```
" pd

    sys open; dotdot; 0
    spa; jms error
    dac df
    law dir-1
    dac 8
1:
    lac df
    sys read; tbuf; 8
    spa; jms error
    sna
    jmp 1f
    lac tbuf
    sna
    jmp 1b
    -8
    dac c1
    law tbuf-1
    dac 9
2:
    lac 9 i
    dac 8 i
    isz c1
    jmp 2b
    jmp 1b

1:
    lac df
    sys close
    law 017
    sys creat; dotdot
    spa; jms error
    dac df
    law dir-2
    cma
    tad 8
    dac .+4
    lac df
    sys write; dir; ..
    spa; jms error
    lac df
    sys close
    sys exit

error; 0
    -1
    tad error
    hlt
    sys save

dotdot:
    056056; 040040; 040040; 040040

c1; .=.+1
df; .=.+1
tbuf; .=.+8
dir:
```

```
" psych

    lac d1
    sys close
    lac d13 "pushbuttons
    sys sysloc
    dac .pbp
    tad d1
    dac .pbp1
restart:
    fld; arg1
    fst; g
    fld; arg2
    fst; fact
    fld; fp01
    fst; d
    jms capture
    -100
    dac j
    fld; fp0
    fst; z
    fst; oldx
    fst; oldy
    lac o400000
    dac i 11
    dac i 11
    lac setx
    dac i 10
    lac sety
    dac i 10

loop:
    sys time
    fld; z
    fad; fm110
    fdv; radians
    fst; b
    fld; z
    fad; g
    fst; z
    fld; g
    fng
    fmp; fact
    fst; g
    fld; fp1
    fdv; fact
    fst; fact
    fld; b
    fad; fpid2
    sin
    fmp; d
    fmp; fp500
    fst; xx
    jms in
    jmp done
    fld; b
    sin
    fmp; d
    fmp; fp500
    fst; yy
    jms in
    jmp done
```

g = arg 1

fact = arg 2

d = 1.0

z = 0

oldx = 0

oldy = 0

loop:

$$b = \frac{z + 110.}{\text{radians}}$$

z = z + g

g = - g * fact

fact = 1/fact

XX = 500. * d * sin(b + π/2)

in()

yy = 500. * d * sin(b)

in()

goto loop

```
    lac o400000
    dac i 11
    dac i 11
    fld; oldx
    fng
    fad; xx
    fix
    spa
    xor o775777
    tad vecx
    dac i 10
    fld; oldy
    fng
    fad; yy
    fix
    spa
    xor o775777
    tad vecy
    dac i 10
    fld; xx
    fst; oldx
    fld; yy
    fst; oldy
    isz j
    skp
    jmp done
    fld; g
    fdv; fp90000
    fad; d
    fst; d
    jmp loop

done:
    -1000
    dac 9f
    sys time
    isz 9f
    jmp .-2
    skp
9f:0
    dzm char
    lac auto
    sza
    jmp doauto
1:
    sys time " swap
    lac i .pbp
    sna
    jmp 1b
    spa ral
    jmp 1f
    spa ral
    jmp 2f
    spa ral
    jmp 3f
    spa ral
    jmp 4f
    spa ral
    jmp 5f
    spa ral
    jmp 6f
```

```
    spa ral
    jmp 7f
    jms release
    sys exit
7:
    cla
    sys read; tmp; 1
    lac tmp
    sad o141000
    skp
    jmp 0f
    dac auto
    dzm i .pbp1
    jmp doauto
0:
    sad o12000
    jmp restart
    dzm g
9:
    cla
    sys read; char; 1
    lac char
    sad o12000
    jmp 9f
    lrss 9
    dac char
    lac g
    alss 3
    tad char
    tad om60
    dac g
    jmp 9b
9:
    lac tmp
    sad o61000
    jmp 9f
    lac g
    dac arg2
    jmp restart
9:
    lac g
    dac arg1
    jmp restart
5:
    isz fp500
    nop
    jmp restart
6:
    -1
    tad fp500
    dac fp500
    jmp restart
4:
    lac o400000
    dac char
3:
    fld; arg1
    fmp; fm056
    lac char
    spa
    fng
```

```
    fad; arg1
    fst; arg1
    jmp restart
2:
    lac o400000
    dac char
1:
    fld; arg2
    fmp; fm056
    lac char
    spa
    fng
    fad; arg2
    fst; arg2
    jmp restart

doauto:
    lac i '.pbp1
    sma
    jmp 1f
    dzm auto
    jmp done
1:
    sys time
    omq
    tad 0
    tad stick
    cll
    mul
    78625
    lls 9
    dac stick
    dzm aexp
    and o177777
    xor o200000
    dac ams
    fad; fmhalf
    fmp; fp128
    lac stick
    and d1
    sna
    jmp 1f
    fst; arg1
    jmp restart
1:
    fst; arg2
    jmp restart

in: 0
    fix
    tad d500
    spa
    jmp i in
    tad dm1000
    spa
    isz in
    jmp i in

capture: 0
    law buf-1
    dac 10
```

```
   dac 11
   lac o400000
   dac i 11
   law buf
   sys capt
   jmp i capture

release: 0
   sys rele
   jmp i release

char: 0
stick: 0
xx: 0; 0; 0
oldx: 0; 0; 0
yy: 0; 0; 0
oldy: 0;0 0; 0
j: 0
auto: 0

setx: 0142000 +512
sety: 0146000 +512
vecx: 0100000
vecy: 0124000
o775777: 0775777
o177777: 0177777
o12000: 012000
o61000: 061000
om60: -060
d13: 13
d500: 500
o141000: 0141000
fp128:8; 0240000; 0
fmhalf: 0; 0600000; 0
dm1000: -1000

fact: 0;0;0
g: 0;0;0
z: 0;0;0
b: 0;0;0
d: 0;0;0

radians: 6;0345136;0
fm110: 7;0734000;0
arg1: 7;0234167;0
arg2: 1;0275531;0
fp0: 0;0;0
fp01: -6;0243656;0
fm056: -4;0745301;0
fp500: 11;0372000;0
fp90000: 021;0257620;0
   .pbp: .=.+1
   .pbp1: 0
buf:
```

```
" rm

    lac 017777
    tad d1
    dac 2f
1:
    lac 017777 i
    sad d4
    sys exit
    tad dm4
    dac 017777 i
    lac 2f
    tad d4
    dac 2f
    sys unlink; 2: 0
    sma
    jmp 1b
    lac 2b
    dac 2f
    lac d1
    sys write; 2: 0; 4
    lac d1
    sys write; 1f; 2f-1f
    jmp 1b
1:
    040077;012000
2:

d1: 1
d4: 4
dm4: -4
```

```
" rn

    lac 017777
    tad d1
    dac name2
loop;
    lac 017777 i
    sad d4
    sys exit
    sad d8
    jmp unbal
    tad dm8
    dac 017777 i
    lac name2
    tad d4
    dac name1
    tad d4
    dac name2
    sys unlink; name2: 0
    lac name2
    dac 1f
    sys rename; name1: 0; 1: 0
    sma
    jmp loop
    lac name1
    dac 1f
    lac d1
    sys write; 1: 0; 4
    lac d1
    sys write; mes; 1
    lac name2
    dac 1f
    lac d1
    sys write; 1: 0; 4
    lac d1
    sys write; mes; 2
    jmp loop
mes;
    040000;077012
unbal;
    lac name2
    tad d4
    dac 1f
    lac d1
    sys write; 1: 0; 4
    lac d1
    sys write; mes; 2
    sys exit

d1; 1
d4; 4
d8; 8
dm8; -8
```

```
" roff

    lac i 017777
    sad d4
    sys exit
    lac 017777
    tad d1
    dac fname
    jms nextfile
    lac obufp
    dac otal

" main i/o loop
1:
    jms readline
    jmp 3f
    lac rawchar
    sad cc
    jmp 2f
    jms text
    jmp 1b
2:
    jms control
    jmp 1b
3:
    jms break
    jms eject
    lac otal
    sma
    jmp 1f
    cla
    jms putsc; otal
1:
    -1
    tad obufp
    cma
    tad otal
    dac 1f
    lac output
    sys write; obuf; 1: 0
    sys exit

" read line routine
readline: 0
    law rawchar-1
    dac 8
1:
    jms getchar
    dac i 8
    sad o12
    skp
    jmp 1b
    isz readline
    jmp i readline

" read character routine
getchar: 0
    lac ital
    sad eibufp
    skp
    jmp 1f
    -64
```

```
    dac 3f
    law ibuf-1
    dac 15
2:
    dzm i 15
    isz 3f
    jmp 2b
    lac input
    sys read; ibuf; 64
    sna
    jms nextfile
    lac ibufp
    dac ital
1:
    jms getsc; ital
    sza
    jmp i getchar
    jmp getchar+1
3: 0

putchar: 0
    jms putsc; otal
    lac otal
    sad eobufp
    skp
    jmp i putchar
    lac output
    sys write; obuf; 64
    lac obufp
    dac otal
    jmp i putchar

laci: 0
    dac 1f
    lac i 1f
    jmp i laci
1: 0

nextfile: 0
    lac i 017777
    sad d4
    jmp i readline
    tad dm4
    dac i 017777
    lac fname
    tad d4
    dac fname
    lac input
    sys close
    sys open; fname: 0; 0
    sma
    jmp 1f
    lac fname
    dac 2f
    lac d1
    sys write; 2: 0; 4
    lac d1
    sys write; 2f; 2
    sys exit
2: 040077;012
1:
```

B14 B15 B16 B17 B18 B19 B20

```
    dac input
    lac eibufp
    dac ital
    jmp i nextfile

getsc: 0
lac i getsc
dac sctalp
isz getsc
lac i sctalp
dac sctal
add o400000
dac i sctalp
ral
lac i sctal
szl
lrss 9
and o177
jmp i getsc

putsc: 0
and o177
lmq
lac i putsc
dac sctalp
isz putsc
lac i sctalp
dac sctal
add o400000
dac i sctalp
sma cla
jmp 1f
llss 27
dac i sctal
lrss 9
jmp i putsc
1:
lac i sctal
omq
dac i sctal
lacq
jmp i putsc

" control card decoder
control: 0
    law 2f-1
    dac 8
    -ncase
    dac c
    lac rawchar+1
    alss 9
    xor rawchar+2
1:
    sad i 8
    jmp i 8
    isz 8
    isz c
    jmp 1b
    jmp i control
2:
    ncase = 0
```

B14 B15 B16 B17 B18 B19 B20

```
   <ad>; jmp casead; ncase = ncase+1
   <bp>; jmp casebp; ncase = ncase+1
   <br>; jmp casebr; ncase = ncase+1
   <cc>; jmp casecc; ncase = ncase+1
   <ce>; jmp casece; ncase = ncase+1
   <ds>; jmp caseds; ncase = ncase+1
   <fi>; jmp casefi; ncase = ncase+1
   <in>; jmp casein; ncase = ncase+1
   <li>; jmp caseli; ncase = ncase+1
   <ll>; jmp casell; ncase = ncase+1
   <ls>; jmp casels; ncase = ncase+1
   <na>; jmp casena; ncase = ncase+1
   <ne>; jmp casene; ncase = ncase+1
   <nf>; jmp casenf; ncase = ncase+1
   <pl>; jmp casepl; ncase = ncase+1
   <sp>; jmp casesp; ncase = ncase+1
   <ss>; jmp casess; ncase = ncase+1
   <ti>; jmp caseti; ncase = ncase+1
   <ul>; jmp caseul; ncase = ncase+1
   <un>; jmp caseun; ncase = ncase+1

" control cases
casead:
   jms break
   -1
   dac ad
   jmp i control

casebp:
   jms break
   jms eject
   jmp i control

casebr:
   jms break
   jmp i control

casecc:
   jms skipcont
   lac i 8
   sad o12
   jmp i control
   dac cc
   jmp i control

casece:
   jms break
   jms number; d0
   spa
   cla
   dac ce
   jms need; ce
   jmp i control

caseds:
   jms break
   lac d2
   dac ls
   jmp i control

casefi:
```

```
    jms break
    -1
    dac fi
    jmp i control

casein:
    jms number; in
    cma
    tad d1
    sma
    cla
    dac in
    dac un
    jmp i control

caseli:
    jms number; d0
    cma
    tad d1
    sma
    jmp i control
    dac 2f
1:
    jms readline
    jmp i control
    jms text
    isz 2f
    jmp 1b
    jmp i control
2: 0

casell:
    jms number; ll
    spa
    cla
    dac ll
    jmp i control

casels:
    jms number; d0
    sza; spa
    lac d1
    dac ls
    jmp i control

casena:
    jms break
    dzm ad
    jmp i control

casene:
    jms number; d0
    spa
    cla
    dac c
    jms need; c
    jmp i control

casenf:
    jms break
    dzm fi
```

```
    jmp i control

casepl:
    jms number; pl
    spa
    cla
    dac pl
    jms topbot
    jmp i control

casesp:
    jms break
    jms number; d0
    cma
    tad d1
    sma
    jmp i control
    dac c
1:
    jms nline
    isz c
    jmp 1b
    jmp i control

casess:
    jms break
    lac d1
    dac ls
    jmp i control

caseti:
    jms break
    jms number; in
    spa
    cla
    dac un
    jmp i control

caseul:
    jms number; d0
    spa
    cla
    dac ul
    jmp i control

caseun:
    jms number; d0
    tad in
    sma
    cla
    dac un
    jmp i control

" selected short routines
skipcont: 0
    law rawchar-1
    dac 8
1:
    lac i 8
    sad o40
    jmp 1f
```

```
    sad o12
    jmp 2f
    jmp 1b
1:
    lac i 8
    sad o12
    jmp 2f
    sad o40
    jmp 1b
2:
    -1
    tad 8
    dac 8
    jmp i skipcont

break: 0
    lac nc
    sna
    jmp i break
    -2
    tad ls
    cma
    sma
    jmp 2f
    dac c
1:
    jms nline
    isz c
    jmp 1b
2:
    lac nl
    sad bl
    jms eject
    lac nl
    sza
    jmp 2f
    -5
    dac c
1:
    lac o55
    jms putchar
    isz c
    jmp 1b
    lac ma1
    dac c
1:
    jms newline
    isz c
    jmp 1b
2:
    law char-1
    dac 8
    lac un
    sma
    jmp 1f
    dac c
2:
    lac o40
    jms putchar
    isz c
    jmp 2b
```

```
1:
    lac i 8
    jms putchar
    isz nc
    jmp 1b
    jms newline
    dzm nwd
    dzm ne
    lac in
    dac un
    jmp i break

newline: 0
    lac o12
    jms putchar
    isz nl
    jmp i newline

nline: 0
    lac nl
    sna
    jmp i nline
    sad b1
    jmp i nline
    jms newline
    jmp i nline

number: 0
    dzm num
    dzm sign
    -1
    dac any
    jms skipcont
1:
    lac i 8
    sad o12
    jmp 3f
    sad o53
    jmp 2f
    sad o55
    jmp 2f
    tad om72
    sma
    jmp 1b
    tad o12
    spa
    jmp 1b
    dac any
    lac num
    cll; mul; 10
    lacq
    tad any
    dac num
    jmp 1b
2:
    dac sign
    jmp 1b
3:
    lac any
    sma
    jmp 1f
```

```
    lac d1
    isz number
    jmp i number
1:
    lac sign
    sza
    jmp 1f
    lac num
    isz number
    jmp i number
1:
    sad o53
    jmp 1f
    lac i number
    jms laci
    cma
    tad num
    cma
    isz number
    jmp i number
1:
    lac i number
    jms laci
    tad num
    isz number
    jmp i number

eject: 0
    lac pl
    sna
    jmp i eject
    lac nl
    sna
    jmp i eject
1:
    sad pl
    jmp 1f
    jms newline
    lac nl
    jmp 1b
1:
    dzm nl
    jmp i eject

storechar: 0
    lmq
    lac nc
    sza
    jmp 1f
    lav char-1
    dac 10
1:
    lacq
    dac i 10
    jms width
    cma
    tad d1
    tad ne
    dac ne
    -1
    tad nc
```

```
    dac nc
    jmp i storechar

getword: 0
    law word-1
    dac 8
    dzm wne
    dzm wch
1:
    lac i 11
    sad o12
    jmp i getword
2:
    dac i 8
    lmq
    jms width
    cma
    tad d1
    tad wne
    dac wne
    -1
    tad wch
    dac wch
    lacq
    sad o40
    jmp 1b
    lac word
    sad o40
    jmp 1f
    lac o40
    dac word
    lacq
    jmp 2b
1:
    lac i 11
    sad o12
    jmp 1f
    sad o40
    jmp 1f
    dac i 8
    jms width
    cma
    tad d1
    tad wne
    dac wne
    -1
    tad wch
    dac wch
    jmp 1b
1:
    -1
    tad 11
    dac 11
    isz getword
    jmp i getword

need: 0
    lac ls
    dac 1f
    lac i need
    jms laci
```

```
    cll; mul; 1: 0
    lacq
    tad nl
    cma
    tad bl
    spa
    jms eject
    isz need
    jmp i need

" text line routine
text: 0
    -1
    tad ul
    sma
    jms undline
    -1
    tad ce
    sma
    jms center
    law rawchar-1
    dac 11
    lac rawchar
    sad o12
    jmp 1f-1
    sad o40
    jmp 1f-1
    lac fi
    sza
    jmp 2f
    skp
    jms break
1:
    lac i 11
    sad o12
    jmp 1f
    jms storechar
    jmp 1b
1:
    lac nc
    sna
    jms nline
    jms break
    jmp i text
2:
    jms getword; jmp i text
    lac wne
    tad ne
    tad un
    tad ll
    spa
    jms adjust
    law word-1
    dac 8
    lac nwd
    sza
    jmp 3f
1:
    lac i 8
    sad o40
    skp
```

B14 B15 B16 B17 B18 B19 B20

```
    jmp 3f+1
    isz wch
    jmp 1b
3:
    lac i 8
    jms storechar
    isz wch
    jmp 3b
    isz nwd
    jmp 2b

" adjust routine
adjust: 0
    lac nwd
    sna
    jmp i adjust
    law char-1
    dac 8
    law tchar-1
    dac 9
    dzm ndiv
    dzm nrem
    lac ad
    sna
    jmp 1f
    -1
    tad nwd
    sna
    jmp 1f
    dac 2f
    lac 11
    tad ne
    tad un
    spa
    jmp 1f
    cll; idiv; 2: 0
    dac nrem
    lacq
    dac ndiv
1:
    lac i 8
    sad o40
    jms fill
    dac i 9
    isz nc
    jmp 1b
    lac o12
    dac i 9
    law tchar-1
    dac 8
2:
    lac i 8
    sad o12
    jmp 2f
    jms storechar
    jmp 2b
2:
    jms break
    jmp i adjust

fill: 0
```

```
    lac nrem
    sna
    jmp 2f
    tad dm1
    dac nrem.
    lac d1
2:
    tad ndiv
    cma
    dac c
    lac o40
2:
    dac i 9
    isz c
    jmp 2b
2:
    isz nc
    lac i 8
    sad o40
    skp
    jmp i fill
    dac i 9
    jmp 2b

" more routines
topbot: 0
    lac pl
    sza
    jmp 1f
    dzm bl
    jmp i topbot
1:
    -11
    tad pl
    spa
    jmp 1f
    dac bl
    cma
    tad d1
    tad nl
    spa
    jmp i topbot
    lac bl
    dac nl
    jmp i topbot
1:
    lac d55
    dac bl
    dac nl
    tad d11
    dac pl
    jmp i topbot

undline: 0
    dac ul
    law rawchar-1
    dac 8
    law tchar-1
    dac 9
1:
    lac i 8
```

mov ac,ro
beq 1f
inc ac
mov $1,ro
1:
add r0,ro
add $1,ro
mov ro,-(sp

out if nl small=

BA14 BA15 BA16 BA17 BA18 BA19 BA20

```
    dac i 9
    sad o12
    jmp 1f
    sad o40
    jmp 1b
    lac o10
    dac i 9
    lac o137
    dac i 9
    jmp 1b
1:
    law tchar-1
    dac 8
    law rawchar-1
    dac 9
1:
    lac i 8
    dac i 9
    sad o12
    jmp i undline
    jmp 1b

center: 0
    dac ce
    law rawchar-1
    dac 8
    law tchar-1
    dac 9
    dzm wne
1:
    lac i 8
    dac i 9
    sad o12
    jmp 1f
    jms width
    tad wne
    dac wne
    jmp 1b
1:
    -1
    tad wne
    spa
    jmp i center
    cma
    tad ll
    tad in
    lrss 1
    cma
    tad d1
    sma
    -1
    dac c
    law tchar-1
    dac 8
    law rawchar-1
    dac 9
    lac o40
1:
    dac i 9
    isz c
    jmp 1b
```

```
1:
   lac i 8
   dac i 9
   sad o12
   jmp i center
   jmp 1b

width: 0
   sad o10
   jmp 1f
   lac d1
   jmp i width
1:
   -1
   jmp i width

eibufp: ibuf+64
ibufp: ibuf
eobufp: obuf+64
obufp: obuf
input: 0
output: 1
ls: 1
ce: 0
in: 0
un: 0
ul: 0
ma1: -5
bl: 55
ll: 50
nwd: 0
nl: 0
nc: 0
ne: 0
pl: 66
ad: -1
fi: -1
cc: .>

o12: 012
o40: 040
o177: 0177
o53: 053
om72: -072
o55: 055
o400000: 0400000
o10: 010
d11: 11
d55: 55
o137: 0137
d1: 1
d2: 2
d4: 4
dm4: -4
d0: 0
dm1: -1

c: .=.+1
nrem: .=.+1
ndiv: .=.+1
num: .=.+1
```

B14 B15 B16 B17 B18 B19 B20

```
any: .=.+1
ital: .=.+1
otal: .=.+1
sctal: .=.+1
sctalp: .=.+1
sign: .=.+1
wch: .=.+1
wne: .=.+1
word: .=.+300
char: .=.+300
tchar: .=.+300
rawchar: .=.+300
ibuf: .=.+64
obuf: .=.+64
```

```
" salv

    lac d1
    sys sysloc
    dac iget

    lac d2
    sys sysloc
    dac inode

    lac d4
    sys sysloc
    dac nxfblk
    tad d1
    dac nfblks
    tad d1
    dac fblks

    lac d5
    sys sysloc
    dac copy

    lac d6
    sys sysloc
    dac copyz

    lac d7
    sys sysloc
    dac betwen

    lac d8
    sys sysloc
    dac dskrd

    lac d10
    sys sysloc
    dac dskbuf

    lac d15
    sys sysloc
    dac free

       dzm indircnt
       dzm icnt
       dzm licnt
       dzm blcnt
       dzm curi
       jms copyz i; usetab; 500

iloop;
       isz curi
       -3400
       tad curi
       sma
       jmp part2
       lac curi
       jms iget i
       jms copy i; inode; 0; linode; 12
       lac iflags
       sma
       jmp iloop
       isz icnt
```

```
	lac iflags
	and o40
	sza
	jmp iloop
	law idskps
	dac t1
	-7
	dac t2
1:
	lac i t1
	sza
	jms dupcheck
	isz t1
	isz t2
	jmp 1b
	lac iflags
	and o200000
	sna
	jmp iloop

	isz licnt
	law idskps
	dac t1
	-7
	dac t2
1:
	lac i t1
	sna
	jmp 3f
	jms dskrd i
	jms copy i; dskbuf; 0; idskbuf; 64
	isz indircnt
	law idskbuf
	dac t3
	-64
	dac t4
2:
	lac i t3
	sza
	jms dupcheck
	isz t3
	isz t4
	jmp 2b
3:
	isz t1
	isz t2
	jmp 1b
	jmp iloop

dupcheck: 0
	isz blcnt
	jms betwen i; d709; d6400
	jmp badadr
	dac t5
	lrss 4
	tad usetabp
	dac t6
	cla
	llss 4
	tad alsscom
	dac 2f
```

```
        lac d1
2:      alss 0
        dac bit
        lac i t6
        and bit
        sza
        jmp dup
        lac i t6
        xor bit
        dac i t6
        jmp i dupcheck

badadr:
        jms print
        lac d1
        sys write; badmes; 3
        jmp i dupcheck
badmes:
        < b>;<ad>;<r 012

dup:
        lac t5
        jms print
        lac d1
        sys write; dupmes; 3
        jmp i dupcheck
dupmes:
        < d>;<up>;<a 012

print:  0
        lmq
        law prbuf-1
        dac 8
        -6
        dac t6
1:
        cla
        llss 3
        tad o60
        dac i 8
        isz t6
        jmp 1b
        lac d1
        sys write; prbuf; 6
        jmp i print

part2:
        lac icnt
        jms print
        lac d1
        sys write; m3; m3s
        lac licnt
        jms print
        lac d1
        sys write; m4; m4s
        lac indircnt
        jms print
        lac d1
        sys write; m5; m5s
        lac blcnt
        jms print
```

```
	lac d1
	sys write; m6; m6s
	dzm blcnt
	dzm nxfblk i
	dzm nfblks i

	lac d709
	dac t1
1:
	isz t1
	lac t1
	sad d6400
	jmp part3
	lrss 4
	tad usetabp
	dac t2
	cla
	llss 4
	tad alsscom
	dac 2f
	lac d1
2: alss 0
	dac bit
	lac i t2
	and bit
	sza
	jmp 1b
	lac t1
	jms free i
	isz blcnt
	jmp 1b

part3:
	lac blcnt
	jms print
	lac d1
	sys write; m7; m7s
	sys exit

d1: 1
d2: 2
d4: 4
d5: 5
d6: 6
d7: 7
d8: 8
d10: 10
d15: 15
o60: 060
o400000: 0400000
o400001: 0400001
o40: 040
o200000: 0200000
alsscom: alss 0
d709: 709
d6400: 6400

m3:
	040;<fi>;<le>;<s 012
  m3s = .-m3
m4:
```

```
      040;<la>;<rg>;<e 012
   m4s = .-m4
m5:
      040;<in>;<di>;<r 012
   m5s = .-m5
m6:
      040;<us>;<ed>;012
   m6s = .-m6
m7:
      040;<fr>;<ee>;012
   m7s = .-m7
m8:
      040;<mi>;<ss>;<in>;<g 012
   m8s = .-m8

usetabp: usetab
curi: 0
bit: 0
blcnt: 0
indircnt: 0
icnt: 0
licnt: 0
t1: 0
t2: 0
t3: 0
t4: 0
t5: 0
t6: 0

iget: 0
nxfblk: 0
nfblks: 0
fblks: 0
copy: 0
copyz: 0
betwen: 0
dskrd: 0
free: 0


ldskbuf: .=.+64
linode: .=.+12
   iflags = linode
   idskps = iflags+1
usetab: .=.+500
prbuf: .=.+6
```

```
" sh

clear:
   jmp shell

1:
   dzm i 8
   isz clear
   jmp 1b
   lacq
   jmp 017771
zerop: .-1

comerr:
   lac d1
   sys write; errmes; 1


shell:
   lac d1
   sys write; ready; 1
shell1:
   lac delim
   sad newln
   jms rline
   jms getcom
   lac narg
   sna
   jmp comretrn
   lac args
   sad chcom
   skp
   jmp 3f
   lac args+1
   sad spsp
   jmp doch
3:
   sys fork
      skp
      jmp loadcom
   spa
   jmp comerr
   lmq
   lac delim
   sad amper
   jmp shell1
   lacq
   clq
   sys smes
comretrn:
   lac delim
   sad newln
   jmp shell
   jmp shell1


loadcom:
   sys open; args; 0
   sma
   jmp 1f
   sys link; system; args; args
   spa
```

Need "done" msg to child

B16 B17 B18 BA19 BA20

```
	jmp 2f
	-1
	dac lnkflg
	jmp loadcom

2:
	lac lnkflg
	sna
	jmp 3f
	sys unlink; args
3:
	lac d1
	sys write; args; 4
	lac d1
	sys write; errmes; 1
	sys exit

1:
	lac lnkflg
	sna
	jmp 1f
	sys unlink; args
1:
	lac in
	sna
	jmp 2f
	cla
	sys close
	sys open; in; 0
	sma
	jmp 2f
	lac d1
	sys write; in; 4
	lac d1
	sys write; errmes; 1
	sys exit
2:
	lac out
	sna
	jmp 1f
	lac d1
	sys close
	lac o17
	sys creat; out
	spa
	sys exit
1:
	lac narg
	cma
	dac t1
	tad o17771
	dac 017777
	tad dm1
	dac 8
	and o7777
	dac boot+2
	cma
	tad d7
	dac clear
	lac nargp
	dac 9
```

```
2:
    lac i 9
    dac i 8
    isz t1
    jmp 2b
    lac bootp
    dac 9
    -6
    dac t1
2:
    lac i 9
    dac i 8
    isz t1
    jmp 2b
    lac d2
    lmq
    lac zerop
    dac 8
    jmp clear+1

boot:
    sys read; 4096; '..
    lacq
    sys close
    jmp 4096

getcom: 0
    lav args-1
    dac 10
    dzm in
    dzm out
    dzm narg
    dzm lnkflg

    jms get
nparm:
    sad gr
    jmp cgr
    sad ls
    jmp cls
    sad amper
    jmp endcom
    sad semic
    jmp endcom
    sad nevln
    jmp endcom
    sad space
    jmp nparm-1
    lmq
    lac narg
    tad d4
    dac narg
    lacq
    jms getparm
    jmp nparm

endcom:
    dac delim
    jmp getcom i

cls:
```

```
    jms get
    jms getparm
    dac t1
    law in-1
    jmp cpio
cgr;
    jms get
    jms getparm
    dac t1
    law out-1
cpio;
    dac 11
    lac 10
    tad dm4
    dac 10
    lmq
    -4
    dac c1
1;
    lac i 10
    dac i 11
    isz c1
    jmp 1b
    lacq
    dac 10
    lac t1
    jmp nparm

getparm;0
    lmq
    -8
    dac c1
    lacq
    skp
1;
    jms get
    sad space
    jmp 1b
    jms checkdlm
    jmp comerr
    jmp 2f
1;
    jms get
    jms checkdlm
    jmp fill1
2;
    alss 9
    isz c1
    lmq
    jms get
    jms checkdlm
    jmp fill
    omq
    dac i 10
    isz c1
    jmp 1b
1;
    jms get
    jms checkdlm
    jmp i getparm
    jmp 1b
```

B16
B17
B18
B19
BA20

```
fill:
    dac t1
    lac space
    omq
    dac i 10
    isz c1
    nop
    lac t1
fill1:
    lmq
    lac c1
    spa
    jmp 1f
    lacq
    jmp i getparm
1:
    lac spsp
    dac i 10
    isz c1
    isz c1
    jmp 1b
    lacq
    jmp i getparm

checkdlm;0
    sad space
    jmp i checkdlm
    sad newln
    jmp i checkdlm
    sad amper
    jmp i checkdlm
    sad semic
    jmp i checkdlm
    isz checkdlm
    jmp i checkdlm

get; 0
    lac i 8
    sad slash
    skp
    jmp i get
    lac i 8
    sad newln
    skp
    jmp comerr
    lacq
    dac 1f
    jms rline
    lac 1f
    lmq
    lac space
    jmp i get

1; 0
rline;0
2:
    lav lineb-1
    dac 15
    dac 8
1:
    jms getcha
```

```
    dac i 15
    sad newln
    jmp i rline
    sad sharp
    jmp psharp
    sad atsign
    jmp 2b
    jmp 1b
psharp;
    -1
    tad 15
    sad 2b
    jmp 2b
    tad dm1
    dac 15
    jmp 1b

getcha; 0
    lac char
    dzm char
    sza
    jmp i getcha
    isz nread
    jmp 1f
    cla
    sys read; inbuf; 64
    spa sna
    jmp lgout
    cma
    tad d1
    dac nread
    lav inbuf-1
    dac 14
1;
    lac i 14
    lmq
    and o777
    dac char
    ecla llss 9
    jmp i getcha

doch;
    lac narg
    lrss 2
    cma
    tad d1
    dac narg
    lav args+4
    dac 1f
2;
    isz narg
    skp
    jmp comretrn
    sys chdir; 1:0
    spa
    jmp cherr
    lac 1b
    tad d4
    dac 1b
    jmp 2b
cherr;
```

```
    lac 1b
    dac .+3
    lac d1
    sys write; ..; 4
    jmp comerr

lgout:
    clq
    lac d1
    sys smes
    sys exit


d1: 1
dm1: -1
d4: 4
dm4: -4
d2: 2
d7: 7
o17: 017
o17771: 017771
o7777: 07777
o777: 0777
gr: 076
ls: 074
amper: &>
semic: 073
space: 040
sharp: 043
atsign: 0100
newln: 012
slash: 057
in: 0;0;0;0
out: 0;0;0;0
errmes: 077012
chcom: <ch>
ready: 0100040
delim: 012
system: <sy>; <st>; <em>; spsp: 040040
nargp: narg-1
bootp: boot-1
char: 0
nread: -1
lineb: .=.+128
inbuf: .=.+64
c1: .=.+1
t1: .=.+1
lnkflg: .=.+1
narg: .=.+1
args:
```

B16 B17 B18 BA19 BA20