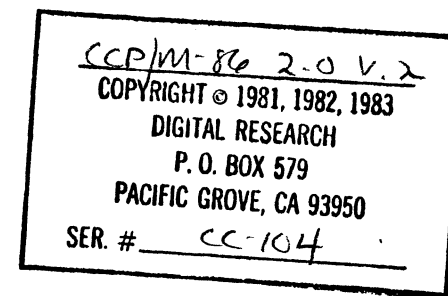


```

1
2      ;*****
3      ;*
4      ;*      Terminal Message Processor
5      ;*
6      ;*      The TMP determines the user interface to CCP/M.
7      ;*      Much of the interface is available through
8      ;*      system calls. This TMP takes advantage of
9      ;*      as much as possible for simplicity. The TMP
10     ;*      could, for instance, be easily modified to
11     ;*      force logins and have non-standard defaults.
12     ;*
13     ;*      With a little more work, the TMP could do all
14     ;*      command parsing and File Loading instead of
15     ;*      using the CLI COMMAND FUNCTION.
16     ;*      Suggestions are given in the CCP/M-86 SYSTEM'S GUIDE.
17     ;*
18     ;*****
19
20
21     title "Terminal Message Processor - CCP/M-86 2.0"
22
23     ;      Some common equates
24
25     00FF      true      equ      0ffh
26     0000      false     equ      0
27     000D      cr        equ      13      ; carriage return
28     000A      lf        equ      10      ; linefeed
29     0009      tab       equ      9       ; tab char
30
31
32     ;      CCP/M-86 system functions used by the TMP
33
34     00E0      osint      equ      224     ; interrupt number for CCP/M
35                                     ; system calls
36
37     0002      c_write     equ      2       ; console functions
38     0009      c_writebuf  equ      9
39     000A      c_readbuf   equ      10
40     0092      c_attachc   equ      146
41     0093      c_detachc   equ      147
42     0094      c_setnum    equ      143
43
44     00A0      l_setnum     equ      160     ; list device functions
45     00A4      l_getnum    equ      164
46
47     000F      f_open      equ      15     ; file functions
48     0010      f_close     equ      16
49     0014      f_read      equ      20
50     001A      f_setdma    equ      26
51     0098      f_parse     equ      152
52
53     000E      drv_set     equ      14     ; drive functions

```



```

54
55      0019      drv_get      equ      25
56      0027      drv_free     equ      39
57
58      0020      dir_usernum   equ      32      ; directory functions
59
60      0096      p_cli         equ      150     ; process control functions
61
62
63      ;          Process descriptor flags
64
65      0000      ps_run        equ      00      ; on ready list root
66      0001      pf_sys        equ      001h    ; system process
67      0002      pf_keep       equ      002h    ; do not terminate
68
69
70      ;          Some locations in the system data segment
71
72      0040      s_ccpmseg      equ      word ptr 40h ;begin CCPM segment
73      004B      s_sysdisk     equ      byte ptr 04bh ;system disk
74      0047      s_ncns        equ      byte ptr 47h ;sys. consoles
75      0078      s_version     equ      word ptr 78h ;ofst ver. str in SUP
76
77
78      ;          Some RSP format equates
79
80      0000      rsp_top        equ      0
81      0008      rsp_nd         equ      008h
82      0010      rsp_pd         equ      010h
83      0040      rsp_uda        equ      040h
84      0140      rsp_bottom     equ      140h
85
86
87      ;          Error codes returned by the CLI
88
89      0003      e_no_memory     equ      3      ; cant find memory
90      000C      e_no_pd        equ      12     ; no free pd's
91      000F      e_q_full       equ      15     ; full queue
92      0017      e_illdisk      equ      23     ; illegal disk #
93      0018      e_badfname     equ      24     ; illegal filename
94      0019      e_badftype     equ      25     ; illegal filetype
95      001C      e_bad_load     equ      28     ; bad ret. from BDOS load
96      001D      e_bad_read     equ      29     ; bad ret. from BDOS read
97      001E      e_bad_open     equ      30     ; bad ret. from BDOS open
98      001F      e_nullcmd      equ      31     ; null command sent
99      0025      e_ill_lst      equ      37     ; illegal list device
100     0026      e_ill_passwd    equ      38     ; illegal password
101     0028      e_abort        equ      40     ; aborted in CLI
102
103
104     ;*****
105     ;*
106     ;*      TMP Shared Code and Constant Area

```

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

SER. # _____

```

107
108
109
110
111
112
113 0000 EB2F      0031      cseg
114 0002 434F50595249      org      0
115      474854202863      jmps tmp
116      292031393833      db "COPYRIGHT (c) 1983, DIGITAL RESEARCH 3/28/83."
117      2C2044494749
118      54414C205245
119      534541524348
120      2020332F3238
121      2F38332E20
122
123      ;===
124      tmp:      ; PROGRAM MAIN - INITIALIZATION
125      ;===
126
127
128      ;      Set default console # = TMP#
129
130 0031 8A160400E81E 0356      mov dl,defconsole l call setconsolenum
131      03      0356
132
133
134      ;      Set default disk = system drive
135
136 0038 1E8E1E0000      push ds l mov ds,sysdatseg
137 003D 8A164B001F      mov dl,.s_sysdisk l pop ds      ;get system drive from
138 0042 E81503      035A      call setdisk      ;system data segment
139
140 0045 32D2      xor dl,dl      ;all TMPs come up user 0
141 0047 E8F602      0340      call setuser
142
143 004A E81D03      036A      call attach      ;print version
144 004D 1E8E1E0000      push ds l mov ds,sysdatseg
145 0052 8B167800      mov dx,.s_version
146 0056 8E1E4000      mov ds,.s_ccpmseg
147 005A E8F5021F      0352      call print_ds_string l pop ds
148 005E E80D03      036E      call detach
149
150 0061 1E07      push ds l pop es
151 0063 BE1B00      mov si,offset pd_ascii_num
152 0066 BF0301      mov di,offset startupnum
153 0069 B90300      mov cx,3
154 006C F3A4      rep movsb
155
156 006E BACA01      mov dx,offset fcb
157 0071 B10F      mov cl,f_open      ;try to open the startup file
158 0073 E80203      0378      call ccpm      ;on default drive which is
159 0076 3CFF      cmp al,0ffh      ;the system drive

```

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

SER. # _____

```

160
161 0078 744C      00C6      je nostartup
162 007A BA4201      mov dx,offset clicb_cmd      ;use the CLI buffer for this
163 007D 611A      mov cl,f_setdma      ;one time one sector read
164 007F E8F602      0378      call ccpm
165 0082 BACA01      mov dx,offset fcb
166 0085 8114      mov cl,f_read
167 0087 E8EE02      0378      call ccpm
168 008A 50      push ax
169 008B BACA01      mov dx,offset fcb
170 008E 8110      mov cl,f_close
171 0090 E8E502      0378      call ccpm
172 0093 58      pop ax
173 0094 84C0      test al,al
174 0096 752E      00C6      jnz nostartup
175 0098 8CD8      mov ax,ds
176 009A 8EC0      mov es,ax
177 009C 800D      mov al,cr
178 009E B98000      mov cx,128
179 00A1 8F4201      mov di,offset clicb_cmd
180 00A4 F2AE      repne scasb
181 00A6 751E      00C6      jne nostartup      ;didn't find a carriage return
182 00A8 47      inc di      ;include cr lf in line
183 00A9 C60524      mov byte ptr [di],'$'
184 00AC 81EF4201      sub di,offset clicb_cmd
185 00B0 88C7      mov ax,di
186 00B2 200200      sub ax,2
187 00B5 A24101      mov read_blen, al
188 00B8 BACB04      mov dx,offset supmsg
189 00BB E88A02      0348      call printstring
190 00BE BA4201      mov dx,offset clicb_cmd
191 00C1 E88E02      0352      call print_ds_string
192 00C4 EB3F      0105      jmps startup
193      nostartup:
194
195      ; THIS IS WHERE A LOGIN ROUTINE MIGHT
196      ; BE IMPLEMENTED. THE DATA FILE THAT
197      ; CONTAINS THE USER NAME AND PASSWORD
198      ; MIGHT ALSO CONTAIN AN INITIAL DEFAULT
199      ; DISK AND USER NUMBER FOR THAT USER.
200
201      ;=====
202      nextcommand:      ; LOOP FOREVER
203      ;=====
204
205      ; free drive
206 00C6 BAFFFFE8AA02 0376      mov dx,0ffffh ! call drive_free
207
208      ; attach console
209 00CC E89802      036A      call attach
210
211      ; print CR,LF if we just sent command
212 00CF 803EC3010074 00DE      cmp cmdsent,false ! je noclearline

```

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

```

213
214      08      000E
215      00D6 C606C30100c      mov cmdsent,false
216      00DB E86702      0345      call crlf
217      noclearline:
218
219      ; set up and print user prompt
220      ; get current default user # and disk
221      ; this call should be made on every
222      ; loop in case the last command
223      ; has changed the default.
224
225      00DE B20DE85602      0339      mov dl,cr | call prchar
226      00E3 E85802      033E      call getuser
227
228      00E6 84DB7405      00EF      test bl,bl | jz nozero      ;don't print user 0 prompt
229      00EA 8A03E83802      0327      mov dl,bl | call prnum
230      nozero:
231      00E7 E86C02      035E      call getdisk
232      00F2 824102D3      mov dl,"A" | add dl,bl
233      00F6 E84002      0339      call prchar
234      00F9 B8B803      mov dx,offset prompt
235      00FC E84902      0348      call print_string
236
237      ; Read Command from Console
238      00FF BA4001E86D02      0372      mov dx,offset read_buf | call conreadbuf
239
240      startup:
241      ; echo newline
242      0105 820AE82F02      0339      mov dl,lf | call prchar
243
244      ; make sure not a null command
245      010A 8D1E4201      lea bx,clicb_cmd
246      010E 803E41010074      0178      cmp read_blen,0 | je gonextcmd
247      63      0178
248      deblank:
249      0115 803F207405      011F      cmp byte ptr [bx],"" | je zapblank
250      011A 803F097507      0126      cmp byte ptr [bx],tab | jne noblanks
251      zapblank:
252      011F 43FE0E4101EB      0115      inc bx | dec read_blen | jmps deblank
253      EF      0115
254      noblanks:
255      0126 800642013BC3      013F      lea ax,clicb_cmd | cmp ax,bx | je chksemi
256      7411      013F
257
258      ; remove leading blanks
259      012E 1E0732ED8A0E      push ds | pop es | xor ch,ch | mov cl,read_blen
260      4101
261      0136 8BF88BF3FCF3      mov di,ax | mov si,bx | cld | rep movsb
262      A4
263      013D 8BD8      mov bx,ax
264      chksemi:
265      013F 803F387434      0178      cmp byte ptr [bx],";" | je gonextcmd

```

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

```

266
267
268 ; see if disk change
269 0144 803E41010275 0178 ; if "X:" change def disk to X
270 30 0178 cmp read_blen,2 ! jne clicall
271 014B 807F013A cmp byte ptr 1[ebx],":"
272 014F 752A 0178 jne clicall
273
274 ; change default disk
275 0151 8A17 mov dl,[ebx] ;get disk name
276 0153 80E25F and dl,5fh ;Upper Case
277 0156 80EA41 sub dl,"A" ;disk number
278
279 ; check bounds
280 0159 80FA00720B 0169 cmp dl,0 ! jb baddrive
281 015E 80FA0F7706 0169 cmp dl,15 ! ja baddrive
282
283 ; select default disk
284 0163 E8F401E90F00 035A call setdisk ! jmp gonextcmd
285
286 0169 BAC003E8D901 0348 baddrive: mov dx,offset errstr ! call printstring
287 016F BA3804E8D301 0348 mov dx,offset driverr ! call printstring ! call crlf
288 E8CD01 0348
289
290 0178 E948FF 00C6 gonextcmd: jmp nextcommand
291
292 ;=====
293 clicall: ; SEND CLI COMMAND
294 ;=====
295
296 017B 8B4201 ; put null at end of input
297 017E A04101B400 mov bx,offset clicb_cmd
298 0183 03D8C60700 mov al,read_blen ! mov ah,0
299 add bx,ax ! mov byte ptr [bx],0
300
301 ; copy command string for error
302 ; reporting later and to check
303 ; for built in commands...
304 0188 B94000 mov cx,64
305 018B BE4201 mov si,offset clicb_cmd
306 018E BFE801 mov di,offset savebuf
307 0191 1E07 push ds ! pop es
308 0193 F3A5 rep movsw
309
310 ; parse front to see if
311 ; built in command
312 0195 BECA01 mov si,offset fcb
313 0198 BFE801 mov di,offset savebuf
314 019E E310 02D0 call parsefilename
315 01A0 2BD98A1E4101 0180 jcxz goodparse
316 01A6 81C3E801 sub bx,bx ! mov bl,read_blen
317 01AA C60724 add bx,offset savebuf
318 01AD E9E300 0293 mov byte ptr [bx],"$"
jmp clierror

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

319
320
321
322 0180 891EC401          goodparse:      mov parseret,bx
323 0184 83F8007508      01C1             cmp bx,0 l jne haveatail
324 0189 8A1E4101          mov bl,read_blen
325 018D 81C3E801          add bx,offset savebuf
326 01C1 C6072443          haveatail:    mov byte ptr [bx],"$" l inc bx
327 01C5 803ECA010074      01CF             cmp fcb,0 l je try_builtin
328 03                      01CF
329 01CC E9A900            0278             jmp not_builtin
330                                     ; is it USER command?
331
332 01CF BECA0146          try_builtin:   mov si,offset fcb l inc si
333 01D3 BF8B04            mov di,offset usercmd
334 01D6 0E07             push cs l pop es
335 01D8 B90400F3A7        mov cx,4 l repz cmpsw
336 01DD 7545              0224             jnz notuser
337 01DF BECA01            mov si,offset fcb
338 01E2 8B3EC401          mov di,parseret
339 01E6 83FF007425        0210             cmp di,0 l je pruser
340 01E8 47                inc di
341 01EC E8EE00            02D3             call parsefilename
342 01EF 83F900751C        0210             cmp cx,0 l jne pruser
343 01F4 BECA01            mov si,offset fcb
344 01F7 46                inc si
345 01F8 8B14              mov dx,[si]
346 01FA E8F000            02ED             call a_to_b
347 01FD 80F80F7708        020A             cmp bl,15 l ja usererr
348 0202 8AD3              mov dl,bl
349 0204 E83901            0340             call setuser
350 0207 E90600            0210             jmp pruser
351 020A 6A5104            usererr:      mov dx,offset usererrmsg
352 020D E83801            0348             call printstring
353 0210 BA7204            pruser:      mov dx,offset usermsg
354 0213 E83201            0348             call printstring
355 0216 E82501            033E             call getuser
356 0219 8AD3E80901        0327             mov dl,bl l call prnum
357 021E E82401            0345             call crlf
358 0221 E9A2FE            00C6             jmp nextcommand
359
360 0224 BECA0146          notuser:      mov si,offset fcb l inc si
361 0228 8FC304            mov di,offset printercmd
362 022B 0E07             push cs l pop es
363 022D B90400F3A7        0278             mov cx,4 l repz cmpsw
364 0232 7544              jnz notprinter
365 0234 BECA01            mov si,offset fcb
366 0237 8B3EC401          mov di,parseret
367 023B 83FF007424        0264             cmp di,0 l je prprinter
368 0240 47                inc di
369 0241 E89900            02DD             call parsefilename
370 0244 83F900751B        0264             cmp cx,0 l jne prprinter
371 0249 BECA01            mov si,offset fcb

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

372
373 024C 46 inc si
374 024D 8B14 mov dx,cs:1
375 024F E89800 call a_to_b
376 0252 80FBFF cmp bl,0ffh
377 0255 7407 je printererr
378 0257 8A03 mov dl,bl
379 0259 E80601 call setlist
380 025C E306 jcxz prprinter
381 025E BA8304 printererr: mov dx,offset printensg
382 0261 E8E400 call printstring
383 0264 BAA704 prprinter: mov dx,offset printermgs
384 0267 E8DE00 call printstring
385 026A E8F900 call getlist
386 026D 8A03E8B500 mov dl,bl | call prnum
387 0272 E8D000 call crlf
388 0275 E94EFE jmp nextcommand
389
390 notprinter:
391 not_builtin:
392
393 0278 C606410100 ; initialize Cli Control Block
394 mov clicb_net,0
395 ; make cli call
396
397 027D C606C301FF mov cmdsent,true
398 0282 8D164101B196 lea dx,clicb | mov cl,p_cli
399 0288 E8ED00 call ccpm
400 028B 83F8007503 cmp bx,0 | jne clierror
401 0290 E933FE jmp nextcommand
402
403 clierror:
404 ;=====
405 ; Cli call unsuccessful, analyze and display err msg
406 ; input: CX = ERROR CODE
407
408 0293 BE7703 mov si,(offset clierrtab)-4
409 nexterr:
410 0296 83C604 add si,4
411 0299 2E838C0000FF 02A8 cmp cs:word ptr [si],0ffffh | je unknownerr
412 7407 02A8
413 02A1 2E388C000075 0296 cmp cx,cs:[si] | jne nexterr
414 EE 0296
415 unknownerr:
416 02A8 2E8B940200 mov dx,cs:2[si]
417 ; jmps showerr
418
419
420 showerr: ; Print Error String
421 ;----- ; input: DX = address of Error
422 ; string in CSEG
423 ; if DX=0 then NULL COMMAND
424

```

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

SER. # _____


```

425
426 02AD 83FA007509 02BA      cmp dx,0 ! jne perr
427 02B2 C606C30100E9 00C6      mov cmdsent,false ! jmp nextcommand
428      0CFE      00C6
429 02BA 52E88700      0345 perr:  push dx ! call crlf
430 02BE BAC003E88400 0348      mov dx,offset errstr ! call printstring
431 02C4 5AE88000E87A 0348      pop dx ! call printstring ! call crlf
432      00      0348
433 02CB 8A4604E87700 0348      mov dx,offset cmdstr ! call printstring
434 02D1 8AE801E87800 0352      mov dx,offset savebuf ! call print_ds_string ! call crlf
435      E86800      0352
436 02DA E9E9FD      00C6      jmp nextcommand
437
438      parsefilename:  ; SI = fcb  DI = string
439 02D0 B99800      mov cx,f_parse
440 02E0 88C601      mov bx,offset pcb
441 02E3 893F897702      mov [bx],di ! mov 2[bx],si
442 02E8 8B03E98800      0378      mov dx,bx ! jmp ccpm
443      a_to_b:
444 02ED 80FE207504      02F6      ;dl = 1st char, dh = 2nd char
445 02F2 8AF28230      cmp dh," " ! jne atob2char
446 02F6 80FE307229      0324      mov dh,dl ! mov dl,"0"
447 02FB 80FE397724      0324      atob2char:  cmp dh,"0" ! jb atoberr
448 0300 80FA30721F      0324      cmp dh,"9" ! ja atoberr
449 0305 80FA39771A      0324      cmp dl,"0" ! jb atoberr
450 030A 80EE3080EA30      cmp dl,"9" ! ja atoberr
451 0310 B800008AC2      sub dh,"0" ! sub dl,"0"
452 0315 52B10A      mov ax,0 ! mov al,dl
453 0318 F6E15A      push dx ! mov cl,10
454 031B 8AD68600      mul cl ! pop dx
455 031F 03C2      mov dl,dh ! mov dh,0
456 0321 8E08C3      add ax,dx
457 0324 B3FFC3      mov bx,ax ! ret
458      atoberr:      mov bl,0ffh ! ret
459 0327 80FA0A720A      0336      prnum:      ; dl = num (0-15)
460 032C 52      cmp dl,10 ! jb prnum_one
461 032D B231E80700      0339      push dx
462 0332 5A80EA0A      mov dl,"1" ! call prchar
463 0336 80C230      0336      pop dx ! sub dl,10
464      prnum_one:  add dl,"0"
465 0339 B102E93A00      0378      ; jmp prchar
466 033E B2FF      mov cl,c_write ! jmp ccpm
467 0340 B120E93300      0378      getuser:  mov dl,0ffh
468 0345 BABD03      0378      setuser:  mov cl,dir_usernum ! jmp ccpm
469      crlf:      mov dx,offset crlfstr
470 0348 1E8CC88ED8      ;jms printstring
471 034D E802001FC3      0352      printstring:  push ds ! mov ax,cx ! mov ds,ax
472 0352 B109EB22      0378      call print_ds_string ! pop ds ! ret
473 0356 B194EB1E      0378      print_ds_string:  mov cl,c_writebuf ! jms ccpm
474 035A B10EEB1A      0378      setconsolenum:  mov cl,c_setnum ! jms ccpm
475 035E B119EB16      0378      setdisk:  mov cl,drv_set ! jms ccpm
476 0362 B1A0EB12      0378      getdisk:  mov cl,drv_get ! jms ccpm
477 0366 B1A4EB0E      0378      setlist:  mov cl,l_setnum ! jms ccpm
478      getlist:  mov cl,l_getnum ! jms ccpm

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

478
479 036A B192EB0A      0378 attach:      mov cl,c_attachc ! jmps ccpm
480 036E B193EB06      0378 detach:     mov cl,c_detachc ! jmps ccpm
481 0372 B104EB02      0378 con_readbuf:  mov cl,c_readbuf ! jmps ccpm
482 0376 B127          drivefree:      mov cl,drv_free ! jmps ccpm
483
484                ;====
485                ccpm:      ; INTERFACE ROUTINE FOR SYSTEM ENTRY POINTS
486                ;====
487
488 0378 CDE0C3          int osint ! ret
489
490                ;*****
491                ;*
492                ;*      CONSTANTS (IN SHARED CODE SEGMENT)
493                ;*
494                ;*****
495
496 037B 1F000000      clierrtab      dw      e_nullcmd,      0      ;null command
497 037F 0300CD03      dw      e_no_memory,      memerr ;No memory
498 0383 0C00DF03      dw      e_no_pd,      pderr ;No unused PD
499 0387 1800ED03      dw      e_badfname,      fnameerr;Ill. command
500 038B 1700ED03      dw      e_illdisk,      fnameerr;Ill. disk
501 038F 2600ED03      dw      e_ill_passwd,      fnameerr;Ill. password
502 0393 1900ED03      dw      e_badftype,      fnameerr;Ill. type
503 0397 1C00FB03      dw      e_bad_load,      loaderr ;
504 039B 1D00FB03      dw      e_bad_read,      loaderr ;
505 039F 1E000604      dw      e_bad_opens,      openerr ;
506 03A3 0F001904      dw      e_q_full,      qfullerr;
507 03A7 28002E04      dw      e_abort,      aborterr;
508
509                ; a few extra entries for future errors
510
511 03AB FFFFFB03      dw      0ffffh,      catcherr;
512 03AF FFFFFB03      dw      0ffffh,      catcherr;
513 03B3 FFFFFB03      dw      0ffffh,      catcherr;
514 03B7 FFFFFB03      dw      0ffffh,      catcherr;
515
516 03BB 3E24          prompt      db      ">$"
517 03BD 0D0A24          crlfstr      db      13,10,"$"
518 03C0 43502F4D2045      errstr      db      "CP/M Error: $"
519 72726F723A20
520 24
521 03CD 4E6F7420456E      memerr      db      "Not Enough Memory$"
522 6F756768204D
523 656D6F727924
524 03DF 504420546162      pderr      db      "PD Table Full$"
525 6C652046756C
526 6C24
527 03ED 426164204669      fnameerr      db      "Bad File Spec$"
528 6C6520537065
529 6324
530 03FB          catcherr      rb      0      ;Unknown Errs give

```

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

SER. # _____

```

531
532 03FB 4C6F61642045 loaderr db "Load Error$" ; Load Error Msg
533 72726F7224
534 0406 43616E277420 openerr db "Can't Find Command$"
535 46696E642043
536 6F6D6D616E64
537 24
538 0419 52535020436F qfullerr db "RSP Command Que Full$"
539 6D6D616E6420
540 517565204675
541 6C6C24
542 042E 434C49204162 aborterr db "CLI Abort$"
543 6F727424
544 0438 496E76616C69 driverr db "Invalid Drive$"
545 642044726976
546 6524
547
548 0446 436F6D6D616E cmdstr db "Command = $"
549 64203D2024
550 0451 0D0A496E7661 usererrmsg db 13,10,"Invalid User Number,"
551 6C6964205573
552 6572204E756D
553 6265722C
554 0467 2049474E4F52 db " IGNORED",13,10,"$"
555 45440D0A24
556 0472 0D0A55736572 usermsg db 13,10,"User Number = $"
557 204E756D6265
558 72203D2024
559
560 0483 0D0A496E7661 printerr db 13,10,"Invalid Printer Number,"
561 6C6964205072
562 696E74657220
563 4E756D626572
564 2C
565 049C 2049474E4F52 db " IGNORED",13,10,"$"
566 45440D0A24
567 04A7 0D0A5072696E printerrmsg db 13,10,"Printer Number = $"
568 746572204E75
569 6D626572203D
570 2024
571
572 048B 555345522020 usercmd db "USER "
573 2020
574 04C3 5052494E5445 printercmd db "PRINTER "
575 5220
576
577 04CB 537461727420 supmsg db "Start up command: $"
578 757020636F6D
579 6D616E643A20
580 24
581
582 ;*****
583 ;*
```

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

```

584
585      ;*      TMP Data Area - this area is copied once for
586      ;*      each system console. The "defconsole"
587      ;*      field is unique for each copy
588      ;*      - Each Data Area is run by a common
589      ;*      shared code segment.
590      ;*
591      ;*****
592
593      JSEG
594      org      rsp_top
595
596      0000 0000      sysdatseg      dw      0
597      0002 4700      sdatvar        dw      s_hcns
598      0004 0000      defconsole     db      0,0
599      0006 000000000000      dw      0,0,0,0,0
600      00000000
601
602      org      rsp_pd
603
604      0010 00000000      pd          dw      0,0      ; link fields
605      0014 00          db      ps_run      ; status
606      0015 C6          db      198        ; priority
607      0016 0300          dw      pf_sys+pf_keep ; flags
608      0018 546070          db      "Tmp"    ; Name
609      0018 20202020      pd_ascii_num db      " " ; Ascii number field set by GENSYS
610      0020 0400          dw      offset uda/10h ; uda seg
611      0022 0000          db      0,0      ; disk,user
612      0024 0000          db      0,0      ; ldisk,luser
613      0026 FFFF          dw      0ffffh    ; mem
614      0028 00000000      dw      0,0      ; dvdract,wait
615      002C 0000          db      0,0      ; org,net
616      002E 0000          dw      0        ; parent
617      0030 0000          db      0,0      ; cns,abort
618      0032 0000          db      0,0      ; cin,cout
619      0034 0000          db      0,0      ; lst,sf3
620      0036 0000          db      0,0      ; sf4,sf5
621      0038 00000000      dw      0,0      ; reserved
622      003C 00000000      dw      0,0      ; pret,scratch
623
624      org      rsp_uda
625
626      0040 000000000000      uda      dw      0,0,0,0      ;0-7 note: no default DMA
627      0000
628      0048 000000000000      dw      0,0,0,0      ;8-fh
629      0000
630      0050 000000000000      dw      0,0,0,0      ;10-17
631      0000
632      0058 000000000000      dw      0,0,0,0      ;18-1f
633      0000
634      0060 000000000000      dw      0,0,0,0      ;20-27
635      0000
636      0068 000000000000      dw      0,0,0,0      ;28-2f

```

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

```
637
638      0000
639 0070 000000003A01      dw      0,0,offset stack_top,0 ;30-37
640      0000
641 0078 000000000000      dw      0,0,0,0 ;38-3f
642      0000
643 0080 000000000000      dw      0,0,0,0 ;40-47
644      0000
645 0088 000000000000      dw      0,0,0,0 ;48-4f
646      0000
647 0090 000000000000      dw      0,0,0,0 ;50-57
648      0000
649 0098 000000000000      dw      0,0,0,0 ;58-5f
650      0000
651 00A0 01                db      1 ;60 INSYS <> 0
652                                     ;don't switch from
653                                     ;from UDA stack
654                                     ;on entry to SUP
655      00A1 00                db      0 ;61
656      00A2 00000000      dw      0,0 ;62-64
657      00A6 00                db      0 ;66
658      00A7 0000      dw      0 ;67-68
659      00A9 00                db      0 ;69
660
661 00AA CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc ;6A-6F
662 00B0 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;70
663      CCCC
664 00B8 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
665      CCCC
666 00C0 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;80
667      CCCC
668 00C8 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
669      CCCC
670 00D0 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;90
671      CCCC
672 00D8 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
673      CCCC
674 00E0 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;A0
675      CCCC
676 00E8 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
677      CCCC
678 00F0 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;B0
679      CCCC
680 00F8 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
681      CCCC
682 0100 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;C0
683      CCCC
684 0108 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
685      CCCC
686 0110 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc ;D0
687      CCCC
688 0118 CCCCCCCCCCCC      dw      0cccc,0cccc,0cccc,0cccc
689      CCCC
```

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

```

690
691 0120 CCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch,0ccccch      ;E0
692      CCCC
693 0128 CCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch,0ccccch
694      CCCC
695 0130 CCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch,0ccccch      ;F0
696      CCCC
697 0138 CCCC            dw      0ccccch
698
699 013A 3100      stack_top      dw      offset tmp      ; code starting point
700 013C 0000      dw      0      ; code seg - set by GENSY
701 013E 0000      dw      0      ; init. flags - set by GENSY
702
703      ; UDA is 100H bytes long
704
705 0080      maxcmdlen      equ      128
706
707      ; the Read Console Buffer and the
708      ; CLI Control Block share the same memory
709
710 0140      read_buf      rb      0
711 0140 80      read_maxcmd      db      128
712 0141      clicb      rb      0
713 0141      clicb_net      rb      0
714 0141      read_blen      rb      1
715 0142      clicb_cmd      rb      maxcmdlen + 1
716
717 01C3 00      cmdsent      db      false
718
719 01C4 0000      parseret      dw      0
720
721 01C6 EB01      pcb      dw      offset savebuf
722 01C8 CA01      dw      offset fcb
723
724 01CA 005354415254      fcb      db      0, "STARTUP "
725      555020
726 01D3 202020      startupnum      db      " "
727 01D6      rb      20
728 01EA 00      db      0      ;current record
729
730 01EB      savebuf      rb      128
731
732 0268 00      db      0      ;ensure hex is formed
733      end
734
735
736 END OF ASSEMBLY.  NUMBER OF ERRORS:  0.  USE FACTOR:  9%

```

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

SER. # _____

ABORTERR	042E V	507	542#																
ATOB	02ED L	346	375	443#															
ATOB2CHAR	02F6 L	444	446#																
ATOBERR	0324 L	446	447	448	449	457#													
ATTACH	036A L	143	209	479#															
BADDRIVE	0169 L	280	281	286#															
CATCHERR	03FB V	511	512	513	514	530#													
CATTACHC	0092 N	40#	479																
CCPM	0378 L	158	164	167	171	398	442	465	467	472	473								
		474	475	476	477	479	480	481	485#										
CDETACHC	0093 N	41#	480																
CHKSEMI	013F L	255	263#																
CLICALL	017B L	269	272	293#															
CLICB	0141 V	397	712#																
CLICBCMD	0142 V	162	179	184	190	245	255	296	304	715#									
CLICBNET	0141 V	393	713#																
CLIERROR	0293 L	318	399	403#															
CLIERRTAB	0378 V	408	496#																
CMDSENT	01C3 V	212	215	396	427	717#													
CMDSTR	0446 V	433	548#																
CUNREADBUF	0372 L	238	481#																
CR	000D N	27#	177	225															
CREADBUF	000A N	39#	481																
CRLF	0345 L	216	287	357	387	429	431	434	468#										
CRLFSTR	038D V	468	517#																
CS	SREG V	334	362	411	413	416	470												
CSETNUM	0094 N	42#	473																
CWRITE	0002 N	37#	465																
CWRITEBUF	0009 N	38#	472																
DEBLANK	0115 L	248#	252																
DEFCUNSOLE	0004 V	130	598#																
DETACH	036E L	148	480#																
DIRUSERNUM	0020 N	58#	467																
DRIVEFREE	0376 L	206	482#																
DRVERR	0438 V	287	544#																
DRVFREE	0027 N	56#	482																
DRVGET	0019 N	55#	475																
DRVSET	000E N	53#	474																
DS	SREG V	136	136	137	144	144	146	147	150	175	258								
		306	470	470	471														
EABORT	0028 N	101#	507																
EBADFNAME	0018 N	93#	499																
EBADFTYPE	0019 N	94#	502																
EBADLOAD	001C N	95#	503																
EBADOPEN	001E N	97#	505																
EBADREAD	001D N	96#	504																
EILLDISK	0017 N	92#	500																
EILLST	0025 N	99#																	
EILLPASSWD	0026 N	100#	501																
ENOMEMORY	0003 N	89#	497																
ENOPD	000C N	90#	498																
ENULLCMD	001F N	98#	496																
EQFULL	000F N	91#	506																

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ERRSTR	03C0 V	286	430	518#							
ES	SREG V	150	176	258	306	334	362				
FALSE	0000 N	26#	212	215	427	717					
FCB	01CA V	156	165	169	311	327	332	337	343	360	365
		371	722	724#							
FCLOSE	0010 N	48#	170								
FNAMEERR	03ED V	499	500	501	502	527#					
FOPEN	000F N	47#	157								
FPARSE	0098 N	51#	439								
FREAD	0014 N	49#	166								
FSETDMA	001A N	50#	163								
GETDISK	035E L	231	475#								
GETLIST	0366 L	385	477#								
GETUSER	033E L	226	355	466#							
GONEXTCMD	0178 L	246	264	284	290#						
GUDDPARSE	0180 L	314	322#								
HAVEATAIL	01C1 L	323	326#								
LF	000A N	28#	242								
LGETNUM	00A4 N	45#	477								
LOADERR	03FB V	503	504	532#							
LSETNUM	00A0 N	44#	476								
MAXCMDLEN	0080 N	705#	715								
MEMERR	03CD V	497	521#								
NEXTCOMMAND	00C6 L	202#	290	358	388	400	427	436			
NEXTERR	0296 L	409#	413								
NOBLANKS	0126 L	250	254#								
NUCLEARLINE	00DE L	212	217#								
NOSTARTUP	00C6 L	161	174	181	193#						
NOIBUILTIN	0278 L	329	390#								
NOIPRINTER	0278 L	364	389#								
NOUSER	0224 L	336	359#								
NOZERO	00EF L	228	230#								
OPENERR	0406 V	505	534#								
OSINT	00E0 N	34#	488								
PARSEFILENAME	02DD L	313	341	369	438#						
PARSERET	01C4 V	322	338	366	719#						
PCB	01C6 V	440	721#								
PCLT	0096 N	60#	397								
PD	0010 V	604#									
PDASCIINUM	0018 V	151	609#								
PDERR	03DF V	498	524#								
PERR	028A L	426	429#								
PFKEEP	0002 N	67#	607								
PFSYS	0001 N	66#	607								
PRCHAR	0339 L	225	233	242	461	465#					
PRINTDSSTRING	0352 L	147	191	434	471	472#					
PRINTMSG	0483 V	381	560#								
PRINTERCMD	04C3 V	361	574#								
PRINTERERR	025E L	377	381#								
PRINTMSG	04A7 V	383	567#								
PRINTSTRING	0348 L	189	235	286	287	352	354	382	384	430	431
		433	470#								
PRNUM	0327 L	229	356	386	458#						

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____