

Direitos reservados por



LIVROS TÉCNICOS E CIENTÍFICOS EDITORA S.A.

Av. Churchill, 94 — 4º andar — Rio de Janeiro — RJ — CEP: 20020

COMANDOS INTERNOS

d: Troca o drive corrente (d = A...P).

Exemplos:

```
A > B: <cr>
B >
B > A: <cr>
A >
```

DIR [[d:] [nomearq. tipo]] Mostra a relação dos arquivos no disquete.

Obs. (1) Não serão mostrados os arquivos com atributo SYS
(2) Mostra somente os arquivos do usuário corrente
(3) S pausa no display

Exemplos:

```
A > DIR <cr>
A: MOVCPM.COM:ASM COM:DDT COM:DUMP COM
A: ED COM:LOAD COM:PIP COM:STAT COM
A: TESTE BAS:ARQ2 DAT
A > DIR *.BAS <cr>
A: PROG1 BAS:PROG2 BAS:PROG3 BAS
```

ERA [d:] nomearq. tipo Elimina arquivo (s) do disquete.

Obs. (1) Não elimina arquivos com atributo R/O
(2) Elimina somente arquivos do usuário corrente

Exemplos:

```
A > ERA *.BAK <cr> Serão eliminados todos os arquivos com
tipo .BAK
A > ERA B:PGM3.BAS <cr> Será eliminado o arquivo PGM3.BAS do
drive B:
A > ERA *.* <cr> Serão eliminados todos os arquivos do dis-
quete. Obs. Será pedido confirmação.
```

REN [d:] novo.tipo = [d:] velho.tipo Altera o nome de um arquivo.

Obs. Não é permitido usar caracteres ambíguos

Exemplos:

```
A > REN ATUAL.DAT = ANTERIOR.DAT <cr>
A > REN B:NOVO.BAS = B:VELHO.BAS <cr>
```

SAVE n [d:] nomearq. tipo Grava o conteúdo da memória (a partir de 100H), no arquivo em disquete.

n = número (decimal) de páginas (blocos de 256 bytes) a serem gravadas

Exemplos:

```
A > SAVE 4 PROG.COM <cr> Grava a memória de 100H até 4FFF
no arquivo PROG.COM
A > SAVE 10 XYZ.COM <cr> Grava a memória de 100H até 0AFFH
no arquivo XYZ.COM
```

COMANDOS INTERNOS (cont.)

A > SAVE 40 B:TRAB.COM <cr> Grava a memória de 100H até 28FFF no arquivo TRAB.COM do drive B:

TYPE [d:] nomearq. tipo Mostra o conteúdo de um arquivo gravado no formato ASCII.

Obs. (1) S pausa no display
(2) Pressione qualquer tecla para cancelar
(3) Não é permitido usar caracteres ambíguos

Exemplo:

```
A > TYPE B:CNSLD.COB <cr> Mostra o conteúdo do arquivo
CNSLD.COB
```

USER n Altera o número do usuário corrente.

n = 0...15 (default = 0)

Exemplo:

```
A > USER 5 <cr>
```

PROGRAMAS TRANSIENTES

ASM nomearq. [P1 P2 P3] Converte um arquivo fonte em ASSEMBLER em um arquivo em formato hexa que será convertido em (.COM) pelo LOAD.

Obs. (1) Não deve ser colocado o tipo do arquivo
(2) Não é permitido o uso de macros
(3) O tipo do arquivo deve ser (.ASM)

P1 { A, B, ... P } Indica o drive que contém o arquivo fonte (.ASM)
P2 { A, B, ... P } Indica o drive que irá conter o arquivo (.HEX)
Z Não será criado o arquivo (.HEX)
A, B, ... P Indica o drive que irá conter o arquivo (.PRN)
P3 { Z Não será criado o arquivo listagem (.PRN)
X O arquivo (.PRN) será enviado ao vídeo da console

Exemplos:

```
A > ASM PROG1 <cr>
A > ASM PROG2.ABZ <cr>
└─ Não será criado o arquivo listagem (.PRN)
└─ O arquivo (.HEX) será criado no drive B:
└─ O arquivo (.ASM) está no drive A:
```

DDT [[d:] nomearq. tipo] Programa depurador usado para carregar, alterar e testar programas em linguagem de máquina (ver apêndice DDT).

nomearq. = arquivo a ser carregado na memória

Exemplos:

```
A > DDT A:XYZ.HEX <cr>
DDT VERSION 2.0
NEXT PC
0180 0100
└─ endereço inicial do programa
└─ endereço final do programa + 1
```

DUMP [d:] nomearq. tipo Mostra sob forma hexadecimal o conteúdo de um arquivo.

Obs. (1) S pausa no display
(2) Pressione qualquer tecla para cancelar o comando

Exemplo:

```
A > DUMP PROG5.COM <cr>
0000 ED 73 CF 02 31 DF 02 2A 01 00 11 18 00 19 22 8D
0010 01 11 03 00 19 22 90 01 19 22 93 01 19 22 96 01
0020 19 22 99 01 19 22 A0 01 0E 00 CD 8C 01 0E 00 CD
```

LOAD [d:] nomearq. Converte o código no formato hexa (.HEX) gerado pelo ASM em um programa objeto (.COM) pronto para ser executado como transiente a partir do endereço 100H.

Obs. O tipo do arquivo deve ser (.HEX)

Exemplo:

```
A > LOAD XYZ.HEX <cr>
FIRST ADDRESS 0100 Endereço inicial do programa
LAST ADDRESS 0234 Endereço final do programa
BYTES READ 0135 Número de bytes do programa
RECORDS WRITTEN 02 Número de registros ocupado pelo programa
```

PROGRAMAS TRANSIENTES (cont.)

MOVCPM [n] [*] Ajusta o CP/M para diferentes tamanhos de memória (entre 20 e 64 Kbytes).

n = tamanho da memória (número inteiro)

Exemplos:

```
A > MOVCPM <cr> O CP/M será ajustado usando a capacidade
total da memória e executado em seguida
Mesma função anterior, sendo 64 Kbytes o
tamanho da memória
A > MOVCPM 64 <cr>
A > MOVCPM * <cr> O CP/M será ajustado usando a capacidade
total da memória e ficará na memória pronto
para ser gravado com SYSGEN ou SAVE
Mesma função anterior, sendo 48 Kbytes o
tamanho da memória
A > MOVCPM 48 * <cr>
```

PIP [comandos] Copia informações de um lugar para outro (ver apêndice PIP).

Exemplo:

```
PIP LST: = PROG3.COB <cr>
```

STAT [comandos] Fornece informações sobre disquetes, arquivos, dispositivos de entrada e saída, e outras funções do sistema (ver apêndice STAT).

Exemplo:

```
STAT B:ARQTT.DAT $R/O
```

SUBMIT nomearq. [p1 p2 ...] Executa uma seqüência de comandos gravados num arquivo com tipo (.SUB).

nomearq. = arquivo (.SUB) que contém os comandos a serem executados
pn = valor que substituirá os parâmetros \$n dentro do arquivo de comandos

Exemplo:

```
A > SUBMIT COMANDOS B:COPIA.COM <cr>
```

Arquivo: A:COMANDOS.SUB

```
PIP $1: = A:$2
```

```
DIR $1:
```

Será executado:

```
A > PIP B: = A:COPIA.COM
```

```
A > DIR B:
```

SYSGEN [[d:] [nomearq. tipo]] Copia o CP/M para as trilhas 0 e 1 de um disquete.

Exemplo:

```
A > SYSGEN <cr>
SYSGEN VER x.x
SOURCE DRIVE NAME
(OR RETURN TO SKIP) A <cr> Pressione <cr> se o sistema já está na
memória
```

```
SOURCE ON A: THEN TYPE RETURN <cr>
```

```
FUNCTION COMPLETE
```

```
DESTINATION DRIVE NAME
```

```
(OR RETURN TO REBOOT) B <cr> Pressione <cr> para deixar o sistema na memória
```

```
DESTINATION ON B: THEN TYPE RETURN <cr>
```

```
FUNCTION COMPLETE
```

```
DESTINATION DRIVE NAME
```

```
(OR RETURN TO REBOOT) <cr>
```

```
A >
```

XSUB Permite fornecer parâmetros pela console a programas dentro do arquivo de comandos do SUBMIT.

Obs. (1) Deve ser o primeiro comando dentro do arquivo de comandos
(2) Deve ser colocado apenas uma vez no arquivo de comandos

Exemplo:

```
XSUB
ERA $1: *.*
Y
SYSGEN
PIP $1: = *.COM
STAT $1: *.*
```

- Ai** Entra com instruções em linguagem ASSEMBLER começando no endereço i.
Obs. Não é permitido entrar com labels ou pseudo-operandes.
Exemplo:
- A0100 <cr>
0100 MOV A, C <cr>
0101 . <cr> Use ponto para interromper o comando ou apenas <cr>
- D** Mostra o conteúdo da memória a partir do endereço corrente.
- Di** Mostra o conteúdo da memória a partir do endereço i.
Exemplo:
- D400 <cr>
400 410A 00 00 00 DE D0 00 01 B7 D8 AC 7E E9
42 43 A BC (serão mostradas 12 linhas)
- Di, f** Mostra o conteúdo da memória do endereço i ao f.
- Fi, f, v** Preenche a memória do endereço i a f com o valor hexa v.
Obs. v = 0 ... FFH
Exemplo:
- F400, 500, FF <cr>
- G** Inicia a execução a partir do endereço no Program Counter (PC).
- Gi** Inicia a execução do programa a partir do endereço i.
Exemplos:
- G100 <cr> Inicia a execução a partir do endereço 100H
- G0 <cr> Volta o controle para o CP/M
- Gi, p1** Inicia a execução no endereço i e interrompe em p1.
- Gi, p1, p2** Inicia a execução no endereço i e interrompe em p1 e p2.
Exemplo:
- G100, 127, 13D
- G, p1, p2** Inicia a execução a partir do endereço no PC e interrompe em p1 e p2.
- Ha, b** Executa aritmética hexadecimal.
Exemplo:
- H980, BC00 <cr>
C580, 4D80
Diferença (a - b) de 980 e BC00
Soma (a + b) de 980 e BC00
- Inomearq** Monta a FCB para o arquivo especificado.
Exemplo:
- ITESTPROG.HEX <cr>
- L** Mostra em linguagem ASSEMBLER o conteúdo da memória.
- Li** Mostra em linguagem ASSEMBLER o conteúdo da memória a partir do endereço i.

- Li, f** Exemplo:
- L100 <cr>
100 MVI A, 50
102 STA 01FE
etc.
- Li, f** Mostra em linguagem ASSEMBLER o conteúdo da memória desde i até f.
- Mi, f, d** Move bloco de memória desde i até f para d.
Exemplo:
- M100, 124, 200 <cr>
- R** Coloca na memória o arquivo especificado pelo comando l.
- Rd** Mesmo que R, sendo que o arquivo será carregado a partir de 100H + d.
- Si** Examina/altera o conteúdo da memória a partir do endereço i.
Exemplo:
- S110 <cr>
110 B5 C0 <cr> B5 → C0
111 00 <cr> O valor será mantido
112 2D C9 <cr> 2D → C9
113 7F . <cr> Um ponto interrompendo o comando
- Tn** Executa n instruções (default = 1) mostrando o conteúdo dos registradores e flags, antes da execução de cada instrução.
Exemplo:
- T3 <cr>
COZOM0E010 A = 00 B = 0000 D = 0000 H = 0000
S = 0100 P = 0100 MVI E, 20
COZOM0E010 A = 00 B = 0000 D = 0020 H = 0000
S = 0100 P = 0102 MVI C, 02
COZOM0E010 A = 00 B = 0002 D = 0020 H = 0000
S = 0100 P = 0104 PUSH D
- Un** Executa n instruções mostrando o conteúdo dos registradores e flags somente após a última instrução.
- X** Mostra o conteúdo dos registradores e flags da CPU.
Exemplo:
- X <cr>
COZOM0E010 A = 00 B = 0000 D = 0000 H = 0000
S = 0100 P = 0100 LXI B, 0FBC
- Xr** Mostra o conteúdo de um registrador ou flag da CPU e opcionalmente modifica o conteúdo.

r	REGISTRADORES	VALORES	r	FLAGS	VALORES
A	Registrador A	0 ... FF	M	Sinal negativo	0/1
B	Registradores BC	0 ... FFFF	C	Carry	0/1
D	Registradores DE	0 ... FFFF	Z	Zero	0/1
H	Registradores HL	0 ... FFFF	E	Paridade par	0/1
S	Stack pointer	0 ... FFFF	I	Interdigit carry	0/1
P	Program Counter	0 ... FFFF			

Exemplos:
- XA <cr>
A = 00
- XS <cr>
S = 0100

APÊNDICE PIP

SUMÁRIO DOS COMANDOS PIP

- d: novo, tipo = d: velho, tipo [ch] <cr>**
Copia o arquivo velho, tipo para o arquivo novo, tipo, usando a(s) chave(s).
Exemplo:
PIP ARQ1.DAT = XYZ.DAT <cr>
- d: novo, tipo = d: velho1, tipo, d: velho2, tipo [ch] <cr>**
Copia os arquivos velho1, tipo e velho2, tipo para o arquivo novo, tipo usando a(s) chave(s).
Exemplo:
PIP A: CADASTRO.DAT = A: JUNHO.DAT, A: JULHO.DAT <cr>
- d: nomearq, tipo = disp: [ch] <cr>**
Copia dados do dispositivo disp para o arquivo nomearq, tipo.
Exemplo:
PIP A: DADOS.DAT = CON: <cr>
- disp: = d: nomearq, tipo [ch] <cr>**
Copia dados do arquivo nomearq, tipo para o dispositivo disp.
Exemplo:
PIP LST: = VENDAS83.DAT <cr>
- disp1: = disp2: [ch] <cr>**
Copia dados do dispositivo disp2 para disp1.
Exemplo:
PIP PTP: = PTR: [EU] <cr>

DISPOSITIVOS DO PIP

DESTINO			FONTE		
LOGICOS	CON:	PUN:	LST:	LOGICOS	CON: RDR:
FISICOS	TTY:	PTP:	LPT:	FISICOS	TTY: PTR:
	CRT:	UP1:	UL1:	CRT:	UR1:
	UC1:	UP2:		UC1:	UR2:
ESPECIAIS	OUT:	PRN:		ESPECIAIS	NUL: EOF: INP:

DISPOSITIVOS ESPECIAIS DO PIP**NUL:**

Envia 40 caracteres nulos (00H) ao dispositivo
 Exemplo:
 PIP PUN: = PROG. HEX, NUL: <cr>

EOF:

Envia um fim-de-arquivo Z (1AH) ao dispositivo
 Exemplo:
 PIP PUN: = XYZ. TXT, EOF: <cr>

OUT:

Dispositivo de saída próprio do usuário
 Exemplo:
 PIP OUT: = XYZ. TXT <cr>

INP:

Dispositivo de entrada próprio do usuário
 Exemplo:
 PIP XYZ. TXT = INP: <cr>

PRN:

Forma especial de LST:
 - imprime 60 linhas por página
 - as linhas são numeradas
 - as tabulações são expandidas a cada 8 colunas
 Exemplo:
 PIP PRN: = EXEMPLO. TXT <cr>

CHAVES DO PIP

- B** Transferência de dados no modo bloqueado (S finaliza bloco)
 Exemplo:
 PIP XYZ. TXT = RDR: [B] <cr>
- Dn** Elimina todos os caracteres após a coluna n (n = 1 . . . 255)
 Exemplo:
 PIP PRN: = LONG. TXT [D52] <cr>
- E** Mostra na console o que está sendo processado.
- F** Remove os caracteres FF (form feed) durante uma operação de cópia.
- Gn** Permite acessar arquivos de outro usuário (n = 0 . . . 15)
- H** Transfere dados no formato hexa. Permite corrigir dados inválidos.
- I** Ignora registros nulos ":00" na transferência em formato hexa.
- L** Converte caracteres maiúsculos em minúsculos

CHAVES DO PIP (cont.)

- N** Adiciona número de linha, dois pontos e espaço para cada linha transferida (zeros significativos no número são suprimidos)
 Exemplo:
 PIP TTY: = EXEMPLO. BAS [N] <cr>
 1: 835983593459385938
 2: 234235245234543523
 3: 354365656333446677
- N2** Similar ao N (zeros significativos são incluídos)
 Exemplo:
 PIP TTY: = EXEMPLO. BAS [N2] <cr>
 000001 321321213213213244
 000002 099887776666453332
 000003 876757656545654123
- O** Transfere arquivos objeto (ignora fim-de-arquivo Z)
- Pn** Gera paginação a cada n linhas (n = 1 . . . 255) (default = 60)
 Exemplos:
 PIP LST: = ARQ1 [P58] <cr> Imprime 58 linhas por página
 PIP LST: = ARQ1 [P] <cr> Imprime saltando os picotes
 PIP LST: = ARQ3 <cr> Imprime sem considerar os picotes
- Qstr Z** Copia dados do arquivo até encontrar o str
- R** Permite copiar um arquivo com atributo SYS
- Sstr Z** Copia dados do arquivo a partir do str
- Tn** Gera tabulação de n colunas a cada ^1 (normalmente a cada 8)
- U** Converte caracteres minúsculos em maiúsculos
- V** Verifica se os dados foram copiados corretamente
- W** Permite gravar num arquivo protegido contra gravação (R/O)
 Exemplo:
 PIP A: XYZ. DAT = B:ARQ3. DAT [W] <cr>
- X** Permite copiar arquivos que não estejam no formato ASCII
- Z** Zera o bit de paridade durante a transferência de dados

APÊNDICE STAT**STAT**

Mostra o atributo e a área livre no (s) disquete (s).
 Exemplos:
 A > STAT <cr>
 A: R/W, SPACE: 2K

APÊNDICE STAT (cont.)

A > STAT <cr>
 A: R/W, SPACE: 2K
 B: R/O, SPACE: 120K

STAT d:

Mostra a área disponível no disquete do drive d:.

Exemplo:
 A > STAT B: <cr>
 BYTES REMAINING ON B: 170K

STAT [d:] nomearq, tipo

Mostra a área ocupada pelo (s) arquivo (s).

Exemplo:
 A > STAT * . COM <cr>
 RECS BYTES EX ACC d: nomearq, tipo
 4 2K 1 R/W A: DUMP. COM
 48 6K 1 R/W A: (ED. COM)
 56 8K 1 R/O A: PIP. COM
 BYTES REMAINING ON A: 252K

RECS Número de registros de 128 bytes usados pelo arquivo
 BYTES Tamanho do arquivo em Kbytes
 EX Número de extensões (blocos de 16Kbytes)
 ACC Atributos do arquivo

(a) Proteção contra gravação ou eliminação
 R/O Arquivo apenas para leitura
 Não é permitido gravar nem eliminar
 R/W Arquivo de leitura e gravação (atributo default)

(b) Mostrar ou não no diretório
 SYS Não aparece no diretório.
 Somente pode ser copiado com a opção R do PIP
 DIR Aparece no diretório (default)

d: Drive (A, B, . . . , P)
 nomearq Nome do arquivo
 tipo Tipo do arquivo

Obs. Os arquivos com atributo SYS aparecem entre parenteses

STAT [d:] nomearq, tipo \$\$

Mostra o tamanho do (s) arquivo (s).

Exemplo:
 A > STAT PIP. COM \$\$
 SIZE RECS BYTES EXT ACC
 55 55 12K 1 R/O A: PIP. COM

SIZE Número de registros de 128 bytes alocados para o arquivo. Nos arquivos sequenciais os campos SIZE e RECS são iguais

Copyright © 1985, 1986, Rubens Prates
 Proibida a reprodução, mesmo parcial, e por qualquer processo, sem autorização do autor e da editora. ISBN: 85-216-0493-9

STAT [d:] nomearq. tipo \$atr

Designa o atributo atr (R/O, R/W, DIR ou SYS) ao arquivo.

Exemplos:

```
A > STAT XYZ.DAT $R/O <cr>
XYZ.DAT set to R/O

A > STAT XYZ.DAT $$SYS <cr>
XYZ.DAT set to SYS
```

STAT DEV:

Mostra as designações atuais dos dispositivos físico-lógicos.

Exemplo:

```
A > STAT DEV: <cr>
CON: IS CRT:
RDR: IS UR1:
PUN: IS UP2:
LST: IS UL1:
      ^
      |
      |-----> dispositivos físicos
      |
      |-----> dispositivos lógicos
```

STAT VAL:

Mostra um sumário dos comandos STAT e as possíveis designações de dispositivos físicos e lógicos.

Exemplo:

```
A > STAT VAL: <cr>
Temp R/O Disk: d: = R/O
Set Indicator: d: filename. typ $R/O $R/W $$SYS $DIR
Disk Status : DSK: d: DSK:
User Status : USR:
lobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

STAT USR:

Mostra o número do usuário corrente e os números de usuários para os quais existem arquivos no disquete.

Exemplo:

```
A > STAT USR: <cr>
ACTIVE USER : 0
ACTIVE FILES: 0 1 3
```

STAT [d:] DSK:

Mostra como os dados estão gravados no disquete do drive d:.

Exemplo:

```
A > STAT DSK: <cr>
A: Drive Characteristics (1)
2720: 128 Byte Record Capacity (2)
340: Kilobyte Drive Capacity (3)
64: 32 Byte Directory Entries (3)
64: Checked Directory Entries (4)
```

- 256: Records/Extent (5)
- 16: Records/Block (6)
- 40: Sectors/Track (7)
- 2: Reserved Tracks (8)

- (1) número máximo de registros de 128 bytes
- (2) capacidade do disquete em Kbytes (não considera as trilhas reservadas)
- (3) número máximo de entradas no diretório
- (4) usualmente o mesmo que (3) para disquetes, Usualmente 0 para discos rígidos
- (5) número de registros por extensão
- (6) número de registros por bloco
- (7) número de setores por trilha
- (8) número de trilhas reservadas pelo CP/M

STAT [d:] R/O

Protege temporariamente contra gravação e eliminação o disquete do drive d:.

Exemplo:

```
A > STAT B:R/O <cr>
```

STAT logico1: = físico 1:

Atribui um dispositivo físico a um dispositivo lógico.

Exemplos:

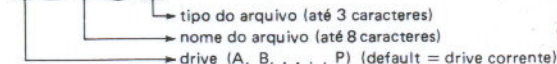
```
A > STAT CON: = CRT: <cr>
A > STAT CON: = CRT:, LST: = UL1: <cr>
```

CONTROLES DE EDIÇÃO DE LINHA E SAÍDA

TECLAS	DEC	HEX	SIGNIFICADO
^C	3	03	Reinicializa o CP/M. "Warm boot"
^E	5	05	Move o cursor para o início da próxima linha. Usada para digitar comandos longos
^H	8	08	Elimina o último caracter. O mesmo que Backspace
^I	9	09	Move o cursor para a próxima tabulação (1, 9, 17, 25, . . .)
^J	10	0A	O mesmo que Line Feed
^M	13	0D	O mesmo que Carriage Return <cr>
^P	16	10	Liga/desliga saída para a impressora a) omite paginação b) qualquer caracter de controle afeta a impressão c) a velocidade de saída para o vídeo e diminuída
^R	18	12	Repete a linha corrente
^S	19	13	Pausa no display. Pressione qualquer tecla para continuar
^U	21	15	Cancela a linha corrente, reinicia na próxima linha
^X	24	18	Volta para o início da linha corrente, apagando-a
^Z	26	1A	Finaliza entrada pela console
DEL	127	7F	Elimina o último caracter e o repete. O mesmo que Rubout

Obs. ^ significa tecla CONTROL

FORMATO: d: nnnnnnnn.ttt



REGRAS:

- . pode conter letras, números e caracteres especiais
- . não é permitido <> . ; = ? * : []
- . letras minúsculas são convertidas em maiúsculas

CARACTERES AMBÍGUOS:

- ? substitui qualquer caracter
- * substitui qualquer string

Exemplos:

```
M?? .COM      M10 .COM      M21 .COM      M32 .COM
?80 .COM      F80 .COM      L80 .COM      M80 .COM
PROG. *      PROG.ASM      PROG.HEX      PROG.PRN
* .ASM        JAN .ASM      FEV .ASM      MAR .ASM
MES* .DAT     MES01.DAT     MES02.DAT     MES03.DAT
* . *         PROG.BAS      ABC .COB      XYZ .ASM
```

TIPOS PREDEFINIDOS DE ARQUIVOS

TIPO	SIGNIFICADO	EXEMPLO
.ASC	Arquivo contendo texto em ASCII	CARTA.ASC
.ASM	Arquivo de fonte em ASSEMBLER	PROG1.ASM
.BAK	Arquivo de reserva (backup)	CADASTRO.BAK
.BAS	Arquivo de fonte em BASIC	MANUT.BAS
.CAL	Arquivo de dados do SUPERCALC	ORTN.CAL
.CMD	Arquivo de comandos	CONTR.CMD
.COB	Arquivo de fonte em COBOL	FPGT0.COB
.COM	Programa transiente executável diretamente	PIP.COM
.DAT	Arquivo de dados	MOVTO.DAT
.DBF	Arquivo de dados do DBASE	CADAST.DBF
.DOC	Arquivo de documento (arquivo de texto)	RELAT.DOC
.FMT	Arquivo de formatação de tela do DBASE	TELA1.FMT
.FOR	Arquivo de programa em FORTRAN	ESTAT.FOR
.FRM	Arquivo de definição de relatórios do DBASE	RELAT1.FRM
.HEX	Arquivo no formato hexadecimal	DRIVE.HEX
.LIB	Arquivo de biblioteca usado pelo macro assembler	SUB.LIB
.MAC	Arquivo de programa fonte em macro assembler	RECUP.MAC
.MEM	Arquivo de variáveis de memória do DBASE	VAR.MEM
.NDX	Arquivo de índices do DBASE	CALC.NDX
.OBJ	Arquivo de programa em código objeto	VERIF.OBJ
.PAS	Arquivo de fonte de PASCAL	EXEMPLO.PAS
.PLI	Arquivo de fonte em PL/I	MENU.PLI
.PRN	Arquivo de listagem de compiladores, assemblers	RECUP.PRN
.REL	Programa relocável em linguagem de máquina	ELIMINA.REL
.SUB	Arquivo de comandos do programa SUBMIT	FECHAM.SUB
.TEX	Arquivo de documento	MEMO.TEX
.TXT	Arquivo de documento (arquivo de texto)	MEMO.TXT
.\$\$\$	Arquivo temporário ou inútil	WORK.\$\$\$

DISPOSITIVOS

LÓGICOS

CON: Console.
RDR: Leitora
LST: Impressora
PUN: Perfuradora.

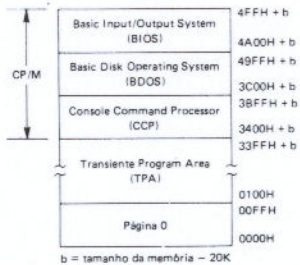
FÍSICOS

TTY: Teletipo (console lenta)
CRT: Vídeo (console rápida)
BAT: Console batch. (Entrada pela RDR: e saída pela LST.)
UC1: Console definida pelo usuário
PTR: Leitora de fita de papel ou leitora de alta velocidade
PTP: Perfuradora de fita de papel ou dispositivo de alta velocidade
UR1: Leitora definida pelo usuário
UR2: Leitora definida pelo usuário
UP1: Perfuradora definida pelo usuário
UP2: Perfuradora definida pelo usuário
LPT: Impressora. Normalmente uma impressora de alta velocidade
UL1: Impressora definida pelo usuário

ASSOCIAÇÕES PERMITIDAS ENTRE OS DISPOSITIVOS LÓGICOS E FÍSICOS

DISPOSITIVOS LÓGICOS		DISPOSITIVOS FÍSICOS			
Console	CON:	TTY:	CRT:	BAT:	UC1:
Leitora	RDR:	TTY:	PTR:	UR1:	UR2:
Perfuradora	PUN:	TTY:	PTP:	UP1:	UP2:
Impressora	LST:	TTY:	CRT:	LPT:	UL1:

MAPA PADRÃO DA MEMÓRIA



Obs. Os endereços de início do CCP, BDOS e BIOS variam conforme o tamanho da memória e configuração utilizada.

PÁGINA 0

ENDEREÇO	CONTEÚDO
0000 ... 0002	Desvio para a "Warm boot" no BIOS (RST 0)
0003	IOBYTE (indica associações correntes entre disp. log. e físicos)
0004	Usuário corrente (bits 7 ... 4) e drive corrente (bits 3 ... 0)
0005 ... 0007	Desvio para o ponto de entrada do BDOS
0008 ... 0028	RST 1 ... RST 5 (disponíveis para o usuário)
0030 ... 0037	RST 6 (reservado para uso futuro do CP/M)
0038 ... 003A	RST 7, usado para interrupção no DDT
003B ... 003F	Reservado para interrupção de máquina
0040 ... 004F	Reservado para uso temporário do BIOS
0050 ... 005B	Não usado
005C ... 007C	FCB default
007D ... 007F	Posição do registro randômico
0080 ... 00FF	Buffer (DMA)

IOBYTE

DISPOSITIVO LÓGICO	BITS	DISPOSITIVOS FÍSICOS				
		00	01	10	11	
Impressora	LST	7,6	TTY:	CRT:	LPT:	UL1:
Perfuradora	PUN	5,4	TTY:	PTP:	UP1:	UP2:
Leitora	RDR	3,2	TTY:	PTR:	UR1:	UR2:
Console	CON	1,0	TTY:	CRT:	BAT:	UC1:

FCB (File Control Block)

BYTE	CONTEÚDO
0	Drive (00H = corrente, 01H = A, ..., 10H = P)
1 ... 8	Nome do arquivo (ajustado a esquerda)
9 ... 11	Tipo do arquivo (ajustado a esquerda) bit 7 do byte 9 = 1 - arquivo com atributo R/O bit 7 do byte 10 = 1 - arquivo com atributo SYS
12	Número da extensão (00H = primeira, 01H = segunda, ...)
13 ... 14	Reservado para uso do CP/M
15	Número de registros nesta extensão
16 ... 31	Relação dos blocos alocados nesta extensão
32	Próximo registro a ser lido ou gravado nesta extensão
33 ... 34	Número do registro randômico (0 ... 65535)
35	Indicador de overflow de registro randômico

TABELA DE PARÂMETROS DE DISCO (DPH)

BYTES	TÍTULO	CONTEÚDO
1 ... 2	XLT	Endereço da tabela de translação de setores (fis x log.)
3 ... 8	nenhum	Reservado para uso do CP/M
9 ... 10	DIRBUF	Endereço do buffer de diretório (128 bytes)
11 ... 12	DPB	Endereço do bloco de parâmetros de disco
13 ... 14	CSV	Endereço da área usada para verificar se houve troca de disco
15 ... 16	ALV	Endereço do mapa de alocação de blocos do disco (cada bit corresponde a um bloco)

BLOCO DE PARÂMETROS DE DISCO (DPB)

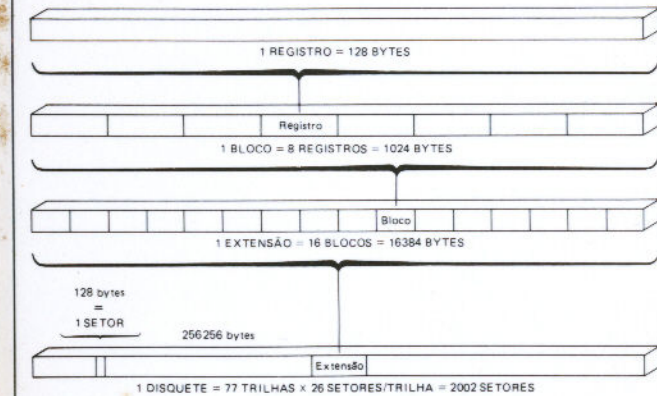
BYTES	TÍTULO	CONTEÚDO
1 ... 2	SPT	Número de setores (de 128 bytes) por trilha
3	BSH	Fator de deslocamento para alocação de blocos
4	BLM	Número de setores por bloco menos 1
5	EXM	Máscara para extensão
6 ... 7	DSM	Número máximo de blocos do disco menos 1
8 ... 9	DRM	Número máximo de entradas no diretório menos 1
10 ... 11	AL0/1	Usados para reservar blocos para o diretório (cada bit corresponde a um bloco)
12 ... 13	CKS	Tamanho da área usada para verificar se houve troca de disco
14 ... 15	OFF	Número de trilha iniciais reservadas para o CP/M (normalmente são 2 trilhas)

Tamanho do bloco	Setores /bloco	BSH	BLM
1024	8	3	07H
2048	16	4	0FH
4096	32	5	1FH
8192	64	6	3FH
16384	128	7	5FH

ENTRADA DO DIRETÓRIO

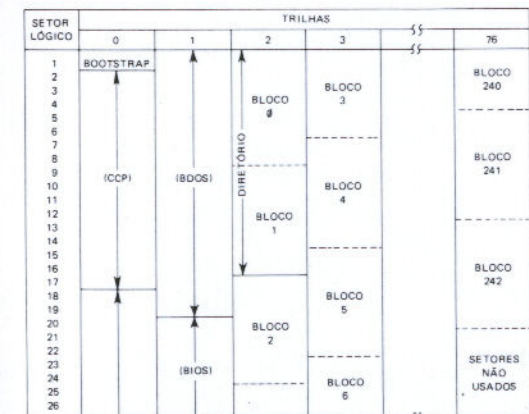
BYTE	CONTEÚDO
0	Número do usuário (Se 0E5H então esta entrada foi eliminada)
1 ... 8	Nome do arquivo (ajustado a esquerda)
9 ... 11	Tipo do arquivo (ajustado a esquerda)
12	Número da extensão (00H = primeira, 01H = segunda, ...)
13 ... 14	Reservado para uso do CP/M
15	Número de registros nesta extensão
16 ... 31	Relação dos blocos alocados nesta extensão

ORGANIZAÇÃO DE UM DISQUETE PADRÃO (IBM 3740)



Souls of CP/M de Mitchell Waite e Robert Lafore - HOWARD W. SAMS & Co., Inc.

LAY-OUT DE UM DISQUETE PADRÃO (IBM 3740)



The Programmer's CP/M Handbook de Andy Johnson-Laird - McGRAW-HILL BOOK COMPANY

