

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE DIR
 OBJECT MODULE PLACED IN DIRRSP.OBJ
 COMPILER INVOKED BY: \$FO: DIRRSP.PLM XREF OPTIMIZE(3) DEBUG

```
$TITLE("CONCURRENT CP/M 86 --- DIR 1.0 ")
$compact
```

```
/* Conditional compile:
   rsp=0ffh      produce a DIR.RSP type of file
   rsp=0         produce a DIR.CMD file
*/
```

```
$set(rsp=0ffh)
```

```
$include(dirm.plm)
```

```
/* dirm:
```

```
   This is the module included by DIRRSP or DIRCMD.
```

```
Revised:
```

```
Jan 80 by Thomas Rolander
July 81 by Doug Huskey
June 82 by Bill Fittler
July 82 by Danny Horovitz (made an RSP)
Dec 82 by Fran Borda (conditional comp)
Mar 83 by Bill Fittler ( " " )
Mar 83 by Danny Horovitz (control C fixes)
```

```
Conditional compile:
```

```
rsp=0ffh      produce a DIR.RSP type of file
rsp=0         produce a DIR.CMD file
*/
```

```
**** Vax commands to compile DIR.RSP and DIR.CMD:
```

```
$ ccpmsetup
$ plm86 dircmd.plm "p1" "p2" "p3" "p4" optimize(3) debug
$ link86 fltscd.obj, dircmd.obj to dircmd.lnk
$ loc86 dircmd.lnk od(sm(code,dats,data,stack,const))-
   ad(sm(code(0),dats(10000h))) ss(stack(+32)) to dircmd.
$ h86 dircmd
$ | DIR.RSP
$ | Note: separate code and data
$ asm86 rhdir.a86 IRsp Header DIR
$ plm86 dirrsp.plm "p1" "p2" "p3" "p4" optimize(3) debug
$ link86 rhdir.obj, dirrsp.obj to dirrsp.lnk
$ loc86 dirrsp.lnk od(sm(code,dats,data,stack,const))-
   ad(sm(code(0),dats(10000h))) ss(stack(0)) to dirrsp.
$ h86 dirrsp
```

```
**** Then, on a micro:
```

```
A>vax dircmd.h86 $fans
A>vax dirrsp.h86 $fans
A>gencmd dircmd data[b1000]
A>ren dir.cmd=dircmd.cmd
A>gencmd dirrsp data[b1000]
```

CCP/M-86 2.0 v.2

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

```

=      A>ren dir.rsp=dirrsp.cmd
=
=      **** Notes: Both DIRCMD.PLN and DIRRSP.PLN include DIRM.PLN, after setting
=      RSP flag appropriately.
=
=      ****/
=
1      =      dir:
=      do:
=
=      $include (:f1:copyrt.lit)
=1
=1      /*
=1      Copyright (C) 1983
=1      Digital Research
=1      P.O. Box 579
=1      Pacific Grove, CA 93950
=1      */
=
=      $include (:f1:comlit.lit)
=1
2      1      =1      declare
=1          lit          literally      'literally',
=1          dcl          lit          'declare',
=1          true         lit          'Offh',
=1          false        lit          '0',
=1          no           lit          'not',
=1          boolean      lit          'byte',
=1          forever      lit          'while true',
=1          cr           lit          '13',
=1          lf           lit          '10',
=1          tab          lit          '9';
=
=      $include (:f1:mfunc.lit)
=1
=1      /* Concurrent CP/M function numbers */
=1
3      1      =1      dcl          m$prtbuf      lit          '9',
=1          m$select      lit          '14',
=1          m$openf       lit          '15',
=1          m$closef      lit          '16',
=1          m$ueletef     lit          '19',
=1          m$readf       lit          '20',
=1          m$writef      lit          '21',
=1          m$makef       lit          '22',
=1          m$getlogin    lit          '24',
=1          m$curdsk      lit          '25',
=1          m$setdma      lit          '26',
=1          m$setatt      lit          '30',
=1          m$setusr      lit          '32',
=1          m$readrf      lit          '33',
=1          m$writerf     lit          '34',
=1          m$resetdrv    lit          '37',
=1          m$errmode     lit          '45',

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

=1      m$dirbios      lit      "50",
=1      m$makeq        lit      "134",
=1      m$openq        lit      "135",
=1      m$deleteq      lit      "136",
=1      m$readq        lit      "137",
=1      m$creadq       lit      "138",
=1      m$writeq       lit      "139",
=1      m$cwriteq      lit      "140",
=1      m$delay        lit      "141",
=1      m$dispatch     lit      "142",
=1      m$setprior     lit      "145",
=1      m$attach       lit      "146",
=1      m$detach       lit      "147",
=1      m$setcns       lit      "148",
=1      m$parse        lit      "152",
=1      m$getcns       lit      "153",
=1      m$sysdat       lit      "154",
=1      m$getpd        lit      "156",
=1      m$abort        lit      "157";

=1      /* Internal calls */
=1
4 1  =1      dcl      m$sleep      lit      "0212H",
=1      m$wakeup      lit      "0213H";
=1
=1      =
=1      $include (:f1:proces.lit)
=1
=1      /*
=1      Proces Literals MP/M-8086 II
=1      */
=1
5 1  =1      declare pnamsiz literally "8";
=1
6 1  =1      declare pd$hdr literally "structure
=1      (link word,thread word,stat byte,prior byte,flag word,
=1      name (8) byte,uda word,dsk byte,user byte,ldsk byte,luser byte,
=1      mem word";
=1
7 1  =1      declare pd$structure literally "pd$hdr,
=1      dvract word,wait word,org byte,net byte,parent word,
=1      cns byte,abort byte,conmode word,lst byte,sf3 byte,sf4 byte,sf5 byte,
=1      reservd (4) byte,pret word,scratch word";
=1
8 1  =1      declare p$run      lit      "00",
=1      p$poll      lit      "01",
=1      p$delay      lit      "02",
=1      p$swap      lit      "03",
=1      p$term      lit      "04",
=1      p$sleep      lit      "05",
=1      p$sdq      lit      "06",
=1      p$snq      lit      "07",
=1      p$flagwait  lit      "08",
=1      p$ciowait   lit      "03";
=1
9 1  =1      declare pf$sys      lit      "00001h",
=1      pf$keep      lit      "00002h",

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

=1      pf$kernal      lit "00004h",
=1      pf$pure        lit "00008h",
=1      pf$table       lit "00010h",
=1      pf$resource    lit "00020h",
=1      pf$raw         lit "00040h",
=1      pf$ctlc        lit "00080h",
=1      pf$active      lit "00100h",
=1      pf$tempkeep    lit "00200h",
=1      pf$ctld        lit "00400h",
=1      pf$childabort  lit "00800h",
=1      pf$noctls      lit "01000h";
=1
10 1      declare pcm$ll      lit "00001h",
=1      pcm$ctls          lit "00002h",
=1      pcm$trout         lit "00004h",
=1      pcm$ctlc          lit "00003h",
=1      pcm$ctlo          lit "00080h",
=1      pcm$rsx           lit "00300h";
=1
=1      $include (:f1:qd.lit)
=1
=1      /* Queue Descriptor */
=1
11 1      =1 dcl qnamsiz lit "8";
=1
12 1      =1 dcl qd$structure lit "structure(
=1      link word,
=1      net byte,
=1      org byte,
=1      flags word,
=1      name(qnamsiz) byte,
=1      msglen word,
=1      nmsgs word,
=1      dq word,
=1      nq word,
=1      asgcnt word,
=1      msgout word,
=1      buffer word)";
=1
=1      /* queue flag values */
=1
13 1      =1 dcl qf$mx      lit "001h"; /* Mutual Exclusion */
14 1      =1 dcl qf$keep    lit "002h"; /* NO DELETE */
15 1      =1 dcl qf$hide    lit "004h"; /* Not User writable */
16 1      =1 dcl qf$rsp     lit "008h"; /* rsp queue */
17 1      =1 dcl qf$stable  lit "010h"; /* from qd table */
18 1      =1 dcl qf$rpl     lit "020h"; /* rpl queue */
19 1      =1 dcl qf$dev     lit "040h"; /* device queue */
=1
=1      /* Queue Parameter Block */
=1
20 1      =1 dcl qpb$structure lit "structure(
=1      flgs byte,
=1      net byte,
=1      qaddr word,
=1      nmsgs word,
=1      buffptr word,

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

=1      name (qname$iz) byte )";
=1
=
=      /*****
=      *
=      *      B D D S   INTERFACE
=      *
=      *****/
=
21  1  =      mon1:
=          procedure (func,info) external;
22  2  =          declare func byte;
23  2  =          declare info address;
24  2  =          end mon1;
=
25  1  =      mon2:
=          procedure (func,info) byte external;
26  2  =          declare func byte;
27  2  =          declare info address;
28  2  =          end mon2;
=
29  1  =      mon3:
=          procedure (func,info) address external;
30  2  =          declare func byte;
31  2  =          declare info address;
32  2  =          end mon3;
=
33  1  =      mon4:
=          procedure (func,info) pointer external;
34  2  =          declare func byte;
35  2  =          declare info address;
36  2  =          end mon4;
=
=
37  1  =      patch: procedure public;      /* dummy area for patching code segments */
38  2  =      declare i address;
=          /* first statement = 9 bytes, rest are 5 bytes */
39  2  =          i=i+5; i=i+5; i=i+5; i=i+5; i=i+5;
44  2  =          i=i+5; i=i+5; i=i+5; i=i+5; i=i+5;
49  2  =          i=i+5; i=i+5; i=i+5; i=i+5; i=i+5;      /* about 79 bytes */
54  2  =          end patch;
=
=
=      $if rsp
55  1  =          declare fcb (36) byte;          /* 1st default fcb      */
56  1  =          declare fcb16 (1) byte at (@fcb(16)); /* 2nd default fcb      */
=      $else
=          declare fcb (1) byte external; /* 1st default fcb      */
=          declare fcb16 (1) byte external; /* 2nd default fcb      */
=      $endif
=
57  1  =      write$console:
=          procedure (char);
58  2  =          declare char byte;
59  2  =          call mon1 (2,char);

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

60 2 =      end write$console;
61 1 =      print$buf:
        =      procedure (buffer$address);
62 2 =          declare buffer$address address;
63 2 =          call mon1 (9,buffer$address);
64 2 =      end print$buf;
65 1 =      check$ctrl$c:
        =      procedure byte;
        =      $if rsp
66 2 =          if (dir$pd.flag and pf$ctlC) <> 0 then
67 2 =              do;
68 3 =                  dir$pd.flag = dir$pd.flag and not double(pf$ctlC);
69 3 =                  return(true);
70 3 =              end;
        =      $endif
71 2 =          return (false);
72 2 =      end check$ctrl$c;
73 1 =      search$first:
        =      procedure (fcb$address) byte;
74 2 =          declare fcb$address address;
75 2 =          return mon2 (17,fcb$address);
76 2 =      end search$first;
77 1 =      search$next:
        =      procedure (fcb$address) byte;
78 2 =          declare fcb$address address;
79 2 =          return mon2 (18,fcb$address);
80 2 =      end search$next;
81 1 =      setdma: procedure(dma);
82 2 =          declare dma address;
83 2 =          call mon1(26,dma);
84 2 =      end setdma;
85 1 =      get$user$code:
        =      procedure byte;
86 2 =          return mon2 (32,0ffh);
87 2 =      end get$user$code;
88 1 =      set$user$code:
        =      procedure(user);
89 2 =          declare user byte;
90 2 =          call mon1 (32,user);
91 2 =      end set$user$code;
92 1 =      terminate:
        =      procedure;
93 2 =          call mon1 (0,0);
94 2 =      end terminate;
95 1 =      declare
        =          parse$fn structure (
        =              buff$adr address,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

=      fcb$adr  address);
=      delimiter based parse$fn.buff$adr byte;
96  1  =      declare tail$len address;
=
97  1  =      parse: procedure address;
98  2  =      return mon3(152,.parse$fn);
99  2  =      end parse;
=
=
100 1  =      crlf:
=      procedure;
101 2  =      call write$console (0dh);
102 2  =      call write$console (0ah);
103 2  =      end crlf;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

/* * * * * *

```

```

* * * GLOBAL VARIABLES * * *

```

```

* * * * *

```

```

104 1  =      declare dir$title (*) byte initial
=      ("Directory for User x:", "$");
=
105 1  =      declare (sys,temp,dcnt,cnt,user) byte;
106 1  =      declare
=      l
=      new$user  byte,
=      sys$exists byte,
=      incl$sys  byte,
=      option    byte;

```

```

107 1  =      declare
=      dirbuf (128) byte;
=

```

```

/* * * * * *

```

```

* * * DIRECTORY DISPLAY * * *

```

```

* * * * *

```

```

=      /* display directory heading */
108 1  =      heading: procedure;
=
109 2  =      if user > 9 then
110 2  =      do;
111 3  =      dir$title(19) = "1";
112 3  =      dir$title(20) = user - 10 + "0";

```

```

113 3 =      end;
      =      else
114 2 =      do;
115 3 =          dir$title(19) = " ";
116 3 =          dir$title(20) = user + "0";
117 3 =      end;
118 2 =      call print$buf (.dir$title);
119 2 =      end heading;

      =
      = /* ----- */
      =
      = /*
      = help: procedure;
      = call mon1(m$prt$buf, .(cr, lf, tab, tab, tab, "DIR EXAMPLES$"));
      = call mon1(m$prt$buf, .(cr, lf, lf, "dir", tab, tab,
      = "(show all directory files on current drive and user)
      = call mon1(m$prt$buf, .(cr, lf, "dir lg3l", tab, tab, tab, tab,
      = "(show non system files under user 3)$");
      = call mon1(m$prt$buf, .(cr, lf, "dir a: b: lsj", tab, tab, tab,
      = tab, "(show all files under current user on a: and b:)$");
      = call terminate;
      = end help;
      = */
      =
      = /* ----- */
      =
      = /* do next directory display */
120 1 = directory: procedure boolean;
      =
121 2 =     shown$nothing = false;
122 2 =     if new$user then do;
124 3 =         call heading;
125 3 =         new$user = false;
126 3 =     end;
127 2 =     syst$exists = false;
128 2 =     cnt = -1;
      =     /* if drive is 0 (default)
      =         then set to current disk */
129 2 =     if fcb(0) = 0
      =         then fcb(0) = mon2 (m$curdisk,0) + 1;
131 2 =     if fcb(1) = " " then
      =         /* check for blank filename => wildcard */
132 2 =         do i = 1 to 11;
133 3 =             fcb(i) = "?";
134 3 =         end;
      =         /* get first file */
135 2 =         if (dcnt := search$first (.fcb)) <> 0ffh then
136 2 =             do while dcnt <> 0ffh;
137 3 =                 temp = shl(dcnt,5);
138 3 =                 sys = ((dirbuf(temp+10) and 80h) = 80h);
139 3 =                 if (dirbuf(temp) = user) and
      =                     (incl$sys or not sys) then
140 3 =                     do;
141 4 =                         if ((cnt:=cnt+1) mod 4) = 0 then
142 4 =                             do;
143 5 =                                 call crlf;
144 5 =                                 call write$console ("A"+fcb(0)-1);

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

145 5 =      end;
      =      else
146 4 =      do;
147 5 =          call write$console (' ');
148 5 =      end;
149 4 =      call write$console (' ');
150 4 =      call write$console (' ');
151 4 =      do i = 1 to 11;
152 5 =          if i = 9 then call write$console (' ');
154 5 =          call write$console
              (dirout(temp+i) and 7fh);
      =          if check$ctrl$c then
155 5 =              return(false);
156 5 =          end;
157 5 =      end;
158 4 =      end;
159 3 =      else if sys then
160 3 =          sys$exists = true;
      =          dcnt = search$next (.fcb);
162 3 =      end;
163 2 =      if cnt = -1 then
164 2 =          do;
165 3 =              call print$buf (.0dh,0ah,
              =              "File not found.", "$");
166 3 =          end;
167 2 =          if sys$exists then
168 2 =              call print$buf (.0dh,0ah,
              =              "System Files Exist", "$");
169 2 =          return(true);
170 2 =      end directory;

```

```

/* * * * * *

```

```

* * * PARSING * * *

```

```

* * * * *

```

```

/* parse one file name, return true if got one */
171 1 = parse$file; procedure boolean;
172 2 =     dcl i address;
173 2 =     dcl buf based parse$fn.buff$adr (1) byte;
174 2 =     dcl parse$ret address;
      =
175 2 =     if (parse$ret := parse$fn.buff$adr) = 0 then
176 2 =         return(false);
177 2 =     fcb(0), i = 0;
178 2 =     parse$ret = parse;          /* kludge around */
179 2 =     do while parse$ret = 0 and buf(i) = '['; /* parse file name bugs */
180 3 =         if (i := findb(@buf(i), "]", tail$len - i)) <> 0ffffh then
181 3 =             do;
182 4 =                 parse$fn.buff$adr = .buf(i) + 1; /* skip right bracket */
183 4 =                 i = 0;
184 4 =                 parse$ret = parse;
185 4 =             end;
      =     else

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

186 3 =      buf(i) = 0;
187 3 =      end;
188 2 =      parse$fn.buff$adr = parse$ret;
      =
      =
189 2 =      if parse$fn.buff$adr <> 0ffffh then
190 2 =      do;
191 3 =          if fcb(1) <> " " then
192 3 =          do;
193 4 =              if parse$fn.buff$adr <> 0 and delimiter <> '[' and delimiter <> 0 then
194 4 =                  parse$fn.buff$adr = parse$fn.buff$adr + 1;
195 4 =              return(true);          /* parse$fn.buff$adr could = 0 */
196 4 =          end;
197 3 =          else if fcb(0) <> 0 and fcb(1) = " " then /* drive spec */
198 3 =          do;
199 4 =              call setb("?", 3fcb(1), 11);
200 4 =              return(true);
201 4 =          end;
      =
      end;
      else /* if parse$fn.buff$adr = 0ffffh then */
203 2 =      do;
204 3 =          call print$buf(.(cr, lf, "Invalid filespec.$"));
205 3 =          shown$nothing = false;      /* don't show directory */
206 3 =          return(false);      /* also if parse$fn.buff$adr = 0 and fcb(0) = " " */
207 3 =      end;
208 2 =      end parse$file;
      =
      =
      /* ----- */
      =
      =
      /* parse & interpret all options - assume global */
209 1 =      parse$options: procedure boolean;
210 2 =      dcl (n,i) word;
211 2 =      dcl (options, in$brackets, error) boolean;
212 2 =      i = 0;          /* parse file name doesn't work with delimiters */
213 2 =      parse$fn.fcb$adr = .dirbuf;
214 2 =      error = false;
215 2 =      options = true;
216 2 =      do while options and not error;
217 3 =          if (n := findb(@tbuf(1), "[", tail$len - 1)) = 0ffffh then
218 3 =              options = false;
      =
      else
219 3 =      do;
220 4 =          i = i + n + 1;
221 4 =          parse$fn.buff$adr = .tbuf(1);
222 4 =          in$brackets = true;
223 4 =          do while in$brackets and not error;
224 5 =              if (parse$fn.buff$adr := parse) <> 0ffffh then
225 5 =              do;
226 6 =                  if dirbuf(1) = "S" then
227 6 =                      incl$sys = true;
228 6 =                  else if dirbuf(1) = "G" then
229 6 =                  do;
230 7 =                      if dirbuf(3) <> " " then
231 7 =                          temp = dirbuf(3) - "0" + 10;
232 7 =                      else if dirbuf(2) <> " " then

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

233 7 =      temp = dirbuf(2) - "0";
      =      if temp < 16 then
235 7 =      do;
236 8 =          call mon1(m$setusr, (user:=temp));
237 8 =          new$usr = true;
238 8 =      end;
239 7 =      end;
      =      else
240 6 =          error = true;
241 6 =      end; /* if parse */
242 5 =      if delimiter = "]" or parse$fn.buff$adr = 0 or
      =      parse$fn.buff$adr = 0ffffh then
243 5 =          in$brackets = false;
244 5 =      end; /* while in$brackets */
245 4 =      end; /* else */
246 3 =      end; /* while options */
      =
247 2 =      if error then
248 2 =      do;
249 3 =          call print$buf(, (cr, lf, "Invalid Command Option$"));
250 3 =          return(false);
      =          /* call help; */
251 3 =      end;
252 2 =      return(true);
253 2 = end parse$options;

```

```

/* * * * * *

```

```

* * * * * MAIN PROGRAM * * * * *

```

```

* * * * *

```

```

$if rsp
254 1 = declare cpd$pointer pointer;          /* Calling PD pointer stuff */
255 1 = declare cpd$ptr structure (
      =      offset address, segment address) at (@cpd$pointer);
256 1 = declare calling$pd based cpd$pointer pd$structure;
257 1 = declare dpd$pointer pointer;          /* DIR RSP PD pointer stuff */
258 1 = declare dpd$ptr structure (
      =      offset address, segment address) at (@dpd$pointer);
259 1 = declare dir$pd based dpd$pointer pd$structure;
260 1 = declare qdbuf (131) byte;
261 1 = declare dirqd qd$structure initial
      =      (0, 0, 0, qf$keep + qf$rsp, "DIR", 131, 1, 0, 0, 0, 0, .qdbuf);
262 1 = declare qpbbuf (131) byte;
263 1 = declare cpd$offset address at (@qpbbuf(0));
264 1 = declare tbuf (128) byte at (@qpbbuf(2));
265 1 = declare dirqpb qpb$structure initial
      =      (0, 0, 0, 0, .qpbbuf, "DIR", );
      = $else
      = declare tbuf (128) byte external;
      = $endif

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

266 1 = declare shown$nothing boolean;
267 1 = plmstart: procedure public;
    =
    = /* initialization */
    =
    = $if rsp
268 2 = call mon1(m$make$q, .dir$qd);
269 2 = call mon1(m$open$q, .dir$qpb);
270 2 = call mon1(m$set$prior, 200); /* Set priority same as other transients*/
    =
    = $else
    = user = get$user$code; /* ????? whf */
    = incl$sys = (fcb(1) = "S"); /* ????? why exclude if rsp? whf */
    = $endif
    =
271 2 = call setdma(.dirbuf);
    = $if rsp
272 2 = cpd$pointer, qpd$pointer = mon1(m$sysdat, 0);
273 2 = qpd$ptr.offset = mon1(m$getpd, 0);
    = /* Don't allow control S, turn on tempkeep for control C checking */
274 2 = dir$pd.flag = dir$pd.flag or pf$noctls or pf$tempkeep;
    = /* Read RSP Queue forever */
275 2 = do forever;
276 3 = call mon1(m$readq, .dir$qpb);
277 3 = dir$pd.flag = dir$pd.flag and not double(pf$ctlC);
    = /* Could be on from last DIR */
    = /* set defaults same as calling process's, have both PDs so will poke */
    = /* and not call O.S. */
278 3 = cpd$ptr.offset = cpd$offset;
279 3 = call mon1(m$setcns, calling$pd.cns);
280 3 = call mon1(m$setusr, (user := calling$pd.user));
281 3 = call mon1(m$select, calling$pd.dsk);
    = $endif
282 3 = new$user = true;
283 3 = sys$exists, incl$sys = false;
284 3 = tail$len = findb($tbuf, 0, 128);
    =
    = /* scan for options - all are global */
    =
285 3 = if not parse$options then
286 3 = goto done; /* option error */
    =
    = /* do command line */
    =
287 3 = shown$nothing = true;
    = $if rsp
288 3 = parse$fn.buff$adr = .tbuf;
    = $else
    = parse$fn.buff$adr = (.tbuf) + 1; /* Skip # of bytes in buffer */
    = $endif
289 3 = parse$fn.fcb$adr = .fcb;
290 3 = do while parse$file; /* false when no more files, sets */
291 4 = if not directory then /* shown$nothing=false if parsing error */
292 4 = goto done; /* directory = false if console input */
293 4 = end;
    =
294 3 = if shown$nothing then /* no files specified on command line */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
295 3 = do;
296 4 = call setb("?", 0fcb(1), 11);
297 4 = if not directory then
298 4 = goto done; /* false on console input */
299 4 = end;
300 3 = done;
    = $if rsp
    = call monl(m$detach, 0);
301 3 = end; /* do forever */
    = $else
    = call terminate;
    = $endif
    =
302 2 = end plmstart;
    =
303 1 = end dir;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

CROSS-REFERENCE LISTING

 DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

256	0021H	1	ABORT.	BYTE MEMBER(CALLINGPD)														
259	0021H	1	ABORT.	BYTE MEMBER(DIRPD)														
2			BOOLEAN.	LITERALLY	120	171	209	211	266									
173	0000H	1	BUFF.	BYTE BASED(PARSEFN.BUFFADR) ARRAY(1)						179	180	182	186					
95	0000H	2	BUFFADR.	WORD MEMBER(PARSEFN)	95	173	175	179	180	182	186	188	189					
					193	194	221	224	242	288								
261	001AH	2	BUFFER.	WORD MEMBER(DIRQD)														
61	0004H	2	BUFFERADDRESS.	WORD PARAMETER AUTOMATIC				62	63									
265	0006H	2	BUFFPTR.	WORD MEMBER(DIRQPB)														
256	0000H	48	CALLINGPD.	STRUCTURE BASED(CPDPOINTER)				279	280	281								
57	0004H	1	CHAR.	BYTE PARAMETER AUTOMATIC				58	59									
65	007AH	29	CHECKCTRLC.	PROCEDURE BYTE STACK=0002H				155										
259	0020H	1	CNS.	BYTE MEMBER(DIRPD)														
256	0020H	1	CNS.	BYTE MEMBER(CALLINGPD)				279										
105	0082H	1	CNT.	BYTE	128	141	163											
256	0022H	2	CONMODE.	WORD MEMBER(CALLINGPD)														
259	0022H	2	CONMODE.	WORD MEMBER(DIRPD)														
263	018FH	2	CPDOFFSET.	WORD AT	278													
254	0010H	4	CPDPOINTER.	POINTER	255	256	272	279	280	291								
255	0010H	4	CPDPTR.	STRUCTURE AT	278													
2			CR.	LITERALLY	204	249												
100	0106H	17	CRLF.	PROCEDURE STACK=000EH				143										
2			DCL.	LITERALLY														
105	0081H	1	DCNT.	BYTE	135	136	137	161										
95	0000H	1	DELIMITER.	BYTE BASED(PARSEFN.BUFFADR)				193	242									
1	0002H		DIR.	PROCEDURE STACK=0000H														
107	0089H	128	DIRBUF.	BYTE ARRAY(128)	138	139	154	213	226	228	230	231	232	233				
					271													
120	0143H	339	DIRECTORY.	PROCEDURE BYTE STACK=0012H				291	297									
259	0000H	48	DIRPD.	STRUCTURE BASED(CPDPOINTER)				66	68	274	277							
261	0018H	28	DIRQD.	STRUCTURE INITIAL	268													
265	0034H	16	DIRQPB.	STRUCTURE INITIAL	269	276												
104	0068H	23	DIRTITLE.	BYTE ARRAY(23) INITIAL	111	112	115	116	118									
81	0004H	2	DMA.	WORD PARAMETER AUTOMATIC				82	83									
300	0593H		DONE.	LABEL	286	292	298											
			DOUBLE.	BUILTIN	68	277												
257	0014H	4	DPOPOINTER.	POINTER	66	68	258	259	272	274	277							
258	0014H	4	DPDPTR.	STRUCTURE AT	273													
261	0012H	2	DQ.	WORD MEMBER(DIRQD)														
256	0012H	1	DSK.	BYTE MEMBER(CALLINGPD)				281										
259	0012H	1	DSK.	BYTE MEMBER(DIRPD)														
256	0018H	2	DVRACT.	WORD MEMBER(CALLINGPD)														
259	0018H	2	DVRACT.	WORD MEMBER(DIRPD)														
211	0108H	1	ERROR.	BYTE	214	216	223	240	247									
2			FALSE.	LITERALLY	71	121	125	127	156	176	205	206	214	218	243			
					250	283												
55	0044H	36	FCB.	BYTE ARRAY(36)	56	129	130	131	133	135	144	161	177	191				
					197	199	289	296										
56	0054H	1	FCB16.	BYTE ARRAY(1) AT														

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

4		MIWAKEUP	LITERALLY																	
3		MMAKEF	LITERALLY																	
3		MMAKEQ	LITERALLY	269																
21	0000H	MON1	PROCEDURE	EXTERNAL(0) STACK=0000H	59	63	83	90	93	235										
				268 269 270 276 279 280 281 300																
25	0000H	MON2	PROCEDURE	BYTE EXTERNAL(1) STACK=0000H		75	79	86	130											
29	0000H	MON3	PROCEDURE	WORD EXTERNAL(2) STACK=0000H		98	273													
33	0000H	MON4	PROCEDURE	POINTER EXTERNAL(3) STACK=0000H		272														
3		MOPENF	LITERALLY																	
3		MOPENQ	LITERALLY	269																
3		MPARSE	LITERALLY																	
3		MPRTBUF	LITERALLY																	
3		MREADF	LITERALLY																	
3		MREADQ	LITERALLY	276																
3		MREADRF	LITERALLY																	
3		MRESETDRV	LITERALLY																	
3		MSELECT	LITERALLY	281																
3		MSETATT	LITERALLY																	
3		MSETCNS	LITERALLY	279																
3		MSETOMA	LITERALLY																	
3		MSETPRIOR	LITERALLY	270																
3		MSETUSR	LITERALLY	236 280																
261	0016H	2 MSGCNT	WORD MEMBER(DIRQD)																	
261	000EH	2 MSGLEN	WORD MEMBER(DIRQD)																	
261	0018H	2 MSGOUT	WORD MEMBER(DIRQD)																	
3		MSYSDAT	LITERALLY	272																
3		MWRITEF	LITERALLY																	
3		MWRITEQ	LITERALLY																	
3		MWRITERF	LITERALLY																	
210	000CH	2 N	WORD	217 220																
256	0008H	8 NAME	BYTE ARRAY(8) MEMBER(CALLINGPD)																	
265	0008H	8 NAME	BYTE ARRAY(8) MEMBER(DIRQPB)																	
261	0006H	8 NAME	BYTE ARRAY(8) MEMBER(DIRQD)																	
259	0008H	8 NAME	BYTE ARRAY(8) MEMBER(DIRPD)																	
256	0010H	1 NET	BYTE MEMBER(CALLINGPD)																	
261	0002H	1 NET	BYTE MEMBER(DIRQD)																	
265	0001H	1 NET	BYTE MEMBER(DIRQPB)																	
259	0010H	1 NET	BYTE MEMBER(DIRPD)																	
106	0085H	1 NEWUSER	BYTE	122 125 237 282																
265	0004H	2 NMSGs	WORD MEMBER(DIRQPB)																	
261	0010H	2 NMSGs	WORD MEMBER(DIRQD)																	
2		NO	LITERALLY																	
261	0014H	2 NO	WORD MEMBER(DIRQD)																	
255	0000H	2 OFFSET	WORD MEMBER(CPOPTR)	278																
258	0000H	2 OFFSET	WORD MEMBER(DPPPTR)	273																
106	0088H	1 OPTION	BYTE																	
211	0109H	1 OPTONS	BYTE	215 216 218																
256	001CH	1 ORG	BYTE MEMBER(CALLINGPD)																	
261	0003H	1 ORG	BYTE MEMBER(DIRQD)																	
259	001CH	1 ORG	BYTE MEMBER(DIRPD)																	
256	001EH	2 PARENT	WORD MEMBER(CALLINGPD)																	
259	001EH	2 PARENT	WORD MEMBER(DIRPD)																	
97	00F7H	15 PARSE	PROCEDURE WORD STACK=0008H	178 184 224																
171	0296H	213 PARSEFILE	PROCEDURE BYTE STACK=000EH	290																
95	0002H	4 PARSEFN	STRUCTURE	95 98 173 175 179 180 182 186 188 189 193																
				194 213 221 224 242 288 289																
209	036BH	269 PARSEOPTIONS	PROCEDURE BYTE STACK=000EH	285																

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

174	000AH	2	PARSERET	WORD	175 178 179 184 188
37	0002H	85	PATCH.	PROCEDURE PUBLIC STACK=0002H	
10			PCM11.	LITERALLY	
10			PCMCTLC.	LITERALLY	
10			PCMCTLD.	LITERALLY	
10			PCMCTLS.	LITERALLY	
10			PCMRQT.	LITERALLY	
10			PCMRXS.	LITERALLY	
6			PDHDR.	LITERALLY	256 259
7			POSTRUCTURE.	LITERALLY	256 259
9			PFACTIVE.	LITERALLY	
9			PFCHILDA8OKT.	LITERALLY	
9			PFCTLC.	LITERALLY	66 68 277
9			PFCTLD.	LITERALLY	
9			PFKEEP.	LITERALLY	
9			PFKERNAL.	LITERALLY	
9			PFNOCTLS.	LITERALLY	274
9			PFPURE.	LITERALLY	
9			PFRW.	LITERALLY	
9			PFRESOURCE.	LITERALLY	
9			PFSYS.	LITERALLY	
9			PFTABLE.	LITERALLY	
9			PFTMPKEEP.	LITERALLY	274
267	0478H	298	PLMSTART.	PROCEDURE PUBLIC STACK=0016H	
5			PNAMSTZ.	LITERALLY	
256	002CH	2	PRET.	WORD MEMBER(CALLINGPD)	
259	002CH	2	PRET.	WORD MEMBER(DIRPD)	
61	006AH	16	PRINTBUF.	PROCEDURE STACK=000AH	118 165 168 204 249
259	0005H	1	PRIOR.	BYTE MEMBER(DIRPD)	
256	0005H	1	PRIOR.	BYTE MEMBER(CALLINGPD)	
8			PSCIOWAIT.	LITERALLY	
8			PSDELAY.	LITERALLY	
8			PSDQ.	LITERALLY	
8			PSFLAGWAIT.	LITERALLY	
8			PSNQ.	LITERALLY	
8			PSPOLL.	LITERALLY	
8			PSRUN.	LITERALLY	
8			PSSLEEP.	LITERALLY	
8			PSSWAP.	LITERALLY	
8			PSTERM.	LITERALLY	
265	0002H	2	QADDR.	WORD MEMBER(DIRQPB)	
260	010CH	131	QDBUF.	BYTE ARRAY(131)	261
12			QDSTRUCTURE.	LITERALLY	261
19			QFDEV.	LITERALLY	
15			QFHIDE.	LITERALLY	
14			QFKEEP.	LITERALLY	261
13			QFMX.	LITERALLY	
18			QFRPL.	LITERALLY	
16			QFRSP.	LITERALLY	261
17			QFTABLE.	LITERALLY	
11			QNAMSTZ.	LITERALLY	261 265
262	018FH	131	QPBDEF.	BYTE ARRAY(131)	263 264 265
20			QPBSTRUCTURE.	LITERALLY	265
256	0028H	4	RESERVED.	BYTE ARRAY(4) MEMBER(CALLINGPD)	
259	0028H	4	RESERVED.	BYTE ARRAY(4) MEMBER(DIRPD)	
256	002EH	2	SCRATCH.	WORD MEMBER(CALLINGPD)	
259	002EH	2	SCRATCH.	WORD MEMBER(DIRPD)	

CCP/M-86 2.0 v.2
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

73	0097H	16	SEARCHFIRST.	PROCEDURE BYTE STACK=000AH	135															
77	00A7H	16	SEARCHNEXT.	PROCEDURE BYTE STACK=000AH	161															
258	0002H	2	SEGMENT.	WORD MEMBER(CDPDTR)																
255	0002H	2	SEGMENT.	WORD MEMBER(CDPDTR)																
			SETB.	BUILTIN 199 296																
81	0087H	16	SETDMA.	PROCEDURE STACK=000AH	271															
88	0006H	19	SETUSERCODE.	PROCEDURE STACK=000AH																
256	0025H	1	SF3.	BYTE MEMBER(CALLINGPD)																
259	0025H	1	SF3.	BYTE MEMBER(DIRPD)																
256	0026H	1	SF4.	BYTE MEMBER(CALLINGPD)																
259	0026H	1	SF4.	BYTE MEMBER(DIRPD)																
256	0027H	1	SF5.	BYTE MEMBER(CALLINGPD)																
259	0027H	1	SF5.	BYTE MEMBER(DIRPD)																
			SHL.	BUILTIN 137																
266	0212H	1	SHOWNOTHING.	BYTE 121 205 287 294																
259	0004H	1	STAT.	BYTE MEMBER(DIRPD)																
256	0004H	1	STAT.	BYTE MEMBER(CALLINGPD)																
105	007FH	1	SYS.	BYTE 138 139 159																
106	0086H	1	SYSEXISTS.	BYTE 127 160 167 283																
			TAB.	LITERALLY																
96	0006H	2	TAILLEN.	WORD 180 217 284																
264	0191H	128	TBUFF.	BYTE ARRAY(128) AT 217 221 284 288																
105	0080H	1	TEMP.	BYTE 137 138 139 154 231 233 234 236																
92	00E9H	14	TERMINATE.	PROCEDURE STACK=0008H																
259	0002H	2	THREAD.	WORD MEMBER(DIRPD)																
256	0002H	2	THREAD.	WORD MEMBER(CALLINGPD)																
			TRUE.	LITERALLY 69 160 169 195 200 215 222 227 237 240 252																
				275 282 287																
256	0010H	2	UDA.	WORD MEMBER(CALLINGPD)																
259	0010H	2	UDA.	WORD MEMBER(DIRPD)																
256	0013H	1	USER.	BYTE MEMBER(CALLINGPD)	280															
105	0083H	1	USER.	BYTE 109 112 116 139 236 230																
88	0004H	1	USER.	BYTE PARAMETER AUTOMATIC 89 90																
259	0013H	1	USER.	BYTE MEMBER(DIRPD)																
256	001AH	2	WAIT.	WORD MEMBER(CALLINGPD)																
259	001AH	2	WAIT.	WORD MEMBER(DIRPD)																
57	0057H	19	WRITECONSOLE.	PROCEDURE STACK=000AH	101 102 144 147 149 150 153 154															

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

MODULE INFORMATION:

CODE AREA SIZE = 05A2H 1442D
 CONSTANT AREA SIZE = 0054H 84D
 VARIABLE AREA SIZE = 0213H 531D
 MAXIMUM STACK SIZE = 0016H 22D
 695 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

ISIS-11 MCS-86 LOCATER, V1.2 INVOKED BY:

IFG: DIRRSP.LNK DD(SHCODE,DATA,STACK,CONST) DD(SHCODE(0), DATA(1000000)) SS(STACK(0)) IS DIRRSP.

WARNING 26: DECREASING SIZE OF SEGMENT

SEGMENT: STACK

SYMBOL TABLE OF MODULE RDIR

READ FROM FILE DIRRSP.LNK

WRITTEN TO FILE DIRRSP.

BASE	OFFSET	TYPE	SYMBOL
1000H	0000H	PUB	RSPLINK
0000H	0036H	PUB	PATCH
0000H	0025H	PUB	MON1
0000H	0025H	PUB	MON3

BASE	OFFSET	TYPE	SYMBOL
0000H	040CH	PUB	PLMSTART
0000H	0025H	PUB	XDOS
0000H	0025H	PUB	MON2
0000H	0025H	PUB	MON4

DIR:	SYMBOLS	AND	LINES
1000H	038GH	SYM	MEMORY
1000H	0140H	SYM	I
1000H	0194H	SYM	FCB16
STACK	0004H	SYM	CHAR
STACK	0004H	SYM	BUFFERADDRESS
0000H	00C4H	SYM	SEARCHFIRST
0000H	00D3H	SYM	SEARCHNEXT
0000H	00EBH	SYM	SETDMA
0000H	00F8H	SYM	GETUSERCODE
STACK	0004H	SYM	USER
1000H	0142H	SYM	PARSEFN
1000H	0146H	SYM	TALLLEN
0000H	013AH	SYM	CRLF
1000H	01BFH	SYM	SYS
1000H	01C1H	SYM	DCNT
1000H	01C3H	SYM	USER
1000H	01C5H	SYM	NEWUSER
1000H	01C7H	SYM	INCLSYS
1000H	01C9H	SYM	DIRBUF
0000H	0177H	SYM	DIRECTORY
1000H	0148H	SYM	I
1000H	014AH	SYM	PARSERET
1000H	014CH	SYM	N
1000H	0249H	SYM	OPTIONS
1000H	024BH	SYM	ERROR
1000H	0150H	SYM	CPDPTR
1000H	0154H	SYM	CPDPINTER
1000H	0154H	BAS	DIRPD
1000H	0156H	SYM	DIRQD
1000H	02CFH	SYM	CPDFFSET
1000H	0174H	SYM	DIRQPB
0000H	04ACH	SYM	PLMSTART
0000H	0036H	LIN	37
0000H	0044H	LIN	40
0000H	0042H	LIN	42
0000H	0058H	LIN	44
0000H	0062H	LIN	46
0000H	006CH	LIN	48
0000H	0076H	LIN	50
0000H	0080H	LIN	52
0000H	0089H	LIN	54

BASE	OFFSET	TYPE	SYMBOL
0000H	038GH	SYM	MEMORY
1000H	0140H	SYM	I
1000H	0194H	SYM	FCB16
STACK	0004H	SYM	CHAR
STACK	0004H	SYM	BUFFERADDRESS
0000H	00C4H	SYM	SEARCHFIRST
0000H	00D3H	SYM	SEARCHNEXT
0000H	00EBH	SYM	SETDMA
0000H	00F8H	SYM	GETUSERCODE
STACK	0004H	SYM	USER
1000H	0142H	SYM	PARSEFN
1000H	0146H	SYM	TALLLEN
0000H	013AH	SYM	CRLF
1000H	01BFH	SYM	SYS
1000H	01C1H	SYM	DCNT
1000H	01C3H	SYM	USER
1000H	01C5H	SYM	NEWUSER
1000H	01C7H	SYM	INCLSYS
1000H	01C9H	SYM	DIRBUF
0000H	0177H	SYM	DIRECTORY
1000H	0148H	SYM	I
1000H	014AH	SYM	PARSERET
1000H	014CH	SYM	N
1000H	0249H	SYM	OPTIONS
1000H	024BH	SYM	ERROR
1000H	0150H	SYM	CPDPTR
1000H	0154H	SYM	CPDPINTER
1000H	0154H	BAS	DIRPD
1000H	0156H	SYM	DIRQD
1000H	02CFH	SYM	CPDFFSET
1000H	0174H	SYM	DIRQPB
0000H	04ACH	SYM	PLMSTART
0000H	0036H	LIN	37
0000H	0044H	LIN	40
0000H	0042H	LIN	42
0000H	0058H	LIN	44
0000H	0062H	LIN	46
0000H	006CH	LIN	48
0000H	0076H	LIN	50
0000H	0080H	LIN	52
0000H	0089H	LIN	54

BASE	OFFSET	TYPE	SYMBOL
0000H	04ACH	PUB	PLMSTART
0000H	0025H	PUB	XDOS
0000H	0025H	PUB	MON2
0000H	0025H	PUB	MON4

BASE	OFFSET	TYPE	SYMBOL
0000H	038GH	SYM	PATCH
1000H	0184H	SYM	FCB
0000H	0088H	SYM	WRITECONSOLE
0000H	009EH	SYM	PRINTBUF
0000H	00AEH	SYM	CHECKCTRLC
STACK	0004H	SYM	FCBADDRESS
STACK	0004H	SYM	FCBADDRESS
STACK	0004H	SYM	DMA
0000H	0104H	SYM	SETUSERCODE
0000H	0110H	SYM	TERMINATE
1000H	0140H	BAS	DELIMITER
0000H	012BH	SYM	PARSE
1000H	01A8H	SYM	DIRTITLE
1000H	01C0H	SYM	TEMP
1000H	01C2H	SYM	CNT
1000H	01C4H	SYM	I
1000H	01C6H	SYM	SYSEXISTS
1000H	01C8H	SYM	OPTION
0000H	014BH	SYM	HEADING
0000H	02CAH	SYM	PARSEFILE
1000H	0140H	BAS	BUF
0000H	039FH	SYM	PARSEDPTIONS
1000H	014EH	SYM	I
1000H	024AH	SYM	INBRACKETS
1000H	0150H	SYM	CPDPINTER
1000H	0150H	BAS	CALLINGPD
1000H	0154H	SYM	CPDPTR
1000H	024CH	SYM	QDBUF
1000H	02CFH	SYM	QPBUFF
1000H	02D1H	SYM	TEUFF
1000H	0352H	SYM	SHOWNNOTHING
0000H	05C7H	SYM	DONE
0000H	0039H	LIN	39
0000H	0049H	LIN	41
0000H	0053H	LIN	43
0000H	005DH	LIN	45
0000H	0067H	LIN	47
0000H	0071H	LIN	49
0000H	007BH	LIN	51
0000H	0085H	LIN	53
0000H	008BH	LIN	57

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93350

SER. # _____

0000H	008EH	LTN	59
0000H	009EH	LTN	61
0000H	00AAH	LIN	64
0000H	0091H	LTN	65
0000H	00C3H	LTN	69
0000H	00C6H	LIN	72
0000H	00CCH	LTN	75
0000H	00D8H	LTN	77
0000H	00E3H	LIN	80
0000H	00EEH	LIN	83
0000H	00F0H	LIN	85
0000H	010AH	LIN	87
0000H	010DH	LIN	90
0000H	011DH	LTN	92
0000H	0129H	LIN	94
0000H	012EH	LTN	98
0000H	013AH	LIN	100
0000H	0143H	LIN	102
0000H	014BH	LTN	108
0000H	0155H	LTN	111
0000H	0161H	LIN	113
0000H	0166H	LTN	116
0000H	0175H	LTN	119
0000H	017AH	LIN	121
0000H	0186H	LTN	124
0000H	018EH	LTN	127
0000H	0193H	LIN	129
0000H	01AEH	LTN	131
0000H	01C1H	LIN	133
0000H	01D2H	LIN	135
0000H	01E4H	LTN	137
0000H	020AH	LTN	139
0000H	0233H	LIN	143
0000H	023FH	LTN	145
0000H	0245H	LTN	149
0000H	0251H	LIN	151
0000H	0264H	LTN	153
0000H	027EH	LIN	155
0000H	0289H	LIN	157
0000H	0291H	LTN	159
0000H	029DH	LTN	161
0000H	02AAH	LIN	163
0000H	02B3H	LTN	167
0000H	02C6H	LTN	169
0000H	02CAH	LTN	171
0000H	02D4H	LTN	177
0000H	02E3H	LTN	179
0000H	032EH	LIN	182
0000H	0338H	LTN	185
0000H	0343H	LTN	187
0000H	0346H	LIN	189
0000H	0357H	LTN	193
0000H	0368H	LTN	195
0000H	036DH	LIN	197
0000H	0389H	LTN	200
0000H	038DH	LIN	204
0000H	0399H	LIN	206
0000H	039FH	LTN	209

0000H	009AE	LTN	66
0000H	00A1H	LIN	68
0000H	00AEH	LTN	65
0000H	00B3H	LTN	68
0000H	00C7H	LTN	71
0000H	00C3H	LTN	73
0000H	00D6H	LTN	76
0000H	00DEH	LTN	79
0000H	00EBH	LIN	81
0000H	00F7H	LTN	84
0000H	00FEH	LTN	86
0000H	010AP	LIN	88
0000H	0119H	LTN	91
0000H	0120H	LTN	93
0000H	012BH	LIN	97
0000H	013AH	LTN	99
0000H	013DH	LTN	101
0000H	0149H	LIN	103
0000H	014EH	LIN	109
0000H	015AH	LTN	112
0000H	0161H	LIN	115
0000H	016EH	LIN	118
0000H	0177H	LTN	120
0000H	017FH	LIN	122
0000H	0189H	LIN	125
0000H	0193H	LIN	128
0000H	019FH	LIN	130
0000H	01B5H	LTN	132
0000H	01CCH	LTN	134
0000H	01E0H	LTN	136
0000H	01F4H	LIN	138
0000H	021EH	LIN	141
0000H	0236H	LIN	144
0000H	023FH	LTN	147
0000H	024BH	LTN	150
0000H	025DH	LIN	152
0000H	026AH	LTN	154
0000H	0285H	LTN	156
0000H	028FH	LIN	158
0000H	0298H	LTN	160
0000H	02A7H	LTN	162
0000H	02B1H	LIN	165
0000H	02B5H	LTN	168
0000H	02CAH	LTN	170
0000H	02CDH	LIN	175
0000H	02E5H	LTN	178
0000H	030DH	LTN	180
0000H	0336H	LIN	183
0000H	0338H	LTN	186
0000H	0345H	LIN	188
0000H	0350H	LIN	191
0000H	0367H	LTN	194
0000H	036DH	LTN	196
0000H	037BH	LIN	199
0000H	038DH	LTN	202
0000H	0394H	LIN	205
0000H	039DH	LIN	208
0000H	03A2H	LTN	212

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

0000H	0379H	LIN	213	0000H	03A0H	LIN	214
0000H	03B3H	LIN	215	0000H	03B8H	LIN	216
0000H	03C8H	LIN	217	0000H	03F1H	LIN	218
0000H	03F8H	LIN	220	0000H	0403H	LIN	221
0000H	0400H	LIN	222	0000H	0411H	LIN	223
0000H	0410H	LIN	224	0000H	0425H	LIN	226
0000H	0430H	LIN	227	0000H	0437H	LIN	228
0000H	043EH	LIN	230	0000H	0445H	LIN	231
0000H	044EH	LIN	232	0000H	0455H	LIN	233
0000H	0450H	LIN	234	0000H	0464H	LIN	236
0000H	0470H	LIN	237	0000H	0475H	LIN	239
0000H	0477H	LIN	240	0000H	047CH	LIN	242
0000H	048EH	LIN	243	0000H	0493H	LIN	244
0000H	0496H	LIN	246	0000H	0495H	LIN	247
0000H	049DH	LIN	249	0000H	04A4H	LIN	250
0000H	04A8H	LIN	252	0000H	04ACH	LIN	253
0000H	04ACH	LIN	267	0000H	04AFH	LIN	268
0000H	04B9H	LIN	269	0000H	04C3H	LIN	270
0000H	04C0H	LIN	271	0000H	04D4H	LIN	272
0000H	04EEH	LIN	273	0000H	04FBH	LIN	274
0000H	0500H	LIN	275	0000H	0500H	LIN	276
0000H	0517H	LIN	277	0000H	0521H	LIN	278
0000H	0527H	LIN	279	0000H	0538H	LIN	280
0000H	054CH	LIN	281	0000H	0550H	LIN	282
0000H	0562H	LIN	283	0000H	056AH	LIN	284
0000H	0583H	LIN	285	0000H	058CH	LIN	287
0000H	0591H	LIN	288	0000H	0597H	LIN	289
0000H	059DH	LIN	290	0000H	05A4H	LIN	291
0000H	05ADH	LIN	293	0000H	05AFH	LIN	294
0000H	05B6H	LIN	296	0000H	05C4H	LIN	297
0000H	05C7H	LIN	300	0000H	05D1H	LIN	301
0000H	05D4H	LIN	302	0000H	0036H	LIN	303

MEMORY MAP OF MODULE RHD1R
 READ FROM FILE DIRRSP.LNK
 WRITTEN TO FILE DIRRSP.

SEGMENT MAP

START	STOP	LENGTH	ALIGN	NAME	CLASS
00000H	00505H	0506H	G	CODE	CODE
10000H	1013FH	0140H	G	DATS	DATA
10140H	10352H	0213H	W	DATA	DATA
10354H	10354H	0000H	W	STACK	STACK
10354H	103A7H	0054H	W	CONST	CONST
103B0H	103B0H	0000H	G	??SEG	
103B0H	103B0H	0000H	W	MEMORY	MEMORY

GROUP MAP

ADDRESS	GROUP OR SEGMENT NAME
10000H	DGROUP
	DATS
	CONST
	DATA

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

000000H CORDUP
CODE

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

IS15-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE RHDTR
 OBJECT MODULE PLACED IN :FO:RHDTR.OBJ
 ASSEMBLER INVOKED BY: :FO:RHDTR.A86

LCL	OBJ	LINE	SOURCE
		1	: Code and Data Interface for DIR.RSP
		2	: August 10, 1981
		3	: July 27, 1982 (updated)
		4	: Jan 1983 whf - added reg save to xdos
		5	: Mar 1983 dh - shrunk to resonable size
		6	
		7	name rhdir ;Rsp Header DIRectory
		8	
		9	cgroun group code
		10	dgroup group dats
		11	public xdos,mon1,mon2,mon3,mon4
		12	public rsplink
		13	extrn plmstart:near
		14	
		15	assume cs:cgroun,ds:dgroup
		16	
		17	dats segment "DATA"
----		18	org 0
0000		19	rsphdr_len equ 16
0010		20	pd_len equ 30H
0030		21	uda_len equ 100H
0100		22	rsp_top equ 0
0000		23	rsp_pd equ rsp_top + rsphdr_len
0010		24	rsp_uda equ rsp_pd + pd_len
0040		25	rsp_bottom equ rsp_uda + uda_len
0140		26	org rsp_top
0000		27	
		28	rsplink dw 0 ;RSP header
0000 0000		29	sdatvar dw 0 ;becomes system data page paragraph
0002 0000		30	ncopies db 0
0004 00		31	dw 0,0,0,0, 0
0005 0000			
0007 0000			
0009 0000			
0008 0000			
0000 0000			
000F 00	32	db 0	
0010	33	org rsp_pd	
0010 0000	34	pd dw 0,0	;link fields
0012 0000			
0014 00	35	db 0	;status
0015 8E	36	db 190	;priority
0016 0300	37	dw 3	;flags - system and keep
0018 44495220202020	38	db "DIR"	;name
20			
0020 0400	39	dw rsp_uda/10h	;uda paragraph
0022 00	40	db 0,0	;disk,user
0023 00			
0024 00	41	db 0,0	;ldisk,luser
0025 00			
0026 0000	42	dw 0	;puremem - not re-entrant

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

LOC	OBJ	LINE	SOURCE
		43	;rest of pd
		44	
0040		45	org rsp_uda ;start of uda
0040 0000		46	uda dw 0
0042 0000		47	dw 0 ;DNA must be explicitly set
0044 0000		48	dw 0,0,0,0, 0,0,0,0, 0,0,0,0
0046 0000			
0048 0000			
004A 0000			
004C 0000			
004E 0000			
0050 0000			
0052 0000			
0054 0000			
0056 0000			
0058 0000			
005A 0000			
005C 0000		49	dw 0,0,0,0, 0,0,0,0, 0,0,0,0
005E 0000			
0060 0000			
0062 0000			
0064 0000			
0066 0000			
0068 0000			
006A 0000			
006C 0000			
006E 0000			
0070 0000			
0072 0000			
0074 3A01		50	dw offset stk_top
00A0		51	org rsp_uda + 60H
00A0 01		52	db 1 ;insys = 1, always use UDA stack
00A1 00		53	db 0
00A2 CCCC		54	dw 0cccc,0cccc,0cccc,0cccc ;62
00A4 CCCC			
00A6 CCCC			
00A8 CCCC			
00AA CCCC		55	dw 0cccc,0cccc,0cccc,0cccc ;6A
00AC CCCC			
00AE CCCC			
00B0 CCCC			
00B2 CCCC		56	dw 0cccc,0cccc,0cccc,0cccc ;72
00B4 CCCC			
00B6 CCCC			
00B8 CCCC			
00BA CCCC		57	dw 0cccc,0cccc,0cccc,0cccc ;7A
00BC CCCC			
00BE CCCC			
00C0 CCCC			
00C2 CCCC		58	dw 0cccc,0cccc,0cccc,0cccc ;82
00C4 CCCC			
00C6 CCCC			
00C8 CCCC			
00CA CCCC		59	dw 0cccc,0cccc,0cccc,0cccc ;8A
00CC CCCC			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

LOC	OBJ	LINE	SOURCE
000E	CCCC		
0000	CCCC		
0002	CCCC	60	dw 0cccc,0cccc,0cccc,0cccc ;92
0004	CCCC		
0006	CCCC		
0008	CCCC		
000A	CCCC	61	dw 0cccc,0cccc,0cccc,0cccc ;9A
000C	CCCC		
000E	CCCC		
0010	CCCC		
0012	CCCC	62	dw 0cccc,0cccc,0cccc,0cccc ;A2
0014	CCCC		
0016	CCCC		
0018	CCCC		
001A	CCCC	63	dw 0cccc,0cccc,0cccc,0cccc ;AA
001C	CCCC		
001E	CCCC		
00F0	CCCC		
00F2	CCCC	64	dw 0cccc,0cccc,0cccc,0cccc ;B2
00F4	CCCC		
00F6	CCCC		
00F8	CCCC		
00FA	CCCC	65	dw 0cccc,0cccc,0cccc,0cccc ;BA
00FC	CCCC		
00FE	CCCC		
0100	CCCC		
0102	CCCC	66	dw 0cccc,0cccc,0cccc,0cccc ;C2
0104	CCCC		
0106	CCCC		
0108	CCCC		
010A	CCCC	67	dw 0cccc,0cccc,0cccc,0cccc ;CA
010C	CCCC		
010E	CCCC		
0110	CCCC		
0112	CCCC	68	dw 0cccc,0cccc,0cccc,0cccc ;D2
0114	CCCC		
0116	CCCC		
0118	CCCC		
011A	CCCC	69	dw 0cccc,0cccc,0cccc,0cccc ;DA
011C	CCCC		
011E	CCCC		
0120	CCCC		
0122	CCCC	70	dw 0cccc,0cccc,0cccc,0cccc ;E2
0124	CCCC		
0126	CCCC		
0128	CCCC		
012A	CCCC	71	dw 0cccc,0cccc,0cccc,0cccc ;EA
012C	CCCC		
012E	CCCC		
0130	CCCC		
0132	CCCC	72	dw 0cccc,0cccc,0cccc,0cccc ;F2
0134	CCCC		
0136	CCCC		
0138	CCCC		
013A	0000	E 73	stk_top dw plmstart,0,0 ;initial RET

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

LOC	OBJ	LINE	SOURCE
013C	0000		
013E	0000		
0140		74	
		75	org rsp_bottom
		76	
0140		77	datsead equ offset 1
----		78	data ends
		79	
----		80	code segment public "CODE"
0000		81	org 0
		82	
0000	434F5059524947	83	db "COPYRIGHT (C) 1982,"
	48542028432920		
	313938322C		
0013	20444947495441	84	db " DIGITAL RESEARCH "
	4C205245534541		
	52434820		
		85	
0025		86	xdos proc
0025	55	87	push bp
0026	88EC	88	mov bp,sp
0028	885604	89	mov dx,[bp+4]
0028	884E06	90	mov cx,[bp+6]
002E	CDE0	91	int 224
0030	5D	92	pop bp
0031	C20400	93	ret 4
		94	xdos endp
		95	
0025		96	mon1 equ xdos
0025		97	mon2 equ xdos
0025		98	mon3 equ xdos
0025		99	mon4 equ xdos
----		100	code ends
		101	end

ASSEMBLY COMPLETE, NO ERRORS FOUND

COPYR.GHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____