Manual do Usuário

DIGITUS

Manual do Usuário

DGP/M

GUIA DO USUÁRIO DO SISTEMA OPERACIONAL DGP/M, DIGITUS.

Este manual não poderá ser reproduzido, total ou parcialmente sem autorização por escrito da **DIGITUS IND. COM. SERV. DE ELETRÔNICA LTDA.**

O conteúdo deste manual tem caráter exclusivamente técnico/informativo e, a **DIGITUS** se reserva o direito de, sem qualquer aviso prévio, fazer as alterações que julgar necessárias.

Projeto e execução editorial.
DIGITUS IND. COM. SERV. DE ELETRÔNICA LTDA.
Rua Gávea, 150 - Jardim América
CEP 30430 - Belo Horizonte - MG.

SUMÁRIO

DGP/M

CONFIGURAÇÃO DO SISTEMA....3
CARACTERÍSTICAS DE ARMAZENAMENTO....4
ACIONAMENTO DO SISTEMA....4
PRESERVANDO SEU DGP/M....5
FORMATANDO DISQUETES....5
COPIANDO DISQUETES....6
ESPECIFICAÇÃO DE ARQUIVOS....9
IDENTIFICAÇÃO DE DRIVES....9

COMANDOS....10

COMANDOS EMBUTIDOS....10
- DIR....11
- ERA....12
- REN....13
- SAVE....13
- d:....14
- USER....14
- TYPE....15
- COMANDOS DIRETOS....15

DGP/M TRANSIENTES....16
- STAT....17
- PIP....22
- ED....27
- DUMP....32
- SUBMIT....33
- XSUB....33

UTILITÁRIOS....34
DDT....34
NOME....41
AUTO....42
CONFIG....43
CONFIGURAÇÃO FÍSICA DE DRIVES....46
CONFIGURAÇÃO LÓGICA DE DRIVES....47
CONFIGURAÇÃO DA INTERFACE SERIAL....49
CONFIGURAÇÃO DA IMPRESSORA....51
CONFIGURAÇÃO DO TECLADO....53

APÊNDICE 1

BIOS....55

APÊNDICE 2

BDOS....61

APÊNDICE 3

CCP....62

APÊNDICE 4

FORTRAN....63

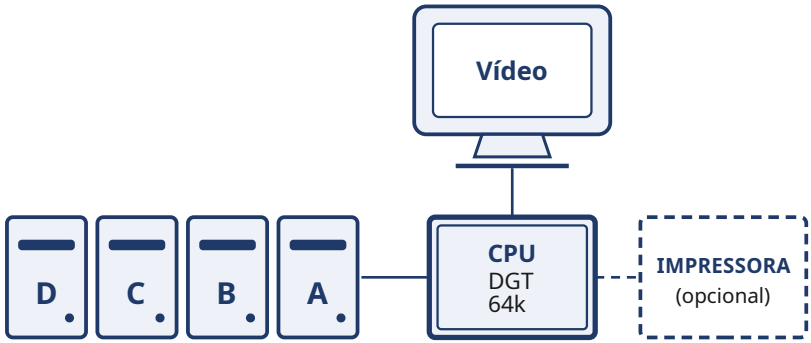
DGP/M

DGP/M é um sistema operacional de discos compatível com o CP/M versão 2.2 desenvolvido para ser implementado na linha TRS-80 de microcomputadores **DIGITUS**.

No microcomputador **DGT-1000** você terá a opção de operar com o tradicional **DIGDOS** ou com o **DGP/M** quando lhe for conveniente. Para implementar o **DGP/M** no seu **DGT-1000**, basta adquirir a placa para tal finalidade e conectá-la no **DGT-1000**. Este Manual, mais o disco com o sistema operacional acompanham a placa.

O CP/M ou **DGP/M** é um sistema operacional de fácil manipulação, de grande versatilidade e que possibilita a implementação de uma variedade de linguagens de alto nível, tais como: FORTRAN, COBOL, PL1 ou PASCAL, dentre outras. Assim, seu **DIGITUS** poderá ser comparado a um sistema de maior porte e ser compatível com outros computadores que utilizam o mesmo sistema.

A configuração mínima e ideal para que o **DGP/M** possa ser utilizado com todos os seus recursos é de 2 drives.



Até 4 drives de 5 1/4"

CARACTERÍSTICAS DE ARMAZENAMENTO

- Até 4 disc-drives de 5 1/4";
- 40 trilhas em densidade dupla;
- 10 setores por trilha;
- 512 bytes por setor;
- As duas primeiras trilhas são reservadas ao sistema;
- Total por unidade de disco 200K bytes;

ACIONAMENTO DO SISTEMA

- Insira o disquete de utilitários no drive A.
- Ligue o computador.
  O sistema DGP/M será carregado na memória e o controle será passado ao DGP/M.
- Várias operações iniciais serão executadas automaticamente.
- Após estas operações iniciais, aparecerá no vídeo um `A>-`, e o DGP/M estará à espera de um comando digitado pelo operador.
- Providencie agora, uma ou mais cópias de seu DGP/M, a fim de evitar perdas acidentais do sistema.

PRESERVANDO SEU DGP/M

Primeiro, você deverá formatar um número de disquetes necessário para receber as cópias. Será utilizado o programa **FORMATE.COM** existente no disquete de utilitários do DGP/M.

FORMATANDO DISQUETES:

É necessário que se formate os disquetes em branco antes de gravar qualquer dado neles. O utilitário **FORMATE** permite esta formatação.

Sintaxe:
FORMATE (drive)
A especificação do drive é obrigatória, e o disco será formatado de acordo com o formato do drive lógico.

EXEMPLO:
Digite: `A>FORMATE B:` **RETURN**

O sistema enviará uma mensagem para ser colocado no drive B, o disquete a ser formatado. Tecle **RETURN**, quando o disquete estiver posicionado. O disquete do drive B será formatado e verificado. Ao final da verificação, o sistema lhe perguntará se há necessidade de formatar outros disquetes. Responda com "S" (SIM) ou "N" (NÃO).

No caso de resposta afirmativa, todo o procedimento de formatação será repetido, caso contrário, o utilitário enviará uma mensagem para montar o disquete SISTEMA no drive e apertar a tecla **RETURN**.
Após ter formatado disquetes suficientes para copiar seu sistema DGP/M, inicie o processo de cópia.

NOTA:
A – O utilitário formata apenas discos no formato **DIGITUS**, e a formatação é feita de acordo com o formato do drive lógico.
B – Pode-se interromper a formatação a qualquer momento, pressionando a tecla **BREAK**.

COPIANDO DISQUETES:

O sistema operacional **DGP/M 2.2** incorpora o utilitário **COPIE**, que permite fazer cópias de discos e a possibilidade de serem feitas cópias utilizando-se apenas um drive.

Sintaxe:

```
(- COPIE ----------------------------------------)RETURN
         (-TUDO----) (-DE-) (-DRIVE-) (-PARA-) (-DRIVE-)
         (-DADOS---)
         (-SISTEMA-)
         (-SIS-----)
```

Todos os parâmetros são opcionais, e quando omitidos, são assumidos defaults (veja parametrização).

O utilitário permite a cópia entre drives distintos ou num drive só.

Quando são utilizados 2 drives, o utilitário, antes de iniciar a cópia, pede a confirmação de que os disquetes **FONTE** e **DESTINO** estão montados nos respectivos drives, permitindo assim, que disquetes que não contenham o utilitário, possam ser duplicados.

Quando um único drive é utilizado, o utilitário pede a troca entre os disquetes **FONTE** e **DESTINO** nos momentos adequados.

OBS:
A cópia integral de discos somente poderá ser feita entre discos com a mesma formatação, enquanto que a cópia de sistemas poderá ser feita entre discos com formatação diferente (número de trilhas), desde que sejam formatos DIGITUS.

EXEMPLOS:

A) COPIE TUDO DE A PARA B

Este comando copia todo o conteúdo do disquete do drive A para o disquete no drive B. São equivalentes a este comando (a menos que os defaults tenham sido alterados):

```
COPIE DE A PARA B
COPIE A B
COPIE TUDO A B
COPIE
```

B) COPIE TUDO DE A PARA A

Este comando copia todo o conteúdo de um disquete para outro no drive A. Nos momentos adequados é pedida a troca dos disquetes. São equivalentes a este comando:

```
COPIE DE A PARA A
COPIE A A
COPIE A
COPIE TUDO A
```

C) COPIE DADOS DE B PARA C

Este comando copia toda a área de dados (trilhas de 2 a 39) do disquete no drive B para o disquete no drive C.

D) COPIE SIS B A

Este comando copia a área de sistema (trilhas 0 e 1) do disquete no drive B para o disquete no drive A.

Ao final de cada cópia será perguntado, se deseja-se outra cópia. Responda adequadamente.

Se a cópia for de sistema, apenas outro disco DESTINO é gravado. Senão, toda a cópia é refeita.

Na cópia de sistema, o sistema destino é configurado de modo que o drive físico 0 tenha as mesmas especificações do drive físico DESTINO e o drive lógico A, as mesmas do drive lógico DESTINO.

PARAMETRIZAÇÃO DO UTILITÁRIO COPIE

O utilitário pode ser parametrizado para uma dada configuração através da alteração de endereços pré-determinados (via DDT, por exemplo). Abaixo, temos a relação destes endereços com seus conteúdos originais (em hexadecimal e em ASCII) e os seus significados:

Endereço	Hexa	ASCII	Significado
0103	41	A	Qual é por default o drive **FONTE**.
0104	42	B	Default para o drive **DESTINO**.
0105	44	D	Qual o último drive.
0106	00		Qual a primeira trilha a ser copiada (0 ou 2).
0107	27		Qual a última trilha a ser copiada (1 ou 27 hexa).
0108	03		Qual o número de tentativas a serem feitas antes de aceitar a operação de I/O no disquete, como errada.
0109	FF		Indica se durante a gravação será utilizado um entrelaçamento de 2 (valor FF) ou não (valor 00). Teste os dois valores, e mantenha aquele que obtiver melhor desempenho quanto à velocidade.

EXEMPLO

Deve-se usar as posições abaixo para especificar o default quanto ao tipo de cópia, da seguinte maneira:

	0106	0107
TUDO	00	27
DADOS	02	27
SISTEMA	00	01

Após fazer as cópias desejadas, guarde os disquetes originais e passe a utilizar as cópias.

ESPECIFICAÇÃO DE ARQUIVOS

Os arquivos são especificados por um conjunto de 8 caracteres alfanuméricos seguidos de "." (ponto) e o máximo de 3 caracteres de extensão (tipo do arquivo). Uma boa idéia é especificar cada arquivo com um nome e extensão que facilmente identifique o arquivo. Algumas extensões comumente usadas pelo sistema são listadas abaixo:

.ASC – Arquivo texto em ASCII.
.ASM – Programa fonte em Assembler.
.BAK – Arquivo backup.
.BAS – Programa fonte em BASIC.
.COB – Programa fonte em COBOL.
.COM – Programa diretamente executável.
.DAT – Arquivos de dados.
.FOR – Programa fonte em FORTRAN.
.INT – Programa BASIC intermediário (CBASIC).
.LIB – Arquivo biblioteca.
.MAC – Programa fonte MACRO-80.
.REL – Programa em códigos relocáveis.
.SUB – Arquivo de comandos SUBMIT.
.$$$ – Arquivo temporário ou impropriamente gravado.

IDENTIFICAÇÃO DE DRIVES

No sistema **DGP/M**, os drives são identificados por letras "A, B, C, ou D" seguidas de : (dois pontos). Toda vez que o sistema está em nível de comando **DGP/M**, é mostrado no vídeo a identificação do drive corrente, isto é, quando "A>" é apresentado no vídeo, quer dizer que o drive corrente é o drive A. Para alterá-lo, simplesmente digite a letra correspondente ao drive seguidos de **:** e **RETURN**, e será mostrado **D>**, onde D é a identificação do drive. Para retornar proceda da mesma maneira.

COMANDOS

Existem dois tipos de comandos, que são:

1 – COMANDOS EMBUTIDOS.
2 – COMANDOS TRANSIENTES.

Os comandos embutidos, como o próprio nome diz, já estão incorporados ao sistema e são executados sem consulta prévia ao disquete, enquanto que os comandos transientes são os programas que são carregados na memória do computador e executados.

COMANDOS EMBUTIDOS

São sete os comandos embutidos:

1 – **DIR**
2 – **ERA**
3 – **REN**
4 – **SAVE**
5 – **d:**
6 – **USER**
7 – **TYPE**

Ainda vários outros comandos diretos são prontamente interpretados como:

CTRL e C,
CTRL e E,
CTRL e P,
CTRL e R,
CTRL e S.

COMANDO DIR

O comando DIR fará com que o diretório do drive especificado seja mostrado no vídeo. Todos os arquivos poderão ser mostrados, selecionados por grupo ou somente um arquivo específico.

Formato geral:
DIR D: arquivo.tip RETURN

EXEMPLOS:

```
A>DIR A: RETURN
```

Mostrará todos os arquivos existentes no disquete montado no drive A. Neste caso, a identificação do drive poderá ser omitida, pois corresponde ao drive corrente.

```
A>DIR D:arquivo.tip RETURN
```

Serão mostrados todos os arquivos do drive D:, obedecendo à especificação de "arquivo.tip". Dois caracteres, "*" e "?" poderão ser usados para agrupar os arquivos. Alguns exemplos são mostrados adiante.

```
A>DIR B:*.FOR RETURN
B: VICON    FOR : VSBE     FOR : FLEX     FOR : FLAMB    FOR
B: ABCDE    FOR
A>-
```

Note que todos os arquivos com extensão **.FOR** foram mostrados.

Esta rotina lerá as informações

```
A>DIR B:AB???.* RETURN
B: ABCDE    COM : ABCDE    REL : ABXXX    BAS : ABCYY    FOR
B: ABCPM    REL : AB000    FOR
A>-
```

Neste, os caracteres "?" substituíram quaisquer caracteres nas respectivas posições onde se encontram, e "*" substituíram a extensão.

```
A>DIR C: *.CO? RETURN
C: COBTEST  COB : CTRL     COM : CTRL     COB : ABCD     COB
C: COBTEST  COM
A>-
```

Neste, o nome foi qualquer um, e a extensão teve os dois primeiros caracteres "CO".

Mensagens de erro:

NO FILE – Não foi encontrado qualquer arquivo com a especificação desejada.

BDOS ERR ON D: – O sistema não pode selecionar o drive D:. Pode ser que o disquete não esteja propriamente formatado ou que tenha sido danificado ou ainda, que a porta do drive esteja aberta. Pode ainda vir a mensagem de REGISTRO NÃO ENCONTRADO ou PORTA ABERTA.

COMANDO ERA

Este comando serve para apagar um arquivo ou grupos de arquivos especificados no comando.

Formato geral:
ERA D:arquivo.tip RETURN

Todos os arquivos especificados por arquivo.tip, existentes no drive D: serão apagados. Também, neste comando ERA poderão ser usados os caracteres "*" e "?" para agrupar os arquivos.

COMANDO REN

Com o comando REN, você poderá alterar os nomes dos arquivos existentes. Agora, os caracteres "*" e "?" não poderão ser utilizados.

Formato geral:
REN D:novonome.tip=D:velhonome.tip RETURN

EXEMPLO:

```
A>DIR RETURN
A: VICON    FOR : PORPLAN  FOR : VSBE     BAS
A>REN PORT1.FOR=PORPLAN.FOR RETURN
A>DIR RETURN
A: VICON    FOR : PORT1    FOR : VSBE     BAS
A>-
```

Neste exemplo, o arquivo PORPLAN.FOR passou a ser especificado por PORT1.FOR.

NOTA:
Caso o arquivo especificado não exista, a mensagem de NO FILE será apresentada, se o novo nome do arquivo já existir, será apresentada a mensagem FILE EXISTS.

COMANDO SAVE

Este comando permite gravar o conteúdo de uma parte da memória em um arquivo de disco.

Formato geral:
SAVE ## D:arquivo.tip RETURN
Onde, ## representa o número de blocos de 256 bytes de memória que serão armazenados no disquete, expresso em decimal, a partir do endereço 0100 hexa.

EXEMPLO:

```
A>SAVE 16 B:PROEXP.COM RETURN
A>-
```

Será criado um arquivo PROEXP.COM no drive B com extensão de 16*256 bytes a partir de 0100 hexa, isto é, de 0100 até 1100 hexa.

COMANDO d:

Troca do drive corrente. Em sistemas de dois ou mais drives, a troca do drive corrente pode ser executada teclando-se a letra correspondente ao drive desejado seguido de **:** (dois pontos) e **RETURN**.

EXEMPLOS:

```
A>B: RETURN
B>C: RETURN
C>A: RETURN
A>-
```

Ao se selecionar qualquer arquivo que não esteja no drive corrente, o drive deverá ser identificado, caso contrário, o sistema informará que o arquivo não existe.

COMANDO USER

Este comando permite que seja trocada a área de usuário do sistema atual. Estas áreas são imaginárias, pois, todo arquivo tem um número de área associado. Podem existir até 16 áreas de usuários, numeradas de 0 a 15 inclusive. Quando o sistema é iniciado o número da área utilizada é 0 (zero).

Formato geral do comando:
USER ## RETURN

EXEMPLO:

Peça o diretório do drive A, digitando:

```
DIR A:RETURN
```

Agora, altere a área de usuário para 2, digite:

```
A>USER 2 RETURN       ; alterar área de usuário.
A>DIR RETURN          ; diretório área.
NO FILE                   ; nenhum arquivo área 2.
A>USER 0 RETURN       ; retornar área 0.
A>-
```

COMANDO TYPE

O comando TYPE serve para listar no vídeo o conteúdo de um arquivo codificado em ASCII.
Usualmente este comando é utilizado para listar o conteúdo de arquivos dos tipos .BAS, .ASM, .BAK, .DAT, .DOC, .TXT ou qualquer outro arquivo que contenha algum texto em ASCII.

Há somente um formato do comando:
TYPE D:arquivo.tip RETURN

COMANDOS DIRETOS

CTRL e C
Control C é usado para ressetar o sistema i.e., usado para interromper um programa transiente retornando ao nível de comando do **DGP/M**, ou quando é introduzido algum disquete diferente em um ou mais drives.

CTRL e E
Control E é usado para continuar na próxima linha um comando longo demais, e o cursor será posicionado no início da próxima linha. Após o término, ao se teclar **RETURN**, todo comando será executado.

CTRL e P
Control P é usado para "ligar ou desligar" a impressora, enviando uma cópia de todas as mensagens enviadas ao vídeo também para a impressora.

CTRL e R
Control R é usado para repetir a linha corrente e poder fazer quaisquer alterações sem perder a linha digitada.

CTRL e S
Control S é usado para fixar o vídeo (pausa do vídeo).

DGP/M TRANSIENTES

Agora, serão mostrados alguns programas complementares dos comandos embutidos vistos anteriormente.
Estes programas irão facilitar as cópias de arquivos, fornecer informações dos periféricos, editar programas, examinar o conteúdo dos arquivos e executar comandos automaticamente.

COMANDO STAT

STAT - STATISTICS

Funções do comando STAT:

- Examinar o espaço livre do disquete.
- Examinar o espaço ocupado por um grupo de arquivos.
- Examinar o espaço ocupado por um arquivo.
- Atribuir R/O temporário a um drive.
- Atribuir R/O ou R/W a um grupo de arquivos.
- Atribuir R/O ou R/W a um arquivo.
- Atribuir DIR ou SYS a um grupo de arquivos.
- Atribuir DIR ou SYS a um arquivo.

Atribuição de arquivo:
Em seu **DGP/M** existem 2 pares de atribuições de arquivos: R/O ou R/W e DIR ou SYS.

R/O
Arquivo somente de leitura. O arquivo não poderá ser atualizado ou apagado usando comando ERA.

R/W
Arquivos com esta atribuição poderão ser lidos ou atualizados sem proteção, a não ser que o disquete esteja com a tarja de proteção. Esta é a atribuição normal de um arquivo.

SYS
Arquivos com atribuição SYS não aparecerão no diretório. Arquivos SYS poderão ser usados normalmente se existir em qualquer das áreas USER.

DIR
Atribuição normal dos arquivos que aparecem nas áreas USER.

OBS:
Os membros de um mesmo par de atribuição são mutuamente exclusivos, não podendo ser R/O e R/W ou SYS e DIR.

FORMATOS DO COMANDO STAT:

STAT RETURN

Informa as atribuições dos disquetes e o espaço livre de cada drive acessado até então.

EXEMPLOS:

```
A>STAT RETURN
A:R/W, Espaço :30k
B:R/O, Espaço :120k
A>-

STAT D: RETURN
Espaço livre do disquete montado no drive D:.

A>STAT B:
Bytes Restantes Em B: 38k
A>-
```

STAT D:Arquivo.tip RETURN

Espaço ocupado e atribuição do arquivo especificado, podendo também conter "*" e "?" para agrupar os arquivos. A identificação do drive é opcional, se omitido o drive corrente é assumido.

EXEMPLOS:

```
A>STAT DDT.COM RETURN
Regs  Bytes  Ext  Acc
 38     6k     1  R/W A:DDT.COM

A>STAT B:*.COM RETURN
Regs  Bytes  Ext  Acc
 12     2k     1  R/W B:COPIA.COM
 38     6k     1  R/W B:DDT.COM
 54     8k     1  R/W B:(ED.COM)
  6     2k     1  R/W B:DUMP.COM
 10     2k     1  R/W B:FORMATE.COM
160    20k     1  R/W B:M80.COM
 16     2k     1  R/W B:LOAD.COM
 42     6k     1  R/W B:STAT.COM
 58     8k     1  R/W B:PIP.COM
 12     2k     1  R/W B:SUBMIT.COM
  8     2k     1  R/W B:XSUB.COM
Bytes Restantes Em B: 130k

A>STAT B:EXEMPLO.?X? RETURN
Regs  Bytes  Ext  Acc
 42     6k     1  R/O B:EXEMPLO.TXT
 16     6k     1  R/W B:EXEMPLO.EXP
Bytes Restantes Em B: 130k
A>-
```

Regs – Número de gravações de blocos de 128 bytes.
Bytes – Tamanho do arquivo em Bytes, 1k = 1024 bytes.
Ext – Número de extensões ocupadas pelo arquivo.
Acc – Atribuição de acesso ao arquivo R/W ou R/O.
B:(ED.COM) – Os parênteses indicam que o arquivo tem atribuição de SYS.

STAT D:Arquivo.tip $atr RETURN

Altera o parâmetro de atribuição (R/O-R/W-DIR ou SYS) do arquivo especificado, podendo também serem utilizados os caracteres "*" e "?" para agrupar os arquivos.
A identificação do drive poderá ser omitida caso o drive seja o drive corrente.

EXEMPLOS:

```
A>STAT PROPOSTA.TXT $R/O RETURN
PROPOSTA.TXT set to R/O

A>STAT B:ED.COM $SYS
B:ED.COM set to SYS

A>STAT C:*.* $R/W RETURN
EXEMPLO1.EXP set to R/W
EXEMPLO2.ABC set to R/W
ABCDEF.GHI   set to R/W
A>-
```

STAT D:DSK: RETURN

Informação do formato de armazenamento do drive D:.

EXEMPLO:

```
A>STAT A:DSK: RETURN
       A: Características Drive;
    1520: Capac. Regist. 128 Bytes;
     190: Capac. Drive Kilobytes;
      64: 32 Bytes/entrada diretório;
      64: Entradas checadas diretório;
     256: Registros/Extensão;
      16: Registros/Bloco;
      40: Setores/Trilha;
       2: Trilhas reservadas;
       A>-
```

As informações fornecidas por este último formato do comando STAT não poderão ser alteradas sem que seu sistema seja modificado, i.e., o **DGP/M**.

STAT D:=R/O RETURN

Atribuir ao drive D: proteção temporária contra gravação.

EXEMPLO:

```
A>STAT B:=R/O RETURN
A>SAVE 3 B:TESTE.ARQ RETURN
BDOS ERR ON B:R/O
A>-
```

NOTAS:

1 – Ao tentar gravar qualquer informação em um disquete com atribuição R/O, ou sobre um arquivo com a mesma atribuição, a mensagem de BDOS ERR ON D: R/O será enviada ao vídeo, onde D: identifica o drive.

2 – Se em um drive existir um disquete com atribuição R/O e este for substituído por algum outro que não tenha esta atribuição, o sistema não o reconhecerá como R/W, até que o sistema seja ressetado. Neste caso, tecle **CTRL C** após a troca.

3 – Se o disquete estiver com a tarja de proteção contra gravações, os arquivos com atribuição R/W ainda continuarão com esta atribuição, pois, o sistema não é capaz de detetar esta proteção física até que seja enviada ao drive uma instrução para gravar.

COMANDO PIP

PIP - PERIPHERAL INTERCHANGE PROGRAM

PIP é utilizado para copiar arquivos de um "lugar" a "outro", com possibilidade de alterar o arquivo conforme os parâmetros especificados no comando.

Há duas maneiras de executar o PIP:

1 – Comando digitado em uma só linha.

```
A>PIP comando RETURN        ;comando de uma linha.
A>-
```

2 – Carregar o programa PIP e executar um conjunto de comandos, um de cada vez.

```
A>PIP RETURN                ;carregar PIP.
*comando1 RETURN            ;executar comando 1.
*comando2 RETURN            ;executar comando 2 ...
*CTRL C                     ;retornar ao DGP/M.
A>-
```

Formato geral do comando PIP:
DESTINO = FONTE (parâmetros)

Funções do PIP:

- Copiar um arquivo de um disquete em outro.
- Copiar um arquivo alterando o nome.
- Copiar um grupo de arquivos em outro disquete.
- Copiar todos os arquivos.
- Criar um arquivo através da concatenação de outros.
- Copiar uma parte do arquivo.
- Copiar um arquivo R/O.
- Passar pelo vídeo o conteúdo do arquivo em processo.
- Copiar arquivos em ASCII transformando caracteres maiúsculos em minúsculos ou vice-versa.

FORMATOS VÁLIDOS:

PIP RETURN

Traz à memória o programa PIP e fica à espera de um comando. Ao teclar **RETURN** ou **CTRL C** no início de qualquer linha, o controle passará ao **DGP/M**.

PIP D1:novo.tip=D2:antigo.tip(p) RETURN

O arquivo especificado por D2:antigo.tip será copiado em outro com a especificação D1:novo.tip, utilizando os parâmetros (p)

EXEMPLOS:

```
A>PIP CONTAS.PCL=B:PAGAM.PCL(V)
A>-
```

O arquivo PAGAM.PCL, existente no drive B, será copiado em outro com nome CONTAS.PCL no drive corrente A, utilizando o parâmetro (V) para verificar o destino.

```
A>PIP A:=B:*.*(V) RETURN
COPYING -
CARTA1.TXT
CARTA2.TXT
CARTA3.TXT
A>-
```

Neste exemplo, todos os arquivos existentes no drive B: foram copiados no drive A:, com verificação.

PIP D1:novo.tip=D2:antigo1.tip(p),D3:antigo2.tip(p) RETURN

Será criado um arquivo novo.tip resultante da concatenação dos arquivos antigo1.tip e antigo2.tip, i.e., a vírgula no comando PIP indica concatenação.

EXEMPLO:

```
A>PIP B:CARTAS=CARTAS1.TXT(V),CARTAS2.TXT(V) RETURN
A>-
```

O arquivo CARTAS será formado por CARTAS1.TXT e CARTAS2.TXT concatenados nesta ordem.

PIP disp:=D:arquivo.tip(p) RETURN

O conteúdo do arquivo.tip será enviado ao dispositivo especificado por "disp".

Dispositivos:
LST - impressora
CON - terminal

EXEMPLO:

```
A>PIP LST:=B:RESUMO.TXT RETURN
A>-
```

Uma cópia do arquivo RESUMO.TXT será enviado à impressora. Este comando poderá ser cancelado ao se pressionar qualquer tecla.

PIP D:arquivo.tip=disp:(p) RETURN

Este formato do comando PIP permite que sejam arquivadas todas as entradas do dispositivo especificado. Este comando poderá ser cancelado ao se pressionar qualquer tecla.

EXEMPLO:

```
A>PIP D:arquivo.
Agora digite a sentença abaixo:
TODAS AS ENTRADAS SERAO GRAVADAS RETURN
NO ARQUIVO ENTRADA.TCL CTRL Z
A>-
```

No exemplo acima foi criado um arquivo contendo a sentença digitada.

OBS:
CTRL Z termina a transferência e marca final de arquivo.
RETURN ou **LF** ↓ poderão ser usados para terminar cada linha.

PARÂMETROS DO COMANDO PIP

Em geral, somente um parâmetro é usado em cada comando PIP, no entanto, poderá ser utilizada uma combinação de 2 ou mais parâmetros entre parênteses, por exemplo: (VEL).

(Dn)
Deleta todos os caracteres a partir da coluna "n". Este número deve ser especificado logo após a letra "D", (ex.D30). Após cada **RETURN** ou **LF**↓ encontrado, o PIP começa a contar os caracteres. Ao encontrar o enésimo caracter, todos outros à sua frente, até encontrar outro **RETURN** ou **LF**↓, são desconsiderados. "n" poderá assumir valores entre 1 e LF↓ encontrado, e o PIP começa a contar arquivos processados pelo comando, estes serão mostrados no vídeo.

(F)
A opção (F) fará com que todos caracteres (0CH - form feed) sejam ignorados no processo de cópia.

(Gn)
Copia arquivos da área "n" de usuário. "n" deve ter valores entre 0 e 15, veja o comando **USER**.

(L)
Converte caracteres alfabéticos "A" até "Z" em caracteres minúsculos.

(N)
Acrescenta o número da linha em cada linha transferida. A cada **RETURN** ou **LF**↓ encontrado, o número da linha é incrementado. A opção N2 adiciona zeros à esquerda do número da linha, formando um conjunto de 6 caracteres.

(O)
Usado na transferência de arquivo fora do formato ASCII ou programa objeto. Este parâmetro não é necessário quando o arquivo for do tipo .COM, porque o comando PIP já assume que os arquivos **.COM** não estão no formato ASCII. O uso do parâmetro (O) diz ao comando PIP para tratar o caracter CTRL Z (1A hexa) como qualquer outro caracter, caso contrário, poderia ser interpretado como um fim de transferência.

(Pn)

Emite um FF (0CH form feed) após a enésima linha especificada. Geralmente as impressoras não imprimem um arquivo saltando o corte do formulário contínuo, para fazê-lo é usado este parâmetro.

FORMATOS:

1 – 3 linhas no topo do formulário em branco.

2 – texto "n" linhas.

3 – 3 linhas no pé do formulário em branco.

"n" poderá assumir valores entre 1 e 255. Se "n" não é especificado, 60 é assumido.

Se o parâmetro (F) está presente junto com (P), todos os FF (form feed) do arquivo são retirados e inseridos nos locais especificados por (P).

(Qstring CTRL Z)

Copia o arquivo até a string especificada no comando. Se este parâmetro é especificado em uma linha de comando do **DGP/M**, a string é convertida em maiúscula. Se digitado como um comando de dentro do PIP, esta conversão não ocorrerá.

(Sstring CTRL Z)

Copia parte do arquivo iniciando pela string especificada. São válidas as mesmas considerações feitas para o parâmetro (Q). Quando os parâmetros (S) e (Q) são utilizados no mesmo comando, pode-se copiar somente uma parte do arquivo.

(E)

Todos os arquivos processados pelo comando serão também mostrados no vídeo.

(R)

Parâmetro exigido na cópia de arquivos com atribuição SYS.

(U)

Transforma caracter alfabético minúsculo em maiúsculo.

(V)

Parâmetro usado para indicar ao sistema, que deverá ser checado o arquivo criado com o conteúdo da memória.

(W)

Possibilita a cópia sobre um arquivo com atribuição R/O.

COMANDO ED

ED - EDITOR DE TEXTOS

Esta é a parte mais importante em seu sistema **DGP/M**, o editor, para que você possa editar seus programas em FORTRAN, COBOL, BASIC ou qualquer outro texto.

Para carregá-lo digite:
A>ED arquivo.tip RETURN

Primeiro, o programa ED.COM é carregado na memória e cria-se um arquivo temporário com a especificação arquivo.$$$ e então, o sistema ficará à espera de um comando.

Após processado todo o texto, o arquivo original (arquivo.tip) é renomeado para arquivo.BAK e o arquivo temporário (arquivo.$$$) é renomeado para arquivo.tip. O arquivo.BAK poderá ser utilizado caso as modificações efetuadas anteriormente não sejam desejadas.
Esta é uma segurança a mais que o editor fornece.

CONVENÇÕES:

(±) Caracteres opcionais +(mais) ou -(menos). Se omitido +(mais) é assumido.
(n) Número entre 0 e 65535 inclusive. O caracter 0(zero) especial para alguns comandos e o símbolo # substitui 65535.

COMANDOS:

nA

Aloca espaço e carrega o número de linhas "n" do arquivo.

#A – Traz todo o arquivo para a memória;

A – Traz somente uma linha;

0A – Ocupará no máximo metade do espaço reservado na memória para edição do texto.

nW

Grava no arquivo temporário "n" linhas.

W – Grava somente uma linha.

#W – Grava todo arquivo existente na memória.

0W – Grava pelo menos a metade do arquivo existente na memória. O comando W sempre inicia a gravação a partir da primeira linha existente na memória, então as linhas gravadas cedem espaço para complementação do texto.

E

Termina a edição. Todo texto restante na memória é gravado no arquivo temporário como no comando W.

O comando E deverá ser executado sozinho.

H

Move para o início do arquivo editado.

1 – Todo o texto restante na memória é gravado no arquivo temporário.

2 – Restante do arquivo original é passado ao arquivo temporário.

3 – O tipo do arquivo original é renomeado para .BAK.

4 – O arquivo temporário passa a ter agora, o mesmo tipo do arquivo original anterior.

5 – Um novo arquivo temporário vazio é criado.

6 – A memória agora está livre, reinicie com o comando A.

Este comando é ideal para se acrescentar vários segmentos a um texto. O comando H deve ser executado como um comando sozinho.

O

Apaga o arquivo editado.

1 – O conteúdo da memória e o arquivo temporário são apagados.

2 – Reinicie a execução com o comando A. O arquivo original não será perdido.
ED perguntará "O - (Y/N)?".
Se a resposta for negativa (N), o comando não será executado.

Q

Não executa qualquer alteração no arquivo.
ED perguntará "Q - (Y/N)?" antes de proceder o comando.
Caso a resposta for afirmativa (Y), o arquivo original não sofrerá qualquer alteração e o comando passará ao **DGP/M**, caso contrário, a edição continuará normalmente.

R

O conteúdo do arquivo criado pelo comando X é introduzido no texto corrente a partir da posição atual do cursor.

R nomelib

Lê um arquivo .LIB especificado, colocando-o logo após a posição atual do cursor.

nX

Cria um arquivo temporário **X$$$$$$$.LIB** com "n" linhas a partir da linha corrente. As linhas não são retiradas da memória.
Este arquivo é chamado novamente à memória utilizando o comando R, e é apagado tão logo a edição do texto seja terminada por um dos comandos E, Q ou O.

COMANDOS DE EDIÇÃO:

±B – Move o cursor para a primeira ou última linha do texto. +(mais) início, -(menos) final.

±nC – Move o cursor para frente ou para trás "n" caracteres.

±nD – Deleta "n" caracteres imediatamente anteriores ou posteriores à posição do cursor.

I RETURN - Entra no modo de inserção.

Istring CTRL Z - Insere uma string a partir do cursor.

Istring RETURN - Insere uma linha.

±nK – Apaga "n" linhas anteriores ou posteriores à atual.

±nL – Posiciona o cursor "n" linhas à frente ou atrás.

±nP – Move o cursor "n" páginas à frente ou atrás.

±nT – Lista "n" linhas anteriores ou posteriores à atual.

±U – Converte caracteres minúsculos em maiúsculos.
+U = inicia a conversão.
-U = termina a conversão.

±V – +V = todas as linhas são numeradas.
-V = as linhas não são numeradas.

0V – Função especial utilizada para informar o espaço ainda restante para edição do texto.

n: – Este comando move o cursor para a linha "n".

COMANDOS DE ATUALIZAÇÃO:

nFstring CTRL Z
Procura a enésima ocorrência da string especificada. A posição corrente do cursor, agora, será um caracter após a string encontrada.

nNstring CTRL Z
Procura a enésima ocorrência da string especificada começando pelo arquivo temporário, a memória e o arquivo original.
O comando F somente procura na memória.

nSstring1 CTRL Zstring2 CTRL Z
Substitui as "n" primeiras ocorrências da string1 por string2 começando pela posição atual do cursor. A procura somente será feita na memória.

nJstring1 CTRL Zstring2 CTRL Zstring3 CTRL Z
Este comando executa "n" vezes a procura da string1, introduzindo logo após a string2 e deleta todos os caracteres até encontrar a string3.

Os comandos de edição e de atualização podem ser usados combinados em uma só linha seguindo as regras abaixo:

1 – Comandos E, H, O e Q devem ser executados sem combinação com outros comandos.

2 – Em comandos que utilizam strings e CTRL Z, como F, I, J, N e S, deverá ser teclado **RETURN** somente no final da linha.

Um conjunto de comandos poderão ser executados repetidas vezes, utilizando um macro-comando:
nMcomando1comando2comando3 RETURN

Se "n" for omitido ou se n=1 ou n=0, os comandos serão executados até que qualquer tecla seja pressionada ou até que termine um comando.

COMANDO DUMP

DUMP enviará ao vídeo todo o arquivo especificado no formato hexadecimal.

Formato geral:
DUMP D:arquivo.tip RETURN

EXEMPLO:

```
   A>DUMP B:FORM.COM RETURN

0000  C3 33 04 20 20 20 4D 69 63 72 6F 73 6F 6C 20 31
0010  39 38 32 20 20 20 20 20 20 20 20 20 20 20 20 20
0020  20 20 20 20 20 20 20 20 20 20 3F 3F 3F 3F 3F 3F
0030  3F 3F 3F 3F 3F 3F 00 00 00 43 4F 4E 3A 52 44 52
0040  3A 50 55 4E 3A 4C 53 54 3A 44 45 56 3A 56 41 4C
0050  3A 55 53 52 3A 44 53 4B 3A 54 54 59 3A 43 52 54
0060  3A 42 41 54 3A 55 43 31 3A 54 54 59 3A 50 54 52
0070  3A 55 52 31 3A 55 52 32 3A 54 54 59 3A 50 54 50
0080  3A 55 50 31 3A 55 50 32 3A 54 54 59 3A 43 52 54
0090  3A 4C 50 54 3A 55 4C 31 3A 52 2F 4F 00 52 2F 57
00A0  00 53 59 53 00 44 49 52 00 52 2F 4F 20 52 2F 57
00B0  20 53 59 53 20 44 49 52 20 2A 2A 20 41 62 6F 72
00C0  74 61 64 20 2A 2A 00 55 73 75 61 72 20 41 74 69
00D0  76 6F 20 3A 00 46 69 6C 65 73 20 41 74 69 76 6F
00E0  73 3A 00 20 20 20 20 00 20 43 61 72 61 63 74 65
00F0  72 69 73 74 69 63 61 73 20 44 72 69 76 65 00 36
0100  35 35 33 36 3A 20 00 43 61 70 61 63 2E 20 52 65
0110  67 69 73 74 2E 20 31 32 38 20 42 79 74 65 73 00
```

Os quatro primeiros dígitos identificam a posição relativa de cada byte no arquivo.

OBS:
CTRL S interromperá a listagem até que seja pressionada novamente. Qualquer outra tecla pressionada fará com que o controle seja passado ao **DGP/M**.

COMANDO SUBMIT

Este comando permite que um conjunto de comandos sejam executados automaticamente. Antes de utilizar o comando SUBMIT, deverá ser criado um arquivo do tipo .SUB, usando o editor. O arquivo deverá conter uma lista de comandos que serão executados pelo sistema.

Formato geral do comando:
SUBMIT arquivo p1 p2 p3 RETURN

Um máximo de nove parâmetros poderão ser especificados. O processo de compilação de programas pode ser grandemente simplificado com a utilização do SUBMIT.
Veja exemplo após a apresentação do XSUB.

COMANDO XSUB

Substituição automática de parâmetros exigidos por um programa executado em SUBMIT. A declaração XSUB deverá ser anterior ao nome do programa a ser executado.
O exemplo a seguir servirá para você compilar um programa em FORTRAN. Faça um arquivo do tipo .SUB usando o editor de textos, contendo:

```
1: XSUB                    ;ativar XSUB.
2: FORTRAN                 ;carregar compilador FORTRAN.
3: $1,TTY:=$1/N            ;compilar programa $1.
4: LINK                    ;carregar LINK.
5: $1,$1/N/E               ;executar linkagem e criar objeto
                            executável com extensão .COM.
```

Supondo que você tenha dado o nome de CPLFOR.SUB para este arquivo, para compilar digite:
SUBMIT CPLFOR nome

Onde "nome" é o nome do programa fonte editado. Não digite a extensão .FOR do programa fonte. Note que, o parâmetro "nome" substituiu todos $1 na execução do SUBMIT.

UTILITÁRIOS

DDT - DYNAMIC DEBUGGING TOOL

DDT é usado para "debugar" programas em linguagem de máquina.

Podendo executar:

- Carregar programas assemblados na memória.
- Procurar possíveis erros em programas.
- Executar pequenas modificações.
- Fazer correções e atualizações do sistema.
- Alterar parâmetros do sistema.
- Examinar e modificar conteúdo da memória.
- Assemblar programas em mneumônicos de INTEL 8080.
- Desassemblar partes de programas.
- Examinar e modificar os conteúdos dos registradores.
- Executar o programa examinando os registradores passo a passo.

Para iniciar DDT, digite:
A>DDT RETURN

Ou então:
A>DDT D:arquivo.tip RETURN
para carregar o arquivo especificado.

TABELA DE COMANDOS:

FUNÇÕES	LETRA
Assemblar instruções.	A
Verificar conteúdo da memória.	D
Preencher memória.	F
Executar programa.	G
Aritmética hexadecimal.	H
Identificar programa.	I
Listar instruções desassembladas.	L
Mover conteúdo da memória.	M
Ler o arquivo identificado em I.	R
Executar o programa no modo TRACE.	T
Executar parcialmente.	U
Examinar conteúdo dos registradores.	X

Ahhhh RETURN

Assemblar instruções. Neste comando você poderá entrar com mneumônicos de INTEL 8080 iniciando pelo endereço "hhhh". Mneumônico e o operando deverão ser separados por um espaço, terminando a linha com **RETURN**. Após a entrada de cada instrução, o próximo endereço será mostrado ficando à espera de uma nova instrução. Para finalizar, simplesmente tecle **RETURN** como primeiro caracter de uma linha.

EXEMPLO:

```
A0100 RETURN
0100  MOV E,A RETURN
0101  MOV A,0 RETURN
0102  RETURN
```

Diiii,jjjj RETURN

O conteúdo da memória é mostrado no vídeo iniciando por iiii até jjjj e com os caracteres ASCII correspondentes à direita. Caracteres não representados em ASCII serão representados por . (ponto).
Se iiii, jjjj forem omitidos, serão mostrados 192 bytes a partir do endereço corrente. O endereço corrente no início do programa é 0100. Caso somente o endereço jjjj for omitido serão mostrados os 192 bytes a partir de iiii.

```
-D
0100 C3CE 04C3 0A01 C340 011A 3ACF 01B7 C214 .......@..:.....
0110 01CD 9301 CD06 DEB7 CA35 01CD 09DE 4F3A .........5....O:
0120 D001 B7CA 2B01 C5CD 0CDE C1C5 CD36 DEC1 ....+........6..
0130 7932 0901 C9CD BB01 CA14 0179 3209 01C9 y2.........y2...
0140 C53A DE01 B7C2 8401 3ACF 01B7 C252 01CD .:......:....R..
0150 9301 CD06 DEB7 CA7E 01CD 09DE 4FFE 1AC2 .......~....O...
0160 6A01 3EFF 32CF 01C3 5201 FE02 CA84 01C5 j.>.2...R.......
0170 CD36 DEC1 3AD0 01B7 CA7E 01CD 0CDE CDBB .6..:....~......
0180 01C3 5201 3EFF 32DE 01C1 C5CD 0CDE C1CD ..R.>.2.........
0190 36DE C921 D101 060D 4EC5 CD0C DEC1 2305 6..!....N.....#.
01A0 C298 01CD 09DE FE59 C2B0 013E FF32 D001 .......Y...>.2..
01B0 0E2B CD0C DE3E FF32 CF01 C9DB 59E6 02C8 .+...>.2....Y...

-D500
0500 32C0 1E11 0000 0E19 CD05 0032 FC1D 31F2 2..........2..1.
0510 1DCD 401A 3AC0 1E32 C11E 216F 1F36 002B ..@.:..2..!o.6.+
0520 3600 2B36 0021 A61E 3601 2336 0021 F31D 6.+6.!..6.#6.!..
0530 3600 2336 FE3A A51E 1FD2 4705 0E2A CD1C 6.#6.:....G..*..
0540 08CD 6F09 CD2E 0821 4E1F 36FF 3ACC 1EFE ..o....!N.6.:...
0550 00C2 5E05 2AFC 1D4D CD5E 08CD 0000 214B ..^.*..M.^....!K
0560 1E36 0021 031E 3600 21A4 1E36 002B 3600 .6.!..6.!..6.+6.
0570 0127 1ECD 2012 3AA9 1EFE 03C2 8105 C324 .'.. .:........$
0580 063A A91E FE04 C2C3 053A 941F 3D32 4B1E .:.......:..=2K.
0590 CD0C 1D01 061E CD20 123A A91E FE02 CAA4 ....... .:......
05A0 05CD 5C18 3AF5 1D1F D2B7 0501 5C00 CDEE ..\.:.......\...
05B0 1CCD 7818 C3C0 0501 271E CDEE 1CCD 821A ..x.....'.......
```

Fiiii,jjjj,aa RETURN

Preenche a memória de iiii até jjjj, inclusive com o hexadecimal aa. Este comando é muito usado para preencher com (00) hexa antes de trabalhar em qualquer nova área de memória. Os primeiros 256 bytes (0000 a 00FF) e a partir de C000 são reservados ao sistema.

G RETURN

Este comando permite iniciar a execução do programa em revisão (endereço inicial 0100).

Variações do comando G:

```
    Giiii RETURN - endereço inicial iiii.
Giiii,jjjj RETURN - iniciar por iiii finalizar em jjjj.
   G,jjjj RETURN - endereço inicial corrente, final jjjj.
```

Haaaa,bbbb RETURN

Aritmética hexadecimal. Serão executados a soma e a diferença em 16 bits. O primeiro valor corresponde à soma e o segundo à diferença.

Iarquivo.tip

Este comando identifica o arquivo que será lido ao utilizar o comando R.

L RETURN

Lista instruções desassembladas no formato INTEL 8080 a partir do endereço corrente. Onze instruções serão listadas na execução do comando. Caso seja encontrado um hexadecimal que não corresponda a qualquer instrução do processador INTEL 8080, será mostrado: ??=##, onde ## é o hexadecimal. Como no comando D, poderão ser especificados os endereços, inicial e final no seguinte formato: Liiii,jjjj. Control S dará uma pausa na execução do comando, pressione novamente Control S para reiniciar.

```
 L
0100   JMP   04CE
0103   JMP   010A
0106   JMP   0140
0109   LDAX  D
010A   LDA   01CF
010D   ORA   A
010E   JNZ   0114
0111   CALL  0193
0114   CALL  DE06
0117   ORA   A
0118   JZ    0135
```

Miiii,jjjj,kkkk RETURN

Este comando permite copiar um bloco de memória para outro local, onde iiii é o endereço inicial do bloco a ser movido, jjjj é o endereço final do bloco a ser movido e kkkk é o endereço inicial da cópia. Tenha o cuidado para não alterar os conteúdos dos endereços de 0000 a 00FF e C000 a FFFF, pois fazem parte do sistema.

Riiii RETURN

Ao executar o comando R, o conteúdo do arquivo identificado em "I" será carregado na memória a partir do endereço iiii. Se iiii for omitido, 0100 é assumido.

Siiii RETURN

Este comando permite alterar o conteúdo da memória a partir de iiii. O endereço e o conteúdo serão mostrados e se desejar alterá-los, entre com o novo valor e tecle **RETURN**, ou então, apenas tecle **RETURN** e o próximo endereço será mostrado. Para finalizar digite um caracter inválido seguido de **RETURN**, por exemplo: **# RETURN**.

EXEMPLO:

```
      S0850 RETURN          ;iniciar pelo endereço 0850.
      0850 CD RETURN        ;este continua.
      0851 3D 3F RETURN     ;alterar 3D para 3F.
      0852 05 03 RETURN     ;alterar 05 para 03.
      0853 CA # RETURN      ;finalizar o comando.
```

Tn RETURN

Executa o programa no modo TRACE. A cada instrução executada, o conteúdo dos registradores são mostrados. "n" informa o número de instruções a serem executadas.
A velocidade de execução cai aproximadamente 500 vezes, pois, o programa DDT simula cada instrução. Qualquer tecla pressionada interromperá a execução do comando.

Formato da linha do vídeo:

```
C0Z0M0E0I0 A=02 B=0354 D=0AF3 H=B000 S=0150 P=01FA LXI SP,0200
```

onde:

C0 –	Flag C é zero;
Z0 –	Flag Z é zero;
M0 –	Flag M é zero;
E0 –	Flag P/V é zero;
I0 –	Flag I é zero;
A=02 –	Acumulador com 02H;
B=0354 –	Par BC com 0354H;
D=0AF3 –	Par DE com 0AF3H;
H=B000 –	Par HL com B000H;
S=0150 –	SP com 0150H;
P=01FA –	PC com 01FAH;
LXI SP,0200 –	próxima instrução a ser executada.

Un RETURN

Este comando tem a mesma função do comando T, porém, são mostrados os conteúdos dos registradores no início da execução. A execução é mais rápida que do comando T e mais lenta que o comando G.

X RETURN

Examina o conteúdo dos registradores.

```
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0000 P=0100 LXI SP,0200
```

XC – Flag C
XZ – Flag Z
XM – Flag M
XE – Flag P/V
XI – Half carry flag
XA – Acumulador
XB – Par BC
XD – Par DE
XH – Par HL
XS – Stack pointer
XP – Program counter

UTILITÁRIO NOME

Ao se carregar um disco, uma mensagem é mostrada. Tal mensagem pode ser modificada através deste utilitário.

Sintaxe:
NOME (drive)
A especificação do drive é opcional. Se omitida, o drive corrente é assumido.

Modo de usar:
Após ser ativado, o utilitário mostra o nome atual do disco, e pede seu novo nome:

```
Nome atual: Digitus
Novo nome: (termine com @)
```

Se não se deseja modificar o nome, tecla-se apenas **RETURN**, caso contrário, tecla-se o novo nome. Este pode ter mais de uma linha. Para tanto, deve-se ao final de cada linha, teclar **RETURN**. Para terminar a especificação do nome deve-se teclar **@** e **RETURN**.

Limitações:
O tamanho máximo do nome é 127 caracteres. Lembre-se que cada mudança de linha ocupa 2 caracteres (CR e LF).

UTILITÁRIO AUTO

Função:
Especifica um comando que é automaticamente executado, quando da carga inicial do sistema.

Sintaxe:
AUTO (drive)

A especificação do drive é opcional. Se omitida, é assumido o drive corrente.

Modo de usar:
Após ser ativado, o utilitário mostra o comando atual e pede a especificação do novo comando:

```
Comando atual: DIR
Novo comando :
```

Um novo comando deve então ser teclado. Se teclar simplesmente **RETURN**, o comando passa a ser um comando nulo. Caso não deseje alterar o comando anterior, deve-se teclar C.

UTILITÁRIO CONFIG

Introdução

O utilitário configura o sistema em 4 áreas:

- Discos;
- Interface serial;
- Impressora;
- Teclado.

A configuração de discos divide-se em 2 partes: a física e a lógica. Na configuração física dos discos é informado ao utilitário o número de drives existentes e a natureza destes (número de trilhas e faces). Na configuração lógica associa-se formatos de discos aos drives.

Na configuração da interface serial é especificado o "baud-rate", tamanho da palavra, etc.

Na configuração da impressora é especificado de que modo ela é ligada (na porta paralela ou na porta serial), se ela possui "line-feed" automático, etc.

Na configuração do teclado é possível reprogramar o teclado (por exemplo, as teclas de setas).

A configuração é feita mudando-se alguns setores nas trilhas de sistema. Quando o sistema é carregado pela primeira vez (cold-start), estes setores são lidos e as opções neles contidas são ativadas.

Modo de usar:

Após a ativação do utilitário, é apresentado um **MENU** com as possíveis configurações:

0 – Configuração física dos drives;
1 – Configuração lógica dos drives;
2 – Configuração da interface serial;
3 – Configuração da impressora;
4 – Configuração do teclado;
BREAK – Termina.

Daqui por diante, este **MENU** será referenciado como **MENU PRINCIPAL**.

Selecionada uma configuração, o utilitário pede a origem do sistema a ser configurado, um dos drives ou a memória (está somente na configuração dos drives).

Entre com o sistema FONTE:
A a F – para drive;
M – para memória;
BREAK – para termina.

Teclando-se BREAK, retorna-se ao **MENU PRINCIPAL**.

Selecionando-se um dos drives, é verificado se ele é válido (se existe, contém sistema, etc.). Caso não seja válido, uma mensagem de erro é mostrada e volta-se à seleção do sistema FONTE. Sendo válido, passa-se à configuração anteriormente selecionada.

Selecionando-se a memória, o sistema atualmente residente é utilizado como fonte para configuração.

Após a configuração ter sido feita, o utilitário pede o destino do sistema configurado:

Entre com o sistema DESTINO:
A a F – para drive;
M – para memória;
BREAK – para terminar;

Como na seleção do sistema FONTE, apenas a configuração de disco (opções 0 e 1 do **MENU PRINCIPAL**) contém a opção de memória. Sendo esta selecionada, o sistema atualmente residente é reconfigurado.

Caso um drive seja selecionado, é feita a mesma consistência na especificação do sistema FONTE, e sendo válido a configuração é gravada no disco deste drive.

Após a seleção do sistema DESTINO, o utilitário pede a especificação de outro, até que se tecle BREAK, quando então volta-se ao **MENU PRINCIPAL**.

É importante notar, que a menos que a memória seja utilizada como destino, a configuração só é ativada no próximo "cold-start" no disco reconfigurado (exceto na configuração de teclado, quando é válido no próximo "warm-start").

CONFIGURAÇÃO FÍSICA DOS DRIVES

Entende-se por drive físico o drive propriamente dito, ou seja, a unidade onde se coloca os disquetes. Os drives físicos são numerados de 0 a 3, sendo o drive 0, o mais perto do computador, é onde o disco sistema é inserido. Existem vários tipos de drives de 5 1/4" (único tamanho com o qual o sistema trabalha), porém, 3 são mais comuns: Face simples, 40 trilhas; Face dupla, 40 trilhas; Face dupla, 80 trilhas.

Na configuração física dos drives especifica-se quantos drives físicos existem e a natureza deles.
Inicialmente o utilitário mostra a configuração atual dos drives físicos.
Por exemplo:

```
DRIVE  TRILHAS  FACES  STEP-RATE
  0      40       1      06ms
  1      40       1      20ms
  2      80       2      03ms
  -      --       -      ----
```

Nesta configuração dada como exemplo, vemos que o drive 3 não está alocado (não existe). O parâmetro step-rate significa o tempo de acesso de uma trilha a outra, e é uma característica do drive. Quanto menor o step-rate, mais rápido será feito o acesso ao disco. Dos modelos existentes no mercado, os "Flexidisk" de 40 trilhas, são os únicos que aceitam no mínimo o step-rate de 20ms.

Após mostrar a configuração atual, o utilitário pede na última linha, o drive a ser configurado: Drive (0-3)
Selecionado um drive, é pedido em seguida o número de trilhas, número de faces e step-rate. Um número de trilhas igual a zero desaloca o drive selecionado.
Qualquer drive físico pode ser desalocado, exceto o drive 0, que é o drive do qual o sistema é carregado quando se liga o microcomputador. Poderíamos ter um sistema com somente os drives 0 e 3 alocados, por exemplo.
Teclando-se **BREAK**, encerra-se a configuração física dos drives.

CONFIGURAÇÃO LÓGICA DOS DRIVES

Enquanto um drive físico é algo realmente físico, um drive lógico não chega a tanto. Entende-se por drive lógico um certo formato de disco associado a um drive físico. Os drives lógicos são numerados de A a F. Normalmente, o drive lógico A está associado ao drive físico 0, o B ao 1, etc. Porém, isto não é rígido. Por exemplo, suponhamos um sistema com 2 drives físicos, 0 e 1. Uma associação possível seria:

Drive lógico A: formato Digitus, associado ao drive físico 0;
Drive lógico B: formato Digitus, associado ao drive físico 1;
Drive lógico C: formato Polymax, associado ao drive físico 1;
Drives lógicos D a F: não alocados.

Como se vê, tanto o drive B quanto o drive C estão associados ao drive 1. Porém, quando se acessa o B, assume-se que lá existe um disco no formato Digitus, e quando se acessa o C, assume-se que lá existe um disco no formato Polymax. É claro que deve-se acessar exclusivamente aos drives B e C, pois, afinal, ambos residem no drive 1.

Inicialmente o utilitário mostra a configuração atual dos drives lógicos e os formatos disponíveis.

EXEMPLO:

```
1 - A>0 : Digitus DD/FS/40t     B>1 : Digitus DD/FS/40t
    C>2 : Digitus DD/FS/80t     D   : (não alocado)
    E>1 : Polymax DD/FS/40t     F>2 : Itautec DD/FS/40t

2 - 0 - Digitus DD/FS/40t       1 - Digitus DD/FS/80t
    2 - Digitus DD/FD/40t       3 - Digitus DD/FD/80t
    4 - Polymax DD/FS/40t       5 - Itautec DD/FS/40t
```

No exemplo número 1, da página anterior, são mostrados os drives lógicos, indicando a qual drive físico o formato está associado. Por exemplo, o drive E está associado ao drive 1 com o formato 4. No exemplo 2 são mostrados os formatos disponíveis.

DD significa dupla densidade;	40T 40 trilhas;
FS face simples;	80T 80 trilhas;
FD face dupla.	

Após mostrar a configuração, o utilitário pede na última linha para se selecionar um drive: Drive (A-F)

Selecionado um drive, se este já está alocado, o utilitário pede para se selecionar um formato ou um drive físico:
Escolha o formato para o drive X (">" muda drive físico)

Deve-se, então, teclar o caracter correspondente ao formato desejado (3 para Digitus DD/FD/80t, por exemplo). Caso se tecle >, o utilitário pede para selecionar o drive físico:
Drive físico (0-3, - desaloca)

Seleciona-se, então, o drive físico, ou tecla-se (-) para desalocar o drive lógico selecionado.

Caso o drive lógico selecionado não esteja alocado, o utilitário pula a seleção de formato, indo direto para a seleção de drive físico e associando o formato 0.

Nem toda associação é possível, por exemplo, não se pode associar um formato de face dupla a um drive lógico cujo drive físico seja de face simples, porém, o contrário é possível (drive lógico face simples com um drive físico de face dupla). O mesmo acontece com drives de 40 e 80 trilhas. Qualquer associação inválida é detectada pelo utilitário, que não a aceita.

Encerra-se a configuração lógica dos drives teclando-se **BREAK**.

CONFIGURAÇÃO DA INTERFACE SERIAL

A interface serial é utilizada como os dispositivos "puncher" e "reader" no CP/M. Na configuração são especificados quatro parâmetros:

1 – Baud-rate;
2 – Tamanho da palavra;
3 – Número de stop-bits;
4 – Tipo de paridade.

Inicialmente o utilitário mostra a configuração atual da interface.

EXEMPLO:

```
      1 - 300 bauds
      2 - 7 bits/palavra
      3 - 1 stop bit
      4 - Sem paridade
```

Para se programar um dos quatro parâmetros, tecla-se o número correspondente (2 para tamanho da palavra, por exemplo).
Teclando-se BREAK, encerra-se a configuração da interface serial.

Para cada parâmetro existe um MENU com as opções possíveis. Tecla-se o número correspondente à opção desejada ou BREAK para não alterar. Logo após, a configuração atualizada é mostrada e volta-se ao passo seguinte.

Os MENUS são os seguintes:

Baud rates:

1 – 300
2 – 600
3 – 1200
4 – 2400
5 – 4800
6 – 9600

Tamanho da palavra:

5 bits
6 bits
7 bits
8 bits

Número de stop-bits:

1 stop bit
2 stop bits

Tipo de paridade:

0 – Sem paridade
1 – Paridade ímpar
2 – Paridade par

CONFIGURAÇÃO DA IMPRESSORA

São configurados três parâmetros:

1 – Porta na qual a impressora está ligada;
2 – Filtragem/conversão;
3 – Teste de fim de papel.

A porta pode ser paralela (Interface Centronics) ou serial. Caso a porta seja a interface serial, esta deve ser programada adequadamente. Existe ainda a porta "dummy" que significa que não existe impressora. Isto impede que um equipamento sem impressora fique travado quando algum programa tenta imprimir algo.

Algumas impressoras contêm uma opção de "auto line-feed", ou seja, após receberem o caracter de voltar o carro ("carriage-RETURN" ou CR), elas provocam um avanço de linha ("line-feed" ou LF). O sistema DIGDOS trabalha somente com este tipo de impressora, enviando somente o CR para mudar de linha.
Porém, o CP/M não foi feito para trabalhar com este tipo de impressora, enviando tanto o CR quanto o LF. Isto ocasiona espaçamento duplo quando se liga o computador a uma destas impressoras.

Através do parâmetro filtragem/conversão, pode-se especificar a filtragem dos CR's ou LF's (ou seja, não são enviados para impressora) ou a conversão da sequência de um ou mais CR's seguidos de um LF em um CR ou um LF. É possível também especificar nenhuma filtragem ou conversão, que é a opção preferível, por permitir o "over-print" (imprimir uma linha sobre outra), recurso utilizado por muitos editores de textos. Porém, esta última opção só pode ser utilizada em impressora sem "line-feed" automático.

Toda impressora contém um sensor de fim-de-papel, que serve para avisar à impressora e ao computador quando o papel acabou, de modo que a impressão seja suspensa. Quando se usa formulário contínuo (maioria dos casos), o sensor funciona perfeitamente, porém, quando se usa folhas separadas, ele acusa fim-de-papel quando se está imprimindo perto do fim da página.

Algumas impressoras (as Epsons, por exemplo) possuem a opção de ignorar o sensor, permitindo o uso mais prático de folhas separadas. Porém, apesar da impressora ignorar o fim-de-papel, continuando a imprimir, ela envia ao computador o código de fim-de-papel, que por sua vez para de enviar caracteres à impressora. Para evitar tal situação existe o parâmetro teste de fim de papel, que pode ser habilitado ou não. Este parâmetro só existe para impressoras ligadas na porta paralela.

No processo de configuração da impressora, inicialmente o utilitário mostra a configuração atual. Por exemplo:

```
1 - Impressora na porta paralela;
2 - Sem filtragem;
3 - NÃO testa fim de papel.
```

Para se programar um dos parâmetros, tecla-se o número correspondente. Teclando-se BREAK, encerra-se a configuração da impressora.

Para cada parâmetro existe um MENU com as opções possíveis. Tecla-se o número correspondente à opção desejada ou BREAK para não alterar. Após isto, a configuração atualizada é mostrada e volta-se ao passo seguinte.

Os MENUS são os seguintes:

Porta da impressora.

0 – "Dummy";
1 – Porta paralela;
2 – Porta serial.

Filtragem/conversão.

0 – Não filtra;
1 – Filtra CR;
2 – Filtra LF;
3 – Converte (CR LF) em CR;
4 – Converte (CR LF) em LF.

Teste de fim-de-papel.

1 – Testa;
2 – NÃO testa.

CONFIGURAÇÃO DO TECLADO

A cada tecla está associado um código. Por exemplo: à tecla com o rótulo A está associado o código 61 hexadecimal ("a" minúsculo em ASCII).

Porém, existem teclas que não produzem códigos e cuja função é modificar o código de outras, são elas: as teclas SHIFT, CTRL e a tecla sem rótulo (que chamaremos de CTRL2) que fica ao lado da tecla BREAK e acima da tecla CLEAR.
Assim, apertando-se simultaneamente a tecla SHIFT e a tecla A, gera-se o código 41 hexadecimal ("A" maiúsculo em hexa). As teclas CTRL e A, quando pressionadas simultaneamente, geram 01 hexadecimal, assim como CTRL2 e A, que também geram 01 hexadecimal.

Porém, esta associação entre teclas e códigos não é fixa, podendo ser reprogramada de acordo com a conveniência. É claro que não tem sentido reprogramar a tecla A, mas as teclas de setas é um exemplo típico.

Além de associar códigos às teclas, pode-se associar funções às teclas. Por exemplo, a tecla CLEAR está associada à função de "Limpar a tela", e as teclas CTRL e BREAK à função de causar um "Warm-start". Pode-se também ter teclas sem nenhum código associado, que seriam teclas nulas.

A programação do teclado é feita da seguinte forma: inicialmente o utilitário pergunta qual a tecla que vai ser programada, então, bata na tecla a ser programada.

Após alguma tecla (ou associação SHIFT tecla, CTRL tecla, CTRL2 tecla) ter sido pressionada, a seguinte tela é mostrada:

```
Programando a tecla de código xxh
Valor anterior = y (yyh)

Escolha novo valor:
0 - Não muda;                4 - (Limpa a tela);
1 - Entra código ASCII;      5 - (Caps Lock);
2 - Entra código em hexa;    6 - (Cancela).
3 - (Tecla nula);
```

O código xxh é um código interno, que não pode ser modificado e que é único para cada tecla. Serve apenas de referência.

Teclando-se "0" ou BREAK, a programação da tecla não é alterada.

Teclando-se "1", o utilitário pede: Bata na tecla ASCII.
Deve-se pressionar, então, a tecla cujo código a tecla programada deve gerar. Por exemplo, CTRL-S.

Teclando-se "2", o utilitário pede: Entre com valor em HEXA.
Deve-se entrar com qualquer valor hexadecimal menor de F8 (os códigos F8 a FF são reservados).

Teclando-se "3", a tecla passa a ser nula, ou seja, não gera nenhum código.

Teclando-se "4" a "6", associa-se à tecla a função selecionada. São elas:
Limpa tela – A tela é limpa, sem que nenhum código seja retornado;
Caps Lock – Comutada entre maiúsculas travadas ou não;
Cancela – Provoca um "warm-start".
Após uma das opções ter sido selecionada, o utilitário mostra: Programando a tecla de código xxh.
Valor anterior = y (yyh) - Valor atual = z (zzh)

E na última linha: Programa outra tecla? (S/N)
Respondendo-se "S" repete o processo, "N" ou BREAK termina a configuração do teclado.

APÊNDICE 1

BIOS

1 - Função Warm Boot

O sistema, agora, não é mais recarregado do drive A e sim do drive em uso, exceto quando este contiver um disco em um formato não DIGITUS, quando, como antes, o sistema é recarregado do drive A.

2 - Função Console Status

A função Console Status, agora, quando retorna em A 0FFh (existe caracter a ser lido), retorna em C o caracter que será lido.

3 - Função Console Output:

A função Console Output responde aos seguintes caracteres de controle:

```
01h : Limpa a tela;
02h : Home;
04h : Limpa até o fim da linha;
06h : Limpa até o fim da tela;
07h : Bell;
08h : Retrocede o cursor;
09h : Tabula;
0ah : Desce uma linha;
0bh : Sobe uma linha;
0ch : Limpa a tela;
0dh : Retorna o cursor para o início da linha;
0eh : Retrocede o cursor;
0fh : Avança o cursor;
12h : Insere uma linha em branco;
13h : Deleta linha;
1ah : Limpa a tela;
1bh : Inicia uma sequência de escape, que são as seguintes:
      esc = (lin+20h)(col+20h) ou esc Y (lin+20h)(col+20h) ou
      esc 12h (lin+20h)(col+20h) : posicionam o cursor;
      esc"~"k : deleta até o fim da tela;
      esc"~"K : deleta até o fim da linha;
      esc":"c : imprime o caracter c literalmente, mesmo
                se for um caracter de controle;
      esc"h"c : imprime o caracter c em hexadecimal;
      esc 10h : home;
      esc 11h : limpa tela;
      esc 13h : limpa até o fim da linha;
      esc 14h : limpa até o fim da tela;
      esc 15h : avança o cursor;
      esc 16h : retrocede o cursor;
      esc 1ch : liga o cursor;
      esc 1eh : desliga o cursor.
```

Através destas funções, o BIOS emula os terminais ADM-3A e Intertec Super-Brain.

4 - Função Puncher

A função envia caracteres através da interface serial, se esta estiver instalada, caso contrário, nada acontece.

5 - Função Reader

A função recebe caracteres da interface serial, se esta estiver instalada. Se não, retorna sempre ctl-Z (1ah).

6 - Função Select Disk

A função Select Disk retorna, agora, em IX, o endereço do DPB do disco selecionado.

7 - Funções Read e Write

Quando houver um erro, estas funções retornam a um valor diferente de zero, que é o código de erro (reconhecido pela nova função **RETURN** Error).

NOVAS FUNÇÕES:

A Jump Table do BIOS foi extendida com as seguintes funções:

```
Função Set Drive
  Função: Especificar um drive físico.
  Recebe:  A - Endereço do drive (de 0 a 3).
  Retorna: IY - Endereço do descritor do drive.

Função Select
  Função: Selecionar um drive físico.
  Recebe: IY - Endereço do descritor do drive.
  Retorna: A - Código de erro.

Função Set Track
  Função: Selecionar uma trilha.
  Recebe: IY - Endereço do descritor do drive.
           C - Trilha.
  Retorna: A - Código de erro.

Função Type I
  Função: Executar um comando do tipo I do controlador de
          disco.
  Recebe: IY - Endereço do descritor do drive.
           A - Comando tipo I.
  Retorna: A - Código de erro.

Função Type II
  Função: Executar um comando do tipo II ou III do
          controlador de disco.
  Recebe: IY - Endereço do descritor do drive.
           A - Comando tipo II ou III.
          Flag Z - Z -> Comando de leitura.
                   NZ -> Comando de escrita.
           B - Número de setores.
           C - Setor inicial.
          HL - Endereço do buffer de leitura/escrita.
  Retorna: A - Código de erro.
```

```
Função RETURN Error
  Função: Decodifica o código de erro retornado pelas
          funções acima.
  Recebe: A - Código de erro.
  Retorna: HL - Endereço da mensagem de erro correspondente
           ao código. A mensagem termina com 00h.

Função Host Error
  Função: Imprime uma mensagem de erro da forma:
          Erro em X: (^C,P,R)
  Recebe: A  - Código de erro.
          HL - Endereço de "retry".
  Retorna: Conforme a resposta:
           C  :Recarrega o sistema;
           P  :Simplesmente retorna;
           R  :Ignora endereço de retorno, indo para o
               endereço de "retry"
          A-F :"loga" no drive especificado e recarrega o
               sistema.
  Modo de usar: Retry: call Read
                       or   a
                       ret  z
                       ld   hl,Retry
                       jp   HostError

Função Print String
  Função: Imprime uma string.
  Recebe: HL - Endereço de uma string terminada por 00h.
```

```
Program Key
Função: Programa o teclado.

Recebe: DE - Endereço da tabela de programação.
        Esta tabela é uma série de pares (tecla, código),
        onde tecla é o código interno gerado pela tecla a
        ser programada e código é o valor que tal tecla
        deve gerar. Existem os seguintes códigos especiais:
        FBh: existe uma string associada à tecla;
        FCh: tecla causa um Warm Start;
        FDh: tecla de Caps Lock;
        FEh: tecla de limpa tela;
        FFh: tecla nula.
        Um valor de tecla 00h indica o fim da tabela.

        BC - Endereço da tabela de strings.
        Tal tabela é uma série de pares (tecla, string),
        onde tecla é uma tecla que tem a ela associada o
        código FBh e a string é uma sequência de bytes,
        onde o primeiro é o tamanho da string e os
        restantes, a string propriamente dita.

Get Key Code
Função: Retorna o código da tecla pressionada.
Retorna: A - Código da tecla pressionada no instante da
         chamada. Se nenhuma tecla foi pressionada, 00h.
```

APÊNDICE 2

MUDANÇAS NO BDOS

1 - Funções 1 e 2: Console input e output

A função de "printer echo", agora, ecoa um ctl-G (bel) quando ativada e não ecoa nada quando desativada.

2 - Função 10: Read Console Buffer

Foram incluídos novos caracteres de controles e retirados outros.

A relação das funções disponíveis são:

CTL-B Se o cursor estiver no início da linha, vai para o final; senão vai para o início.
CTL-C Se o cursor estiver no início da linha, recarrega o sistema.
CTL-D Avança o cursor uma posição para direita.
CTL-G Deleta o caracter sob o cursor.
CTL-H (Back-Space) - Deleta o caracter à esquerda do cursor.
CTL-I Avança para próxima posição de tabulação.
CTL-J (Line-Feed) - Termina a função.
CTL-M (RETURN) - Termina a função.
CTL-R Recupera a última linha digitada.
CTL-S Retrocede o cursor uma posição para esquerda.
CTL-X Deleta todos os caracteres à esquerda do cursor, exclusive.
CTL-Y Deleta todos os caracteres à direita do cursor, inclusive.

3 - Função 13: Reset Disk System

O drive resselecionado agora é o drive em uso, e não mais o drive A.

APÊNDICE 3

CCP

1 - COMANDO TYPE

Durante a execução do comando TYPE, teclando-se um número de 0 a 9, introduz-se um "delay" entre a impressão de duas linhas, sendo que 0 corresponde a nenhum "delay" e 9 ao "delay" máximo.

2 - COMANDOS EXTERNOS:

Agora, quando se executar um comando externo, ou seja, ativar-se um programa (caso o arquivo .COM não exista e se existir o arquivo .SUB), o comando é interpretado como se fosse prefixado pela sequência SUBMIT.

EXEMPLO:

```
Suponhamos o seguinte comando:
A>monta prog

No caso de não existir o arquivo MONTA.COM, o comando é
interpretado da seguinte forma:
A>submit monta prog

Surtiria o mesmo efeito o seguinte comando:
A>monta.sub prog
```

APÊNDICE 4

FORTRAN

Esta breve apresentação do FORTRAN não tem o objetivo de listar todos os comandos da linguagem, pois, esta versão inclui todos os comandos do FORTRAN ANSI - 1966, exceto a manipulação de variáveis complexas.

Fazem parte do disquete de sistema FORTRAN:

```
FORTRAN.COM          ; compilador FORTRAN
FORLIB.REL           ; library FORTRAN
LINK.COM             ; LINK
ED.COM               ; editor de textos
```

O processo de compilar um programa em FORTRAN é dividido em duas etapas:

1 – Compilar o programa, criando um programa objeto de códigos relocáveis.

2 – Linkagem, transformar em objeto executável.

COMPILANDO PROGRAMAS FORTRAN:

Digite:
FORTRAN RETURN

O compilador FORTRAN retornará com "*", indicando que está à espera de um comando. Um comando para o compilador consiste em nomes de arquivos com parâmetros opcionais. Alguns tipos de arquivos são automaticamente reconhecidos pelos compiladores.

Arquivo	Extensão
Programa fonte FORTRAN.	.FOR
Programa fonte COBOL.	.COB
Programa formato MACRO80.	.MAC
Programa em códigos relocáveis.	.REL
Programa listável.	.PRN
Programa executável.	.COM

Formato geral do comando:
D1:objeto,D2:listavel=D3:fonte

Somente o sinal de igual (=) e o nome do programa fonte são necessários para criar o programa objeto relocável com extensão .REL. Para compilar um programa sem que seja criado o objeto relocável ou o listável, inicie a linha de comando com uma vírgula (,). Este recurso é muito utilizado para detectar algum erro de sintaxe.

EXEMPLOS:

```
*=B:TESTE             ;Compila o programa B:TESTE.FOR e
                        cria o objeto relocável B:TESTE.REL.
*TESTEOBJ=TESTE.FOR   ;Compila o programa TESTE.FOR e cria
                        o relocável TESTEOBJ.REL.
*TESTE,TESTE=TESTE    ;Compila TESTE.FOR e cria o relocável
                        TESTE.REL e o listável TESTE.PRN.
*,=TESTE.FOR          ;Compila TESTE.FOR sem criar o
                        relocável e o listável.
*,TTY:=C:TESTE        ;Compila TESTE.FOR listando-o no
                        vídeo sem criar o relocável.
*TESTE0,LST:=B:TESTE  ;Compila B:TESTE.FOR e cria o
                        relocável TESTE0.REL e lista o
                        programa compilado na impressora.
```

PARÂMETROS

Os parâmetros opcionais são letras acrescentadas ao final da especificação do arquivo precedidas por uma barra (/). Mais de um parâmetro poderá ser usado, porém, cada um deve ser precedido por uma barra.

Os parâmetros existentes são:

/O – Listar todos os endereços em octal.
/H – Listar todos os endereços em hexadecimal, é assumido se omitido.
/N – Não listar os códigos gerados pelo compilador.
/R – Força a criação do objeto relocável.
/L – Força a criação do listável.
/P – Cada /P alocará 100 bytes extras no stack durante a compilação.

EXEMPLOS:

```
*,TTY:=PROGRAMA/N   ; Compila PROGRAMA.FOR e lista-o no
                      vídeo sem criar o relocável.
*=TESTE/L           ; Compila TESTE.FOR e cria TESTE.REL
                      e TESTE.PRN.
*=TESTE/P/P         ; Compila TESTE.FOR e cria TESTE.REL
                      alocando 200 bytes extras no stack.
```

NOTA:
Lembre-se que as seis primeiras colunas na codificação do FORTRAN são reservadas e também a partir da coluna 72. Nos comandos READ e WRITE são utilizados números que identificam os dispositivos de entrada e saída:
1,3,4,5 identificam o vídeo.
2 identifica a impressora.
6 a 10 arquivos em discos.

FAZENDO "LINKAGEM"

Com o "LINK" será possível ligar vários programas ou somente um, transformando-os em objetos executáveis.
Digite: **LINK RETURN**

LINK retornará com "*", indicando que está à espera de um comando. Cada comando é formado por uma lista de especificações de arquivos separados por vírgula e às vezes seguidos de parâmetros opcionais. LINK também poderá ser usado em uma só linha, isto é, carregar o programa e executar os comandos na mesma linha.

EXEMPLO:

```
    LINK PROG1,PROG2 RETURN
```

Formato geral do comando:
D1:objeto1,D2:objeto2,D3:objeto3,...Dn:objeton

Os parâmetros existentes são:

/R – Reinicia. Usado para recarregar um arquivo errado se deseja corrigi-lo.

/E – Finaliza LINK, retornando ao DGP/M. Ao final serão mostrados 3 números que indicam: endereço de início, endereço do primeiro byte livre e o número de blocos de 256 bytes usados.

/G – Inicia a execução do programa tão logo a linha tenha sido interpretada.

/N – Grava o programa objeto executável com extensão .COM.

/P e /D – Indica os endereços de origem do próximo programa a ser carregado.
Formato: **/P:endereço** e **/D:endereço**.

```
EXEMPLO:
A>LINK RETURN
*EQ3,EQ3/N/E
Data  0103  2240
[02FB 2240  34]
A>-
```

```
C PROGRAMA FORTRAN EXEMPLO-EQ3.FOR-DGT1000 COM CP/M
C EQUACAO DO TERCEIRO GRAU-AX3+BX2+CX+D=0
  SUBROUTINE RAIZ (C1,RD)
  RD=(1-2*C1)/(3*C1-1)
  DO 5 K=1,20
  Y0=RD
5 RD=(2*C1*Y0*Y0*Y0*+1)/(3*C1*Y0*Y0-1)
  RETURN
  END
C PROGRAMA PRINCIPAL
  INTEGER*1K
  REAL A,B,C,D,A1,B1,C1,D1,RD,Y0,DT,IM,X1,X2,X3
10WRITE(1,20)
20FORMAT('ENTRE COM A B C D OU 0 0 0 0')
  WRITE (1,25)
25FORMAT('#####.####  #####.#### #####.#### #####.####',/)
  READ(1,30) A,B,C,D
30FORMAT (4(F10.4,2X))
  WRITE (1,30) A,B,C,D
  IF(A.EQ.0)GOTO200
  CALCULO DOS PARAMETROS A1,B1,C1
  DT=B*B-3*A*C
  A1=-B/3/A
  B1=(B*C-3*A*D-2*B*B*B/9/A)/DT
  C1=3*A*A*B1*B1/DT
C CALCULO DA RAIZ DICUBICA
  CALL RAIZ(C1,RD)
C CALCULO DA 1A. SOLUCAO
  X1=A1+B1*RD
C CALCULO DAS OUTRAS SOLUCOES
  D1=A*X1+B
  DT=D1*D1+4*A*D/X1
C SE DT<0 ENTAO DUAS RAIZES IMAGINARIAS
  WRITE(1,40)X1
40FORMAT('X1=',F10.4)
  IF(DT)150,100,100
100X2=(-D1+SQRT(DT))/2/A
  X3=(-D1-SQRT(DT))/2/A
  WRITE(1,50)X2,X3
50FORMAT('X2=',F10.4,/,'X3=',F10.4)
  GOTO 10
150IM=SQRT(ABS(DT))/2/A
  RL=-D/2/A
  WRITE(1,60)RL,M
60FORMAT('RAIZES IMAGINARIASX2EX3=',F10.4,'+/-',F10.4,'I')
  GOTO 10
200 STOP FIM
  END
```



MATRIZ: Rua Gávea, 150 · Tel: (031) 332.8300 · Telex: 3352 30.430
Belo Horizonte - MG
RIO DE JANEIRO: Rua Barata Ribeiro, 391 Sl. 404
SÃO PAULO: Rua Faxina, 47 - Centro

ERRATA

1 - O utilitário DDT, que acompanha o sistema operacional, só aceita mneumônicos do microprocessador Z-80 e não mais o **Intel 8080** como descrito na página 34.

2 - Pg. 8
Deve-se desconsiderar os parâmetros dos endereços 106 e 107, já que não são mais úteis devido à implementação do CONFIG. Desconsiderar, também, o exemplo desta página.

NOVOS COMANDOS

Foram incorporadas as seguintes sequências de escape na função **CONSOLE OUTPUT**:

```
ESC"L""0"(1BH,4CH,30H) --> Desliga o avanço de linha
                           automático.
ESC"L""1"(1BH,4CH,31H) --> Liga o avanço de linha automático.
```

Entende-se por avanço de linha automático, a passagem do cursor para a primeira coluna da próxima linha, assim que seja preenchida a última posição da linha. Ao se desativar esse avanço automático, o cursor não se moverá, ao se preencher a última posição.

```
ESC"G""0"(1BH,47H,30H) --> Ativa o mascaramento do Bit 7.

ESC"G""1"(1BH,47H,31H) --> Desativa o mascaramento do Bit 7.
```

Para se permitir a impressão de caracteres gráficos, ou qualquer caracter cujo código seja maior que 127, deve desativar o mascaramento do Bit 7.

OBS:
Toda vez que se executa um Warm-Boot, são ativados o avanço de linha automático e o mascaramento do Bit 7.

