000 003735'   10690             JRST    R3;               SEND CHAR STRING TO BUFFER
003705  254000 003740'   10700             JRST    R4;               HALT ROUTINE
                         10710
003706  550440 003673'   10720     PR1:    HRRZ    J,PROP
003707  307440 000000    10730             CAIG    J,INTBEG
003710  254000 003713'   10740             JRST    H3;               GO TO SIMULATE HALT 3
003711  201000 000075    10750             MOVEI   A,75
003712  254000 004162'   10760             JRST    KILL;             GO TO ELIMINATE THE USER
                         10770
003713  201040 000003    10780     H3:     MOVEI   B,3;              ILLEGAL UUO--NON-INTERPRETER
003714  202040 000040    10790             MOVEM   B,40
003715  254000 003740'   10800             JRST    R4
                         10810
003716  200040 010535'   10820     XMTR:   MOVE    B,[POINT 7,0]
003717  540040 000040    10830             HRR     B,40
003720  202040 002112'   10840             MOVEM   B,MISC;           POINTER FOR INPUT STRING
003721  135040 010540'   10850             LDB     B,[POINT 7,40,12]; GET "M" THE CHARACTER COUNT
003722  322040 003713'   10860             JUMPE   B,H3;             ZERO CHAR COUNT - PROBABLY AN ERROR
003723  134000 002112'   10870             ILDB    A,MISC
003724  136000 000002    10880             IDPB    A,C;              MOVE THE CHARACTERS
003725  367040 003723'   10890             SOJG    B,.-2
003726  254120 003673'   10900             JRST    2,@PROP;          RETURN FROM PROGRAMMED OPERATOR
                         10910
003727  200000 000040    10920     INSR:   MOVE    A,40;             GET THE CHARACTER
003730  136000 000002    10930             IDPB    A,C;              DEPOSIT IN OUTPUT STRING
003731  254120 003673'   10940             JRST    2,@PROP;          RETURN
                         10950
003732  550200 000040    10960     R2:     HRRZ    E,40;             GET DESTINATION STATE
003733  260740 010233'   10970             TSX     CHS.;             GO TO ROUTINE
003734  254120 003673'   10980             JRST    2,@PROP;          RETURN FROM PROGRAMMED OP
                         10990
003735  504200 000040    11000     R3:     HRL     E,40;             ADDRESS OF CONTROL WORD
003736  260740 010433'   11010             TSX     BJ9.1;            TO SEND ROUTINE
003737  254120 003673'   11020             JRST    2,@PROP
```

```
                        11040      ,                HALT ROUTINE
                        11050      ;
                        11060      ;                THIS ROUTINE EITHER HALTS ON ERROR OR ATTEMPTS
                        11070      ;                A RECOVERY OF THE SYSTEM FROM DRUM AND RESTART
                        11080      ;                OF JOSS FROM INITIAL CONDITIONS. APPROPRIATE ERROR
                        11090      ;                MESSAGES AND DUMPS ARE WRITTEN AT THE CONSOLE.
                        11100      ;                REGISTER 1 (B) HAS BEEN SAVED IN LOCATION 30.
                        11110
                        11120      R4:      FSW B
003740  200040 000000          MOVE B,SWITCH
003741  606040 020000   11130               TRNN     B,HALTS;         SKIP IF HALTING
003742  254000 003750'  11140               JRST     R4.1
003743  540040 000040   11150               HRR      B,40
003744  271040 777000   11160               ADDI     B,777000
003745  542040 003747'  11170               HRRM     B,R4.99;         SIMULATE LOCAL HALT
003746  200040 000030   11180               MOVE     B,30;            RECOVER REGISTER
003747  254200 777000   11190      R4.99:   JRST     4,777000
                        11200
                        11210
                        11220      ;        ATTEMPT RECOVERY FROM THE DRUM
                        11230
003750  700600 110000   11240      R4.1:    CONO     PI,110000;       KILL PI SYSTEM AND PARITY FLAG
003751  700200 210000   11250               CONO     APR,210000;      KILL PROCESSOR INTERRUPTS
003752  200040 010541'  11260               MOVE     B,[XWD YEAR,31]
003753  251040 000035   11270               BLT      B,35;            SAVE DATE AND TIME
003754  202100 000042   11280               MOVEM    C,42
003755  202140 000043   11290               MOVEM    D,43
003756  200057 000000   11300               MOVE     B,0(17)
003757  202040 000114   11310               MOVEM    B,SA+2;          @17
003760  550040 000000   11320               HRRZ     B,0
003761  200041 000000   11330               MOVE     B,0(B)
003762  202040 000115   11340               MOVEM    B,SA+3;          @0
003763  200051 777777   11350               MOVE     B,-1(J)
003764  202040 000113   11360               MOVEM    B,SA+1;          @J
003765  254000 000000   11370               JRST     RJD;             GO TO READ DRUM
```

```
                          11390     ;        CHANNEL 7 INTERRUPTS
                          11400
003766  000000 000000     11410     APRR:    Z
003767  700340 001000     11420              CONSO    APR,1000;        IT IS THE CLOCK ISNT IT?
003770  254000 004141'    11430              JRST     CHECK;           PARITY, NOEX MEM, PDL OV
003771  350000 000042'    11440     CLOCK:   AOS      TIME;            BUMP SYSTEM CLOCK
003772  376000 000000     11450              SOSN     D.TIME;          TIME OUT THE DISC ROUTINES
003773  264000 000000     11460              JSR      DISC.C;          TRY TO RECOVER DROPPED TRAP
003774  700200 001007     11470              CONO     APR,1000+CAPR;   TURN OFF THE FLAG
003775  750200 000001     11480              CONO     500,1;           PULSE THE "JOSS HERE" FLOP
003776  350000 000043'    11490              AOS      T.CU;            BUMP THE CURRENT USER CLOCK
003777  350000 000034'    11500              AOS      TICK;            INCREMENT TICKER
004000  350000 000044'    11510              AOS      SECONDS;         TICK COUNT (SOURCE OF TIMER)
004001  250040 000034'    11520              EXCH     B,TICK
004002  305040 000074     11530              CAIGE    B,!D60;          SKIP IF ONE SECOND
004003  254000 004051'    11540              JRST     CL3
004004  202100 000035'    11550              MOVEM    C,CL9;           SAVE REGISTER
004005  350040 000033'    11560              AOS      B,SEC;           BUMP SECOND COUNT
004006  602040 000003     11570              TRNE     B,3;             SKIP IF NOT 4 SEC BOUNDARY
004007  254000 004063'    11580              JRST     CL6;             GO TO BEEP SIGNALLED CONSOLES
004010  550040 003766'    11590     CL8:     HRRZ     B,APRR;          FETCH INTERRUPT ADDRESS
004011  316040 000040'    11600              CAMN     B,CL12;          SKIP IF DIFFERENT THAN LAST TIME
004012  254000 004075'    11610              JRST     CL14
004013  542040 000040'    11620              HRRM     B,CL12;          SAVE LOC COUNTER
004014  402000 000041'    11630     CL18:    SETZM    CL13;            ZERO COUNT
                          11640     CL5:     FSW B
004015  200040 003740'        MOVE B,SWITCH
004016  602040 400000     11650              TRNE     B,BEEPS;         DO WE WANT TO BEEP
004017  264000 000000     11660              JSR      C30;             GO MAKE NOISE
004020  606040 200000     11670              TRNN     B,SDS;           SKIP IF SHUTTING DOWN
004021  254000 004025'    11680              JRST     CL4
004022  200040 000033'    11690              MOVE     B,SEC
004023  307040 000005     11700              CAIG     B,!D5;           BEEP FOR 5 SEC EACH MIN
004024  264000 004017'    11710              JSR      C30
```

```
004025  200040 002214'  11730    CL4:   MOVE   B,CT46;        INPUT INTERRUPTS
004026  221040 003720   11740           IMULI  B,↑D2000
004027  230040 002160'  11750           IDIV   B,USERS
004030  264000 004267'  11760           JSR    CVTL1
004031  350002 002356'  11770           AOS    IINT(C)
004032  200040 002215'  11780           MOVE   B,CT47;        OUTPUT INTERRUPTS
004033  221040 003720   11790           IMULI  B,↑D2000
004034  230040 002160'  11800           IDIV   B,USERS
004035  264000 004267'  11810           JSR    CVTL1
004036  350002 002370'  11820           AOS    OINT(C)
004037  200040 002214'  11830           MOVE   B,CT46
004040  270040 002215'  11840           ADD    B,CT47
004041  221040 003720   11850           IMULI  B,↑D2000
004042  230040 002160'  11860           IDIV   B,USERS
004043  264000 004267'  11870           JSR    CVTL1
004044  350002 002402'  11880           AOS    TINT(C)
004045  402000 002214'  11890           SETZM  CT46
004046  402000 002215'  11900           SETZM  CT47
004047  200100 000035'  11910           MOVE   C,CL9
004050  201040 000000   11920           MOVEI  B,0;           ZERO THE TICKER
004051  250040 000034'  11930    CL3:   EXCH   B,TICK
004052  250040 000043'  11940           EXCH   B,T.CU
004053  311040 000326'  11950           CAML   B,T.MAX
004054  254000 004057'  11960           JRST   CL2;           YES TIME IS UP
004055  250040 000043'  11970    CL1:   EXCH   B,T.CU;        NO INTERRUPT YET
004056  254520 003766'  11980           JRST   12,@APRR
                         11990
004057  336000 005471'  12000    CL2:   SKIPN  SS99
004060  336000 002107'  12010           SKIPN  DMBY;          NOT MUCH POINT IF DRUM IS BUSY
004061  350000 002111'  12020           AOS    COMEBACK;      SET INTERPRETER RETURN FLAG
004062  254000 004055'  12030           JRST   CL1
                         12040
```

```
                                12060
                                12070     ;       BEEP SIGNALLED CONSOLES EACH 4 SECONDS
                                12080
004063  200040  000000          12090     CL6:    MOVE    B,CONSOL
004064  350000  000001          12100             AOS     B;              START WITH FIRST JOSS CONSOLE
004065  307040  000120          12110     CL7:    CAIG    B,N.S*S.M;
004066  254000  004010'         12120             JRST    CL8;            FINISHED - ALL STATIONS TESTED
004067  336001  005667'         12130             SKIPN   TTT(B)
004070  344040  004065'         12140             AOJA    B,CL7;          NO SIGNAL - TRY NEXT
004071  246040  777777          12150             LSHC    B,-1;           GET "TO" CONSOLE NUMBER
004072  264000  000000          12160             JSR     C32;            BEEP THE CONSOLE
004073  246040  000001          12170             LSHC    B,1
004074  344040  004065'         12180             AOJA    B,CL7;          CONTINUE SEARCH
                                12190
                                12200     ;       DETECT INDIRECT ADDRESS LOOPS
                                12210
004075  350040  000041'         12220     CL14:   AOS     B,CL13;         COUNT INTERRUPTS WITH SAME PC
004076  307040  000012          12230             CAIG    B,↑D10
004077  254000  004015'         12240             JRST    CL5;            NOT EXCESSIVE YET
                                12250
                                12260     ;       TEN EVEN SECONDS WITH THE SAME PC - TEST FOR ADDRESS LOOP
                                12270
004100  202140  000036'         12280             MOVEM   D,CL10
004101  201100  000100          12290             MOVEI   C,100
004102  202100  000037'         12300             MOVEM   C,CL11;         INDIRECT DEPTH COUNT
004103  200120  003766'         12310             MOVE    C,@APRR;        INITIAL WORD
004104  200140  000002          12320     CL15:   MOVE    D,C
004105  135040  004127'         12330             LDB     B,IRPTR;        GET INDEX REGISTER ADDRESS
004106  322040  004120'         12340             JUMPE   B,CL16;         JUMP IF NO INDEX
004107  305040  000004          12350             CAIGE   B,4
004110  550101  000033'         12360             HRRZ    C,TICK-1(B);    GET STORED REGISTER
004111  301040  000004          12370             CAIL    B,4
004112  550101  000000          12380             HRRZ    C,(B)
004113  270100  000003          12390             ADD     C,D;            ADDRESS PLUS IR CONTENTS
004114  301040  000004          12400             CAIL    B,4
004115  200102  000000          12410             MOVE    C,(C);          GET ADDRESSED WORD
004116  305040  000004          12420             CAIGE   B,4
004117  200101  000033'         12430             MOVE    C,TICK-1(B);    IF SAVED
004120  607140  000020          12440     CL16:   TLNN    D,20;           SKIP IF INDIRECT BIT SET
004121  254000  004125'         12450             JRST    CL17;           END OF ADDRESS CHAIN
004122  373000  000037'         12460             SOSLE   CL11;           COUNT INDIRECT LEVELS
004123  254000  004104'         12470             JRST    CL15
004124  040000  000033          12480     H33:    HALT    33;             INDIRECT ADDRESS LOOP
                                12490
004125  200140  000036'         12500     CL17:   MOVE    D,CL10;         RESTORE REGISTER D
004126  254000  004014'         12510             JRST    CL18;           WASNT A LOOP
                                12520
004127  220400  000002          12530     IRPTR:  POINT   4,C,17;         POINTER TO INDEX FIELD
                                12540
```

CONSTANTS

```
004130  201004 020100    12560    BLANKS: ASCII   ?     ?
004131  201004 020100    12570            ASCII   ?     ?
004132  000000 000000    12580    ZERO:   0
004133  064321 506432    12590    CRS:    OCT     064321506432;   FIVE CARRIAGE RETURNS
004134  064240 000000    12600    CRLF:   OCT     064240000000;   CR, LF, AND ZEROS
                         12610    BELF:   ASCII   ?\
004135  034240 700000    12620    \?
                         12630
004136  440700 000000    12640    C3:     POINT   7,INITIAL
004137  440700 000000    12650    C4:     POINT   7,JOBNO
004140  220600 004015'   12660    C5:     POINT   6,SWITCH,↑D17;  POINTER TO DISPLAY LINE NUMBER
```

```
                                 12680     ;        TRY TO SAVE THE SYSTEM FROM HARDWARE ERRORS
                                 12690
                                 12700     ;                NOTE THAT PROPER RECOVERY FROM MACHINE ERRORS
                                 12710     ;                DEPENDS, IN PART, ON A SPECIFIC LOADING ORDER
                                 12720     ;                FOR JOSS COMPONENTS.  IN PARTICULAR, THE ORDER
                                 12730     ;                SHOULD BE: ARITH,DISTRIB, DISC, SU, CPU, DDT, IU.
                                 12740
004141  700340 010000            12750     CHECK:  CONSO   APR,10000;       SKIP IF NOEX MEM
004142  254000 004151'           12760             JRST    CHECK1
004143  550440 003766'           12770             HRRZ    J,APRR;          GET LOCATION OF ERROR
004144  307440 003707'           12780             CAIG    J,INTBEG
004145  040000 000027            12790     H27:    HALT    27;              NOT A REFERENCE FROM JOE
004146  700200 110007            12800             CONO    APR,110007;      RESET FLAGS
004147  201000 000072            12810             MOVEI   A,72
004150  254400 004162'           12820             JRST    10,KILL
                                 12830
004151  550440 003766'           12840     CHECK1: HRRZ    J,APRR;          PICK UP OFFENDING LOCATION
004152  700340 020000            12850             CONSO   APR,20000;       SKIP IF CPA ILLEGAL OP
004153  254000 004176'           12860             JRST    CHECK2
004154  550440 003766'           12870             HRRZ    J,APRR
004155  307440 004144'           12880             CAIG    J,INTBEG
004156  040000 000031            12890     H31:    HALT    31;              MEMORY PROTECT VIOLATION--NON-INTERPRETER
004157  700200 120007            12900             CONO    APR,120007
004160  200000 000074            12910             MOVE    A,74
004161  254400 004162'           12920             JRST    10,KILL
                                 12930
                                 12940     ,        DISCARD A BAD USER BLOCK
                                 12950
004162  200740 010542'           12960     KILL:   MOVE    PP,[XWD -N.PP1,PP1-1];  RESTORE MONITOR PUSH REGIS
004163  200700 000021'           12970             MOVE    S,CUI
004164  135040 000267'           12980             LDB     B,S.DU
004165  322040 004174'           12990             JUMPE   B,CHK1
004166  402000 002110'           13000             SETZM   DKBY;            DISABLE POSSIBLE DISC ACTION
004167  336700 002131'           13010             SKIPN   S,DQ
004170  254000 004174'           13020             JRST    CHK1
004171  350000 002110'           13030             AOS     DKBY;            ENABLE NEXT FOR DISC
004172  275700 000141'           13040             SUBI    S,S.Q
004173  020000 000007            13050             CHS     COM.S
004174  260740 004211'           13060     CHK1:   TSX     OGKM
004175  254000 004446'           13070             JRST    QUIT
                                 13080
                                 13090     ;        HANDLE PARITY ERRORS
                                 13100
004176  700740 100000            13110     CHECK2: CONSO   PI,100000;       SKIP IF PARITY ERROR
004177  040000 000002            13120     H2:     HALT    2;               PDL OVERFLOW
004200  700600 100000            13130             CONO    PI,100000;       TURN OFF FLAG
004201  332000 002125'           13140             SKIPE   CU;              SKIP IF NO CURRENT USER
004202  254000 004206'           13150             JRST    CHK2
004203  350000 002167'           13160             AOS     CT15;            RECORD IN LOG
004204  256000 004162'           13170             XCT     KILL;            RESTORE PUSH REGISTER
004205  254400 007264'           13180             JRST    10,SIGPR;        GO TO MPL AND HOPE
                                 13190
004206  201000 000076            13200     CHK2:   MOVEI   A,76
004207  202456 002572'           13210             MOVEM   J,MINT(S);       DISPLAY OFFENDING LOCATION
004210  254400 004162'           13220             JRST    10,KILL;         DISCARD USER
```

```
                                13240      ;        ON ENTRY REGISTER A CONTAINS THE GRONK CODE
                                13250      ;        A GRONK IS REPORTED AT THE CONSOLE TTY BY A LINE OF DATA
                                13260      ;        GIVING THE GRONK CODE (ONE OF THOSE LISTED BELOW), THE
                                13270      ;        NUMBER OF THE CONSOLE GRONKED, AND TWO DATA WORDS WITH
                                13280      ;        CONTENTS AS LISTED BELOW.
                                13290
                                13300      ;        71- CKSUM FAILURE; 1) CORRECT, 2) INCORRECT
                                13310      ;                IF THE DP OR DR HARDWARE DETECTS AN ERROR THE
                                13320      ;                CONTENTS OF THE LEFT AND RIGHT HALVES OF
                                13330      ;                THE INCORRECT CHECKSUM WORD CONTAIN THE RESULTS
                                13340      ;                OF CONI INSTRUCTIONS ON THE DP AND DR.
                                13350      ;                  FAILURE TO COMPARE FINAL DATAI REGISTER CONTENTS
                                13360      ;                WITH COMPUTED VALUES ALSO TAKE THIS EXIT. IN THIS
                                13370      ;                CASE THE LEFT HALF WILL BE ZERO AND THE RIGHT
                                13380      ;                HALF WILL SHOW NO ERROR FLAG.
                                13390      ;        72- NOEX MEM FROM INTERPRETER
                                13400      ;        73- INITIALS CHECK; 1) BAD INITIALS, 2) GOOD
                                13410      ;        74- CPA ILL OP = PROTECT VIOLATION ON USER BLOCK REFERENCE
                                13420      ;        75- ILLEGAL UUO FROM INTERPRETER CODE
                                13430      ;        76- PARITY ERROR;        2) OFFENDING LOCATION
                                13440
004211  200700 000021'          13450      OGKM:    MOVE    S,CUI
004212  200440 000000           13460               MOVE    J,A
004213  260740 006274'          13470               TSX     OFFNQ;           DO OFF PROCEDURE
004214  260740 010343'          13480               TSX     GETBUF
004215  040000 000026           13490      H26:     HALT    26;              IF SO WE ARE IN TERRIBLE TROUBLE
004216  200100 010543'          13500               MOVE    C,[POINT 7,1(E)]
                                13510               XMT 3,BELF
                                    OPDEF X[3B12]
004217  000140 004135'          X BELF
004220  303440 000100           13520               CAILE   J,100;           SKIP IF A MONITOR CODE
004221  200451 000000           13530               MOVE    J,0(J);          GET CONTENTS OF JWS PUSH LIST
004222  200000 000011           13540               MOVE    A,J
004223  260740 004274'          13550               TSX     OCTW
                                13560               XMT 2,BLANKS
                                    OPDEF X[2B12]
004224  000100 004130'          X BLANKS
004225  200000 000021'          13570               MOVE    A,CUI
004226  260740 006700'          13580               TSX     OCT
                                13590               XMT 10,GKM
                                    OPDEF X[10B12]
004227  000400 004247'          X GKM
                                13600               XMT 2,BLANKS
                                    OPDEF X[2B12]
004230  000100 004130'          X BLANKS
```

```
OUTPUT A GRONK MESSAGE

004231  200700 000021'   13620              MOVE    S,CUI
004232  200016 002642'   13630              MOVE    A,SUM(S)
004233  260740 004274'   13640              TSX     OCTW
                         13650              XMT 2,BLANKS
                             OPDEF X[2B12]
004234  000100 004130'   X BLANKS
004235  200016 002572'   13660              MOVE    A,MINT(S);      POSSIBLE BAD CHECKSUM
004236  260740 004274'   13670              TSX     OCTW
                         13680              XMT 3,BELF
                             OPDEF X[3B12]
004237  000140 004135'   X BELF
                         13690              XMT 3,CRLF
                             OPDEF X[3B12]
004240  000140 004134'   X CRLF
004241  571700 777777    13700              HRREI   S,-1
004242  260740 010305'   13710              TSX     PUTB
004243  200700 000021'   13720              MOVE    S,CUI
004244  402016 002572'   13730              SETZM   MINT(S)
004245  350000 002224'   13740              AOS     CT21;           COUNT ERROR
004246  263740 000000    13750              DONE
                         13760
004247  036172 247634    13770      GKM:    ASCII   ?\GRONK!\?;             8 CHARACTERS
004250  455020 700000
```

```
                                13790     ;        TABLE FOR CONVERSION TO LOGARITHMIC RANGE
                                13800
004251  000000 072460           13810     CVTLOG:  DEC      500*60
004252  000000 027340           13820              DEC      200*60
004253  000000 013560           13830              DEC      100*60
004254  000000 005670           13840              DEC      50*60
004255  000000 002260           13850              DEC      20*60
004256  000000 001130           13860              DEC      10*60
004257  000000 000454           13870              DEC      5*60
004260  000000 000170           13880              DEC      2*60
004261  000000 000074           13890              DEC      1*60
                                13900     CVTS=.-CVTLOG
                                13910
                                13920     ,        CONVERT TO LOG RANGE
                                13930
004262  000000 000000           13940     CVTL:    Z
004263  201100 000011           13950              MOVEI    C,CVTS
004264  311042 004250'          13960              CAML     B,CVTLOG-1(C)
004265  367100 004264'          13970              SOJG     C,.-1
004266  254120 004262'          13980              JRST     2,@CVTL
                                13990
                                14000     ,        TRAP TIME COPY OF CONVERT ROUTINE
                                14010
004267  000000 000000           14020     CVTL1:   Z
004270  201100 000011           14030              MOVEI    C,CVTS
004271  311042 004250'          14040              CAML     B,CVTLOG-1(C)
004272  367100 004271'          14050              SOJG     C,.-1
004273  254120 004267'          14060              JRST     2,@CVTL1
                                14070
                                14080     ,        CONVERT A WORD TO OCTAL ASCII
                                14090     ,          CLOBBERS D,B, INPUT IS A.
                                14100
004274  201140 000014           14110     OCTW:    MOVEI    D,↑D12
004275  200040 000000           14120              MOVE     B,A
004276  402000 000000           14130     OCTW1:   SETZM    A
004277  246000 000003           14140              LSHC     A,3
004300  435000 000060           14150              ORI      A,60
004301  136000 000002           14160              IDPB     A,C
004302  367140 004276'          14170              SOJG     D,OCTW1
004303  263740 000000           14180              DONE
```

```
                          14200     ;        DECODE INTERPRETER SIGNALS
                          14210
004304  250700 000021'    14220     MONENT:  EXCH    S,CUI;          RESTORE THE USER INDEX
004305  200100 000043'    14230              MOVE    C,T.CU
004306  135140 000262'    14240              LDB     D,S.TM
004307  274100 000003     14250              SUB     C,D;            TICKS SINCE LAST ENTRY
004310  272116 002522'    14260              ADDM    C,USTAT(S);     ACCUMULATE IN USER STAT BLOCK
004311  272100 000000     14270              ADDM    C,COMTIM;       UPDATE USERS COMPUTE TIME
004312  272100 002165'    14280              ADDM    C,CT13;         UPDATE COMPUTE TIME THIS MINUTE
004313  272100 002213'    14290              ADDM    C,CT44;         ACCUMULATE TIME COMPUTING
004314  135000 000264'    14300              LDB     A,S.BLOCK;      NUMBER OF CORE BLOCKS
004315  220000 000002     14310              IMUL    A,C
004316  272000 000000     14320              ADDM    A,SPARE3;       INTEGRATE TIME X CORE SIZE="CHARGE UNITS"
004317  403000 000043'    14330              SETZB   A,T.CU
004320  137000 000266'    14340              DPB     A,S.INR;        RESET POSSIBLE IN SIGNAL
004321  270100 000003     14350              ADD     C,D
004322  137000 000262'    14360              DPB     A,S.TM
004323  250740 000020'    14370              EXCH    PP,PPSAV;       RESTORE MONITOR PP REGISTER
004324  301040 000021     14380              CAIL    B,L.ETSW;       CHECK FOR WITHIN LIMIT.
004325  040000 000005     14390     H5:      HALT    5;              BAD SIGNAL FROM JOE
004326  254021 004327'    14400              JRST    @ENTSW(B);      INTERPRET ENTRY.
                          14410
                          14420     ;        INTERPRETER SIGNAL TYPES
                          14430
004327  254000 004372'    14440     ENTSW:   JRST    SU;             0 - SWITCH TO USER
004330  254000 004430'    14450              JRST    BUFFER;         1 - GET A BUFFER
004331  254000 004444'    14460              JRST    REBUF;          2 - RETURN A BUFFER
004332  254000 004366'    14470              JRST    TL;             3 - TRANSMIT BUFFER TO USER
004333  254000 004370'    14480              JRST    TLSU;           4 - TRANSMIT AND SWITCH TO USER
004334  254000 004355'    14490              JRST    CONT;           5 - BREAK POINT RESPONSE
004335  254000 004604'    14500              JRST    DREQ;           6 - REQUEST DISC
004336  254000 004616'    14510              JRST    DCONT;          7 - CONTINUE DISC ACTION
004337  254000 004625'    14520              JRST    DCOMP;          8 - DISC ACTION COMPLETE
004340  254000 004760'    14530              JRST    MORCOR;         9 - GET ANOTHER BLOCK OF CORE
004341  254000 005013'    14540              JRST    MORCI;          10 - IMPERATIVE MORE CORE
004342  254000 004501'    14550              JRST    PAGE;           11 - MAKE UP AND SEND PAGE HEADING
004343  254000 004446'    14560              JRST    QUIT;           12 - GET USER OFF
004344  254000 004726'    14570              JRST    RCOR;           13 - RETURN BLOCKS OF CORE
004345  254000 004416'    14580              JRST    PAUSE;          14 - PAUSE FOR SIGNALS AND (E) SECONDS
004346  254000 004420'    14590              JRST    WAIT;           15 - PAUSE (E) SECONDS
                          14600     L.ETSW=.-ENTSW+1;                LENGTH OF THE SWITCH
004347  000000 000000     14610              Z;                      SPARE
                          14620
                          14630     ;        RETURN TO INTERPRETER
                          14640
004350  201300 002150'    14650     MONEXIT:MOVEI   G,INTENT;        INTERPRETER CONTEXT
004351  250700 000021'    14660     MONE1:   EXCH    S,CUI
004352  250740 000020'    14670              EXCH    PP,PPSAV;       SAVE MONITOR PP REGISTER
004353  402000 002111'    14680              SETZM   COMEBACK
004354  254026 000000     14690              JRST    @0(G);          GO TO PROPER INTERPRETER
```

```
                          14710     ,        COMPUTATION BREAK
                          14720
004355  315100 000326'    14730     CONT:    CAMGE    C,T.MAX;        HAS HE HIT MAX?
004356  254000 004361'    14740              JRST     CONT1;          NO
004357  020000 000007     14750              CHS      COM.S;          PUT CURRENT USER ON END OF LIST
004360  254000 007264'    14760              JRST     SIGPR1
                          14770
004361  350000 002113'    14780     CONT1:   AOS      CSS;            SET  TOP OF QUEUE SWITCH FOR CHS
004362  137100 000262'    14790              DPB      C,S.TM
004363  020000 000007     14800              CHS      COM.S;          PUT ON TOP OF COMPUTE QUEUE
004364  402000 002113'    14810              SETZM    CSS;            RESET SWITCH
004365  254000 007264'    14820              JRST     SIGPR
                          14830
                          14840     ,        TRANSMIT LINE TO USER
                          14850
004366  260740 010305'    14860     TL:      TSX      PUTB;           ATTACH BUFFER TO USER
004367  254000 004350'    14870              JRST     MONEXIT;          GO BACK TO INTERPRETER
004370  260740 010305'    14880     TLSU:    TSX      PUTB;           ATTACH BUFFER TO USER
004371  254000 004404'    14890              JRST     SU1
                          14900
                          14910     ;        SWITCH CONSOLE TO USER
                          14920
004372  311700 004063'    14930     SU:      CAML     S,CONSOL;       SKIP IF TTY CONSOLE
004373  254000 004401'    14940              JRST     SU.5
004374  260740 010343'    14950              TSX      GETBUF
004375  254000 004414'    14960              JRST     SU3;            NO BUFFER--GO TO WAIT
004376  515040 574000     14970              HRLZI    B,574000;       LEFT ARROW CHARACTER
004377  202044 000003     14980              MOVEM    B,3(E);         STASH IN BUFFER
004400  260740 010305'    14990              TSX      PUTB;           TRANSMIT TO USER
004401  550216 000331'    15000     SU.5:    HRRZ     E,S.BUF(S);     PICK UP BUFFER HEADER
004402  402000 000000     15010              SETZM    RISIG;          RESET "IN" SIGNAL.
004403  322200 004406'    15020              JUMPE    E,SU2;          JUMP IF NO BUFFER ATTACHED.
004404  020000 000016     15030     SU1:     CHS      DSU.S;          PUT ON END ON DSU QUEUE.
004405  254000 007264'    15040              JRST     SIGPR1;         TO PROCESS SIGNALS.
                          15050
004406  260740 010343'    15060     SU2:     TSX      GETBUF
004407  254000 004414'    15070              JRST     SU3
004410  504200 000004     15080              HRL      E,E
004411  202216 000331'    15090              MOVEM    E,S.BUF(S);     LINK BUFFER TO USER
004412  260740 006560'    15100              TSX      SG;             SWITCH GREEN
004413  254000 007264'    15110              JRST     SIGPR1
                          15120
004414  020000 000017     15130     SU3:     CHS      ABG.S;          TO WAIT BUFFER FOR GREEN QUEUE.
004415  254000 007264'    15140              JRST     SIGPR
                          15150
```

```
                                15170     ;        WAIT FOR SIGNALS AND TIME
                                15180
004416  201040 000001           15190     PAUSE:   MOVEI    B,1
004417  137040 000271'          15200              DPB      B,S.SIG
                                15210
                                15220     ;        DELAY USER FOR  (E)  SECONDS
                                15230
004420  020000 000020           15240     WAIT:    CHS      QP.S;            TO PAUSE STATE
004421  260740.006421'          15250              TSX      ISEC
004422  270040 000004           15260              ADD      B,E
004423  303040 250600           15270              CAILE    B,↑D24*↑D3600
004424  275040 250600           15280              SUBI     B,↑D24*↑D3600
004425  332000 000004           15290              SKIPE    E;               NO TIME SPECIFIED
004426  506056 000141'          15300              HRLM     B,S.Q(S);        SAVE WAKEUP TIME IN QUEUE WORD
004427  254000 007264'          15310              JRST     SIGPR
```

```
                          15330     ;        GET A BUFFER
                          15340
004430  200200 000323'    15350     BUFFER:  MOVE    E,N.CK;          GET THE CHOKE NUMBER
004431  550056 000331'    15360              HRRZ    B,S.BUF(S);      GET TOP BUFFER LOCATION
004432  322040 004437'    15370              JUMPE   B,BU2;           OK - GET A BUFFER FOR HIM
004433  550041 000000     15380              HRRZ    B,0(B);          GET NEXT ON LIST
004434  367200 004432'    15390              SOJG    E,.-2;           JUMP IF HE SHOULD NOT BE CHOKED
004435  020000 000013     15400              CHS     CK.S;            HE HAS ENOUGH ALREADY--CHOKE UP
004436  254000 007264'    15410              JRST    SIGPR1
                          15420
004437  260740 010343'    15430     BU2:     TSX     GETBUF
004440  254000 004442'    15440              JRST    .+2;             NO BUFFER
004441  254000 004350'    15450              JRST    MONEXIT
004442  020000 000005     15460     BUF1:    CHS     UC.S;            PUT ON ABR-UC QUEUE
004443  254000 007264'    15470              JRST    SIGPR1
                          15480
                          15490     ;        RETURN A BUFFER
                          15500
004444  260740 010401'    15510     REBUF:   TSX     MBA1;            PUT BUFFER ON AVAILABLE
004445  254000 004350'    15520              JRST    MONEXIT;         GO BACK TO INTERPRETER
```

```
                                15540      ,        FINAL "OFF" FOR USER
                                15550
004446  135100 000264'          15560      QUIT:    LDB     C,S.BLOCK
004447  200140 010544'          15570               MOVE    D,[POINT 6,CT48A]
004450  134040 000003           15580               ILDB    B,D
004451  367100 004450'          15590               SOJG    C,.-1
004452  275040 000001           15600               SUBI    B,1
004453  137040 000003           15610               DPB     B,D;            DECREMENT BLOCK SIZE DISPLAY
004454  402016 002572'          15620               SETZM   MINT(S)
004455  135040 000263'          15630               LDB     B,S.COR
004456  322040 004465'          15640               JUMPE   B,Q2;           NOT IN CORE--MUST BE A GRONK
004457  275040 000020           15650               SUBI    B,BBLOCK;       COMPUTE PHYSICAL LOCATION
004460  135100 000264'          15660               LDB     C,S.BLOCK;      GET NUMBER OF BLOCKS
004461  402001 000272'          15670               SETZM   CORE(B) ;       RESET CORE MAP
004462  350000 000001           15680               AOS     B
004463  367100 004461'          15690               SOJG    C,.-2;          FOR ALL ASSIGNED BLOCKS
004464  370000 002202'          15700               SOS     CT39;           DECREMENT # OF IN CORE USERS
004465  135000 000261'          15710      Q2:      LDB     A,S.STA
004466  402016 000211'          15720               SETZM   S.S(S) ;        ZERO HIS STATUS
004467  137000 000261'          15730               DPB     A,S.STA;        EXCEPT STATE
004470  201000 000001           15740               MOVEI   A,1
004471  137000 000264'          15750               DPB     A,S.BLOCK;      INITIAL BLOCK SIZE
004472  020000 000023           15760               CHS     OF.S
                                15770               FSW D
004473  200140 004140'            MOVE D,SWITCH
004474  606140 100000           15780               TRNN    D,OFFS;         DONT RE-ENABLE IF BLASTING OFF
004475  264000 000000           15790               JSR     C31
004476  200140 010545'          15800               MOVE    D,[ASCII ?OFF-?] ;    DISPLAY OFFS ON CONSOLE TTY
004477  260740 006664'          15810               TSX     DOF
004500  254000 007264'          15820               JRST    SIGPR1
                                15830
```

```
                        15850      ;        SEND PAGE HEADING
                        15860
                        15870      ;                OUTPUT PAGE HEADING LINE TO USER. ON INPUT  REGISTER E
                        15880      ;                CONTAINS A BUFFER ADDRESS
                        15890
004501  200100 010546'  15900      PAGE:    MOVE    C,[POINT 7,3(E)]; POINTER TO BUFFER BEGINNING
004502  331000 000004   15910               SKIPL   E;               PAGED ALREADY!
004503  010000 000014   15920               INS     14;              FORM FEED
004504  200000 000031'  15930               MOVE    A,HR
004505  260740 006410'  15940               TSX     CTDR;            CONVERT HOURS WITH LEADING BLANKS
004506  010000 000072   15950               INS     ":";               INSERT A ":"
004507  200000 000032'  15960               MOVE    A,MIN
004510  260740 006403'  15970               TSX     CTD;             CONVERT MINUTES WITH LEADING ZEROS
                        15980               XMT 2,BLANKS
                           OPDEF X[2B12]
004511  000100 004130'  X BLANKS
                        15990               XMT 12,DATE
                           OPDEF X[12B12]
004512  000500 000023'  X DATE
004513  010000 000043   16000               INS     "#"
004514  200000 000016   16010               MOVE    A,S
004515  260740 004715'  16020               TSX     BTA;             CONVERT AND OUTPUT CONSOLE #
                        16030               XMT 2,BLANKS
                           OPDEF X[2B12]
004516  000100 004130'  X BLANKS
                        16040               XMT 5,INITIALS
                           OPDEF X[5B12]
004517  000240 004136'  X INITIALS
                        16050               XMT 5,JOBNO
                           OPDEF X[5B12]
004520  000240 004137'  X JOBNO
004521  311700 004372'  16060               CAML    S,CONSOL;        SKIP IF TTY CONSOLE
                        16070               XMT 4,BLANKS;          CENTER PAGE NUMBER FOR JOSS CONSOLE
                           OPDEF X[4B12]
004522  000200 004130'  X BLANKS
004523  350040 000000   16080               AOS     B,PAGNO;         FETCH AND INCREMENT PAGE NUMBER
004524  305040 000144   16090               CAIGE   B,^D100;         CENTER PAGE NUMBER A LITTLE
                        16100               XMT 1,BLANKS
                           OPDEF X[1B12]
004525  000040 004130'  X BLANKS
004526  010000 000133   16110               INS     "["
004527  200000 004523'  16120               MOVE    A,PAGNO;         REMEMBER REG. A IS ZERO
004530  260740 004715'  16130               TSX     BTA;             CONVERT TO ASCII BASE TEN
004531  010000 000135   16140               INS     "]"
                        16150               XMT 2,BLANKS
                           OPDEF X[2B12]
004532  000100 004130'  X BLANKS
004533  260740 004535'  16160               TSX     HDM;             MAKE UP HEADING MESSAGE
004534  254000 004366'  16170               JRST    TL;              TRANSMIT THE LINE
```

```
MAKE UP PAGE HEADING MESSAGE

                                16190     HDM:     FSW D;       FETCH DATA SWITCHES
004535  200140 004473'              MOVE D,SWITCH
004536  602140 000040           16200              TRNE    D,PRMES;         SKIP IF NO PRIORITY MESSAGE
004537  254000 004551'          16210              JRST    PG4
004540  606140 200000           16220              TRNN    D,SDS;           SKIP IF SHUTTING DOWN
004541  254000 004544'          16230              JRST    PG3
                                16240              XMT 34,HSDM; PUT SHUTDOWN MESSAGE IN HEADER
                                      OPDEF X[34B12]
004542  001600 004571'          X HSDM
004543  254000 004557'          16250              JRST    PG1
004544  606140 040000           16260     PG3:     TRNN    D,DCLOB
004545  254000 004550'          16270              JRST    PG2
                                16280              XMT 30,DCLOBM;         SEND DISC CLOBBERED MESSAGE
                                      OPDEF X[30B12]
004546  001400 004577'          X DCLOBM
004547  254000 004557'          16290              JRST    PG1
004550  602140 000001           16300     PG2:     TRNE    D,DAM;           SKIP IF NO DISPLAY
004551  336040 004562'          16310     PG4:     SKIPN   B,L.OPM;         IS THERE A MESSAGE
004552  254000 004557'          16320              JRST    PG1;             NOTHING MORE FOR THIS LINE
004553  200140 010536'          16330              MOVE    D,[POINT 7,OPMSG]
004554  134000 000003           16340              ILDB    A,D
004555  136000 000002           16350              IDPB    A,C;             MOVE OPERATOR MESSAGE TO HEADING LINE
004556  367040 004554'          16360              SOJG    B,.-2
                                16370     PG1:     XMT 6,CRS;  SIX CRS
                                      OPDEF X[6B12]
004557  000300 004133'          X CRS
004560  010000 000000           16380              INS     0
004561  263740 000000           16390              DONE
                                16400
004562  000000 000000           16410     L.OPM:   DEC     0;               # CHARACTERS IN OPERATOR MESSAGE
004563  201004 020100           16420     OPMSG:   ASCII   /                                /; ONLY 30 CHAR
004564  201004 020100
004565  201004 020100
004566  201004 020100
004567  201004 020100
004570  201004 020100
                                16430     ;        SHUTDOWN MESSAGE: "PREPARE FOR JOSS SHUTDOWN.".
                                16440
004571  503454 570302           16450     HSDM:    BYTE    (7)  120,162,145,160,141,162,145,40,146,157
004572  713124 063336
004573  710164 045236           16460              BYTE    (7)  162,7,40,112,117,123,123,40,163,150
004574  516464 071720
004575  727514 467756           16470              BYTE    (7)  165,164,144,157,167,156,56,7
004576  671340 700000
                                16480
                                16490     ,        DISC CLOBBERED MESSAGE: "FILES ARE NOT USEABLE."
                                16500
004577  036155 166312           16510     DCLOBM:  BYTE    (7)  7,106,151,154,145,163,40,141,162,145
004600  715014 171312
004601  200175 667750           16520              BYTE    (7)  40,7,156,157,164,40,165,163,7,141
004602  203536 303702
004603  613314 527000           16530              BYTE    (7)  142,154,145,56
```

```
                        16550     ,       DISC ACTION ROUTINES
                        16560
004604  201000 000001   16570     DREQ:   MOVEI   A,1
004605  402000 000000   16580             SETZM   SPARE4;         RESET DISC AVAILABLE FLAG
004606  137000 000267'  16590             DPB     A,S.DU;         FLAG AS DISC USER
004607  336000 002110'  16600             SKIPN   DKBY;           SKIP IF DISC IS BUSY
004610  254000 004613'  16610             JRST    DREQ1
004611  020000 000014   16620             CHS     DQ.S;           DISC BUSY - PUT ON QUEUE
004612  254000 007264'  16630             JRST    SIGPR1
004613  350000 002110'  16640     DREQ1:  AOS     DKBY;           SET DISC BUSY SWITCH
004614  476000 004605'  16650             SETOM   SPARE4;         TELL JOE HE GOT THE DISC
004615  254000 004350'  16660             JRST    MONEXIT;        RETURN TO INTERPRETER
                        16670
                        16680
004616  700740 000100   16690     DCONT:  CONSO   PI,CHDC
004617  254000 004622'  16700             JRST    DCONT1;         JUMP IF DC IDLE
004620  020000 000011   16710             CHS     DCT.S;          WAIT FOR DC TO IDLE
004621  254000 007264'  16720             JRST    SIGPR1
004622  020000 000012   16730     DCONT1: CHS     DIP.S;          DISC IN PROGRESS QUEUE
004623  264000 003773'  16740             JSR     DISC.C;         CONTINUE DISC ACTION
004624  254000 007264'  16750             JRST    SIGPR1
                        16760
004625  403040 002110'  16770     DCOMP:  SETZB   B,DKBY;         INDICATE DISC IDLE
004626  137040 000267'  16780             DPB     B,S.DU;         RESET DISC FLAG
004627  250700 002112'  16790             EXCH    S,MISC
004630  336700 002131'  16800             SKIPN   S,DQ;           SKIP IF SOMEONE IN THE QUEUE
004631  254000 004635'  16810             JRST    DCP1;           NONE IN QUEUE
004632  350000 002110'  16820             AOS     DKBY;           BUSY THE DISC
004633  275700 000141'  16830             SUBI    S,S.Q;
004634  020000 000007   16840             CHS     COM.S;          ACTIVATE USER
004635  250700 002112'  16850     DCP1:   EXCH    S,MISC
004636  135000 000265'  16860             LDB     A,S.OFR;        GET OFF FLAG
004637  322000 004350'  16870             JUMPE   A,MONEXIT;      RETURN TO INTERPRETER
004640  260740 006274'  16880             TSX     OFFNQ;          DO OFF PROCESSING
004641  254000 007264'  16890             JRST    SIGPR1
                        16900
                        16910     ,       DISC RESTART ROUTINE
                        16920
004642  700700 000100   16930     DSTRT:  CONSZ   PI,CHDC
004643  263740 000000   16940             DONE;                   NO RESTART IF DC IS BUSY
004644  275700 000141'  16950             SUBI    S,S.Q
004645  020000 000012   16960             CHS     DIP.S
004646  264000 004623'  16970             JSR     DISC.C;         CONTINUE DISC ACTION
004647  263740 000000   16980             DONE
```

```
                                17000      ,        ACCOUNT FOR DISC USAGE AT DISCARD TIME
                                17010      ,        USES REGISTERS B,E,S; ENTERED VIA PUSHJ PP,ADIS.
                                17020
004650  201100 000001           17030      ADIS:    MOVEI    C,1
004651  204042 000000           17040               MOVS     B,DISC.D(C)
004652  316042 004651'          17050               CAMN     B,DISC.D(C);     SKIP IF DATES ARE DIFFERENT
004653  263740 000000           17060               DONE;                     NO ACCOUNTING
004654  260740 010343'          17070               TSX      GETBUF
004655  263740 000000           17080               DONE;                     IN THIS CASE THEY GO FREE
004656  200040 000000           17090               MOVE     B,FILE
004657  202044 000007           17100               MOVEM    B,↑D7(E)
004660  200040 000000           17110               MOVE     B,KEY
004661  202044 000010           17120               MOVEM    B,↑D8(E)
004662  200056 002572'          17130               MOVE     B,MINT(S)
004663  202044 000011           17140               MOVEM    B,↑D9(E)
004664  200040 010547'          17150               MOVE     B,[XWD ↑D9,↑D9]
004665  202044 000001           17160               MOVEM    B,1(E)
004666  201044 000002           17170               MOVEI    B,2(E);          TO ADDRESS
004667  505040 004652'          17180               HRLI     B,DISC.D;        ADDRESS OF ACCOUNTING DATA
004670  200700 000001           17190               MOVE     S,B
004671  251056 000004           17200               BLT      B,4(S);          MOVE TO BUFFER
004672  550040 000000           17210               HRRZ     B,PROG
004673  542044 000002           17220               HRRM     B,2(E)
004674  200004 000006           17230               MOVE     A,6(E)
004675  603000 777777           17240               TLNE     A,777777;        CHECK FOR BINARY
004676  260740 004704'          17250               TSX      TLT;             NO - CONVERT FROM ASCII
004677  603000 777777           17260               TLNE     A,777777
004700  202244 000006           17270               MOVEM    F,6(E)
004701  200700 010550'          17280               MOVE     S,[DEC -2];      CONTEXT FOR TAPE
004702  260740 010305'          17290               TSX      PUTB
004703  263740 000000           17300               DONE
                                17310
004704  402000 000005           17320      TLT:     SETZM    F
004705  200100 010535'          17330               MOVE     C,[POINT 7,A]
004706  134140 000002           17340      TLT1:    ILDB     D,C
004707  306140 000040           17350               CAIN     D," ";             WE ARE DONE ON THE FIRST BLANK
004710  263740 000000           17360               DONE
004711  405140 000017           17370               ANDI     D,17
004712  221240 000012           17380               IMULI    F,↑D10
004713  270240 000003           17390               ADD      F,D
004714  254000 004706'          17400               JRST     TLT1
```

```
                                17420     ;                THE NUMBER IN A IS CONVERTED AND STUFFED IN THE STRING
                                17430     ;                AT C, B IS CLOBBERED
                                17440
004715  402000 000001           17450     BTA:     SETZM   B
004716  231000 000012           17460              IDIVI   A,↑D10
004717  506057 000000           17470              HRLM    B,0(PP);         PUT REMAINDER AWAY ON PDL
004720  322000 004722'          17480              JUMPE   A,BTA1;          END, GO TO OUTPUT
004721  260740 004715'          17490              TSX     BTA;             RECURSE
                                17500
004722  554017 000000           17510     BTA1:    HLRZ    A,0(PP);         RECOVER DIGIT
004723  435000 000060           17520              ORI     A,60;            PUT IN THE ASCII BITS
004724  136000 000002           17530              IDPB    A,C;             DEPOSIT IN STRING
004725  263740 000000           17540              DONE
```

RETURN BLOCKS OF CORE

```
004726  135100 000264'  17560   RCOR:  LDB    C,S.BLOCK
004727  200140 010544'  17570          MOVE   D,[POINT 6,CT48A]
004730  134000 000003   17580          ILDB   A,D
004731  367100 004730'  17590          SOJG   C,.-1
004732  275000 000001   17600          SUBI   A,1;            DECREMENT SIZE DISTRIBUTION
004733  137000 000003   17610          DPB    A,D
004734  135100 000264'  17620          LDB    C,S.BLOCK
004735  274100 000004   17630          SUB    C,E
004736  305100 000001   17640          CAIGE  C,1
004737  040000 000023   17650   H23:   HALT   23;             ATTEMPT TO RETURN ALL USER CORE
004740  137100 000264'  17660          DPB    C,S.BLOCK;      SET NEW USER SIZE
004741  200140 010544'  17670          MOVE   D,[POINT 6,CT48A]
004742  134000 000003   17680          ILDB   A,D
004743  367100 004742'  17690          SOJG   C,.-1
004744  271000 000001   17700          ADDI   A,1
004745  137000 000003   17710          DPB    A,D;            INCREMENT SIZE DISTRIBUTION
004746  135100 000264'  17720          LDB    C,S.BLOCK
004747  135000 000263'  17730          LDB    A,S.COR
004750  275000 000020   17740          SUBI   A,BBLOCK
004751  270100 000000   17750          ADD    C,A
004752  270100 000004   17760          ADD    C,E;            LOCATION OF BLOCKS TO BE RELEASED
004753  275100 000001   17770          SUBI   C,1
004754  402002 000272'  17780          SETZM  CORE(C);        ZERO THE FREED BLOCKS
004755  367200 004753'  17790          SOJG   E,.-2
004756  260740 005110'  17800          TSX    SRR;            RESET PROTECT
004757  254000 004350'  17810          JRST   MONEXIT
```

PROVIDE MORE CORE FOR USERS BLOCK

```
004760  201040 000000   17830   MORCOR: MOVEI   B,0;              PRESET TO DENY REQUEST
004761  135100 000264'  17840           LDB     C,S.BLOCK;        CURRENT # OF BLOCKS
004762  311100 000312'  17850           CAML    C,N.CB;           COMPARE WITH MAX ALLOWABLE
004763  254000 004350'  17860           JRST    MONEXIT;          ENOUGH - DENY REQUEST
004764  200140 010544'  17870   MORC1:  MOVE    D,[POINT 6,CT48A]
004765  134040 000003   17880           ILDB    B,D
004766  367100 004765'  17890           SOJG    C,.-1;            COUNT DOWN TO PROPER BYTE
004767  275040 000001   17900           SUBI    B,1
004770  137040 000003   17910           DPB     B,D;              ONE LESS AT THIS SIZE
004771  134040 000003   17920           ILDB    B,D
004772  271040 000001   17930           ADDI    B,1
004773  137040 000003   17940           DPB     B,D;              ONE MORE AT NEXT SIZE
                         17950
                         17960   ,       IS BLOCK JUST ABOVE HIM FREE?
                         17970
004774  135100 000264'  17980           LDB     C,S.BLOCK;        CURRENT # OF BLOCKS
004775  135140 000263'  17990           LDB     D,S.COR;          CURRENT BLOCK LOCATION
004776  275140 000020   18000           SUBI    D,BBLOCK
004777  270140 000002   18010           ADD     D,C;              INDEX TO NEXT HIGHER BLOCK
005000  350000 000002   18020           AOS     C;                NUMBER OF REQUIRED BLOCKS
005001  311140 000313'  18030           CAML    D,N.C;            SKIP IF NOT TOP USER
005002  254000 005020'  18040           JRST    MOR6;             TO MOR1 FOR INCREASE IN CORE *******
005003  332043 000272'  18050           SKIPE   B,CORE(D) ;       SKIP IF BLOCK IDLE
005004  254000 005020'  18060           JRST    MOR6;             TO MOR1 TO INCREASE IN CORE************
005005  661040 400000   18070           TLO     B,400000;         SET IN USE
005006  202043 000272'  18080           MOVEM   B,CORE(D)
005007  137100 000264'  18090           DPB     C,S.BLOCK;        NOW HE HAS ONE MORE
                         18100
005010  201040 000001   18110   M1:     MOVEI   B,1;              TELL JOE HE GOT ONE
005011  260740 005110'  18120           TSX     SRR
005012  254000 004350'  18130           JRST    MONEXIT;          TELL INTERPRETER HE  GOT ONE.
                         18140
                         18150   ;       IMPERATIVE REQUEST FOR MORE CORE
                         18160
005013  135100 000264'  18170   MORCI:  LDB     C,S.BLOCK
005014  315100 000313'  18180           CAMGE   C,N.C;            SKIP IF TOO BIG FOR MACHINE
005015  254000 004764'  18190           JRST    MORC1
005016  201040 000000   18200           MOVEI   B,0
005017  254000 004350'  18210           JRST    MONEXIT;          DENY REQUEST
                         18220
                         18230   ,       NOW WE MUST SET UP A SWAP
                         18240
005020  137100 000264'  18250   MOR6:   DPB     C,S.BLOCK;        SET NEW SIZE
005021  336000 002107'  18260           SKIPN   DMBY;             SKIP IF DRUM IS BUSY
005022  254000 005025'  18270           JRST    MOR7
005023  020000 000021   18280           CHS     QDM.S;            PUT ON QUEUE FOR OUT
005024  254000 007264'  18290           JRST    SIGPR
                         18300
005025  275100 000001   18310   MOR7:   SUBI    C,1;              SIZE TO SHIP OUT
005026  476000 003331'  18320           SETOM   DMIN;             INDICATE NO USER IN
005027  260740 003234'  18330           TSX     OSWAP;            START HIM OUT
005030  020000 000006   18340           CHS     QC.S;             PUT ON QUEUE TO COME BACK
005031  254000 007264'  18350           JRST    SIGPR
                         18360
```

```
                          18380     ;        FIND A FREE BLOCK OF CORE
                          18390     ;           (OF REQUIRED SIZE ANYWHERE)
                          18400
                          18410     ,        C- REQUIRED # OF BLOCKS, E IS DESTROYED
                          18420     ,        B= INDEX OF TOP FOUND BLOCK, D= CORE CELL  (=0)  OF LAST BLOCK
                          18430     ;        ROUTINE SKIPS ON NO SUCCESS
                          18440
005032  201040 000000     18450     FINDB:   MOVEI    B,0
005033  201200 000000     18460     MOR2:    MOVEI    E,0
005034  332141 000272'    18470     MOR2.5:  SKIPE    D,CORE(B)
005035  254000 005046'    18480              JRST     MOR3.5
005036  350000 000004     18490              AOS      E
005037  317100 000004     18500              CAMG     C,E;            IS BLOCK OF REQUIRED SIZE?
005040  263740 000000     18510              DONE;                    YES, TAKE SUCCESS EXIT
005041  350000 000001     18520     MOR3:    AOS      B
005042  315040 000313'    18530              CAMGE    B,N.C;          SKIP IF SEARCHED ALL BLOCKS
005043  254000 005034'    18540              JRST     MOR2.5
005044  350017 000000     18550              AOS      0(PP);          BUMP TO NO SUCCESS EXIT
005045  263740 000000     18560              DONE
005046  201200 000000     18570     MOR3.5:  MOVEI    E,0
005047  254000 005041'    18580              JRST     MOR3
                          18590
                          18600     ZZ=.
```

```
                          18620     ;        FIND A FREE BLOCK OF THE REQUIRED SIZE
                          18630     ;          (C = REQUIRED SIZE)
                          18640
005050  260740 005032'    18650     MOR1:    TSX     FINDB
005051  254000 005100'    18660              JRST    M2;              FOUND GO TO MOVE
                          18670
                          18680     ,        FIND A FREE BLOCK BELOW USER
                          18690
                          18700     ,        EXITS WITH B= INDEX TO FREE BLOCK
                          18710
005052  135040 000263'    18720     MOR4:    LDB     B,S.COR
005053  275040 000020     18730              SUBI    B,BBLOCK
005054  322040 005064'    18740              JUMPE   B,MOR5;          JUMP IF NO BLOCKS BELOW
005055  275040 000001     18750              SUBI    B,1
005056  200141 000272'    18760     MOR41:   MOVE    D,CORE(B)
005057  603140 200000     18770              TLNE    D,200000;        SKIP IF DRUM NOT USING BLOCK
005060  254000 005064'    18780              JRST    MOR5;            DRUM USING THIS AREA -- CANT TAMPER
005061  336000 000003     18790              SKIPN   D
005062  254000 005116'    18800              JRST    M4;              FOUND ONE: GO TO MOVE
005063  365040 005056'    18810              SOJGE   B,MOR41;         GO FOR NEXT BLOCK
                          18820
```

```
                        18840     ,       FIND A FREE BLOCK ABOVE USER; C= # OF REQUIRED BLOCKS
                        18850
005064  135040 000263'  18860     MOR5:   LDB     B,S.COR
005065  275040 000020   18870             SUBI    B,BBLOCK
005066  270040 000002   18880             ADD     B,C;            START LOOKING ABOVE
005067  275040 000002   18890             SUBI    B,2
005070  311040 000313'  18900     MOR51:  CAML    B,N.C
005071  254000 005020'  18910             JRST    MOR6;           NONE , TRY SWAPS
005072  200141 000272'  18920             MOVE    D,CORE(B)
005073  603140 200000   18930             TLNE    D,200000
005074  254000 005020'  18940            .JRST    MOR6;           DRUM IS USING CORE
005075  336000 000003   18950             SKIPN   D
005076  254000 005135'  18960             JRST    M5;             FOUND ONE: GO TO MOVE
005077  344040 005070'  18970             AOJA    B,MOR51;        GO TO LOOK AT NEXT BLOCK
                        18980
```

```
                          19000     ;        MOVE USER TO FREE AREA
                          19010
                          19020     ,        C= # OF BLOCKS IN NEW USER AREA
                          19030     ,        B= CORE MAP INDEX OF LAST BLOCK IN "TO" REGION
                          19040
005100  661140 400000     19050     M2:      TLO     D,400000;        BUSY THE HIGH BLOCK
005101  202141 000272'    19060              MOVEM   D,CORE(B)
005102  137100 000264'    19070              DPB     C,S.BLOCK;       NEW NUMBER OF BLOCKS
005103  275100 000001     19080              SUBI    C,1;             BLOCKS TO MOVE
005104  274040 000002     19090              SUB     B,C;             "TO" BLOCK LOCATION
005105  135200 000263'    19100              LDB     E,S.COR
005106  275200 000020     19110              SUBI    E,BBLOCK;        INDEX OF "FROM" BLOCK
005107  200300 000004     19120              MOVE    G,E;             SAVE "FROM" LOCATION
005110  260740 005073'    19130              TSX     BLTC;            MOVE USER AND CORE MAP
005111  402006 000272'    19140     M25:     SETZM   CORE(G) ;        ZERO OLD MAP
005112  350000 000006     19150              AOS     G
005113  367100 005111'    19160              SOJG    C,M25
005114  260740 005050'    19170              TSX     RUL;             RESET USER LOCATIONS
005115  254000 005010'    19180              JRST    M1;              SUCCESS RETURN TO JOE
                          19190
                          19200     ;        MOVE DOWN MULTIPLE USERS (B=FREE BLOCK INDEX)
                          19210
                          19220     ,        C= NEW SIZE
                          19230
005116  135240 000263'    19240     M4:      LDB     F,S.COR
005117  275240 000020     19250              SUBI    F,BBLOCK
005120  137100 000264'    19260              DPB     C,S.BLOCK;       NEW SIZE
005121  270100 000005     19270              ADD     C,F
005122  200000 000002     19280              MOVE    A,C;             SAVE INDEX OF HIGH BLOCK
005123  275100 000002     19290              SUBI    C,2
005124  274100 000001     19300              SUB     C,B;             BLOCKS TO MOVE
005125  200200 000001     19310              MOVE    E,B;             "TO" LOCATION
005126  271200 000001     19320              ADDI    E,1;             "FROM" LOCATION
005127  260740 005073'    19330              TSX     BLTC;            MOVE USER
005130  260740 005050'    19340              TSX     RUL;             RESET LOCATION FOR AFFECTED USERS
005131  201040 000001     19350              MOVEI   B,1
005132  250000 000001     19360              EXCH    A,B
005133  137000 000314'    19370              DPB     A,S.IU;          BUSY THE FREED BLOCK
005134  254000 005010'    19380              JRST    M1;              BACK TO JOE
                          19390
```

PROVIDE MORE CORE FOR USERS BLOCK

```
                                19410      ,        MOVE USER(S) UP
                                19420
                                19430      ,        B= FREE BLOCK INDEX, C= NEW SIZE
                                19440
005135  137100  000264'         19450      M5:      DPB     C,S.BLOCK;       SET NEW SIZE
005136  135200  000263'         19460               LDB     E,S.COR
005137  275200  000020          19470               SUBI    E,BBLOCK
005140  270200  000002          19480               ADD     E,C
005141  275200  000001          19490               SUBI    E,1
005142  261740  000001          19500               PUSH    PP,B;            SAVE FREE BLOCK LOCATION
005143  200001  000271'         19510      M51:     MOVE    A,CORE-1(B)
005144  202001  000272'         19520               MOVEM   A,CORE(B);       MOVE THE CORE MAP
005145  370000  000001          19530               SOS     B
005146  313040  000004          19540               CAMLE   B,E
005147  254000  005143'         19550               JRST    M51
005150  201100  000001          19560               MOVEI   C,1
005151  137100  000314'         19570               DPB     C,S.IU;          BUSY THE NEW BLOCK
005152  262740  000001          19580               POP     PP,B
005153  271040  000020          19590               ADDI    B,BBLOCK
005154  240040  000012          19600               ASH     B,↑D10
005155  271200  000020          19610               ADDI    E,BBLOCK
005156  240200  000012          19620               ASH     E,↑D10;          LAST WORD TO BE MOVED
005157  370000  000001          19630      M52:     SOS     B
005160  200141  000000          19640               MOVE    D,0(B)
005161  202141  002000          19650               MOVEM   D,2000(B)
005162  313040  000004          19660               CAMLE   B,E
005163  254000  005157'         19670               JRST    M52
005164  260740  005050'         19680               TSX     RUL;             RESET USER LOCATIONS
005165  254000  005010'         19690               JRST    M1;              RETURN TO JOE
                                19700
                                19710      RELOC    ZZ
```

```
                          19730     ;        RESET USER LOCATIONS
                          19740
                          19750     ,        SCAN CORE MAP AND UPDATE USER LOCATIONS
                          19760
005050  250700 002112'    19770     RUL:     EXCH    S,MISC
005051  201040 000000     19780              MOVEI   B,0
005052  336201 000272'    19790     RUL1:    SKIPN   E,CORE(B);       SKIP IF BUSY BLOCK
005053  344040 005067'    19800              AOJA    B,RUL2;          BUMP TO NEXT
005054  603200 200000     19810              TLNE    E,200000
005055  344040 005067'    19820              AOJA    B,RUL2;          DONT MESS WITH DRUM CORE
005056  135700 000317'    19830              LDB     S,S.UR;          USER FOR THIS BLOCK
005057  200200 000001     19840              MOVE    E,B
005060  271200 000020     19850              ADDI    E,BBLOCK
005061  137200 000263'    19860              DPB     E,S.COR
005062  135200 000264'    19870              LDB     E,S.BLOCK
005063  135240 000261'    19880              LDB     F,S.STA
005064  306240 000021     19890              CAIN    F,QDM.S
005065  275200 000001     19900              SUBI    E,1
005066  270040 000004     19910              ADD     B,E;             BUMP PAST THIS USER
005067  315040 000313'    19920     RUL2:    CAMGE   B,N.C
005070  254000 005052'    19930              JRST    RUL1;            AROUND FOR NEXT
005071  250700 002112'    19940              EXCH    S,MISC
005072  263740 000000     19950              DONE
                          19960
```

```
                                19980     ;        BLOCK TRANSFER CORE
                                19990
                                20000     ,        MOVE BLOCKS DOWN - USERS AND CORE MAP
                                20010     ,                C= BLOCKS TO BE MOVED
                                20020     ,                B= "TO" BLOCK INDEX
                                20030     ,                E= "FROM" BLOCK INDEX
                                20040     ,        CLOBBERS F,E,B.
                                20050
005073  504040 000004           20060     BLTC:    HRL     B,E;              BLT POINTER WORD
005074  200200 000001           20070              MOVE    E,B
005075  270200 000002           20080              ADD     E,C
005076  275200 000001           20090              SUBI    E,1
005077  200240 000001           20100              MOVE    F,B;              F MAY BE DESTROYED BY INTERRUPT
005100  270240 010551'          20110              ADD     F,[XWD CORE,CORE]
005101  251244 000272'          20120              BLT     F,CORE(E);        MOVE CORE MAP
005102  270040 010552'          20130              ADD     B,[XWD BBLOCK,BBLOCK]
005103  240040 000012           20140              ASH     B,↑D10
005104  271200 000020           20150              ADDI    E,BBLOCK
005105  240200 000012           20160              ASH     E,↑D10
005106  251044 001777           20170              BLT     B,2000-1(E)
005107  263740 000000           20180              DONE
                                20190
                                20200     ,        SET RELOCATION REGISTER
                                20210
005110  135140 000263'          20220     SRR:     LDB     D,S.COR
005111  240140 000012           20230              ASH     D,↑D10
005112  270140 010553'          20240              ADD     D,[OCT 100000700000];
005113  135400 000264'          20250              LDB     I,S.BLOCK
005114  275400 000001           20260              SUBI    I,1
005115  240400 000012           20270              ASH     I,↑D10
005116  207000 000010           20280              MOVSS   I
005117  270140 000010           20290              ADD     D,I
005120  700140 000003           20300              DATAO   APR,D;            SET RELOCATION
005121  263740 000000           20310              DONE
```

```
                          20330
                          20340      ,        COMPACT CORE
                          20350
005122  403040 000007     20360      COMPACT: SETZB   B,H
005123  336001 000272'    20370               SKIPN   CORE (B)
005124  350000 000007     20380               AOS     H;              COUNT FREE BLOCKS IN H
005125  350000 000001     20390               AOS     B
005126  315040 000313'    20400               CAMGE   B,N.C
005127  254000 005123'    20410               JRST    .-4
005130  260740 005133'    20420               TSX     COMP1;          DO A SINGLE COMPACT
005131  263740 000000     20430               DONE
005132  254000 005130'    20440               JRST    .-2
                          20450
                          20460      ;        DO ONE COMPACT
                          20470
005133  260740 005152'    20480      COMP1:   TSX     FM;             FIND A MOVE
005134  263740 000000     20490               DONE;                   NONE
005135  200140 000002     20500               MOVE    D,C
005136  270140 000001     20510               ADD     D,B;            BEGINNING OF FREED AREA
005137  260740 005073'    20520               TSX     BLTC;           DO THE MOVE
005140  402003 000272'    20530      COMP2:   SETZM   CORE (D) ;      ZERO FREED CORE
005141  350000 000003     20540               AOS     D
005142  315140 000006     20550               CAMGE   D,G
005143  254000 005140'    20560               JRST    COMP2
005144  200200 000002     20570               MOVE    E,C
005145  272100 002230'    20580               ADDM    C,CT50
005146  272200 002445'    20590               ADDM    E,T7+5
005147  260740 005050'    20600               TSX     RUL;            RESET USER LOCATIONS
005150  350017 000000     20610               AOS     0 (PP)
005151  263740 000000     20620               DONE;                   TAKE "DID ONE" EXIT
                          20630
```

```
                                20650     ;        FIND A CORE MOVE
                                20660
                                20670     ,        EXITS +1 IF NONE, +2 IF A MOVE IS FOUND
                                20680     ;        ON ENTRY H= # OF FREE BLOCKS
                                20690     ;        ON EXIT:
                                20700     ;                B= 1ST FREE LOCATION ("TO" BLOCK)
                                20710     ;                E= 1ST NON-ZERO BLOCK ("FROM")
                                20720     ;                C= # OF BLOCKS TO MOVE
                                20730     ;                G= # OF FIRST UNAFFECTED BLOCK
                                20740     ;        F AND I ARE CLOBBERED
                                20750
                                20760
005152  201040 000000           20770     FM:      MOVEI    B,0
005153  200141 000272'          20780     FM1:     MOVE     D,CORE(B)
005154  322140 005161'          20790              JUMPE    D,FM2;          SEARCH FOR 1ST UNUSED BLOCK
005155  350000 000001           20800              AOS      B
005156  312040 000313'          20810              CAME     B,N.C;          END TEST
005157  254000 005153'          20820              JRST     FM1
005160  263740 000000           20830              DONE;                    EXIT: NO FREE BLOCKS IN CORE
                                20840
005161  200200 000001           20850     FM2:     MOVE     E,B;            B IS "TO" LOCATION (1ST FREE BLOCK)
005162  200400 000007           20860              MOVE     I,H
005163  332004 000272'          20870     FM3:     SKIPE    CORE(E);        SEARCH FOR 1ST NON-ZERO BLOCK
005164  254000 005173'          20880              JRST     FM4
005165  350000 000004           20890              AOS      E
005166  377000 000010           20900              SOSG     I
005167  263740 000000           20910              DONE;                    WE HAVE ALL FREE BLOCKS TOGETHER
005170  312200 000313'          20920              CAME     E,N.C;          END TEST
005171  254000 005163'          20930              JRST     FM3
005172  263740 000000           20940              DONE;                    CORE IS ALREADY COMPACT
                                20950
005173  200100 000004           20960     FM4:     MOVE     C,E;            E IS "FROM" LOCATION (1ST NON-ZERO)
005174  200242 000272'          20970     FM5:     MOVE     F,CORE(C)
005175  322240 005203'          20980              JUMPE    F,FM7
005176  603240 200000           20990              TLNE     F,200000;       SKIP IF DRUM NOT USING BLOCK
005177  254000 005203'          21000              JRST     FM7;            DO NOT MOVE DRUM CORE
005200  350000 000002           21010              AOS      C
005201  312100 000313'          21020              CAME     C,N.C;          END TEST
005202  254000 005174'          21030              JRST     FM5
                                21040
005203  200300 000002           21050     FM7:     MOVE     G,C;            SAVE # OF FIRST UNAFFECTED BLOCK
005204  274100 000004           21060              SUB      C,E;            C HAS # OF BLOCKS TO MOVE
005205  337000 000002           21070              SKIPG    C
005206  263740 000000           21080              DONE;                    NO NEED TO MOVE ZERO BLOCKS
005207  350017 000000           21090              AOS      0(PP);          TO SECOND EXIT--MOVE FOUND
005210  263740 000000           21100              DONE
                                21110
```

SELECT A USER FOR DRUM SWAP

```
                          21130     ,        FIND A USER TO BRING IN
                          21140
005211  336000 002116'    21150     SELSWP:  SKIPN   ON
005212  332000 000320'    21160              SKIPE   N.DRM
005213  254000 005215'    21170              JRST    .+2;            SELECT FOR SWAP IF "ON" USERS OR SOME ON DRUM
005214  263740 000000     21180              DONE
005215  200100 010554'    21190              MOVE    C,[POINT 5,SI1];  TABLE OF CANDIDATES
005216  134040 000002     21200     SS1:     ILDB    B,C
005217  306040 000037     21210              CAIN    B,END.S
005220  263740 000000     21220              DONE;                   NONE REQUIRED
005221  201000 000001     21230              MOVEI   A,1
005222  550041 002115'    21240              HRRZ    B,S.QUE(B);     QUEUE HEADER
005223  322040 005216'    21250              JUMPE   B,SS1;          JUMP IF LIST EMPTY
005224  550700 000001     21260     SS2:     HRRZ    S,B
005225  275700 000141'    21270              SUBI    S,S.Q;          STATION INDEX
005226  135140 000263'    21280              LDB     D,S.COR;        GET CORE LOCATION
005227  135240 000270'    21290              LDB     F,S.GK
005230  326240 005235'    21300              JUMPN   F,SS3;          DONT BRING IN A GRONKED USER
005231  135240 000261'    21310              LDB     F,S.STA;        AND STATE
005232  306240 000001     21320              CAIN    F,ON.S
005233  254000 005262'    21330              JRST    SS10;           DEAL WITH "ON" SEPARATELY
005234  322140 005240'    21340              JUMPE   D,SS4;          JUMP IF NOT IN CORE
005235  550041 000000     21350     SS3:     HRRZ    B,0(B);         GET NEXT ON LIST
005236  322040 005216'    21360              JUMPE   B,SS1;          JUMP TO GET NEXT LIST
005237  344000 005224'    21370              AOJA    A,SS2
                          21380
                          21390     ,        USER # IN S, STATE IN F, PLACE IN QUEUE IN A
                          21400
                          21410     ,        FIND A FREE BLOCK FOR HIM
                          21420
005240  552700 003331'    21430     SS4:     HRRZM   S,DMIN;         SET "IN" CANDIDATE
005241  506000 003331'    21440              HRLM    A,DMIN;         SET PLACE IN QUEUE
005242  135100 000264'    21450     SS5:     LDB     C,S.BLOCK;      GET SIZE
005243  260740 005032'    21460              TSX     FINDB;          FIND A FREE BLOCK OF THE RIGHT SIZE
005244  254000 005277'    21470              JRST    SS15;           FOUND, GO TO START IN
                          21480
                          21490     ,        TRY TO FIND SCATTERED FREE BLOCKS
                          21500
005245  403040 000004     21510              SETZB   B,E
005246  336001 000272'    21520     SS6:     SKIPN   CORE(B)
005247  350000 000004     21530              AOS     E;              COUNT FREE BLOCKS
005250  350000 000001     21540              AOS     B
005251  315040 000313'    21550              CAMGE   B,N.C
005252  254000 005246'    21560              JRST    SS6
005253  315200 000002     21570              CAMGE   E,C;            SKIP IF ENOUGH FREE TO COMPACT
005254  254000 005305'    21580              JRST    SS17;           CANT COMPACT
005255  260740 005122'    21590              TSX     COMPACT;        COMPACT CORE
005256  135100 000264'    21600              LDB     C,S.BLOCK
005257  260740 005032'    21610              TSX     FINDB
005260  254000 005277'    21620              JRST    SS15;           FOUND
005261  254000 005305'    21630              JRST    SS17;           STILL CANT FIND--TRY SOMETHING ELSE
```

```
                          21650     ,       DEAL WITH "ON" SEPARATELY
                          21660
005262  200040 000313'    21670     SS10:   MOVE    B,N.C
005263  275040 000001     21680             SUBI    B,1
005264  336001 000272'    21690             SKIPN   CORE(B)
005265  263740 000000     21700             DONE;                   WE HAVE A FREE BLOCK. SELINT WILL FIND
005266  365040 005264'    21710             SOJGE   B,.-2;          SEARCH CORE FOR A FREE BLOCK
005267  200040 010555'    21720             MOVE    B,[POINT 5,S12]; 89SEARCH LIST
005270  200340 010556'    21730             MOVE    H,[TSX SS18];   TEST ROUTINE
005271  260740 005472'    21740             TSX     FIND
005272  263740 000000     21750             DONE;                   CANT FIND ONE BEST GET COMPUTING
005273  562700 003331'    21760             HRROM   S,DMIN;         SET NO IN REQUIRED
005274  135100 000264'    21770     SS11:   LDB     C,S.BLOCK
005275  260740 003234'    21780             TSX     OSWAP
005276  263740 000000     21790             DONE
                          21800
                          21810     ,       READY FOR "IN" SWAP; B= TOP OF IN BLOCK
                          21820
005277  135100 000264'    21830     SS15:   LDB     C,S.BLOCK
005300  275100 000001     21840             SUBI    C,1
005301  274040 000002     21850             SUB     B,C
005302  202040 003324'    21860             MOVEM   B,DMIBK;        SET "IN" BLOCK LOCATION
005303  260740 003333'    21870             TSX     ISWAP
005304  263740 000000     21880             DONE
                          21890
                          21900     ,       FIND A LOW PRIORITY USER OF THE PROPER SIZE
                          21910
005305  200240 000002     21920     SS17:   MOVE    F,C
005306  200040 010557'    21930             MOVE    B,[POINT 5,S11]
005307  200340 010556'    21940             MOVE    H,[TSX SS18]
005310  260740 005472'    21950             TSX     FIND
005311  254000 005327'    21960             JRST    SS30;           CANT FIND A SINGLE - TRY MULTIPLE
005312  254000 005274'    21970             JRST    SS11;           FOUND - GO TO START OUT
                          21980
005313  135300 000263'    21990     SS18:   LDB     G,S.COR
005314  336000 000006     22000             SKIPN   G;              SKIP IF IN CORE
005315  263740 000000     22010             DONE
005316  315140 000005     22020             CAMGE   D,F;            SKIP IF BIG ENOUGH
005317  263740 000000     22030             DONE
005320  350017 000000     22040             AOS     0(PP);          BUMP TO SUCCESS EXIT
005321  263740 000000     22050             DONE
                          22060
```

SELECT MULTIPLE USERS FOR TRANSFER TO DRUM

```
                                22080     ;        FIND CANDIDATE(S)  FOR DRUM
                                22090
005322  407563 261324           22100     S10:     BYTE     (5)  QP.S,ABG.S,DSU.S,GR.S,DQ.S,CK.S,DIP.S,DCT.S,END.S
005323  237400 000000
005324  407563 374000           22110     S11:     BYTE     (5)  QP.S,ABG.S,DSU.S,GR.S,END.S
005325  407563 261324           22120     S12:BYTE  (5)  QP.S,ABG.S,DSU.S,GR.S,DQ.S,CK.S,DIP.S,DCT.S,COM.S,END.S
005326  223770 000000
                                22130
                                22140     ,        FIND MULTIPLE USERS TO KICK OUT
                                22150     ,        E= # OF FREE BLOCKS, F= REQUIRED SIZE TO FREE
                                22160
005327  200040 010560'          22170     SS30:    MOVE     B,[POINT 5,S10]
005330  200340 010561'          22180              MOVE     H,[TSX SS35];    TEST ROUTINE
005331  200300 003305'          22190              MOVE     G,DMNR
005332  274240 000004           22200              SUB      F,E;             REQUIRED BLOCKS TO FREE
005333  260740 005472'          22210              TSX      FIND
005334  254000 005376'          22220              JRST     SS40;            NO SUCCESS - TRY COMPUTE QUEUE
                                22230
                                22240     ,        START OUT, POSSIBLY MULTIPLE, USERS
                                22250
005335  262300 000016           22260     SS32:    POP      G,S;             GET FIRST USER TO WRITE
005336  135100 000264'          22270              LDB      C,S.BLOCK
005337  260740 003234'          22280              TSX      OSWAP;           START HIM TO DRUM
005340  202300 003305'          22290              MOVEM    G,DMNR;          SAVE LIST OF USERS TO WRITE
005341  603300 777777           22300              TLNE     G,777777;        SKIP IF ONLY ONE TO WRITE
005342  254000 005351'          22310              JRST     SS33
005343  135040 000264'          22320              LDB      B,S.BLOCK;       BLOCKS WRITTEN
005344  550700 003331'          22330              HRRZ     S,DMIN;          USER TO COME IN
005345  135100 000264'          22340              LDB      C,S.BLOCK;       BLOCKS TO COME IN
005346  315040 000002           22350              CAMGE    B,C;             SKIP IF WE ARE FREEING ENOUGH SPACE
005347  350000 005470'          22360              AOS      SS98;            FIRE COMPACT ACTION
005350  263740 000000           22370              DONE;                     DONE WITH SWAP SELECTION
                                22380
005351  201100 000001           22390     SS33:    MOVEI    C,1
005352  350000 005470'          22400              AOS      SS98;            SIGNAL CORE COMPACT NEEDED
005353  262300 000016           22410     SS34:    POP      G,S
005354  135140 000264'          22420              LDB      D,S.BLOCK
005355  135040 000263'          22430              LDB      B,S.COR
005356  275040 000020           22440              SUBI     B,BBLOCK
005357  137100 000315'          22450              DPB      C,S.ID;          BUSY ALL BLOCKS INVOLVED IN SWAP
005360  350000 000001           22460              AOS      B
005361  367140 005357'          22470              SOJG     D,.-2
005362  603300 777777           22480              TLNE     G,777777
005363  254000 005353'          22490              JRST     SS34
005364  263740 000000           22500              DONE
                                22510
005365  135000 000263'          22520     SS35:    LDB      A,S.COR
005366  336000 000000           22530              SKIPN    A
005367  263740 000000           22540              DONE;                     NOT IN CORE TRY NEXT
005370  274240 000003           22550              SUB      F,D
005371  261300 000016           22560              PUSH     G,S;             SAVE AS USER TO WRITE
005372  333000 000005           22570              SKIPLE   F;               SKIP IF WE HAVE SELECTED ENOUGH
005373  263740 000000           22580              DONE
005374  350017 000000           22590              AOS      0(PP)
005375  263740 000000           22600              DONE;                     TAKE SUCCESS EXIT
                                22610
```

```
                                22630     ,        NOW WE MUST KICK OUT A COMPUTE USER
                                22640     ,        F= REQUIRED BLOCKS TO FREE, DMIN: R=USER FOR IN,L= PLACE IN QUEUE
                                22650     ,        G= CURRENT LIST TO WRITE
                                22660     ,        E= # OF FREE BLOCKS
                                22670
005376  200700 003331'          22680     SS40:    MOVE     S,DMIN
005377  135040 000261'          22690              LDB      B,S.STA;          GET STATE OF GUY COMING IN
005400  201140 000001           22700              MOVEI    D,1
005401  201000 000001           22710              MOVEI    A,1;
005402  306040 000007           22720              CAIN     B,COM.S;          SKIP IF "IN" USER NOT ON COM QUEUE
005403  554000 000016           22730              HLRZ     A,S;              GET "INS" PLACE IN QUEUE
005404  200100 002124'          22740              MOVE     C,COM;            GET COMPUTE QUEUE HEADER
005405  322100 005442'          22750              JUMPE    C,SS45;           NONE IN COMPUTE - BEST GET TO COMPUTING
005406  201040 000037           22760              MOVEI    B,END.S
005407  261740 000001           22770              PUSH     PP,B;             PUT A FLAG ON THE LIST
005410  550700 000002           22780     SS41:    HRRZ     S,C
005411  275700 000141'          22790              SUBI     S,S.Q
005412  261740 000016           22800              PUSH     PP,S;             UNRAVEL COM LIST TO SEARCH BACKWARD
005413  350000 000003           22810              AOS      D
005414  506157 000000           22820              HRLM     D,0(PP);          RECORD PLACE IN QUEUE
005415  550102 000000           22830              HRRZ     C,0(C)
005416  326100 005410'          22840              JUMPN    C,SS41;           JUMP IF NOT LAST ON QUEUE
005417  262740 000016           22850     SS42:    POP      PP,S;             GET USER FROM INVERTED QUEUE
005420  306700 000037           22860              CAIN     S,END.S
005421  254000 005442'          22870              JRST     SS45;             END OF QUEUE
005422  554140 000016           22880              HLRZ     D,S;              PLACE IN QUEUE
005423  317140 000000           22890              CAMG     D,A
005424  254000 005437'          22900              JRST     SS44;             CANT FIND USER OF LOW ENOUGH PRIORITY
005425  135040 000263'          22910              LDB      B,S.COR
005426  322040 005417'          22920              JUMPE    B,SS42;           JUMP IF OUT OF CORE
005427  274240 000001           22930              SUB      F,B;              FOUND ONE, DECREMENT REQUIRED SIZE
005430  261300 000016           22940              PUSH     G,S;              PUT ON WRITE LIST
005431  327240 005417'          22950              JUMPG    F,SS42;           GO FOR MORE IF NEEDED
005432  262740 000001           22960              POP      PP,B
005433  302040 000037           22970              CAIE     B,END.S
005434  254000 005432'          22980              JRST     .-2
005435  350000 002446'          22990              AOS      T7+6;             COUNT COMPUTE USERS KICKED OUT
005436  254000 005335'          23000              JRST     SS32;             SUCCESS--GO FOR FIRST WRITE
                                23010
005437  262740 000001           23020     SS44:    POP      PP,B
005440  302040 000037           23030              CAIE     B,END.S
005441  254000 005437'          23040              JRST     .-2
                                23050
005442  200040 010562'          23060     SS45:    MOVE     B,[XWD 0,DMNR]
005443  202040 003305'          23070              MOVEM    B,DMNR;           RESET WRITE QUEUE HEADER
005444  263740 000000           23080              DONE;                      CANNOT SET UP WRITE--NEED TO COMPUTE
                                23090
005445  201700 000047           23100     TEST:    MOVEI    S,N.S-1
005446  135040 000263'          23110     TE1:     LDB      B,S.COR
005447  322040 005454'          23120              JUMPE    B,TE2;            JUMP IF NOT IN CORE
005450  240040 000012           23130              ASH      B,↑D10
005451  200041 000002           23140              MOVE     B,2(B);           INITIALS FROM USERS BLOCK
005452  312056 002572'          23150              CAME     B,MINT(S)
005453  040000 000022           23160     H22:     HALT     22;               NOT THE SAME AS CORE RECORD
005454  365700 005446'          23170     TE2:     SOJGE    S,TE1
005455  263740 000000           23180              DONE
```

```
                                23200      ,        COMPACT CORE FOR SWAP IN
                                23210
005456  260740 005445'          23220      SS90:    TSX      TEST
005457  260740 005122'          23230               TSX      COMPACT;          COMPACT CORE
005460  260740 005445'          23240               TSX      TEST
005461  200700 003331'          23250               MOVE     S,DMIN
005462  402000 005471'          23260               SETZM    SS99;             RESET COMPACT SWITCH
005463  135100 000264'          23270               LDB      C,S.BLOCK
005464  260740 005032'          23280               TSX      FINDB
005465  254000 005277'          23290               JRST     SS15
005466  402000 002107'          23300               SETZM    DMBY
005467  263740 000000           23310               DONE;                       NOT ROOM ENOUGH - MORECORE MUST HAVE STOLEN ONE
                                23320
005470  000000 000000           23330      SS98:    Z;       SET IF MULTI WRITE
005471  000000 000000           23340      SS99:    Z;       SET TO REQUEST COMPACTION OF CORE AND IN SWAP
```

```
                         23360     ,       FIND FIRST USER IN STATES LISTED AT POINTER IN B
                         23370     ;       WHICH SATISFIES TEST IN H (THAT SKIPS ON SUCCESS).
                         23380
                         23390     ,       C IS DESTROYED, S IS SET TO FOUND USER
                         23400
005472  134100 000001    23410     FIND:   ILDB    C,B;            INDEX TO STATE HEADER
005473  302100 000037    23420             CAIE    C,END.S;        SKIP IF END
005474  254000 005476'   23430             JRST    F1
005475  263740 000000    23440             DONE;                   FAILURE EXIT
                         23450
005476  550102 002115'   23460     F1:     HRRZ    C,S.QUE(C);     LIST HEADER
005477  322100 005472'   23470             JUMPE   C,FIND;         JUMP IF LIST EMPTY
005500  550700 000002    23480     F2:     HRRZ    S,C
005501  275700 000141'   23490             SUBI    S,S.Q;          COMPUTE STATION INDEX
005502  135140 000264'   23500             LDB     D,S.BLOCK
005503  256000 000007    23510             XCT     H;              DO TEST
005504  254000 005507'   23520             JRST    F3
005505  350017 000000    23530             AOS     0(PP)
005506  263740 000000    23540             DONE;                   TEST SUCCEEDS
                         23550
005507  550102 000000    23560     F3:     HRRZ    C,0(C);         TEST ROUTINE FAILS; GET NEXT ON LIST
005510  326100 005500'   23570             JUMPN   C,F2;           JUMP IF AN ENTRY
005511  254000 005472'   23580             JRST    FIND;           GO FOR NEXT LIST
                         23590
```

```
                                23610     ;        FORCE USERS OFF
                                23620
005512  402000 000324'          23630     BOFF:    SETZM    N.SON;            ALLOW NO MORE ON
005513  201000 200010           23640              MOVEI    A,SDS+DOAF;
005514  436000 004535'          23650              ORM      A,SWITCH;         SET SHUTDOWN AND OFF SIGNAL DISPLAY
005515  505000 000003           23660              HRLI     A,3;              GET AN OFF SIGNAL
005516  700600 001030           23670              CONO     PI,1000+CH630;    TURN OFF 630 INTERRUPTS
005517  200100 000115'          23680              MOVE     C,SG.L;           SIGNAL TABLE SIZE
005520  201700 000047           23690              MOVEI    S,N.S-1;          STATION # FROM TOP DOWN
005521  135040 000261'          23700     BOF1:    LDB      B,S.STA;          GET STATE
005522  135240 000267'          23710              LDB      F,S.DU;           GET DISC USE FLAG
005523  326240 005536'          23720              JUMPN    F,BOF5;           DONT KICK DISC USER
005524  200240 010563'          23730              MOVE     F,[POINT 5,BOF10]
005525  134200 000005           23740     BOF2:    ILDB     E,F;              GET EXEMPT STATE
005526  306200 000037           23750              CAIN     E,END.S;          SEARCHED ALL EXEMPTS?
005527  254000 005533'          23760              JRST     BOF3;             YES, NOT EXEMPT
005530  316200 000001           23770              CAMN     E,B;
005531  254000 005536'          23780              JRST     BOF5;             EXEMPT STATE- NO OFF SIGNAL
005532  254000 005525'          23790              JRST     BOF2;             GO FOR NEXT
005533  540000 000016           23800     BOF3:    HRR      A,S;              MAKE UP SIGNAL
005534  202002 000045'          23810              MOVEM    A,SIGTBL(C) ;     PUT ON SIGNAL LIST
005535  350000 000002           23820              AOS      C;                COUNT LIST
005536  365700 005521'          23830     BOF5:    SOJGE    S,BOF1;           GO FOR NEXT
005537  202100 000115'          23840              MOVEM    C,SG.L;           NUMBER OF ENTRIES
005540  700600 002030           23850              CONO     PI,2000+CH630;    TURN ON 630 INTERRUPTS
005541  263740 000000           23860              DONE
                                23870
                                23880     ,        LIST OF STATES WHICH SHOULD NOT BE SUMMARILY KICKED OFF
                                23890
005542  460370 000000           23900     BOF10:   BYTE     (5) OF.S,TOF.S,END.S
                                23910
```

```
                        23930     ;        PROCESS USERS WAITING FOR BUFFERS
                        23940
005543  275700 000141'  23950     MSGPR:   SUBI     S,S.Q;            COMPUTE USER NUMBER
005544  260740 010343'  23960              TSX      GETBUF
005545  263740 000000   23970              DONE;                      SHOULDNT HAPPEN THE FIRST TIME
005546  311700 004521'  23980              CAML     S,CONSOL;         SKIP IF TTY CONSOLE
005547  254000 005555'  23990              JRST     MSGPR1
005550  515040 574000   24000              HRLZI    B,574000;         LEFT ARROW CODE
005551  202044 000003   24010              MOVEM    B,3(E);           STASH IN BUFFER
005552  260740 010305'  24020              TSX      PUTB
005553  020000 000016   24030              CHS      DSU.S
005554  263740 000000   24040              DONE
005555  504200 000004   24050     MSGPR1:  HRL      E,E
005556  202216 000331'  24060              MOVEM    E,S.BUF(S);       LINK TO USER
005557  260740 006560'  24070              TSX      SG;               SWITCH TO GREEN
005560  263740 000000   24080              DONE
                        24090
                        24100     ,        PROCESS THE PAUSE QUEUE
                        24110
005561  544116 000000   24120     PQP:     HLR      C,0(S);           GET TIME COUNT
005562  322100 005571'  24130              JUMPE    C,PQP2
005563  540716 000000   24140              HRR      S,0(S);           NEXT ON LIST
005564  260740 006421'  24150              TSX      ISEC
005565  302040 000002   24160              CAIE     B,C
005566  254000 005571'  24170              JRST     PQP2;             TIME NOT UP YET
005567  020000 000007   24180              CHS      COM.S;            ITS TIME - ACTIVATE HIM
005570  553016 000141'  24190              HRRZS    S.Q(S);           ZERO REQUEST TIME
005571  326700 005561'  24200     PQP2:    JUMPN    S,PQP;            JUMP IF MORE IN PAUSE QUEUE
005572  263740 000000   24210              DONE
                        24220
```

```
                          24240     ;        PROCESS DISC INTERRUPT SIGNAL
                          24250
005573  302140 000003     24260     DISCP:   CAIE    D,3;            SKIP IF DISC RE-ENTRY
005574  254000 005600'    24270              JRST    DISC2
005575  402000 000000     24280              SETZM   DISC.S
005576  264000 004646'    24290              JSR     DISC.C
005577  263740 000000     24300              DONE
                          24310
005600  332000 006101'    24320     DISC2:   SKIPE   DSS;            SKIP IF NOT SKULKING
005601  254000 006032'    24330              JRST    SKULK
005602  336700 002127'    24340              SKIPN   S,DIP;           SKIP IF WE DO HAVE A USER
005603  040000 000010     24350     H10:     HALT    10;              DISC INTERRUPT BUT NO DISC USER
005604  275700 000141'    24360              SUBI    S,S.Q;           COMPUTE USER NUMBER
005605  020000 000007     24370              CHS     COM.S;           ACTIVATE USER FOR INFORMATION TRANSFER
005606  306140 000002     24380              CAIN    D,2;             SKIP IF NOT A DISCARD
005607  260740 004650'    24390              TSX     ADIS;            ACCOUNT FOR DISC USE
005610  402000 005575'    24400     DISC1:   SETZM   DISC.S;          RESET THE SIGNAL
005611  260740 003532'    24410              TSX     TRST;            RESTART THE TAPE
005612  263740 000000     24420              DONE
                          24430
                          24440     ;        SEARCH FOR GRONKED USERS
                          24450
005613  201700 000047     24460     GRONK:   MOVEI   S,N.S-1
005614  135040 000270'    24470     GR1:     LDB     B,S.GK
005615  326040 005621'    24480              JUMPN   B,GR3
005616  365700 005614'    24490     GR2:     SOJGE   S,GR1
005617  402000 003162'    24500              SETZM   CKER
005620  263740 000000     24510              DONE
                          24520
005621  202700 000021'    24530     GR3:     MOVEM   S,CUI
005622  201000 000071     24540              MOVEI   A,71
005623  254000 004162'    24550              JRST    KILL
```

INTER-CONSOLE SYNCHRONIZING SIGNALS

```
                                24570     ;        THESE ROUTINES ARE CALLED BY THE INTERPRETER TO SET (SSIG),
                                24580     ;        RESET (RSIG), AND TEST (TSIG) THE INTER-CONSOLE SIGNAL BITS.
                                24590     ;        ENTRY IS VIA PUSHJ 0,ROUTINE.
                                24600     ;        REGISTERS LISTED BELOW ARE PRESUMED FREE AND NOT SAVED.
                                24610     U=1
                                24620     V=2
                                24630     W=3
                                24640     X=4
                                24650     Y=5
                                24660     Z=6
                                24670
                                24680     ;        GET SIGNAL TABLE INDEX
                                24690
                                24700     ;        INPUT:  "TO" USER IN CUI
                                24710     ;                "FROM" USER IN U
                                24720
                                24730     ;        RETURNS: "TO" INDEX IN Z
                                24740     ;                 "FROM" BIT IN V FOR SETTING AND TESTING
                                24750
                                24760     ;        DESTROYS: W
                                24770
005624  200100 000001           24780     GTI:     MOVE    V,U
005625  240100 777773           24790              ASH     V,-5;          HIGH ORDER PART OF "FROM"
005626  200300 000021'          24800              MOVE    Z,CUI
005627  221300 000002           24810              IMULI   Z,S.M
005630  270300 000002           24820              ADD     Z,V;           COMPUTE "TO" INDEX
005631  200140 000001           24830              MOVE    W,U
005632  405140 000037           24840              ANDI    W,37;          MASK LOW ORDER  OF "FROM"
005633  201100 000001           24850              MOVEI   V,1
005634  240103 000000           24860              ASH     V,0(W);        SHIFT TO PROPER POSITION (HIGH ON LEFT)
005635  263000 000000           24870              POPJ    0,0
                                24880
                                24890     ;        RESET A SIGNAL BIT
                                24900
                                24910     ;        THE BIT "TO" CURRENT USER "FROM" (U) IS RESET
                                24920
005636  260000 005624'          24930     RSIG:    PUSHJ   GTI;           GET TABLE INDEX
005637  200146 005667'          24940              MOVE    W,TTT(Z)
005640  630140 000002           24950              TDZ     W,V
005641  202146 005667'          24960              MOVEM   W,TTT(Z)
005642  263000 000000           24970              POPJ    0,0
                                24980
                                24990     ;        TEST A SIGNAL BIT
                                25000
                                25010     ;        THE BIT "TO" CURRENT USER "FROM" (U) IS TESTED.
                                25020     ;        REGISTER V IS NON-ZERO IF BIT IS SET.
                                25030
005643  260000 005624'          25040     TSIG:    PUSHJ   GTI
005644  200146 005667'          25050              MOVE    W,TTT(Z)
005645  616140 000002           25060              TDNN    W,V
005646  402000 000002           25070              SETZM   V
005647  263000 000000           25080              POPJ    0,0
```

```
                          25100     ;        SET A SIGNAL BIT
                          25110
                          25120     ;        THE BIT "TO" (U) "FROM" CURRENT USER IS SET
                          25130
005650  250040 000021'    25140     SSIG:    EXCH    U,CUI
005651  260000 005624'    25150              PUSHJ   GTI
005652  200146 005667'    25160              MOVE    W,TTT(Z)
005653  670140 000002     25170              TDO     W,V
005654  202146 005667'    25180              MOVEM   W,TTT(Z)
005655  250040 000021'    25190              EXCH    U,CUI
                          25200
                          25210     ;        RE-ENABLE SIGNALLED USER IF WAITING
                          25220
005656  250740 000020'    25230              EXCH    PP,PPSAV;       GET MONITOR PUSH REGISTER
005657  200300 000016     25240              MOVE    Z,S;            SAVE REGISTER S
005660  200700 000001     25250              MOVE    S,U;            "TO" USER TO S
005661  135140 000271'    25260              LDB     W,S.SIG
005662  332000 000003     25270              SKIPE   W;              SKIP IF NOT WAITING
005663  020000 000007     25280              CHS     COM.S;          ACTIVATE USER
005664  200700 000006     25290              MOVE    S,Z;            RESTORE S
005665  250740 000020'    25300              EXCH    PP,PPSAV;       RESTORE PP
005666  263000 000000     25310              POPJ    0,0
                          25320
                          25330
                          25340
                          25350     ;        TABLE OF INTER-CONSOLE SIGNALS
                          25360
                          25370     ;        ONE WORD PER 32 STATIONS PER STATION
                          25380     ;        BITS ACROSS THE WORD CORRESPOND TO THE "FROM" CONSOLE
                          25390     ;        AND WORDS OR GROUPS DOWN THE TABLE TO THE "TO" CONSOLE.
                          25400
                          25410     TTT:     ZBLOK S.M*N.S
                            REPEAT S.M*N.S,
005667  000000 000000     Z
                          25420
```

```
                                25440      ,        MIDNIGHT DISC SKULKER
                                25450
006007  332000 006101'          25460      MDS:     SKIPE    DSS
006010  254000 006013'          25470               JRST     RESK
006011  332000 002110'          25480               SKIPE    DKBY
006012  254000 006030'          25490               JRST     MDS2
                                25500
                                25510      ,        START OR RESTART SKULK
                                25520
006013  700700 000100           25530      RESK:    CONSZ    PI,CHDC;        SKIP IF DATA CONTROL IDLE
006014  254000 006030'          25540               JRST     MDS2
006015  402000 006102'          25550               SETZM    SKT
006016  350000 006103'          25560               AOS      SKR;            FLAG DISC USING DC
006017  402000 000001           25570               SETZM    B
006020  336000 006101'          25580               SKIPN    DSS
006021  505040 000001           25590               HRLI     B,1;            INITIALIZATION BIT FOR IDG
006022  350000 006101'          25600               AOS      DSS;            FLAG SKULKING
006023  350000 002110'          25610               AOS      DKBY;           BUSY THE DISC
006024  541040 000144           25620               HRRI     B,↑D100;        REQUEST SKULK FROM DISC PROCESSOR
006025  202040 000000           25630               MOVEM    B,ACTION
006026  264000 005576'          25640               JSR      DISC.C
006027  263740 000000           25650               DONE
                                25660
006030  350000 006102'          25670      MDS2:    AOS      SKT;            FLAG START REQUEST
006031  263740 000000           25680               DONE
                                25690
```

```
                        25710     ;        DISC ACTION COMPLETE
                        25720
006032  402000 006103'  25730     SKULK:   SETZM   SKR;            DISC NOT USING DC
006033  200040 000000   25740              MOVE    B,RESULT
006034  306040 000014   25750              CAIN    B,↑D12
006035  254000 006072'  25760              JRST    SK5;             DISC ERROR
006036  302040 000144   25770              CAIE    B,↑D100;         SKIP IF RECORD READY
006037  254000 006055'  25780              JRST    SK1
006040  201040 000011   25790              MOVEI   B,↑D9
006041  200001 000000   25800     SK3:     MOVE    A,DBUF(B)
006042  603000 777777   25810              TLNE    A,777777
006043  260740 004704'  25820              TSX     TLT;             CONVERT ASCII TO BINARY
006044  603000 777777   25830              TLNE    A,777777
006045  202241 006041'  25840              MOVEM   F,DBUF(B)
006046  271040 000005   25850              ADDI    B,5
006047  305040 000206   25860              CAIGE   B,↑D134
006050  254000 006041'  25870              JRST    SK3
006051  200700 010550'  25880              MOVE    S,[DEC -2];      CONTEXT FOR TAPE
006052  201200 006045'  25890              MOVEI   E,DBUF
006053  260740 010305'  25900              TSX     PUTB;            START TO TAPE
006054  254000 005610'  25910              JRST    DISC1;           GO TO RESTART THE TAPE
                        25920
006055  302040 000310   25930     SK1:     CAIE    B,↑D200;         SKIP IF NO MORE RECORDS
006056  040000 000021   25940     H21:     HALT    21;              UNKNOWN ENTRY FROM DISC ROUTINES
006057  260740 010422'  25950              TSX     CMESS
                        25960              XMT 10,SK11; SAY SKULK COMPLETE
                            OPDEF X[10B12]
006060  000400 006077'  X SK11
006061  402000 005610'  25970     SK4:     SETZM   DISC.S
006062  402000 006101'  25980              SETZM   DSS
006063  336700 002131'  25990              SKIPN   S,DQ
006064  254000 006070'  26000              JRST    SK2
006065  275700 000141'  26010              SUBI    S,S.Q
006066  020000 000007   26020              CHS     COM.S;           ENABLE WAITING DISC USER
006067  263740 000000   26030              DONE
                        26040
006070  402000 002110'  26050     SK2:     SETZM   DKBY
006071  263740 000000   26060              DONE
                        26070
006072  260740 010422'  26080     SK5:     TSX     CMESS
                        26090              XMT 10,SK10; SAY NO SKULK
                            OPDEF X[10B12]
006073  000400 006075'  X SK10
006074  254000 006061'  26100              JRST    SK4
                        26110
006075  472364 051626   26120     SK10:    ASCII   ?NO SKULK?
006076  526311 300000
006077  516272 546226   26130     SK11:    ASCII   ?SKULK OK?
006100  202371 300000
                        26140
006101  000000 000000   26150     DSS:     Z;                       SET IF WE ARE SKULKING
006102  000000 000000   26160     SKT:     Z;                       SET WHEN TAPE OUTPUT COMPLETE
006103  000000 000000   26170     SKR:     Z;                       SET IF DISC USING DC FOR SKULK
```

SHUTDOWN PROCEDURE

```
                                26190      ,        SHUTDOWN JOSS SYSTEM
                                26200
006104  403140 000324'          26210      SDP:     SETZB   D,N.SON;          ALLOW NO MORE ON
006105  201100 000001           26220               MOVEI   C,1
006106  436100 005514'          26230               ORM     C,SWITCH;         BE SURE A MESSAGE IS GOING OUT
006107  201100 000022           26240               MOVEI   C,QM-S.QUE;       TEST ALL BUT QUEUE AND OF
006110  434142 002114'          26250               OR      D,S.QUE-1(C);
006111  367100 006110'          26260               SOJG    C,.-1
006112  332000 000003           26270               SKIPE   D;                SKIP IF ALL IS QUIET
006113  263740 000000           26280               DONE    ;                 STILL SOME ACTION
006114  571700 777777           26290               HRREI   S,-1;             TTY CONTEXT
006115  260740 006711'          26300               TSX     ODIS
006116  260740 003470'          26310               TSX     TWAIT
006117  722200 001400           26320               CONO    MTC,1400;         END FILE LOG
006120  260740 003470'          26330               TSX     TWAIT
006121  722200 007400           26340               CONO    MTC,7400;         BACK OVER IT
006122  260740 010422'          26350               TSX     CMESS
                                26360               XMT 23,TERM; SYSTEM ALL DOWN
                                    OPDEF X[23B12]
006123  001140 006130'          X TERM
006124  700600 001001           26370               CONO    PI,1000+CHAPR;    TURN OFF PROCESSOR INTERRUPTS
006125  402000 006106'          26380               SETZM   SWITCH;           BE SURE BEEP WONT INTERFERE
006126  264000 000000           26390               JSR     SHUT;             TURN OFF ALL CONSOLES
006127  254000 006127'          26400               JRST    .
                                26410
006130  516632 352212           26420      TERM:    ASCII   ?SYSTEM HAS SHUTDOWN?
006131  465011 040646
006132  202471 052650
006133  422372 747000
```

```
                        26440     ;        PROCESS DISTRIBUTOR SIGNALS
                        26450
006134  201300 000000   26460     PRSIG:   MOVEI   G,0;              ZERO ENTRY COUNT
006135  550706 000045'  26470     SP1:     HRRZ    S,SIGTBL(G) ;     GET ENTRY (XWD SIGTYPE,STN#)
006136  554046 000045'  26480              HLRZ    B,SIGTBL(G) ;     TYPE
006137  313700 000325'  26490              CAMLE   S,S.OK
006140  254000 006146'  26500              JRST    SP3;              WE ARE IGNORING THIS GUY
006141  303040 000005   26510              CAILE   B,!D5;            TEST FOR WITHIN LIMIT
006142  040000 000006   26520     H6:      HALT    6;                BAD SIGNAL FROM IRWIN
006143  135240 000261'  26530              LDB     F,S.STA;          GET STATE
006144  135101 006173'  26540              LDB     C,T2(B) ;         GET NAME OF PROPER ROUTINE
006145  256002 006224'  26550              XCT     T3(C) ;           DO IT
006146  700600 001030   26560     SP3:     CONO    PI,1000+CH630;    TURN OFF 630 INTERRUPTS
006147  350000 000006   26570              AOS     G;                BUMP ENTRY COUNT
006150  311300 000115'  26580              CAML    G,SG.L;           COMPARE WITH LENGTH
006151  254000 006154'  26590              JRST    SP2
006152  700600 002030   26600              CONO    PI,2000+CH630;    TURN ON 630 INTERRUPTS
006153  254000 006135'  26610              JRST    SP1;              GO BACK FOR NEXT ITEM
006154  402000 000115'  26620     SP2:     SETZM   SG.L;             ZERO THE SIGNAL LIST
006155  700600 002030   26630              CONO    PI,2000+CH630;    TURN ON 630 INTERRUPTS
006156  550700 000000   26640              HRRZ    S,FAKE
006157  554040 006156'  26650              HLRZ    B,FAKE
006160  626040 400000   26660              TRZN    B,400000;         SKIP IF PSEUDO SIGNAL
006161  263740 000000   26670              DONE
006162  303040 000004   26680              CAILE   B,4;              SKIP IF LEGITIMATE SIGNAL
006163  263740 000000   26690              DONE
006164  303700 000050   26700              CAILE   S,N.S;            SKIP IF LEGITIMATE STATION
006165  263740 000000   26710              DONE
006166  135240 000261'  26720              LDB     F,S.STA;          GET STATE
006167  135101 006173'  26730              LDB     C,T2(B)
006170  256002 006224'  26740              XCT     T3(C) ;           DO THE ACTION ROUTINE
006171  402000 006157'  26750              SETZM   FAKE;             FLAG DONE
006172  263740 000000   26760              DONE
                        26770
```

TABLES FOR SIGNAL INTERPRETATION

```
006173  350705 006200'  26790   T2:     POINT   7,T1(F),6;      TABLE OF POINTER TO COLUMNS IN T1
006174  260705 006200'  26800           POINT   7,T1(F),↑D13
006175  170705 006200'  26810           POINT   7,T1(F),↑D20
006176  100705 006200'  26820           POINT   7,T1(F),↑D27
006177  010705 006200'  26830           POINT   7,T1(F),↑D34
                        26840
                        26850           RADIX   10
                        26860   ;               ROUTINE NAME TABLE FOR GIVEN SIGNAL AND STATE
                        26870   ;               COLUMNS ARE: TO,IN,ON,OFF,CR
                        26880
                        26890   T1=.
006200  004020 100402   26900           BYTE    (7)   1,1,1,1,1; TOF
006201  004020 104400   26910           BYTE    (7)   1,1,1,9,0; ON
006202  000140 104400   26920           BYTE    (7)   0,6,1,9,0; RC
006203  064020 104400   26930           BYTE    (7)  13,1,1,9,0; RI
006204  064020 104400   26940           BYTE    (7)  13,1,1,9,0; RIB
006205  064120 104400   26950           BYTE    (7)  13,5,1,9,0; UC
006206  064140 107400   26960           BYTE    (7)  13,6,1,15,0;  QC
006207  064100 107400   26970           BYTE    (7)  13,4,1,15,0; COM
006210  000000 000000   26980           BYTE    (7)   0,0,0,0,0; CU
006211  064140 107000   26990           BYTE    (7)  13,6,1,14,0; DCT
006212  064140 107000   27000           BYTE    (7)  13,6,1,14,0; DIP
006213  060120 104400   27010           BYTE    (7)  12,5,1,9,0; CK
006214  064400 107000   27020           BYTE    (7)  13,16,1,14,0; DQ
006215  000020 104406   27030           BYTE    (7)   0,1,1,9,3; GR
006216  054020 104400   27040           BYTE    (7)  11,1,1,9,0; DSU
006217  064020 104400   27050           BYTE    (7)  13,1,1,9,0; ABG
006220  064100 107400   27060           BYTE    (7)  13,4,1,15,0; QP
006221  064140 107400   27070           BYTE    (7)  13,6,1,15,0;  QDM
006222  064160 104000   27080           BYTE    (7)  13,7,1,8,0; QM
006223  004021 200402   27090           BYTE    (7)   1,1,10,1,1;  OF
                        27100           RADIX   8
                        27110
                        27120   ;               TABLE OF ROUTINES FOR SIGNAL INTERPRETATION
                        27130
006224  040000 000020   27140   T3:     HALT    20;             0 - MACHINE ERROR HALT
006225  255000 000000   27150           NOP;                    1 - IGNORE SIGNAL
006226  255000 000000   27160           NOP;                    2 - UNUSED
006227  260740 006457'  27170           TSX     CR.R;           3 - GENERAL CR RESPONSE
006230  260740 006427'  27180           TSX     IN10;           4 - CHANGE TO RI STATE
006231  260740 006434'  27190           TSX     IN20;           5 - CHANGE TO RIB STATE
006232  260740 006454'  27200           TSX     IN50;           6 - RECORD IN SIGNAL
006233  260740 006441'  27210           TSX     IN40;           7 - SEND QUEUE MESSAGE
006234  260740 006261'  27220           TSX     OFFQ;           8 - OFF FROM USER IN THE QUEUE
006235  260740 006274'  27230           TSX     OFFNQ;          9 - OFF FROM ALL OTHERS
006236  260740 006326'  27240           TSX     ON.R;           10 -  ON FROM OFF STATION
006237  260740 006537'  27250           TSX     TODSU;          11 - TO FROM STATION IN DSU
006240  260740 006542'  27260           TSX     TOCK;           12 - TO FROM CHOKED STATION
006241  260740 006555'  27270           TSX     TO99;           13 - TO FROM ALL OTHERS
006242  260740 006323'  27280           TSX     OFFD;           14 - OFF FROM A DISCING USER
006243  260740 006321'  27290           TSX     OFF5;           15 - OFF FROM MORE CORE
006244  260740 006436'  27300           TSX     IN30;           16 - INTERRUPT FROM DISC QUEUE
006245  000000 000000   27310           Z;                      SPARE
                        27320   H20=T3
```

SEND MESSAGES TO ALL IN THE QUEUE

```
006246  551240 000001    27340    SAQM:   HRRZI   F,1;          INITIALIZE PLACE IN QUEUE
006247  550140 002137'   27350            HRRZ    D,QM;         ADDRESS OF FIRST
006250  550700 000003    27360    SA1:    HRRZ    S,D
006251  275700 000141'   27370            SUBI    S,S.Q;        COMPUTE USER INDEX
006252  260740 010343'   27380            TSX     GETBUF
006253  263740 000000    27390            DONE
006254  260740 006342'   27400            TSX     SQM;          SEND MESSAGE
006255  550143 000000    27410            HRRZ    D,0(D) ;      NEXT USER
006256  350000 000005    27420            AOS     F;            BUMP PLACE IN QUEUE
006257  326140 006250'   27430            JUMPN   D,SA1;        GO AROUND IF MORE USERS
006260  263740 000000    27440            DONE
```

```
006261  370000 002164'  27460     OFFQ:   SOS     CT28;            OFF FROM USER IN THE QUEUE
006262  402016 002572'  27470             SETZM   MINT(S);         ZERO INITIALS
006263  020000 000023   27480             CHS     OF.S
                        27490             FSW H
006264  200340 006125'      MOVE H,SWITCH
006265  606340 100000   27500             TRNN    H,OFFS;          DONT RE-ENABLE IF BLAST OFF
006266  264000 004475'  27510             JSR     C31;             RE-ENABLE
006267  200340 000016   27520             MOVE    H,S
006270  332000 002137'  27530             SKIPE   QM;              SEND MESSAGES TO ALL IN QUEUE
006271  260740 006246'  27540             TSX     SAQM
006272  200700 000007   27550             MOVE    S,H
006273  254000 006276'  27560             JRST    OFF
006274  370000 002160'  27570     OFFNQ:  SOS     USERS;           OFF FROM USER NOT IN QUEUE
006275  020000 000000   27580             CHS     TOF.S
006276  200216 000331'  27590     OFF:    MOVE    E,S.BUF(S);      STATION BUFFER HEADER
006277  322200 006302'  27600             JUMPE   E,OF1;           JUMP IF NO MORE BUFFERS AVAILABLE
006300  260740 010365'  27610             TSX     MBA;             PUT BUFFER ON AVAILABLE
006301  254000 006276'  27620             JRST    OFF;             CONTINUE TILL ALL ARE PUT AWAY
                        27630
                        27640     ;               TURN ON A USER FROM THE QUEUE IF PROPER
                        27650
006302  336000 002137'  27660     OF1:    SKIPN   QM;              SKIP IF SOME IN QUEUE
006303  263740 000000   27670             DONE
                        27680
006304  200200 002160'  27690     OF2:    MOVE    E,USERS
006305  311200 000324'  27700             CAML    E,N.SON;         SKIP IF USERS < MAX
006306  263740 000000   27710             DONE
                        27720
006307  550700 002137'  27730             HRRZ    S,QM
006310  275700 000141'  27740             SUBI    S,S.Q;           INDEX OF TOP USER IN QM
006311  332016 000331'  27750             SKIPE   S.BUF(S)
006312  263740 000000   27760             DONE;                    WELL GET HIM LATER IF NOW OUTPUTTING
006313  020000 000001   27770             CHS     ON.S;            CHANGE STATE TO "ON"
006314  370000 002164'  27780             SOS     CT28;            DECREMENT # IN QUEUE
006315  350000 002160'  27790             AOS     USERS
006316  332000 002137'  27800             SKIPE   QM;              TEST FOR NONE IN THE QUEUE
006317  260740 006246'  27810             TSX     SAQM;            SEND MESSAGES TO ALL IN QUEUE
006320  263740 000000   27820             DONE
                        27830
006321  135000 000267'  27840     OFF5:   LDB     A,S.DU
006322  322000 006274'  27850             JUMPE   A,OFFNQ;         JUMP IF NOT USING DISC
006323  201000 000001   27860     OFFD:   MOVEI   A,1
006324  137000 000265'  27870             DPB     A,S.OFR;         FLAG USER OFF DURING DISC TRANSFER
006325  263740 000000   27880             DONE
```

ON-SIGNAL ROUTINES

```
006326  550200 002160'  27900    ON.R:  HRRZ   E,USERS;      GET NUMBER OF USERS
006327  315200 000324'  27910           CAMGE  E,N.SON;      BIGGER THAN MAX?
006330  254000 006337'  27920           JRST   ON5;          NO, GO TO TURN ON.
006331  020000 000022   27930           CHS    QM.S;         CHANGE STATE TO "QUEUE MESSAGE"
006332  350240 002164'  27940           AOS    F,CT28;       COUNT AND FETCH QUEUE #
006333  260740 010343'  27950           TSX    GETBUF;       GET A BUFFER
006334  263740 000000   27960           DONE;                NONE, FORGET IT
006335  260740 006342'  27970           TSX    SQM;          SEND QUEUE MESSAGE
006336  263740 000000   27980           DONE
                        27990
006337  020000 000001   28000    ON5:   CHS    ON.S;         CHANGE STATE TO "ON"
006340  350000 002160'  28010           AOS    USERS;        COUNT USERS
006341  263740 000000   28020           DONE
```

```
SEND A QUEUE MESSAGE

                                28040
006342  200100 010564'          28050      SQM:    MOVE    C,[POINT 7,MINT(S)]
006343  010000 000121           28060              INS     "Q"
006344  010000 000043           28070              INS     "#"
006345  200000 000005           28080              MOVE    A,F
006346  260740 006405'          28090              TSX     CTDB;           PUT QUEUE # IN FOR INITIALS
006347  010000 000040           28100              INS     " "
006350  305240 000012           28110              CAIGE   F,!D10
006351  010000 000040           28120              INS     " "
006352  200000 000005           28130              MOVE    A,F;            RECOVER QUEUE #
006353  200100 010565'          28140              MOVE    C,[POINT 7,QM2,6]
006354  260740 006410'          28150              TSX     CTDR;           CONVERT AND DEPOSIT QUEUE #
006355  200100 010566'          28160              MOVE    C,[POINT 7,QM4]
006356  200000 000031'          28170              MOVE    A,HR
006357  260740 006410'          28180              TSX     CTDR;           CONVERT AND DEPOSIT HOURS
006360  133000 000002           28190              IBP     C;              SKIP PAST THE COLON
006361  200000 000032'          28200              MOVE    A,MIN
006362  260740 006403'          28210              TSX     CTD;            CONVERT AND DEPOSIT MINUTES
006363  030000 006371'          28220              SEND    QM1;            "YOU ARE # IN THE QUEUE"
                                28230              FSW A
006364  200000 006264'              MOVE A,SWITCH
006365  602000 200000           28240              TRNE    A,SDS;          IF SHUTTING DOWN
006366  030000 010512'          28250              SEND    SDM;            SEND SHUT DOWN MESSAGE TO QUEUE
006367  260740 010305'          28260              TSX     PUTB;           SEND TO USER
006370  263740 000000           28270              DONE
                                28280
                                28290
006371  006372'000011           28300      QM1:    XWD     .+1,QM3-.-1
006372  064241 203416           28310              OCT     64241203416;    CR,LF,LF,BELL,BELL
006373  442207 246632           28320      QM4:    ASCII   /HH:MM/
006374  202635 772500           28330              OCT     202635772500;    YOU
006375  607454 520334           28340              OCT     607454520334;   ARE N
006376  727334 262744           28350              OCT     727334262744;   UMBER
006377  202613 020322           28360      QM2:    OCT     202613020322;    XX I
006400  671016 464312           28370              OCT     671016464312;   N THE
006401  203436 562752           28380              OCT     203436562752;    QUEU
006402  625341 500000           28390              OCT     625341500000;   E."CR"
                                28400      QM3=.
```

CONVERT TO TWO DECIMAL CHARACTERS

```
                                28420     ;        (A) ARE CONVERTED TO CHARACTERS AT POINTER IN C, B DESTROYED
                                28430
006403  231000 000012           28440     CTD:     IDIVI   A,↑D10;
006404  254000 006414'          28450              JRST    CTD1;            OUTPUT HIGH ORDER ZEROS
                                28460
006405  231000 000012           28470     CTDB:    IDIVI   A,↑D10;          CONVERT TWO DIGITS TO ASCII
006406  322000 006416'          28480              JUMPE   A,CTD2;          SUPPRESS LEADING ZEROS
006407  254000 006414'          28490              JRST    CTD1
                                28500
006410  231000 000012           28510     CTDR:    IDIVI   A,↑D10
006411  326000 006414'          28520              JUMPN   A,CTD1
006412  201000 000040           28530              MOVEI   A," ";             REPLACE LEADING ZEROS WITH BLANKS
006413  254000 006415'          28540              JRST    .+2
006414  435000 000060           28550     CTD1:    ORI     A,60
006415  136000 000002           28560              IDPB    A,C
006416  435040 000060           28570     CTD2:    ORI     B,60
006417  136040 000002           28580              IDPB    B,C
006420  263740 000000           28590              DONE
                                28600
                                28610     ,                COMPUTE BINARY SECONDS FROM CLOCK CELLS
                                28620
006421  200040 000031'          28630     ISEC:    MOVE    B,HR
006422  221040 000074           28640              IMULI   B,↑D60
006423  270040 000032'          28650              ADD     B,MIN
006424  221040 000074           28660              IMULI   B,↑D60
006425  270040 000033'          28670              ADD     B,SEC
006426  263740 000000           28680              DONE
```

```
006427  135040 000267'  28700   IN10:  LDB     B,S.DU
006430  332000 000001   28710          SKIPE   B;              SKIP IF NOT A DISC USER
006431  254000 006454'  28720          JRST    IN50;           FROM DISC USER: FLAG IT
006432  020000 000003   28730   IN11:  CHS     RI.S;           CHANGE TO REQUEST IN STATE
006433  254000 006452'  28740          JRST    IN99
                        28750
006434  020000 000004   28760   IN20:  CHS     RIB.S
006435  254000 006452'  28770          JRST    IN99
                        28780
006436  402000 000001   28790   IN30:  SETZM   B
006437  137040 000267'  28800          DPB     B,S.DU;         RESET DISC FLAG
006440  254000 006432'  28810          JRST    IN11
                        28820
006441  260740 010343'  28830   IN40:  TSX     GETBUF
006442  254000 006452'  28840          JRST    IN99;           FORGET IT IF NO BUFFER
006443  201240 000001   28850          MOVEI   F,1;            INITIALIZE QUEUE COUNT
006444  550040 002137'  28860          HRRZ    B,QM
006445  306056 000141'  28870          CAIN    B,S.Q(S);       SKIP IF NOT THE  ONE OF INTEREST
006446  254000 006451'  28880          JRST    .+3
006447  550041 000000   28890          HRRZ    B,0(B)
006450  344240 006445'  28900          AOJA    F,.-3;          COUNT PLACE IN QUEUE
006451  260740 006342'  28910          TSX     SQM;            SEND HIM THE MESSAGE
006452  350000 002170'  28920   IN99:  AOS     CT27;           COUNT INS WHICH RESULT IN ACTION
006453  263740 000000   28930          DONE
                        28940
006454  201000 000001   28950   IN50:  MOVEI   A,1
006455  137000 000266'  28960          DPB     A,S.INR;        FLAG INTERRUPT SIGNAL
006456  254000 006452'  28970          JRST    IN99
                        28980
```

CARRIER-RETURN-SIGNAL ROUTINES

```
006457  350000 002173'  29000   CR.R:   AOS     CT31;            COUNT CARRIAGE RETURNS
006460  350000 002442'  29010           AOS     T7+2
006461  550216 000331'  29020           HRRZ    E,S.BUF(S);      GET BUFFER LOCATION
006462  574144 000002   29030           HLRE    D,2(E);          GET CHARACTER COUNT
006463  335000 000003   29040           SKIPGE  D
006464  462000 000003   29050           SETCMM  D;               COMPLEMENT IF TTY STATION
006465  272140 002174'  29060           ADDM    D,CT25;          GROSS SUM
006466  260740 006507'  29070           TSX     CVTLIN;          FORM DISTRIBUTION
006467  350001 002332'  29080           AOS     CPIL(B)
006470  550056 002522'  29090           HRRZ    B,USTAT(S);      TICKS SINCE LAST CR
006471  221040 000074   29100           IMULI   B,↑D60;          SCALE
006472  264000 004262'  29110           JSR     CVTL
006473  350002 002320'  29120           AOS     DCPI(C);         COUNT COMP/INTERACTION IN RANGE
006474  260740 006421'  29130           TSX     ISEC;            CURRENT TIME IN SECS
006475  554116 002522'  29140           HLRZ    C,USTAT(S);      LAST GREEN TIME
006476  516056 002522'  29150           HRLZM   B,USTAT(S);      THIS GREEN TIME
006477  274040 000002   29160           SUB     B,C
006500  335000 000001   29170           SKIPGE  B
006501  271040 250600   29180           ADDI    B,↑D24*↑D3600;   MIDNIGHT CORRECTION
006502  221040 000012   29190           IMULI   B,↑D10;          UNITS ARE NOW 1/10TH MINUTE
006503  264000 004262'  29200           JSR     CVTL
006504  350002 002306'  29210           AOS     DTIM(C);         INTERACTION TIME DISTRIBUTION
006505  020000 000002   29220           CHS     RC.S;            CHANGE STATE TO CR
006506  263740 000000   29230           DONE
                       29240
                       29250   ;       CONVERT TO LINEAR RANGE
                       29260
006507  201040 000000   29270   CVTLIN: MOVEI   B,0
006510  201100 000005   29280           MOVEI   C,5
006511  317140 000002   29290   CVT1:   CAMG    D,C
006512  263740 000000   29300           DONE
006513  271100 000005   29310           ADDI    C,5
006514  305040 000011   29320           CAIGE   B,↑D9
006515  344040 006511'  29330           AOJA    B,CVT1
006516  263740 000000   29340           DONE
                       29350
```

```
006517  336016 000331'  29370   TO:     SKIPN   S.BUF(S) ;      HALT IF NO BUFFER
006520  040000 000007   29380   H7:     HALT    7;              NO SUCH BUFFER
006521  350000 002171'  29390           AOS     CT16;           COUNT LINES
006522  350000 002443'  29400           AOS     T7+3
006523  550216 000331'  29410           HRRZ    E,S.BUF(S) ;    CHARACTER COUNT
006524  550144 000002   29420           HRRZ    D,2(E)
006525  272140 002175'  29430           ADDM    D,CT26;         GLOBAL COUNT
006526  260740 006507'  29440           TSX     CVTLIN;         FIND DISTRIBUTIONAL RANGE
006527  350001 002344'  29450           AOS     CPOL(B)
006530  204056 000331'  29460           MOVS    B,S.BUF(S) ;    PICK UP BUFFER HEADER
006531  430056 000331'  29470           XOR     B,S.BUF(S) ;    SEE IF LEFT=RIGHT
006532  322040 006536'  29480           JUMPE   B,TO1;          JUMP IF LAST BUFFER
006533  260740 010365'  29490           TSX     MBA;            PUT BUFFER ON AVAILABLE LIST
006534  260740 000000   29500           CALL    C27;            INITIATE TRANSMISSION
006535  350017 000000   29510           AOS     0(PP) ;         BUMP TO SECOND EXIT
006536  263740 000000   29520   TO1:    DONE
                       29530
006537  260740 006517'  29540   TODSU:  TSX     TO;             TO PREAMBLE
006540  260740 006560'  29550           TSX     SG;             LAST BUFFER SO CHANGE TO GREEN
006541  263740 000000   29560           DONE
                       29570
006542  260740 006517'  29580   TOCK:   TSX     TO;             "TO" PREAMBLE
006543  254000 006552'  29590           JRST    TOCK1;          ALWAYS UNCHOKE ON LAST BUFFER
006544  200200 000322'  29600           MOVE    E,N.UC;         GET UNCHOKE #
006545  550056 000331'  29610           HRRZ    B,S.BUF(S)
006546  322040 006553'  29620           JUMPE   B,TO3;          GET UNCHOKED (# BUFFERS <= CK #)
006547  550041 000000   29630           HRRZ    B,0(B)
006550  363200 006546'  29640           SOJLE   E,.-2
006551  263740 000000   29650           DONE;                   STAY CHOKED (# BUFFERS > CK #)
                       29660
006552  260740 010365'  29670   TOCK1:  TSX     MBA
006553  020000 000005   29680   TO3:    CHS     UC.S;           CHANGE STATE TO UNCHOKE
006554  263740 000000   29690           DONE
                       29700
006555  260740 006517'  29710   TO99:   TSX     TO;             "TO" PREAMBLE
006556  260740 010365'  29720           TSX     MBA;            LAST BUFFER- MAKE IT AVAILABLE
006557  263740 000000   29730           DONE
```

SWITCH CONSOLE TO GREEN STATE

```
                                29750
006560  550056  000331'  29760    SG:    HRRZ    B,S.BUF(S);     POINTER TO BUFFER
006561  200200  004130'  29770           MOVE    E,BLANKS
006562  202201  000003   29780           MOVEM   E,3(B);         BLANK FIRST TEXT WORD
006563  200200  000001   29790           MOVE    E,B
006564  271200  000022   29800           ADDI    E,BUFSIZ;       LAST BUFFER LOCATION
006565  271040  000003   29810           ADDI    B,3;            BUMP DOWN TO FIRST TEXT WORD
006566  504040  000001   29820           HRL     B,B
006567  350000  000001   29830           AOS     B;              MAKE UP BLT CONTROL WORD
006570  251044  000000   29840           BLT     B,0(E);         BLANK THE BUFFER
006571  260740  006421'  29850           TSX     ISEC
006572  554116  002522'  29860           HLRZ    C,USTAT(S)
006573  274040  000002   29870           SUB     B,C;            TASK TIME: CR TO SG
006574  335000  000001   29880           SKIPGE  B
006575  271040  250600   29890           ADDI    B,↑D24*↑D3600
006576  221040  000074   29900           IMULI   B,↑D60;         DISPLAY UNITS ARE 1 SECOND
006577  264000  004262'  29910           JSR     CVTL
006600  350002  002452'  29920           AOS     T8(C)
006601  020000  000015   29930           CHS     GR.S;           SWITCH TO GREEN STATE
006602  260740  000000   29940           CALL    C28;            SWITCH TO USER
006603  263740  000000   29950           DONE
```

```
ACCUMULATE COUNTS ON THE MINUTE

006604  402000 000000   29970     QCTR:   SETZM   A
006605  322040 006611'  29980             JUMPE   B,.+4
006606  550041 000000   29990             HRRZ    B,0 (B)
006607  350000 000000   30000             AOS     A
006610  326040 006606'  30010             JUMPN   B,.-2
006611  263740 000000   30020             DONE
                        30030
                        30040     GETCT:  QCT GR,CT33; COUNT GREEN QUEUE

006612  550040 002132'  HRRZ B,GR
006613  260740 006604'  TSX QCTR
006614  202000 002161'  MOVEM A,CT33
                        30050             QCT COM,CT34;          COUNT COMPUTE QUEUE

006615  550040 002124'  HRRZ B,COM
006616  260740 006604'  TSX QCTR
006617  202000 002162'  MOVEM A,CT34
                        30060             QCT CK,CT35; COUNT WAIT FOR BUFFER QUEUE

006620  550040 002130'  HRRZ B,CK
006621  260740 006604'  TSX QCTR
006622  202000 002163'  MOVEM A,CT35
006623  200000 002213'  30070             MOVE    A,CT44;        CUMULATIVE COMPUTE TIME
006624  221000 023420   30080             IMULI   A,↑D10000;     SCALE
006625  230000 000042'  30090             IDIV    A,TIME
006626  202000 002212'  30100             MOVEM   A,CT43;        1/100THS % OF TOTAL TIME
006627  201100 000001   30110             MOVEI   C,1
006630  200002 002176'  30120     GT1:    MOVE    A,CT27A (C)
006631  231000 000010   30130             IDIVI   A,↑D8
006632  221000 000002   30140             IMULI   A,2
006633  272002 002176'  30150             ADDM    A,CT27A (C) ;  MAKE LOG PRINT OCTAL STATION #
006634  365100 006630'  30160             SOJGE   C,GT1;         CT27A AND CT27B
006635  200040 002165'  30170             MOVE    B,CT13;        TICKS THIS MINUTE
006636  221040 001750   30180             IMULI   B,↑D1000
006637  231040 000002   30190             IDIVI   B,2;           DISPLAY AT 2 MS.
006640  230040 002166'  30200             IDIV    B,CT14;        1/10THS OF MS./STATEMENT
006641  264000 004262'  30210             JSR     CVTL
006642  332000 002166'  30220             SKIPE   CT14;          SKIP IF NO STATEMENTS INTERPRETED
006643  350002 002414'  30230             AOS     T5 (C) ;       RECORD IN DISTRIBUTION
006644  263740 000000   30240             DONE
```

DISPLAY USERS INITIALS AND STATION NUMBER

```
                                30260      ;        USES REGISTERS A,B,C,D,E,F,G. S MUST CONTAIN CONTEXT FOR OUTPUT
                                30270
006645  200140 010567'          30280      DISINT:  MOVE    D,[XWD -N.S,0]; WELL CHECK ALL STATIONS
006646  201300 000010           30290      DI1:     MOVEI   G,!D8;          AND OUTPUT 8 PER LINE
006647  260740 010422'          30300               TSX     CMESS
006650  260740 006653'          30310               TSX     DI2;MAKE UP INITIALS LINE
006651  321140 006646'          30320               JUMPL   D,DI1;          BACK AROUND IF MORE STATIONS
006652  263740 000000           30330               DONE
                                30340
006653  336243 002572'          30350      DI2:     SKIPN   F,MINT(D);      SKIP IF INITIALS ARE THERE
006654  254000 006661'          30360               JRST    DI3;            NONE ON THAT STATION
006655  550000 000003           30370               HRRZ    A,D
006656  260740 006700'          30380               TSX     OCT;            CONVERT STA # TO OUTPUT
                                30390               XMT 5,F;     AND THE INITIALS
                                     OPDEF X[5B12]
006657  000240 000005           X F
006660  370000 000006           30400               SOS     G;              COUNT INITIALS IN BUFFER
006661  252140 006663'          30410      DI3:     AOBJP   D,DI4;          JUMP IF WE HAVE FINISHED THE TABLE
006662  327300 006653'          30420               JUMPG   G,DI2;          JUMP IF BUFFER NOT FULL
006663  263740 000000           30430      DI4:     DONE
```

```
                                30450      ,        DISPLAY LOG-ONS AND LOG-OFFS IF PSEUDO SWITCH DOAF IS SET
                                30460      ,        D CONTAINS "ON-" OR "OFF-"
                                30470      ;        DESTROYS A,B,C,E,S.
                                30480
                                30490      DOF:     FSW A;       GET THE SWITCHES
006664  200000 006364'              MOVE A,SWITCH
006665  606000 000010           30500               TRNN     A,DOAF;          SKIP IF DISPLAY INDICATED
006666  263740 000000           30510               DONE                      ;FORGET IT IF NONE
006667  550240 000016           30520               HRRZ     F,S
006670  260740 010422'          30530               TSX      CMESS
006671  260740 006673'          30540               TSX      DOF1;            MAKE UP LOG ON/OFF LINE
006672  263740 000000           30550               DONE
                                30560
                                30570      DOF1:    XMT 4,D;     "ON" OR "OFF"
                                     OPDEF X[4B12]
006673  000200 000003           X D
006674  550000 000005           30580               HRRZ     A,F
006675  260740 006700'          30590               TSX      OCT;             CONVERT STA # TO OUTPUT
                                30600               XMT 5,INITIALS
                                     OPDEF X[5B12]
006676  000240 004517'          X INITIALS
006677  263740 000000           30610               DONE
                                30620
                                30630
                                30640      ,        CONVERT STATION # TO OCTAL ASCII (TWO DIGITS ONLY)
                                30650
006700  260740 006703'          30660      OCT:     TSX      OCT1
006701  010000 000055           30670               INS      "-"
006702  263740 000000           30680               DONE
                                30690
006703  231000 000010           30700      OCT1:    IDIVI    A,10
006704  435000 000060           30710               ORI      A,60
006705  136000 000002           30720               IDPB     A,C;             CONVERT AND OUTPUT STA #
006706  435040 000060           30730               ORI      B,60
006707  136040 000002           30740               IDPB     B,C
006710  263740 000000           30750               DONE
```

DISPLAY STATISTICAL DISTRIBUTIONS

```
006711  201240 000020   30770   ODIS:   MOVEI   F,STAT2;        DISPLAY TABLE SIZE
006712  200445 006716'  30780           MOVE    J,STAT1(F);     STAT TABLE POINTER
006713  260740 007057'  30790           TSX     TABO;           FORMAT AND OUTPUT STAT LINE
006714  365240 006712'  30800           SOJGE   F,.-2
006715  263740 000000   30810           DONE
                       30820
                       30830   STAT1:
006716  000000 002510'  30840           XWD     0,T11;          T11, T10, AND T9 ARE USED
006717  000000 002476'  30850           XWD     0,T10;          BY JOE TO RECORD COUNTS OF EXECUTION
006720  000000 002464'  30860           XWD     0,T9;           BY VERB TYPE
006721  000000 002402'  30870           XWD     0,TINT;         TOTAL I/O CHARACTERS
006722  000000 002370'  30880           XWD     0,OINT;         OUTPUT INTERRUPTS
006723  000000 002356'  30890           XWD     0,IINT;         INPUT INTERRUPTS
006724  000000 002344'  30900           XWD     0,CPOL;         CHAR/OUTPUT LINE
006725  000000 002332'  30910           XWD     0,CPIL;         CHAR/INPUT LINE
006726  000000 002274'  30920           XWD     0,DSIZE;        SIZE
006727  000000 002320'  30930           XWD     0,DCPI;         COMPUTE TIME PER INTERACTION
006730  000000 002306'  30940           XWD     0,DTIM;         INTERACTION TIME
006731  000000 002262'  30950           XWD     0,CCTIM;        COMPUTE TIME
006732  000000 002250'  30960           XWD     0,SESTIM;       SESSION TIME
006733  000000 002414'  30970           XWD     0,T5;           INTERPRETATION RATE
006734  000000 002452'  30980           XWD     0,T8;           TASK TURN-AROUND TIME
006735  000000 002440'  30990           XWD     0,T7;           MISC. STATISTICS
006736  000000 002426'  31000           XWD     0,T6;           GENERAL COUNTS AND BLOCKSIZE DISTRIB.
                       31010   STAT2=.-STAT1-1
                       31020
```

COUNT TIME AND GATHER STATISTICS

```
006737  200040 000033'  31040     STAT:   MOVE    B,SEC
006740  306040 000036   31050             CAIN    B,↑D30
006741  260740 003532'  31060             TSX     TRST;           RESTART TAPE
006742  275040 000074   31070             SUBI    B,↑D60
006743  321040 007135'  31080             JUMPL   B,ST2.6;        DONE IF NOT A NEW MINUTE
                        31090
                        31100     ;       NEW MINUTE
                        31110
006744  571700 777777   31120             HRREI   S,-1;           CONTEXT FOR CONSOLE I/O
006745  202040 000033'  31130             MOVEM   B,SEC
006746  260740 007212'  31140             TSX     UP1;            CONVERT DATE TO ASCII IN CASE OF RECENT CHANGE
006747  350040 000032'  31150             AOS     B,MIN;          BUMP THE MINUTE COUNT
006750  231040 000017   31160             IDIVI   B,K3;
                        31170             FSW D
006751  200140 006664'      MOVE D,SWITCH
006752  606140 000002   31180             TRNN    D,DUI;          CHECK DISPLAY SWITCH
006753  336000 000002   31190             SKIPN   C;              DISPLAY INITIALS IF FP(MIN/K3)=0
006754  260740 006645'  31200             TSX     DISINT
                        31210             FSW D
006755  200140 006751'      MOVE D,SWITCH
006756  626140 000020   31220             TRZN    D,DSTAT;        SKIP IF WE ARE TO DISPLAY STATISTICS
006757  254000 006762'  31230             JRST    STA3
006760  202140 006755'  31240             MOVEM   D,SWITCH
006761  260740 006711'  31250             TSX     ODIS;           DISPLAY STATISTICS
                        31260     STA3:   FSW D
006762  200140 006760'      MOVE D,SWITCH
006763  606140 000100   31270             TRNN    D,OLSW;         SKIP IF DISPLAY
006764  254000 006775'  31280             JRST    STA1
006765  201240 000021   31290             MOVEI   F,STAT2+1
006766  135440 004140'  31300             LDB     J,C5
006767  303440 007043'  31310             CAILE   J,STA2
006770  254000 006775'  31320             JRST    STA1
006771  322440 006775'  31330             JUMPE   J,STA1
006772  274240 000011   31340             SUB     F,J
006773  200445 006716'  31350             MOVE    J,STAT1(F)
006774  260740 007057'  31360             TSX     TABO
```

COUNT TIME AND GATHER STATISTICS

```
006775  200140 010544'  31380    STA1:   MOVE    D,[POINT 6,CT48A]
006776  201040 000004   31390            MOVEI   B,4
006777  350000 002426'  31400            AOS     T6
007000  134100 000003   31410            ILDB    C,D
007001  272101 002433'  31420            ADDM    C,T6+5(B)
007002  367040 007000'  31430            SOJG    B,.-2
007003  200040 002204'  31440            MOVE    B,CT18
007004  272040 002433'  31450            ADDM    B,T6+5;         TOTAL SAVES
007005  200040 002207'  31460            MOVE    B,CT19
007006  272040 002432'  31470            ADDM    B,T6+4;         TOTAL LOADS
007007  200040 002203'  31480            MOVE    B,CT19A
007010  272040 002431'  31490            ADDM    B,T6+3;         TOTAL DRUM ACTIONS
007011  200040 002166'  31500            MOVE    B,CT14
007012  272040 002430'  31510            ADDM    B,T6+2;         TOTAL STATEMENTS
007013  200040 002165'  31520            MOVE    B,CT13
007014  272040 002427'  31530            ADDM    B,T6+1;         TOTAL TICKS
007015  200040 002205'  31540            MOVE    B,CT18A
007016  272040 002440'  31550            ADDM    B,T7;           DISCARDS
007017  332000 002201'  31560            SKIPE   CT11;           SKIP IF A FULL USE MINUTE
007020  254000 007025'  31570            JRST    STA1.5
007021  350000 002447'  31580            AOS     T7+7;           COUNT THEM
007022  201040 007020   31590            MOVEI   B,↑D3600
007023  274040 002165'  31600            SUB     B,CT13;         COMPUTE OVERHEAD TICKS
007024  272040 002450'  31610            ADDM    B,T7+8;         AND ACCUMULATE THEM
007025  200040 000032'  31620    STA1.5: MOVE    B,MIN
007026  275040 000074   31630            SUBI    B,↑D60
007027  321040 007114'  31640            JUMPL   B,ST1;          JUMP IF NOT A NEW HOUR
                        31650
                        31660    ;       NEW HOUR
                        31670
007030  402000 002210'  31680            SETZM   CT17
007031  402000 002211'  31690            SETZM   CT23;           CLEAR IDG COUNTERS
                        31700
007032  202040 000032'  31710            MOVEM   B,MIN;          UPDATE MINUTES
007033  350040 000031'  31720            AOS     B,HR
007034  606040 000003   31730            TRNN    B,3;            OUTPUT STATISTIC EACH FOUR HOURS
007035  260740 006711'  31740            TSX     ODIS
007036  200040 000031'  31750            MOVE    B,HR
007037  275040 000030   31760            SUBI    B,↑D24
007040  321040 007043'  31770            JUMPL   B,STA2;         JUMP IF NOT A NEW DAY
007041  202040 000031'  31780            MOVEM   B,HR
007042  260740 007162'  31790            TSX     UDATE;          GO TO INCREMENT DATE
007043  260740 007100'  31800    STA2:   TSX     HMES;           HEADING LINE
007044  254000 007114'  31810            JRST    ST1
```

TAPE OUTPUT ROUTINE

```
                          31830     ,       HANDLES ALL TAPE OUTPUT EXCEPT ACCOUNTING RECORDS
                          31840
007045  260740 010343'    31850     SOUT:   TSX     GETBUF
007046  263740 000000     31860             DONE
007047  504140 000001     31870             HRL     D,B
007050  544141 000000     31880             HLR     D,(B);          CONTROL WORD FOR MOVE
007051  275200 000002     31890             SUBI    E,2;            FAKE UP FOR MOVE ROUTINE
007052  505200 000003     31900             HRLI    E,D;            CONTROL WORD ADDRESS
007053  260740 010433'    31910             TSX     BJ9.1;          MOVE WORDS TO BUFFER
007054  271200 000002     31920             ADDI    E,2;            RESTORE
007055  260740 010305'    31930             TSX     PUTB;           START OUTPUT
007056  263740 000000     31940             DONE
                          31950
```

OUTPUT CUMULATIVE STATISTICS LINE

```
                          31970      ;        OUTPUTS TABLE OF SIZE 10 FROM (J), USES LOTS OF REGISTERS
                          31980
007057  260740 010422'    31990      TABO:    TSX      CMESS
007060  260740 007062'    32000               TSX      TABO1;          FORMAT THE LINE
007061  263740 000000     32010               DONE
                          32020
007062  571340 777766     32030      TABO1:   HRREI    H,-12
007063  201300 000006     32040      ST.1:    MOVEI    G,6;            FIELD SIZE
007064  201400 000040     32050               MOVEI    I," ";            ALTERNATE OUTPUT CHARACTER
007065  402000 000003     32060               SETZM    D;              INITIAL OVERPUNCH
007066  200011 000000     32070               MOVE     A,0(J);         GET ELEMENT FROM TABLE
007067  350000 000011     32080               AOS      J
007070  260740 007240'    32090               TSX      FRMT;           FORMAT ELEMENT FOR OUTPUT LINE
007071  346340 007063'    32100               AOJN     H,ST.1
                          32110               XMT 5,ST.12
                              OPDEF X[5B12]
007072  000240 007077'    X ST.12
007073  201000 000021     32120               MOVEI    A,STAT2+1
007074  274000 000005     32130               SUB      A,F
007075  260740 006403'    32140               TSX      CTD;            OUTPUT LINE NUMBER ON STATISTICS
007076  263740 000000     32150               DONE
                          32160
007077  201004 021500     32170      ST.12:   ASCII    ?   # ?
```

OUTPUT HOURLY HEADING LINE

```
007100  260740 010422'   32190     HMES:   TSX     CMESS
007101  260740 007103'   32200             TSX     HMES1
007102  263740 000000    32210             DONE
                         32220
007103  010000 000012    32230     HMES1:  INS     12;             LINE FEED
007104  200000 000031'   32240             MOVE    A,HR
007105  260740 006403'   32250             TSX     CTD;            HOURS TO HEADING LINE
007106  010000 000072    32260             INS     ":"
007107  010000 000040    32270             INS     " "
                         32280             XMT 10,DATE
                             OPDEF X[10B12]
007110  000400 000023'   X DATE
                         32290             XMT 2,CRLF
                             OPDEF X[2B12]
007111  000100 004134'   X CRLF
                         32300             XMT 110,HEAD;         72 HEADING CHARACTERS
                             OPDEF X[110B12]
007112  004400 002231'   X HEAD
007113  263740 000000    32310             DONE
```

```
007114  260740 006612'  32330   ST1:    TSX     GETCT;          COMPUTE NEW COUNTS
007115  332000 000032'  32340   ST1B:   SKIPE   MIN
007116  254000 007121'  32350           JRST    .+3
007117  402000 000042'  32360           SETZM   TIME;           RESET COMPUTE % ON THE HOUR
007120  402000 002213'  32370           SETZM   CT44
007121  200040 002174'  32380           MOVE    B,CT25
007122  270040 002175'  32390           ADD     B,CT26
007123  270040 002176'  32400           ADD     B,CT27A
007124  270040 002177'  32410           ADD     B,CT27B
007125  322040 007130'  32420           JUMPE   B,ST2.5-1;      NO OUTPUT IF NO ERRORS OR CHAR.
007126  260740 010422'  32430           TSX     CMESS
007127  260740 007136'  32440           TSX     ST1A;           FORMAT TO BUFFER
007130  571040 777745   32450           HRREI   B,-CTPARS
007131  200201 002745'  32460   ST2.5:  MOVE    E,CTPAR+CTPARS(B);  GET PARAMETER WORD
007132  607200 400000   32470           TLNN    E,400000;       SKIP IF SHOULD NOT ZERO THE COUNTER
007133  402004 000000   32480           SETZM   (E)
007134  341040 007131'  32490           AOJL    B,ST2.5
007135  263740 000000   32500   ST2.6:  DONE
                        32510
                        32520   ;       FORMAT A LINE FOR THE CONSOLE TTY
                        32530
007136  200000 000032'  32540   ST1A:   MOVE    A,MIN
007137  260740 006403'  32550           TSX     CTD;            CONVERT MINUTES
007140  571340 777745   32560           HRREI   H,-CTPARS;      TABLE SIZE
007141  201400 000040   32570   ST2:    MOVEI   I," ";            SET ALTERNATE OUTPUT CHARACTER
007142  402000 000003   32580           SETZM   D;              INITIALIZE TO NO OVERPUNCH
007143  135300 010570'  32590           LDB     G,[POINT 5,CTPAR+CTPARS(H),5]; GET FIELD SIZE
007144  135240 010571'  32600           LDB     F,[POINT 12,CTPAR+CTPARS(H),17]; GET SCALE FACTOR
007145  200047 002745'  32610           MOVE    B,CTPAR+CTPARS(H)
007146  200001 000000   32620           MOVE    A,(B);          GET COUNT
007147  326000 007152'  32630           JUMPN   A,.+3
007150  201140 000056   32640           MOVEI   D,".";            ALTERNATE CHAR IF ALL ZERO
007151  254000 007157'  32650           JRST    ST2.4
007152  230000 000005   32660           IDIV    A,F;            SCALE
007153  326000 007157'  32670           JUMPN   A,ST2.4;        GET REMAINDER IF QUOTIENT IS ZERO
007154  200000 000001   32680           MOVE    A,B
007155  221000 000012   32690           IMULI   A,↑D10;         MAKE A PLACE FOR THE *
007156  201140 000052   32700           MOVEI   D,"*";            MAY HELP BUT IT AINT PERFECT
007157  260740 007240'  32710   ST2.4:  TSX     FRMT;           FORMAT THE OUTPUT LINE
007160  341340 007141'  32720           AOJL    H,ST2;          GO AROUND IF MORE COUNTERS
007161  263740 000000   32730           DONE
                        32740
```

ROUTINE TO UPDATE THE DATE

```
007162  350040 000030'  32760   UDATE:  AOS     B,DAY;            ROUTINE TO INCREMENT AND UPDATE DATE.
007163  200140 010572'  32770           MOVE    D,[POINT 5,MONS]
007164  200100 000027'  32780           MOVE    C,MONTH
007165  133000 000003   32790           IBP     D
007166  366100 007165'  32800           SOJN    C,.-1;            COUNT DOWN TO THE CURRENT MONTH
007167  200100 000027'  32810           MOVE    C,MONTH
007170  135000 000003   32820           LDB     A,D
007171  302100 000002   32830           CAIE    C,2
007172  254000 007176'  32840           JRST    UD2;              NOT FEBRUARY
007173  200100 000026'  32850           MOVE    C,YEAR
007174  606100 000003   32860           TRNN    C,3;              CHECK FOR LEAP YEAR
007175  354000 000000   32870           AOSA    A;                LEAP YEAR - AN EXTRA FOR FEB.
007176  274040 000000   32880   UD2:    SUB     B,A
007177  323040 007212'  32890           JUMPLE  B,UP1;            NOT A NEW MONTH
007200  350040 000027'  32900           AOS     B,MONTH;          BUMP THE MONTH
007201  201000 000001   32910           MOVEI   A,1
007202  202000 000030'  32920           MOVEM   A,DAY;            SET DAY = 1
007203  260740 006007'  32930           TSX     MDS;              TIME FOR DISC ACCOUNTING
007204  200040 000027'  32940           MOVE    B,MONTH
007205  275040 000014   32950           SUBI    B,↑D12
007206  323040 007212'  32960           JUMPLE  B,UP1;            NOT A NEW YEAR
007207  350000 000026'  32970           AOS     YEAR
007210  201000 000001   32980           MOVEI   A,1
007211  202000 000027'  32990           MOVEM   A,MONTH;          SET MONTH = 1
                       33000
                       33010   ,       CONVERT DATE AND TIME TO ASCII
                       33020
007212  200000 004130'  33030   UP1:    MOVE    A,BLANKS
007213  202000 000023'  33040           MOVEM   A,DATE
007214  202000 000024'  33050           MOVEM   A,DATE +1;        BLANK THE DATE
007215  202000 000025'  33060           MOVEM   A,ATIM;           AND TIME
007216  200100 010573'  33070           MOVE    C,[POINT 7,DATE]
007217  200000 000027'  33080           MOVE    A,MONTH
007220  260740 006405'  33090           TSX     CTDB
007221  010000 000057   33100           INS     "/"
007222  200000 000030'  33110           MOVE    A,DAY
007223  260740 006405'  33120           TSX     CTDB
007224  010000 000057   33130           INS     "/"
007225  200000 000026'  33140           MOVE    A,YEAR
007226  260740 006405'  33150           TSX     CTDB
007227  200100 010574'  33160           MOVE    C,[POINT 7,ATIM]
007230  200000 000031'  33170           MOVE    A,HR
007231  260740 006405'  33180           TSX     CTDB;             CONVERT - SUPPRESS LEAD ZEROS
007232  010000 000072   33190           INS     ":"
007233  200000 000032'  33200           MOVE    A,MIN
007234  260740 006403'  33210           TSX     CTD;              CONVERT - WITH LEAD ZEROS
007235  263740 000000   33220           DONE
                       33230
                       33240   ADATE=UP1
                       33250           RADIX   10
007236  776377 577676  33260   MONS:   BYTE    (5) 31,28,31,30,31,30,31,31,30,31,30,31
007237  777377 574000
                       33270           RADIX   8
```

```
                        33290     ;        A CONTAINS SCALED COUNTER VALUE, C THE POINTER TO THE
                        33300     ;        OUTPUT LINE, G THE FIELD SIZE, AND I THE ALTERNATE
                        33310     ;        OUTPUT CHARACTER.
                        33320     ;        D, B ARE CLOBBERED.
                        33330
007240  315006 000000   33340     FRMT:    CAMGE   A,T80(G);        TOO BIG FOR FIELD?
007241  254000 007246'  33350              JRST    ST4;             NO
007242  201140 000041   33360              MOVEI   D,"!";             YES, SET OVERPUNCH
007243  231000 000012   33370     ST3:     IDIVI   A,↑D10;          SCALE BY TEN
007244  311006 007240'  33380              CAML    A,T80(G);        IF STILL TOO BIG:
007245  344140 007243'  33390              AOJA    D,ST3;           BUMP TO NEXT SPECIAL CHARACTER
007246  306300 000001   33400     ST4:     CAIN    G,1
007247  201400 000060   33410              MOVEI   I,"0";             SET ALTERNATE TO ZERO ON LAST COLUMN
007250  230006 000000   33420              IDIV    A,T80.99(G)
007251  326000 007254'  33430              JUMPN   A,ST5
007252  200000 000010   33440              MOVE    A,I;             IF ZERO GET ALTERNATE
007253  254000 007256'  33450              JRST    ST6
007254  201400 000060   33460     ST5:     MOVEI   I,"0";             SET ALTERNATE TO ZERO SINCE WE NOW HAVE A DIG
007255  435000 000060   33470              ORI     A,60;            ASCII BITS
007256  136000 000002   33480     ST6:     IDPB    A,C;             DUMP IN OUTPUT
007257  250000 000001   33490              EXCH    A,A+1;           GET REMAINDER
007260  367300 007246'  33500              SOJG    G,ST4;           GO AROUND IF MORE COLUMNS
007261  322140 007263'  33510              JUMPE   D,ST7;           JUMP IF NO OVERPUNCH
007262  137140 000002   33520              DPB     D,C;             DEPOSIT OVERPUNCH
007263  263740 000000   33530     ST7:     DONE
```

```
                          33550      ;        MASTER MONITOR ENTRY POINT
                          33560
007264  402000 000043'    33570      SIGPR:   SETZM   T.CU;            KEEP IT FROM COUNTING TOO HIGH
                          33580      SIGPR1=SIGPR
                          33590               FSW B;       GET DATA SWITCHES
007265  200040 006762'        MOVE B,SWITCH
007266  602040 200000     33600               TRNE     B,SDS
007267  260740 006104'    33610               TSX      SDP;            GO TO SHUTDOWN PROCEDURE
007270  602040 100000     33620               TRNE     B,OFFS
007271  260740 005512'    33630               TSX      BOFF;           GO TO BLAST OFF PROCEDURE
007272  332000 003162'    33640               SKIPE    CKER
007273  260740 005613'    33650               TSX      GRONK
007274  332700 002126'    33660               SKIPE    S,DCT;          SKIP IF NO DISC RESTART WAITING
007275  260740 004642'    33670               TSX      DSTRT;          TRY THE RESTART
007276  332140 006061'    33680               SKIPE    D,DISC.S;       COMPLETION SIGNAL FROM DISC?
007277  260740 005573'    33690               TSX      DISCP;          KEEP TABS ON THE DISC
007300  332000 006102'    33700               SKIPE    SKT
007301  260740 006007'    33710               TSX      MDS;            TIME FOR NEXT SKULK RECORD
007302  332000 000115'    33720               SKIPE    SG.L
007303  260740 006134'    33730               TSX      PRSIG;          PROCESS SIGNALS FROM DISTRIBUTOR
007304  260740 006737'    33740               TSX      STAT;           DO STATISTICS
007305  332700 002134'    33750               SKIPE    S,ABG
007306  260740 005543'    33760               TSX      MSGPR;          GET A BUFFER FOR AWAITING GREEN
007307  332700 002135'    33770               SKIPE    S,QP
007310  260740 005561'    33780               TSX      PQP;            PROCESS THE PAUSE QUEUE
007311  332000 002137'    33790               SKIPE    QM
007312  260740 006304'    33800               TSX      OF2;            TRY TO ENABLE USER IN THE QUEUE
007313  332000 005471'    33810               SKIPE    SS99;           SKIP IF NO COMPACT FOR IN SWAP
007314  260740 005456'    33820               TSX      SS90;           GO COMPACT
007315  332700 002136'    33830               SKIPE    S,QDM
007316  260740 007322'    33840               TSX      DM90;           TAKE CARE OF SWAPS TO INCREASE CORE
007317  336000 002107'    33850               SKIPN    DMBY
007320  260740 005211'    33860               TSX      SELSWP;         SELECT FOR SWAP
007321  254000 007335'    33870               JRST     SELINT;         GO TO SELECT FOR INTERPRET
                          33880
                          33890      ;        PROCESS THE QUEUE FOR TRANSFER TO DRUM
                          33900
007322  332000 002107'    33910      DM90:    SKIPE    DMBY
007323  263740 000000     33920               DONE
007324  275700 000141'    33930               SUBI     S,S.Q
007325  020000 000006     33940               CHS      QC.S;           PRIORITY TO COME IN
007326  135100 000264'    33950               LDB      C,S.BLOCK
007327  275100 000001     33960               SUBI     C,1;            BLOCKS TO GO OUT
007330  476000 003331'    33970               SETOM    DMIN;           INDICATE NO IN REQUESTED
007331  260740 003234'    33980               TSX      OSWAP;          START HIM OUT
007332  263740 000000     33990               DONE
                          34000
                          34010
```

```
                         34030
                         34040     ,                TABLE OF STATES WHICH ARE SEARCHED TO FIND A USER
                         34050
007333  000420 620514    34060     SI1:     BYTE    (5) TOF.S,ON.S,RC.S,RI.S,RIB.S,UC.S,QC.S,COM.S,END.S
007334  177400 000000
                         34070
007335  201440 000010    34080     SELINT:  MOVEI   J,10;            COUNT TO SAVE US FROM THE MACHINE
007336  200040 010554'   34090               MOVE    B,[POINT 5,SI1]; GET TABLE POINTER
007337  134100 000001    34100     SI2:     ILDB    C,B;             INDEX TO STATE HEADER
007340  306100 000037    34110               CAIN    C,END.S;         SKIP IF NOT END
007341  254000 007360'   34120               JRST    SI4;             END OF SEARCH
007342  375000 000011    34130               SOSGE   J
007343  040000 000030    34140     H30:     HALT    30;              CAIN OR ILDB HAS FAILED
007344  550102 002115'   34150               HRRZ    C,S.QUE(C);      LIST HEADER
007345  322100 007337'   34160               JUMPE   C,SI2;           JUMP IF NO ENTRY
007346  550700 000002    34170     SI3:     HRRZ    S,C
007347  275700 000141'   34180               SUBI    S,S.Q;           STATION INDEX
007350  135140 000263'   34190               LDB     D,S.COR;         GET CORE CELL
007351  135240 000261'   34200               LDB     F,S.STA;         GET STATE
007352  326140 007406'   34210               JUMPN   D,INTINT;        FOUND ONE; JUMP TO INITIALIZE
007353  306240 000001    34220               CAIN    F,ON.S;          IS IT ON?
007354  254000 007367'   34230               JRST    SI7;             YES, GO TO FIND CORE
007355  550102 000000    34240     SI3.5:   HRRZ    C,0(C);          NOT IN SO LOOK FURTHER
007356  326100 007346'   34250               JUMPN   C,SI3;           JUMP IF ENTRY EXISTS
007357  254000 007337'   34260               JRST    SI2;             END OF LIST - GO FOR NEXT LIST
                         34270
                         34280     ,                MONITOR IDLE LOOP
                         34290
007360  336000 002107'   34300     SI4:     SKIPN   DMBY;            CANT FIND WORK; IS DRUM BUSY?
007361  354000 002201'   34310               AOSA    CT11;            NO WORK ADD TO IDLE TIME.
                         34320
007362  336000 002124'   34330     SI4.5:   SKIPN   COM
007363  254000 007264'   34340               JRST    SIGPR;           NOT UNOVERLAPPED IF NO COMPUTE
007364  350000 002200'   34350               AOS     CT12;            UNOVERLAPPED TICKS
007365  350000 002441'   34360               AOS     T7+1
007366  254000 007264'   34370               JRST    SIGPR;           GO PROCESS SIGNALS
                         34380
                         34390
```

```
                                34410     ;        PROCESS AN "ON" USER
                                34420
007367  201140  000000          34430     SI7:     MOVEI   D,0
007370  336003  000272'         34440              SKIPN   CORE(D)
007371  254000  007376'         34450              JRST    SI8;            NOPE
007372  350000  000003          34460              AOS     D
007373  311140  000313'         34470              CAML    D,N.C;          HAVE WE SEARCHED ALL
007374  254000  007355'         34480              JRST    SI3.5;          YES, THERE ARE NONE AVAILABLE
007375  254000  007370'         34490              JRST    SI7+1;          NO, GO FOR NEXT
                                34500
007376  200003  000272'         34510     SI8:     MOVE    A,CORE(D)
007377  661000  400000          34520              TLO     A,400000;       SET "IN USE" BIT
007400  202003  000272'         34530              MOVEM   A,CORE(D)
007401  200040  000003          34540              MOVE    B,D
007402  137700  000317'         34550              DPB     S,S.UR;         SET USER # IN CORE BLOCK
007403  271140  000020          34560              ADDI    D,BBLOCK;       COMPUTE RELOCATION
007404  137140  000263'         34570              DPB     D,S.COR;        AND SET IN STATUS WORD
007405  350000  002202'         34580              AOS     CT39;           COUNT IN-CORE USERS
```

```
                                34600     ,        ON ENTRY F CONTAINS STATE, S THE SELECTED USER, AND
                                34610     ,        D HIS CORE LOCATION
                                34620
007406  275140 000020           34630     INTINT:  SUBI    D,BBLOCK
007407  200143 000272'          34640              MOVE    D,CORE(D)
007410  603140 200000           34650              TLNE    D,200000
007411  254000 007355'          34660              JRST    SI3.5;          NO GO - DRUM IS USING
007412  331000 000016           34670              SKIPL   S
007413  303700 000050           34680              CAILE   S,N.S;          S MUST BE 0 <= S <= N.S
007414  040000 000001           34690     H1:      HALT    1;              BAD USER NUMBER
007415  200305 002141'          34700              MOVE    G,T10.(F);      GET ACTION TABLE ENTRY
007416  607300 000004           34710              TLNN    G,BUFBIT;       SKIP IF HE NEEDS A BUFFER
007417  254000 007423'          34720              JRST    INT2
007420  260740 010343'          34730              TSX     GETBUF
007421  254000 007355'          34740              JRST    SI3.5;          GET HIM LATER IF NONE AVAILABLE
007422  200000 000004           34750              MOVE    A,E;            SAVE THE BUFFER POINTER
007423  260740 005110'          34760     INT2:    TSX     SRR;            SET RELOCATION REGISTER
007424  603300 000040           34770              TLNE    G,ONBIT;        SKIP IF NOT ON STATE
007425  254000 007431'          34780              JRST    INT3;           NO TEST IF TURNING ON
007426  200040 006676'          34790              MOVE    B,INITIALS
007427  312056 002572'          34800              CAME    B,MINT(S);      COMPARE INITIALS - CORE COPY W/ USER BLOCK COPY
007430  254000 007456'          34810              JRST    KILL1;          THEY DIFFER - BAD USER BLOCK DATA
007431  603300 000002           34820     INT3:    TLNE    G,INBIT;        SKIP IF NO INTERRUPT SIGNAL
007432  350000 004402'          34830              AOS     RISIG;          SET IN REQUEST
007433  135040 000266'          34840              LDB     B,S.INR
007434  436040 007432'          34850              ORM     B,RISIG;        INTERRUPT SIGNAL TO USER BLOCK
007435  402000 000001           34860              SETZM   B
007436  137040 000266'          34870              DPB     B,S.INR;        RESET INTERRUPT FLAG
007437  135040 000267'          34880              LDB     B,S.DU
007440  332000 000001           34890              SKIPE   B
007441  476000 004614'          34900              SETOM   SPARE4;         TELL JOE HE HAS THE DISC
007442  607300 000001           34910              TLNN    G,RCBIT;        SKIP IF CARRIER RETURN
007443  254000 007446'          34920              JRST    INT1
007444  550016 000331'          34930              HRRZ    A,S.BUF(S);     GET BUFFER POINTER
007445  402016 000331'          34940              SETZM   S.BUF(S);       UNLINK FROM USER,
007446  020000 000010           34950     INT1:    CHS     CU.S;           SET AS CURRENT USER AND
007447  200200 000000           34960              MOVE    E,A;            BUFFER POINTER TO COMMUNICATION REG
007450  135040 000262'          34970              LDB     B,S.TM
007451  202040 000043'          34980              MOVEM   B,T.CU;         SET CORRECT TIME IN SHOT
007452  201040 000000           34990              MOVEI   B,0
007453  603300 000020           35000              TLNE    G,CORBIT
007454  201040 000001           35010              MOVEI   B,1;            TELL JOE HE GOT MORE CORE
007455  254000 004351'          35020              JRST    MONE1;          GO EXIT FROM MONITOR
                                35030
007456  202700 000021'          35040     KILL1:   MOVEM   S,CUI;          PLANT USER # , AND
007457  202056 002642'          35050              MOVEM   B,SUM(S);       SAVE BAD INITIALS
007460  201000 000073           35060              MOVEI   A,73
007461  254000 004162'          35070              JRST    KILL;           KICK OUT
```

```
                                35090
007462  254000 007463'   35100   BJ:     JRST    .+1
007463  200140 010544'   35110           MOVE    D,[POINT 6,CT48A]
007464  134040 000003    35120           ILDB    B,D
007465  271040 000001    35130           ADDI    B,1
007466  137040 000003    35140           DPB     B,D;            COUNT IN BLOCK SIZE DISTRIBUTION
007467  201040 007550'   35150           MOVEI   B,NEXT
007470  202040 004350'   35160           MOVEM   B,INTENT;       INITIALIZE ENTRY
007471  200100 000021'   35170           MOVE    C,CUI
007472  200040 010575'   35180           MOVE    B,[ASCII /???  /]
007473  202042 002572'   35190           MOVEM   B,MINT(C);      INITIALIZE USER INITIALS
007474  202040 007426'   35200           MOVEM   B,INITIALS
007475  200040 004130'   35210           MOVE    B,BLANKS
007476  202040 004520'   35220           MOVEM   B,JOBNO
007477  402000 004311'   35230           SETZM   COMTIM
007500  402000 004527'   35240           SETZM   PAGNO
007501  402000 000000    35250           SETZM   SPARE1;         ZERO BINARY JOB NUMBER
007502  402000 000000    35260           SETZM   SPARE2;         ZERO DEPT NUMBER
007503  402000 004316'   35270           SETZM   SPARE3;         CHARGE UNITS
007504  250740 000020'   35280           EXCH    PP,PPSAV
007505  402000 000000    35290           SETZM   SEQ;            INITIALIZE SEQUENCER
007506  260740 006421'   35300           TSX     ISEC
007507  202040 000000    35310           MOVEM   B,ONTIME;       SET USERS ON TIME
007510  200700 000021'   35320           MOVE    S,CUI
007511  516056 002522'   35330           HRLZM   B,USTAT(S);     SIMULATE FIRST GREEN TIME
007512  201040 000001    35340           MOVEI   B,1;            GO FOR A BUFFER
007513  254000 007546'   35350           JRST    BJLEV
                         35360
007514  030000 010444'   35370   BJ1:    SEND    BJM1;           "JOSS HERE ... INITIALS PLEASE"
007515  350000 007505'   35380           AOS     SEQ;            INCREMENT SEQUENCER
007516  254000 007545'   35390           JRST    BJSU
                         35400
007517  260740 007605'   35410   BJ2:    TSX     BJA1;           GO TO APPROVE INITIALS
007520  254000 007524'   35420           JRST    BJ2.1;          NO GOOD
007521  030000 010456'   35430           SEND    BJM2;           GOOD - "JOB # PLEASE"
007522  350000 007515'   35440           AOS     SEQ;            INCREMENT SEQUENCER
007523  254000 007545'   35450           JRST    BJSU
                         35460
007524  030000 010463'   35470   BJ2.1:  SEND    BJM3;           "INITIALS AGAIN"
007525  254000 007545'   35480           JRST    BJSU
```

```
                                35500
007526  260740  007656'  35510   BJ3:     TSX     BJA5;          GO TO APPROVE JOB #
007527  254000  007536'  35520            JRST    BJ3.2;         NO GOOD
007530  350000  007522'  35530            AOS     SEQ
007531  350000  007530'  35540   BJ3.1:   AOS     SEQ;           GOOD - INCREMENT  SEQUENCER
007532  301000  001750   35550            CAIL    A,↑D1000;      SKIP IF NOT AN RPN
007533  254000  007567'  35560            JRST    BJ6;           YES, GO GET DEPARTMENT
007534  201040  000002   35570   BJ3.3:   MOVEI   B,2;          RETURN BUFFER
007535  254000  007546'  35580            JRST    BJLEV;         RETURN WILL BE TO INTERP. FIRST ENTRY
                         35590
007536  030000  010472'  35600   BJ3.2:   SEND    BJM4;          "JOB # AGAIN"
007537  350000  007531'  35610            AOS     SEQ
007540  254000  007545'  35620            JRST    BJSU
                         35630
007541  260740  007656'  35640   BJ4:     TSX     BJA5;          GO TO APPROVE JOB #
007542  254000  007544'  35650            JRST    .+2
007543  254000  007531'  35660            JRST    BJ3.1;         JOB # OK
007544  030000  010502'  35670            SEND    BJM5;          "ARE YOU PLAYING?"
007545  201040  000004   35680   BJSU:    MOVEI   B,4
007546  250740  000020'  35690   BJLEV:   EXCH    PP,PPSAV
007547  254000  004304'  35700            JRST    MONENT;        GO TO MONITOR
                         35710
007550  550040  007537'  35720   NEXT:    HRRZ    B,SEQ;         GET THE SEQUENCER
007551  250740  000020'  35730            EXCH    PP,PPSAV
007552  254021  007553'  35740            JRST    @BJSW(B);      DISPATCH TO PROPER SEQUENCE
                         35750
007553  000000  007514'  35760   BJSW:    XWD     0,BJ1
007554  000000  007517'  35770            XWD     0,BJ2
007555  000000  007526'  35780            XWD     0,BJ3
007556  000000  007541'  35790            XWD     0,BJ4
007557  000000  007562'  35800            XWD     0,BJ5
007560  000000  007572'  35810            XWD     0,BJ7
007561  000000  007601'  35820            XWD     0,BJ9
                         35830
```

```
007562  200700 000021'  35850   BJ5:   MOVE   S,CUI;            RECOVER STATION #
007563  200140 010576'  35860          MOVE   D,[ASCII ? ON-?]
007564  260740 006664'  35870          TSX    DOF;              DISPLAY INITIALS ON TTY
007565  250740 000020'  35880          EXCH   PP,PPSAV
007566  254000 004155'  35890          JRST   INTBEG;           GO TO BEGINNING OF INTERPRETER
                         35900
007567  350000 007550'  35910   BJ6:   AOS    SEQ
007570  030000 010520'  35920          SEND   BJM7;             "DEPARTMENT"
007571  254000 007545'  35930          JRST   BJSU
                         35940
007572  260740 007763'  35950   BJ7:   TSX    BJAD;             GO TO APPROVE DEPT
007573  254000 007576'  35960          JRST   BJ8;              NO GOOD
007574  370000 007567'  35970          SOS    SEQ;              OK EXIT
007575  254000 007534'  35980          JRST   BJ3.3
                         35990
007576  350000 007574'  36000   BJ8:   AOS    SEQ
007577  030000 010524'  36010          SEND   BJM8;             "DEPT NAME OR NUMBER"
007600  254000 007545'  36020          JRST   BJSU
                         36030
007601  260740 007763'  36040   BJ9:   TSX    BJAD;             TRY FOR APPROVAL AGAIN
007602  254000 007577'  36050          JRST   BJ8+1;            NO GOOD - KEEP AT HIM
007603  370000 007576'  36060          SOS    SEQ
007604  254000 007574'  36070          JRST   BJ7+2;            TAKE OK EXIT
```

ROUTINE TO APPROVE INITIALS

```
007605  200040 004130'  36090   BJA1:  MOVE    B,BLANKS
007606  202040 007474'  36100          MOVEM   B,INITIALS
007607  201400 000004   36110          MOVEI   I,4;            MAX OF 4 LETTERS
007610  201100 007632'  36120          MOVEI   C,SCT1;         SCAN TABLE LOCATION
007611  200040 004136'  36130          MOVE    B,C3;           OUTPUT POINTER
007612  260740 010103'  36140          TSX     SCAN;           GO TO SCAN THE BUFFER
007613  306400 000004   36150          CAIN    I,4;            NOT APPROVED IF NO INITIALS
007614  254000 007627'  36160          JRST    BJA4
007615  350017 000000   36170          AOS     0(PP);          BUMP TO EXIT FOR APPROVED
007616  200040 007606'  36180          MOVE    B,INITIALS
007617  200100 000021'  36190          MOVE    C,CUI
007620  202042 002572'  36200          MOVEM   B,MINT(C);      SAVE IN MONITOR AREA
007621  263740 000000   36210          DONE
                         36220
007622  262740 000000   36230   BJA2:  POP     PP,0;           BUMP UP ONE LEVEL TO SCAN EXIT
007623  263740 000000   36240          DONE;                   TAKE NOT APPROVED EXIT
                         36250
007624  371000 000010   36260   ALPH1: SOSL    I;              ALPHA SCAN FOR INITIALS - COUNT CHARACTERS
007625  254003 000003   36270          JRST    3(D);           ADD TO STRING
                         36280
                         36290          NOT APPROVED EXIT FOR INITIALS SCAN
                         36300
007626  262740 000000   36310   BJA3:  POP     PP,0
007627  200040 010575'  36320   BJA4:  MOVE    B,[ASCII /???  /]
007630  202040 007616'  36330          MOVEM   B,INITIALS;     RESTORE TEST CELL (FOR GOOD DRUM DATA)
007631  263740 000000   36340          DONE;                   TAKE NOT APPROVED EXIT
```

SCAN TABLE FOR APPROVAL OF INITIALS

```
                                36360
007632  255000 000000           36370     SCT1:   NOP;                      BLANK
007633  265140 007624'           36380             JSP     D,ALPH1;          ALPHA
007634  254000 007626'           36390             JRST    BJA3;             NUMERIC
007635  254000 007626'           36400             JRST    BJA3;             SPECIAL
007636  254000 007626'           36410             JRST    BJA3;             ILLEGAL
007637  255000 000000            36420             NOP;                      PERIOD
007640  254000 007626'           36430             JRST    BJA3;             TAB
007641  255000 000000            36440             NOP;                      DASH
007642  265140 007655'           36450             JSP     D,END;            END OF STRING
                                36460
007643  405100 000017           36470     DIGIT:  ANDI    C,17;             MASK TO NUMERIC
007644  326000 007647'           36480             JUMPN   A,DIG1;           JUMP IF NOT TESTING FOR LEADING ZEROS
007645  326100 007647'           36490             JUMPN   C,.+2;            IT IS NON-ZERO
007646  254003 000000            36500             JRST    0(D);             IGNORE LEADING ZEROS
007647  221000 000012            36510     DIG1:   IMULI   A,↑D10
007650  270000 000002            36520             ADD     A,C;              CONSTRUCT BINARY RPN IN REG A
007651  435100 000060            36530             ORI     C,60;             MAKE IT AN ASCII AGAIN
007652  371000 000010            36540     ALPHA:  SOSL    I;                COUNT CHARACTERS
007653  254003 000003            36550     DASH:   JRST    3(D);             ADD TO STRING
007654  254000 007622'           36560             JRST    BJA2;             MORE THAN 4 INITIALS, NOT APPROVED EXIT
007655  254003 000002            36570     END:    JRST    2(D);             EXIT FROM SCAN
```

ROUTINE TO APPROVE JOB NUMBER

```
                                36590
007656  200040 004130'   36600   BJA5:   MOVE    B,BLANKS
007657  202040 007476'   36610           MOVEM   B,JOBNO
007660  201000 000000    36620           MOVEI   A,0;            PRESET LEADING ZERO SWITCH
007661  201400 000004    36630           MOVEI   I,4;            MAX OF 4 DIGITS
007662  201100 007752'   36640           MOVEI   C,SCT2;         SCAN TABLE LOCATION
007663  200040 004137'   36650           MOVE    B,C4;           OUTPUT POINTER
007664  260740 010103'   36660           TSX     SCAN;           GO TO SCAN BUFFER
007665  202000 007501'   36670           MOVEM   A,SPARE1;       SAVE BINARY JOB NUMBER
007666  322000 007723'   36680           JUMPE   A,BJAN;         NO GOOD IF  ZERO
007667  307000 000004    36690           CAIG    A,4;            1-4 ARE OK
007670  254000 007722'   36700           JRST    BJAY;           TAKE OK EXIT
007671  201040 000011    36710           MOVEI   B,RPNS;         SIZE OF RPN RANGE TABLE
007672  554141 007740'   36720   BJA5.2: HLRZ    D,RPN-1(B);     LOW END OF OK RANGE
007673  315000 000003    36730           CAMGE   A,D
007674  254000 007700'   36740           JRST    .+4
007675  550141 007740'   36750           HRRZ    D,RPN-1(B);     HIGH END OF OK RANGE
007676  317000 000003    36760           CAMG    A,D
007677  254000 007722'   36770           JRST    BJAY;           ITS OK
007700  367040 007672'   36780           SOJG    B,BJA5.2;       GO FOR NEXT TABLE ENTRY
                         36790
                         36800   ,       SEARCH SPECIAL TABLE
                         36810
007701  201040 000011    36820           MOVEI   B,SRPNS
007702  135140 007737'   36830   BJS1:   LDB     D,PLOW
007703  322140 007711'   36840           JUMPE   D,BJS2;         IGNORE ZERO ENTRIES
007704  315000 000003    36850           CAMGE   A,D
007705  254000 007711'   36860           JRST    BJS2
007706  135140 007740'   36870           LDB     D,PHIG
007707  317000 000003    36880           CAMG    A,D
007710  254000 007713'   36890           JRST    BJS3
007711  367040 007702'   36900   BJS2:   SOJG    B,BJS1
007712  254000 007723'   36910           JRST    BJAN;           NOT IN TABLE
                         36920
                         36930   ,       HERE WE HAVE A JN IN OK RANGE
                         36940
007713  135140 007736'   36950   BJS3:   LDB     D,PSTN;         STATION NUMBER
007714  336000 000021'   36960           SKIPN   CUI
007715  254000 007720'   36970           JRST    BJS4;           ZERO IS OK
007716  312140 000021'   36980           CAME    D,CUI
007717  254000 007723'   36990           JRST    BJAN;           NOT ALLOWED ON THIS STATION
007720  135140 007735'   37000   BJS4:   LDB     D,PDEP
007721  202140 007502'   37010           MOVEM   D,SPARE2;       SET DEPT NO.
007722  350017 000000    37020   BJAY:   AOS     0(PP);          BUMP TO APPROVAL EXIT
007723  263740 000000    37030   BJAN:   DONE
                         37040
```

TABLES OF RPNS

```
                                37060     SRPN=.;          TABLE OF SPECIAL RPNS
                                37070               RADIX    10
007724  234027 555716            37080               BYTE     (8)  78,↑O5   (10)  950,974; SPARE
007725  232023 411634            37090               BYTE     (8)  77,↑O4   (10)  900,924; RAND, BETHESDA
007726  230017 101470            37100               BYTE     (8)  76,↑O3   (10)  800,824; AFCRL
007727  226012 571324            37110               BYTE     (8)  75,↑O2   (10)  700,724; TAC LANGLEY, VA.
007730  224212 261211            37120               BYTE     (8)  74,↑O42  (10)  600,649; OSD
007731  222235 751045            37130               BYTE     (8)  73,↑O47  (10)  500,549; ARPA
007732  220215 440701            37140               BYTE     (8)  72,↑O43  (10)  400,449; AFCSA
007733  216175 130617            37150               BYTE     (8)  71,↑O37  (10)  300,399; AIR ACADEMY
007734  214154 620453            37160               BYTE     (8)  70,↑O33  (10)  200,299; SACRAMENTO
                                37170               RADIX    8
                                37180     SRPNS=.-SRPN;    TABLE SIZE
                                37190
007735  341001 007723'           37200     PDEP:    POINT    8,SRPN-1(B),7
007736  241001 007723'           37210     PSTN:    POINT    8,SRPN-1(B),↑D15
007737  121201 007723'           37220     PLOW:    POINT    10,SRPN-1(B),↑D25
007740  001201 007723'           37230     PHIG:    POINT    10,SRPN-1(B),↑D35
                                37240
                                37250     ,        TABLE OF LEGITIMATE RPN RANGES
                                37260
                                37270               RADIX    10
007741  023254 023417            37280     RPN:     XWD      9900,9999
007742  022745 023253            37290               XWD      9701,9899
007743  022674 022674            37300               XWD      9660,9660
007744  021640 022516            37310               XWD      9120,9550
007745  020022 020034            37320               XWD      8210,8220
007746  015530 015732            37330               XWD      7000,7130
007747  015055 015100            37340               XWD      6701,6720
007750  013114 013147            37350               XWD      5708,5735
007751  001751 003410            37360               XWD      1001,1800
                                37370     RPNS=.-RPN;                        SIZE OF RPN TABLE
                                37380               RADIX    8
```

SCAN TABLE FOR JOB NUMBER APPROVAL

```
                          37400
007752  255000 000000     37410     SCT2:   NOP;                    BLANK
007753  254000 007622'    37420             JRST    BJA2;           ALPHA
007754  265140 007643'    37430             JSP     D,DIGIT;        NUMERIC
007755  254000 007622'    37440             JRST    BJA2;           SPECIAL
007756  254000 007622'    37450             JRST    BJA2;           ILLEGAL
007757  255000 000000     37460             NOP;                    PERIOD
007760  254000 007622'    37470             JRST    BJA2;           TAB
007761  254000 007622'    37480             JRST    BJA2;           DASH
007762  265140 007655'    37490             JSP     D,END;          END OF STRING
```

ROUTINE TO APPROVE DEPARTMENT NAME OR NUMBER

```
007763  200040 010577'  37510   BJAD:  MOVE   B,[POINT 7,F];   OUTPUT POINTER
007764  201100 010014'  37520          MOVEI  C,SCT3;          CHAR TYPE EXECUTE TABLE
007765  402000 000005   37530          SETZM  F
007766  403340 007721'  37540          SETZB  H,SPARE2
007767  403400 000011   37550          SETZB  I,J
007770  260740 010103'  37560          TSX    SCAN
007771  322400 010006'  37570          JUMPE  I,BJA9;          JUMP IF NUMERICS RECEIVED
007772  404240 010600'  37580          AND    F,[BYTE (7) 137,137,137]; FORCE UPPER CASE
007773  201040 000036   37590   BJA7:  MOVEI  B,DEPTS;         GET SIZE OF DEPARTMENT TABLE
007774  200101 010044'  37600          MOVE   C,DEPT-1(B)
007775  404100 010601'  37610          AND    C,[BYTE (7) 177,177,177];  TAKE THREE CHARACTERS
007776  316100 000005   37620          CAMN   C,F
007777  254000 010002'  37630          JRST   BJA8;            FOUND
010000  367040 007774'  37640          SOJG   B,BJA7+1;        LOOK FOR NEXT
010001  263740 000000   37650          DONE;                   NO GOOD
                        37660
010002  135040 010602'  37670   BJA8:  LDB    B,[POINT 7,DEPT-1(B),35]; GET DEPT NUMBER
010003  202040 007766'  37680          MOVEM  B,SPARE2;        SAVE IN USER BLOCK
010004  350017 000000   37690          AOS    0(PP);           BUMP TO APPROVAL EXIT
010005  263740 000000   37700          DONE
                        37710
010006  201040 000036   37720   BJA9:  MOVEI  B,DEPTS
010007  135100 010602'  37730          LDB    C,[POINT 7,DEPT-1(B),35]; GET DEPT NUMBER
010010  316100 000007   37740          CAMN   C,H
010011  254000 010002'  37750          JRST   BJA8;            HE GOT ONE!
010012  367040 010007'  37760          SOJG   B,BJA9+1;        AROUND FOR NEXT
010013  263740 000000   37770          DONE;                   NONE FOUND - NOT OK
```

```
010014  255000 000000   37790    SCT3:   NOP;                    IGNORE BLANKS
010015  265140 010025'  37800            JSP     D,BJD1;         ALPHA
010016  265140 010032'  37810            JSP     D,BJD2;         NUMERIC
010017  265140 010025'  37820            JSP     D,BJD1;         SPECIAL - TREAT AS ALPHA
010020  254000 007622'  37830            JRST    BJA2;           ILLEGAL - DENY
010021  255000 000000   37840            NOP;                    IGNORE PERIODS
010022  255000 000000   37850            NOP;                    IGNORE TABS
010023  255000 000000   37860            NOP;                    IGNORE DASH
010024  265140 007655'  37870            JSP     D,END;          END OF STRING
                        37880
010025  326440 007622'  37890    BJD1:   JUMPN   J,BJA2;         ALPHA SCAN - DENY IF PRIOR NUMERIC
010026  350000 000010   37900            AOS     I
010027  301400 000003   37910            CAIL    I,3;            NEED THREE ALPHAS
010030  254003 000001   37920            JRST    1(D);           ENOUGH SEEN ADD TO STRING AND EXIT
010031  254003 000003   37930            JRST    3(D);           ADD TO STRING
                        37940
010032  326400 007622'  37950    BJD2:   JUMPN   I,BJA2;         NUMERIC SCAN - BAD IF ALPHA
010033  405100 000017   37960            ANDI    C,17;           MASK INTO NUMBER
010034  326440 010037'  37970            JUMPN   J,BJD22;        JUMP IF WE HAVE A DIGIT
010035  336000 000002   37980            SKIPN   C
010036  254003 000000   37990            JRST    0(D);           IGNORE LEADING ZEROS
010037  221340 000012   38000    BJD22:  IMULI   H,↑D10
010040  270340 000002   38010            ADD     H,C;            COMPUTE BINARY
010041  350000 000011   38020            AOS     J
010042  301440 000002   38030            CAIL    J,2;            NEED 2 DIGITS
010043  254003 000001   38040            JRST    1(D);           ENOUGH
010044  254003 000003   38050            JRST    3(D);           ADD TO STRING
```

DEPARTMENT NAME-NUMBER TABLE

```
                                38070              RADIX    10
010045  406111 500027            38080      DEPT:   BYTE     (21) "ADM" (15) 23
010046  406132 200040            38090              BYTE     (21) "AER" (15) 32
010047  406472 400043            38100              BYTE     (21) "AST" (15) 35
010050  416371 500121            38110              BYTE     (21) "COM" (15) 81
010051  416372 300013            38120              BYTE     (21) "COS" (15) 11
010052  416470 400121            38130              BYTE     (21) "CSD" (15) 81
010053  426310 500041            38140              BYTE     (21) "ELE" (15) 33
010054  426071 700012            38150              BYTE     (21) "ECO" (15) 10
010055  434270 100043            38160              BYTE     (7) "G",↑O13,"A" (15) 35;   LOWER CASE +
010056  434370 100043            38170              BYTE     (7) "G",↑O17,"A" (15) 35;   LOWER CASE /
010057  436131 700043            38180              BYTE     (21) "GEO" (15) 35
010060  462230 200025            38190              BYTE     (21) "LIB" (15) 21
010061  462370 700014            38200              BYTE     (21) "LOG" (15) 12
010062  466032 400120            38210              BYTE     (21) "MAT" (15) 80
010063  466152 300024            38220              BYTE     (21) "MFS" (15) 20
010064  464272 200032            38230              BYTE     (7) "M",↑O13,"R" (15) 26
010065  464372 200032            38240              BYTE     (7) "M",↑O17,"R" (15) 26
010066  502132 200034            38250              BYTE     (21) "PER" (15) 28
010067  502213 100050            38260              BYTE     (21) "PHY" (15) 40
010070  512100 000074            38270              BYTE     (14) "RD" (7) 0 (15) 60
010071  512132 000074            38280              BYTE     (21) "REP" (15) 60
010072  512132 300036            38290              BYTE     (21) "RES" (15) 30
010073  516130 300033            38300              BYTE     (21) "SEC" (15) 27
010074  516370 300132            38310              BYTE     (21) "SOC" (15) 90
010075  516470 400132            38320              BYTE     (21) "SSD" (15) 90
010076  516460 000132            38330              BYTE     (14) "SS" (7) 0 (15) 90
010077  516631 700042            38340              BYTE     (21) "SYO" (15) 34
010100  516632 300042            38350              BYTE     (21) "SYS" (15) 34
010101  536032 300026            38360              BYTE     (21) "WAS" (15) 22
010102  536360 000026            38370              BYTE     (14) "WO" (7) 0 (15) 22
                                 38380      DEPTS=.-DEPT
                                 38390              RADIX    8
```

```
                        38410     ;        C IS SCAN TABLE LOCATION, B IS OUTPUT BYTE POINTER LOCATION
                        38420     ;        E LEFT IS CLOBBERED, SCAN TABLE RETURNS ARE: 0=IGNORE CHAR
                        38430     ;        1=ADD TO OUTPUT STRING AND EXIT FROM SCAN, 2=EXIT FROM SCAN
                        38440     ;        3=ADD CHARACTER TO OUTPUT STRING.
                        38450
010103  271200 000003   38460     SCAN:    ADDI     E,3;            BUMP POINTER DOWN TO TEXT
010104  505200 440700   38470              HRLI     E,440700;       INPUT BYTE POINTER
010105  202200 002151'  38480              MOVEM    E,SCP1
010106  275200 000003   38490              SUBI     E,3;            RESTORE BUFFER POINTER
010107  322040 010111'  38500     MSCAN:   JUMPE    B,.+2;          SKIP IF CONTINUING OUTPUT
010110  202040 002153'  38510              MOVEM    B,SCP3;         OUTPUT BYTE POINTER
010111  542100 002155'  38520              HRRM     C,SC9;          EXECUTE TABLE LOCATION
010112  134040 002151'  38530     SC3:     ILDB     B,SCP1;         LOAD NEXT INPUT BYTE
010113  202040 000002   38540              MOVEM    B,C;            SAVE THE BYTE
010114  405040 000007   38550              ANDI     B,7;            MASK LOW ORDER
010115  211041 777770   38560              MOVNI    B,-8(B);        COMPUTE 8-(B)
010116  240040 000002   38570              ASH      B,2;            4(8-(B))
010117  137040 002154'  38580              DPB      B,SCP4;         SET UP BYTE POINTER
010120  202100 000001   38590              MOVEM    C,B;            RECOVER INPUT BYTE
010121  240040 777775   38600              ASH      B,-3;           HIGH 4 BITS FOR INDEX
010122  135040 002152'  38610              LDB      B,SCP2;         GET THE TYPE
010123  256020 002155'  38620              XCT      @SC9;           EXECUTE THE SCAN TABLE INSTRUCTION
010124  254000 010112'  38630              JRST     SC3;            SKIP CHARACTER
010125  136100 002153'  38640              IDPB     C,SCP3;         ADD TO STRING AND EXIT
010126  263740 000000   38650              DONE;                    EXIT FROM SCAN
010127  136100 002153'  38660              IDPB     C,SCP3;         ADD CHARACTER TO OUTPUT
010130  254000 010112'  38670              JRST     SC3
```

```
                                38690
                                38700      ,       TYPES ARE:
                                38710      ,               0=BLANK
                                38720      ,               1=ALPHA
                                38730      ,               2=NUMERIC
                                38740      ,               3=SPECIAL
                                38750      ,               4=ILLEGAL
                                38760      ,               5=PERIOD
                                38770      ,               6=COMMA
                                38780      ,               7=DASH
                                38790      ,               8=END OF STRING
                                38800
                                38810      ,       THERE ARE EIGHT 4 BIT CODES PER WORD WITH 4 ZEROS AT
                                38820      ,       THE END OF EACH WORD.
                                38830
010131  410421  042100          38840      CART:   BYTE    (4) 8,4,4,4,4,4,4,4,0; COLUMN 1
010132  154316  102100          38850              BYTE    (4) 3,6,3,3,8,8,4,4,0
010133  210421  042100          38860              BYTE    (4) 4,4,4,4,4,4,4,4,0
010134  210421  042100          38870              BYTE    (4) 4,4,4,4,4,4,4,4,0
010135  006314  631460          38880              BYTE    (4) 0,3,3,3,3,3,3,3,0;      COLUMN 2
010136  146314  672460          38890              BYTE    (4) 3,3,3,3,3,7,5,3,0
010137  104210  421040          38900              BYTE    (4) 2,2,2,2,2,2,2,2,0
010140  104314  631460          38910              BYTE    (4) 2,2,3,3,3,3,3,3,0
010141  142104  210420          38920              BYTE    (4) 3,1,1,1,1,1,1,1,0;      COLUMN 3
010142  042104  210420          38930              BYTE    (4) 1,1,1,1,1,1,1,1,0
010143  042104  210420          38940              BYTE    (4) 1,1,1,1,1,1,1,1,0
010144  042114  631600          38950              BYTE    (4) 1,1,1,3,3,3,3,8,0
010145  202104  210420          38960              BYTE    (4) 4,1,1,1,1,1,1,1,0;      COLUMN 4
010146  042104  210420          38970              BYTE    (4) 1,1,1,1,1,1,1,1,0
010147  042104  210420          38980              BYTE    (4) 1,1,1,1,1,1,1,1,0
010150  042121  042100          38990              BYTE    (4) 1,1,1,4,4,4,4,4,0
```

```
010151  254000 010152'  39010     EJ:     JRST    .+1
010152  201040 010160'  39020             MOVEI   B,EJ1
010153  202040 007470'  39030             MOVEM   B,INTENT;       SET NEXT ENTRY IN CASE WE DONT GET A BUFFER
010154  200040 007500'  39040             MOVE    B,PAGNO
010155  322040 010231'  39050             JUMPE   B,EJ3;          NO RECORD IF INITIALIZATION INCOMPLETE
010156  201040 000001   39060             MOVEI   B,1;            GO TO MONITOR FOR A BUFFER
010157  254000 004304'  39070             JRST    MONENT
                        39080
010160  250740 000020'  39090     EJ1:    EXCH    PP,PPSAV
010161  260740 006421'  39100     EJ2:    TSX     ISEC
010162  202044 000002   39110             MOVEM   B,2(E);         SET OFF TIME IN ACCOUNTING RECORD
010163  315040 007507'  39120             CAMGE   B,ONTIME
010164  271040 250600   39130             ADDI    B,↑D3600*↑D24;  HE WORKS OVER THE WITCHING HOUR
010165  274040 010163'  39140             SUB     B,ONTIME;       COMPUTE SESSION TIME
010166  264000 004262'  39150             JSR     CVTL
010167  350002 002250'  39160             AOS     SESTIM(C);      INCREMENT PROPER COUNTER
010170  202044 000010   39170             MOVEM   B,8(E);         SET IN ACCOUNTING RECORD
010171  272040 002444'  39180             ADDM    B,T7+4
010172  200040 010603'  39190             MOVE    B,[XWD ↑D14,1]; COUNT AND TYPE FOR TAPE RECORD
010173  202044 000001   39200             MOVEM   B,1(E)
010174  200100 010546'  39210             MOVE    C,[POINT 7,3(E)]
                        39220             XMT 12,DATE
                            OPDEF X[12B12]
010175  000500 000023'  X DATE
                        39230             XMT 5,INITIALS
                            OPDEF X[5B12]
010176  000240 007630'  X INITIALS
010177  200040 007665'  39240             MOVE    B,SPARE1;       BINARY PROJECT NUMBER
010200  202044 000006   39250             MOVEM   B,6(E)
010201  200040 007477'  39260             MOVE    B,COMTIM
```

```
010202  202044 000007   39280          MOVEM   B,7(E);        SAVE HIS COMPUTING TIME
010203  231040 000012   39290          IDIVI   B,↑D10;        SCALE BY 10 FOR DISPLAY
010204  264000 004262'  39300          JSR     CVTL
010205  350002 002262'  39310          AOS     CCTIM(C);      ACCUMULATE IN COMPUTE TIME DISTRIBUTION
010206  200700 000021'  39320          MOVE    S,CUI
010207  402016 002572'  39330          SETZM   MINT(S);       CLEAR MONITOR INITIALS
010210  202704 000011   39340          MOVEM   S,↑D9(E);      STATION  # TO ACCOUNTING RECORD
010211  200040 000000   39350          MOVE    B,USIZE
010212  202044 000012   39360          MOVEM   B,↑D10(E)
010213  221040 000006   39370          IMULI   B,↑D6;         LOW END = 10 CELLS
010214  264000 004262'  39380          JSR     CVTL
010215  350002 002274'  39390          AOS     DSIZE(C);      COUNT IN SIZE DISTRIBUTION
010216  135040 000264'  39400          LDB     B,S.BLOCK
010217  202044 000013   39410          MOVEM   B,↑D11(E);     RECORD # OF CORE BLOCKS USED
010220  571700 777776   39420          HRREI   S,-2;          GET CONTEXT FOR TAPE I/O
010221  200040 010154'  39430          MOVE    B,PAGNO
010222  202044 000014   39440          MOVEM   B,↑D12(E);     NUMBER OF PAGES OUTPUT TO ACCT RECORD
010223  200040 010003'  39450          MOVE    B,SPARE2
010224  202044 000015   39460          MOVEM   B,↑D13(E);     DEPT # TO ACCT RECORD
010225  200040 007503'  39470          MOVE    B,SPARE3;      CHARGE UNITS
010226  202044 000016   39480          MOVEM   B,↑D14(E)
010227  260740 010305'  39490          TSX     PUTB;          OUTPUT THE RECORD
010230  250740 000020'  39500          EXCH    PP,PPSAV
010231  201040 000014   39510   EJ3:   MOVEI   B,↑D12;        TO MONITOR FOR FINAL OFF
010232  254000 004304'  39520          JRST    MONENT
```

```
                          39540     ,                CHANGES STATE OF USER IN S TO STATE IN E (AN
                          39550     ,                INDEX TO THE STATE TABLE)
                          39560     ,                CLOBBERS B, C, E, AND F.
                          39570
010233  135040 000261'    39580     CHS.:   LDB      B,S.STA;          CURRENT STATE
010234  316040 000004     39590             CAMN     B,E
010235  263740 000000     39600             DONE;                      EXIT IF OLD STATE EQUALS NEW
010236  551041 002115'    39610             HRRZI    B,S.QUE(B);       ADDRESS OF HEADER
010237  200100 000001     39620             MOVE     C,B;              SAVE POINTER TO HEADER
010240  550241 000000     39630     CHS1:   HRRZ     F,0(B);           CONTENTS OF ITEM
010241  326240 010243'    39640             JUMPN    F,.+2;            WE HAVENT GOT SOMETHING WRONG, HAVE WE?
010242  040000 000024     39650     H24:    HALT     24;               USER STATE AND QUEUE ARE DIFFERENT
010243  306256 000141'    39660             CAIN     F,S.Q(S);         DOES ITEM POINT TO DESIRED USER?
010244  254000 010247'    39670             JRST     CHS2;             YES
010245  551045 000000     39680             HRRZI    B,0(F);           NEW ADDRESS OF ITEM
010246  254000 010240'    39690             JRST     CHS1
010247  200245 000000     39700     CHS2:   MOVE     F,0(F);           GET CELL TO BE UNLINKED
010250  542241 000000     39710             HRRM     F,0(B);           TAKE OUT OF LIST
010251  602240 777777     39720             TRNE     F,-1;             DID WE JUST MOVE UP THE LAST IN THE LIST?
010252  254000 010257'    39730             JRST     CHS2.5;           NO
010253  201241 000000     39740             MOVEI    F,0(B);           ADDRESS OF LAST ON LIST
010254  506242 000000     39750             HRLM     F,0(C);           END LINK TO HEADER
010255  316240 000002     39760             CAMN     F,C;              DOES END LINK POINT TO HEADER?
010256  402002 000000     39770             SETZM    0(C);             YEA VERILY--ZERO IT OUT
010257  137200 000261'    39780     CHS2.5: DPB      E,S.STA;          UPDATE STATE
                          39790
                          39800     ,       PUT ON NEW LIST
                          39810
010260  541204 002115'    39820     CHS4:   HRRI     E,S.QUE(E)
010261  336044 000000     39830             SKIPN    B,0(E);           BRING UP LIST HEADER AND
010262  254000 010300'    39840             JRST     CHS5;             JUMP IF LIST IS EMPTY
010263  336000 002113'    39850             SKIPN    CSS;              SKIP IF LIST TOP SIGNAL IS ON
010264  254000 010271'    39860             JRST     CHS4.5;           NO--PUT ON BOTTOM OF LIST
010265  542056 000141'    39870             HRRM     B,S.Q(S);
010266  541056 000141'    39880             HRRI     B,S.Q(S);         NEW ITEM TO LIST TOP
010267  202044 000000     39890             MOVEM    B,0(E);           RESTORE HEADER
010270  263740 000000     39900             DONE
                          39910
010271  554044 000000     39920     CHS4.5: HLRZ     B,0(E);           GET ADDRESS OF END OF LIST
010272  200241 000000     39930             MOVE     F,0(B);           GET LAST ITEM ON LIST
010273  541256 000141'    39940             HRRI     F,S.Q(S);         LINK IN CURRENT USER
010274  202241 000000     39950             MOVEM    F,0(B);           PUT ON END OF LIST
010275  506244 000000     39960             HRLM     F,0(E);           END LINK TO LIST HEADER
010276  513016 000141'    39970             HLLZS    S.Q(S);           ZERO LAST IN LIST
010277  263740 000000     39980             DONE
                          39990
010300  201256 000141'    40000     CHS5:   MOVEI    F,S.Q(S);
010301  504240 000005     40010             HRL      F,F;              LINK AS TOP OF QUEUE
010302  202244 000000     40020             MOVEM    F,0(E)
010303  513016 000141'    40030             HLLZS    S.Q(S);           ZERO END ITEM
010304  263740 000000     40040             DONE
```

```
                                40060      ,        STATION # IN REGISTER S RIGHT
                                40070      ;        INPUT IS POINTER TO BUFFER IN E RIGHT, ANYTHING IN LEFT
                                40080      ,        B IS SCRATCH REGISTER.  STARTS THE TRANSMISSION IF IT IS
                                40090      ,        THF ONLY BUFFER ATTACHED.
                                40100
010305  700600 001002           40110      PUTB:    CONO    PI,1000+CHCTY;  TURN OFF THE TRAPS
010306  312700 010550'          40120               CAME    S,[DEC -2] ;    SKIP IF TAPE
010307  254000 010316'          40130               JRST    PUTB5;          NOT TAPE
                                40140               FSW B
010310  200040 007265'              MOVE B,SWITCH
010311  602040 000004           40150               TRNE    B,TPQ;          QUIET TAPE SWITCH UP?
010312  254000 010341'          40160               JRST    PUTB4;          YES - RETURN BUFFER
010313  550040 000401'          40170               HRRZ    B,N.BUF
010314  307040 000005           40180               CAIG    B,5;            FORGET TAPE OUTPUT IF FIVE OR
010315  254000 010341'          40190               JRST    PUTB4;          FEWER BUFFERS AVAILABLE
010316  554056 000331'          40200      PUTB5:   HLRZ    B,S.BUF(S) ;    POINTER TO LAST BUFFER
010317  326040 010334'          40210               JUMPN   B,PUTB1;        JUMP IF BUFFERS ATTACHED
010320  504200 000004           40220               HRL     E,E
010321  513004 000000           40230               HLLZS   0(E) ;          ZERO LINK
010322  202216 000331'          40240               MOVEM   E,S.BUF(S) ;    NEW BUFFER HEADER.
010323  325700 010332'          40250               JUMPGE  S,PUTB2;        TEST FOR JOSS STATION OUTPUT
010324  315700 010550'          40260               CAMGE   S,[DEC -2] ;    IS THE RANGE OK?
010325  040000 000025           40270      H25:     HALT    25;             ILLEGAL I/O CONTEXT
010326  256016 010332'          40280               XCT     PUTB2(S) ;      DO THE START ROUTINE
010327  254000 010337'          40290               JRST    PUTB3
010330  260740 003475'          40300               TSX     CIO;            TAPE I/O START
010331  260740 003602'          40310               TSX     BIO;            CONSOLE TTY I/O START
010332  260740 006534'          40320      PUTB2:   CALL    C27;            JOSS CONSOLE I/O START
010333  254000 010337'          40330               JRST    PUTB3
010334  202201 000000           40340      PUTB1:   MOVEM   E,0(B) ;        CHAIN NEW BUFFER TO LAST.
010335  513004 000000           40350               HLLZS   0(E) ;          ZERO CURRENT BUFFER LINK
010336  506216 000331'          40360               HRLM    E,S.BUF(S) ;    LAST BUFFER POINTER TO BUFFER HEADER.
010337  700600 002002           40370      PUTB3:   CONO    PI,2000+CHDAT+CHCTY;  TURN ON THE TRAPS
010340  263740 000000           40380               DONE
010341  260740 010401'          40390      PUTB4:   TSX     MBA1;           GIVE BACK THE BUFFER
010342  254000 010337'          40400               JRST    PUTB3
```

GET A BUFFER FROM AVAILABLE LIST

```
                        40420     ;        EXITS +2 WITH BUFFER LOCATION IN E OR
                        40430     ;              +1 IF NONE AVAILABLE.
                        40440     ;        USES ONLY REGISTER E.
                        40450
010343  700600 001002   40460     GETBUF:  CONO    PI,1000+CHDAT+CHCTY;  TURN OFF THE TRAPS
010344  371000 000401'  40470              SOSL    N.BUF;          COUNT DOWN AVAIL BUFFERS AND TEST
010345  254000 010350'  40480              JRST    GET1
010346  402000 000401'  40490              SETZM   N.BUF;          NONE AVAILABLE, EXIT 1
010347  254000 010363'  40500              JRST    GET2
010350  550200 000402'  40510     GET1:    HRRZ    E,L.BUF;        POINTER TO TOP BUFFER
010351  540204 000000   40520              HRR     E,0(E);         POINTER TO NEXT BUFFER
010352  504200 000402'  40530              HRL     E,L.BUF;        POINTER TO AVAIL BUFFER
010353  542200 000402'  40540              HRRM    E,L.BUF;        UPDATE HEADER
010354  606200 777777   40550              TRNN    E,-1;           SKIP IF NOT LAST ON LIST
010355  402000 000402'  40560              SETZM   L.BUF;          LAST, SO ZERO ENTIRE HEADER
010356  554200 000004   40570              HLRZ    E,E;            POINT TO BUFFER IN E RIGHT
010357  513004 000000   40580              HLLZS   0(E);           ZERO LINK
010360  326200 010362'  40590              JUMPN   E,.+2
010361  040000 000011   40600     H11:     HALT    11;             ZERO CANT BE A BUFFER
010362  350017 000000   40610              AOS     0(PP)
010363  700600 002002   40620     GET2:    CONO    PI,2000+CHDAT+CHCTY;  TURN ON THE TRAPS
010364  263740 000000   40630              DONE;                   TAKE EXIT 2
```

```
                        40650     ;        RETURNS BUFFER ATTACHED TO S TO AVAILABLE
                        40660     ,        CLOBBERS B,E
                        40670
010365  700600 001002   40680     MBA:     CONO    PI,1000+CHDAT+CHCTY;  TURN OFF THE TRAPS
010366  550216 000331'  40690              HRRZ    E,S.BUF(S);      GET BUFFER HEADER
010367  336000 000004   40700              SKIPN   E
010370  040000 000012   40710     H12:     HALT    12;              ZERO CANT BE A BUFFER
010371  550044 000000   40720              HRRZ    B,0(E);          GET LINK FROM ATTACHED BUFFER
010372  336000 000001   40730              SKIPN   B;               SKIP IF NOT LAST ON LIST
010373  402016 000331'  40740              SETZM   S.BUF(S);        LAST, SO ZERO HEADER
010374  542056 000331'  40750              HRRM    B,S.BUF(S);      UPDATE HEADER
010375  302200 006052'  40760              CAIE    E,DBUF;          SKIP IF SPECIAL DISC BUFFER
010376  254000 010401'  40770              JRST    MBA1
010377  350000 006102'  40780              AOS     SKT;             FLAG OUTPUT COMPLETE
010400  254000 010416'  40790              JRST    MBA5;            AND DONT PUT ON AVAILABLE
                        40800
                        40810     ;                RETURN BUFFER IN E TO AVAILABLE
                        40820
010401  700600 001002   40830     MBA1:    CONO    PI,1000+CHDAT+CHCTY;  TURN OFF THE TRAPS
010402  550040 000402'  40840              HRRZ    B,L.BUF
010403  316040 000004   40850     MBA4:    CAMN    B,E;             THIS BUFFER SHOULD NOT BE ON THE LIST ALREADY
010404  040000 000013   40860     H13:     HALT    13;              BUT IT IS!!!
010405  322040 010410'  40870              JUMPE   B,.+3;           IF WE ARE AT THE END
010406  550041 000000   40880              HRRZ    B,0(B);          GET THE NEXT ON THE LIST
010407  254000 010403'  40890              JRST    MBA4;            AND GO TO TEST IT
010410  554040 000402'  40900              HLRZ    B,L.BUF;         ADDRESS OF LAST ON AVAILABLE
010411  322040 010420'  40910              JUMPE   B,MBA3;          ARE THERE NONE? JUMP IF SO.
010412  542201 000000   40920              HRRM    E,0(B);          LINK OBJECT BUFFER TO LAST
010413  506200 000402'  40930     MBA2:    HRLM    E,L.BUF;         AND TO HEADER
010414  402004 000000   40940              SETZM   0(E);            ZERO HEADER OF OBJECT BUFFER
010415  350000 000401'  40950              AOS     N.BUF;           COUNT ONE MORE AVAILABLE
010416  700600 002002   40960     MBA5:    CONO    PI,2000+CHDAT+CHCTY;  TURN ON THE TRAPS
010417  263740 000000   40970              DONE
                        40980
010420  542200 000402'  40990     MBA3:    HRRM    E,L.BUF;         SET UP FOR ONLY BUFFER ON LIST
010421  254000 010413'  41000              JRST    MBA2
```

SEND **A MESSAGE** TO THE CONSOLE TTY

```
                                41020     ;        ROUTINE AT CALLER + 1 IS EXECUTED TO FILL BUFFER
                                41030     ;                CLOBBERS S,E,C,B.
                                41040
010422  260740 010343'          41050     CMESS:   TSX      GETBUF
010423  254000 010431'          41060              JRST     CM1
010424  571700 777777           41070              HRREI    S,-1;            CONSOLE OUTPUT CONTEXT
010425  200100 010543'          41080              MOVE     C,[POINT 7,1(E)]; POINT TO BUFFER
010426  256037 000000           41090              XCT      @0(PP);          DO ROUTINE TO FILL THE BUFFER
                                41100              XMT 3,CRLF; TERMINATE THE BUFFER
                                    OPDEF X[3B12]
010427  000140 004134'          X CRLF
010430  260740 010305'          41110              TSX      PUTB;            TRANSMIT IT
010431  350017 000000           41120     CM1:     AOS      0(PP);           BUMP TO EXIT
010432  263740 000000           41130              DONE
                                41140
```

STUFF MESSAGE IN BUFFER

```
                        41160
                        41170      ,        ON ENTRY E IS  XWD MSGPTR,BUFFERPTR
                        41180      ,        B IS CLOBBERED, E IS UNCHANGED
                        41190
010433  554040 000004   41200      BJ9.1:   HLRZ    B,E;            ADDR OF BLT CONTROL WORD IN MESSAGE
010434  200100 000004   41210               MOVE    C,E;            SAVE BUFFER POINTER
010435  200041 000000   41220               MOVE    B,0(B);         PICK UP XWD FIRST LOC, CT
010436  500200 000001   41230               HLL     E,B;            FIRST LOC TO BLT CONT WORD
010437  271200 000003   41240               ADDI    E,3;            BUMP TP TEXT LOCATION
010440  270040 000004   41250               ADD     B,E;            CT + FIRST LOC = LAST WD + 1
010441  251201 777777   41260               BLT     E,-1(B);        TRANSFER MSG TO BFR
010442  200200 000002   41270               MOVE    E,C;            RESTORE BUFFER POINTER
010443  263740 000000   41280               DONE
```

```
010444  010445'000011   41300   BJM1:   XWD     .+1,BJM1E-.
                        41310   ;       JOSS AT YOUR SERVICE. INITIALS PLEASE:
010445  066251 751646   41320           BYTE    (7) 15,112,117,123,123,40,141,164,40,171;
010446  203036 420362
010447  677536 220346   41330           BYTE    (7) 157,165,162,40,163,145,162,166,151,143;
010450  627456 664706
010451  625341 544734   41340           BYTE    (7) 145,56,15,111,156,151,164,151,141,154;
010452  647515 160730
010453  715016 066312   41350           BYTE    (7) 163,40,160,154,145,141,163,145,72,40;
010454  607474 535100
010455  000000 000000   41360   BJM1E:  BYTE    (7) 0;
010456  010457'000004   41370   BJM2:   XWD     .+1,BJM2E-.;    PROJECT NUMBER PLEASE:
010457  503455 765312   41380           BYTE    (7) 120,162,157,152,145,143,164,40,156,165
010460  617504 067352
010461  667054 571164   41390           BYTE    (7) 155,142,145,162,72
010462  200000 000000   41400   BJM2E:  BYTE    (7) 40,0
                        41410
010463  010464'000006   41420   BJM3:   XWD     .+1,BJM3E-.;    ONE TO FOUR LETTERS PLEASE:
010464  477354 520350   41430           BYTE    (7) 117,156,145,40,164,157,40,146,157,165
010465  675014 667752
010466  711015 462750   41440           BYTE    (7) 162,40,154,145,164,164,145,162,163,40
010467  723136 271500
010470  703314 560746   41450           BYTE    (7) 160,154,145,141,163
010471  625644 000000   41460   BJM3E:  BYTE    (7) 145,72,40,0
                        41470
010472  010473'000007   41480   BJM4:   XWD     .+1,BJM4E-.;    PLEASE USE A LEGITIMATE NUMBER:
010473  503314 560746   41490           BYTE    (7) 120,154,145,141,163,145,40,165,163,145
010474  625016 571712
010475  203024 066312   41500           BYTE    (7) 40,141,40,154,145,147,151,164,151,155
010476  637236 464732
010477  607514 520334   41510           BYTE    (7) 141,164,145,40,156,165,155,142,145,162
010500  727334 262744
010501  351000 000000   41520   BJM4E:  BYTE    (7) 72,40,0
                        41530
010502  010503'000007   41540   BJM5:   XWD     .+1,BJM5E-.;    IF YOU ARE LEARNING USE "1" :
010503  447144 074736   41550           BYTE    (7) 111,146,40,171,157,165,12,40,141,162
010504  724244 060744
010505  625015 462702   41560           BYTE    (7) 145,40,154,145,141,162,156,151,156,147
010506  713355 167316
010507  203536 362500   41570           BYTE    (7) 40,165,163,145,40,42,61,42,40,72
010510  211424 220164
010511  200000 000000   41580   BJM5E:  BYTE    (7) 40,0
```

LOG-ON AND SHUTDOWN MESSAGES

```
010512  010513'000005   41600   SDM:    XWD     .+1,BJM6E-.;    SYSTEM SHUTTING DOWN
010513  066477 171750   41610           BYTE    (7)  15,123,171,163,164,145,155,40,163,150
010514  627324 071720
010515  727516 464734   41620           BYTE    (7)  165,164,164,151,156,147,40,144,157,167
010516  635014 467756
010517  671341 500000   41630   BJM6E:  BYTE    (7)  156,56,15,0
010520  010521'000003   41640   BJM7:   XWD     .+1,BJM7E-.;    DEPARTMENT:
010521  423136 060744   41650           BYTE    (7)  104,145,160,141,162,164,155,145,156,164
010522  723334 567350
010523  351000 000000   41660   BJM7E:  BYTE    (7)  72,40
010524  010525'000006   41670   BJM8:   XWD     .+1,BJM8E-.;    DEPARTMENT NAME OR NUMBER:
010525  423136 060744   41680           BYTE    (7)  104,145,160,141,162,164,155,145,156,164
010526  723334 567350
010527  203354 166712   41690           BYTE    (7)  40,156,141,155,145,40,157,162,40,156
010530  203376 220334
010531  727334 262744   41700           BYTE    (7)  165,155,142,145,162
010532  351000 000000   41710   BJM8E:  BYTE    (7)  72,40,0
```

LITERAL TABLE

```
010533  720100 003527'   41750              END
010534  264000 002745'
010535  440700 000000
010536  440700 004563'
010537  360300 000040
010540  270700 000040
010541  000026'000031
010542  777760 000120'
010543  440704 000001
010544  440600 002216'
010545  476150 626400
010546  440704 000003
010547  000011 000011
010550  777777 777776
010551  000272'000272'
010552  000020 000020
010553  100000 700000
010554  440500 007333'
010555  440500 005325'
010556  260740 005313'
010557  440500 005324'
010560  440500 005322'
010561  260740 005365'
010562  000000 003305'
010563  440500 005542'
010564  440716 002572'
010565  350700 006377'
010566  440700 006373'
010567  777730 000000
010570  360507 002745'
010571  221407 002745'
010572  440500 007236'
010573  440700 000023'
010574  440700 000025'
010575  375767 720100
010576  202371 626400
010577  440700 000005
010600  576773 700000
010601  777777 700000
010602  000701 010044'
010603  000016 000001
```

THERE ARE NO ERRORS

PROGRAM BREAK IS 010604

SYMBOL TABLE

A	000000	
ABG	002134'	
ABG.S	000017	
ACTION	006025'	EXT
ADATE	007212'	INT
ADIS	004650'	
ALPH1	007624'	
ALPHA	007652'	
APRR	003766'	INT
ATIM	000025'	
B	000001	
BBLOCK	000020	
BEEPS	400000	
BELF	004135'	
BIO	003602'	
BJ	007462'	
BJ1	007514'	
BJ2	007517'	
BJ2.1	007524'	
BJ3	007526'	
BJ3.1	007531'	
BJ3.2	007536'	
BJ3.3	007534'	
BJ4	007541'	
BJ5	007562'	
BJ6	007567'	
BJ7	007572'	
BJ8	007576'	
BJ9	007601'	
BJ9.1	010433'	
BJA1	007605'	
BJA2	007622'	
BJA3	007626'	
BJA4	007627'	
BJA5	007656'	
BJA5.2	007672'	
BJA7	007773'	
BJA8	010002'	
BJA9	010006'	
BJAD	007763'	
BJAN	007723'	
BJAY	007722'	
BJD1	010025'	
BJD2	010032'	
BJD22	010037'	
BJLEV	007546'	
BJM1	010444'	
BJM1E	010455'	
BJM2	010456'	
BJM2E	010462'	
BJM3	010463'	
BJM3E	010471'	
BJM4	010472'	
BJM4E	010501'	
BJM5	010502'	

SYMBOL TABLE

BJM5E	010511'	
BJM6E	010517'	
BJM7	010520'	
BJM7E	010523'	
BJM8	010524'	
BJM8E	010532'	
BJS1	007702'	
BJS2	007711'	
BJS3	007713'	
BJS4	007720'	
BJSU	007545'	
BJSW	007553'	
BLANKS	004130'	
BLKSWR	003332'	
BLTC	005073'	
BOF1	005521'	
BOF10	005542'	
BOF2	005525'	
BOF3	005533'	
BOF5	005536'	
BOFF	005512'	
BTA	004715'	
BTA1	004722'	
BU2	004437'	
BUF	000403'	
BUF1	004442'	
BUFBIT	000004	
BUFFER	004430'	
BUFSIZ	000022	
C	000002	
C20	000000	EXT
C27	010332'	EXT
C28	006602'	EXT
C3	004136'	
C30	004024'	EXT
C31	006266'	EXT
C32	004072'	EXT
C4	004137'	
C5	004140'	
CAPR	000007	
CART	010131'	
CCTIM	002262'	
CCTY	000006	
CDAT	000006	
CDC	000001	
CDRM	000002	
CH630	000030	
CHAPR	000001	
CHCTY	000002	
CHDAT	000000	
CHDC	000100	
CHDRM	000040	
CHECK	004141'	
CHECK1	004151'	
CHECK2	004176'	

SYMBOL TABLE

CHK1	004174'	
CHK2	004206'	
CHS.	010233'	
CHS1	010240'	
CHS2	010247'	
CHS2.5	010257'	
CHS4	010260'	
CHS4.5	010271'	
CHS5	010300'	
CI630	000003	
CIO	003475'	
CIO1	003524'	
CK	002130'	
CK.S	000013	
CKER	003162'	
CKF	002114'	
CKS10	003232'	
CKS11	003233'	
CKS5	003225'	
CKS7	003231'	
CKSUM	003207'	
CL1	004055'	
CL10	000036'	
CL11	000037'	
CL12	000040'	
CL13	000041'	
CL14	004075'	
CL15	004104'	
CL16	004120'	
CL17	004125'	
CL18	004014'	
CL2	004057'	
CL3	004051'	
CL4	004025'	
CL5	004015'	
CL6	004063'	
CL7	004065'	
CL8	004010'	
CL9	000035'	
CLOCK	003771'	
CM1	010431'	
CMESS	010422'	INT
CO.BUF	000330'	
CO630	000004	
COM	002124'	
COM.S	000007	
COMEBA	002111'	INT
COMP1	005133'	
COMP2	005140'	
COMPAC	005122'	
COMTIM	010201'	EXT
CONSOL	005546'	EXT
CONT	004355'	
CONT1	004361'	
CORBIT	000020	

SYMBOL TABLE

CORE	000272'	INT
CPIL	002332'	
CPOL	002344'	
CR.R	006457'	
CRLF	004134'	
CRS	004133'	
CSS	002113'	
CT10	002156'	
CT10A	002157'	
CT11	002201'	INT
CT12	002200'	
CT13	002165'	INT
CT14	002166'	INT
CT15	002167'	INT
CT16	002171'	INT
CT17	002210'	INT
CT18	002204'	INT
CT18A	002205'	INT
CT18B	002206'	INT
CT19	002207'	INT
CT19A	002203'	
CT2	002162'	
CT21	002224'	
CT22	002226'	INT
CT22A	002225'	INT
CT23	002211'	INT
CT24	002227'	INT
CT25	002174'	
CT26	002175'	
CT27	002170'	
CT27A	002176'	INT
CT27B	002177'	INT
CT28	002164'	INT
CT2A	002174'	
CT31	002173'	INT
CT32	002160'	
CT33	002161'	
CT34	002162'	
CT35	002163'	
CT37	002172'	INT
CT39	002202'	
CT3A	002214'	
CT41	002221'	
CT42	002222'	
CT43	002212'	
CT44	002213'	
CT45	002223'	
CT46	002214'	INT
CT47	002215'	INT
CT48A	002216'	
CT48B	002217'	
CT48C	002220'	
CT49	000320'	
CT50	002230'	
CT7	002221'	

SYMBOL TABLE

CT7A	002230'	
CTD	006403'	INT
CTD1	006414'	
CTD2	006416'	
CTDB	006405'	INT
CTDR	006410'	INT
CTPAR	002712'	
CTPARS	000033	
CTYR	003544'	INT
CU	002125'	
CU.S	000010	
CUI	000021'	
CVT1	006511'	
CVTL	004262'	
CVTL1	004267'	
CVTLIN	006507'	
CVTLOG	004251'	
CVTS	000011	
D	000003	
D.TIME	003772'	EXT
DAM	000001	
DASH	007653'	
DATE	000023'	INT
DATIME	000000	EXT
DAY	000030'	INT
DBASE	000321'	
DBUF	010375'	EXT
DCLOB	040000	
DCLOBM	004577'	
DCOMP	004625'	
DCONT	004616'	
DCONT1	004622'	
DCP1	004635'	
DCPI	002320'	
DCT	002126'	
DCT.S	000011	
DDT	000000	EXT
DE10	003154'	
DECT	003163'	
DEDR	003164'	
DEDR1	003165'	
DEER	003126'	
DELTA	000004	
DEPT	010045'	
DEPTS	000036	
DERR	003133'	
DI1	006646'	
DI2	006653'	
DI3	006661'	
DI4	006663'	
DIG1	007647'	
DIGIT	007643'	
DIP	002127'	
DIP.S	000012	
DISC	000000	EXT

SYMBOL TABLE

DISC.C	006026'	EXT
DISC.D	004667'	EXT
DISC.S	007276'	EXT
DISC1	005610'	
DISC2	005600'	
DISCP	005573'	
DISINT	006645'	
DISMIS	002745'	
DKBIT	000010	
DKBY	002110'	
DM10	003113'	
DM90	007322'	
DMBY	002107'	
DMERR	003137'	
DMIBK	003324'	
DMIN	003331'	
DMNR	003305'	
DMT	000020	
DMUSR	003327'	
DMWR	003330'	
DOAF	000010	
DOF	006664'	
DOF1	006673'	
DP	000010	
DPWD	003325'	
DPWD1	003326'	
DQ	002131'	
DQ.S	000014	
DR	000400	
DR2	003014'	
DREQ	004604'	
DREQ1	004613'	
DRM10	003104'	
DRM11	003105'	
DRM12	003106'	
DRM13	003107'	
DRM14	003110'	
DRM2	003037'	
DRM3	002762'	
DRM4.5	003031'	
DRM5	003032'	
DRM6	003033'	
DRM6.5	003035'	
DRM7	003043'	
DRM7.5	003061'	
DRM7.6	003077'	
DRM7.7	003100'	
DRM8	003102'	
DRM9	003103'	
DRMPP	003111'	
DRMR	002750'	INT
DSIZE	002274'	
DSS	006101'	
DSTAT	000020	
DSTRT	004642'	

SYMBOL TABLE

DSU	002133'	
DSU.S	000016	
DT.BUF	000327'	
DTIM	002306'	
DUI	000002	
DUMP	000000	EXT
E	000004	
EJ	010151'	
EJ1	010160'	
EJ2	010161'	
EJ3	010231'	
END	007655'	
END.S	000037	
ENTSW	004327'	
F	000005	
F1	005476'	
F2	005500'	
F3	005507'	
FAIL	003166'	
FAKE	006171'	EXT
FILE	004656'	EXT
FIND	005472'	
FINDB	005032'	
FM	005152'	
FM1	005153'	
FM2	005161'	
FM3	005163'	
FM4	005173'	
FM5	005174'	
FM7	005203'	
FRMT	007240'	
G	000006	
GET1	010350'	
GET2	010363'	
GETBUF	010343'	
GETCT	006612'	
GKM	004247'	
GR	002132'	
GR.S	000015	
GR1	005614'	
GR2	005616'	
GR3	005621'	
GRONK	005613'	
GT1	006630'	
GTI	005624'	
H	000007	
H1	007414'	
H10	005603'	
H11	010361'	
H12	010370'	
H13	010404'	
H14	003235'	
H15	002755'	
H17	002777'	
H2	004177'	

SYMBOL TABLE

H20	006224'	
H21	006056'	
H22	005453'	
H23	004737'	
H24	010242'	
H25	010325'	
H26	004215'	
H27	004145'	
H3	003713'	
H30	007343'	
H31	004156'	
H33	004124'	
H5	004325'	
H6	006142'	
H7	006520'	
HALTS	020000	
HDM	004535'	
HEAD	002231'	
HMES	007100'	INT
HMES1	007103'	
HR	000031'	INT
HSDM	004571'	
I	000010	
IINT	002356'	
IN10	006427'	
IN11	006432'	
IN20	006434'	
IN30	006436'	
IN40	006441'	
IN50	006454'	
IN99	006452'	
INBIT	000002	
INITIA	010176'	EXT
INSR	003727'	
INT1	007446'	
INT2	007423'	
INT3	007431'	
INTBEG	007566'	EXT
INTENT	010153'	EXT
INTINT	007406'	
IRPTR	004127'	
ISEC	006421'	
ISWAP	003333'	
J	000011	
JOBNO	007657'	EXT
K	000012	
K1	000074	
K2	000054	
K3	000017	
KEY	004660'	EXT
KILL	004162'	INT
KILL1	007456'	
L	000013	
L.BUF	000402'	
L.ETSW	000021	

SYMBOL TABLE

L.OPM	004562'	
LASTB	002064'	
M	000014	
M1	005010'	
M2	005100'	
M25	005111'	
M4	005116'	
M5	005135'	
M51	005143'	
M52	005157'	
MBA	010365'	
MBA1	010401'	
MBA2	010413'	
MBA3	010420'	
MBA4	010403'	
MBA5	010416'	
MDS	006007'	
MDS2	006030'	
MIN	000032'	INT
MINT	002572'	
MISC	002112'	
MONE1	004351'	
MONENT	004304'	INT
MONEXI	004350'	
MONS	007236'	
MONTH	000027'	INT
MOR1	005050'	
MOR2	005033'	
MOR2.5	005034'	
MOR3	005041'	
MOR3.5	005046'	
MOR4	005052'	
MOR41	005056'	
MOR5	005064'	
MOR51	005070'	
MOR6	005020'	
MOR7	005025'	
MORC1	004764'	
MORCI	005013'	
MORCOR	004760'	
MSCAN	010107'	
MSGPR	005543'	
MSGPR1	005555'	
MT1	000224	
MT2	000230	
MTC	000220	
N	000015	
N.BUF	000401'	
N.C	000313'	
N.CB	000312'	INT
N.CK	000323'	
N.COR	000020	
N.DRM	000320'	
N.PP1	000020	
N.S	000050	INT

SYMBOL TABLE

```
N.SG              000050
N.SON             000324' INT
N.UC              000322'
NEXT              007550'
OCT               006700'
OCT1              006703'
OCTW              004274' INT
OCTW1             004276'
ODIS              006711'
OF                002140'
OF.S              000023
OF1               006302'
OF2               006304'
OFF               006276'
OFF5              006321'
OFFD              006323'
OFFNQ             006274'
OFFQ              006261'
OFFS              100000
OGKM              004211'
OINT              002370'
OLSW              000100
ON                002116'
ON.R              006326'
ON.S              000001
ON5               006337'
ONBIT             000040
ONTIME            010165' EXT
OPMSG             004563'
OSWAP             003234'
PAGE              004501'
PAGNO             010221' EXT
PATCH             000000'
PAUSE             004416'
PDEP              007735'
PG1               004557'
PG2               004550'
PG3               004544'
PG4               004551'
PHIG              007740'
PLOW              007737'
PP                000017
PP1               000121'
PPSAV             000020'
PPW               000120' INT
PQP               005561'
PQP2              005571'
PR1               003706'
PRMES             000040
PROG              004672' EXT
PROP              003673' INT
PRSIG             006134'
PSTN              007736'
PUTB              010305'
PUTB1             010334'
```

SYMBOL TABLE

PUTB2	010332'	
PUTB3	010337'	
PUTB4	010341'	
PUTB5	010316'	
Q2	004465'	
QC	002123'	
QC.S	000006	
QCTR	006604'	
QDM	002136'	
QDM.S	000021	
QM	002137'	
QM.S	000022	
QM1	006371'	
QM2	006377'	
QM3	006403'	
QM4	006373'	
QP	002135'	
QP.S	000020	
QUIT	004446'	
R2	003732'	
R3	003735'	
R4	003740'	
R4.1	003750'	
R4.99	003747'	
RC	002117'	
RC.S	000002	
RCBIT	000001	
RCOR	004726'	
REBUF	004444'	
RESK	006013'	
RESULT	006033'	EXT
RI	002120'	
RI.S	000003	
RIB	002121'	
RIB.S	000004	
RISIG	007434'	EXT
RJD	003765'	EXT
RPN	007741'	
RPNS	000011	
RSIG	005636'	INT
RUL	005050'	
RUL1	005052'	
RUL2	005067'	
S	000016	
S.BLOC	000264'	
S.BUF	000331'	INT
S.COR	000263'	
S.DU	000267'	
S.GK	000270'	
S.ID	000315'	
S.INR	000266'	
S.IU	000314'	
S.M	000002	
S.OFR	000265'	
S.OK	000325'	INT

SYMBOL TABLE

S.Q	000141'	
S.QUE	002115'	INT
S.S	000211'	INT
S.SIG	000271'	
S.STA	000261'	
S.TM	000262'	
S.UD	000316'	
S.UR	000317'	
S10	005322'	
S11	005324'	
S12	005325'	
SA	000112	
SA1	006250'	
SAQM	006246'	
SC3	010112'	
SC9	002155'	
SCAN	010103'	
SCP1	002151'	
SCP2	002152'	
SCP3	002153'	
SCP4	002154'	
SCT1	007632'	
SCT2	007752'	
SCT3	010014'	
SDM	010512'	
SDP	006104'	
SDS	200000	
SEC	000033'	INT
SECOND	000044'	INT
SELINT	007335'	
SELSWP	005211'	
SEQ	007603'	EXT
SESTIM	002250'	
SG	006560'	
SG.L	000115'	INT
SG.LIM	000117'	INT
SG.M	000116'	INT
SHUT	006126'	EXT
SI1	007333'	
SI2	007337'	
SI3	007346'	
SI3.5	007355'	
SI4	007360'	
SI4.5	007362'	
SI7	007367'	
SI8	007376'	
SIGPR	007264'	INT
SIGPR1	007264'	
SIGTBL	000045'	INT
SK1	006055'	
SK10	006075'	
SK11	006077'	
SK2	006070'	
SK3	006041'	
SK4	006061'	

SYMBOL TABLE

SK5	006072'	
SKR	006103'	
SKT	006102'	
SKULK	006032'	
SOUT	007045'	INT
SP1	006135'	
SP2	006154'	
SP3	006146'	
SPARE1	010177'	EXT
SPARE2	010223'	EXT
SPARE3	010225'	EXT
SPARE4	007441'	EXT
SQM	006342'	
SRPN	007724'	
SRPNS	000011	
SRR	005110'	
SS1	005216'	
SS10	005262'	
SS11	005274'	
SS15	005277'	
SS17	005305'	
SS18	005313'	
SS2	005224'	
SS3	005235'	
SS30	005327'	
SS32	005335'	
SS33	005351'	
SS34	005353'	
SS35	005365'	
SS4	005240'	
SS40	005376'	
SS41	005410'	
SS42	005417'	
SS44	005437'	
SS45	005442'	
SS5	005242'	
SS6	005246'	
SS90	005456'	
SS98	005470'	
SS99	005471'	
SSIG	005650'	INT
ST.1	007063'	
ST.12	007077'	
ST1	007114'	
ST1A	007136'	
ST1B	007115'	
ST2	007141'	
ST2.4	007157'	
ST2.5	007131'	
ST2.6	007135'	
ST22	003574'	
ST23	003575'	
ST24	003576'	
ST3	007243'	
ST31	003577'	

SYMBOL TABLE

ST32	003600'	
ST33	003557'	
ST34	003555'	
ST35	003601'	
ST4	007246'	
ST40	003572'	
ST41	003612'	
ST42	003627'	
ST43	003643'	
ST44	003652'	
ST45	003660'	
ST46	003664'	
ST47	003665'	
ST48	003666'	
ST49	003667'	
ST5	007254'	
ST50	003670'	
ST51	003671'	
ST52	003672'	
ST6	007256'	
ST7	007263'	
STA1	006775'	
STA1.5	007025'	
STA2	007043'	
STA3	006762'	
STAT	006737'	
STAT1	006716'	
STAT2	000020	
SU	004372'	
SU.5	004401'	
SU1	004404'	
SU2	004406'	
SU3	004414'	
SUM	002642'	
SWITCH	010310'	EXT
T.CU	000043'	
T.MAX	000326'	
T1	006200'	
T10	002476'	INT
T10.	002141'	
T11	002510'	INT
T11.1	002511'	INT
T11.2	002512'	INT
T11.3	002513'	INT
T11.4	002514'	INT
T11.5	002515'	INT
T11.6	002516'	INT
T11.7	002517'	INT
T11.8	002520'	INT
T11.9	002521'	INT
T2	006173'	
T3	006224'	
T5	002414'	
T6	002426'	
T7	002440'	

SYMBOL TABLE

T7.9	002451'	INT
T8	002452'	INT
T80	007244'	EXT
T80.99	007250'	EXT
T9	002464'	INT
TABO	007057'	
TABO1	007062'	
TAPES	003526'	
TCW	003527'	
TE1	005446'	
TE2	005454'	
TE20	003432'	
TE21	003434'	
TE22	003436'	
TEND	003410'	
TEND1	003424'	
TEND2	003425'	
TEND4	003444'	
TEND5	003445'	
TEND6	003446'	
TEND7	003447'	
TEND8	003450'	
TEND9	003451'	
TEND93	003452'	
TEND95	003461'	
TERASE	002222'	INT
TERM	006130'	
TERROR	002221'	
TEST	005445'	
TICK	000034'	
TIME	000042'	INT
TINT	002402'	
TIP	003405'	
TL	004366'	
TLSU	004370'	
TLT	004704'	
TLT1	004706'	
TO	006517'	
TO1	006536'	
TO3	006553'	
TO99	006555'	
TOCK	006542'	
TOCK1	006552'	
TODSU	006537'	
TOF	002115'	
TOF.S	000000	
TPDL	003530'	
TPQ	000004	
TRST	003532'	
TRST1	003541'	
TSIG	005643'	INT
TTT	005667'	
TWAIT	003470'	
TYPE6	000022'	INT
U	000001	

SYMBOL TABLE

```
UC                  002122'
UC.S                000005
UD2                 007176'
UDATE               007162'
UP1                 007212'
USERS               002160'  INT
USIZE               010211'  EXT
USTAT               002522'
V                   000002
W                   000003
WAIT                004420'
WJD                 000000   EXT
X                   000004
XMTR                003716'
Y                   000005
YEAR                000026'  INT
Z                   000006
ZERO                004132'
ZZ                  005050'
```

END OF ASSEMBLY

EN 5031
165

PART II

PREAMBLE AND RECOVERY

```
00040     ;       JOSS INITIALIZATION AND RECOVERY PROGRAMS
00050
00060
00070     ;            G. E. BRYAN
00080
00090
00100     ;       ASSEMBLED 8/1/67 FROM TAPE 56  (A SEQUENCED COPY OF 31)
00110     ;          BINARIES ON TAPE 24 AS IU
```

```
;       INITIALIZATION DIALOGUE FOR JOSS STARTUP.

,       THIS CODE MUST BE LOADED LAST, DIRECTLY
;       PRECEDED BY DDT; A (THE ARITHMETIC) MUST
,       BE LOADED FIRST TO PRESERVE A DDT ENTRY AT 140.

;       ASSEMBLY PARAMETERS FOR THE PI CHANNELS

CDC=1;  1 - DATA CONTROL
CDRM=2; 2 - I/O PROCESSOR AND DRUM CONTROL
CI630=3;        3 - 630 INPUT
CO630=4;        4 - 630 OUTPUT
CDAT=6;         INTERRUPT CHANNEL FOR TAPE
CCTY=6; 6 - CONSOLE TELETYPE
CAPR=7; 7 - PROCESSOR

CHDC=100;       THESE ARE CORRESPONDING DEFINITIONS FOR PI CONOS
CHDRM=40;
CH630=30
CHDAT=0
CHCTY=2
CHAPR=1
MTC=220;                        DEVICE CODES FOR MAG TAPE REGISTERS
MT1=224
MT2=230
DR=400
DP=010
DMT=20;         DATA CONTROL DEVICE NUMBER FOR MAG TAPE
DDK=50;         AND FOR DISC

A=0
B=1
C=2
D=3
E=4
F=5
G=6
H=7
I=10
J=11
K=12;           USED BY DISTRIBUTOR
L=13;           USED BY DISTRIBUTOR
M=14;           USED BY DISTRIBUTOR
N=15;           USED BY DISTRIBUTOR
S=16;           USED FOR STATION INDEX
PP=17;          PUSH-POP REGISTER

        DEFINE XMT (M,A); TRANSMIT M CHARACTERS FROM A
<OPDEF X[M'B12]
X A>

        OPDEF   TSX[PUSHJ PP,0]
        OPDEF   DONE[POPJ PP,0]
```

```
                                 00670              EXTERN DDT,MONTH,YEAR,HR,MIN,ADATE,DATE
                                 00680              EXTERN  CMESS,PROP,DRMR,C25,C26,CTYR,APRR
                                 00690              EXTERN  OCTW
                                 00700              EXTERN  CORE,C20,PPW,TYPE6,SOUT,HMES,SIGPR
                                 00710
                                 00720              INTERN  DUMP,RJD,WJD,DATIME
                                 00730              INTERN  SWITCH,FAKE,L100,BEGIN
                                 00740
                                 00750
                                 00760
000000  000000 000000           00770      DISMIS: Z
000001  700600 001100           00780              CONO    PI,1000+CHDC;    TURN OFF INTERRUPT CHANNEL
000002  254520 000000'          00790              JRST    12,@DISMIS;      DISMISS INTERRUPT
```

```
                                00810     SAVE=120;        SAVE LOCATION FOR THE ACS
                                00820     IOA=777700
000003  700600 000200           00830     DUMP:   CONO     PI,200;          TURN ON THE INTERRUPT SYSTEM
000004  250040 000564'          00840             EXCH    B,[XWD 0,SAVE]
000005  251040 000137           00850             BLT     B,SAVE+17;       SAVE THE ACCUMULATORS
000006  250040 000564'          00860             EXCH    B,[XWD 0,SAVE]
000007  202040 000121           00870             MOVEM   B,SAVE+1
000010  722200 000000           00880             CONO    MTC,0
000011  722600 000002           00890             CONO    MT1,2;           READY UNIT ZERO
000012  200040 000565'          00900             MOVE    B,[JSR DISMIS]
000013  202040 000043           00910             MOVEM   B,41+2*CDC
000014  200040 000566'          00920             MOVE    B,[BLKO DC,S]
000015  202040 000042           00930             MOVEM   B,40+2*CDC
000016  201200 000000           00940             MOVEI   E,0;             LOCATION COUNTER
000017  201240 000020           00950             MOVEI   F,20;            A BLANK
000020  200140 000567'          00960             MOVE    D,[POINT 3,SAVE]
000021  254000 000024'          00970             JRST    TD1;             FIRST THE ACCUMULATORS
000022  306200 000020           00980     TD05:   CAIN    E,20;            AND THEN CORE
000023  200140 000570'          00990             MOVE    D,[POINT 3,20]; INPUT POINTER
000024  200300 000571'          01000     TD1:    MOVE    G,[POINT 6,IOA]; OUTPUT POINTER
000025  201040 000000           01010             MOVEI   B,0
000026  136240 000006           01020             IDPB    F,G;             CARRIAGE CONTROL
000027  200100 000572'          01030             MOVE    C,[POINT 3,E,↑D20]; POINT TO LOCATION COUNTER
000030  134000 000002           01040     TD2:    ILDB    A,C
000031  136000 000006           01050             IDPB    A,G;             ADDRESS
000032  305040 000004           01060             CAIGE   B,4
000033  344040 000030'          01070             AOJA    B,TD2
000034  201040 000010           01080             MOVEI   B,10;            EIGHT WORDS PER LINE
000035  136240 000006           01090     TD3:    IDPB    F,G
000036  136240 000006           01100             IDPB    F,G;             TWO BLANKS BETWEEN WORDS
000037  201340 000014           01110     TD4:    MOVEI   H,14;            TWELVE CHARACTERS PER WORD
000040  134000 000003           01120             ILDB    A,D
000041  136000 000006           01130             IDPB    A,G;             CONVERT TO BCD
000042  367340 000040'          01140             SOJG    H,.-2
000043  367040 000035'          01150             SOJG    B,TD3;           JUMP IF LINE NOT FULL
000044  136240 000006           01160             IDPB    F,G
000045  136240 000006           01170             IDPB    F,G
000046  722740 040000           01180             CONSO   MT1,40000;       WAIT FOR XFER NEW COMMAND
000047  254000 000046'          01190             JRST    .-1;             WAIT FOR TAPE CONTROL FREE
000050  200700 000573'          01200             MOVE    S,[XWD -↑D20,IOA-1]; DATA CONTROL WORD
000051  720200 003421           01210             CONO    DC,CDC+DMT+3400
000052  722200 031000           01220             CONO    MTC,31000
000053  700600 002100           01230             CONO    PI,2000+CHDC
000054  700700 000100           01240             CONSZ   PI,CHDC;         WAIT FOR DATA CONTROL FINISHED
000055  254000 000054'          01250             JRST    .-1;             WAIT FOR END OF RECORD
000056  271200 000010           01260             ADDI    E,↑D8
000057  305200 100000           01270             CAIGE   E,100000;        AINT NONE UP THERE
000060  254000 000022'          01280             JRST    TD05
000061  722740 040000           01290             CONSO   MT1,40000
000062  254000 000061'          01300             JRST    .-1
000063  722200 001400           01310             CONO    MTC,1400;        END FILE
000064  722740 040000           01320             CONSO   MT1,40000;       WAIT FOR COMMAND OK
000065  254000 000064'          01330             JRST    .-1
000066  722200 000400           01340             CONO    MTC,400;         REWIND
000067  254000 000000           01350             JRST    DDT;             GO TO DDT
```

ROUTINE TO INPUT DATE AND TIME

```
000070  712200 003600   01370   DATIME: CONO    TTY,3600;        ALL TTY FLAGS OFF
000071  260740 000423'  01380           TSX     SLM;             SET LOW MEMORY
000072  260740 000000   01390           TSX     ADATE;           CONVERT TO ASCII
000073  260740 000172'  01400           TSX     D50;             ASK ABOUT DDT
000074  260740 000254'  01410           TSX     DTO;             OUTPUT CURRENT DATE AND TIME
                        01420
000075  403140 000007   01430   D1:     SETZB   D,H;             SLASH COUNT AND CONVERTER
000076  260740 000274'  01440           TSX     LM
000077  201040 000006   01450           MOVEI   B,6
000100  200100 000574'  01460           MOVE    C,[POINT 7,S1]
000101  265300 000231'  01470           JSP     G,TS;            SEND "DATE: "
000102  265300 000262'  01480   D2:     JSP     G,TTCH;          GET TTY CHARACTER
000103  306240 000057   01490           CAIN    F,"/"
000104  254000 000121'  01500           JRST    D10;             ITS A SLASH
000105  306240 000015   01510           CAIN    F,15
000106  254000 000131'  01520           JRST    D12;             ITS A CR
000107  405240 000160   01530           ANDI    F,160
000110  302240 000060   01540           CAIE    F,60
000111  254000 000075'  01550           JRST    D1;              NON-NUMERIC -- TRY AGAIN
000112  712040 000005   01560           DATAI   TTY,F;           RECOVER CHARACTER
000113  405240 000017   01570           ANDI    F,17;            MASK OFF ASCII BITS
000114  221140 000012   01580           IMULI   D,↑D10
000115  270140 000005   01590           ADD     D,F
000116  301140 000144   01600           CAIL    D,↑D100;
000117  254000 000075'  01610           JRST    D1;              DATE COMPONENT MUST BE < 100
000120  254000 000102'  01620           JRST    D2;              GET ANOTHER
                        01630
000121  301340 000002   01640   D10:    CAIL    H,2;             HERE WE HAVE A SLASH
000122  254000 000075'  01650           JRST    D1;              BUT ONLY 2 ARE ALLOWED
000123  322140 000075'  01660           JUMPE   D,D1;            AND I MUST HAVE A NUMBER > 0
000124  313147 000170'  01670           CAMLE   D,R1(H);         CHECK RANGE
000125  254000 000075'  01680           JRST    D1;              GO ASK AGAIN
000126  202147 000000   01690           MOVEM   D,MONTH(H);      SAVE MONTH OR DAY
000127  402000 000003   01700           SETZM   D;               ZERO CONVERTER
000130  344340 000102'  01710           AOJA    H,D2
                        01720
000131  322140 000135'  01730   D12:    JUMPE   D,D14;           HERE A CR, IF NO DIGITS USE OLD VALUES
000132  305140 000102   01740           CAIGE   D,↑D66;          CHECK YEAR
000133  254000 000075'  01750           JRST    D1;              NOT UP TO DATE
000134  202140 000000   01760           MOVEM   D,YEAR
```

ROUTINE TO INPUT DATE AND TIME

```
000135  402000 000003   01780   D14:   SETZM   D
000136  260740 000274'  01790          TSX     LM
000137  201040 000006   01800          MOVEI   B,6
000140  200100 000575'  01810          MOVE    C,[POINT 7,S2]
000141  265300 000231'  01820          JSP     G,TS
000142  265300 000262'  01830   D16:   JSP     G,TTCH;        GET A CHAR FROM  TTY
000143  306240 000072   01840          CAIN    F,":"
000144  254000 000161'  01850          JRST    D20;           ITS A COLON
000145  306240 000015   01860          CAIN    F,15
000146  254000 000164'  01870          JRST    D22;           ITS A CR
000147  405240 000160   01880          ANDI    F,160
000150  302240 000060   01890          CAIE    F,60;          MUST BE NUMERIC
000151  254000 000135'  01900          JRST    D14;           NON-NUMERIC
000152  712040 000005   01910          DATAI   TTY,F;         RECOVER CHARACTER
000153  405240 000017   01920          ANDI    F,17;          MASK TO NUMERIC
000154  221140 000012   01930          IMULI   D,↑D10
000155  270140 000005   01940          ADD     D,F;           CONVERT TO BINARY
000156  301140 000074   01950          CAIL    D,↑D60;
000157  254000 000135'  01960          JRST    D14;           TIME COMPONENT MUST BE < 60
000160  254000 000142'  01970          JRST    D16;           GET NEXT
                        01980
000161  202140 000000   01990   D20:   MOVEM   D,HR
000162  402000 000003   02000          SETZM   D
000163  254000 000142'  02010          JRST    D16
                        02020
000164  202140 000000   02030   D22:   MOVEM   D,MIN
000165  260740 000072'  02040          TSX     ADATE;         CONVERT TO ASCII
000166  260740 000117   02050          TSX     WJD;           DUMP SYSTEM ON DRUM
000167  263740 000000   02060          DONE;                  DONE WITH INITIALIZATION
                        02070
000170  000000 000014   02080   R1:    DEC     12;            BIGGEST MONTH
000171  000000 000037   02090          DEC     31;            BIGGEST DAY
```

```
                        02110     ;       ASK ABOUT DDT AND ALLOCATE USE OF USER BLOCKS ACCORDINGLY
                        02120
000172  201440 000070'  02130     D50:    MOVEI   J,DATIME
000173  305440 040000   02140             CAIGE   J,40000
000174  254000 000216'  02150             JRST    D70;            DDT AVAILABLE ANYWAY
000175  260740 000274'  02160             TSX     LM
000176  201040 000012   02170             MOVEI   B,↑D10
000177  200100 000576'  02180             MOVE    C,[POINT 7,S3]
000200  265300 000231'  02190             JSP     G,TS;           ASK FOR DDT
                        02200
000201  265300 000262'  02210     D51:    JSP     G,TTCH;         GET TTY CHAR
000202  306240 000131   02220             CAIN    F,"Y"
000203  254000 000216'  02230             JRST    D70;            YES - DDT
000204  306240 000116   02240             CAIN    F,"N"
000205  254000 000207'  02250             JRST    D60;            NO DDT
000206  254000 000172'  02260             JRST    D50;            BAD CHAR - TRY AGAIN
                        02270
000207  260740 000274'  02280     D60:    TSX     LM
000210  201040 000006   02290             MOVEI   B,6
000211  200100 000577'  02300             MOVE    C,[POINT 7,S5]
000212  265300 000231'  02310             JSP     G,TS;           SAY: NO DDT
000213  260740 000274'  02320             TSX     LM
000214  201440 000067'  02330             MOVEI   J,DDT
000215  254000 000223'  02340             JRST    D71
                        02350
000216  260740 000274'  02360     D70:    TSX     LM
000217  201040 000015   02370             MOVEI   B,↑D13
000220  200100 000600'  02380             MOVE    C,[POINT 7,S4]
000221  265300 000231'  02390             JSP     G,TS;           SAY: DDT AVAILABLE
000222  260740 000274'  02400             TSX     LM
000223  201100 000000   02410     D71:    MOVEI   C,0
000224  305440 040000   02420     D73:    CAIGE   J,40000
000225  263740 000000   02430             DONE
000226  476002 000000   02440             SETOM   CORE(C);        BUSY BLOCKS USED BY SYSTEM
000227  275440 002000   02450             SUBI    J,2000
000230  344100 000224'  02460             AOJA    C,D73
                        02470
```

```
                          02490     ;        WRITE AND READ CONSOLE TTY
                          02500
                          02510     ,        SEND STRING TO CONSOLE TTY
                          02520     ,                B IS CHAR COUNT, C IS POINTER, F IS USED
                          02530
000231  134240 000002     02540     TS:      ILDB     F,C
000232  712140 000005     02550              DATAO    TTY,F
000233  712340 000010     02560              CONSO    TTY,10;
000234  254000 000233'    02570              JRST     .-1;              WAIT FOR FLAG
000235  712200 000200     02580              CONO     TTY,200;          TURN OFF FLAG
000236  372000 000001     02590              SOSE     B;                COUNT DOWN
000237  254000 000231'    02600              JRST     TS;               MORE TO DO
000240  254006 000000     02610              JRST     (G) ;             RETURN
                          02620
000241  422032 442564     02630     S1:      ASCII    $DATE: $
000242  200000 000000
000243  522231 542564     02640     S2:      ASCII    $TIME: $
000244  200000 000000
000245  422112 437532     02650     S3:      ASCII    $DDT?-Y/N: $
000246  545371 635100
000247  422112 420202     02660     S4:      ASCII    $DDT AVAILABLE$
000250  532031 146202
000251  412310 500000
000252  472364 042210     02670     S5:      ASCII    $NO DDT$
000253  520000 000000
                          02680     ,        OUTPUT DATE AND TIME TO CONSOLE
                          02690
000254  260740 000274'    02700     DTO:     TSX      LM
000255  201040 000017     02710              MOVEI    B,↑D15
000256  200100 000301'    02720              MOVE     C,PDATE
000257  265300 000231'    02730              JSP      G,TS
000260  260740 000274'    02740              TSX      LM
000261  263740 000000     02750              DONE
                          02760
                          02770     ,        READ AND ECHO ON TTY
                          02780
000262  712340 000040     02790     TTCH:    CONSO    TTY,40
000263  254000 000262'    02800              JRST     .-1;              WAIT FOR A CHAR
000264  712200 001000     02810              CONO     TTY,1000;         FLAG OFF
000265  712040 000005     02820              DATAI    TTY,F;            READ CHAR
000266  712140 000005     02830              DATAO    TTY,F;            ECHO BACK
000267  712340 000010     02840              CONSO    TTY,10
000270  254000 000267'    02850              JRST     .-1;              WAIT ON OUTPUT
000271  712200 000200     02860              CONO     TTY,200;          FLAG OFF
000272  405240 000177     02870              ANDI     F,177;            JUST 7 BITS
000273  254006 000000     02880              JRST     (G) ;             RETURN
                          02890
                          02900     ,        MOVE CARRIER TO LEFT MARGIN
                          02910
000274  201040 000002     02920     LM:      MOVEI    B,2
000275  200100 000601'    02930              MOVE     C,[POINT 7,CRLF]
000276  265300 000231'    02940              JSP      G,TS
000277  263740 000000     02950              DONE
                          02960
000300  064240 000000     02970     CRLF:    OCT      064240000000;     CARIAGE RETURN,LINE FEED AND ZERO
000301  440700 000000     02980     PDATE:   POINT    7,DATE;           POINTER TO DATE
```

```
                                03020     ;        ENTERED FOLLOWING READING JOSS FROM DRUM
                                03030
000302  200100 000042           03040     RECOVER:MOVE    C,42
000303  200140 000043           03050             MOVE    D,43
000304  201040 000525'          03060             MOVEI   B,ACS;          PLACE TO SAVE THE ACS FOR A MOMENT
000305  251040 000544'          03070             BLT     B,ACS+17
000306  200040 000030           03080             MOVE    B,30
000307  202040 000526'          03090             MOVEM   B,ACS+1;        AND REG 1
000310  256000 000546'          03100             XCT     BEG3;           SET UP PUSH REGISTER
000311  260740 000322'          03110             TSX     CLEAR;          CLEAR MACHINE FLAGS
000312  260740 000333'          03120             TSX     DISPLAY;        SEND ERROR MESSAGE AND DUMP
000313  260740 000415'          03130             TSX     RESTORE;        SET UP TO GO
000314  712200 000006           03140             CONO    TTY,6;          ENABLE TTY OUTPUT CHANNEL
000315  700600 002402           03150             CONO    PI,2402;        ENABLE CHANNEL FOR TTY
000316  712300 000020           03160             CONSZ   TTY,20
000317  254000 000316'          03170             JRST    .-1;            WAIT UNTIL NON-BUSY
000320  260740 000165'          03180             TSX     ADATE
000321  254000 000551'          03190             JRST    BEG5;           AND GO.
                                03200
```

CLEAR MACHINE STATUS

```
000322  201040 000007   03220    CLEAR:  MOVEI   B,7
000323  254400 000324'  03230            JRST    10,.+1;          DISMISS ALL INTERRUPTS
000324  367040 000323'  03240            SOJG    B,.-1
000325  712200 001200   03250            CONO    TTY,1200;        TTY FLAGS OFF
000326  700200 111000   03260            CONO    APR,111000;      PROCESSOR FLAGS OFF
000327  722600 000000   03270            CONO    MT1,0;           TAPE INTERRUPTS OFF
000330  720200 000000   03280            CONO    DC,0;            CLEAR DATA CONTROL
000331  727200 000000   03290            CONO    270,0;           CLEAR DISC FILE
000332  263740 000000   03300            DONE
                        03310
```

```
000333  260740 000000    03330     DISPLAY:TSX     CMESS
000334  260740 000356'   03340             TSX     MR0;            SEND ERROR NUMBER
000335  260740 000333'   03350             TSX     CMESS
                         03360             XMT ↑D13,M100;        AC TITLE
                             OPDEF X[↑D13B12]
000336  000640 000374'   X M100
000337  201240 000525'   03370             MOVEI   F,ACS;          PLACE THE ACS ARE STORED
000340  260740 000335'   03380     EC1:    TSX     CMESS;          DISPLAY ACS
000341  260740 000363'   03390             TSX     MR1
000342  305240 000545'   03400             CAIGE   F,ACS+20
000343  254000 000340'   03410             JRST    EC1
000344  260740 000340'   03420             TSX     CMESS
                         03430             XMT ↑D50,M101;        VARIOUS @ CELLS
                             OPDEF X[↑D50B12]
000345  003100 000377'   X M101
000346  200040 000536'   03440             MOVE    B,ACS+11;       REGISTER 11
000347  202040 000112    03450             MOVEM   B,SA
000350  201240 000112    03460             MOVEI   F,SA
000351  260740 000344'   03470             TSX     CMESS
000352  260740 000363'   03480             TSX     MR1
000353  260740 000351'   03490             TSX     CMESS
                         03500             XMT 5,M102; MORE LINE FEEDS
                             OPDEF X[5B12]
000354  000240 000411'   X M102
000355  263740 000000    03510             DONE
                         03520
```

```
                                03540      ;        MAKE UP HALT NUMBER MESSAGE
                                03550
000356  200240 000040           03560      MR0:     MOVE     F,40;            SAVE HALT CODE
                                03570               XMT ↑D18,M102
                                    OPDEF X[↑D18B12]
000357  001100 000411'          X M102
                                03580
000360  200000 000005           03590               MOVE     A,F
000361  260740 000000           03600               TSX      OCTW
000362  263740 000000           03610               DONE
                                03620
                                03630      ;        FORMAT FOUR OCTAL WORDS FROM (F) INTO A LINE
                                03640
000363  201300 000004           03650      MR1:     MOVEI    G,4
000364  200005 000000           03660      MR1.1:   MOVE     A,(F)
000365  260740 000371'          03670               TSX      O1
000366  350000 000005           03680               AOS      F
000367  367300 000364'          03690               SOJG     G,MR1.1
000370  263740 000000           03700               DONE
                                03710
000371  260740 000361'          03720      O1:      TSX      OCTW
                                03730               XMT 3,BLANKS
                                    OPDEF X[3B12]
000372  000140 000422'          X BLANKS
000373  263740 000000           03740               DONE
                                03750
000374  406070 352632           03760      M100:    ASCII    ?ACCUMULATORS:?
000375  526310 152236
000376  512467 200000
000377  476150 642634           03770      M101:    ASCII    ?OFFENDING LOCATION (11):  CONTENTS:(@11): @17: @0:?
000400  422231 643500
000401  462370 340650
000402  446371 620120
000403  305425 135100
000404  202071 747250
000405  426352 451564
000406  242006 130522
000407  351010 030556
000410  351010 030164
000411  064241 205024           03780      M102:    OCT      064241205024;   CR AND 4 LFS
000412  442031 452100           03790               ASCII    ?HALT NUMBER: ?
000413  472531 541212
000414  511644 000000
                                03800
```

RESTORE DATE, TIME, ETC.

```
000415  200040 000421'   03820     RESTORE:MOVE    B,RE1
000416  251040 000164'   03830             BLT     B,MIN
000417  260740 000423'   03840             TSX     SLM;            SET LOWER MEMORY
000420  263740 000000    03850             DONE
                         03860
000421  000031 000134'   03870     RE1:    XWD     31,YEAR
000422  201004 020100    03880     BLANKS: ASCII   ?     ?
                         03890
                         03900     ;       SET UP LOWER MEMORY
                         03910
000423  200000 000602'   03920     SLM:    MOVE    A,[XWD LOC41,41]
000424  251000 000137    03930             BLT     A,137
000425  263740 000000    03940             DONE
                         03950
                         03960
```

```
03980     ;       0-17     EXEC LOADER (SHADOW MEMORY)
03990     ;       20-27    PAPER TAPE LOADER - RIM
04000     ;       30       SAVE CELL FOR REG 1 DURING RECOVERY
04010     ;       31-35    SAVE CELLS FOR DATE AND TIME IN RECOVERY
04020     ;       36-37    USED BY DDT
04030     ;       40-41    UUO CELLS
04040     ;       42-57    INTERRUPT SYSTEM TRAP CELLS
04050     ;       60-77    PART OF DRUM READ ROUTINE
04060     ;       100      PSEUDO SWITCHES
04070     ;       101      DUMP ENTRY
04080     ;       102      PSEUDO CONSOLE SIGNAL CELL
04090     ;       103      ENTRY TO READ DRUM ROUTINE
04100     ;       104-137  DRUM READ AND WRITE ROUTINES
04110
```

```
                                04130     ;        CONTENTS OF CELLS 41-137
                                04140
000426  264000 000000           04150     LOC41:   JSR      PROP
000427  000000 000000           04160              Z
000430  264000 000214'          04170              JSR      DDT;            43
000431  264000 000000           04180              JSR      DRMR;           44 - CHANNEL 2, THE DRUM
000432  000000 000000           04190              Z
000433  264000 000000           04200              JSR      C25;            46 - CHANNEL 3, 630 INPUT
000434  000000 000000           04210              Z
000435  264000 000000           04220              JSR      C26;            50 - CHANNEL 4, 630 OUTPUT
000436  000000 000000           04230              Z
000437  254000 000430'          04240              JRST     DDT;            52 - CHANNEL 5, DISC FILE
000440  000000 000000           04250              Z
000441  264000 000000           04260              JSR      CTYR;           54 - CHANNEL 6, TAPE AND CONSOLE TTY
000442  000000 000000           04270              Z
000443  264000 000000           04280              JSR      APRR;           56 - CHANNEL 7, PROCESSOR
000444  000000 000000           04290              Z
                                04300
                                04310     ;        DEFINITIONS FOR WHEN DRUM CODE IS MOVED TO LOW MEMORY
                                04320
                                04330     BTW=116
                                04340     RJ=60
                                04350     RIO=72
                                04360     DS=73
                                04370     DS1=74
                                04380     SWITCH=100
                                04390     FAKE=102
                                04400     RJD=103
                                04410     WJD=117
                                04420     WD=132
                                04430     RC=135
                                04440     SA=112
                                04450     CKD=63
```

```
                                04470   ;       READ AND WRITE JOSS ON DRUM
                                04480
000445  701200  000000          04490   RJ.:    CONO    DP,0;           167 TO READ
000446  265140  000074          04500           JSP     D,DS1
000447  740200  000230          04510           CONO    DR,230;         READ
000450  740340  001000          04520   CKD.:   CONSO   DR,1000;        CHECK DRUM ERRORS
000451  701300  100060          04530           CONSZ   DP,100060;       IOP ERRORS
000452  254200  007770          04540           JRST    4,7770
000453  740340  000100          04550           CONSO   DR,100;          WAIT FOR DONE
000454  254000  000063          04560           JRST    CKD
000455  740200  000270          04570           CONO    DR,270;          DESELECT
000456  254002  000000          04580           JRST    (C)
000457  740000  040000          04590   RIO.:   XWD     -40000,40000;    I/O WORD - 16K FROM UPPER BOX
000460  701200  000100          04600   DS.:    CONO    DP,100;          SELECT FOR WRITE
000461  701140  000072          04610   DS1.:   DATAO   DP,RIO;          DRUM SELECT
000462  740140  000001          04620           DATAO   DR,B;            UNIT AND TRACK
000463  740200  000260          04630           CONO    DR,260;          SELECT
000464  254003  000000          04640           JRST    (D)
                                04650
                                04660   HALTS=20000
000465  000000  000000          04670   L100:   Z;      100 - PSEUDO SWITCH CELL
000466  254000  000003'         04680           JRST    DUMP
000467  000000  000000          04690   FAKE.:  Z;                       102 - PSEUDO SIGNAL CELL
000470  201040  000000          04700   RJD.:   MOVEI   B,0;             103 - RECOVERY ENTRANCE
000471  265100  000060          04710           JSP     C,RJ
000472  200040  000116          04720           MOVE    B,BTW;           MOVE TO LOW MEMORY
000473  251040  037777          04730           BLT     B,37777
000474  201040  002000          04740           MOVEI   B,2000
000475  265100  000060          04750           JSP     C,RJ;            FILL UPPER BOX
000476  254000  000302'         04760           JRST    RECOVER
                                04770   SA.:    BLOCK   4;               FOUR SAVE CELLS
000503  040140  000140          04780   BTW.:   XWD     40140,140;       BLT CONTROL WORD
000504  201040  002000          04790   WJD.:   MOVEI   B,2000;          WRITE JOSS ON DRUM
000505  265100  000132          04800           JSP     C,WD;            WRITE
000506  265100  000135          04810           JSP     C,RC;            READ COMPARE
000507  201040  040000          04820           MOVEI   B,40000
000510  251040  077777          04830           BLT     B,77777;         MOVE LOW TO HIGH MEMORY
000511  201040  000000          04840           MOVEI   B,0
000512  265100  000132          04850           JSP     C,WD;            WRITE LOW BOX
000513  265100  000135          04860           JSP     C,RC;            READ COMPARE
000514  201040  002000          04870           MOVEI   B,2000
000515  265100  000060          04880           JSP     C,RJ;            READ BACK UPPER BOX
000516  263740  000000          04890           DONE
                                04900
000517  265140  000073          04910   WD.:    JSP     D,DS;            WRITE DRUM - SELECT
000520  740200  000220          04920           CONO    DR,220;          WRITE DRUM
000521  254000  000063          04930           JRST    CKD;             GO CHECK
                                04940
000522  265140  000073          04950   RC.:    JSP     D,DS;            READ COMPARE - SELECT
000523  740200  000210          04960           CONO    DR,210;          COMPARE DRUM AND CORE
000524  254000  000063          04970           JRST    CKD
                                04980
                                04990   ACS:    BLOCK   20;              TEMP STORAGE FOR ACS
                                05000
```

```
000545  700200 210000    05020    BEGIN:  CONO    APR,210000;     RESET STUFF
000546  200740 000000    05030    BEG3:   MOVE    PP,PPW
000547  260740 000070'   05040            TSX     DATIME;         ASK FOR DATE AND TIME
000550  712200 000206    05050            CONO    TTY,206;        TURN OFF ITO FLAG AND ENABLE CHANNEL
000551  260740 000000    05060    BEG5:   TSX     C20;            INITIALIZE IRWIN
000552  700200 122007    05070            CONO    APR,122007;     SET PROCESSOR FLAGS AND CHANNEL
000553  722200 000000    05080            CONO    MTC,0
000554  722600 000002    05090            CONO    MT1,2;          GET TAPE DRIVE ZERO READY
000555  700600 020000    05100            CONO    PI,20000;       ENABLE PARITY ERROR TRAP
000556  571700 777776    05110            HRREI   S,-2;           CONTEXT FOR TAPE
000557  201040 000000    05120            MOVEI   B,TYPE6
000560  260740 000000    05130            TSX     SOUT
000561  571700 777777    05140            HRREI   S,-1;           CONTEXT FOR CONSOLE TTY
000562  260740 000000    05150            TSX     HMES;           LOG HEADING MESSAGE
000563  254000 000000    05160            JRST    SIGPR
                         05170
000564  000000 000120    05180            END
000565  264000 000000'
000566  720100 000016
000567  440300 000120
000570  440300 000020
000571  440600 777700
000572  170300 000004
000573  777754 777677
000574  440700 000241'
000575  440700 000243'
000576  440700 000245'
000577  440700 000252'
000600  440700 000247'
000601  440700 000300'
000602  000426'000041
```

THERE ARE NO ERRORS

PROGRAM BREAK IS 000603

SYMBOL TABLE

A	000000	
ACS	000525'	
ADATE	000320'	EXT
APPR	000443'	EXT
B	000001	
BEG3	000546'	
BEG5	000551'	
BEGIN	000545'	INT
BLANKS	000422'	
BTW	000116	
BTW.	000503'	
C	000002	
C20	000551'	EXT
C25	000433'	EXT
C26	000435'	EXT
CAPR	000007	
CCTY	000006	
CDAT	000006	
CDC	000001	
CDRM	000002	
CH630	000030	
CHAPR	000001	
CHCTY	000002	
CHDAT	000000	
CHDC	000100	
CHDRM	000040	
CI630	000003	
CKD	000063	
CKD.	000450'	
CLEAR	000322'	
CMESS	000353'	EXT
CO630	000004	
CORE	000226'	EXT
CRLF	000300'	
CTYR	000441'	EXT
D	000003	
D1	000075'	
D10	000121'	
D12	000131'	
D14	000135'	
D16	000142'	
D2	000102'	
D20	000161'	
D22	000164'	
D50	000172'	
D51	000201'	
D60	000207'	
D70	000216'	
D71	000223'	
D73	000224'	
DATE	000301'	EXT
DATIME	000070'	INT
DDK	000050	
DDT	000437'	EXT
DISMIS	000000'	

SYMBOL TABLE

DISPLA	000333'	
DMT	000020	
DP	000010	
DR	000400	
DRMR	000431'	EXT
DS	000073	
DS.	000460'	
DS1	000074	
DS1.	000461'	
DTO	000254'	
DUMP	000003'	INT
E	000004	
EC1	000340'	
F	000005	
FAKE	000102	INT
FAKE.	000467'	
G	000006	
H	000007	
HALTS	020000	
HMES	000562'	EXT
HR	000161'	EXT
I	000010	
IOA	777700	
J	000011	
K	000012	
L	000013	
L100	000465'	INT
LM	000274'	
LOC41	000426'	
M	000014	
M100	000374'	
M101	000377'	
M102	000411'	
MIN	000416'	EXT
MONTH	000126'	EXT
MR0	000356'	
MR1	000363'	
MR1.1	000364'	
MT1	000224	
MT2	000230	
MTC	000220	
N	000015	
O1	000371'	
OCTW	000371'	EXT
PDATE	000301'	
PP	000017	
PPW	000546'	EXT
PROP	000426'	EXT
R1	000170'	
RC	000135	
RC.	000522'	
RE1	000421'	
RECOVE	000302'	
RESTOR	000415'	
RIO	000072	

SYMBOL TABLE

RIO.	000457'	
RJ	000060	
RJ.	000445'	
RJD	000103	INT
RJD.	000470'	
S	000016	
S1	000241'	
S2	000243'	
S3	000245'	
S4	000247'	
S5	000252'	
SA	000112	
SA.	000477'	
SAVE	000120	
SIGPR	000563'	EXT
SLM	000423'	
SOUT	000560'	EXT
SWITCH	000100	INT
TD05	000022'	
TD1	000024'	
TD2	000030'	
TD3	000035'	
TD4	000037'	
TS	000231'	
TTCH	000262'	
TYPE6	000557'	EXT
WD	000132	
WD.	000517'	
WJD	000117	INT
WJD.	000504'	
YEAR	000421'	EXT

END OF ASSEMBLY

REFERENCES

1. Bryan, G. E., JOSS: User Scheduling and Resource Allocation, The RAND Corporation, RM-5216-PR, January 1967.
2. Bryan, G. E., JOSS: Accounting and Performance Measurement, The RAND Corporation, RM-5217-PR, June 1967.
3. Programmed Data Processor-6 Handbook F-65, Digital Equipment Corporation, Maynard, Mass., August 1964 (including F-65 Change Notice No. 3).
4. PDP-6 Programming Manual: MACRO-6 Assembly Language, DEC-6-0-TP-MAC-LM-F-ACT01, Digital Equipment Corporation, Maynard, Mass.

JOSS BIBLIOGRAPHY

PUBLICATIONS OF CURRENT INTEREST

Baker, C. L., JOSS: Introduction to a Helpful Assistant, The RAND Corporation, RM-5058-PR, July 1966.

-----, JOSS: Console Design, The RAND Corporation, RM-5218-PR, February 1967.

Bryan, G. E., JOSS: Accounting and Performance Measurement, The RAND Corporation, RM-5217-PR, June 1967.

-----, JOSS: Introduction to the System Implementation, The RAND Corporation, P-3486, November 1966; also published by the Digital Equipment Computer Users Society, DECUS Proceedings, Fall 1966.

-----, JOSS: 20,000 Hours at the Console--A Statistical Summary, The RAND Corporation, RM-5359-PR, August 1967.

-----, JOSS: User Scheduling and Resource Allocation, The RAND Corporation, RM-5216-PR, January 1967.

Bryan, G. E., and E. W. Paxson, The JOSS Notebook, The RAND Corporation, RM-5367-PR, August 1967.

Bryan, G. E., and J. W. Smith, JOSS Language (Aperçu and Précis, Pocket Précis, Poster Précis), The RAND Corporation, RM-5377-PR, August 1967.

Gimble, E. P., JOSS: Problem Solving for Engineers, The RAND Corporation, RM-5322-PR, May 1967.

Greenwald, I. D., JOSS: Arithmetic and Function Evaluation Routines, The RAND Corporation, RM-5028-PR, September 1966.

-----, JOSS: Console Service Routines (The Distributor), The RAND Corporation, RM-5044-PR, September 1966.

-----, JOSS: Disc File System, The RAND Corporation, RM-5257-PR, February 1967.

Marks, S. L., and G. W. Armerding, The JOSS Primer, The RAND Corporation, RM-5220-PR, August 1967.

Smith, J. W., JOSS: Central Processing Routines, The RAND Corporation, RM-5270-PR, August 1967.

PUBLICATIONS OF HISTORICAL INTEREST

Baker, C. L., JOSS: Scenario of a Filmed Report, The RAND Corporation, RM-4162-PR, June 1964.

"The JOSS System: Time-Sharing at RAND," Datamation, Vol. 10, No. 11, November 1964, pp. 32-36. (This article is based on RM-4162-PR above.)

Shaw, J. C., JOSS: A Designer's View of an Experimental On-Line Computing System, The RAND Corporation, P-2922, August 1964; also published in AFIPS Conference Proceedings (1964 FJCC), Vol. 26, Spartan Books, Inc., Baltimore, Md., 1964, pp. 455-464.

-----, JOSS: Conversations with the Johnniac Open-Shop System, The RAND Corporation, P-3146, May 1965.

-----, JOSS: Examples of the Use of an Experimental On-Line Computing Service, The RAND Corporation, P-3131, April 1965.

-----, JOSS: Experience with an Experimental Computing Service for Users at Remote Typewriter Consoles, P-3149, May 1965.