

```

BRS IDENT H19 3/7/68
* ENTRY POINTS
ENTRY T,BSX,SS01,SS02,SS03,BST,EPOPX,BSE
ENTRY MGET,MPUT,PTRAP,TRAP,MDEL,TRAPS
ENTRY MX01,MS03,MX13
ENTRY TRAPI,TRAPM
ENTRY RFK,TFK,FKSTW,GFK,SCFK,DFK,EFK
IF -1
ENTRY MSP,MSX,NRTRAP,NRPOPX,NRXPOP
ENDF

```

142 MONIT
/BRS/

```

*
* 'SBRM','SBRR','EXS' 6/28/66
*

```

```

SBRM POPD 17000000B,1,1,0,1
SMB STX SS03
EAX* 0
XXA
ETR ADMSK
MRG =40040000B
XMA 0
ETR =37737777B
STA* 0
CXA
LDX SS03
BRR 0

```

```

SBRR POPD 17100000B,1,1,0,1
SRB STX SS03
LDX* 0
EAX 0,6
STX T
LDX SS03
BRR* T (TIME 40 US)

```

```

EXS POPD 15200000B,1,1,0,1
XSP STX SS03
LDX PACPTR
SKN PQU,2
BRU TRAP
LDX SS03
LDX* 0
STX XSP1
LDX SS03
XSP1 ZRO
BRR 0
MIN 0
BRR 0

```

```

*
* 'BRS' 9/27/65
*
* THIS ROUTINE DISPATCHES ON THE ADDRESS OF A 'BRS'
*
T ZRO
* UNIVERSAL TEMPORARY STORAGE
BSX ZRO

```

```

* BRS EXIT
SS01 ZRO
* SAVE (A)
SS02 ZRO
* SAVE (B)
SS03 ZRO
* SAVE (X)
SSR EXT

```

```

*
BRS POPD 17300000B, 1, 1, 0, 1
BS STA SS01
STB SS02
STX SS03
EAX* 0
LDA =-40000B
ADM 0
CXA
ETR ADMSK
SKG =BSTU-BST
BRU BS1
BST BRU TRAP- (0)
BRU MONOPN Open file
BRU MONCLS close file
EAX SPFDC -
BRU MPT Release Port (4)
BRU TRAP
BRU TRAP
BRU CDMP-
BRU IOH Close all files (8)
BRU FKST Start fork
BRU PPAR Program panic
BRU CIB Clear Input Buffer
BRU CET Set Echo Table (12)
BRU SKI skip if Input Empt
BRU DOB Delay on output buffer
NOP 0 Read input file name
NOP 1 Open input (16)
NOP 2 Close all
NOP 3 Read name, look up in PD
NOP 4 Open output
NOP 5 Close type RTs (20)
BRU FNA Floating Negate
BRU RUPAR -
BRU LNKS Link set
BRU LNKC Link clear (24)
BRU MSGS Accept messages
BRU SKROUT skip if escape
BRU TTYOFF - BE+6 (?)
BRU CLCW- (28)
BRU COB Clear output buffer
BRU FKRD Read fork status
BRU FKWT wait for fork panic
BRU FKTM Terminate fork (32)
EAX GETSTR Read string

```

EAX	OUTMSG	Output message	
EAX	OUTSTR	Output string	
EAX	OUTNUM	Output number (36)	
BRU	TRAP		
EAX	GETNUM	Input number	skip if PMT entry already in use ???
BRU	SPMTB -		
BRU	RDET	Read echo table (40)	
BRU	TRAP		
BRU	RREAL	Read RTC	
BRU	RDRL	Read RT pseudo RL	
BRU	STRL	set RL (44)	
BRU	SQO	Simulate short quantum overflow	
BRU	NROUT	Turn escape off	
BRU	SROUT	Turn escape on	
NOP	6	Look up % filename (48)	
BRU	SRIR	Read interrupts	
BRU	FFIX	Fix	
BRU	FFLT	Float	
NOP	1002B	Formatted in (52)	
NOP	1003B	out	
BRU	TRAP -		
BRU	TRAP -		
BRU	TRAP	(56)	
BRU	COO	Guarantee 16ms	
BRU	CLKTIC -		Pick up clock cycle time
IF RANFIL; BRU CBRF; ELSEF 1; BRM TRAPX; ENDF -			
NOP	9	Look up % insert (60)	
BRU	ARD - BE+1	Read disc	
BRU	AWD - BE+2	write disc	
BRU	TRAP -		
BRU	TRAP -	(64)	
BRU	A2K -	Assign 2K block	
BRU	DFDL	Delete file descriptor	
BRU	DFER	Delete file index block	
BRU	TRAP -	(68)	
BRU	TRAP -		
BRU	D2K -	Delete 2K block	
BRU	SKXEC	skip if exec - skip if bit 0 of NEXFLG is set	
BRU	TRAP	(72)	
BRU	TRAP		
BRU	RD2K -	BE+9 (?)	Read from attached 2K block
BRU	WR2K -	BE+10 (?)	write on attached 2K block
BRU	DA2K -	(76)	Delete all 2K blocks
BRU	RDUSSR -		Read USS relabeling
BRU	SAIR	Arm software interrupts	
BRU	SIIR	Cause software interrupt	
BRU	TRAP	(80)	
BRU	WREAL	Dismiss for specified time	
BRU	TRAP -	(UNTIL SWSF IS FIXED)	
BRU	TRAP -		
BRU	TRAP -	(84)	
BRU	SET8P	Set 8-bit mode	
BRU	CLR8P	Clear 8-bit mode	
BRU	DFRX	Read index block	

BRU	RTEX	Read time of execution (88)	
BRU	TRAP		
BRU	DFR	Declare fork for subcall	
BRU	TRAP		
BRU	TRAP		(92)
BRU	TRAP		
BRU	TRAP		
BRU	TRAP		(BRS 95 AND 96 ARE DANGEROUS)
BRU	TRAP		(96)
NOP	10		
BRU	TRAP		
BRU	TRAP		
BRU	TRAP		(100)
BRU	TRAP		
BRU	TRAP		
BRU	TRAP		
BRU	RSYB	Read page	} RFD (104)
BRU	WSYB	Write page	
BRU	FKWA	wait for fork to terminate	
BRU	FKRA	Read all lower fork statuses	
BRU	FKTA	Terminate all lower forks	(108)
BRU	DMS	Dismiss	
BRU	RDU	Read device, unit	
BRU	BRSRET	Return from BRS	
BRU	TSOFF	Turn off TTY	(112)
BRU	DFCD	Compute disc file size	
BRU	TRAP		
BRU	TRAP		
BRU	RURL	Read user RL	(116)
BRU	SURL	Set user RL	
BRU	TGET	Get tape	
BRU	TREL	Release tape	
BRU	APMTE	Assign PMT entry	(120)
BRU	DPMTE	Delete PMT entry	
BRU	MPAN	Memory panic	
BSTU	BRU	TRAP	
BS1	LDA	BST, 2	
	STA	BSX	
	LDA	SS01	
	LDX	PACPTR	
	EXU	BSX	(GO IF DIRECT)
* SAVE RETURN IF CLASS II			
BSE	LDX	Ø	
	STX	SBRST	
	LDX	=UBRSET	
	STX	Ø	
	EXU	BSX	
	STX	T	(STORE DISPATCH)
	LDX	SS03	
	BRU*	T	(GO TO CLASS II)
* RETURN FROM CLASS II BRS'S			
EPOPX	BRR	SBRST	

*

* CLASS 3/4 BRS SETUP AND RETURN ROUTINE FOR SYS 1.8 FF

*

```
UBRSET BRM      GFK
        BRU      FKSTW
        BRM      STFK
        MRG      X6
        LDX      FK04
        STA      PQU,2
        LDA      BSX
        RCY      9
        ETR      =37B
        CAX
        LDA      BRSTV,2
        RSH      12
        STA      T
        LDA      =4001B
        LCY      12
        STA      0
        LDX      PACPTR
        LDA      RL1,2
        STA      UBRL1
        LDB      RL2,2
        STB      UBRL2
        LDA      =NCMEM*1000000B+10000B
        MRG      T
        CLB
        LDX      FK04
        STA      RL1,2
        STB      RL2,2
        LDA      SS01
        STA      PA,2
        STA      UBA
        LDA      SS02
        STA      PB,2
        STA      UBB
        LDA      SS03
        STA      PX,2
        STA      UBX
        LDA      BSX
        ETR      =777B
        ADD      0
        STA      PL,2
        CLA
        STA      PIM,2
        LDA      =EXECL
        RSH      15
        LDA      JOB
        LSH      15
        MRG      X4
        STA      PTAB,2
        LDA      SBRST
        STA      0
        STA      UPL
        LDB      =700004B
```

```

BRU      POPST
BRSRET  LDA      PPTR, 2
        MRG      PLMSK
        CAX
        LDA      PTEST, 2
        SKE      = 700004B
        BRU      TRAP
        SUB      = 3
        STA      PTEST, 2
        LDA      UPL
        STA      0
        STX      TF01
        LDX      PACPTR
        BRM      DFK
        LDX      TF01
        STX      PACPTR
        BRU      STRL1

```

IF -1

* BRANCH TO NON-RESIDENT MONITOR PAGE

```

MSP     ZRO
        STA      MSR1
        STB      MSR2
        STX      MSR3
        MIN      MSP
MSP1    LDX      JOB
        LDA      RL3, 2
        LRSH     6
        LDA*     MSP
        ADD      =NMSMT
        LSH      6
        LDX      PACPTR
        LDB      RL2, 2
        LDX      RL1, 2
        XXA
        BRM      SWAP
        BRU      *-1
        BRM      LABEL
        LDA      MSR1; LDB MSR2; LDX MSR3; BRR MSP

```

*

```

MSX     ZRO
        STA      MSR1
        STB      MSR2
        STX      MSR3
        LDX      PACPTR
        BRM      CHRL
        BRU      *-2
        LDA      MSR1
        LDB      MSR2
        LDX      MSR3
        BRR      MSX

```

*

```

MSR1    ZRO

```

MSR2 ZRO
MSR3 ZRO
ENDF

NLDB EQU 100
FDBLST RPT NLDB
DATA *+1
ENDR
DATA -1
ETA2K EQU *

*BRS 65: ASSIGN 2K BLOCK

A2K LDB UEXFLG; SKN PQU,2; SKB =BIT4; BRU *+2; BRM TRAPX
SKN FDBLST; BRU *+2; BRU POPX
LDB =NLDB/3-1; STB A2K1
LDA JOB; MRG =BIT0; LDB =BIT0+177B; LDX =-NLDB
A2K3 SKM ETA2K,2; BRU A2K2
SKR A2K1; BRU A2K2; BRU POPX
A2K2 BRX A2K3; LDA SS01; LRSB 11
BRM ADB; BRU POPX; LSH 11; STA SS01
CAB; LDA* FDBLST; XMA FDBLST; COPY AX,BA
MRG JOB; MRG =BIT0
STA 0,2; MIN 0; BRU POPX
A2K1 ZRO

*BRS 70: DELETE 2K BLOCK

D2K BRM V2K; BRM TRAPX; BRM R2K; BRU POPX

*BRS 76: DELETE ALL 2K BLOCKS

DA2K LDX =-NLDB
DA2K1 LDB =BIT0+177B; LDA JOB; MRG =BIT0
SKM ETA2K,2; BRU DA2K3; BRM R2K
DA2K3 BRX DA2K1; BRU POPX

*BRS 74: READ FROM ATTACHED 2K BLOCK

RD2K BRM OK2K
STB 0,6; BRM DTH
LDB SS02; LDX SS03; BRM RTC
WT2K LDX ERCL; LDA =2; STA RD2K2; ADM 3,2
DIR; BRM IRTCT; BRM RTS
SKN RD2K2; BRU *-1; BRU POPX
RD2K2 ZRO

*BRS 75: WRITE ON ATTACHED 2K BLOCK

WR2K BRM OK2K; BRM DTH; LDB SS02; LDX SS03
BRM RTW; BRU WT2K

SRD2KI ZRO; SKN RADTRY; MIN 0; LDA =-1; STA RD2K2; BRR RD2KI

V2K ZRO; ETR =3774000B; MRG JOB; MRG =BIT0; LDX =-NLDB
V2K2 SKE ETA2K,2; BRU V2K1; MIN V2K; BRR V2K
V2K1 BRX V2K2; BRR V2K

R2K ZR0; STX R2K1; LDA ETA2K,2; LRSR 11; ETR =777B
BRM RDB; LDX R2K1; CXA; ADD =ETA2K
XMA FDBLST; STA* FDBLST; BRR R2K
R2K1 ZR0

OK2K ZR0; LDA UEXFLG; SKN PQU,2; SKA =BIT4
SKB =77B; BRM TRAPX; LDA SS01; ETR =3777B
ADD SS03; SKG =4000B; BRU **2; BRM TRAPX
CBA; LDB SS03; ETR =3777B; ADD SS03
SKG =4000B; SKB =77B; BRM TRAPX

DRMLIM LDA SS02; SKG UPLIM; SKG LOWLIM IS THIS PAGE IN SHARED LISP?
BRU **2 IT IS NOT
BRU OK2K1 IT IS

SKG =3777700B; BRM V2K; BRM TRAPX
OK2K1 LDA SS01; CAX; LDB 0,6; BRR OK2K
UPLIM 3617777B SHARED LISP ON BANDS 120-119. 2/10/69 D.C.
LOWLIM 3377777B
CDMP LDX =-30000B; LDA 30000B,2; STA 30000B,6; BRX *-2
BRU POPX

CLKTIC LDA =CLKTIM BRS 58D. PICKS UP THE CLOCK CYCLE TIME. USED
STA SS01; BRU POPX BY BBN LISP.

* SET NON-TERMINABILITY

NROUT SKN PQU,2
BRU TRAP
LDA PIM,2
MRG X1
STA PIM,2
BRU POPX

* SKIP IF TERMINATION PENDING

SKROUT LDA PIM,2
SKA X2
MIN 0
BRU POPX

* CLEAR NON-TERMINABILITY

SROUT SKN PQU,2
BRU TRAP
LDA PIM,2
SKA X1
BRU **2
BRU POPX
ETR =47777777B
XMA PIM,2
SKA X2
BRU PACQE
BRU POPX

* DECLARE FORK FOR RUBOUT

DFR CXA
LDX UTTY
STA TTYASG,2
BRU POPX

*

* MEMORY ALLOCATION LOGIC

*

* ASSIGN A POSITION IN PMT

```
PMGET  ZRO
        LDX      JOB
        LDA      PMA,2
        SKA      =27700000B
        BRU      *+2
        BRR      PMGET
        ETR      =77B
        STA      SMGET
        CLA
        LDX      =NCMEM-1
        BRU      MGET4
MGET1  EAX      1,2
        SKE*     PMTJOB
        BRU      MGET4
        MIN      PMGET
        BRR      PMGET
MGET4  SKR      SMGET
        BRU      MGET1
        BRR      PMGET
```

* ASSIGN A POSITION IN SMT

```
SMGET  ZRO
        IF -1
        ABC
        LDX      =-NSMT
SMG1   SKE      SMTE,2
        BRU      SMG2
        CXA
        ADD      =NSMT
        MIN      SMGET
        BRR      SMGET
SMG2   BRX      SMG1
        BRR      SMGET
        ENDF
```

* GET A BLOCK OF MEMORY

```
MGET  ZRO
        BRM      MX01
        SKE      =0
        BRU      MX03
        LDA      =-2
        STA      MX09
        LDX      MS03
```

* SCAN FOR LOCAL OR FIXED MEMORY FORK

```
MGET11 STX     MX08
        LDA     RL1,2
        LDB     RL2,2
        LDX     MX07
        MIN     MX09
        LCY     0,2
        SKA     =77000000B
        BRU     MGET12
```

```

LDX      MX08
SKN      PIM,2
BRU      *+2
BRU      MX03
SKN      PTAB,2
BRU      MGET13
LDA      PMTA
SKA      =40000B
BRU      MX03
MIN      MX09
BRM      PMGET
BRU      MX03
STX      MGTS2
BRM      PMTA
BRU      MX03
LDX      MS03

```

* PROPAGATE NEW BYTE AS NECESSARY

```

MGET10 STX      MX08
LDA      RL1,2
LDB      RL2,2
LDX      MX07
LCY      0,2
RCY      18
MRG      MGTS2
LCY      18
RCY      0,2
LDX      MX08
STA      RL1,2
STB      RL2,2
LDA      PPTR,2
MRG      PLMSK
CAX
SKR      MX09
BRU      MGET10

```

* SET UP NEW MAP AND CLEAR BLOCK

```

BRU      MX04
MGET12 LRSB      18
LDX      MS03
BRU      MGET10
MGET13 LDA      PPTR,2
MRG      PLMSK
CAX
BRU      MGET11

```

* ASSIGN BLOCK IN PMT

```

PMTA    ZRO
STX      MGTS1
MGET9  LDA      JOB; ADD MGTS1; ADD =1; BRM ADB; BRR PMTA
LSH      6
MRG      X1
LDX      MGTS1
STA*     PMTJOB
LDX      JOB
LDA      PMA,2
SKG      X2

```

ADD MS02
STA PMA,2
MIN PMTA
BRR PMTA

* RELEASE MEMORY

PMTA1 ZRO

*

MGTS1 ZRO

MGTS2 ZRO

MGTS3 ZRO

MGTS4 ZRO

ADB ZRO; LDB =NSEC-1; STB PMTA1

MGET3 ETR SECMSK; SUB =NDRAT*NSEC; AXC; SKE EDRAT,2; BRU MGET5

BRX *+1; BRX *-3; CAX; ADD =1; SKR PMTA1

BRU MGET3; BRR ADB

MGET5 STX MGTS3; LDB EDRAT,2; CLA; LDX =46; NOD 48

EOR X2; LCY 2,2; STX MGTS4; LDX MGTS3; STB EDRAT,2

CAX; ADD =NDRAT*NSEC; CAB; ETR SECMSK; STA MGTS3

CBA; LRSH L2NSEC; MUL =12; BAC; ADD MGTS4

LSH L2NSEC

ADD MGTS3

ADD =8*OSBND

MIN ADB

BRR ADB

MDEL1 ZRO

MDEL2 ZRO

MDEL3 ZRO

*

MPUTA ZRO

BRM MX01

SKG =NCMEM

BRU TRAP

CAX

LDA* PMTJOB

SKE =0

SKA X2

BRU TRAP

SKA =100000B

BRU *+2

BRU MPUT2

CXB

LDX MS03

SKN POU,2

BRU TRAP

CBX

MPUT2 BRM MDELA

BRU MX04

*

MDELA ZRO

STX MDEL1

LDX =-NPAC*NPPAR

MDEL4 STX MDEL2

LDA PTAB,2

```

RCY      15
ETR      =177B
SKE      JOB
BRU      MDEL7+1
LDA      RL1,2
LDB      RL2,2
LDX      JOB
LDX      RL3,2
BRM      UPRL
LDX      =-10
LDA      MDEL1
STA      MDEL3
MDEL5   SKE      SRTE,2
        BRU      MDEL6
        STX      MDEL3
        CLA
        STA      SRTE,2
MDEL6   LDA      MDEL1
        BRX      MDEL5
        SKN      MDEL3
        BRU      MDEL7
        BRM      PTRL
        STX      MDEL3
        LDX      MDEL2
        STA      RL1,2
        STB      RL2,2
        LDX      JOB
        LDA      MDEL3
        STA      RL3,2
MDEL7   LDX      MDEL2
        EAX      NPPAR-1,6
        BRX      MDEL4
        LDX      JOB
        LDA      PMA,2
        SKG      X2
        ADD      =100000B-100B
        STA      PMA,2
        LDA      MDEL1
        COPY AX,B
        LDA =76577777B
        DIR; ETR* PMTJOB; EIR; STA* PMTJOB
        ETR =37B; SKG =NSMEM-1; BRU MDEL9
        XXA; SKN RMC,2; BRU *-1
        STB RMT,2; XXA
MDEL9   CLA; XMA* PMTJOB
MPUT3   ETR =77700B; RSH 6; BRM RDB
        LDX PACPTR; BRM CHRL; BRU *-2
        BRR MDELA

RDB     ZRO; SUB =OSBND*8; CAB; ETR SECMSK
        STA PMTA1; CBA; RSH L2NSEC; CLB; RSH 23; DIV =24
        COPY BX,B; LSH L2NSEC; ADD PMTA1; SUB =NDRAT*NSEC
        XXA; LDB EDRAT,2; XXA; RCY 1,2; MRG X4; LCY 1,2
        CAX; STB EDRAT,2; BRR RDB

```

MPUT EQU MPUTA
MDEL EQU MDELA
* COMMON ENTRY FOR MGET,MPUT

MX01 ZRO
STA PMTA
ETR =34000B
STA MS01
STB MS02
STX MS03
LDX MX01
LDB -1,2
STB MX00
LRSH 11
MUL =3
LDX MS03
LDA RL1,2
LDX RL2,2
XXB
STX MX07
RCY 18
LCY 0,2
ETR =77B
STA MX06
BRR MX01
MK13 LDX MS03
BRM CHRL
BRU *-2
MX04 MIN MX00
MX03 LDA MS01
LDX MS03
BRR MX00

*
MS01 ZRO
MS02 ZRO
MS03 ZRO
MX00 ZRO
MX06 ZRO
MX07 ZRO
MX08 ZRO
MX09 ZRO

*
* 'MBEX', 'MBRO', 'APMTE', 'DPMTE', 'MPT' 6/30/66
*

* BRS'S FOR MODIFYING THE MEMORY TABLES
*

IF -1
MBEX SKN PQU,2
BRU TRAP
ETR =77B
BRM CRTA
BRU TRAP
MRG =100000B

	SKN	SS01
	EOR	=100000B
	XMA	0,2
	LCY	8
MBEX1	ETR	X4
	XMA	SS01
	ETR	=77B
	ADM	SS01
	BRU	POPX
*		
MBRO	ETR	=77B
	STA	RTW
	BRM	CRTA
	BRU	TRAP
	STA	CRTA
	SKA	=100000B
	BRU	MBRO1
	LDA	RTW
	SKG	=NCMEM-1
	BRU	MBRO1
MBRO2	LDA	0,2
	SKE	=0
	BRU	*+2
	BRU	TRAP
	SKN	SS01
	BRU	MBRO3
	SKA	=40B
	BRU	MBRO5
	MRG	=40B
	STA	0,2
	SKA	X4
	BRU	MBRO5
	BRM	OMW
	BRM	RTS
	BRU	MBRO4
MBRO3	ETR	=(NOT) 40B
	STA	0,2
	SKA	X4
	BRU	MBRO5
MBRO4	LDX	PACPTR
	BRM	CHRL
	BRU	TRAP
MBRO5	LDA	CRTA
	LCY	18
	BRU	MBEX1
MBRO1	XXB	
	SKN	PQU,2
	BRU	TRAP
	CBX	
	BRU	MBRO2
	ENDF	
*		
APMTE	SKN	PQU,2
	BRU	TRAP

```

SKG      =NCMEM
BRU      TRAP
AXC
SKE*    PMTJOB
BRU      TRAP
LDB      =-1000000B+100B
STB      MS02
LDA      =TRAP-1
STA      MX00
BRM      PMTA
BRU      MTRAP
BRU      POPX

```

```

*
DPMTE   ETR      =77B
        SKG      =0
        BRU      POPX
        SKG      =NCMEM
        BRU      TRAP
        STA      MDEL
        BRM      CRTA
        BRU      POPX
        CBX
        SKA      =100000B
        BRU      DPMTE1
DPMTE2  LDX      MDEL
        BRM      MDEL
        BRU      POPX
DPMTE1  SKN      POU,2
        BRU      TRAP
        BRU      DPMTE2

```

```

*
MPT     BRM      MPUT
        BRU      TRAP
        BRU      POPX

```

```

*
* COMPUTE RELABELING TABLE ADDRESS

```

```

CRTA    ZRO
        COPY     XB,AX
        SKG      =NCMEM-1
        BRU      CRTA1
        EAX*    PMTJOB
        LDA      0,2
        SKA      X2
        BRU      CRTA1
CRTA2   SKE      =0
        MIN      CRTA
        BRR      CRTA
CRTA1   ADD      =SMT
        CAX
        LDA      0,2
        BRU      CRTA2

```

```

*
* 'RDRL', 'RURL', 'STRL', 'SURL' 6/28/66
*

```

* READ AND SET RELABELING

*

RORL LDA RL1,2
LDB RL2,2
LDX SS03
BRU XPOP
RJRL LDA UPRRL1
LDB UPRRL2
LDX SS03
BRU XPOP
STRL BRM SCRL
BRU TRAP
LDA SS01
LDB SS02
STA RL1,2
STB RL2,2
STRL1 BRM CHRL; BRU SS00; BRU POPX
SJRL BRM SCRL
BRU TRAP
LDA SS01
LDB SS02
STA UPRRL1
STB UPRRL2
BRU POPX
SPMTB SKG =77B; SKG =NCMEM; BRM TRAPX
AXC; SKE* PMTJOB; MIN 0; BRU POPX

*

SCRL ZRO
CLX
BRM UPRL
CLA
LDX PACPTR
SKN PQU,2
LDA =100000B
STA STRL4
LDX =-10
STRL3 LDA SRTE,2
BRM CRTA
BRR SCRL
CBX
SKA STRL4
BRR SCRL
BRX STRL3
MIN SCRL
LDX PACPTR
BRR SCRL
STRL4 ZRO
RDUSSR LDA USSRL1; LDB USSRL2; BRU XPOP
RUPAR BRM GATT; LDA UEXFLG; LDB WERIS,2
LDX WERISB,2; BRU XPOP

*

*

* TRAP ROUTINES


```

*
PPAN   LDA     0
      CLB
      BRU     PTRAP
* SKIP IF EXFLAG SET
SKXEC  SKN     UEXFLG
      BRU     POPX
      MIN     0
      BRU     POPX
* ILLEGAL INSTRUCTION TRAP
TRAPI  ZRO
      STA     SS01
      STB     SS02
      STX     SS03
      LDA     TRAPI
      LDB     =1
*GENERAL TRAP LOGIC
PTRAP  LDX     PACPTR
      SKG     =-1
      BRU     *+2
      LDA     0
      ETR     =50037777B
      STA     PL,2
      LDA     SS01
      STA     PA,2
      LDA     SS02
      STA     PB,2
      LDA     SS03
      STA     PX,2
      BRM     RFX
      BRM     TFX
      BRU     PACGO
*ILLEGAL SYSPOP EXIT
TRAPS  ZRO
      BRU     TRAP
      IF -1
NRTRAP BRM     MSX
      ENDF
TRAPX  ZRO 0   REPLACE BRU TRAP WITH BRM TRAPX FOR DEBUGGING
TRAP   LDA     0
      LDB     =1
      BRU     PTRAP
* NON-RESIDENT RETURNS
      IF -1
NRPOPX BRM     MSX
      BRU     POPX
NRXPOP BRM     MSX
      BRU     XPOP
      ENDF
*READ-ONLY TRAP
*MEMORY TRAP
TRAPM  ZRO
      STA     TX01
      STB     TX02

```

```

STX      TX03
LDA      TRAPM
ETR      =50037777B
STA      TX00
BRM      CAE
MRG      =400000B
LDX      PACPTR
LDB      =-1000000B+100B
BRM      MGET
BRU TX07; SKN TX00; BRU TX06
LDX PACPTR; BRM CHRL; BRU **2
BRU TX08; LDX PACPTR; BRM CHRL; BRU **2
BRU TX08; BRM TX05
LDB PACDMB; LDX =QSQ; BRU POPDMS
TX07    BRM TX05; BRU MTRAP
TX06    LDX PACPTR; BRM CHRL; BRU *-2
TX08    LDA      TX01
        LDB      TX02
        LDX      TX03
        BRI      TX00
TX05    ZR0; LDA      TX01
        STA      SS01
        LDA      TX02
        STA      SS02
        LDA      TX03
        STA      SS03
        LDA      TX00
        SKA      X4
        STA      0
        BRR TX05
SMTRAP  LDX      PACPTR
        LDA      =10000000B
        BRM      IIR
        BRU      MTPAN
        BRU      POPINT
MTPAN   LDA      0
        LDB      =2
        BRU      PTRAP
* COMPUTE EFFECTIVE OUT-OF-BOUNDS ADDRESS
*      TIME = 114 + N CY
SCAE    ZR0
        STA      CAE1
        STX      CAE3
CAE11   BRM      CEX
        LDA*     CAE1
        BRU      CAE4
* ADDRESS FOUND
CAE5    LDA      CAE1
        BRR      CAE
CAE4    LDA      CEX3
        STA      CAE2
        SKA      X1
        BRU      CAE6
        LDA      IABIT

```

ADM CAE1
LDX CAE3
BRM CEX
EAX* CAE1
BRU CAE7

* INDIRECT ADDRESS CHAIN OUT-OF-BOUNDS

LDA =-40000B

ADM CAE1
BRU CAE9

CAE8 LDA CAE1
ETR X4

MRG CEX3

ETR =6003777B

STA CAE1

CAE9 LDX CAE3

BRM CEX

LDA* CAE1

BRU CAE8

BRU CAE5

* POP CAUSED TROUBLE

CAE6 LDA X4

BRR CAE

* CHECK FOR EXU- OTHERWISE X CONTAINS BAD ADDRESS

CAE7 CXA

ETR =4003777B

STA CAE1

LDA CAE2

EOR EXUW

SKA =17700000B

BRU CAE10

LDX CAE3

BRU CAE11

CAE10 MIN CAE

BRU CAE5

CAE1 ZRO

CAE2 ZRO

CAE3 ZRO

EXUW EXU 0

IABIT ZRO* 0

* EXECUTE NEXT INSTRUCTION AND SKIP IF OUT-OF-BOUNDS-

* TIME = 31 + N CY

CEX ZRO

MIN CEX

STA CEX1

LDA CEX2

XMA 41B

STA CEX3

EXU* CEX

CEX4 XMA CEX3

STA 41B

LDA CEX1

BRR CEX

* OUT-OF BOUNDS IF WE COME HERE

CEX2 BRU *+1
MIN CEX
BRU CEX4

CEX1 ZRO
CEX3 ZRO

* SIMULATE MEMORY TRAP

MPAN LDA PQU,2
EOR X6
SKA X6
BRU TRAP
LDA PPTR,2
MRG PLMSK
CAX
LDA SS01
LDB =-1000000B+100B
BRM MGET
BRU *+2
BRU SS00
LDA SS01
STA UPL
BRU MTRAP

TX00 ZRO
TX01 ZRO
TX02 ZRO
TX03 ZRO

*

* FORK LOGIC

*

* FIND HIGHEST FORK IN STRUCTURE

EFK ZRO 0
LDA PPTR,2
MRG PLMSK
CAB
ETR PRMSK
SKG =0
BRR EFK
CBX
BRU EFK+1

* GET FORK ENTRY

GFK ZRO
LDA FPLST
SKG =0
BRR GFK
SUB =PPTR
COPY AX,A,B
XMA PPTR,2
STA FPLST

```

MIN      GFK
BRR      GFK
* PUT NEW FORK ON QIO
* SET PDOWN(OLD)=NEW, PDOWN(NEW)=0
*      PFORK(NEW)=OLD, PPAR(NEW)=PDOWN(OLD)
STFK     ZRO
         CXA
         STA      FK04
         LDX      =QTI
         BRM      QPUT
         LDX      PACPTR
         LSH      12
         ETR      PLMSK
         XMA      PPTR,2
         CAB
         ETR      PRMSK
         ADM      PPTR,2
         LDA      FK04
         XXA
         ETR      PRMSK
         STA      PPTR,2
         LDA      PACDMB
         STA      PTEST,2
         LDX      PACPTR
         CLA
         RSH      3
         LDA      QUTAB
         LSH      15
         BRR      STFK
* DELETE PAC ENTRY WHOSE PACT PTR IS GIVEN IN X
DFK      ZRO
         STX      DF01
         LDA      =700000B
         BRM      QSCH
* REMOVED FROM QUEUE IF DISMISSED
         LDA      PPTR,2
         SKA      PRMSK
         BRU      *+2
         BRR      DFK          (EXEC NOT DELETED)
         MRG      PLMSK
         COPY     AB,XA
         LDX      UTTY
* PFORK IN B, PACT PTR IN A
         SKE      TTYASG,2
         BRU      *+2
         STB      TTYASG,2    (PROPAGATE RUBOUT 'UP')
* PUT PFORK IN X, PDOWN(PFORK) IN A
         CBX
         LDA      PPTR,2
         RSH      12
         SKE      DF01
         BRU      DF02
         COPY     XB,AX
         LDA      PQU,2

```

```

        SKA      PRMSK
        BRU      DF03
* LAST FORK ON PDOWN TO BE CLEARED- PUT PFORK IN X, PACT PTR IN B
        XXB
        LDA      PPTR,2
        ETR      PRMSK
        STA      PPTR,2
        COPY     BX,BA
* PUT PACT ENTRY ON FPLST
DF04   ADD      =PPTR
        XMA      FPLST
        STA      PPTR,2
        BRR      DFK
* MOVE PDOWN OVER
DF03   COPY     BX,B          (NEW PDOWN IN A,PFORK IN X)
        LCY      12
        XMA      PPTR,2
        ETR      PRMSK
        ADM      PPTR,2
DF05   LDX      DF01
        CXA
        BRU      DF04
* SEARCH PPAR THEN REMOVE
DF02   CAX
        LDA      PQU,2
        MRG      PLMSK
        SKE      DF01
        BRU      DF02
        COPY     XB,AX
        LDA      PQU,2
        ETR      PRMSK
        CBX
        XMA      PQU,2
        ETR      PLMSK
        ADM      PQU,2
        BRU      DF05
DF01   ZRO
* SEARCH SUBSIDIARY FORK STRUCTURE FOR SPECIFIED FORK
*     SKIP IF PRESENT, NO SKIP IF TERMINATED
SHFK   ZRO
        LDA      PPTR,2
        SKA      PLMSK
        BRU      **+2
        BRR      SHFK
        RSH      12
SHF01  CAX
        LDA      PTAB,2
        EOR      SS01
        SKA      ADMSK
        BRU      **+3
        MIN      SHFK
        BRR      SHFK
        LDA      PQU,2
        SKA      PRMSK

```

```

BRU      *+2
BRR      SHFK
MRG      PLMSK
BRU      SHF01
* SCAN FORK STRUCTURE ANF OPERATE
*      A= OPERATION
*      X= EXIT
*      THE OPERATION SPEC. BY A IS CALLED WITH CORRECT PACPTR IN X
SCFK     ZRO
        STA      SCF01
        STB      SCF02
        STX      SCF03
* SCAN TO LOCAL 'BOTTOM'
SCF06    SKN      PPTR,2
        BRU      SCF07
        LDA      PPTR,2
        RSH      12
        CAX
        BRU      SCF06
SCF07    CXA
        SKE      SCF03
        BRU      SCF05
* PERFORM OP AND EXIT
        BRM*     SCF01
        LDB      SCF02
        LDX      SCF03
        BRR      SCFK
SCF05    LDA      PQU,2
        SKA      PRMSK
        BRU      SCF08
* PERFORM OP AND GO 'UP'
        LDA      PPTR,2
        MRG      PLMSK
        STA      SCF04
        BRM*     SCF01
        LDX      SCF04
        BRU      SCF07
* PERFORM OP AND GO 'PARALLEL'
SCF08    MRG      PLMSK
        STA      SCF04
        BRM*     SCF01
        LDX      SCF04
        BRU      SCF06
SCF01    ZRO
SCF02    ZRO
SCF03    ZRO
SCF04    ZRO
* TERMINATE FORK STRUCTURE
TFK      ZRO
        LDA      PPTR,2
        STA      TF01
        LDA      PTAB,2
        STA      TF06
        LDA      =DFK

```

```

BRM      SCFK
LDA      TF01
SKA      PRMSK
BRU      TF02
* EXEC TOP-LEVEL PANIC
LDA      PACDMB
STA      PTEST,2
LDA      EXECL6,4
STA      PX,2
LDA      TFC01
STA      PL,2
LDA      PQU,2
ETR      =47777777B
STA      PQU,2
BRU      TF09

```

* ORDINARY PANIC

```

TF02 MRG      PLMSK
      COPY    AX,AB
      LDA     PTEST,2
      EOR     =7000000B
      SKA     =77000000B
      BRU     TF03
      CAX
      BRU     *,2
      BRU     TF03
      BRU     TF04
      BRU     TF05
      BRU     TF03
      BRU     TF04

```

```

(ACTIVE)
(BRS31)
(BRS 106)
(PERFORMING EXEC BRS)
(BRS 109)

```

* CAUSE INTERRUPT IF ARMED

```

TF03 LDA      =4000000B
      CBX
      BRM     IIR
      NOP
      BRR     TFK

```

* BRS 106

```

TF05 CBX
      MIN     PL,2
TF08 LDA      PACDMB
      CBX
      STA     PTEST,2
      MIN     PL,2
      LDA     TF06
      ETR     ADMSK
      STA     PA,2
      LDA     SCF02
      STA     PX,2
TF09 CXA
      STX     TF06
      LDX     =QIO
      BRM     QPUT
      LDX     TF06
      BRR     TFK

```

* BRS 31

TF04	CBX		
	LDA	TF06	
	EOR	PA,2	
	SKA	ADMSK	
	BRU	TF03	
	BRU	TF03	
TF01	ZRO		
TF06	ZRO		
TFC01	ZRO	EXECPT,4	
* READ	FORK	STATUS	
RFK	ZRO		
	LDA	PQU,2	
	EOR	X6	
	SKA	X6	
	BRU	RF12	
	LDA	PPTR,2	
	MRG	PLMSK	
	CAX		
RF12	LDA	=RF06	
	BRM	SCFK	
	LDX	PUPAC	
	BRM	CHRL	
	BRU	*-2	
	LDB	SCF02	
	LDX	SCF03	
	BRR	RFK	
RF06	ZRO		
	STX	RF08	
	LDX	JOB	
	LDX	RL3,2	
	CLAB		
	BRM	SWAP	
	BRU	*-1	(CHANG)
	BRM	LABEL	
	LDX	RF08	
	LDA	PL,2	
	SKA	X4	
	BRU	*+2	
	LDA	SBRST	
	STA	RF14	
	CXA		
	EOR	SCF03	
	SKA	ADMSK	
	BRU	RF10	
	LDB	SCF02	
	SKB	X4	
	BRU	*+2	
	BRU	RF07	
RF10	LDA	PTEST,2	
	LDB	=-1	
	SKE	PACDMB	
	LDB	=-2	
RF07	LDA	PQU,2	
	EOR	X6	

	SKA	X6	
	BRU	*+3	
	CXA		
	BRU	RF11	(EXEC. BRS)
	LDA	PPTR,2	
	SKA	PRMSK	
	BRU	*+2	
	CXA	(TOP-LEVEL EXEC)	
RF11	MRG	PLMSK	
	STA	RF13	
	STB	FK04	
	LDA	PTAB,2	
	ETR	ADMSK	
	MRG	X4	
	STA	RF08	
	CXA		
	ADD	RFC01	
	STA	RF09	
	LDA	RF08	
	ETR	=34000B	
	LRSR	11	
	MUL	=3	
	LDX	RF13	
	LDA	RL1,2	
	LDX	RL2,2	
	XXB		
	RCY	18	
	LCY	0,2	
	ETR	=77B	
	SKE =0;	BRU *+2;	BRR RF06
	CLB		
	RCY	0,2	
	LCY	18	
	CLX		
	BRM	SWAP	
	BRU	*-1	(CHANG)
	BRM	LABEL	
	LDA	RF14	
	STA*	RF08	
	MIN	RF08	
	LDX	=-5	
RF09	LDA	PL6,2	
	STA*	RF08	
	MIN	RF08	
	BRX	RF09	
	LDA	FK04	
	STA*	RF08	
	BRR	RF06	
RF08	ZRO		
RF13	ZRO		
RF14	ZRO		
RFC01	LDA	PL6,2	
* START SUBSIDIARY FORK			
FKST	ETR	=3777B	

```

ADD      =6
SKA      =40000B
BRU      TRAP          (PANIC TABLE OVERLAP)
LDA      SS01
BRM      SHFK
BRU      *+2
BRU      TRAP          (TRIED TO RESTART SAME FORK)
BRM      GFK
BRU      FKSTW        (DISMISS UNTIL PACT SLOT FREE)
BRM      STFK
LDB      SS01
SKB      X4
SKN      PQU,2
BRU      *+2
MRG      X4
LDX      FK04
STA      PQU,2
CLA
SKB      =4000000B
MRG      X4
STA      PIM,2
CBA
ETR      ADMSK
LRSH     15
LDA      JOB
LSH      15
SKB      =2000000B
MRG      X4
STA      PTAB,2

```

* PROPAGATE TTYASG DOWN IF CALLED FOR

```

LDA      UTTY
SKG      =-1
BRU      FK07A
CAX
LDA      FK04
SKB      X1
STA      TTYASG,2
FK07A   LDX      PACPTR
LDA      SS01
SKA      X2
BRU      FK08
FK07    LDA      RL1,2
LDB      RL2,2

```

* STORE RELABELLING, INITIAL A,B,X,L

```

FK09    LDX      FK04
STA      RL1,2
STB      RL2,2
LDX      SS01
LDA      2,6
LDB      3,6
LDX      FK04
STA      PB,2
STB      PX,2
LDX      SS01

```

```
LDA      =-1
STA      6,6      (SET STATUS WD TO RUNNING)
LDB      1,6
LDA      0,6
ETR      =5003777B
SKN      0
BRU      *+2
MRG      X4
LDX      FK04
STA      PL,2
STB      PA,2
BRU      POPX
```

* PICK UP RELABELLING FROM PANIC TABLE

```
FK03     CAX
LDA      4,6
LDB      5,6
BRM      SCRL
BRU      FK03B
LDX      SS01
LDA      4,6
LDB      5,6
BRU      FK09
FK03B    LDX      FK04
BRM      DFK
BRU      TRAP
```

* DISMISS UNTIL PACT SLOT IS RELEASED

```
FKSTW    LDB      =FPLST
BRU      IOQDMS
```

* WAIT FOR SPECIFIED FORK TO TERMINATE

```
FKWT     BRM      SHFK
BRU      FK01
LDB      =700002B
BRU      POPST
```

* READ STATUS OF SPECIFIED FORK

```
FKRD     BRM      SHFK
BRU      FK01
LDB      =-1
BRM      RFK
FK01     LDX      SS01
LDA      6,6
STA      SS03
BRU      POPX
```

* TERMINATE SPECIFIED FORK

```
FKTM     BRM      SHFK
BRU      FK01
LDB      =-1
BRM      RFK
BRM      TFK
BRU      FK01
```

* WAIT FOR ANY FORK TO TERMINATE

```
FKWA     SKN      PPTR,2
BRU      POPX
LDB      =700003B
BRU      POPST
```

* TERMINATE ALL SUBSIDIARY FORKS

```
FKTA SKN PPTR,2
      BRU POPX
      LDB =FK02
FK03 STB FK04
      LDA PPTR,2
      RSH 12
FK05 MRG PLMSK
      CAX
      LDB --1
      BRM* FK04
      LDA PQU,2
      SKA PRMSK
      BRU FK05
      BRU POPX
FK02 ZRO
      BRM RFK
      BRM TFK
      BRR FK02
```

* READ STATUS OF ALL SUBSIDIARY FORKS

```
FKRA SKN PPTR,2
      BRU POPX
      LDB =RFK
      BRU FK03
FK04 ZRO
```

* PROGRAMMED INTERRUPT LOGIC

```
SRIR LDA PIM,2
      ETR =3777777B
      LDX SS03
      BRU XPOP
SIIR SKG =20
      SKG =4
      BRU TRAP
      CAX
      LDA =4000000B
      RSH 0,2
      LDX PACPTR
      BRM SIR
      BRU POPX
      CXA; SKE PACPTR; BRU POPX; BRU POPINT
SAIR DIR; EOR PIM,2; ETR =3777777B; EOR PIM,2; EIR
      STA PIM,2
      BRU POPX
      END
```