DUMP.ASM

Dump program, reads input file and displays hex

; Dump program, reads input file and displays hex data

;

0100 org 100h

0005 = bdos equ 0005h ;dos entry point

0001 = cons equ 1 ;read console

0002 = typef equ 2 ;type function

0009 = printf equ 9 ;buffer print entry

000B = brkf equ 11 ;break key function (true if char ready)

000F = openf equ 15 ;file open

0014 = readf equ 20 ;read function

;

005C = fcb equ 5ch ;file control block address

0080 = buff equ 80h ;input disk buffer address

;

; non graphic characters

000D = cr equ 0dh ;carriage return

000A = lf equ 0ah ;line feed

;

; file control block definitions

005C = fcbdn equ fcb+0 ;disk name

005D = fcbfn equ fcb+1 ;file name

0065 = fcbft equ fcb+9 ;disk file type (3 characters)

0068 = fcbrl equ fcb+12 ;file's current reel number

006B = fcbrc equ fcb+15 ;file's record count (0 to 128)

007C = fcbcr equ fcb+32 ;current (next) record number (0 to 127)

007D = fcbln equ fcb+33 ;fcb length

;

; set up stack

0100 210000 lxi h,0

0103 39 dad sp

; entry stack pointer in hl from the ccp

0104 221502 shld oldsp

; set sp to local stack area (restored at finis)

0107 315702 lxi sp,stktop

; read and print successive buffers

010A CDC101 call setup ;set up input file

010D FEFF cpi 255 ;255 if file not present

010F C21B01 jnz openok ;skip if open is ok

;

; file not there, give error message and return

0112 11F301 lxi d,opnmsg

0115 CD9C01 call err

0118 C35101 jmp finis ;to return

;

openok: ;open operation ok, set buffer index to end

011B 3E80 mvi a,80h

011D 321302 sta ibp ;set buffer pointer to 80h

; hl contains next address to print

0120 210000 lxi h,0 ;start with 0000

;

gloop:

0123 E5 push h ;save line position

0124 CDA201 call gnb

0127 E1 pop h ;recall line position

0128 DA5101 jc finis ;carry set by gnb if end file

012B 47 mov b,a

; print hex values

; check for line fold

012C 7D mov a,l

012D E60F ani 0fh ;check low 4 bits

012F C24401 jnz nonum

; print line number

0132 CD7201 call crlf

;

; check for break key

0135 CD5901 call break

; accum lsb = 1 if character ready

0138 0F rrc ;into carry

0139 DA5101 jc finis ;don't print any more

;

013C 7C mov a,h

013D CD8F01 call phex

0140 7D mov a,l

0141 CD8F01 call phex

nonum:

0144 23 inx h ;to next line number

0145 3E20 mvi a,' '

0147 CD6501 call pchar

014A 78 mov a,b

014B CD8F01 call phex

014E C32301 jmp gloop

;

finis:

; end of dump

0151 CD7201 call crlf

0154 2A1502 lhld oldsp

0157 F9 sphl

; stack pointer contains ccp's stack location

0158 C9 ret ;to the ccp

;

;

; subroutines

;

break: ;check break key (actually any key will do)

0159 E5D5C5 push h! push d! push b; environment saved

015C 0E0B mvi c,brkf

015E CD0500 call bdos

0161 C1D1E1 pop b! pop d! pop h; environment restored

0164 C9 ret

;

pchar: ;print a character

0165 E5D5C5 push h! push d! push b; saved

0168 0E02 mvi c,typef

016A 5F mov e,a

016B CD0500 call bdos

016E C1D1E1 pop b! pop d! pop h; restored

0171 C9 ret

;

crlf:

0172 3E0D mvi a,cr

0174 CD6501 call pchar

0177 3E0A mvi a,lf

0179 CD6501 call pchar

017C C9 ret

;

;

pnib: ;print nibble in reg a

017D E60F ani 0fh ;low 4 bits

017F FE0A cpi 10

0181 D28901 jnc p10

; less than or equal to 9

0184 C630 adi '0'

0186 C38B01 jmp prn

;

; greater or equal to 10

0189 C657 p10: adi 'a' - 10

018B CD6501 prn: call pchar

018E C9 ret

;

phex: ;print hex char in reg a

018F F5 push psw

0190 0F rrc

0191 0F rrc

0192 0F rrc

0193 0F rrc

0194 CD7D01 call pnib ;print nibble

0197 F1 pop psw

0198 CD7D01 call pnib

019B C9 ret

;

err: ;print error message

; d,e addresses message ending with "$"

019C 0E09 mvi c,printf ;print buffer function

019E CD0500 call bdos

01A1 C9 ret

;

;

gnb: ;get next byte

01A2 3A1302 lda ibp

01A5 FE80 cpi 80h

01A7 C2B301 jnz g0

; read another buffer

;

;

01AA CDCE01 call diskr

01AD B7 ora a ;zero value if read ok

01AE CAB301 jz g0 ;for another byte

; end of data, return with carry set for eof

01B1 37 stc

01B2 C9 ret

;

g0: ;read the byte at buff+reg a

01B3 5F mov e,a ;ls byte of buffer index

01B4 1600 mvi d,0 ;double precision index to de

01B6 3C inr a ;index=index+1

01B7 321302 sta ibp ;back to memory

; pointer is incremented

; save the current file address

01BA 218000 lxi h,buff

01BD 19 dad d

; absolute character address is in hl

01BE 7E mov a,m

; byte is in the accumulator

01BF B7 ora a ;reset carry bit

01C0 C9 ret

;

setup: ;set up file

; open the file for input

01C1 AF xra a ;zero to accum

01C2 327C00 sta fcbcr ;clear current record

;

01C5 115C00 lxi d,fcb

01C8 0E0F mvi c,openf

01CA CD0500 call bdos

; 255 in accum if open error

01CD C9 ret

;

diskr: ;read disk file record

01CE E5D5C5 push h! push d! push b

01D1 115C00 lxi d,fcb

01D4 0E14 mvi c,readf

01D6 CD0500 call bdos

01D9 C1D1E1 pop b! pop d! pop h

01DC C9 ret

;

; fixed message area

01DD 66696C6520signon: db 'file dump version 2.0$'

01F3 0D0A6E6F20opnmsg: db cr,lf,'no input file present on disk$'

; variable area

0213 ibp: ds 2 ;input buffer pointer

0215 oldsp: ds 2 ;entry sp value from ccp

;

; stack area

0217 ds 64 ;reserve 32 level stack

stktop:

;

0257 end