```
                        eject 1 include cpyright.def
                        ;*******************************************************
                        ;*                                                     *
                        ;*               Concurrent CP/M-86 V2.1               *
                        ;*               =======================               *
                        ;*                                                     *
                        ;*               Copyright (c) 1982                    *
                        ;*                                                     *
                        ;*                  Digital Research                   *
                        ;*                    P.O.Box 579                      *
                        ;*          Pacific Grove, California 93950            *
                        ;*                                                     *
                        ;*                   (408) 649-3896                    *
                        ;*                   TWX 9103605001                    *
                        ;*                                                     *
                        ;* All information contained in this source listing is *
                        ;*                                                     *
                        ;*                     PROPRIETORY                     *
                        ;*                     ===========                     *
                        ;*                                                     *
                        ;* All rights reserved.  No part of this document      *
                        ;* may be reproduced, transmitted, stored in a         *
                        ;* retrieval system, or translated into any language   *
                        ;* or computer language, in any form or by any means   *
                        ;* without the prior written permission of Digital     *
                        ;* Research, P.O Box 579, Pacific Grove, California.   *
                        ;*                                                     *
                        ;*******************************************************


                        ;*******************************************************
                        ;*                                                     *
                        ;*      System Data Area - Initialized Data            *
                        ;*                                                     *
                        ;*******************************************************
```

```
                        eject 1 include system.def
                        ;*******************************************************
                        ;*                                                     *
                        ;*       SYSTEM DEFINITIONS                            *
                        ;*                                                     *
                        ;*******************************************************
                        
FFFF                    true        equ  0ffffh     ; value of TRUE
0000                    false       equ  0          ; value of FALSE
0000                    unknown     equ  0          ; value to be filled in
0080                    dskrecl     equ  128        ; log. disk record len
0020                    fcblen      equ  32         ; size of file control block
0008                    pnamsiz     equ  8          ; size of process name
0008                    qnamsiz     equ  pnamsiz    ; size of queue name
0008                    fnamsiz     equ  pnamsiz    ; size of file name
0003                    ftypsiz     equ  3          ; size of file type
00E0                    osint       equ  224        ; int vec for O.S. entry
00E1                    debugint    equ  osint+1    ; int vec for debuggers
0100                    ulen        equ  0100h      ; size of uda
015E                    u8087len    equ  ulen + 94  ; size of uda when process uses 8087
0030                    pdlen       equ  030h       ; size of Process Descriptor
0005                    todlen      equ  5          ; size of Time of Day struct
0001                    flag_tick   equ  1          ; flag 0 = tick flag
0002                    flag_sec    equ  2          ; flag 1 = second flag
0003                    flag_min    equ  3          ; flag 2 = minute flag
00AA                    ldtabsiz    equ  0aah       ; ldtablen=11, 10 entries
                        
0000                    mpm     equ     false       ; MP/M-86 or
FFFF                    ccpm    equ     not mpm     ; CCP/M-86
                        
0000                    BCPM    equ     false       ; CP/M-86 BDOS
FFFF                    BMPM    equ     not BCPM    ; multi-process BDOS
```

```
                        eject 1 include pd.def
                        ;*******************************************************
                        ;*
                        ;*      Process Descriptor - with the UDA associated
                        ;*              with the PD, describes the current
                        ;*              state of a Process under MP/M-CCP/M
                        ;*
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 00|   link    |  thread   |stat |prior|   flag    |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 08|                       Name                    |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 10|    uda    | dsk | user| ldsk|luser|    mem    |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 18| cmod|tkcnt|   wait    | reserved  |  parent   |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 20| cns |abort|  conmode  | lst |     reserved    |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;* 28|       reserved        |   pret    |  scratch  |
                        ;*   +-----+-----+-----+-----+-----+-----+-----+-----+
                        ;*
                        ;*      link    - Used for placement into System Lists
                        ;*      thread  - link field for Thread List
                        ;*      stat    - Current Process activity
                        ;*      prior   - priority
                        ;*      flag    - process state flags
                        ;*      name    - name of process
                        ;*      uda     - Segment Adress of User Data Area
                        ;*      dsk     - Current default disk
                        ;*      user    - Current default user number
                        ;*      ldsk    - Disk program loaded from
                        ;*      luser   - User number loaded from
                        ;*      mem     - pointer to MD list of memory owned
                        ;*                by this process
                        ;*      cmod    - compatibility mode bits.
                        ;*                80 -> F1 bit
                        ;*                40 -> F2 bit
                        ;*                20 -> F3 bit
                        ;*                10 -> F4 bit
                        ;*      tkcnt   - temp keep count, # times tempkeep has been
                        ;*                turned on
                        ;*      wait    - parameter field while on System Lists
                        ;*      orj     - Network node that originated this process
                        ;*      net     - Network node running this process
                        ;*      parent  - process that created this process
                        ;*      cns     - default console #
                        ;*      abort   - abort code
                        ;*      lst     - default list #
                        ;*      conmode - console mode for function 109
                        ;*      reserved- not currently used
                        ;*      pret    - return code at termination
                        ;*      scratch - scratch word
```

```
                        ;*
                        ;*******************************************************
                        ;
0000                    p_link          equ     word ptr 0
0002                    p_thread        equ     word ptr p_link + word
0004                    p_stat          equ     byte ptr p_thread + word
0005                    p_prior         equ     byte ptr p_stat + byte
0006                    p_flag          equ     word ptr p_prior + byte
0008                    p_name          equ     byte ptr p_flag + word
0010                    p_uda           equ     word ptr p_name + pnamsiz
0012                    p_dsk           equ     byte ptr p_uda + word
0013                    p_user          equ     byte ptr p_dsk + byte
0014                    p_ldsk          equ     byte ptr p_user + byte
0015                    p_luser         equ     byte ptr p_ldsk + byte
0016                    p_mem           equ     word ptr p_luser + byte
0018                    p_cmod          equ     byte ptr p_mem + word
0019                    p_tkcnt         equ     byte ptr p_cmod + byte
001A                    p_wait          equ     word ptr p_tkcnt + byte
001E                    p_parent        equ     word ptr p_wait + 4
0020                    p_cns           equ     byte ptr p_parent + word
0021                    p_abort         equ     byte ptr p_cns + byte
0022                    p_conmode       equ     word ptr p_abort + byte
0024                    p_lst           equ     byte ptr p_conmode + word
002A                    p_pret          equ     word ptr p_lst + 8
002E                    p_scratch       equ     byte ptr p_pret + word
002E                    p_wscrtch       equ     word ptr p_scratch
                        ;
                        ;       Process descriptor pd_status values
                        ;
0000                    ps_run          equ     0       ; in ready list root
0001                    ps_poll         equ     1       ; in poll list
0002                    ps_delay        equ     2       ; in delay list
0003                    ps_swap         equ     3       ; in swap list
0004                    ps_term         equ     4       ; terminating
0005                    ps_sleep        equ     5       ; sleep processing
0006                    ps_dq           equ     6       ; in dq list
0007                    ps_nq           equ     7       ; in nq list
0008                    ps_flagwait     equ     8       ; in flag table
0009                    ps_ciowait      equ     9       ; waiting for character
000A                    ps_sync         equ     10      ; waiting for sync structure
                        ;
                        ;       Process descriptor pd_flag bit values
                        ;
0001                    pf_sys          equ     00001h  ; system process
0002                    pf_keep         equ     00002h  ; do not terminate
0004                    pf_kernal       equ     00004h  ; resident in kernal
0008                    pf_pure         equ     00008h  ; pure memory descibed
0010                    pf_table        equ     00010h  ; from pd table
0020                    pf_resource     equ     00020h  ; waiting for resource
0040                    pf_raw          equ     00040h  ; raw console i/o
```

```
                        pf_ctlc         equ     00080h  ; abort pending
                        pf_active       equ     00100h  ; active tty
                        pf_tempkeep     equ     00200h  ; don't terminate yet...
                        pf_ctld         equ     00400h  ; explicit detach occured
                        pf_childabort   equ     00800h  ; child terminated abnormally
                        pf_noctls       equ     01000h  ; control S not allowed
                        pf_dskld        equ     02000h  ; process was loaded from disk
                        pf_nokbd        equ     04000h  ; ignore keyboard
                        pf_8087         equ     08000h  ; process uses 8087
                        ;
                        ;       Process descriptor pcm_flag bit values
                        ;               (console mode set by function 109)
                        ;
                        pcm_11          equ     00001h  ; control C status for function 11
                        pcm_ctls        equ     00002h  ; disable control S-Q
                        pcm_rout        equ     00004h  ; raw output, no tabs or control P
                        pcm_ctlc        equ     00008h  ; disable control C
                        ;pcm_rsrv       equ     00040h  ; used by CP/M 3.0
                        pcm_ctlo        equ     00080h  ; disable control O
                        pcm_rsx         equ     00300h  ; 2 bits used by RSX for status
```

```
                    ;
                    ;
                    eject 1 include uda.def
                    ;
                    ;       UDA offsets
                    ;
0060                ud_insys        equ     60h
```

```
                        eject | include qd.def
                        ;*******************************************************
                        ;*
                        ;*      Queue Descriptor - This is structure is used
                        ;*              to create a queue.  One is maintained
                        ;*              in the system data area for each queue
                        ;*
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*   00 |  link   |net |org |  flags  |  name...
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*   08              ...name           | msglen  |
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*   10 |  nmsgs  |   dq    |   nq    | msgcnt  |
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*   18 | msgout  | buffer  |
                        ;*      +----+----+----+----+
                        ;*
                        ;*      link    - used to link QDs is system lists
                        ;*      net     - which machine in the network
                        ;*      org     - origin machine in the network
                        ;*      flags   - Queue Flags
                        ;*      name    - Name of Queue
                        ;*      msglen  - # of bytes in one message
                        ;*      nmsgs   - maximum # of messages in queue
                        ;*      dq      - Root of PDs waiting to read
                        ;*      nq      - Root of PDs list waiting to write
                        ;*      msgcnt  - # of messages currently in queue
                        ;*      msgout  - next message # to read
                        ;*      buf     - pointer to queue message buffer
                        ;*                (for MX queues, owner of queue)
                        ;*
                        ;*******************************************************
0000                    q_link          equ     word ptr 0
0002                    q_net           equ     byte ptr q_link + word
0003                    q_org           equ     byte ptr q_net + byte
0004                    q_flags         equ     word ptr q_org + byte
0006                    q_name          equ     byte ptr q_flags + word
000E                    q_msglen        equ     word ptr q_name + qnamsiz
0010                    q_nmsgs         equ     word ptr q_msglen + word
0012                    q_dq            equ     word ptr q_nmsgs + word
0014                    q_nq            equ     word ptr q_dq + word
0016                    q_msgcnt        equ     word ptr q_nq + word
001d                    q_msgout        equ     word ptr q_msgcnt + word
001A                    q_buf           equ     word ptr q_msgout + word
001C                    qdlen           equ     q_buf + word
                        ;
                        ;       Q_FLAGS values
                        ;
0001                    qf_mx           equ     001h    ; Mutual Exclusion
0002                    qf_keep         equ     002h    ; NO DELETE
0004                    qf_hide         equ     004h    ; Not User writable
```

```
                        qf_rsp          equ     008h    ; rsp queue
                        qf_table        equ     010h    ; from qd table
                        qf_rpl          equ     020h    ; rpl queue
                        qf_dev          equ     040h    ; device queue

                        ;*******************************************************
                        ;*
                        ;*      QPB - Queue Parameter Block
                        ;*
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*  00  |flgs|net |  qaddr  |  nmsgs  | buffptr |
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*  08  |                 name                  |
                        ;*      +----+----+----+----+----+----+----+----+
                        ;*
                        ;*      flgs    - unused
                        ;*      net     - unused (which machine to use)
                        ;*      qaddr   - Queue ID, address of QD
                        ;*      nmsgs   - number of messages to read/write
                        ;*      buffptr - address to read/write into/from
                        ;*      name    - name of queue (for open only)
                        ;*
                        ;*******************************************************

                        qpb_flgs        equ     byte ptr 0
                        qpb_net         equ     byte ptr qpb_flgs + byte
                        qpb_qaddr       equ     word ptr qpb_net + byte
                        qpb_nmsgs       equ     word ptr qpb_qaddr + word
                        qpb_buffptr     equ     word ptr qpb_nmsgs + word
                        qpb_name        equ     byte ptr qpb_buffptr + word
                        qpblen          equ     qpb_name + qnamsiz
```

```
                        eject 1 include modfunc.def
                        ;*******************************************************
                        ;*
                        ;*      CCP/M-86, MP/M-86 Inter-Module Function Definitions
                        ;*
                        ;*      Same calling conventions as User programs
                        ;*      except CX = function instead of CL
                        ;*             BX = 2nd parameter on entry
                        ;*      (CH=module, CL=function # in module)
                        ;*
                        ;*******************************************************

                                ; Module definitions
0000                    user            equ     0
0001                    sup             equ     1
0002                    rtm             equ     2
0003                    mem             equ     3
0004                    cio             equ     4
0005                    bdos            equ     5
0006                    xios            equ     6
0007                    net             equ     7

                                ; Bits that represent present modules
                                ; in module_map
0001                    supmod_bit      equ     001h
0002                    rtmmod_bit      equ     002h
0004                    memmod_bit      equ     004h
0008                    bdosmod_bit     equ     008h
0010                    ciomod_bit      equ     010h
0020                    xiosmod_bit     equ     020h
0040                    netmod_bit      equ     040h

                                ; Supervisor Functions
                        ;f_sysreset     equ     (user * 0100h) + 0
                        ;f_conin        equ     (user * 0100h) + 1
0002                    f_conout        equ     (user * 0100h) + 2
                        ;f_rconin       equ     (user * 0100h) + 3      ;supported internally
                        ;f_rconout      equ     (user * 0100h) + 4      ;under CCP/M
0005                    f_lstout        equ     (user * 0100h) + 5
                        ;f_rawconio     equ     (user * 0100h) + 6
                        ;f_getiobyte    equ     (user * 0100h) + 7
                        ;f_setiobyte    equ     (user * 0100h) + 8
0009                    f_conwrite      equ     (user * 0100h) + 9
000A                    f_conread       equ     (user * 0100h) + 10
                        ;f_constat      equ     (user * 0100h) + 11
                        ;f_bdosversion  equ     (user * 0100h) + 12
                        ;f_diskreset    equ     (user * 0100h) + 13
                        ;f_diskselect   equ     (user * 0100h) + 14
000F                    f_fopen         equ     (user * 0100h) + 15
0010                    f_fclose        equ     (user * 0100h) + 16
                        ;f_searchfirst  equ     (user * 0100h) + 17
                        ;f_searchnext   equ     (user * 0100h) + 18
```

```
345
346  =                    ;f_fdelete      equ     (user * 0100h) + 19
347  =  0014              f_freadseq      equ     (user * 0100h) + 20
348  =                    ;f_fwriteseq    equ     (user * 0100h) + 21
349  =                    ;f_fmake        equ     (user * 0100h) + 22
350  =                    ;f_frename      equ     (user * 0100h) + 23
351  =                    ;f_loginvector  equ     (user * 0100h) + 24
352  =  0019              f_getdefdisk    equ     (user * 0100h) + 25
353  =  001A              f_setdma        equ     (user * 0100h) + 26
354  =                    ;f_getallocvec  equ     (user * 0100h) + 27
355  =                    ;f_writeprotect equ     (user * 0100h) + 28
356  =                    ;f_getrovector  equ     (user * 0100h) + 29
357  =                    ;f_setfileattr  equ     (user * 0100h) + 30
358  =                    ;f_getdpb       equ     (user * 0100h) + 31
359  =  0020              f_usercode      equ     (user * 0100h) + 32
360  =  0021              f_freadrdm      equ     (user * 0100h) + 33
361  =                    ;f_fwriterdm    equ     (user * 0100h) + 34
362  =                    ;f_filesize     equ     (user * 0100h) + 35
363  =  0024              f_setrndrec     equ     (user * 0100h) + 36
364  =                    ;f_resetdrive   equ     (user * 0100h) + 37
365  =                    ;f_accessdrive  equ     (user * 0100h) + 38
366  =                    ;f_freedrive    equ     (user * 0100h) + 39
367  =                    ;f_writerndzero equ     (user * 0100h) + 40
368  =                    ;f_chain        equ     (user * 0100h) + 47
369  =  0032              f_callbios      equ     (user * 0100h) + 50
370  =  0033              f_setdmab       equ     (user * 0100h) + 51
371  =                    ;f_getdma       equ     (user * 0100h) + 52
372  =                    ;f_getmaxmem    equ     (user * 0100h) + 53
373  =                    ;f_getabsmaxmem equ     (user * 0100h) + 54
374  =                    ;f_allocmem     equ     (user * 0100h) + 55
375  =                    ;f_allocabsmem  equ     (user * 0100h) + 56
376  =  0039              f_freemem       equ     (user * 0100h) + 57
377  =  003A              f_freeall       equ     (user * 0100h) + 58
378  =                    ;f_userload     equ     (user * 0100h) + 59
379  =  006E              f_delim         equ     (user * 0100h) + 110
380  =  0080              f_malloc        equ     (user * 0100h) + 128
381  =  0082              f_memfree       equ     (user * 0100h) + 130
382  =                    ;f_polldev      equ     (user * 0100h) + 131
383  =  0084              f_flagwait      equ     (user * 0100h) + 132
384  =  0085              f_flagset       equ     (user * 0100h) + 133
385  =  0086              f_qmake         equ     (user * 0100h) + 134
386  =  0087              f_qopen         equ     (user * 0100h) + 135
387  =                    ;f_qdelete      equ     (user * 0100h) + 136
388  =  0089              f_qread         equ     (user * 0100h) + 137
389  =  008A              f_cqread        equ     (user * 0100h) + 138
390  =  008B              f_qwrite        equ     (user * 0100h) + 139
391  =  008C              f_cqwrite       equ     (user * 0100h) + 140
392  =                    ;f_delay        equ     (user * 0100h) + 141
393  =  008E              f_dispatch      equ     (user * 0100h) + 142
394  =  008F              f_terminate     equ     (user * 0100h) + 143
395  =  0090              f_createproc    equ     (user * 0100h) + 144
396  =  0091              f_setprior      equ     (user * 0100h) + 145
397  =  0092              f_conattach     equ     (user * 0100h) + 146
```

```
                        f_condetach     equ     (user * 0100h) + 147
                        ;f_setdefcon    equ     (user * 0100h) + 148
0095                    f_conassign     equ     (user * 0100h) + 149
                        ;f_clicmd       equ     (user * 0100h) + 150
                        ;f_callrsp      equ     (user * 0100h) + 151
0098                    f_parsefilename equ     (user * 0100h) + 152
                        ;f_getdefcon    equ     (user * 0100h) + 153
                        ;f_sdataddr     equ     (user * 0100h) + 154
                        ;f_timeofday    equ     (user * 0100h) + 155
                        ;f_pdaddress    equ     (user * 0100h) + 156
                        ;f_abortprocess equ     (user * 0100h) + 157
009A                    f_lstattach     equ     (user * 0100h) + 158
009F                    f_lstdetach     equ     (user * 0100h) + 159
                        ;f_setdeflst    equ     (user * 0100h) + 160
                        ;f_clstattch    equ     (user * 00h) + 161
00A2                    f_cconattch     equ     (user * 0100h) + 162
                        ;f_osvernum     equ     (user * 0100h) + 163
                        ;f_getdeflst    equ     (user * 0100h) + 164


                                ; Internal SUP functions
010A                    f_load          equ     (sup * 0100h) + 10
010E                    f_cload         equ     (sup * 0100h) + 14


                                ; Internal RTM functions
0203                    f_inflagset     equ     (rtm * 0100h) + 3
0212                    f_sleep         equ     (rtm * 0100h) + 18
0213                    f_wakeup        equ     (rtm * 0100h) + 19
0214                    f_findpdname    equ     (rtm * 0100h) + 20
0215                    f_sync          equ     (rtm * 0100h) + 21
0216                    f_unsync        equ     (rtm * 0100h) + 22
0217                    f_no_abort      equ     (rtm * 0100h) + 23
0218                    f_ok_abort      equ     (rtm * 0100h) + 24
0219                    f_no_abort_spec equ     (rtm * 0100h) + 25


                                ; Internal MEM functions
0308                    f_share         equ     (mem * 0100h) + 8
0309                    f_maualloc      equ     (mem * 0100h) + 9
030A                    f_maufree       equ     (mem * 0100h) + 10
030B                    f_mlalloc       equ     (mem * 0100h) + 11
030C                    f_mlfree        equ     (mem * 0100h) + 12


                                ; Internal CIO functions
040E                    f_conprint      equ     (cio * 0100h) + 14
0417                    f_cioterm       equ     (cio * 0100h) + 23
0418                    f_ciostat       equ     (cio * 0100h) + 24
0402                    f_rconin        equ     (cio * 0100h) + 2
0403                    f_rconout       equ     (cio * 0100h) + 3
```

```
451
452  =
453  =                              ; Internal BDOS functions
454  =  0520                f_bdosterm      equ     (bdos * 0100h) + 45
455
```

```
                        eject 1 include sysdat.dat
                        ;*******************************************************
                        ;*
                        ;*      System Data Area
                        ;*
                        ;*******************************************************
                        
                                CSEG
                                org     0ch
                        ;dev_ver
000C 06                         db      6       ;development system data version
                                                ;SYSDAT.CON has 16 byte code segment
                        
                                DSEG
                                org     0
                        
                                ;
                                ;This data is initialized by GENCCPM
                                ;
                        
                                ;Module Table - contains the FAR CALL addresses
                                ;       of each module for their initialization
                                ;       and entry routines.
                                ;
                                ;       +---+---+---+---+---+---+---+---+
                                ;       |     entry     |   initialize  |
                                ;       +---+---+---+---+---+---+---+---+
                                ;
                                ;               entry   init
                                ;               -----   ----
                        
0000                    module_table    equ     dword ptr (offset $)
0000                    supmod          equ     (offset $)
0000 030000000000               dw      3,0,    0,0     ;SUP
     0000
                        
0008                    rtmmod          equ     (offset $)
0008 030000000000               dw      3,0,    0,0     ;RTM
     0000
                        
0010                    memmod          equ     (offset $)
0010 030000000000               dw      3,0,    0,0     ;MEM
     0000
                        
0018                    ciomod          equ     (offset $)
0018 030000000000               dw      3,0,    0,0     ;CIO
     0000
                        
0020                    bdosmod         equ     (offset $)
0020 030000000000               dw      3,0,    0,0     ;BDOS
     0000
                        
```

```
                                xiosmod         equ     (offset $)
=0028 030C0000000C                              dw      0C03H,0,        0C00H,0 ;XIOS
      0000
=
=0030                           netmod          equ     (offset $)
=0030 030000000000                              dw      3,0,    0,0     ;NET
      0000
=
=0038                           dispatcher      equ     (offset $)
=0038 00000000                                  dw      0,0             ;far dispatcher (does IRET)
=
=003C                           rtm_pdisp       equ     (offset $)
=003C 00000000                                  dw      0,0             ;far dispatcher (does RETF)
=
=                                       ; location in memory of MP/M-86 or CCP/M-86
=
=0040 0810                      osseg           dw      1008h   ;1st parag. of MP/M-86 or CCP/M
=0042 0000                      rspseg          dw      0       ;segment of first RSP
=0044 0000                      endseg          dw      0       ;1st parag. outside of MP/M or CCP/M
=
=0046 3F                        module_map      db      03fh    ;bit map of modules that exist
=                                                               ; in this system.  low order bit
=                                                               ; corresponds to 1st module in
=                                                               ; module table.  If bit is on,then
=                                                               ; module exists.
=
=                                       ; some answers to GENCCPM questions
=
=0047 04                        ncns            db      4       ;# system console devices
=0048 01                        nlst            db      1       ;# system list devices
=0049 05                        nccb            db      5       ;# character control blocks
=004A 20                        nflags          db      20h     ;# flags
=004B 01                        srchdisk        db      1       ;system disk
=004C 0040                      mmp             dw      04000h  ;Max Memory per process
=004E 00                        nslaves         db      0       ;Number of network requestors
=004F 00                        dayfile         db      0       ;if 0ffh,display command info
=0050 01                        tempdisk        db      1       ;Temporary disk
=0051 3C                        tickspersec     db      60      ;Number of ticks per second
=
=                                       ; data lists created by GENCCPM
=
=0052 0000                      free_root       dw      0       ;locked unused list
=0054 0000                      ccb             dw      0       ;addr. Console Ctrl Blk Table
=0056 0000                      flags           dw      0       ;addr. Flag Table
=0058 2000                      mdul            dw      020h    ;Mem descr. Unused List
=005A 0000                      mfl             dw      0       ;Memory Free List
=005C 1400                      pul             dw      014h    ;Proc. descr. Unused List
=005E 2000                      qul             dw      020h    ;QCB Unused List
=                                                               ;MAU for queue buffer info
=0060 0000                      qmau            dw      0       ;link
=0062 0000                                      dw      0       ;start segment
=0064 0004                                      dw      400h    ;length
```

```
562
563  =0066 0000                          dw      0       ;plist
564  =
565  =                           ;
566  =                           ;This data is initialized at Assembly time
567  =                           ;
568  =
569  =0068 7704          rlr             dw      initpd  ;Ready List Root
570  =006A 0000          dlr             dw      0       ;Delay List Root
571  =006C 0000          drl             dw      0       ;Dispatcher Ready List
572  =006E 0000          plr             dw      0       ;Poll List Root
573  =0070 0000          scl             dw      0       ;Shared Code List
574  =0072 7704          thrdrt          dw      initpd  ;Process Thread Root
575  =0074 9401          qlr             dw      mxloadqd;Queue List Root
576  =0076 0000          mal             dw      0       ;Memory Alloc List
577  =
578  =                           ; Version Information
579  =
580  =0078 0000          version         dw      unknown ;addr. version str in SUP code segment
581  =                                                   ;set by GENCCPM if CCP/M
582  =                   if mpm
583  =                   bvernum         dw      01130h  ;MPM-86 w/BDOS v3.0
584  =                   osvernum        dw      01121h  ;MPM-86 V2.1
585  =                   endif
586  =
587  =                   if ccpm
588  =007A 3114          bvernum         dw      01431h  ;CCP/M w/BDOS 3.1
589  =007C 2014          osvernum        dw      01420h  ;CCP/M V2.0
590  =                   endif
591  =
592  =                           ; Time of Day Structure
593  =
594  =007E               tod             rw      0
595  =007E 7E06          tod_day         dw      067EH   ;day since 1/1/78 (09 Jul 82)
596  =0080 12            tod_hr          db      12h     ;hour of day
597  =0081 00            tod_min         db      00h     ;minute of hour
598  =0082 00            tod_sec         db      00h     ;second of minute
599  =
600  =                           ; info from XIOS
601  =
602  =0083 00            ncondev         db      0               ;# console devs in XIOS
603  =0084 00            nlstdev         db      0               ;# character devs in XIOS
604  =0085 00            nciodev         db      0               ;# character i/o devices
605  =                                                           ; supported by XIOS.
606  =0086 0000          lcb             dw      0               ; list control block address
607  =0088 0000          openvec         dw      0               ; open file vector
608  =008A 20            lock_max        db      20h             ; Max Locked Records/process
609  =008B 20            open_max        db      20h             ; Max Open Files/process
610  =008C 0000          owner8087       dw      0               ; no one owns it initially
611  =008E               rm              rw      1               ; RESERVED
612  =0090 FF            cmod            db      0ffh            ; BDOS Compatibility
613  =0091 00            ndp8087         db      false           ; true 8087 exits
614  =                                                           ; (Numeric Data Processor)
```

```
615
616  =0092 00000000          err_intercept   dw      0,0             ; BDOS does a callf here
617  =                                                               ; to print error msgs,
618  =                                                               ; if second word is <> 0
619  =0096 4106              slr             dw      offset mem_spb  ; Sync List Root
620  =0098 000000000000                      dw      0,0,0,0         ; RESERVED
621        0000
622  =
623  =
624  =                                ; SYSENT Table - MP/M-86, CCP/M-86 system function information
625  =                                ;       The supervisor calls the appropriate module
626  =                                ;       through this table.
627  =                                ;
628  =                                ;        Low Byte  High Byte
629  =                                ;       +----+----+--------+
630  =                                ;       |function |flgs|mod|
631  =                                ;       +----+----+--------+
632  =                                ;
633  =                                ;       flgs    - 001h - network intercept
634  =                                ;                 if on, the network module is called
635  =                                ;                 first, on return, either the function
636  =                                ;                 is called or it is considered complete
637  =                                ;                 depending on the return.
638  =                                ;       mod     - module number (0-15)
639  =                                ;       function- function to call within module
640  =
641  =                                ;       note: sup function 0 returns not the
642  =                                ;                 implemented error code to the caller,
643  =                                ;                 and sup function 1 returns the illegal
644  =                                ;                 function error code.
645  =
646  =                                ; standard CPM-2 functions
647  =
648  =                                ;       func,   module
649  =
650  =                                org     ((offset $) + 1) and 0fffeh
651  =
652  =00A0 0002              sysent  db      0,      rtm     ;  0-system reset
653  =00A2 0004                      db      0,      cio     ;  1-conin
654  =00A4 0104                      db      1,      cio     ;  2-conout
655  =00A6 0001                      db      0,      sup     ;  3-raw conin/aux in
656  =00A8 0001                      db      0,      sup     ;  4-raw conout/aux out
657  =00AA 0404                      db      4,      cio     ;  5-list out
658  =00AC 0504                      db      5,      cio     ;  6-raw conio
659  =00AE 0001                      db      0,      sup     ;  7-getiobyte
660  =00E0 0001                      db      0,      sup     ;  8-setiobyte
661  =00B2 0604                      db      6,      cio     ;  9-conwrite
662  =00B4 0704                      db      7,      cio     ; 10-conread
663  =00B6 0804                      db      8,      cio     ; 11-constat
664  =00B8 0201                      db      2,      sup     ; 12-get version
665  =00BA 0005                      db      0,      bdos    ; 13-diskreset
666  =00BC 0105                      db      1,      bdos    ; 14-diskselect
667  =00BE 0205                      db      2,      bdos    ; 15-file open
```

```
668
669  =00C0 0305                    db      3,      bdos    ; 16-file close
670  =00C2 0405                    db      4,      bdos    ; 17-search first
671  =00C4 0505                    db      5,      bdos    ; 18-search next
672  =00C6 0605                    db      6,      bdos    ; 19-file delete
673  =00C8 0705                    db      7,      bdos    ; 20-file read seq
674  =00CA 0805                    db      8,      bdos    ; 21-file write seq
675  =00CC 0905                    db      9,      bdos    ; 22-file make
676  =00CE 0A05                    db      10,     bdos    ; 23-file rename
677  =00D0 0C05                    db      11,     bdos    ; 24-login vector
678  =00D2 0C05                    db      12,     bdos    ; 25-get def disk
679  =00D4 0D05                    db      13,     bdos    ; 26-set dma
680  =00D6 0E05                    db      14,     bdos    ; 27-get alloc vector
681  =00D8 0F05                    db      15,     bdos    ; 28-write protect
682  =00DA 1005                    db      16,     bdos    ; 29-get r/O vector
683  =00DC 1105                    db      17,     bdos    ; 30-set file attr.
684  =00DE 1205                    db      18,     bdos    ; 31-get disk parm block
685  =00E0 1305                    db      19,     bdos    ; 32-user code
686  =00E2 1405                    db      20,     bdos    ; 33-file read random
687  =00E4 1505                    db      21,     bdos    ; 34-file write random
688  =00E6 1605                    db      22,     bdos    ; 35-file size
689  =00E8 1705                    db      23,     bdos    ; 36-set random record
690  =00EA 1805                    db      24,     bdos    ; 37-reset drive
691  =00EC 1905                    db      25,     bdos    ; 38-access drive
692  =00EE 1A05                    db      26,     bdos    ; 39-free drive
693  =00F0 1B05                    db      27,     bdos    ; 40-file write random w/zero fill
694  =
695  =                                             ;CPM-3 extensions
696  =
697  =00F2 0001                    db      0,      sup     ; 41-Test and Write (NOT IMPLEMENTED)
698  =                                                     ; Would be BDOS func # 28
699  =00F4 1C05                    db      28,     bdos    ; 42-Lock Record
700  =00F6 1D05                    db      29,     bdos    ; 43-Unlock Record
701  =00F8 1E05                    db      30,     bdos    ; 44-Set Multi-sector
702  =00FA 1F05                    db      31,     bdos    ; 45-Set Bdos Error Mode
703  =00FC 2005                    db      32,     bdos    ; 46-Get Disk Free Space
704  =00FE 0C01                    db      12,     sup     ; 47-Chain to Program
705  =                                                     ; In CP/M-86 BDOS func # 34
706  =0100 2105                    db      33,     bdos    ; 48-Flush Buffers
707  =0102 0101                    db      1,      sup     ; 49-
708  =
709  =                                             ;CPM-86 extensions
710  =
711  =0104 0301                    db      3,      sup     ; 50-call xios
712  =0106 2205                    db      34,     bdos    ; 51-set dma base
713  =0108 2305                    db      35,     bdos    ; 52-get dma
714  =010A 0003                    db      0,      mem     ; 53-get max mem
715  =010C 0103                    db      1,      mem     ; 54-get abs max mem
716  =010E 0203                    db      2,      mem     ; 55-alloc mem
717  =0110 0303                    db      3,      mem     ; 56-alloc abs mem
718  =0112 0403                    db      4,      mem     ; 57-free mem
719  =0114 0503                    db      5,      mem     ; 58-free all mem
720  =0116 0401                    db      4,      sup     ; 59-load
```

```
                =0118 0101                      db      1,      sup     ; 60-
                =011A 0101                      db      1,      sup     ; 61-
                =011C 0101                      db      1,      sup     ; 62-
                =011E 0101                      db      1,      sup     ; 63-
                =
                =                                       ;CP/NET functions
                =
                =0120 4007                      db      64,     net     ; 64-network login
                =0122 4107                      db      65,     net     ; 65-network logoff
                =0124 4207                      db      66,     net     ; 66-network send msg
                =0126 4307                      db      67,     net     ; 67-network rcv msg
                =0128 4407                      db      68,     net     ; 68-network status
                =012A 4507                      db      69,     net     ; 69-get network config addr
                =
                =                               ;CP/M-3 extensions
                =
                =012C 2405                      db      36,     bdos    ; 98-Reset Alloc Vector
                =012E 2505                      db      37,     bdos    ; 99-Truncate File
                =0130 2605                      db      38,     bdos    ;100-Set Dir Label
                =0132 2705                      db      39,     bdos    ;101-Return Dir Label
                =0134 2805                      db      40,     bdos    ;102-Read File XFCB
                =0136 2905                      db      41,     bdos    ;103-Write File XFCB
                =0138 2A05                      db      42,     bdos    ;104-Set Date and Time
                =013A 2B05                      db      43,     bdos    ;105-Get Date and Time
                =013C 2C05                      db      44,     bdos    ;106-Set Default Password
                =013E 0D01                      db      13,     sup     ;107-Return Serial Number
                =0140 0001                      db      0,      sup     ;108-(not implemented)
                =0142 1904                      db      25,     cio     ;109-Get/Set Console Mode
                =0144 1A04                      db      26,     cio     ;110-Get/Set Output Delimiter
                =0146 1B04                      db      27,     cio     ;111-Print Block
                =0148 1C04                      db      28,     cio     ;112-List Block
                =
                =                               ; MP/M functions
                =
                =014A 0603                      db      6,      mem     ;128-mem req
                =014C 0603                      db      6,      mem     ;129-(same function as 128)
                =014E 0703                      db      7,      mem     ;130-mem free
                =0150 0102                      db      1,      rtm     ;131-poll device
                =0152 0202                      db      2,      rtm     ;132-flag wait
                =0154 0302                      db      3,      rtm     ;133-flag set
                =0156 0402                      db      4,      rtm     ;134-queue make
                =0158 0502                      db      5,      rtm     ;135-queue open
                =015A 0602                      db      6,      rtm     ;136-queue delete
                =015C 0702                      db      7,      rtm     ;137-queue read
                =015E 0802                      db      8,      rtm     ;138-cond. queue read
                =0160 0902                      db      9,      rtm     ;139-queue write
                =0162 0A02                      db      10,     rtm     ;140-cond. queue write
                =0164 0B02                      db      11,     rtm     ;141-delay
                =0166 0C02                      db      12,     rtm     ;142-dispatch
                =0168 0D02                      db      13,     rtm     ;143-terminate
                =016A 0E02                      db      14,     rtm     ;144-create process
                =016C 0F02                      db      15,     rtm     ;145-set priority
```

```
                                        db      9,      cio     ;146-console attach
                                        db      10,     cio     ;147-console detach
                                        db      11,     cio     ;148-set def console
                                        db      12,     cio     ;149-console assign
                                        db      5,      sup     ;150-CLI
                                        db      6,      sup     ;151-call RPL
                                        db      7,      sup     ;152-parse filename
                                        db      13,     cio     ;153-get def console
                                        db      8,      sup     ;154-sysdat addr
                                        db      9,      sup     ;155-time of day
                                        db      16,     rtm     ;156-get.PD addr
                                        db      17,     rtm     ;157-abort process
                                ;
                                        ; MPM II extensions
                                ;
                                        db      15,     cio     ;158-attach list
                                        db      16,     cio     ;159-detach list
                                        db      17,     cio     ;160-set list dev
                                        db      18,     cio     ;161-Cond. Attach list
                                        db      19,     cio     ;162-Cond. Attach Console
                                        db      11,     sup     ;163-MP/M Version Number
                                        db      20,     cio     ;164-get list dev
                                ;
                                ;
                                        ; Initialized Queues
                                ;
                                        org     ((offset $) + 1) and 0fffeh
                                ;
                        mxloadqd        dw      mxdiskqd
                                        db      0,0
                                        dw      qf_keep+qf_mx
                                        db      'MXLoad  '
                                        dw      0,1,0,0,1,0,0
                        mxloadqpb       db      0,0
                                        dw      mxloadqd,1,0
                        ;               db      'MXLoad  '
                                ;
                                ;       ; Data Used by Load Program
                                ;
                                        org     ((offset $) + 1) and 0fffeh
                                ;
                        lod_uda         dw      0
                        lod_lstk        dw      0
                        lod_basep       dw      0
                        lod_nldt        dw      0
                        lod_pd          dw      0
                        lod_fcb         rs      36
                        lod_indma       dw      0
                        lod_nrels       db      0
```

```
827
828  =01F9 00                lod_chain       db      0
829  =01FA 00                lod_user        db      0
830  =01FB 00                lod_disk        db      0
831  =01FC 00                lod_fifty       db      0
832  =01FD 00                lod_8080        db      0
833  =01FE 00                lod_lbyte       db      0
834  =01FF 0000              lod_fixrec      dw      0
835  =0201 0000              lod_fixrecl     dw      0
836  =0203                   lod_dma         rb      dskrecl
837  =0273                   ldtab           rb      ldtabsiz
838  =
839  =031D                   cli_dma_ofst    rw      1
840  =031F                   cli_dma_seg     rw      1
841  =0321                   cli_pflag       rw      1
842  =0323                   cli_chain       rb      1
843  =0324                   cli_term        rb      1
844  =0325                   cli_dma         rb      dskrecl         ;dma buffer
845  =
846  =                                       ;copy of user's clicb
847  =03A5                   cli_net         rb      1               ;net
848  =03A6                   cli_ppd         rw      1               ;parent PD
849  =03A8                   cli_cmdtail     rb      129             ;command sent
850  =0429                                   rb      1
851  =
852  =042A                   cli_fcb         rb      fcblen+1        ;internal FCB
853  =
854  =044B 0000              cli_cuspqpb     db      0,0             ;QPB of command
855  =044D 00000000                          dw      0,0
856  =0451 A603                              dw      offset cli_ppd
857  =0453 242424242424                      db      '$$$$$$$$'
858        2424
859  =
860  =045B 0000              cli_acb         db      0,0             ;cns,match
861  =045D 0000                              dw      0               ;pd
862  =045F 242424242424                      db      '$$$$$$$$'      ;name
863        2424
864  =
865  =0467 A803              cli_pcb         dw      offset cli_cmdtail      ;parse
866  =0469 2A04                              dw      offset cli_fcb          ;ctl bk
867  =
868  =046B 0000              cli_pd          dw      0       ;pd of load prog
869  =046D 0000              cli_err         dw      0       ;error return
870  =046F 0000              cli_bpage       dw      0       ;base page
871  =0471 01                cli_lddsk       db      1       ;load disk
872  =
873  =                               ;parent information
874  =
875  =0472 00                cli_cns         db      0       ;pd.p_cns save
876  =0473 00                cli_user        db      0       ;pd.p_dsk save
877  =0474 00                cli_dsk         db      0       ;pd.p_user save
878  =0475 00                cli_err_mode    db      0       ;u_error_mode save
879  =0476 00                cli_dfil        db      0       ;dayfile flag
```

```
                                ;
                                ;System Initialization Variables
                                ;
initpd          dw      0               ;link
                dw      0               ;thread
                db      ps_run          ;stat
                db      1               ;prior
                dw      pf_sys+pf_kernal;flag
                db      'Init    '      ;name
                db      2020
                dw      unknown         ;uda segment
                db      0               ;disk
                db      0               ;user
                db      0,0             ;iosk,iuser
                dw      0               ;mem
                dw      0               ;dvract
                dw      0               ;wait
                db      0,0             ;org,net
                dw      0               ;parent
                db      0               ;cns
                db      0               ;abort
                db      0,0             ;cin,cout
                db      0               ;lst
                db      0,0,0           ;sf3,4,5
                rb      4               ;reserved
                dw      0,0             ;pret,scratch


                ;User Data Area of Init process
                ;paragraph aligned

                org     ((offset $)+0fh) AND 0fff0h
inituda         rb      ulen
init_tos        rw      0
                org     offset inituda + ud_insys
                db      1               ;keep the SUP from doing stack
                org     offset init_tos ;switches


                ; RTM data
                ; is word aligned from init uda

                dw      0cccch,0cccch,0cccch
                dw      0cccch,0cccch,0cccch
                dw      0cccch,0cccch,0cccch

                dw      0cccch,0cccch,0cccch
                dw      0cccch,0cccch,0cccch
                dw      0cccch,0cccch,0cccch
                dw      0cccch,0cccch,0cccch
```

```
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch

                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch

                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch
                                        dw      0cccch,0cccch,0cccch

                        dsptchtos       rw      0

                        indisp          db      false   ;?currently in dispatch?
                        intflag         db      0       ;if 0, interrupts not enabled -
                                                        ;not implemented
                        es_sav          dw      0       ;(staying word aligned)
                        bx_sav          dw      0
                        ax_sav          dw      0

                                ; MEM Data

                        beststart       dw      0
                        bestlen         dw      0
                        bestsi          dw      0
                        bestmau         dw      0
                        currmau         dw      0
                        currsi          dw      0
                        currmpb         dw      0,0,0,0,0
                                00000000

                        ;       SYNC Parameter Blocks
                        ;
                        ;       The MEM ENTRY: point uses the following for
                        ;       mutual exclusion and recursion.

                        mem_cnt         db      0       ;how many times a process has recursivly
                                                        ;called the memory manager

                        mem_spb         dw      q_spb   ;link   Mem Sync Parameter Block
                                        dw      0       ;owner
                                        dw      0       ;wait
                                        dw      0       ;next

                        ;       The queue sub-system in the RTM uses the following
                        ;       structure for mutual exclusion
```

```
 986
 987  =
 988  =0649 5105              q_spb           dw      cli_spb ;link   Queue Sync Parameter Block
 989  =064B 0000                              dw      0       ;owner
 990  =064D 0000                              dw      0       ;wait
 991  =064F 0000                              dw      0       ;next
 992  =
 993  =
 994  =                       ;       The CLI uses the CLI_SPB for mutual exclusion
 995  =
 996  =0651 5905              cli_spb         dw      thrd_spb;link   CLI Sync Parameter Block
 997  =0653 0000                              dw      0       ;owner
 998  =0655 0000                              dw      0       ;wait
 999  =0657 0000                              dw      0       ;next
1000  =
1001  =                       ;       When the thread is accessed, the THRD_SPB must be owned
1002  =                       ;       first.
1003  =
1004  =0659 A002              thrd_spb        dw      msg_spb ;link   Thread Sync Parameter Block
1005  =065B 0000                              dw      0       ;owner
1006  =065D 0000                              dw      0       ;wait
1007  =065F 0000                              dw      0       ;next
1008  =
1009  =                       ; Currently the order in which the SYNCs must be obtained if
1010  =                       ; more than one is needed is:
1011  =
1012  =                       ;       CLI
1013  =                       ;       QUEUE                   ;called by CLI for RSPs
1014  =                       ;       MEM                     ;called by make queue
1015  =                       ;       THREAD                  ;used from the MEM module
1016  =
1017  =                       ; The SYNCs must be released in reverse order
1018  =                       ; MSG_SPB is used by the BDOS to protect the BDOS error message
1019  =                       ; buffer.  MSG_SPB is in DATA.BDO
1020
```

```
1021
1022  =                     eject ! include data.bdo
1023  =
1024  =                     ;*********************************************************
1025  =                     ;*
1026  =                     ;*      bdos data area
1027  =                     ;*
1028  =                     ;*********************************************************
1029  =
1030  =  FFFF              CCPMOFF equ     true
1031  =
1032  =                     if 8CPM
1033  =
1034  =                     ;
1035  =                     ;       8086 variables that must reside in code segment
1036  =                     ;
1037  =                     cseg    $
1038  =                     ;
1039  =                     axsave          dw      0       ; register saves
1040  =                     ss_save         dw      0
1041  =                     sp_save         dw      0
1042  =                     stack_begin     dw      endstack
1043  =                     ;
1044  =                     ;       variables in data segment:
1045  =                     ;
1046  =                     dseg    cpmsegment
1047  =                             org     bdosoffset+bdoscodesize
1048  =
1049  =                     header  rs      128
1050  =                             rs      72
1051  =
1052  =                     pag0    dw      0               ;address of user's page zero
1053  =                     ip0     db      0               ;initial page value for ip register
1054  =                     ;
1055  =                     ;     memory control block
1056  =                     ;
1057  =                     umembase        dw      0       ;user'sbase for memory request
1058  =                     umemlg  dw      0               ;length of memory req
1059  =                     contf   db      0               ;flag indicates added memory is avail
1060  =                     ;
1061  =                     ;
1062  =                     hold_info       dw      0       ;save info
1063  =                     hold_spsave     dw      0       ;save user sp during program load
1064  =                     hold_sssave     dw      0       ;save user ss during program load
1065  =
1066  =                     mod8080         db      0
1067  =                     ;
1068  =                     ;       byte i/o variables:
1069  =                     ;
1070  =                     compcol db      0               ;true if computing column position
1071  =                     strtcol db      0               ;starting column position after read
1072  =                     column  db      0               ;column position
1073  =                     listcp  db      0               ;listing toggle
```

```
1074
1075  =                       kbchar  db      0               ;initial key char = 00
1076  =
1077  =                       endif
1078  =
1079  =                               dseg
1080  =                       if BNPM
1081  =                               org     0c00h   ;org for MP/M system
1082  =                       endif
1083  =
1084  =                       if BNPM and CCPMOFF
1085  =                               org     0800h   ;org for CCP/M system
1086  =                       endif
1087  =
1088  =                       if BNPM
1089  =0800 0000              rlog            dw      0       ;removeable logged-in disks
1090  =0802 0000              tlog            dw      0       ;removeable disk test login vector
1091  =0804 0000              ntlog           dw      0       ;new tlog vector
1092  =
1093  =                       endif
1094  =
1095  =0806 0000              rodsk           dw      0       ;read only disk vector
1096  =0808 0000              dlog            dw      0       ;logged-in disks
1097  =
1098  =080A 00                open_fnd        db      0               ;open file found in srch_olist
1099  =
1100  =                       ;the following variables are set to zero upon entry to file system
1101  =
1102  =080B 00                fcbdsk          db      0               ;disk named in fcb
1103  =080C 00                parlg           db      0               ;length of parameter block copied
1104  =080D 0000              aret            dw      0               ;adr value to return
1105  =  080D                 lret            equ     byte ptr aret   ;low(aret)
1106  =080F 00                resel           db      0               ;reselection flag
1107  =0810 00                rd_dir_flag     db      0               ;must read/locate directory record
1108  =0811 00                comp_fcb_cks    db      0               ;compute fcb checksum flag
1109  =0812 00                search_user0    db      0               ;search user 0 for file (open)
1110  =0813 00                make_xfcb       db      0               ;make & search xfcb flag
1111  =0814 00                find_xfcb       db      0               ;search find xfcb flag
1112  =0815 0000              xdcnt           dw      0               ;empty directory dcnt
1113  =0817 00                DIR_CNT         DB      0               ;DIRECT I/O COUNT
1114  =0818 00                MULT_NUM        DB      0               ;MULTI-SECTOR COUNT used in bdos
1115  =0819 00                olap_type       db      0               ;record lock overlap type
1116  =081A 00                ff_flag         db      0               ;0ffh xios error return flag
1117  =081B 00                err_type        db      0               ;type of error to print if not zero
1118  =081C                                   rb      4               ;reserved bytes
1119  =  0015                 zerolength      equ     (offset $)-(offset fcbdsk)
1120  =0820 01                mult_sec        db      1               ;multi sector count passed to xios
1121  =0821 00                RELOG           DB      0               ;AUTOMATIC RELOG SWITCH
1122  =
1123  =0822 00                BLK_OFF         DB      0               ;RECORD OFFSET WITHIN BLOCK
1124  =0823 0000              blk_num         dw      0               ;block number
1125  =                       ;
1126  =0825 2708              ua_lroot        dw      offset ua0      ;unallocated block list root
```

```
1127
1128  =                       ;
1129  =0827 20080000          ua0     dw      offset ua1,0
1130  =082B FF00                      db      0ffh,0
1131  =082D 30080000          ua1     dw      offset ua2,0
1132  =0831 FF00                      db      0ffh,0
1133  =0833 39080000          ua2     dw      offset ua3,0
1134  =0837 FF00                      db      0ffh,0
1135  =0839 3F080000          ua3     dw      offset ua4,0
1136  =083D FF00                      db      0ffh,0
1137  =083F 45080000          ua4     dw      offset ua5,0
1138  =0843 FF00                      db      0ffh,0
1139  =0845 00000000          ua5     dw      0,0
1140  =0849 FF00                      db      0ffh,0
1141  =
1142  =084B                   HASH            RB      4               ;HASH CODE WORK AREA
1143  =084F 00                HASHL           DB      0               ;HASH SEARCH LENGTH
1144  =
1145  =0850 0B                LOG_FXS         db      11              ;number of log_fxs entrys
1146  =                                       ;       fxi15,fxi17,fxi19,fxi22,fxi23
1147  =0851 020406090A                        db      02h  ,04h  ,06h  ,09h  ,0ah
1148  =                                       ;       fxi30,fxi35,fxi99,fxi100,fxi102,fxi103
1149  =0856 111625262829                      db      11h  ,16h  ,25h  ,26h   ,28h   ,29h
1150  =085C 00000000                          db      0,0,0,0         ;reserved
1151  =0860 07                rw_fxs          db      7               ;number of rw_fxs entrys
1152  =                                       ;       fxi20,fxi21,fxi33,fxi34,fxi40,fxi42,fxi43
1153  =0861 07081415181C                      db      07h  ,08h  ,14h  ,15h  ,1bh  ,1ch  ,1dh
1154          1D
1155  =0868 0000                              db      0,0             ;reserved
1156  =086A 02                sc_fxs          db      2               ;number of sc_fxs entrys
1157  =                                       ;       fxi16,fxi18
1158  =086B 0305                              db      03h  ,05h
1159  =086D 0000                              db      0,0             ;reserved
1160  =
1161  =086F 00                DEBLOCK_FX      DB      0               ;DEBLOCK FUNCTION #
1162  =0870 00                PHY_OFF         DB      0               ;RECORD OFFSET WITHIN PHYSICAL RECORD
1163  =0871 0000              CUR_BCBA        DW      0               ;CURRENT BCB OFFSET
1164  =0873 0000              ROOT_BCBA       DW      0               ;ROOT BCB OFFSET
1165  =0875 0000              EMPTY_BCBA      DW      0               ;EMPTY BCB OFFSET
1166  =
1167  =                       if BMPM
1168  =
1169  =0877 0000              P_LAST_BCBA     DW      0               ;PROCESS'S LAST BCB OFFSET
1170  =0879 00                P_BCB_CNT       DB      0               ;PROCESS BCB COUNT IN BCB LIST
1171  =
1172  =                       endif
1173  =
1174  =087A 00                fx_intrn        db      0               ;internal BDOS function number
1175  =
1176  =087B 0000              TRACK           DW      0       ;BCB RECORD'S TRACK
1177  =087D 0000              SECTOR          DW      0       ;BCB RECORD'S SECTOR
1178  =
1179  =                       ;       seldsk,usrcode are initialized as a pair
```

```
1180
1181  =087F 00              seldsk          db      0       ;selected disk num
1182  =0880 00              usrcode         db      0       ;curr user num
1183  =
1184  =0881 0000            info            dw      0       ;info adr
1185  =0883 0000            searcha         dw      0       ;search adr
1186  =
1187  =                     ;the following variable order is critical
1188  =
1189  =                     ;variables copied from uda for mp/m                     x
1190  =
1191  =                     ;variables included in fcb checksum for mp/m and cp/m     x
1192  =
1193  =                     ;variables used to access system lock list for mp/m         x
1194  =
1195  =0885 0000            dma_ofst        dw      0       ;dma offset             1
1196  =0887 0000            dma_seg         dw      0       ;dma segment            2
1197  =0889 00              func            db      0       ;bdos function #        3
1198  =088A 00              searchl         db      0       ;search len             4
1199  =
1200  =                     if BNPM
1201  =
1202  =088B 0000            searchaofst     dw      0       ;search adr ofst        5
1203  =088D 0000            searchabase     dw      0       ;search adr base        6
1204  =
1205  =                     endif
1206  =
1207  =088F 0000            dcnt            dw      0       ;directory counter      7
1208  =0891 0000            dblk            dw      0       ;directory block        8 ?? - not used - ??
1209  =0893 00              error_mode      db      0       ;bdos error mode        9
1210  =0894 00              mult_cnt        db      0       ;bdos multi-sector cnt 10
1211  =0895                 df_password     rb      8       ;process default pw    11
1212  =
1213  =                     if BNPM
1214  =
1215  =089D 00              pd_cnt          db      0       ;bdos process cnt      12 1
1216  =
1217  =                     endif
1218  =
1219  =089E 00              high_ext        db      0       ;fcb high extent bits     2
1220  =089F 00              xfcb_read_only  db      0       ;xfcb read only flag      3
1221  =08A0 FF              curdsk          db      0ffh    ;current disk             4 1
1222  =
1223  =                     if BNPM
1224  =
1225  =08A1 00              packed_dcnt     db      0       ;packed dblk+dcnt           2
1226  =08A2 00                              db      0
1227  =08A3 00                              db      0
1228  =08A4 0000            pdaddr          dw      0       ;process descriptor addr    3
1229  =
1230  =                     endif
1231  =
1232  =                             org ((offset $) + 1) and 0fffeh
```

```
1233
1234  =
1235  =                       ;       curtrka - alloca are set upon disk select
1236  =                       ;       (data must be adjacent)
1237  =
1238  =08A6 0000              cdrmaxa         dw      0       ;ptr to cur dir max val
1239  =08A8 0000              drvlbla         dw      0       ;drive label data byte addr
1240  =08AA 0000              buffa           dw      0       ;ptr to dir dma addr
1241  =08AC 0000              dpbaddr         dw      0       ;curr disk param block addr
1242  =08AE 0000              checka          dw      0       ;curr checksum vector addr
1243  =08B0 0000              alloca          dw      0       ;curr alloc vector addr
1244  =08B2 0000              DTR_BCBA        DW      0       ;DIRECTORY BUFFER CONTROL BLOCK ADDR
1245  =08B4 0000              DAT_BCBA        dw      0       ;DATA BUFFER CONTROL BLOCK ADDR
1246  =08B6 0000              HASH_SEG        dw      0       ;HASH TABLE SEGMENT
1247  =  000C                 addlist         equ     12      ;"$-buffa" = addr list size
1248  =
1249  =                       ;       sectpt - offset obtained from disk parm block at dpbaddr
1250  =                       ;       (data must be adjacent)
1251  =
1252  =08B8 0000              sectpt          dw      0       ;sectors per track
1253  =08BA 00                blkshf          db      0       ;block shift factor
1254  =08BB 00                blkmsk          db      0       ;block mask
1255  =08BC 00                extmsk          db      0       ;extent mask
1256  =08BD 0000              maxall          dw      0       ;max alloc num
1257  =08BF 0000              dirmax          dw      0       ;max dir num
1258  =08C1 0000              dirblk          dw      0       ;reserved alloc bits for dir
1259  =08C3 0000              chksiz          dw      0       ;size of checksum vector
1260  =08C5 0000              offsetv         dw      0       ;offset tracks at beginning
1261  =08C7 00                PHYSHF          DB      0       ;PHYSICAL RECORD SHIFT FACTOR
1262  =08C8 00                PHYMSK          DB      0       ;PHYSICAL RECORD MASK
1263  =08C9                   endlist         rs      0       ;end of list
1264  =  0011                 dpblist         equ     (offset endlist)-(offset sectpt)
1265  =                                               ;size
1266  =
1267  =                       ;       local variables
1268  =
1269  =08C9                   common_dma      rb      16      ;copy of user's dma 1st 16 bytes
1270  =08D9 00                make_flag       db      0       ;make function flag
1271  =08DA 00                actual_rc       db      0       ;directory ext record count
1272  =08DB 00                save_xfcb       db      0       ;search xfcb save flag
1273  =08DC 00                save_mod        db      0       ;open_reel module save field
1274  =08DD 00                pw_mode         db      0       ;password mode
1275  =08DE 00                attributes      db      0       ;fcb interface attributes hold byte
1276  =
1277  =                       if BMPM
1278  =
1279  =                       ;       number of lock list items required for lock operation
1280  =08DF 010002020101      required_table  db      1,0,2,2,1,1,2,2,1,1,2,2,1,1,2,2
1281          020201010202
1282          01010202
1283  =
1284  =08EF 00                chk_olist_flag  db      0       ;check | test olist flag
1285  =08F0 0000              lock_sp         dw      0       ;lock stack ptr
```

```
1286
1287  =08F2 00              lock_shell      db      0       ;lock shell flag
1288  =08F3 00              check_fcb_ret   db      0       ;check_fcb return switch
1289  =08F4 00              lock_unlock     db      0       ;lock | unlock function flag
1290  =08F5 00              incr_pdcnt      db      0       ;increment process_cnt flag ??
1291  =08F6 00              free_mode       db      0       ;free lock list entries flag ??
1292  =                                                     ;1=free entries for curdsk
1293  =                                                     ;0=free all entries
1294  =08F7 0000            cur_pos         dw      0       ;current position in lock list
1295  =08F9 0000            prv_pos         dw      0       ;previous position in lock list
1296  =
1297  =08FB 00              dont_close      db      0       ;inhibit actual close flag
1298  =08FC 00              open_cnt        db      0       ;process open file count
1299  =08FD 00              lock_cnt        db      0       ;process locked record count
1300  =08FE 0000            file_id         dw      0       ;address of file' lock list entry
1301  =0900 00              set_ro_flag     db      0       ;set drive r/o flag
1302  =0901 00              check_disk      db      0       ;disk reset open file check flag
1303  =0902 00              flushed         db      0       ;lock list open file flush flag
1304  =
1305  =                     ;free_root, lock_max, open_max initialized by sysgen
1306  =
1307  =0903 5200                            dw      offset free_root
1308  =0905 0000            open_root       dw      0       ;lock list open file list root
1309  =0907 0000            lock_root       dw      0       ;lock list locked record list root
1310  =
1311  =                     endif
1312  =
1313  =0909 0000            sdcnt           dw      0       ;saved dcnt of file's 1st fcb
1314  =090B 0000            sdcnt0          dw      0       ;saved dcnt (user 0 pass)
1315  =
1316  =                     if BCPM
1317  =
1318  =                     chain_flag      db      0       ;chain flag ??
1319  =                     tod             db      0ffh,0ffh,0ffh,0ffh ??
1320  =
1321  =                     endif
1322  =
1323  =
1324  =090D 00              rmf             db      0       ;read mode flag for open$reel
1325  =090E 00              wflag           db      0       ;xios/bios write flag
1326  =090F 00              dirloc          db      0       ;directory flag in rename, etc.
1327  =0910 00              linfo           db      0       ;low(info)
1328  =0911 00              dminx           db      0       ;local for diskwrite
1329  =0912 00              single          db      0       ;set true if single byte
1330  =                                                     ;alloc map
1331  =0913 00              rcount          db      0       ;record count in curr fcb
1332  =0914 00              extval          db      0       ;extent num and extmsk
1333  =0915 00              vrecord         db      0       ;curr virtual record
1334  =0916 00              ADRIVE          DB      0       ;CURRENT DISK - must preceed arecord
1335  =0917 0000            arecord         dw      0       ;curr actual record
1336  =0919 00                              db      0       ;curr actual record high byte
1337  =091A 0000            ARecCOR01       dw      0       ;curr actual block# * blkmsk
1338  =091C 0000            drec            dw      0       ;curr actual directory record
```

```
1339
1340  =091E 0000            CUR_dma_seg     DW      0
1341  =0920 0000            CUR_DMA         DW      0
1342  =
1343  =                     ;       local variables for directory access
1344  =
1345  =0922 00              dptr            db      0       ;directory pointer 0,1,2,3
1346  =  008F               ldcnt           equ     byte ptr dcnt   ;low(dcnt)
1347  =0923 00              user_zero_pass  db      0       ;search user zero flag
1348  =
1349  =                     ;       shell variables
1350  =
1351  =0924 0000            shell_si        dw      0       ;bdos command offset
1352  =0926 000000          shell_rr        db      0,0,0   ;r0,r1,r2 save area
1353  =
1354  =                     ;       special 8086 variables:
1355  =
1356  =0929 0000            uda_save        dw      0       ;user data area saved value
1357  =092B 0000            parametersegment dw     0       ;user parameter segment
1358  =092D 0000            returnseg       dw      0       ;user return segment
1359  =
1360  =                     ;       error messages
1361  =
1362  =092F 00              err_drv         db      0
1363  =0930 0000            err_pd_addr     dw      0
1364  =
1365  =0932 0D0A43502F4D    dskmsg          db      13,10,'CP/M Error On '
1366        204572726F72
1367        204F6E20
1368  =0942 203A2000        dskerr          db      ' : ',0
1369  =
1370  =0946 000060096909    xerr_list       dw      0,permsg,rodmsg,rofmsg,selmsg
1371        78098709
1372  =0950 B309C709DC09                    dw      xe5,xe6,xe7,xe8,xe9,xe10,xe11,xe3
1373        EB09FF09100A
1374        290A9509
1375  =
1376  =0960 4469736B2049    permsg          db      'Disk I/O',0
1377        2F4F00
1378  =0969 526561642F4F    rodmsg          db      'Read/Only Disk',0
1379        6E6C79204469
1380        736B00
1381  =0978 526561642F4F    rofmsg          db      'Read/Only File',0
1382        6E6C79204669
1383        6C6500
1384  =0987 496E76616C69    selmsg          db      'Invalid Drive',0
1385        642044726976
1386        6500
1387  =
1388  =099D 46696C65204F    xe3             db      'File Opened in Read/Only Mode',0
1389        70656E656420
1390        696E20526561
1391        642F4F6E6C79
```

```
1392
1393            20436F6e46500
1394    =
1395    =0983   46696C652043        xe5             db      'File Currently Open',0
1396            757272656E74
1397            6C79204F7065
1398            6E00
1399    =09C7   436C6F736520        xe6             db      'Close Checksum Error',0
1400            436865636B73
1401            756D20457272
1402            6F7200
1403    =09DC   50617373776F        xe7             db      'Password Error',0
1404            726420457272
1405            6F7200
1406    =09EB   46696C652041        xe8             db      'File Already Exists',0
1407            6C7265616479
1408            204578697374
1409            7300
1410    =09FF   496C6C656761        xe9             db      'Illegal ? in FCB',0
1411            6C203F20696E
1412            2046434200
1413    =0A10   4F70656E2046        xe10            db      'Open File Limit Exceeded',0
1414            696C65204C69
1415            6D6974204578
1416            63656564656
1417            00
1418    =0A29   4E6F20526F6F        xe11            db      'No Room in System Lock List',0
1419            6020696E2053
1420            797374656D20
1421            4C6F636B204C
1422            69737400
1423    =
1424    =0A45   0D0A00              crlf_str        db      13,10,0
1425    =0A48   0D0A42646F73        pr_fx           db      13,10,'Bdos Function = '
1426            2046756E6374
1427            696F6E203D20
1428    =0A5A   202020              pr_fx1          db      '   '
1429    =0A5D   2046696C6520        pr_fcb          db      ' File = '
1430            3D20
1431    =0A65                       pr_fcb1         rs      12
1432    =0A71   00                                  db      0
1433    =
1434    =0A72   0D0A44697361B       deniedmsg       db      13,10,'Disk reset denied, Drive '
1435            207265736574
1436            20646F6E6965
1437            642C20447269
1438            766520
1439    =0A8D   003A                deniedrv        db      0,':'
1440    =0A8F   20436F6E736F                        db      ' Console '
1441            6C6520
1442    =0A98   00                  deniedcns       db      0
1443    =0A99   2050726F6772                        db      ' Program '
1444            616D20
```

```
1445
1446  =0AA2 313233343536        deniedprc       db      '12345678',0
1447        373800
1448  =
1449  =                         if BMPM
1450  =
1451  =                         ;       bdos stack switch variables and stack
1452  =                         ;       used for all bdos disk functions
1453  =
1454  =                                 org     ((offset $) + 1) and 0fffeh
1455  =
1456  =                                         ; 69 word bdos stack
1457  =
1458  =0AAC CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1459  =0AB2 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1460  =0AB8 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1461  =0ABE CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1462  =0AC4 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1463  =0ACA CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1464  =0AD0 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1465  =0AD6 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1466  =0ADC CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1467  =0AE2 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1468  =0AE8 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1469  =0AEE CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1470  =0AF4 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1471  =0AFA CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1472  =0B00 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1473  =0B06 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1474  =0B0C CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1475  =0B12 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1476  =0B18 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1477  =0B1E CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1478  =0B24 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1479  =0B2A CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1480  =0B30 CCCCCCCCCCCC                       dw      0cccch,0cccch,0cccch
1481  =0B36                     bdosstack       rw      0
1482  =
1483  =0B36                     save_sp         rw      1
1484  =0B38                     ss_save         rw      1
1485  =0B3A                     sp_save         rw      1
1486  =
1487  =                         ;       local buffer area:
1488  =
1489  =0B3C                     info_fcb        rb      40      ;local user FCB
1490  =0B64                     save_fcb        rb      16      ;fcb save area for xfcb search
1491  =
1492  =0B74 0000                mxdiskqd        dw      0       ;link
1493  =0B76 0000                                db      0,0     ;net,org
1494  =0B78 0300                                dw      qf_keep+qf_mx   ;flags (MX queue)
1495  =0B7A 4D5864697368                        db      'MXdisk  '
1496        2020
1497  =0B82 00000100                            dw      0,1     ;msglen,nmsgs
```

```
1498
1499  =0886 00000000                    dw      0,0     ;nq,dq
1500  =088A 01000000                    dw      1,0     ;msgcnt,out
1501  =088E 0000                        dw      0       ;buffer ptr
1502  =
1503  =0890 00              mxdiskqpb   db      0       ;flgs
1504  =0891 00                          db      0       ;net
1505  =0892 7409                        dw      mxdiskqd        ;qaddr
1506  =0894 0100                        dw      1       ;nmsgs
1507  =0896 0000                        dw      0       ;buffer
1508  =0898 4D5864697368                db      'MXdisk  '
1509        2020
1510  =
1511  =                       ;       Error Message sync param block for mutual exclusion
1512  =
1513  =08A0 0000              msg_spb     dw      0       ;link   Error Message sync parameter block
1514  =08A2 0000                          dw      0       ;owner
1515  =08A4 0000                          dw      0       ;wait
1516  =08A6 0000                          dw      0       ;next
1517  =                       endif
1518  =
1519  =                       if BCPM
1520  =
1521  =                       ;
1522  =                       ;       special 8086 variables:
1523  =                       ;
1524  =                       ioloc           db      0       ;iobyte
1525  =                       user_parm_seg   dw      0       ;holds user parameter seg during load
1526  =                       nallocmem       db      0       ;no. of allocated memory segments
1527  =                       ncrmem          db      0       ;no. of available memory segments
1528  =                       crmem           dw      0,0     ;memory table (16 elements)
1529  =                                       dw      0,0,0,0,0,0,0,0
1530  =                                       dw      0,0,0,0,0,0,0,0
1531  =                                       dw      0,0,0,0,0,0,0,0
1532  =                                       dw      0,0,0,0,0,0,0,0
1533  =                       ;
1534  =                       mem_stack_length        equ     40
1535  =                       memstack        rs      mem_stack_length
1536  =                                       ;8 possible allocations
1537  =                       stbase  equ     word ptr 0
1538  =                       stlen   equ     word ptr 2
1539  =                       ccpflag equ     byte ptr 4
1540  =                       nccpalloc       db      0       ;number of current ccp allocations
1541  =                       mem_stk_ptr     dw      0       ;current memory stack location
1542  =
1543  =                       stackarea       rw      ssize   ;stack size
1544  =                       endstack        rb      0       ;top of stack
1545  =                       ;
1546  =                       endif
1547  =                                       org     0fffh
1548  =                       if CCPMOFF
1549  =                                       org     0bffh
1550  =                       endif
```

```
1551
1552  =0BFF 00                              db      0
1553
1554
1555
1556  END OF ASSEMBLY.  NUMBER OF ERRORS:   0.  USE FACTOR: 34%
```

```
ACTUALRC        080A V  1271#
ADDLIST         000C N  1247#
ADRIVE          0916 V  1334#
ALLOCA          0B80 V  1243#
ARECORD         0917 V  1335#
ARECORD1        091A V  1337#
ARET            0800 V  1104# 1105
ATTRIBUTES      080E V  1275#
AXSAV           0628 V   957#
BDOS            0000 N    68#   69  1032  1316  1519
BDOS            0005 N   311#  454   665   666   667   669   670   671   672   673
                         674   675   676   677   678   679   680   681   682   683
                         684   685   686   687   688   689   690   691   692   693
                         699   700   701   702   703   706   712   713   738   739
                         740   741   742   743   744   745   746
BDOSMOD         0020 N   505#
BDOSMODBIT      0008 N   320#
BDOSSTACK       0536 V  1481#
BESTLEN         062C V   962#
BESTMAU         0630 V   964#
BESTSI          062E V   963#
BESTSTART       062A V   961#
BLKMSK          088B V  1254#
BLKNUM          0823 V  1124#
BLKOFF          0822 V  1123#
BLKSHF          0889 V  1253#
BMPM            FFFF N    69# 1080  1084  1088  1167  1200  1213  1223  1277  1449
BUFFA           088A V  1240#
BVERNUM         007A V   583#  588#
BXSAV           0626 V   956#
CCB             0054 V   552#
CCPM            FFFF N    66#  587
CCPMOFF         FFFF N  1030# 1084  1548
CDRMAXA         0886 V  1238#
CHECKA          088E V  1242#
CHECKDISK       0901 V  1302#
CHECKRECORET    08F3 V  1288#
CHKOLISTFLAG    08EF V  1284#
CHKSIZ          08C3 V  1259#
CIO             0004 N   310#  445   446   447   448   449   653   654   657   658
                         661   662   663   749   750   751   752   775   776   777
                         778   782   790   791   792   793   794   796
CIOMOD          0018 N   501#
CIOMODBIT       0010 N   321#
CLIACB          0458 V   860#
CLIBPAGE        046F V   870#
CLICHAIN        0323 V   842#
CLICHDTAIL      0348 V   849#  865
CLICNS          0472 V   875#
CLICUSPQPB      044B V   854#
CLIDFIL         0476 V   879#
CLIDMA          0325 V   844#
CLIDMAOFST      031D V   839#
```

```
CLIDMASEG       031F V   840#
CLIDSK          0474 V   877#
CLIERR          046D V   869#
CLIERRMODE      0475 V   878#
CLIFCB          042A V   852#  866
CLILDDSK        0471 V   871#
CLINET          03A5 V   847#
CLIPCB          0467 V   865#
CLIPD           046B V   868#
CLIPFLAG        0321 V   841#
CLIPPD          03A6 V   848#  856
CLISPB          0651 V   988   996#
CLITERM         0324 V   843#
CLIUSER         0473 V   876#
CMDD            0090 V   612#
COMMAND1A       08C9 V  1269#
COMPCBCKS       0811 V  1108#
CRLFSTR         0A45 V  1424#
CS              SREG V
CURBCNA         0871 V  1163#
CURDMA          0920 V  1341#
CURDMASEG       091E V  1340#
CURDSK          0840 V  1221#
CURPOS          08F7 V  1294#
CURRMBU         0632 V   965#
CURRMPB         0636 V   967#
CURRSI          0634 V   966#
DATBCNA         0884 V  1245#
DAYFILE         004F V   545#
DBLK            0891 V  1208#
DCNT            088F V  1207# 1346
DEBLOCKFX       086F V  1161#
DEBUGINT        00E1 N    55#
DENIEDCNS       0A98 V  1442#
DENIEDDRV       0A8D V  1439#
DENIEDMSG       0A72 V  1434#
DENIEDPRC       0AA2 V  1446#
DFPASSWORD      0695 V  1211#
DIRBCNA         0882 V  1244#
DIRBLK          08C1 V  1258#
DIRCNT          0817 V  1113#
DIRLOC          090F V  1326#
DIRMAX          088F V  1257#
DISPATCHER      0038 N   518#
DLOG            0808 V  1096#
DLR             006A V   570#
DMADEST         0885 V  1195#
DMASEG          0887 V  1196#
DMAX            0911 V  1328#
DONTCLOSE       08F8 V  1297#
DPBADDR         08AC V  1241#
DPBLIST         0011 N  1264#
DPIR            0922 V  1345#
```

```
DREC            091C V  1338#
DRL             006C V   571#
DRVLBLA         0A8B V  1239#
DS              SREG V
DSKERR          0942 V  1368#
DSKMSG          0932 V  1365#
DSKRECL         0080 N    48#   836    844
DSPICHIOS       0622 V   950#
EMPTYPCBA       0875 V  1165#
ENDLIST         0AC9 V  1263#  1264
ENDSEG          0044 V   528#
ERRORV          092F V  1362#
ERRINTERCEPT    0092 V   616#
ERRORMODE       0893 V  1209#
ERRPDADDR       0930 V  1363#
ERRTYPE         0818 V  1117#
ES              SREG V
ESSAV           0624 V   955#
EXTMSK          08BC V  1255#
EXTVAL          0914 V  1332#
FALSE           0000 N    46#    65     68    513    952
FBOOSTERR       052D N   454#
FCALLMIOS       0032 N   369#
FCBDSK          0808 V  1102#  1119
FCBLEN          0020 N    49#   852
FCONATTCH       00A2 N   414#
FCIOSTAT        0418 N   447#
FCIOTERM        0417 N   446#
FCLOAD          010E N   421#
FCONASSIGN      0095 N   401#
FCONATTACH      0092 N   397#
FCONDETACH      0093 N   399#
FCONOUT         0002 N   328#
FCONPRINT       040E N   445#
FCONREAD        000A N   336#
FCONWRITE       0009 N   335#
FCREAD          008A N   389#
FCWRITE         008C N   391#
FCREATEPROC     0090 N   395#
FDELIM          006E N   379#
FDISPATCH       008E N   393#
FFCLOSE         0010 N   342#
FFFLAG          081A V  1116#
FFINDPDNAME     0214 N   428#
FFLAGSET        0085 N   384#
FFLAGWAIT       0084 N   383#
FFOPEN          000F N   341#
FFREADRDM       0021 N   360#
FFREADSEQ       0014 N   347#
FFREEALL        003A N   377#
FFREEMEM        0039 N   376#
FGETDEFDISK     0019 N   352#
FILEAD          08FE V  1300#
```

```
FINDXFCB        0614 V   1111#
FINFLAGSET      0203 N    425#
FLAGMIN         0003 N     62#
FLAGS           0056 V    553#
FLAGSET         0002 N     61#
FLAGTICK        0001 N     60#
FLOAD           010A N    420#
FLSTATTACH      009E N    410#
FLSTDETACH      009F N    411#
FLSTOUT         0005 N    331#
FLUSHED         0902 V   1303#
FMALLOC         0080 N    380#
FMAUALLOC       0309 N    438#
FMAUFREE        030A N    439#
FMEMFREE        0082 N    381#
FMLALLOC        030B N    440#
FMLFREE         030C N    441#
FNAMSIZ         0008 N     52#
FNDABORT        0217 N    431#
FNDABORTSPEC    0219 N    433#
FOKABORT        0218 N    432#
FPARSEFILENAME  0098 N    404#
FQMAKE          0086 N    385#
FQOPEN          0087 N    386#
FQREAD          0089 N    388#
FQWRITE         008B N    390#
FRCONIN         0402 N    448#
FRCONOUT        0403 N    449#
FREEMODE        08F6 V   1291#
FREEKOUT        0052 V    551# 1307
FSETDMA         001A N    353#
FSETDMAB        0033 N    370#
FSETPRIOR       0091 N    396#
FSETRNDREC      0024 N    363#
FSHARE          0308 N    437#
FSLEEP          0212 N    426#
FSYNC           0215 N    429#
FTERMINATE      008F N    394#
FTYPSIZ         0003 N     53#
FUNC            0889 V   1197#
FUNSYNC         0216 N    430#
FUSERCODE       0020 N    359#
FWAKEUP         0213 N    427#
FXINTRN         087A V   1174#
HASH            0846 V   1142#
HASHL           084F V   1143#
HASHSEG         0886 V   1246#
HIGHEXT         089E V   1219#
INCRPDCNT       08F5 V   1290#
INDISP          0622 V    952#
INFO            0881 V   1184#
INFOFCB         083C V   1489#
INITPD          0477 V    569   574   886#
```

```
INITIOS         0580 V   916#  919
INITODA         0480 V   915#  917
INITFLAG        0623 V   953#
LCB             0086 V   606#
LDCNT           089F V  1346#
LDIRB           0273 V   837#
LDIALSIZ        00AA N    63#  837
LINFO           0910 V  1327#
LOCKCNT         08FD V  1299#
LOCKMAX         008A V   608#
LOCKROOT        0907 V  1309#
LOCKSHELL       08F2 V  1287#
LOCKSP          08F0 V  1285#
LOCKUNLOCK      08F4 V  1289#
LODBORO         01ED V   832#
LODBASEP        01BC V   821#
LODCHAIN        01E9 V   828#
LODDISK         01EB V   830#
LODDMA          01F3 V   836#
LODFCB          01C2 V   824#
LODFIFTY        01EC V   831#
LODFIXREC       01EF V   834#
LODFIXREC1      01F1 V   835#
LODINDMA        01E6 V   825#
LODLBYTE        01EE V   833#
LODLSTK         018A V   820#
LODNLDT         018E V   822#
LODNRECS        01C8 V   826#
LODPD           01C0 V   823#
LODUDA          0188 V   819#
LODUSER         01EA V   829#
LOGFXS          0850 V  1145#
LRET            080D V  1105#
MAKEFLAG        0809 V  1270#
MAKEXFCB        0813 V  1110#
MAL             0076 V   576#
MAXALL          088D V  1256#
MDUL            0058 V   554#
MEM             0003 N   309#  437   438   439   440   441   714   715   716   717
                         718   719   756   757   758
MEMCNT          0640 V   976#
MEMMOD          0010 N   497#
MEMMODBIT       0004 N   319#
MEMSPB          0641 V   619   979#
MFL             005A V   555#
MMP             004C V   543#
MODULEMAP       0046 V   530#
MODULETABLE     0000 N   488#
MPM             0000 N    65#   66   582
MSGSPB          08A0 V  1004  1513#
MULTCNT         0894 V  1210#
MULTNUM         0818 V  1114#
MULTSEC         0820 V  1120#
```

```
MXDISKQD          0B74 V    803  1492#  1505
MXDISKQPB         0B90 V   1503#
MXLOADQD          0194 V    575    803#   812
MXLOADQPB         0190 V    811#
NCCP              0049 V    540#
NCLODEV           0085 V    604#
NCNS              0047 V    538#
NCONDEV           0083 V    602#
NDP8087           0091 V    613#
NEI               0007 N    513#   729    730    731    732    733    734
NEIMOD            0030 N    514#
NEIMODBIT         0040 N    323#
NFLAGS            004A V    541#
NLST              0048 V    539#
NLSTDEV           0084 V    603#
NSLAVES           004E V    544#
NTLOG             0B04 V   1091#
OFFSETV           08C5 V   1260#
OLAPTYPE          0819 V   1115#
OPENCNT           08FC V   1298#
OPENMD            080A V   1096#
OPENMAX           008B V    609#
OPENROOT          0905 V   1308#
OPENVEC           0088 V    607#
OSINT             00E0 N     54#    55
OSSEG             0040 V    526#
OSVERNUM          007C V    584#   589#
OWNER8087         008C V    610#
PABORT            0021 N    145#   146
PACKEDDCNT        08A1 V   1225#
PARAMETERSEGMENT  0928 V   1357#
PARLG             080C V   1103#
PBLOCNT           0879 V   1170#
PCALL             0001 N    191#
PCALLC            0008 N    194#
PCALLD            0080 N    196#
PCALLS            0002 N    192#
PCMOD             0018 N    140#   141
PCNROOT           0004 N    193#
PCNRSX            0300 N    197#
PCNS              0020 N    144#   145
PCONMODE          0022 N    146#   147
PDADDR            08A4 V   1228#
PDCNT             089D V   1215#
PDLEN             0030 N     58#
PDSK              0012 N    135#   136
PERMSG            0960 V   1370   1376#
PF8087            8000 N    186#
PFACTIVE          0100 N    179#
PFCHILDABORT      0800 N    182#
PFCTLC            0080 N    178#
PFCTLD            0400 N    181#
PFDSKLD           2000 N    184#
```

PFKEEP	0002	N	171#			
PFKERNAL	0004	N	172#	890		
PFLAG	0006	N	132#	133		
PFNOCTLS	1000	N	183#			
PFNOKBD	4000	N	185#			
PFPURE	0008	N	173#			
PFRAW	0040	N	176#			
PFRESOURCE	0020	N	175#			
PFSYS	0001	N	170#	890		
PFTABLE	0010	N	174#			
PFTEMPKEEP	0200	N	180#			
PHYMSK	08C8	V	1262#			
PHYOFF	0870	V	1162#			
PHYSOF	08C7	V	1261#			
PLASTBCBA	0877	V	1169#			
PLDSK	0014	N	137#	138		
PLINK	0000	N	128#	129		
PLR	006E	V	572#			
PLST	0024	N	147#	148		
PLUSER	0015	N	138#	139		
PMEM	0016	N	139#	140		
PNAME	0008	N	133#	134		
PNAMSIZ	0008	N	50#	51	52	134
PPARENT	001E	N	143#	144		
PPREL	002C	N	148#	149		
PPRIOR	0005	N	131#	132		
PRFCB	0A5D	V	1429#			
PRFCB1	0A65	V	1431#			
PRFX	0A48	V	1425#			
PRFX1	0A5A	V	1428#			
PRVPOS	08F9	V	1295#			
PSCIOWAIT	0009	N	165#			
PSCRATCH	002E	N	149#	150		
PSDELAY	0002	N	158#			
PSDQ	0006	N	162#			
PSFLAGWAIT	0008	N	164#			
PSNQ	0007	N	163#			
PSPOLL	0001	N	157#			
PSRUN	0000	N	156#	888		
PSSLEEP	0005	N	161#			
PSSWAP	0003	N	159#			
PSSYNC	000A	N	166#			
PSTAT	0004	N	130#	131		
PSTERM	0004	N	160#			
PTHREAD	0002	N	129#	130		
PTKCNT	0019	N	141#	142		
PUDA	0010	N	134#	135		
PUL	005C	V	556#			
PUSER	0013	N	136#	137		
PWAIT	001A	N	142#	143		
PWMODE	08DD	V	1274#			
PWSCRTCH	002E	N	150#			
QBUF	001A	N	251#	252		

```
QDLEN              001C N   2524#
QDQ                0012 N   247#  248
QFDEV              0040 N   263#
QFHIDE             0004 N   258#
QFKEEP             0002 N   257#  805  1494
QFLAGS             0004 N   243#  244
QFMX               0001 N   256#  805  1494
QFRPL              0020 N   262#
QFRSP              0008 N   260#
QFTABLE            0010 N   261#
QLINK              0000 N   240#  241
QLR                0074 V   575#
QMAU               0060 V   559#
QMSGCNT            0016 N   249#  250
QMSGLEN            000E N   245#  246
QMSGOUT            0018 N   250#  251
QNAME              0006 N   244#  245
QNAMSIZ            0008 N    51#  245   290
QNET               0002 N   241#  242
QNMSGS             0010 N   246#  247
QNQ                0014 N   248#  249
QORG               0003 N   242#  243
QPBBUFFPTR         0006 N   288#  289
QPBFLGS            0000 N   284#  285
QPBLEN             0010 N   290#
QPBNAME            0008 N   289#  290
QPBNET             0001 N   285#  286
QPBNMSGS           0004 N   287#  288
QPBQADDR           0002 N   286#  287
QSPB               0649 V   979   938#
QUL                005E V   557#
RCOUNT             0913 V  1331#
RDDIRFLAG          0810 V  1107#
RELOG              0821 V  1121#
REQUIREDTABLE      080F V  1280#
RESEL              080F V  1106#
RETURNSEG          092D V  1358#
RLOG               0800 V  1089#
RLR                0068 V   569#
RMF                090D V  1324#
RODMSG             0969 V  1370  1378#
RODSK              0806 V  1095#
RDFMSG             0978 V  1370  1381#
ROOTBCBA           0873 V  1164#
RSPSEG             0042 V   527#
RTM                0002 N   308#  425   426   427   428   429   430   431   432   433
                            652   759   760   761   762   763   764   765   766   767
                            768   769   770   771   772   773   785   786
RTMMOD             0008 N   493#
RTMMODBIT          0002 N   318#
RTMPDISP           003C N   521#
RWFXS              0860 V  1151#
SAVEFCB            0B64 V  1490#
```

```
SAVEMOD          030C V  1273#
SAVESP           0b36 V  1483#
SAVEXFCB         0806 V  1272#
SCFXS            086A V  1156#
SCL              0070 V   573#
SDCNT            0909 V  1313#
SDCNTO           090b V  1314#
SEARCHA          0883 V  1185#
SEARCHABASE      088D V  1203#
SEARCHADFST      088B V  1202#
SEARCHL          088A V  1198#
SEARCHUSERO      0312 V  1109#
SECTOR           087D V  1177#
SECTPT           0888 V  1252# 1264
SELDSK           087F V  1181#
SELMSG           0987 V  1370  1384#
SEIRQFLAG        0900 V  1301#
SHELLRR          0926 V  1352#
SHELLST          0924 V  1351#
SINGLE           0912 V  1329#
SLR              0096 V   619#
SPSAVE           083A V  1041# 1485#
SRCHDISK         004B V   542#
SS               SREG V
SSSAVE           0b38 V  1040# 1484#
SUP              0001 N   307#  420   421   655   656   659   660   664   697   704
                          707   711   720   722   723   724   725   747   748   779
                          780   781   783   784   795
SUPMOD           0000 N   489#
SUPMODBIT        0001 N   317#
SYSENT           00A0 V   652#
TEMPDISK         0050 V   546#
THRDRT           0072 V   574#
THRDSPb          0659 V   996  1004#
TICKSPERSEC      0051 V   547#
TLOG             0302 V  1090#
TOD              007E V   594# 1519#
TODDAY           007E V   595#
TODHR            0080 V   596#
TODLEN           0005 N    59#
TODMIN           0081 V   597#
TODSEC           0082 V   598#
TRACK            087B V  1176#
TRUE             FFFF N    45# 1030
U8087LEN         015E N    57#
UA0              0827 V  1126  1129#
UA1              082D V  1129  1131#
UA2              0833 V  1131  1133#
UA3              0839 V  1133  1135#
UA4              083F V  1135  1137#
UA5              0845 V  1137  1139#
UALRDDT          0825 V  1126#
UDASAVE          0929 V  1356#
```

```
UDIRSYS          0060 N   204#  917
ULEN             0100 N    56#   57   915
UNKNOWN          0060 N    47#  580   693
USER             0000 N   306#  328   331   335   336   341   342   347   352   353
                          359   360   363   369   370   376   377   379   380   381
                          383   384   385   386   388   389   390   391   393   394
                          395   396   397   399   401   404   410   411   414
USERZEROPASS     0923 V  1347#
USRCODE          0880 V  1182#
VERSION          0078 V   580#
VRECORD          0915 V  1333#
WFLAG            090E V  1325#
XDCNT            0815 V  1112#
XE10             0A10 V  1372  1413#
XE11             0A29 V  1372  1418#
XE3              0995 V  1372  1388#
XE5              0983 V  1372  1395#
XE6              09C7 V  1372  1399#
XE7              09DC V  1372  1403#
XE8              09EB V  1372  1406#
XE9              09FF V  1372  1410#
XERRLIST         0946 V  1370#
XFCBREADONLY     089F V  1220#
XIOS             0006 N   312#
XIOSMOD          0028 N   510#
XIOSMODBAT       0020 N   322#
ZEROLENGTH       0015 N  1119#
```