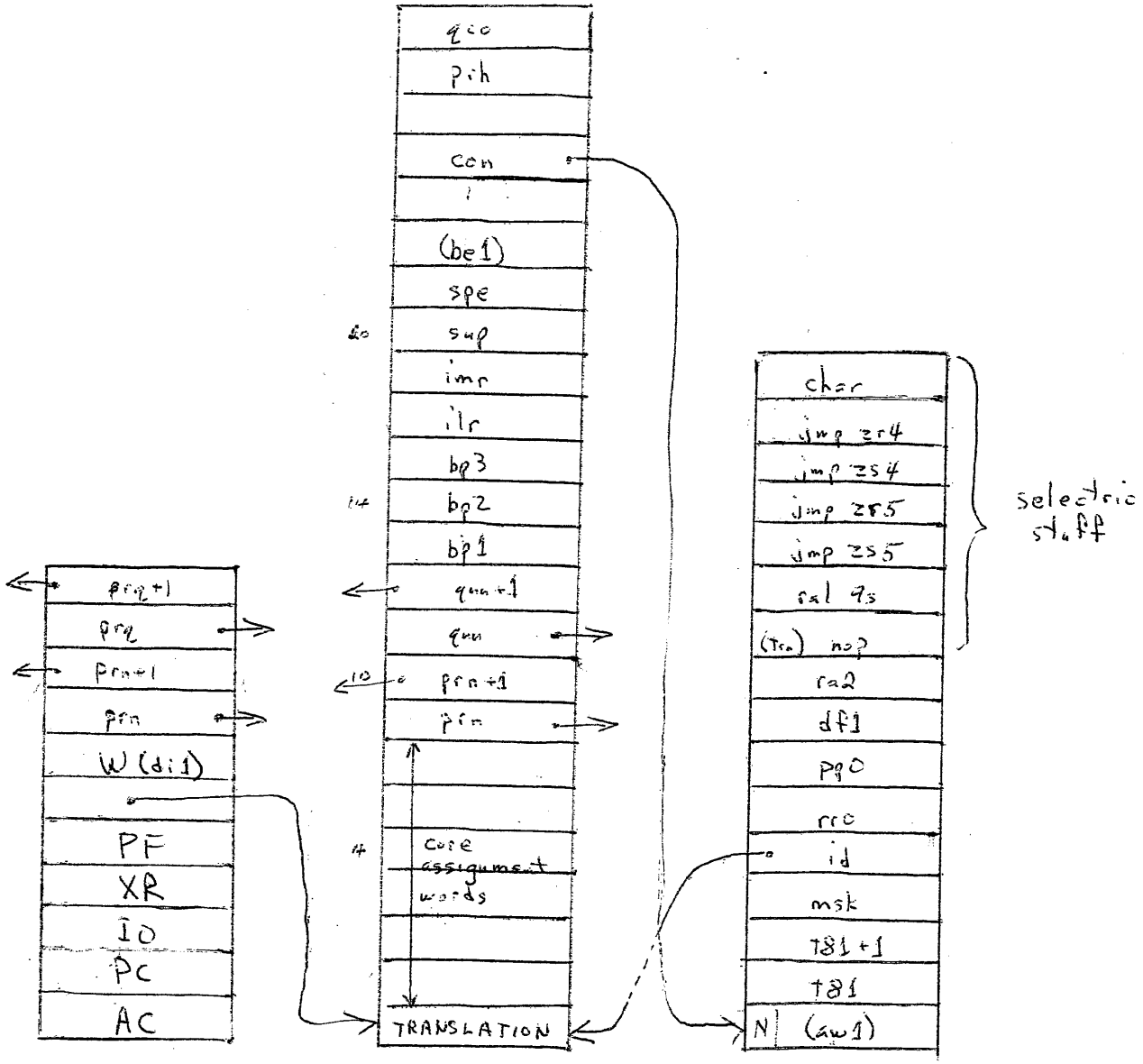

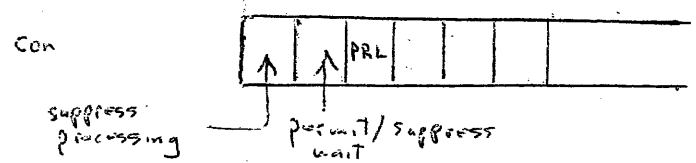

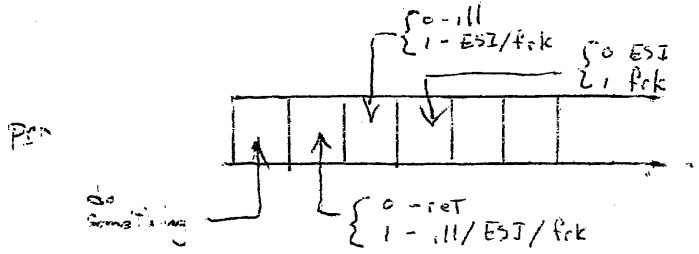

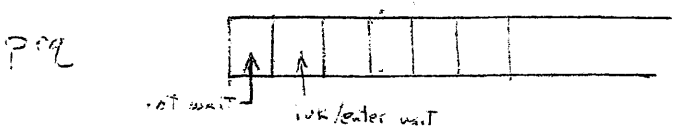

prn+1 = 0 ⇒ process deleted while in instruction wait

```
executive routine  17 feb 1970

psf=iot 0077          psn=iot 1077          sps=iot 3077
bef=iot 0177          ben=iot 1177          sbe=iot 3177
bff=iot 0277          bfn=iot 1277          sbf=iot 3277
rsf=iot 4177          rsn=iot 4077          srs=iot 4277
usf=iot 5777          usn=iot 5677          srw=iot 2677
spn=iot 1477          scn=iot 1577          lar=iot 0677
rpn=iot 0477          rcn=iot 0577          sti=iot 3377
lbe=iot 1377          rbe=iot 3777          sbr=iot 2577
rsb=iot 2077          sdl=iot 3477          siw=iot 3577
lqn=iot 4377          soq=iot 4477          sei=iot 2777

rpp=770000
rcp=770002 rqp=770003
rfa=770007
lpp=770010 cqt=770011
scp=770012 sqp=770013
ubn=770020 ubs=770021
ubf=770022
rin=770030 rfn=770031
ioc=770032 lcr=770037

lok=770040 ulk=770041
sfa=770045

ncb_=12     /size of typewriter buffer
ewv_=5      /restart level

npb_=140    /punch buffer size
pwm_=30     /restart level

rwm_=40     /reader restaqt level
nuf_=20     /number of user fields
ntl_=3      /number of user entries

/process words

di1=6       /dia word
prn=7       /process ring
prq=11      /process queue

cms=6630    /origin of computation blocks

/console words

aw1=0       /assignment word
t81=1       /1 and 2 are translator variables
msk=3       /console mask
id=4
rr0=5       /reader switch
pp0=6       /punch switch
df1=7       /drum field table
ra2=10      /selectric switch
trn=11      /7 words of typewriter junk
```

```
/computation words

quu=11      /computation queue (two words)
bp1=13      /location of breakpoint
bp2=bp1+1   /proceed count
bp3=bp2+1   /instruction under breakpoint
ilr=bp3+1   /illegal instruction return
imr=ilr+1   /illegal memory reference return
sup=imr+1   /superior sphere
spe=sup+1   /fault entry to superior
be1=spe+1   /break enable
con=be1+2   /pointer back to console
prh=con+2   /process hoard
qco=prh+1   /quantum count

define console n,r1,m
pb'n,       n'04000
            0
            0
            m
            0
            jmp ill
            jmp ill
            dd'n+nuf
            skp r1*i
            repeat 1-r1,[nop     ral 9s
            jmp zs5   jmp zr5
            jmp zs4   jmp zr4   0]
terminate
```

```
0/          eem       /initial entry
            lat
            sad (2
            dac nuc
            iam
            cli
            lcr
            dia
            lio (210000          /adm. rt. on field 21, loc. 3200
            law 4600
            dcc
            hlt
            lxr (-100
            dzm i 100
            SXXP
            jmp .-2
            lem
            law cms-[cms-end]>13*13
            dac t
            lio frp
            aam
            dio t
            dac frp
            law 13
            adm t
            sas (cms-54          /leave last four out
            jmp .-7
            law 5000
sut,        lia
            lar
            scn
            ben
            bff
            psf
            spn
            ben
            bff
            add (xct
            sas (5001
            jmp sut
            rsf
            usf
            ioc
            law 7400
            ivk 121    /initialize microtape address
            lxr (1
            dzm i 0
            SXXA
            sas (.-1
            jmp .-3
            jmp 131
            constants

74/
            340000+qqt-prq
unt=ivk .   jmp 20
dat=ivk .   jmp 21
mot=ivk .   jmp 77
```

```
100,    jmp tot   /0 - interrupt
        jmp dsp   /1 - iot
        jsp trp   /2 - illegal
        jsp trp   /3 - lock fault
        jmp .     /4 - function tardy
        jsp trp   /5 - function busy
        jmp str   /6 - function started
        jsp trp   /7 - hlt
        jmp adf   /10 - extend snag
        jmp bp    /11 - bpt
        jmp xe0   /12 - esi
        jmp ivw   /13 - ill ivk
        jmp pre   /14 - preempt
        jmp rbn   /15 - rnd rbn
        jmp fr1   /16 - frk
        jmp qt1   /17 - qit
        jmp atm   /20 - meta
        jmp ivt   /21 - enter
        jmp ivt   /22 - ivk
        jmp adx   /23 - index snag
        jmp adf+2 /24 - last snag
125,    /four words of space
131/    ioc
        ivk 120
beg,    lac (xor i tot+3
        dac 141
        lac (add
        dac 144
        ubn
140,    repeat 6,0
        jmp 147
        cli       /system death
        lar
        szs 70
        jmp .
        lio (6500
        dia
        lio (250700
        lac (77100
        dcc
        hlt
        jmp 7777

dd2,    repeat nuf,0
dd3,    repeat nuf,0
dd4,    repeat nuf,0
dd5,    repeat nuf,0
dd6,    repeat nuf,0
```

```
sd,         lac
            repeat 17,0
            lac
            lac
            0
            lac
            lac
            0

wf,         10000

            0           0

pmt,        76
csi,        0           /pseudo console switches
onn,        0           /consoles logged in
            jmp ill     /constant
uc,         arc         repeat 3,0              /absolute core words
            sr0         /constants for adm rt
            ntb
            ubs
frp,        0           /free process pool
            qqt
            bop
ctb,        pb2
            pb3
            pb4
            pb5
            pb6
            srr
            mtt
bc,         0           repeat 3,1000           /core flags
nuc,        3
            rs1
cpp,        cpp-prq     cpp-prq     /process chain

/programmed queues

qqt,        -1          .-prq-0     /queue for microtapes
            repeat 4,.-prq          .-prq-1
```

```
dsp,        add (d7    /iot trap
            dap .+10
            lxr cmp
            law 7777
            and i con
            sza i
            jmp ill
            dac t1
            lxr prc
            jmp .

ill,        lxr cmp    /recoverable illegal instruction
            lio i ilr
            TIA>P
            jmp 102
ill+4,      lxr prc
            ral 3s
            rcr 3s
            lio i 1
            rcl 3s
            rar 3s
            rir 3s
            dio i di1
            dac i 1
            ubn

b,
b ncb*5 npb/
erb=b+200

/dispatch table for iot traps

d7,         jmp .      /old break 0
            jmp .      /old break 1
            jmp wa     /wat
            jmp ra     /rpa
            jmp rb     /rpb
            jmp to     /tyo
            jmp ti     /tyi
            jmp pa     /ppa
            jmp pb     /ppb
            jmp di     /dia
            jmp ill    /dba
            jmp dc     /dcc
            jmp da     /dra
            jmp .      /old break 15
            jmp ar     /arq
            jmp ill    /iot 2377
            jmp rr     /rrb
```

```
str,        rfn
            law 77
            A←IA
            sas (1
            sad (2
            add (13     /drum
            sub (14
            TAAX>P
            jmp .       /wrong device
            sub (11
            sma
            jmp .       /wrong device
            law 7777
            and i iow
            sza
            jmp .       /process already hung
            lac prc
            dap i iow
hng,        lxr prc     /hang this process
            lac (200000
            dip i prq
            jmp wa0

iow,        0           /drum (1)
            0           /drum (2)
            050000      /ttyin (16)
            050000      /ttyout (17)
            030000 mtp
            030000
            060000      /crock (22)
            060000      /kludge (23)
            060000      /lossage (24)

ntb=.-1     101         /adm rt
            c7e         /entry for core 7 stuff
            repeat ntl-2,0
ntb+ntl+1,
            arc
            exc
            repeat ntl-2,0
```

```
atm,        rfa         /meta processor
            lxr (-070000
            X+IX
            lac i 0
            rar 3s
            and i 0
            and (77
            dac t
atm+10,     sub (mtz-mtb
            sma
            jmp mt9
            add (mtz
            dap mtc
            lxr prc
            lac i 0
            lio i 2
mtc,        xct .
            dio i 2
rta,        lxr prc     /return value to AC
            dac i 0
            jmp ret

mtb,        dac i di1   /mta 000 - AC to drum address
            dio i di1   /mta 001 - IO to drum address
            lac i di1   /mta 002 - drum address to AC
            lio i di1   /mta 003 - drum address to IO
            jmp atl     /mta 004
            jmp atl     /mta 005
            jmp atl     /mta 006
            jmp atl     /mta 007
            jmp fr2     /770170 - temporary fork
            jmp rnj     /770171 - wait for switch change
            jmp ill     /770172
            jmp ill     /770173
            jmp rdd     /mta 104 - read drum
            jmp rdd     /mta 105 - read drum
mtz,

mt9,        law 1
            dac t6
            jmp ntr
```

```
atl,        law 4
            add t
            jmp spr

/wait for switch change (770171)

rnj,        law arl
            sas prc
            jmp ill    /probably Plummer
            lac tsb
            and pmt
            sad csi
            jmp rnk
            dac csi
            dac i 0
            jmp ret
rnk,        lac rnk
            dac i prq
            jmp wa0

rdd,        lac i di1 /read drum (770174, 771175)
            dac t
            dio t2
            cla
            jmp dc1
```

```
qt1,        lac i prn /quit
            lxr i prn+1
            dap i prn
            X→AX
            dap i prn+1
            lxr cmp
            lio i prh
            spi
            idx i prh /decrease debt
qt2-1,      lxr (cpp-prq        /check process chain
qt2,        law 7777
            and i prq
            sad (cpp-prq
            jmp qt9
            dac t
            X→AX
            dac t1
            law i 7777          /find sphere for which process is needed
            and i prn
            sza i
            jmp qt5    /wants to fork
            ral 6s     /wants to enter
            sas (1
            jmp .+4
            lxr i 5    /wants to enter superior
            lac i sup
            jmp .+3
            TAX
            lac i ntb+ntl-1
            and (7777
            skp i
qt5,        lac i 5
            spi i      /AC = sphere
            sad cmp
            jmp qt6    /found a deserving one
            lxr t
            jmp qt2
qt6,        dac cmq    /sphere to which process will be given
            TAX
            law i 1
            spi
            adm i prh /increase debt
            lxr t     /unlink from chain
            law 7777
            and i prq
            lxr t1
            dap i prq
            TXI
            sad (cpp-prq
            dio cpp+1 /process being removed is last
            lxr t
            law i 7777
            and i prn
            sza
            jmp ntw    /enter
            lio i di1 /fork
            lac (740000
            dip i prn
            lac prc
```

```
            dac i di1
            X→AX
            dio i di1
            jda acp
            jmp wa0
/warning - do not allow di1 or PC to change

ntw,        ral 6s      /restart enter
            sub (1
            dac t6
            lac prc
            dac t7      /new proc
            lac t
            dac prc     /old proc
            lac i prq+1
            dac t
            jmp nty

qt9,        lxr cmp     /return it to hoard
            spi
            lxr (frp-prh          /or pool
            lio i prh
            lac prc
            aam
            dio prc
            dac i prh
            jmp wa0
```

```
rbn,        TXXP|       /`round robin` trap
            jmp wa0
            law wa0

rpc,        rcp         /put process at end of queue
            dap rpx     /process in XR, priority in CP or IO
            ril 1s
            law pqu-prq
            A+II
rpc+5,      X→IX
            lac i prq+1
            dio i prq+1
            X→IX
            dio i prq
            dac i prq+1
            X→AX
            dap i prq
rpx,        jmp .

pre,        TXXP|       /preempt trap
            jmp wa0
            rcp
            ril 1s
            lax pqu-prq
            A+II
            X→IX
            lac i prq
            dio i prq
            X→IX
            dio i prq+1
            dap i prq
            X→AX
            dap i prq+1
            jmp wa0
```

```
fr2,        stf 6       /temporary fork (mta 100)

fr1,        lxr cmp
            lio i prh
            TII_<
            jmp .+7     /hoard is not empty
            lio frp     /hoard empty, check pool
            TIIAP|
            jmp fr8-2   /lose
            law i 1
            adm i prh   /increase debt
            lxr (frp-prh
            aam         /unlink
            lac i prh
            dac i prh
            dio t       /new process block
fr7,        rcp
            SIA         /demote old process
            sad (10
            TIA
            rqp
            swp
            AMI_<
            sqp
            lxr prc
            law 3       /crock for temp fork
            add i 1
            szf 6
            dap i 1
            jsp rpc     /put old process back on queue
            lxr prc     /old proc
            lio i prn
            lac t       /new proc
            dac i prn
            X→AX
            dac i prn+1
            dio i prn
            X→IX
            dio i prn+1
            TAX
            lac i 5
            lio i di1
            lxr t
            dac i 5
            dio i di1
            TXI
            dio prc
            ubf

/hang process until it gets another
/reason in AC

fr8-2,      szf 6
            jmp ret
fr8,        lxr prc
            dip i prn
            law cpp-prq
            dac i prq
            TXA
```

```
            lxr cpp+1
            dac cpp+1
            dap i prq
            jmp hng

/restart fork

fr6,        lxr prc
            lac i di1
            dac t
            X→AX
            lio i di1
            X→AX
            dio i di1
            jmp fr7
```

```
svc,        rin
            cla
            rcl 6s
            sas (1
            sad (2
            add (13     /drum
            sub (14
            TAAX>P
            jmp .       /wrong device
            sub (11
            TA<M
            jmp .       /wrong device
            lio i iow
            dap i iow
            rcl 6s
            dac pri
            rir 6s
            TIAP|
            jmp .       /no suspended process
            jda acp
            lxr acp
            lac (add
            dip i prn
            jmp rm3
```

```
/service io

srv,        dap sr1
            srw
srr,        skp         /skip if reader running
            jmp sr8
srr+2,      rrb
rip,        lac .
            ral 8s
            rcr 8s
            aam
            dac rip
            rpa-i
            idx rip
            sad (lac erb
            lac (lac b
            dac rip
            lio c1
            dio rrs     /buffer not empty
            sub rop
            sza i
            dio srr     /full, shut off reader
            sas (erb-b-rwm
            sad (-rwm
            rsn
            srw
            xct srr
            jmp sr6
            jmp srr+2

sr8,        srs i
rs1,        jmp .+1     /or rr9

sr0,        rpn
            sni i
            jmp sr5
            rcn
            sni
sr1,        jmp .
            ril 4s
            TIX
            sps
            jmp sr2
            sti
            jmp sr3
            jsp if0+1
            psf
sr2,        tyi
            jsp itf
            TXI         /restart both processes
            lxr (6
            I+XXA|
rct,        TXXAI       /restart a process
            dap rc2
            dio t4
            lac i bdc
            sza i
            jmp rc2
            jda rms
```

```
            law 6
            dac pri
            lac rms
            jea acp
rc2,        law .
            lxr t4
            A$XP
            jmp rct
            jmp sr4
sr3,        jsp ite
            tyo
            jmp sr4-2
sr5,        lxr (1
            jsp ite
            ppa
            sbf
            jmp rct
sr4,        idx sr1
            jmp sr1
sr6,        srs
            jmp sr4
            lxr (7      /reactivate for reader
            jmp rct
```

```
/index and test if buffer empty

ite,        dap ie7
            law 377
            aam
            and i bop
            lia
            idx i bop
            sad i bor+1
            lac i bor
            dac i bop
            sad i bew
            bff
            sad i bip
            ben
ie7,        jmp .

/index and test if buffer full

itf,        dap if7
            aam
            lac i bip
            rcr 8s
            ral 8s
            aam
            dac i bip
            bef
            idx i bip
            sad i bor+1
            lac i bor
            dac i bip
            sad i bop
            bfn
            idx i bew
            sad i bor+1
            lac i bor
            dac i bew
if7,        jmp .

/clear typewriter buffer

            law to3
if0,        psn
            dap if3
            bff
            lac i bip
            dac i bop
if3,        jmp .

/buffer pointer table

bop=.-1     z=0
            b+z         z=z+npb     /1  (punch)
            b+z         z=z+ncb     /2
            b+z         z=z+ncb     /3
            b+z         z=z+ncb     /4
            b+z         z=z+ncb     /5
            b+z         z=z+ncb     /6
```

```
bip=.-1     z=0
            b+z         z=z+npb    /1 (punch)
            b+z         z=z+ncb    /2
            b+z         z=z+ncb    /3
            b+z         z=z+ncb    /4
            b+z         z=z+ncb    /5
            b+z         z=z+ncb    /6

bew=.-1     z=0
            b+z+npb-pwm+1           z=z+npb     /1 (punch)
            b+z+ncb-ewv+1           z=z+ncb     /2
            b+z+ncb-ewv+1           z=z+ncb     /3
            b+z+ncb-ewv+1           z=z+ncb     /4
            b+z+ncb-ewv+1           z=z+ncb     /5
            b+z+ncb-ewv+1           z=z+ncb     /6

bor=.-1     z=0
            b+z         z=z+npb    /1 (punch)
            b+z         z=z+ncb    /2
            b+z         z=z+ncb    /3
            b+z         z=z+ncb    /4
            b+z         z=z+ncb    /5
            b+z         z=z+ncb    /6
            b+z

bdc=.-1     /IO deactivate table
            0           /1 (punch)
            0           /tyo 2
            0           /tyo 3
            0           /tyo 4
            0           /tyo 5
            0           /tyo 6
            0           /7 (reader)
            0           /tyi 2
            0           /tyi 3
            0           /tyi 4
            0           /tyi 5
            0           /tyi 6
```

```
/remove process from IO wait

rms,        0
            dap msx
            lxr rms
            lac i prq
            sma
            jmp msx    /not in IO wait
            and (7777
            TAAI>
            jmp rm4    /not in sbm chain
            lxr i prq+1          /remove from sbm chain
            dap i prq
            X→IX
            dio i prq+1
            lxr rms
rm4,        lac i prq
            dzm i prq
            ral 6s
            and (17
            TAXP
            dzm i bdc /remove from IO wait
msx,        jmp .
```

```
acp,        0           /activate process
            dap acx
            lxr acp
            cla
            sad i prn+1
            jmp ac3     /process has been abandoned
            dip i prq /turn off inactive flag
            lac i 5
            sas cm1
            sad (exc
            jmp ac2     /in core, run it directly
ac0,        TAAX        /enter here from enb
            lio i con
            spi
            jmp acx-2 /computation is stopped
            lio i quu
            sni i
            jmp acx-2
/put computation on queue
            dzm i qco /give it a new quantum
            dac cmm
            lio i quu+1
            sni
            lio (cqu-quu+12.
            law 7
            xor pri
            sza
            law i 3
            A+IA
            sad (cqu-quu-3
            law cqu-quu
            jda rpm
            lac qua
            CAAM
            dac qua     /terminate infinite quantum
            jmp acx-2

ac2,        lac pri     /process is in core
            rqp
            swp
            ANI_<
            sqp
            lxr acp
            jsp rpc+1
acx-2,      law 7
            dac pri
acx,        jmp .

ac3,        lac frp
            dac i 0
            TXA
            dac frp
            jmp acx-2

pri,        7
```

```
rb,         law rb1     /rpb
c2,         skp 600

ra,         law ra1     /rpa
            lxr t1
            xct i rr0
            nop
            dap rab
            rsf
            law sr0
            dap rs1
            xct i rr0
            jmp rr8

rr7,        jsp srv
            nop
            law 600
rrs,        skp 600     /skip if buffer empty
            jmp rop-1
            lxr (7      /normal entry
            siw i
            jmp dms
            law rr9
            dap rs1
            jmp ret

rr8,        rpa-i       /set up
            law i 3
            dac r00
            law 600
            dap i rr0
            dap rrs     /buffer empty
            dap srr     /reader running
            law b
            dap rip
            dap rop
            jmp rr7

roq,        idx sr1
            cla

            dap rs2
rop,        lac .
            dac t
            lio c2
            idx rop
            sad (lac erb
            lac (lac b
            dac rop
            sub rip
            sza i
            dio rrs     /buffer empty
            sas (erb-b-rwm
            sad (-rwm
            dio srr     /buffer nearly empty
            lio t
rab,        jmp .       /rpa-rpb switch
```

```
ra1,        cla+swp
            rcl 8s
            dio prb

res,        usn
            law sr0
            dap rs1
rs2,        skp
            jmp sr1

            lxr prc     /rpa complete
            siw i
ret-1,      dio i 2
ret,        lxr prc
            lac (400000
            dip i prn
            jmp rm3

rb1,        spi i
            jmp rb2
            lac prb
            ril 2s
            rcl 6s
            dac prb
            isp r00
            jmp rb2
            law i 3
            dac r00
            lio prb
            jmp res

rb2,        xct rs2
            jmp sr1
            jmp rrs-1

r00,        0           /rpb count

rr9,        xct rrs     /this is part of srv
            jmp roq
            rsf
            jmp sr0

prb,        0           /reader buffer

rr,         lxr t1      /rrb
            xct i rr0
            nop
            usf
            lio prb
            jmp rei
```

```
pb,         law 2       /ppb
            lio i 2
            rcl 6s
            jmp pa+1

pa,         lac i 2     /ppa
            lxr t1
            xct i pp0
            dac t
            spn
            lxr (1
            sbf i
            jmp dms
            lio t
            jsp itf
            jmp ret

ti,         lxr t1      /tyi
            xct i ra2
            jmp z3
ti+3,       scn
            ril 4s
            TIX
            sps
            sbe i
            jmp dms-2
            jsp ite
            lxr t1
            xct i ra2
            jmp z10
rei,        lxr prc     /return with IO
            jmp ret-1

to,         lio i 2     /tyo
            dio t
            idx t1
z25,        scn
            ril 4s
            TIIX
            dio t2
            sps
            jmp if0-1
            sbf i
            jmp dms
to3,        lxr t1
            xct i ra2-1
            jmp z50
            lio t
z51,        lxr t2
            jsp itf
            jmp ret
```

```
di,     law 1       /dia
        jmp atm+10          /simulate 770071

dc,     lio i di1 /dcc
        dio t       /write field
        lio i 2
        dio t2      /read field
        jsp trf
        dip t2
        lio t
        jsp trf
        jmp dc1
        xct tr7
        lxr t1
        and i msk
        sza i
        jmp ill
        jmp . 2
dc1,    dip t       /enter here from direct drum read
        lxr prc
        lac i 0
        dac t1
        sfa
        jmp adc     /not in core
        lio i 4
        ril 5s
        and (070000
        spi
        sza
        jmp dc2
        law 7700    /references PRL field
        and t1
        sza i
        jmp ill
        law 7777
        and t2
        sub (1
        TAH>P
        jmp ill
        law i 7777
        ior t1
        A+I_<
        jmp ill     /wraps around

dc2,    dra         /enter here from read/write sphere
        xct . 2
        lai
        sub t
        and (7777
        sub (7652
        and (-77
        sza
        jmp dc3
        spn
        scn
        lio t
        dia
        lio t2
        lac t1
```

```
            dcc
            jmp ret
skk,        law 1
            lxr prc
            add i 1
            dap i 1
            jmp ret

dc3,        jsp srv
c1,         skp
            jmp dc2
```

```
trf,        dap trx
            ril 1s
            cla
            rcl 5s
            sza i
            jmp trx
            rir 6s
            spi
            jmp abs
            sub (nuf
            sma
            jmp ill
            lxr t1
            add i df1
            dap . 1
tr7,        lac .
            and (700000
            sza
            jmp ill
            xct tr7
            and (77
            rar 6s
            sza i
            jmp ill
trx,        jmp .

abs,        sub (27
            sma
            jmp ret   /selection error
            add (sd 26
            dap trx-7
            idx trx
            jmp trx

da,         dra       /dra
            law 145
            A+IA
            and (7777
            dac i 2
            jmp ret
```

```
/entry from interrupts

tot,        sei
            jmp svc
            jsp srv
            jmp .+2
            jmp .-2

            rsb         /read switches and buttons
            lxr tsb
            X$IP|
            jmp bs0     /no change
            CXX
            dio tsb
            X←IA
            sar 7s
            and onn
            dac t0      /call buttons that have been pressed
            lac tsb
            and pmt
            sad csi
            jmp bs1
            lia         /switches have changed
            lac arl+prq
            sas rnk
            jmp bs1     /login process isn't hung on mta 101
            dio csi
            dio arl
            law 6
            dac pri
            lac (400000
            dip arl+prn
            law arl
            jda acp
            jmp bs1

console 2,0,40
console 3,0,20
console 4,1,10
console 5,1,4
console 6,1,2
```

```
bs1,    law ctb
        dac t6
        lac t0
        rar 6s
ub0,    and (-7777
        sza i
        jmp bs0
        dac t4
        sma
        jmp ubx
        lxr t6      /console hit call
        lac i 0
        TAX
        law 14
        dac t       /transmitted word
        law 7777
        and i id
        TAAX
        stf 1
        sad i prn
        jmp ntc
        lxr i prn
        lio i prq
        ril 1s
        law 40
        and i 4
        sza         /check ID's flag 1
        spi
        jmp ubx     /in enter, can't hit call
        dip i prn /clear process control flags
        law 102
        dac i 1
        TXA
        rir 1s
        spi i
        jmp .+4
        jda rms     /in iot wait
        lac rms
        jda acp
ubx,    idx t6
        lac t4
        ral 1s
        jmp ub0
```

```
bs0,        lac sbm    /check sbm chain
sb1,        sad (sbm-prq
            jmp rm1
            dac rms
            TAX
            rbe
            dio t1
            law 7777
            and i prq
            dac t0
            lxr i 5
            lio i be1
            lbe
            law 7777
            and i con
            TAIXP
            lio i aw1
            lar
            spn
            scn
            sbr
            jmp .+6
            jsp rms+1
            law 6
            dac pri
            lac rms
            jda acp
            lio t1
            lbe
            lac t0
            jmp sb1

rm1,        lxr cm1
            law 7777
            and i con
            TAIXP
            lio i aw1
            lar
            spn
            scn
            soq
            jmp rm3
pac,        lxr cm1
            TXXP|
            jmp pad
            lac qua
            TA>
            jmp pad-1 /computation had infinite quantum
            isp i qco
            jmp paf
            law 3
            add cpr
            sas (cqu-quu+15.
            dac cpr    /demote unless at bottom level
            jmp pad
paf,        law cqu-quu-3
            lio (3
            jmp .+3
            sas i quu
```

```
jmp pad
A+IAX
sas cpr
jmp .-4
lio (74    /start another quantum
lqn
jmp rm3
```

```
dms-2,      law 6
            A+XX
dms,        lac prc    /deactivate process, device number in XR
            lio i bdc
            sni i
            jmp 105    /function busy
            dac i bdc
            TXA|

wa,         cla        /deactivate, no IO device
            rar 6s
            ior (400000
            lxr prc
            dac i prq /reason for deactivation
            lac i 4
            and (160000
            sas (40000
            jmp wa0
            law sbm-prq
            dac i prq+1
            lac sbm
            dap i prq
            X→AX
            dac i prq+1
            dac sbm

/search process queue

wa0,        law cpp-prq          /check process chain
            lio frp
            sas cpp
            TIIXP|
            jmp w0a
            lac i 0
            dac frp
            dio prc
            cli↓cmi
            jmp qt2-1
w0a,        law pqu-prq-2
            lio (2
            A+IAX
            sad i prq
            jmp .-2
            sub (pqu-prq
            sar 1s
            sad (10
            jmp p5e    /queue is empty
            lia
            scp
            lac i prq
            dac prc
            X→AXI
            lpp
            lio i 5
            dio cmp
            lio i prq
            X→IX
            dap i prq+1
            X→AX
```

```
            dap i prq
            lio (2
            TXXA|
            A+IAX
            sad i prq
            jmp .-2
            sub (pqu-prq
            sar 1s
            lia
            sqp
rm3-2,      spn
            scn
rm3,        lac qua
            spa
            jmp pac    /end infinite quantum
            lac 0
            sza i
            idx cs1
            dac 0
            lxr prc
            TXXP|
            jmp wa0    /running process has disappeared
            lio i prn
            spi i
            ubn
            cla↓clf 7
            dip i prn
            ril 1s
            TIIA<M
            ubs
            A+IAI<M
            jmp ill
            A+I<M
            jmp xe1
            jmp fr6

p5e,        dzm prc    /process queue empty
            lio cm1
            sni↓szf 4 i
            jmp pad    /try another computation
            cli        /run hung process
            lar
            lqn
            lio (cs1
            lpp
            lio (10
            sqp
            scp
            dzm qua
            ubn

rmv,        dap pax    /remove computation, put on queue at level in cpr
            lxr prc
            TXXP
            jsp rpc    /remove running process
            dzm prc
            lxr cm1
            TXX|=
            jmp pax    /there is none
            clf 2
```

```
          rbe
          dio i be1

rmi,      law 7777  /remove all processes belonging to this computation
          and i prn /from process queue
          TAAX
          dac cmm
          sad cm1
          jmp pab   /done
          law i 7777
          and i prq
          sza
          jmp rml   /wasn't active
          lio i prq
          lxr i prq+1
          dio i prq
          X→IX
          dio i prq+1
          lxr cmm
          stf 2     /indicate active process found
          jmp rml

pab,      dzm cm1
          lac i con
          spa
          jmp .     /stopped?
          lac cpr
          dac i quu+1          /save priority
          szf 2 i
          jmp pag   /there were no active proc's
          jda rpm   /put on comp queue
          jmp pax
pag,      dzm i quu /enter here also from dsb
          lio i 0   /mark all cores inactive
          TIIM
pax,      jmp .     /done
          lac (700000
          rcl 3s
          sas (6
          sad (7
          jmp pax-1
          TAX
          dip i bc
          jmp pax-1
```

```
/place computation in XR, cmm on on queue at level in AC

rpm,        0
            dap pmx
            lio i qco
            lac rpm
            dac i quu
            dac i quu+1
            sni
            idx rpm    /to put at end of queue instead of front
            lxr rpm
            lac i quu
            lxr cmm
            sni
            jmp .+5
            dac i quu /put at front of queue
            X→AX
            dac i quu+1
            jmp .+4
            dac i quu+1          /put at end of queue
            X→AX
            dac i quu
            lxr rpm
            dac i quu
pmx,        jmp .
```

```
t0,         0
t1,         0
t2,         0
t3,         0
t4,         0
t5,         0
t6,         0

sbm,        sbm-prq     /seq. brk. deactivate chain
            sbm-prq

cs1,        525252      /hung process
            sub .+2
            dac i .
            dac cs1+2
            520052
            (667666

cmm,        0
cpr,        0
who,        0
qua,        0
cqu,        .-quu       .-quu-1     -1          /.13 sec
            .-quu       .-quu-1     -2          /.27
            .-quu       .-quu-1     -5          /.67
            .-quu       .-quu-1     -12.        /1.6
            .-quu       .-quu-1     -15.        /2.0

pqu,        repeat 10,.-prq     .-prq-1
cmp,        0           /current computation
cmq,        0
prc,        0           /current process
cm1,        0
```

```
pad-1,      dzm i qco
pad,        lac cm1
            dac who
            jsp rmv
            stf 4       /to indicate that computation search will happen
            law cqu-quu-3       /search computation queue
            lio (3
            A+IAX
            sad (cqu-quu+15.
            jmp wa0     /empty
            sad i quu
            jmp .-4
            dac cpr     /found one
            cli
            sas (cqu-quu+12.    /if at bottom, maybe infinite quantum
            lio (74
            lac i quu
            d+c cmq
            sas who
            lio (74
            dio qua
            TAX
            lio i be1
            lbe
            law 7777
            and i con
            TAIXP
            lio i aw1
            laq

/bring in core 0

            lxr cmq
            lac i 1
            TAP|
            jmp .       /does not exist
            lio i 0
            rcl 3s
            sas (6
            jmp p5b     /already in core
/select absolute core to use
            clc
            dac t0
            ZAIX
            lac i bc    /look for oldest inactive core
            AMI_>
            jmp .+3
            X→AI
            dac t0
            SWXA
            sas nuc
            jmp .-7
            lac t0
            TAAM
            jmp p5c+1   /found one
            ZX
            law 7
            and i uc
            X→AP
```

```
                jmp p5c     /not a core 0
                SXXA
                sas nuc
                jmp .-6
                law i 1
                add nuc
p5c,            dac t0      /absolute core
                dzm t4      /pseudo core = 0
                jsp bru
```

```
p5b,    lac cmq
        dac cm1
        lxr cm1    /remove it from comp queue
        lac i quu
        lxr i quu+1
        dac i quu
        X→AX
        dac i quu+1
        lxr cpr
        lio i quu+2
        lxr cm1
        lac i qco
        sza i
        dio i qco /give it a new quantum
        lio qua
        lqn
        lxr cm1    /put all active processes on process queue
p5f,    law 7777
        and i prn
        sad cm1
        jmp wa0    /done
        dac t3
        TAX
        law i 7777
        and i prq
        lio (7
        sza i
        jsp rpc+1 /put on proc queue if active
        lxr t3
        jmp p5f
```

```
/stop processing in a computation, remove IO waits
/computation in AC

stp,        0
            dap spx
            lac stp
            sad cm1
            jsp rmv    /is running
spx,        law .
            dap pax
            lxr stp
            lac i con
            spa
            jmp pax    /already stopped
            ior (400000
            dac i con
            lio i quu /remove from computation queue
            sni
            jmp .+5    /not active
            lxr i quu+1
            dio i quu
            X→IX
            dio i quu+1
            lxr stp
            dzm i quu+1          /crock for acp
            law 7777
            and i prn
            TAAX
            sad stp
            jmp pag    /clear quu, give cores low priority, exit
            jda rms    /remove each process from iot wait
            lxr rms
            jmp .-7
```

```
trp,    sub (103    /program trap

spr,    and (17     /start superior sphere
        dac t
        dzm t6
        lxr cmp
        lac i sup
        TAXP
        jmp .       /no superior
        jmp ntr
```

```
/resume processing in computation in AC. Must be stopped

ust,        0
            dap acx
            lxr ust
            lac i con
            sma
            jmp acx    /wasn't stopped
            and (377777
            dac i con /turn off stop bit
            law 7777  /check each process
            and i prn
            sad ust
            jmp acx    /done, no active proc
            TAX
            law i 7777
            and i prq
            sza
            jmp .-10   /not active
            law 6
            dac pri    /crock
            lac ust    /active proc found
            jmp ac0    /acp will put it on comp queue
```

```
bp,         rfa         /bpt
            lai
            lxr cmp
            sad i bp1
            isp i bp2
            jmp b3      /not primary,or count expired
            lac i bp3 /multiple proceed
            lxr (-070000
            X+IX
            dac i 0     /replace instruction
ses,        lxr prc
            law 4000    /set ESI bit
            ior i 4
            dac i 4
            ubn

b3,         dio i bp1 /report breakpoint to superior
            law 4
            jmp spr

ila,        lxr cmp     /memory protection violation
            lac i imr
            sma
            jmp ill+4
            law 6
            jmp spr

adf,        cli         /extend snag
            jmp adf+7
adf+2,      lio i 3     /last snag
            lac i 1
            TAAX
            A+X<M
            jmp ady
adf+7,      lac (77777
            jmp ady+3

adx,        lio i 3     /xsum snag
            lac i 1
            TAAX
            A+X>P
            cla
ady,        and (70000
            A+II
            law 7777
ady+3,      dio t
            rfa
            lxr (-070000
            X+IX
            and i 0
            add t
```

```
adc,        ral 6s
            and (7
            TAAIP|      /attempted core in AC
            jmp .
            dac t4
            sub (6
            sma
            jmp ila     /core can't exist
            lac cmp
            dac cmq
            A+IX
            lac i 1
            szа i
            jmp ila     /core doesn't exist
            law rm3

/bring program field t4 of computation cmq into core, preserving
/core 0 of running computation

br0,        dap brx
            lxr cm1
            lio i 0
            cla
            rcl 3s
            dac t2      /this sphere's core 0
            ŻAX
            lio i uc
            sni i
            jmp . 3
            sas t2
            jmp żaż     /found empty core
            SXXA
            sas nuc
            jmp .-7
            ŻAX
            lio i bc
            spi
            jmp . 3
            sas t2
            jmp żaż     /inactive core
            SXXA
            sas nuc
            jmp .-7
            ŻAIX
            sad t2
            jmp . 7
            law 7777
            and i bc
            AMI_>       /to be sure of getting at least one
            jmp . 3
            X→AI
            dac t0
            SXXA
            sas nuc
            jmp .-12
            lac t0
żaż,        dac t0
            jmp bru+1
```

```
/bring program field into core
/computation in cmq, absolute core (already selected for priority) in t0
/pseudo core in t4, must exist and be on the drum (translation = 6)

bru,        dap brx
            idx bc
            idx bc+1
            idx bc+2
            lxr t0
            lac (600000
            dac i bc
            lac i uc
            sza
            jmp br2
            lac wf      /no previous inhabitant
            ral 6s
            and (37
            TAX
            dzm i sd-1
            dzm wf
            jmp br3

br2,        dac t1      /primary field word
ct1,        dac t2      /current field word
            and (7770
            dac t3      /computation block
            TAAX
            lio i 0
            xor t2
            TAX
            xct i r1
            CXX
            law 6
            rcr 3s
            xct i r2
            lxr t3
            dio i 0     /clear translation of previous inhabitant
            lxr t2
            law 7777
            and i 1     /get next attachment
            sas t1
            jmp ct1
            lac wf
            lxr t1
            dip i 1     /mark last inhabitant on drum

br3,        lac t4
            add cmq
            dac t1      /assignment word
br4,        dac t2
            TAAX
            and (7770
            dac t3
            law i 7777
            and i 1
            sza i
            jmp br5     /just an attachment
            dac rf      /the real field
            lac (add
```

```
            dip i 1
            TXA
            lxr t0
            dac i uc
br5,        lxr t3
            lio i 0
            law 7
            and t2
            TAX
            xct i r1
            lac t0
            rcr 3s
            CXX
            xct i r2
            lxr t3
            dio i 0     /fix up translation
            sas (600000
            jmp .       /was already in core
            lxr t2
            law 7777
            and i 1
            sas t1
            jmp br4
            lac t0
            rcr 3s
            lcr
            dra
            lac .
            lai
            add (30
            dap wf
            dap cf
            dzm dec
            lio wf
            dia
cf,         law .
            lio rf
            dcc
            jmp dre
px,         lac rf
            dac wf
brx,        jmp .

rf,         0           /last read field
```

```
/drum error recovery

dre,        dra
            lac .
            spi i
            jmp .       /not parity error
            isp dec
            jmp de9     /try again
            spq
            jmp .       /unrecoverable
            law i 20
            dac dec
de9,        law 7777
            and wf      /clear write field
            lia
            jmp cf-1

dec,        0
```

```
/ESI trap

xe0,        law i 4000
            and i 4
            dac i 4
xe1,        lxr cmp
            lio i 0
            lcr
            lac i bp1
            TAAI<M
            jmp .+7    /interpreting breakpoint
            law 2
            dac t
            TIM
            isp i bp2 /counting instructions
            jmp spr+2 /cause trap 2
            jmp ses   /turn ESI back on and proceed
            sfa
            jmp xe2   /not in core
            lio (bpt
            sub (070000
            TAX
            lac i 0
            dio i 0
            lxr cmp
            dac i bp3
            ubn

xe2,        lac (700000
            lxr prc
            dip i prn
            lai
            jmp adc
```

```
t,          0

tsb,        0

/tables to rotate translation word

r1,         ril 3s      ril 6s      ril 9s
            rir 6s      rir 3s
r2,         nop
```

```
ivw,        lxr i 5     /ivk trap without PRL
            law 7777
            and i con
            TAXP|
            jmp ill
            rfa
            X→IX
            eem
            law nuf-1
            and i 0
            lem
            add (-nuf
            TIX
            add i df1
            TAXI
            jmp ivt+3

ivt,        rfa         /ivk trap with user PRL
            lxr (-070000
            X+IXI
ivt+3,      dio t2
            lio i 0
            dio t       /capability word
            law 7777
            A←HAP|
            jmp ill     /drum field or does not exist
            dac acp     /low 12 bits of capability
            cla
            rcl 3s
            sad (7
            jmp etr     /enter
            lxr prc
            lxr i 0
            X→AIX
            A$IA
            rcr 3s
            A→IP
            jmp ill     /improper code
            xct i .+1
            jmp ill     /0
            jmp ssp     /1 - entered process
            jmp ifs     /2 - sphere
            jmp pgq     /3 - programmed queue
            jmp ill     /4 - directory
            jmp ill     /5 - file
            jmp .       /6?
```

```
ssp,        law 17      /entered process ivk
            A←IXA
            sub (12
            sma
            jmp ill
            lac acp
            xct i .+1
            jmp sp0     /01 - read state
            jmp sp1     /11 - set state
            jmp sp2     /21 - continue
            jnp sp3     /31 - return
            jmp sp4     /41 - cause illegal inst.
            jmp sp5     /51 - return and skip
            jmp sp6     /61 - read process number
            jmp ill     /71
            jmp mrw     /101 - write memory
            jmp mrr     /111 - read memory

sp1,        stf 6
sp0,        lio (-1     /read/write process state
            stf 2
            lxr prc
            lac i 2
/transmit info with user's core
/AC = user core address
/IO = 1-number of words
/acp = core 7 address
            dac t2      /core address
            sfa
            jmp adc     /not in core
            AMIA
            sfa
            jmp adc     /check for crossing cores
            lac (070000
            ior acp
            TAX
            eem
s01,        aam
            lac t2
            szf i 6
            lac i 0
            aam
            dac t2
            dac i 0
            idx t2
            SII<=
            jmp s02
            SXX
            szf i 2     /to skip over PC
            sni↓szf 4   /to skip over core rename
            SXX
            jmp s01

s02,        lem
            szf i 4
            jmp ret
            lxr t1      /doing read/write process state
            lio i con
            ril 2s
```

```
            lxr acp
            lac i 4
            and (-013700
            spi
            ior (010000
            dac i 4    //replace PRL
            jmp skk

sp5,        TAX
            law 1
            add i 1
            dap i 1
sp3,        lac (400000
            jmp sp4+1
sp2,        ZAP
sp4,        lac (600000
            lxr t2
            dzm i 0
            lxr acp
            dip i prn
            law 6
            dac pri
            jsp acp+1
            jmp ret

sp6,        cli
            TAX
            SII
            law 7777
            and i prn+1
            sza i
            jmp ret    /abandoned
            sas i 5
            jmp .-7
            lxr prc
            dac i 2    /computation
            dio i 0    /process number
            jmp skk

mrw,        stf 6
mrr,        TAX
            sad i prn+1
            jmp ret    /logged out
            lac i 5
            dac acp
            jmp rrr
```

```
ifs,    lxr prc    /sphere ivk
        lio i 2
        and (77
        TAX
        lac t2
        dac t
        law 60
        A←XP
        jmp mt9    /let George do it
        law i 12
        X+A<M
        jmp ill
        lac cmp
        dac cmq
        lac acp
        xct i .+1
        jmp dsb    /02 - suppress processing
        jmp enb    /12 - permit processing
        jmp coa    /22 - attach
        jmp ill    /32
        jmp rdp    /42 - read process state
        jmp wrp    /52 - write process state
        jmp rbs    /62 - read bpt state
        jmp wbs    /72 - write bpt state
        jmp rrr    /102 - read
        jmp www    /112 - write

dsb,    jda stp
        jmp ret

enb,    jda ust
        jmp ret

coa,    A→IAX      /attach
        ral 6s
        and (7
        dac t3     /attaching field
        sub (6
        X→A<M
        jmp ret
        and (7
        dac t4     /attached field
        sub (6
        AAIX
        law arc
        sas cmp
        lac i con
        ral 2s
        swp
        spi        /check for attaching PRL field
        sas (-6
        sma
        jmp ret
```

```
        lio i 0
        lxr t4
        xct i r1
        dio t2      /translation from attachee
        lac acp
        adm t4
        TAX
        sad i 1     /see if attachee exists
        jmp ret     /no
        lac cmq
        add t3
        dac t0
        TAX
        lac i 1
        sza i
        jmp co8
        and (7777
        sas i 1
        jmp ret     /attacher is real core
        dap .+6
        TAX
        law 7777
        and i 1     /follow attachment ring around
        sas t0
        jmp .-4
        law .
        dap i 1
co8,    lxr cmq
        lac t2
        lio i 0
        lxr t3
        xct i r1
        rcr 3s
        CXX
        xct i r2
        lxr cmq
        dio i 0     /insert new translation
        lxr t4      /put attacher in ring
        lio i 1
        lac t0
        dap i 1
        TAX
        dio i 1
        dip i 1
        jmp skk
```

```
wrp,        stf 6
rdp,        cmi↓stf 4 /read/write process state
            dac t1
            TAX
            lac i con
            sma
            jmp ret   /not stopped
rdp+6,      law 7777
            and i prn
            sad acp   /look for selected process
            jmp ret   /does not exist
            TAAX
            SIIP
            jmp rdp+6
            dac acp
            lxr prc
            lac i di1 /core address
            lio (-5
            jmp sp0+4

wbs,        stf 6
rbs,        law bp1   /read/write breakpoint status
            aem acp
            lio (-2
            jmp sp0+2
```

```
www,    stf 6
rrr,    lxr prc   /read/write
        lio i 4
        ril 5s    /own PRL bit
        lac i di1 /own core address
        dac t1
        sfa
        jmp adc   /not in core
        and (077700
        sza
        jmp .+3
        spi
        jmp ret   /violates own PRL
        and (070000
        ral 6s
        dac t3    /own core field
        law 7777
        and i di1
        dac t4    /own address
        lio i 2
        law 7740
        A←II
        dio t     /referenced aedress
        lax i 3777
        and i 0
        rar 6s
        dac t2    /word count
        sub (1
        spa
        law 7777
        dac t5    /count-1
        sub (010000
        A+I<
        jmp ret   /wraps around in referenced computation
        add t4    /own address
        sma
        jmp ret   /wraps around in self
        lac i 2
        and (077700
        lia
        lxr acp   /referenced computation
        law arc
        sas cmp
        lac i con
        ral 2s    /PRL bit of referenced sphere
        spa↓sni   /unless self = adm rt
        jmp ret   /violates PRL
        lai
        and (070000
        ral 6s
        dac t6    /referenced core field
        A+XX
        sub (6
        sma
        jmp ret   /illegal field
fs3,    lxr i 1
        TXXIP|
        jmp ret   /referenced field not assigned
        law 7777
```

```
        A←XX
        X$IAIP|
        jmp fs3     /trace attachment ring
        spa
        jmp fsc     /in core
        szf 6       /on drum
        cla
        A$II
        adm t2      /read field, word count
        lai
        adm t       /write field, drum address
        jmp dc2

fsc,    lxr acp
        lio i 0
        lxr t6      /referenced core field
        xct i r1
        lai         /translated core
        lxr cmp
        lio i 0
        lxr t3      /own core field
        xct i r1    /own translated core
        szf 6
        swp
        rir 3s
        rcr 3s
        lcr         /read core 0, write core 1
        lxr t5      /count-1
        lac t       /referenced address
        lio t4      /own address
        szf 6
        swp
        X+AA
        dap fsr
        lai
        ior (010000
        CXX
        eem
fsr,    lio i .
        X→AX
        dio i 0
        X→AX
        SAA
        SXX>
        jmp fsr
        lem
        jmp skk
```

```
pgq,        and (3      /programmed queue ivk
            TAX
            law 7777
            and t
            xct i .+1
            jmp enq     /03 - enter queue
            jmp rlq     /13 - releate queue
            jmp rqs     /23 - release or skip
            jmp ill

enq,        TAX
            lac i prq
            spa
            jmp eq8
            lac (200000
            A↓XI
            law wa0
            dap rpx
            lxr prc
            lac (400000
            dip i prn /so instruction will complete
            jmp rpc+5
eq8,        SAA<
            TXXA
            dac i prq
            jmp ret

rlq,        TAAX
            lio (-1
            lxr i prq
            TXX>
            jmp rq3
            A$XP
            jmp rqs+5
rq2,        TAX
            dio i prq
            jmp ret
rq3,        I+XI<
            SII
            jmp rq2

rqs,        TAAX
            lxr i prq
            TXX<
            A$XP|
            jmp skk     /queue is empty
rqs+5,      lio i prq
            X→IX
            dap i prq+1
            X→AX
            dap i prq
            lai
            jda acp
            jmp ret
```

```
etr,        law 2       /enter
            dac t6
            lio t
            ril 5s
            spi
            jmp ntr
            sir 5s
            law 77
            A←IAX
            dac t6
            X$II
            law i 7777
            and i ntb+ntl
            A↓IA
            dac t

/enter, object in t6
/transmitted word in t (goes to IO)

ntr,        lxr prc
            lac (200000
            dac i prq /hang entering process
            lxr cmp
            lac i sup
            lxr t6
            TXXP
            lac i ntb+ntl
ntc,        and (7777
            dac cmq
            TAAX
            lio i prh
            TII_<
            jmp .+7     /hoard is not empty
            lio frp     /hoard empty, check pool
            sni
            jmp ntz     /too bad
            law i 1
            adm i prh /increase debt
            lxr (frp-prh
            aam         /unlink from hoard or pool
            lac i prh
            dac i prh
            dio t7      /new process
nty,        lxr cmq
            lio i con
            dio t5
            law 100
            szf 1
            jmp nts     /entering ID from call button
            ril 2s
            spi i       /check for core 0 C-list
            jnp ntp
            dzm t4
            lac i 0     /see if entered comp is in core
            and (700000
            sad (600000
            jsp br0     /bring it in
            lxr cmq
            lio i 0
```

```
            lcr
            lxr (-070000+1
            law 100
ntq,        dac t4
            lio (1
            lac i 0
            sza i
            jmp .+6
            SXX
            SIIA
            sas t4
            jmp .-6
            jmp .       /can't
            lac prc
            ior (150000
            dac i 0
            lxr prc
            lxr i 5
            lac i spe
            lxr t6
            TXXP
            lac i ntb /start address
nts,        lxr t7
            dio i 0    /AC has capability index
            dac i 1    /PC
            lac t
            dac i 2    /transmitted word
            lac cmq
            dac i prn
            dac i 5
            X→AX
            lio i prn+1
            dac i prn+1
            TIX
            dap i prn
            X→AX
            dac i prn+1
            lac t5
            rar 3s
            and (010000
            dac i 4    /initialize PRL
            TXA
            jda acp
            szf 1
            jmp ubx    /call button enter
            jmp wa0

ntp,        law 7777
            and i con
            TAXP|      /check for core 7 C-list
            jmp .
            law i nuf
            add i df1
            SAX
            law 20
            jmp ntq

t7,         0

ntz,        lxr prc
```

```
lac t
dac i prq+1          /transmitted word
idx t6
rar 6s
jmp fr8
```

```
mus,    rar 7s      /end of tailspin
        spa
        jsp mst+4   /if moving, stop
mtg,    unt 100     /unit wait
        unt         /read unit number
        rir 9s
        law 30      /or 170 for 20 units
        A←IX
        ril 9s
        lac (100000
        lok
        mot         /motion select
        ior i mtt+7         /turn on ready bit
        mot 100     /skip ready
        and (7777
        A→I<M       /skip if block or end mark
        jmp un5
        lac (-200000
        mot 300     /skip EOT
        jmp un4     /block mark
        A↓IM|       /in end zone
        jmp un5     /alreaey know about it
        AMIA        /turn on end
        ior (070000         /clear lastrev, need, moving
        CAA<M
        ior (040000         /turn on lastrev if not fwd
        lia
un5,    dio i mtt+7
        law 10      /check whether to end block wait
        A←IP|
        jmp mtg     /not waiting
        lac i mtt+4
        TAAM        /-0 means just waiting to leave end zone
        sub i mtt+5
        A→IA<M
        cmi         /IO has number of blocks to go
        ral 5s
        sma
        jmp .+6     /not ready, or not moving
        rar 4s
        spa
        jmp mtg     /in end zone
        TI_<
        jmp mtf
        law i 10    /terminate block wait
        adm i mtt+7         /clear wait flag
        rar 6s
        sma
        jmp mus     /in tailspin, stop tape
        frk
        mtl
        jmp mtg
```

```
mtf,    ral 3s      /check whether to search
        TA>P
        sas dtf
        jmp mtg     /busy, or don't need to search
        idx dtf
        law 03
        ivk 74
        frk
        mtg
        TXXI
        ril 9s
        dat         /data select
        dat 400     /search
        dat 300     /read status
        spi i
        jmp mth     /block delay or end mark
        dat 200     /read block number
        law 1777
        A←II
        dio i mtt+5           /new block
        lac (-020000
        lok
        and i mtt+7
        dac i mtt+7           /clear need
mth,    law 13      /release data control
        ivk 74
        law i 1
        adm dtf
        qit

un4,    A←II>P      /block mark, clear end bit
        cma
        ral 2s      /+1 or -1, depending on direction
        adm i mtt+5
        jmp un5
```

```
c7e,        lxr (30    /or 170 for 20 units
            X←IX
            iam

tbc_=4      /tape beginning coast distance
tec_=1      /tape ending coast distance

/microtape entry
/index in AC, 10*unit number in XR

mte,        lok
            lio i mtt+7
            rir 6s
            spi i       /busy flag
            jmp .+5
            dap .+2     /unit is busy
            law 41
            ivk .
            qit

            dap i mtt+6          /set up entered process
            lax 40     /mark it busy
            dap i mtt+7
            ulk
            law mtt
            A+XI
            law 1
            xct i mtt+6          /get state of calling process
mtd,        law 777    /translate block number
            and i mtt+1
            ral 1s
            sub (1000
            sma
            CAA|
            add (1001-776
            add (776
            lio i mtt+0
            ril 1s
            spi
            law i 5000           /rewind, set desired block negative
            dac i mtt+3
            rir 1s
            law 10
            rcl 2s
            dap i mtt+7          /set up control flags, clear attempt count
            lac (-200000
            A←XX       /to indicate data is not in buffer
```

```
mt1,        law 100    /decide what to do next
            lok
            adm i mtt+7          /count attempts
            ral 6s
            TAAI>P
            jmp mt0+1 /too many
            lpf        /load tape flags
            iam
            szf i 3
            jmp mt0    /tape not ready
            and (000125
            s+d (000124
            jmp mdo    /rewind complete
            law 341
            A←IA
            sad (301
            jmp mdo    /rewind complete
            lac i mtt+3
            sub i mtt+5          /actual block
            szf i 6
            jmp ms1    /tape not moving
            cli↓cmi↓swp
            szf 2
            jmp ms9+3 /leaving end zone
            szf 1
            cmi
            law tbc+tec+2
            A+II_>
            jmp ms9    /a long way to go, wait
            AMI_<
            jmp mh1    /very close
mh2,        law 2*tbc+tec+3      /went past, or can't get control
            TII=       /skip if can stop in time
            AMI>       /must go past and turn around
            jmp mst    /far enough past, stop
            law i tbc+1          /wait
ms9,        szf 1
            cma
            add i mtt+3          /get waiting block number
ms9+3,      dac i mtt+4
            law 10
            adm i mtt+7          /block wait flag
mda,        TXX<M
            qit
            jmp mth    /release data control

mh1,        TXX>P      /try to get data control
            jmp m12-1 /already have it
            cla
            sas dtf
            jmp mh2    /busy
            idx dtf
            law 3
            ivk 74
            TXXI
            ril 9s
            dat        /data select
            jmp m12
```

```
mst,        law msu
            TXXI
            ril 9s
            mot         /select
mst+4,      dap msv     /stop tape
            lac (-010000
            adm i mtt+7
            TA<M
            law i tec*2
            add (tec
            adm i mtt+5          /fudge block number
            mot 500     /stop
msv,        jmp .

ms1,        szf 2       /tape stopped
            jmp ms4     /in end zone
            CAI<
            cma
            sub (2*tbc+tec+1
            szm
            jmp srt     /quite far away
            add (tbc+tec         /fairly close
            szm
            jmp mr3
            cmi         /too close, go away
srt,        law tbc     /start tape, direction in IO
            spi i
            cma
            adm i mtt+5          /fudge block number
srt+4,      X→IA
            ril 9s
            mot         /motion select
            spa
            mot 600     /forward
            TAI<M
            mot 700     /reverse
            mot 400     /go
            lac i mtt+7          /turn on moving, need
            ior (430000
            spi i
            and (370000          /and direction bit
msu-1,      dip i mtt+7
msu,        ulk
            jmp mtl

ms4,        lio (1000 /start from end zone
            szf 4
            lio (-1
            dio i mtt+5          /set up block number
            jmp srt+4
```

```
mr3,        lac i mtt+7          /stopped a reasonable distance away
            rcl 1s
            rar 1s
            dac i mtt+7          /put in direction bit
            ulk
            TXXI>P
            jmp .+7
            idx dtf    /get data control
            law 03
            ivk 74     /wait as long as necessaqy
            ril 9s
            dat
            skp i
            ril 9s
            lac (030000
            lok
            ior i mtt+7
            dac i mtt+7          /turn on moving, need
            mot        /motion select
            spa
            mot 600    /forward
            sma
            mot 700    /reverse
            mot 400    /go
m12-1,      ulk
m12,        law 7400
            mta
            lac (400000
            A↓XXA      /to indicate that this unit hat data control
            A+X>P
            jmp m15    /stuff is in buffer, too
            lac (200000
            A↓XX
            law i 37
            and i mtt+0
            sas i mtt+0
            jmp mt2    /not on 40 word boundary
            lio i mtt+7
            rir 2s
            A→I<M
            jmp m15    /write
            lac (040111          /read
            xct i mtt+6          /move stuff into buffer
            jmp mt2
```

```
m15,    lio i mtt+7         /ready to try the transfer
        rir 2s
        lac i mtt+3
        spi
        dat 600     /write
        spi i
        dat 500     /read
        dat 300     /get status
        lac (140000
        A←IP
        jmp mtl     /block delay or end of tape
        lac i mtt+3
        dac i mtt+5         /store correct block number
        lac (-020000
        lok
        and i mtt+7
        dac i mtt+7         /clear need bit
        ulk
        spi
        jmp m16     /wrong block number
        ril 1s
        dio tpb
        rar 2s
        spa
        jmp .+6     /was a write
        lio i mtt+0         /read
        lac (040101
        xct i mtt+6         /move stuff out of buffer
        jmp mt2     /bad core address
        lio tpb
        spi
        jmp mdn     /transfer was ok
        cla         /error
        ril 1s
        SAA
        TII_<
        jmp .-3
        jmp mt0+3

mt2,    law 2       /error 2 - bad core address
        jmp mt0+3
mt0,    ZAP         /error 0 - tape not ready
mt0+1,  law 1       /error 1 - can't find block
        ulk
mt0+3,  dac i mtt+0         /error code
        clf 6
        jmp mdf

m16,    dat 200     /read block number
        law 1777
        A←IA
        dac i mtt+5
        jmp mtl

mdn,    law 400     /block transfer complete
        adm i mtt+0
        lio (770000
        idx i mtt+1
        A←IP|
```

```
            jmp mdo
            lac (-010000
            adm i mtt+1
            A←IP
            jmp mtd
mdo,        ulk         /operation complete
            stf 6       /to step PC
mdf,        lio i mtt+7
            law tbc+4-tec       /set up tailspin
            spi i
            cma
            add i mtt+3
            dac i mtt+4
            law mtt
            A+XI
            lax 11
            xct i mtt+6         /write out new AC and IO
            law 10
            lok
            dap i mtt+7
            law 31
            szf 6
            law 51
            xct i mtt+6         /return
            jmp mda   /release data control if have it, then qit

/microtape unit tables

mtt,        repeat 4,[repeat 6,0
            ivk
            0]

tpb,        0           /status

dtf,        0           /number of processes trying to use data control
```

```
ar,     clc     /arq
        siw
        cla
        dac t
        jmp mt9
```

```
/selectric translator

ẓ10,        law 76      /tyi translator
            A$IA
            rar 4s
            spa
            xor (240000
            ral 4s
            sas (16
            sad (15
            jmp ẓ11
/XR = t1 = cns
ẓ55,        ior i t81
            dac t0
            sub (1
            TAAX
            and (17
            lio (11
            AMI_>
            jmp ẓs1
            lio t1
            lxr (ktb-kte-1
ẓs0,        and (277
            sas (200
            SXXP|
            jmp ẓs3
            lac i kte
            X→IX
            xct i trn
            X→IX
            ral 9s
            and (777
            xor t0
            sẓa
            jmp ẓs0
            lac i kte
            TIX
            xct i trn
            xct i trn+2          /jmp ẓs5 or ẓr5
ẓs3,        lac i kte
            TIX
            xct i trn
            ior i t81
            xct i trn+2          /jmp ẓs5 or ẓr5
```

```
zs5,    and (177
        lia
        and (100
        A$II
        sad i t81+1
        jmp rei
        dap i t81+1
        dio i trn+6          /need to save char
        lio (72    /and type in a case shift
        sza
        lio (74
        jsp .+3
        law ti+3
        lio i trn+6
        dap z3
        jmp rei

z11,    cli
        sas (15
        lio (100
        dio i t81
        jmp ti+3

z3,     jmp ti+3

zs1,    A$XA
        sas (100
        jmp .+5
        lac i uut-100
        lxr t1
        xct i trn
        xct i trn+2          /jmp zs5 or zr5

        lxr t1
        lac t0
        xct i trn+4          /jnp zs4 or zr4
zs4,    sad (21
        law 173
        sad (121
        law 106
        jmp zs5
```

```
ƶ50,        law 77      /tyo trap
            and t
            sas (74
            sad (72
            jmp ƶ56
/XR = t1 =  cns+1
            jmp ƶ55

ƶr5,        rar 4s
            spa
            xor (240000
            ral 4s
            xor (76
            lia
            and (100
            sad i t81-1
            jmp ƶ51
            dap i t81-1
            lxr t2
            lio (65
            sƶa i
            lio (66
            jsp itf
            jmp ƶ25

ƶr4,        sad (21
            law 111
            sad (121
            law 113
            jmp ƶr5

ƶ56,        cli
            sas (72
            lio (100
            dio i t81
            jmp ƶ55
```

```
ktb,        277277      /cr
            257275      /backspace
            276275      /line feed
            275236      /tab
            073073      /period
            173040      /colon, centerdot
            033033      /comma
            133056      /semicolon, overbar
            215272      /lower case
            216274      /upper case
            253257      /(, [
            220255      /), ]
            060154      /+
            160120      /→
            040054      /-
            140140      /underbar
            000020      /0
            100104      /backslash, \
            021173      /*
            101156      /|
            013133      /=
            113121      /?
            234234      /black
            237235      /red
kte,        074000

uut,        103156      /upper case numbers
            104103
            102101
            100102
            110107
            121110
            105111
            106105
            107021

constants
end,
```

```
cms-54/     0           /hoard for adm. rt.
cms-41/     0           /hoard for tapes
cms-26/
arl,        0           /login/logout process
            103
            0
            0
            i
            arc
            0
            arc
            arc
            lac rnk
            0

mtp,        0           /microtape unit monitor
            mtg
            0
            0
            add i
            exc
            0
            exc         /proc. ring
            exc
            lac
            0
```

```
cms,
arc,        006676      /computation for adm. rt.
            add arc+1
            arc
            0
            0
            0
            0
            arl         /proc. ring
            arl
            repeat 5,0
            -0
            -0
            0
            0
            0
            0
            100000      /not stopped, PRL
            0
            cms-54      /hoard
            0

exc,        766666
            add exc
            0
            0
            0
            0
            0
            mtp         /proc. ring
            mtp
            repeat 5,0
            -0
            -0
            0
            0
            0
            0
            100000
            0
            cms-41      /hoard
            0
```

```
7740/       0
            jmp sys
7756/       6500
            240000
            250700
sys,        lat↓cli
            TAP
            jmp ysy
            dia         /new system
            lio sys-2
            law i 7777
            jmp 7776
ysy,        lio sys-3 /saver
            dia
            dzm 7776
            lio sys-1
            law i 677
            dcc
            dcc
7777,       hlt
start
```