

110.15

MANUAL DO PROGRAMADOR

MANUAL DO PROGRAMADOR

INDICE

1	Apresentação	1
2	Descrição Física	
2.1	Descrição geral	5
2.2	Unid. de Video	6
2.3	Unid. Teclado	6
2.4	Unid. Impressora	9
2.5	Unid.de Disco Flexivel e Adicional	9
2.6	Unid. de Disco Rápido	14
2.7	Unid. de Fita Magnética	14
2.8	Unid. Central	15
2.9	Painel Frontal/Conectores	15
3	Introdução ao Sistema Operacional	
3.1	Alocação do Sistema	17
3.2	Designação de Arquivos	20
3.3	Recursos e Facilidades do Sistema	23
3.4	Comandos Internos	26
3.5	Cuidados	29
3.6	Mensagens de Erro	30
4	Utilitários	
4.1	Preparação de Discos	32
	Reform	32
	Sysgen	34
	Setup	35
4.2	Uso de densidades diferentes	38
	Mode	38

MANUAL DO PROGRAMADOR

4.3	Copia de Discos	40
	Copy	40
	DCopy	43
4.4	Transferências de Arquivos	44
	Pip	44
4.5	Encadeamento	55
	Submit	55
	XSub	57
4.6	Outros	59
	Stat	59
	Dumpas	62
	Despool	63
5	Introdução ao Editor de Textos	
5.1	Descrição Funcional	65
5.2	Descrição Grafica do Editor	68
5.3	Org. do Buffer de Memória	69
5.4	Comandos de movim. do Buffer	71
5.5	Comandos de movim. do Apontador	74
5.6	Comandos de finalização de Textos	75
5.7	Comandos Manipulação de Textos	77
5.8	Comandos de uso geral	78
5.9	Comandos mais complexos	80
5.10	Mensagens de Erro	84

MANUAL DO PROGRAMADOR

Apêndices

- A Resumo do Editor
- B Tabela de Conversão
Binário, Octal, Decimal e Hexadecimal
- C Tabela ASCII
- D Tabela dos Controles Disponíveis
- E Potências de 2
- F Resumo Utilitários
- G Conversão de Arquivos através do SORT

MANUAL DO PROGRAMADOR

Neste manual foram agrupadas todas as informações necessárias sobre o sistema operacional e seus utilitários, como também informações sobre o Hardware do Sistema POLYMAX, acompanhados de resumos e tabelas para otimização do serviço, que juntamente com os manuais da linguagem adotada, o usuário terá em mãos os recursos para desenvolvimento de seus programas contando também com o apoio de uma equipe de suporte qualificada para qualquer esclarecimento posteriores.

Agradecemos as críticas e sugestões para o aprimoramento desta edição, contando com sua colaboração para isto.

DEPTO. TREINAMENTO

1. APRESENTAÇÃO

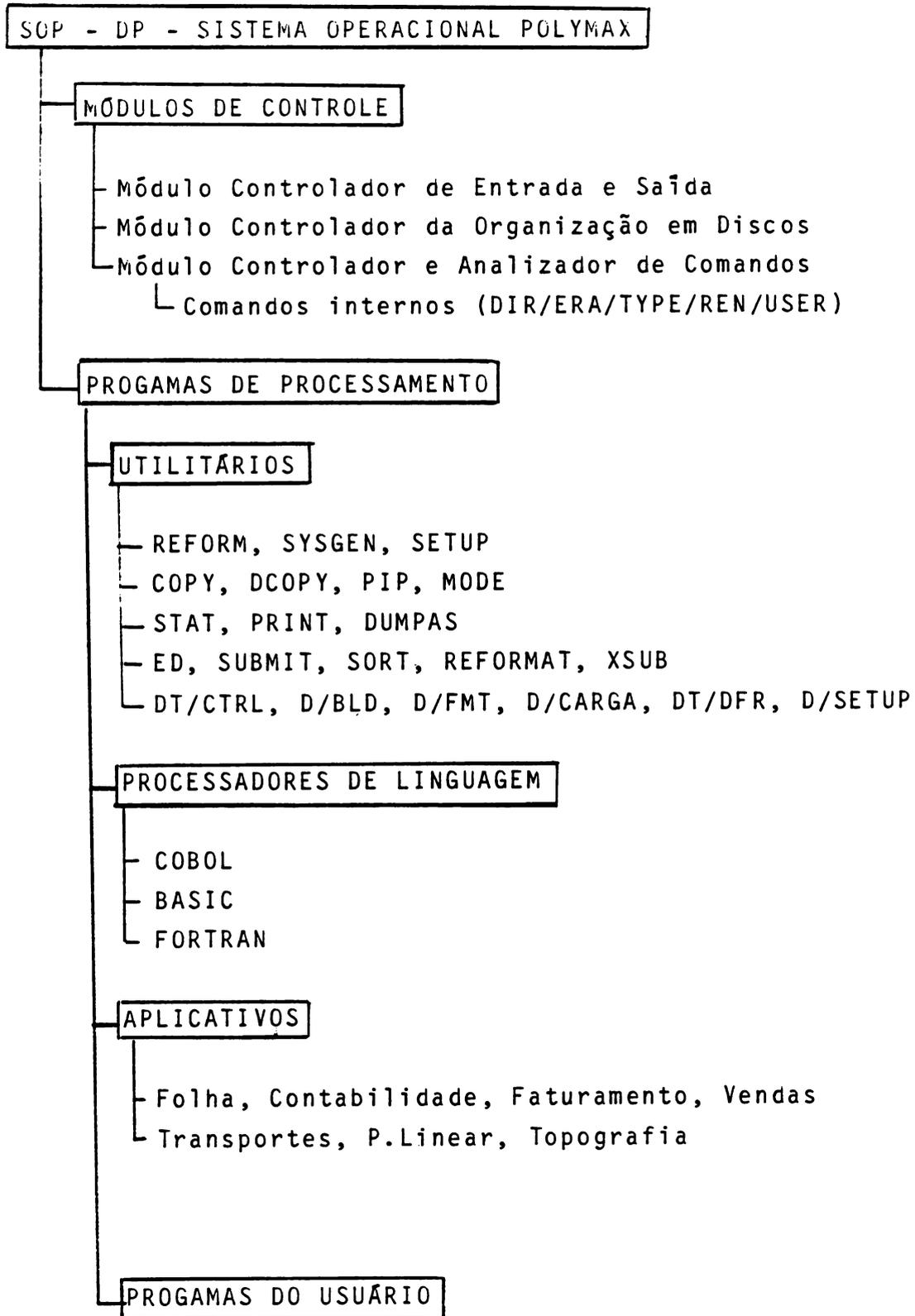
A POLYMAX desenvolveu um microcomputador voltado para o processamento de dados e que atende uma faixa de mercado com um índice de custo e desempenho invejável. E isto se deve em parte ao microprocessador Z80A.

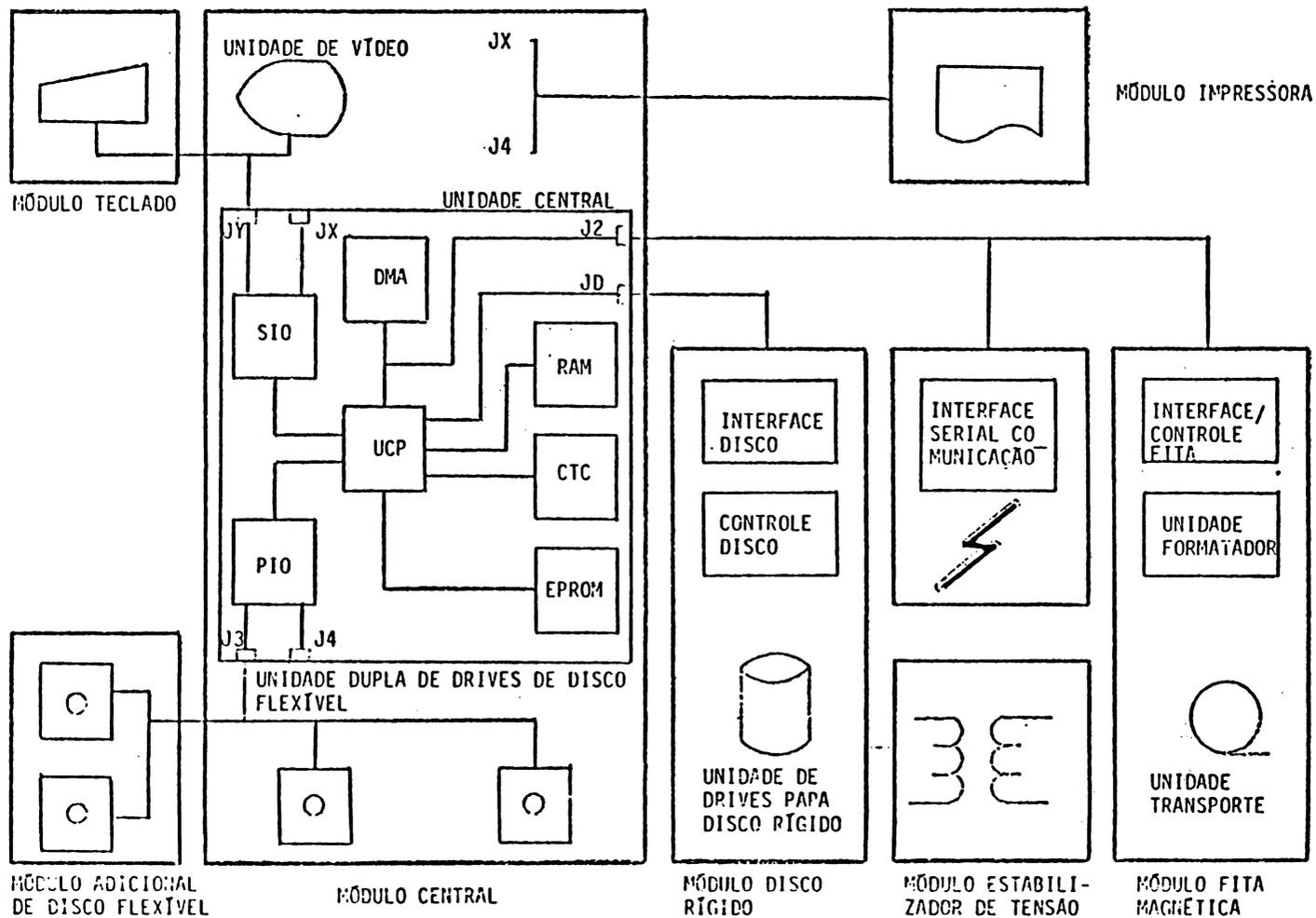
Decorrente das características de microprocessadores, a POLYMAX selecionou para compor seu SOFTWARE, um sistema operacional de recursos poderosos e que encontra compatibilidade quase sem precedente no mercado de microcomputadores. É um sistema operacional voltado para o processamento de dados com características idênticas as encontradas em sistemas de uso geral (sistemas de computadores de pequeno a grande porte), como poderão ser identificadas a seguir.

Por ser modular quanto a sua estrutura de HARDWARE, permite uma gama de configurações expandido seu uso a várias aplicações nas mais diversas áreas.

MANUAL DO PROGRAMADOR

ESTRUTURA LÓGICA DO SOFTWARE POLYMAX





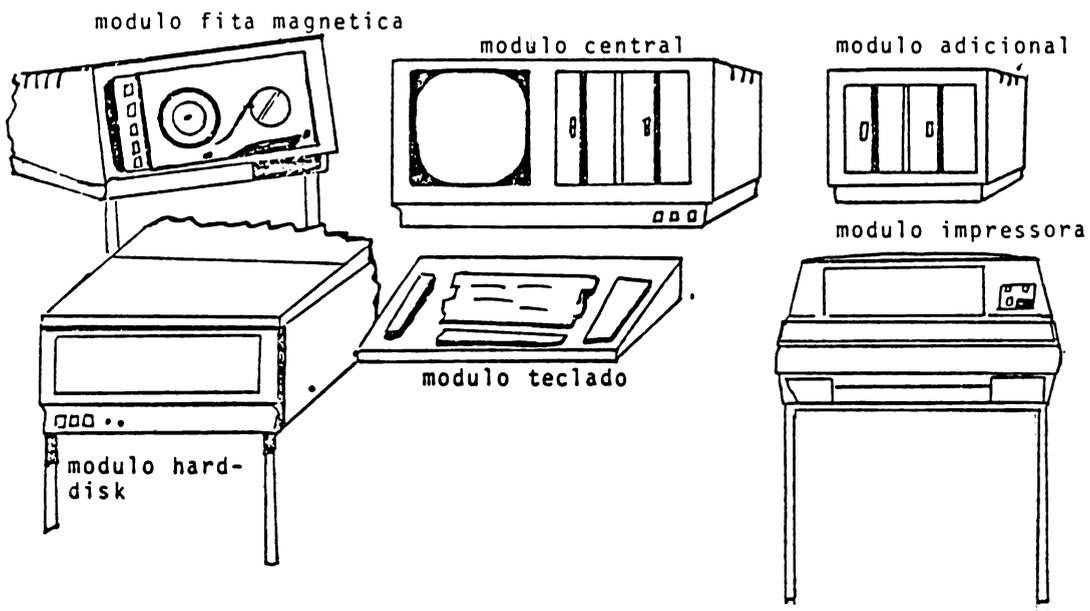
2. DESCRIÇÃO FÍSICA

2.1. Descrição Geral:

O Sistema POLYMAX é constituído fisicamente dos seguintes módulos:

- Módulo Central - unidade vídeo
- unidade central
- unidade disco flexível
- painel frontal/conector

- Módulo Teclado - unidade teclado
- Módulo Impressora - unidade impressora
- Módulo Disco Rígido - unidade disco rígido
- Módulo Fita Magnética - unidade fita magnética
- Módulo Adicional Disco Flexível - unidade disco flexível

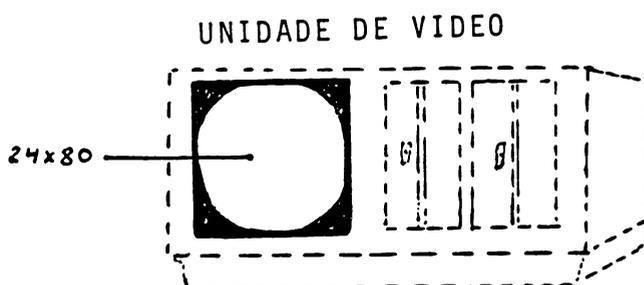


MANUAL DO PROGRAMADOR

2.2. UNIDADE DE VÍDEO

Esta unidade localizada no módulo central, é composta de um cinescópio de 12", com fosfatização em verde, munido de tela anti-refletores.

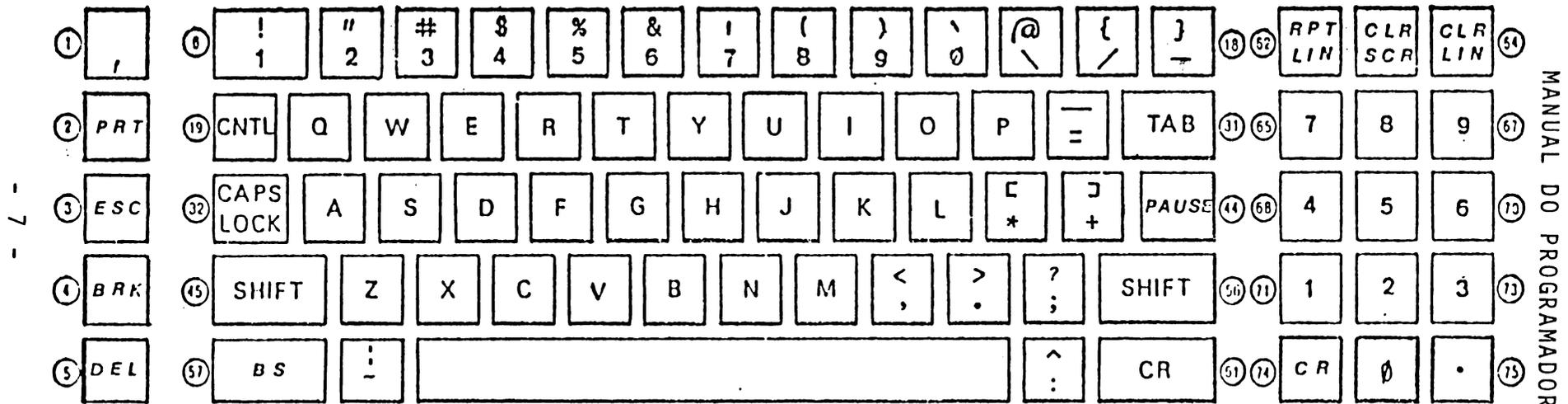
Permite ao usuário o controle de brilho através de potenciômetro externo. Possui tela com 24 linhas de 80 caracteres.



2.3. UNIDADE TECLADO

Esta unidade é semelhante a uma máquina de escrever, contém algumas teclas de funções especiais para auxiliar o usuário na edição de execução dos programas.

TECLADO POLY 201 DP



MANUAL DO PROGRAMADOR

- 7 -

Ⓢ NÚMERO DA TECLA

MANUAL DO PROGRAMADOR

DESCRIÇÃO DO TECLADO E FUNÇÕES

O teclado do Sistema Polymax apresenta uma série de funções já preparadas para facilitar a edição de programa.

Descrição das principais teclas:

No. da tecla	Nome	Função
2	PRT	ativa a impressora, para desativar tecla-a novamente.
3	ESC	Escape usada no BASIC
4	BRK	BREAK - Atualiza diretório
5	DEL	Apaga o ultimo caracter e ecoa no video.
6-18	sinais gráficos	
19	CNTL	CONTROL - usado simultaneamente com outras teclas
20-30	alfabeto	
31	TAB	TABULAÇÃO - 8 espaços
32	CAPS LOCK	Ativa o alfabeto para maiuscula e as teclas 6-18
33-43	alfabeto/sinais	
44	PAUSE	Parada momentanea de uma listagem, qualquer outra tecla desativa
45 e 56	SHIFT	Maiuscula e Minuscula
46-55	alfabeto/sinais	
57	BS	BACK-SPACE - retorna o cursor e apaga o ultimo caracter.
58-60	sinais	
61	CR	CARRIAGE - RETURN Finalização de entradas
62	RPT LIN	reescreve a ultima linha
63	CLR SCR	limpa o video (após um PRT salta página)
64	CLR LIN	apaga a ultima linha
65-73	numeros	
74	CR	
75-76	0 .	zero e ponto

Mais detalhes apêndices C e D.

MANUAL DO PROGRAMADOR

2.4. UNIDADE IMPRESSORA

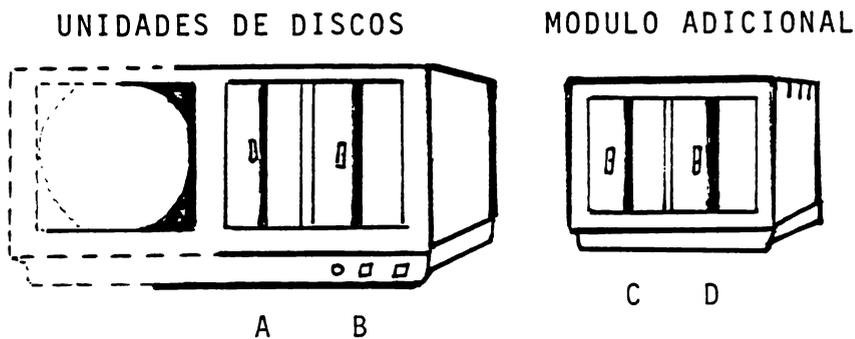
É possível modular o sistema a necessidade do usuário, podemos conectar ao Sistema POLYMAX as seguintes impressoras:

- Matricial 160 CPS
- Linear 300 LPM
- Linear 600 LPM

O modo de operação é explicado em um manual específico da impressora adquirida ou em curso.

2.5. UNIDADE DE DISCO FLEXÍVEL E ADICIONAL

O sistema básico usa 2 unidades de discos flexíveis que podem ser expandidos até 4 unidades, bastando acoplar uma unidade adicional.



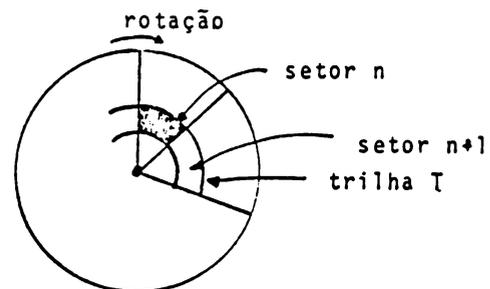
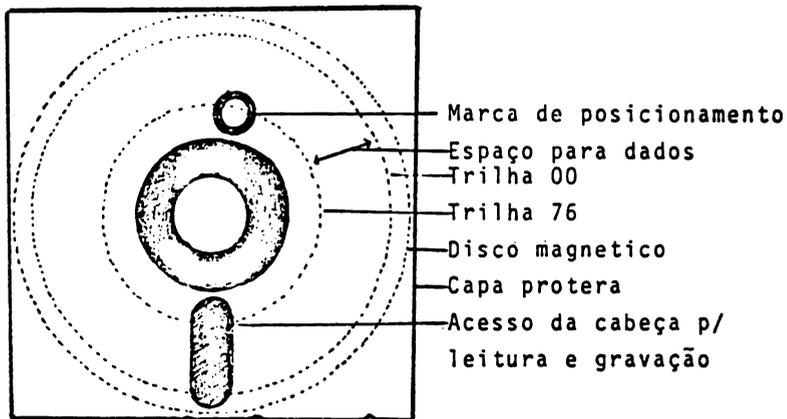
MANUAL DO PROGRAMADOR

Estas unidades tem a função de leitura e gravação de discos flexíveis, são denominadas DRIVES e dentro do sistema operacional serão referenciadas como drive A, drive B, etc...

Os discos flexíveis são confeccionados em material magnético, acondicionados por uma capa protetora.

Os discos flexíveis de 8" estão divididos em 77 trilhas concentricas numeradas de 0 a 76. Essas trilhas são divididas logicamente em setores (Setorização por Software).

DISCO FLEXIVEL (Soft Sector)



MANUAL DO PROGRAMADOR

Podem ser formatados em densidades simples ou dupla.

Especificação em densidade simples:

77 trilhas de 26 setores com 128 bytes por setor
243 k bytes livres

Especificação em densidade dupla:

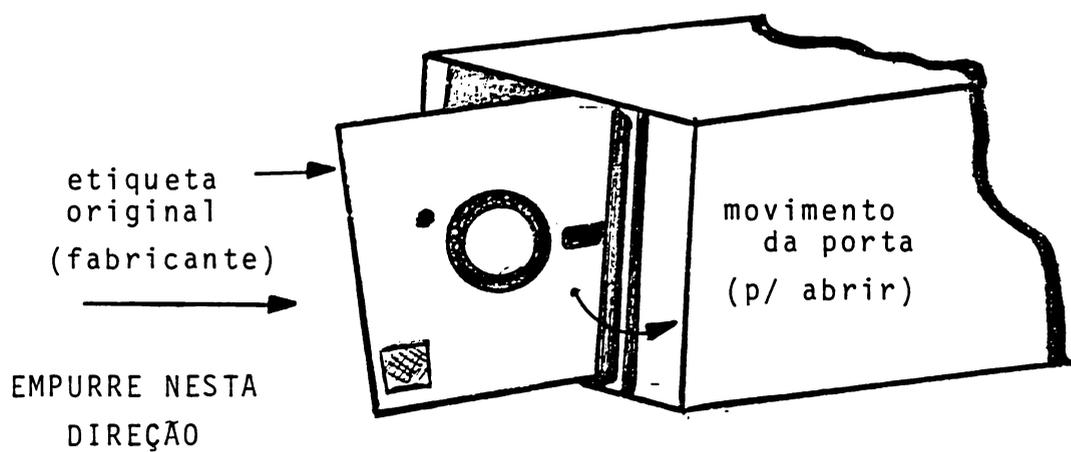
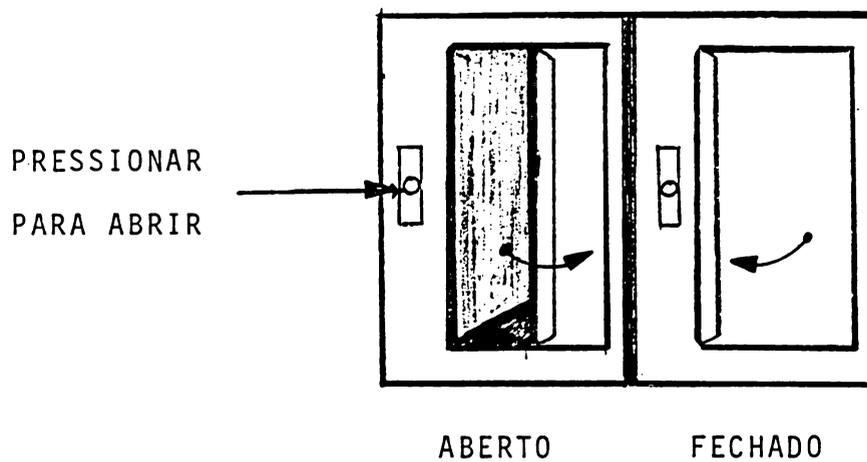
75 trilhas de 48 setores com 128 bytes por setor
2 trilhas de 26 setores com 128 bytes por setor
450 k bytes livres

OBS.: 1- Em ambos casos as trilhas 0 e 1 estarão reservadas para o Sistema Operacional.

2- Embora o setor contenha 128 bytes, a gravação de registro com tamanho variável é permitida; e não será perdido espaço. O sistema operacional trabalha logicamente com registros de tamanho variável.

MANUAL DO PROGRAMADOR

MODO DE OPERAÇÃO



CUIDADOS

Material de constante manuseio e de enorme importancia, pois, nele estará contido todo o nosso trabalho. É importante observarmos com absoluta rigidez alguns cuidados, tais como:

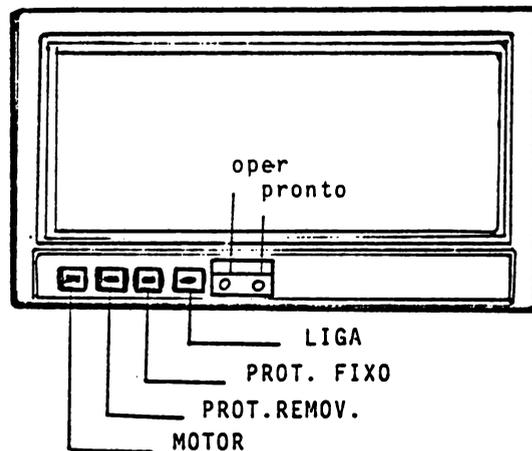
- Guardar os discos na posição vertical (isso evitará sua deformação)
- Escrever na etiqueta antes de cola-la no disco. Caso tenhamos que escrever algo na etiqueta após te-la colado no disco, devemos sempre fazê-lo utilizando caneta hidrografica que, pela sua maciez, não riscará o disco.
- Nunca dobrar ou vergar o disco.
- Nunca deveremos segurar o disco pela sua abertura.
- Nunca colocar grampos ou clips para juntar vários discos.
- Não deverá estar em ambientes que apresentem temperaturas aquém 10 graus centígrados ou além de 52 graus centígrados.
- Não deverá permanecer próximo a um campo magnético.
- Não deverá ficar exposto a luz intensa.
- Uma vez fora do drive, deverá ser imediatamente colocado no envelope de proteção.
- Não desligue a máquina com Diskette nos drives.
- Nunca tire o Diskette durante uma operação de leitura ou gravação.

MANUAL DO PROGRAMADOR

2.6. UNIDADE DE DISCO RIGIDO

Esta unidade é do tipo IBM 5440 ou similar (carregamento superior) com capacidade de armazenamento de 5 MB fixo + 5 MB removível. Consultar o manual de operação e manutenção Disco Rigido para maiores informações.

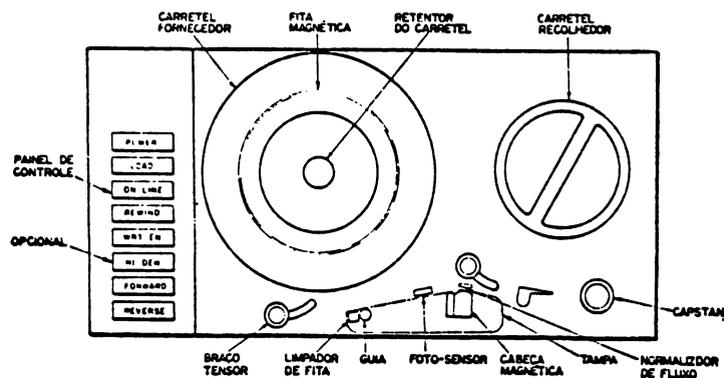
PAINEL FRONTAL



2.7. UNIDADE DE FITA MAGNÉTICA

Esta unidade é utilizada na forma OFF-LINE e tem a função de BACK-UP de arquivos gravados pelo sistema POLYMAX ou outro sistema. Consultar o manual de operação da Fita Magnética para maiores informações.

PAINEL FRONTAL



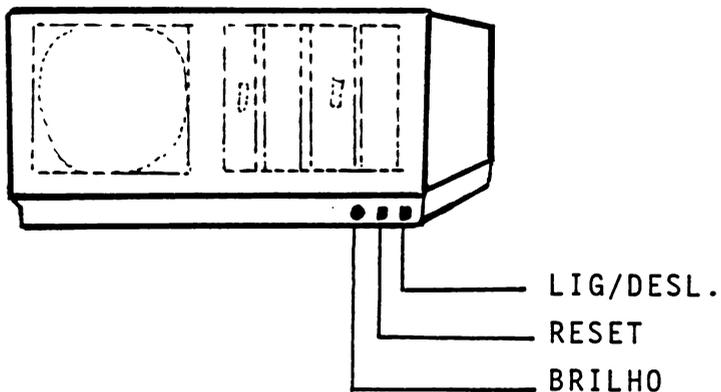
MANUAL DO PROGRAMADOR

2.8. UNIDADE CENTRAL

Esta unidade é embutida no módulo central, contém os componentes para controle de todas as operações do sistema. Consta de um microprocessador Z80A, 64k RAM. Interfaces Paralelas e Serial.

2.9. PAINEL FRONTAL/CONECTORES

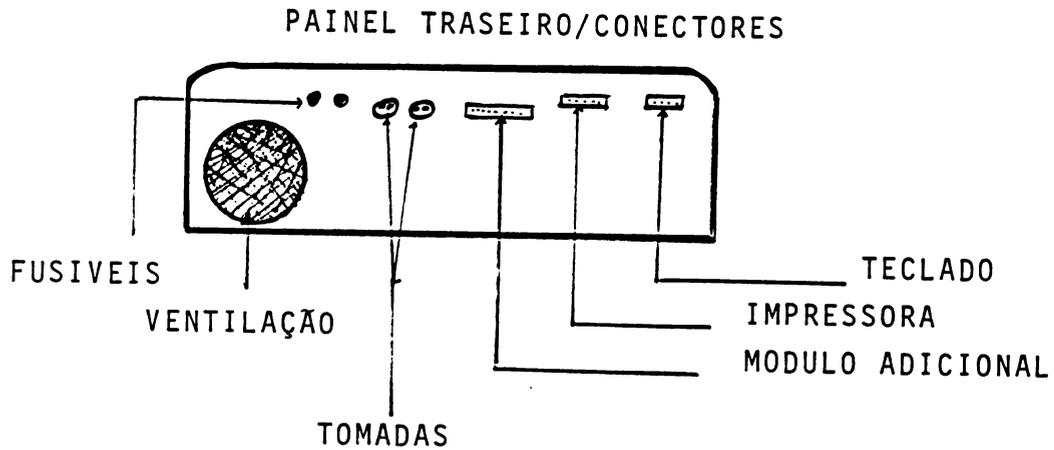
PAINEL FRONTAL



O painel frontal contém tres botões de controle a saber.

- LIG/DESL. para ligar o módulo central
- RESET para carregar e recarregar o sistema operacional
- BRILHO aumenta ou diminui o brilho do vídeo

MANUAL DO PROGRAMADOR



Posicionando na parte posterior do módulo central, é através deste painel que são interligadas as unidades periféricas de interface serial, interface paralela e onde temos a expansão de drives para disco flexível neste mesmo painel é instalada uma tomada dupla com terra para alimentação da impressora ou de algum outro periférico.

3. INTRODUÇÃO AO SISTEMA OPERACIONAL

3.1. ALOCAÇÃO DO SISTEMA

O Sistema Operacional POLYMAX (SOP), é um sistema composto de várias rotinas destinadas ao gerenciamento das operações básicas do sistema, como organização do espaço em Diskette, comunicação do usuário com a máquina e procedimentos de entrada e saída de dados. Podemos apresentar o sistema em tres módulos básicos:

MC - ES Módulo controlador de entrada e saída, que permite operações de acesso aos Diskettes e a comunicação com os periféricos padrões (impressora, vídeo) e possíveis expansões.

MC - OD Módulo controlador de organização em disco, que providencia o gerenciamento de Diskettes, controlando um ou mais drives, executando colocação estratégica no disco, permitindo a construção dinâmica de arquivos e minimizando o movimento do cabeçote durante as operações de leitura e gravação.

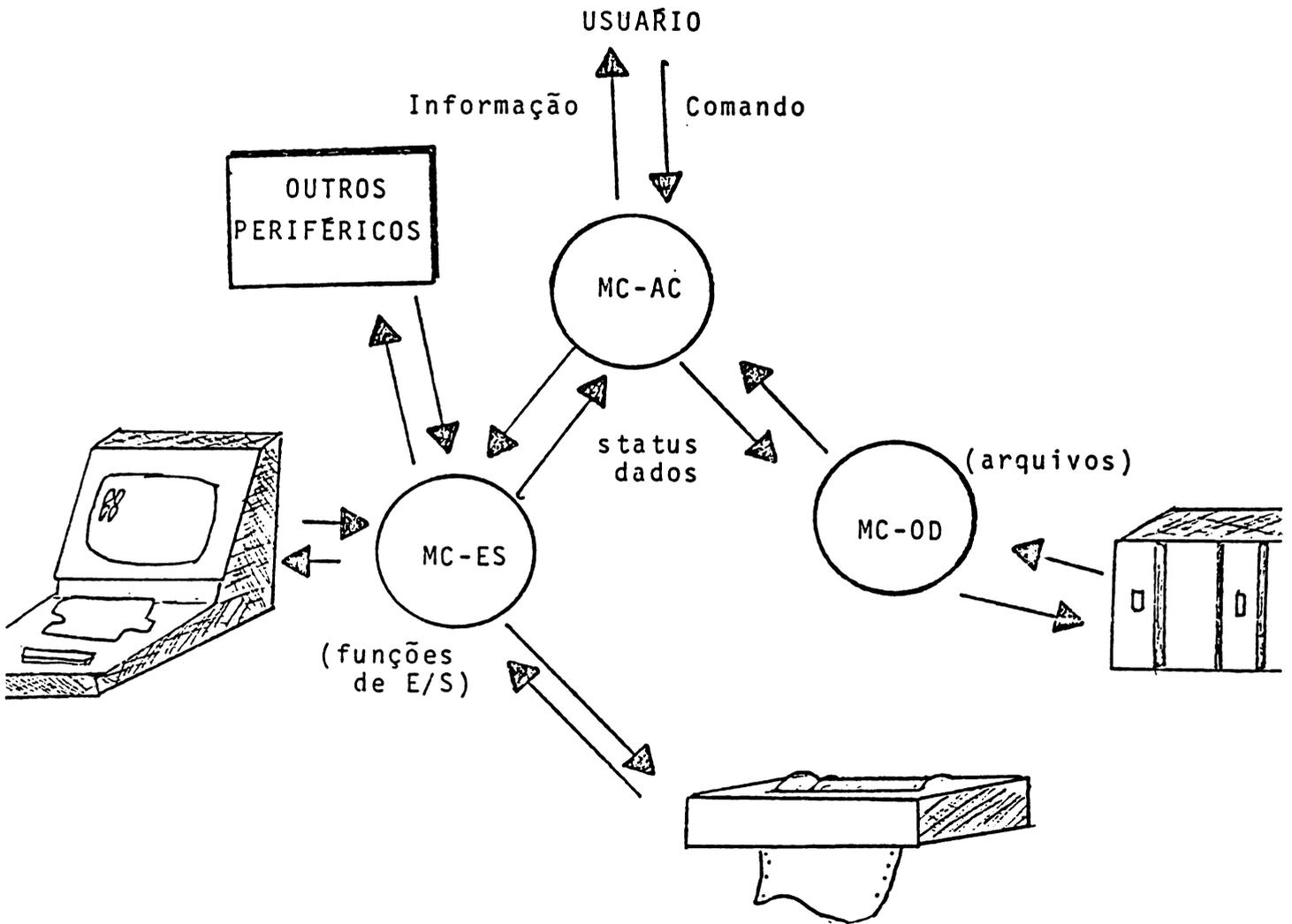
MC - AC Módulo controlador e analisador de comandos, encarregado da comunicação usuário via console e o resto do sistema e internamente com o MC - OD e M - E/S para distribuição e controle de tarefas a serem executadas.

Esses módulos ocupam aproximadamente 8 kbytes, deixando o restante da memória para a ÁREA DO USUÁRIO.

Esta divisão lógica e transparente para o usuário.

MANUAL DO PROGRAMADOR

ESQUEMA GRÁFICO DE COMUNICAÇÃO DO SOP NA MEMÓRIA

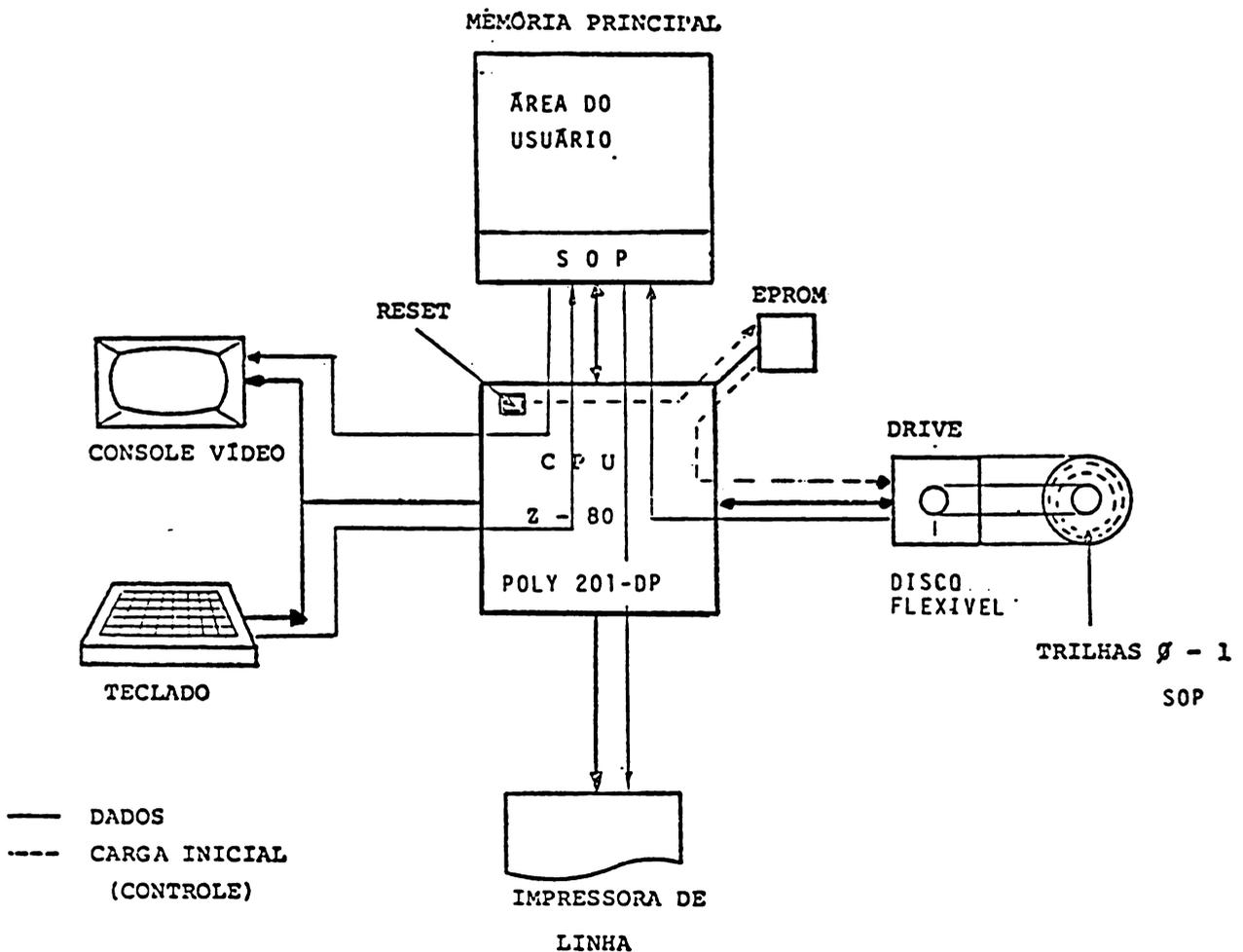


MANUAL DO PROGRAMADOR

O sistema é carregado das trilhas 00 e 01 do Diskette posicionado na unidade de discos de endereço lógico A, e colocado na memória onde fica residente.

A carga do Sistema é efetuada através do botão RESET ou automática quando a CPU é ligada.

ESQUEMA GRÁFICO DE CARGA DO SOP



Uma parte do modulo controlador de entrada e saída é carregado do disco.

3.2. DESIGNAÇÃO DE ARQUIVOS

Qualquer espaço físico ocupado em disco referenciado por um nome é denominado arquivo.

A mesma sintaxe do nome para referenciar um arquivo contendo dados é usada para referenciar programas fontes em qualquer linguagem, programas objetos criados pelo usuário ou utilitários do sistema.

O nome de um arquivo é usado para a gravação, leitura ou execução dos mesmos ou ainda operações para apagar, copiar ou transferir. Podemos referenciar se desejarmos vários arquivos em uma operação que chamaremos de referência MULTIPLA ou um único arquivo que será uma referência ÚNICA.

REFERÊNCIA ÚNICA:

DRIVE: NOME.TIPO

DRIVE: uma letra indicando o endereço lógico do drive onde está o arquivo (A, B, C,...).
Na omissão é assumido o drive corrente.

NOME: Qualquer sequência contendo no máximo 8 caracteres.

TIPO: qualquer sequência de 0 a 3 caracteres indicando a espécie do arquivo.

MANUAL DO PROGRAMADOR

EX.: CLIENTES.001(nome = CLIENTES; tipo = 001)
ETIQUETA (nome = ETIQUETA; não possui tipo)
P001/X.COM (nome = P001/X; tipo = COM)
COPY.COM (nome = COPY; tipo = COM)
B:ABC3.4 (drive = B; nome = ABC3; tipo = 4)

Para os arquivos de dados criados pelo usuário não há obrigatoriedade de um tipo, o nome é de livre escolha dentro dos padrões apresentados.

Alguns tipos especiais de arquivos usados no SOP:

.COM =arquivo contendo programa objeto, para ser carregado na memória.
.FOR =arquivo com programa fonte em FORTRAN
.BAS =arquivo com programa fonte em BASIC
.COB =arquivo com programa fonte em COBOL
.REL =arquivo com código relocável (gerado por compilação)
.INT =arquivo em código intermediário
.PRN =arquivo com listagem de compilação
.LST =arquivo com listagem de compilação
.BAK =arquivo Back-up gerado pelo editor do texto
.LIB =arquivo biblioteca, pode ser acessado através do editor
.\$\$\$ =arquivo temporário, usado somente pelo sistema, deve ser apagado quando aparecer no diretório de um Diskette.
.SUB =arquivo usado pelo SUBMIT

MANUAL DO PROGRAMADOR

Os seguintes caracteres não podem ser incluídos como parte do nome ou tipo de uma referência única:

* . , ; : = ? [] < >

Para uma referência múltipla, faremos uso dos caracteres "?" e "*" que indicaram:

? : quando colocado no lugar de um caracter, indica que aquela posição pode ser substituída por qualquer caracter.

* : quando colocado no lugar do nome e ou do tipo, indica que qualquer nome e ou tipo pode ser referenciado.

EX.: temos um diskette com os seguintes arquivos:

1 - CLIENTES.DAT	4 - EST.COM
2 - ESTUQ1.DAT	5 - EST.COB
3 - ESTUQ2.DAT	6 - ES.COM

Referência Múltipla Arquivos Referenciados

ESTUQ?.DAT	2, 3
EST????????	2, 3, 4, 5
EST????.*	2, 3, 4, 5
?????????.DAT	1, 2, 3
*.DAT	1, 2, 3
?????????.???	1, 2, 3, 4, 5, 6
.	1, 2, 3, 4, 5, 6
E?.DAT	nenhum

MANUAL DO PROGRAMADOR

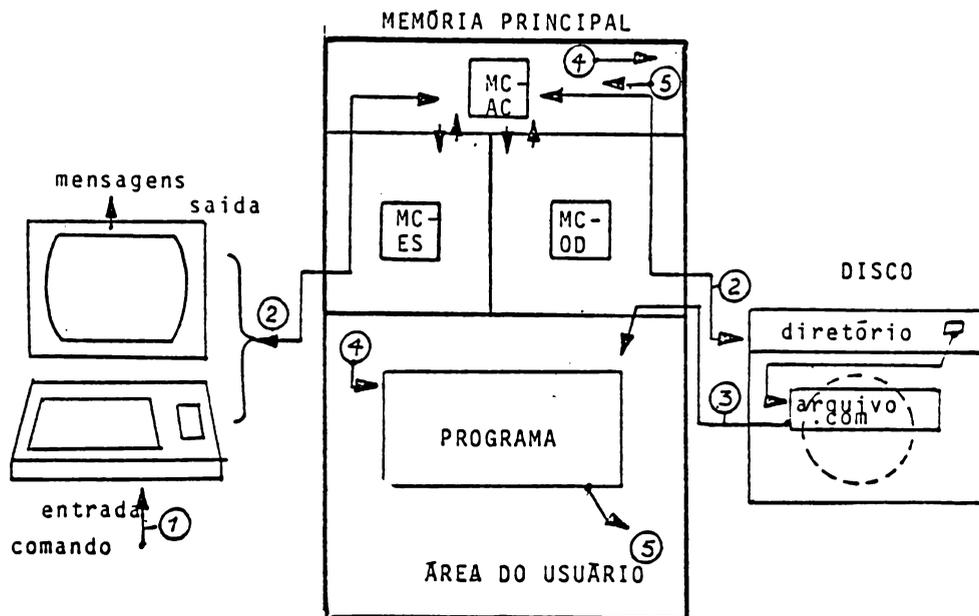
3.3. RECURSOS E FACILIDADES DO SISTEMA

1- Execução de um programa (comando)

O MC-AC verifica se o nome introduzido pelo console vídeo/teclado é um comando interno do sistema, caso contrário ordena ao MC-OD que procure o nome introduzido na unidade de disco especificada com o tipo COM.

Se o nome existir, será carregado na área do usuário, então o MC-AC passa o controle de processamento para o programa, ao término da execução o controle é devolvido ao MC-AC.

FLUXO BÁSICO DE EXECUÇÃO



MANUAL DO PROGRAMADOR

Alguns programas necessitam de parâmetros para a execução e exigem que esses parâmetros sejam passados na carga.

EX. B:PRINT A:CLIENTES.DAT - deve existir no drive B o arquivo PRINT.COM, que será carregado para a área do usuário, assumindo o controle de execução, logo em seguida irá ler o parâmetro introduzido no console (A:CLIENTES.DAT)

2- MUDANÇA NO DRIVE CORRENTE

Quando o sistema é carregado na memória o drive que é assumido como corrente e o drive com endereço lógico A.

Para efetuar a mudança de drive basta introduzir a letra que corresponde ao endereço lógico do drive desejado, seguido de dois pontos.

EX. A>drive corrente

A>B: mudança

B> drive corrente

Qualquer operação dentro do sistema em que não esteja especificado uma unidade de disco, será assumido o drive corrente.

3- DELIMITADORES DE REGISTROS E FIM DE ARQUIVO

O sistema operacional usa para separar registros dois bytes delimitadores ODH, OAH CARRIAGE-RETURN, LINE-FEED e para fim de arquivos textos (ASCII), complementa o último setor de 128 bytes com o byte 1AH (CONTROL-Z)

4- CONTROLES DOS ARQUIVOS

O sistema reserva para cada arquivo um espaço em disco para controle, que são chamados de PUNTOS DE ENTRADA, esses blocos formam o diretório do diskette, sendo alocados na trilha 02 de cada unidade de disco.

Podemos dispor para:

- diskettes em densidade simples (241k) de 64 entradas
- diskettes em densidade dupla (448k) de 96 entradas
- disco rígido (4,8MB) de 1024 de entradas.

O sistema operacional permite a gravação de arquivos com até 8MB. Registros podem ser considerados logicamente contínuos na área de dados de discos. Os arquivos são divididos em blocos de 16k, chamados extensões lógicas. Não existe nenhuma consequencia para o usuário com estas divisões pois elas são acessadas tanto sequencialmente como randomicamente. Não podem existir dois arquivos com referencias iguais num mesmo disco.

3.4. COMANDOS INTERNOS

O sistema operacional POLYMAX contém internamente 5 comandos disponíveis para o usuário. Esses comandos ficam residentes na memória dentro do módulo controlador e analizador de comandos.

1- DIR

Lista no vídeo os arquivos contidos no diskette, podem ser feitas referencias multiplas.

EX. DIR - lista o diretório do diskette no drive corrente

DIR B - lista o diretório do diskette no drive B

DIR *.COM - lista o diretório do diskette no drive corrente, somente dos arquivos com o tipo COM.

DIR COPY.COM - lista o nome do arquivo COPY.COM do drive corrente.

Quando não é encontrado os arquivos que foram referidos a mensagem "NO FILE" é apresentada no vídeo.

2- ERA

Apaga no disco os arquivos que foram referenciados.

EX. ERA CLIENTES.DAT - será apagado o arquivo referenciado com o nome CLIENTES e tipo DAT do disco no drive corrente.

ERA B:*.COM - serão apagados todos os arquivos com tipo COM do diskette no drive B.

MANUAL DO PROGRAMADOR

ERA *.* - serão apagados todos os arquivos do drive corrente, porém antes será enviada a mensagem: "ALL FILES (Y/N)?". Somente se for introduzida a tecla "Y" é que a operação será executada.

3- REN

Modifica o nome do arquivo no disco. Somente é possível modificar um nome de arquivo por vez.

Forma Geral = REN NOME-NOVO=NOME-VELHO

EX. REN ARQ2.COM=ARQ1.COM - o arquivo como o nome de ARQ1 e tipo COM será renomeado com o nome ARQ2 e tipo COM no diskette do drive corrente.

REN B:X.DAT=F.DAT - o arquivo com a referência F.DAT do drive B será renomeado para X.DAT.

REN.B:X.DAT = B:F.DAT - idem ao exemplo acima.

Se o arquivo que se quer renomear não existir, será enviada a mensagem "NOT FOUND"

Se o novo nome já existir no diretório será emitida a mensagem "FILE EXISTS".

4- TYPE

Lista o conteúdo de um arquivo. Este comando não deve ser executado em arquivos que não estiverem no código ASCII. Só poderá ser listado um arquivo por vez.

MANUAL DO PROGRAMADOR

EX. TYPE CLIENTES.DAT = listará o conteúdo do arquivo CLIENTES.DAT no formato em que foi gravado, sem qualquer característica de relatório (não faz saltar de página).

5- USER

Aloca uma nova área lógica no mesmo diretório do disco. Forma Geral:

USER n

onde n é um valor de 0 a 15.

Sendo assim podemos ter arquivos gravados em diferentes áreas lógicas em um mesmo disco e sendo acessadas somente quando estivermos na mesma área. Na carga do Sistema é assumido como USER 0.

Somente o comando PIP com o parametro GN consegue fazer a transferência de uma área para outra.
(ver capítulo 4 - utilitários)

3.5. CUIDADOS

Por ser um sistema de microcomputador, o usuário terá que tomar o maior cuidado em alguns aspectos como:

- troca de discos deve ser sempre acompanhada de um CTRL-C ou BREAK para posicionar o novo diretório, ou se interna em programas, usar as rotinas adequadas. Quando um disco é trocado seu diretório ainda permanece na memória e o novo disco está disponível apenas para leitura (R/O).
- O sistema operacional não tem proteção quanto a queda de energia ou qualquer interrupção similar, que limpe a memória (reset, por exemplo). É necessário que o usuário faça proteção em seus programas com os recursos disponíveis na linguagem usada para não perder um volume grande de dados.

É também de importância que o usuário faça o controle de capacidade do disco por programa.

O tamanho dos programas deve ser compatível com a memória disponível.

MANUAL DO PROGRAMADOR

3.6. MENSAGENS DE ERRO

O Sistema Operacional deteta três tipos de erro através do Módulo Controlador de Organização em Disco. Quando ocorre um erro é enviada a mensagem:

BDOS ERR ON D: CONDIÇÃO.

onde: D: é o drive que ocorreu o erro.

CONDIÇÃO pode ser:

BAD SECTOR

SELECT

READ ONLY

BAD SECTOR - Foi detetado uma condição de erro na leitura ou gravação do disco; isso pode indicar que o disco está danificado, mau colocado, não está presente ou o controlador está danificado.

Verifique a causa e reinicie a operação com RESET.

SELECT - Ocorre quando um endereçamento é feito para uma unidade de drive não existente (diferente de A, B, C ou D)

Qualquer tecla pressionada, fará o Sistema retornar.

READ ONLY - Ocorre quando tentamos gravar um disco que não contem mais espaços livres ou o diretório não está reconhecido pelo Sistema (o disco foi trocado e não foi teclado um CTRL-C para atualizar o diretório)

Tambem quando o disco ou o arquivo está na condição R/O dada pelo comando STAT.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO - Quando uma trilha defeituosa é encontrada, o Sistema envia a mensagem:

```
PERMANENT    DISK    ERROR,TRACK    tt,SECTOR  
ss,STATUS xx
```

onde: tt numero da trilha
 ss numero do setor
 xx condição de erro

exemplo: STATUS descrição
 10 RECORD NOT FOUND
 08 CRC ERROR
 80 DISK NOT READY

Um erro de trilha pode acontecer, se a trilha estiver fisicamente danificada ou se estiver mau gravada (flutuação magnética)

No último caso reformatar o disco resolverá o problema.

4- UTILITÁRIOS

4.1- PREPARAÇÃO DE DISCOS

INTRODUÇÃO:

Qualque diskette para ser utilizado no SOP-DP deverá passar por um processo de preparação envolvendo os seguintes utilitários.

UTILITÁRIO 1:

REFORM

FUNÇÃO:

É usado para formatar logicamente no padrão SOP-DP, os diskettes virgens em densidade simples ou dupla.

Algumas vezes é necessário formatar um diskette, caso sua formatação tenha sido destruída, com o uso de uma função errada em um programa, ou uma má alocação das trilhas.

Notar que todos os dados gravados no diskette, serão destruídos sem uma verificação prévia, portanto tenha certeza quanto ao diskette que será formatado para não destruir arquivos importantes.

MANUAL DO PROGRAMADOR

EXECUÇÃO:

REFORM <CR>

Após a carga o programa apresentará as funções disponíveis; selecione pelo número a função desejada, a seguir o programa irá requisitar o diskette a ser formatado, quando estiver com o diskette colocado tecle CR e o processo de formatação começará.

Após o término será emitida uma mensagem indicando o fim do processo e requisitando nova função.

OBSERVAÇÃO:

Em caso de erro o programa indicará uma mensagem adequada.

MANUAL DO PROGRAMADOR

UTILITÁRIO 2:

SYSGEN

FUNÇÃO:

É usado para gerar o Sistema operacional no diskette; faz uma cópia do conteúdo das trilhas 0 e 1 do diskette do drive indicado como fonte, para as trilhas 0 e 1 do diskette no drive destino.

Deve ser executado logo após o programa REFORM.

Esse programa não interfere na área de dados, portanto se o disco contiver arquivo, estes não serão afetados.

EXECUÇÃO:

SYSGEN <CR>

Após a carga o programa pedirá o nome do drive fonte, indique com a letra correspondente (A, B, C, D) em seguida tecle CR; nesse instante é carregado o conteúdo das trilhas 0 e 1 do drive indicado para memória, e é requisitado o nome do drive destino, tecle a letra correspondente (A, B, C, D), o programa faz a cópia do conteúdo anteriormente carregado na memória para as trilhas 0 e 1 deste diskette; para terminar tecle CR.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO:

- Em caso de erro aperte RESET e recomece.
- Normalmente usamos:
drive fonte - A
drive destino - B
- As trilhas 0 e 1 estão sempre em densidade simples, independente da densidade do disco.

UTILITARIO 3

SETUP

Função:

Atualizar os parametros do sistema operacional no diskette.
Após a execução do programa SYSGEN, algumas vezes é necessário atualizar a velocidade da impressora ou a nova densidade do disco.

Execução:

SETUP < CR >

Após a carga, o programa emite uma série de mensagens requisitando:

- A taxa de transmissão da console (0-7) que deverá ser sempre 7

MANUAL DO PROGRAMADOR

-A taxa da transmissão da impressora (0-7) que poderá variar de acordo com a impressora usada:

160 CPS	- 4
340 CPS	- 4
300/600 LPM	- 7
300 LPM-Paralela	- 0

-A taxa de transmissão da porta auxiliar (0-7) que pode ser omitida (tecle CR).

-A densidade dos diskettes, 0 para simples, 2 para dupla, selecione de acordo com a densidade dada ao disco no programa REFURM.

-Tipo do Leitor, 1 para face simples, 2 para face dupla, selecione de acordo com o seu equipamento.

-O drive que será feita a alteração: A ou B.

Após o término tecle CR com o diskette do sistema no drive A. Para carregar o novo sistema aperte RESET com o diskette atualizado no drive A.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO:

Quando aparecer a mensagem:

ALTERAÇÃO SERÁ FEITA NO DRIVE E?
TECLE Y OU N

Tecle sempre "N".

Conclusão:

Após a execução dos utilitários descritos acima é necessário que o diskette recém-preparado contenha o programa CBIOS64.COM, que é uma extensão do módulo controlador de entrada e saída (esta extensão é residente também na área de dados para permitir futuras implementações no SOP).

Para a gravação do programa CBIOS64.COM no novo diskette, use o utilitário PIP, com o seguinte comando:

PIP B:=A:CBIOS64.COM[V] (ver transferências de arquivos).

MANUAL DO PROGRAMADOR

4.2- USO DE DENSIDADES DIFERENTES

Introdução:

Após a carga do sistema, todos os drives estão com a mesma densidade do diskette posicionado no drive A; é possível alterar a densidade de qualquer drive permitindo a operação de dois ou mais drives em densidades diferentes.

UTILITARIO 4:

MODE

Função:

É usado para alterar a frequência de operação de um drive para simples ou dupla densidade.

Execução:

MODE d: n <CR>

Após a carga do programa na memória os parâmetros d e n serão analisados, sendo que:

d é o drive desejado deve ser A,B,C ou D.

n é o número do modo da densidade:

0 para densidade simples

2 para densidade dupla.

Em caso de erro o programa enviará uma mensagem adequada, caso contrário confirma a execução com a mensagem.

Drive A está no modo n.

MANUAL DO PROGRAMADOR

Conclusão:

Se tentarmos acessar um drive e ocorrer erro na trilha 02 devemos imediatamente verificar se o disco alocado nesse drive não está em densidade diferente. Para isso usaremos o utilitário MODE.

MANUAL DO PROGRAMADOR

4.3 - CÓPIAS DE DISCOS

Introdução: Uma boa medida de segurança no sistema é o hábito de manter cópias bem atualizadas de todos os arquivos e programas, para evitar a perda de informações.

UTILITARIO 5: COPY

Função: Serve para copiar o conteúdo das trilhas de um diskette em outro. Permite a cópia de todas as trilhas, das trilhas do sistema ou das trilhas de dados.

MANUAL DO PROGRAMADOR

* Execução:

COPY tipo <CR>

Após a carga do sistema na memória será verificado se a sintaxe do tipo está correta.

O tipo pode ser:

ALL para copiar todas as trilhas (00 a 76)

SYSTEM para copiar somente o sistema (00 e 01)

DATA para copiar somente os dados (02 a 76)

Se o tipo estiver errado ou for omitido será enviada a mensagem apropriada.

Caso contrário enviará a mensagem:

FONTE NO A

COPIA NO B

Tecla CR

Neste momento deve-se colocar o disco fonte no drive A e o disco em que será feita a cópia no drive B.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO:

O programa não irá verificar o conteúdo do disco do drive B, por isso não há necessidade deste estar vazio, seu conteúdo anterior será apagado automaticamente.

-Em caso de erro repita a operação mesmo que aparentemente o conteúdo esteja correto.

-Os discos devem estar em densidade simples.

MANUAL DO PROGRAMADOR

UTILITARIO 6:

DCOPY

Função:

Idêntica ao utilitário COPY porem altera as densidades dos drives para dupla antes de começar a cópia.

Execução:

DCOPY tipo <CR>

O programa tem execução analoga ao programa COPY.

Conclusão:

Para copiar em densidade simples usar o programa COPY, para copias em dupla usar o programa DCOPY. Em caso de densidades diferentes ver transferencias de arquivos. (utilitario PIP).

MANUAL DO PROGRAMADOR

4.4- TRANSFERENCIAS DE ARQUIVOS

INTRODUÇÃO:

O Sistema operacional permite a transferencia de arquivos entre os periféricos facilitando, assim a organização dos discos, listagens rápidas, ou cópias em densidades diferentes entre outras coisas.

UTILITARIO 7:

PIP

Função:

Transferencias de arquivos entre os periféricos (video, impressora e drive).

Execução:

1 - PIP linha de comando <CR>
ou
2 - PIP <CR>

No primeiro formato o Programa PIP será carregado para a memória irá analisar a linha de comando, se correta, após a sua execução retornará o controle para o sistema.

No segundo formato o programa após a carga, emitirá no video um asterisco (*), e ficará aguardando uma linha de comando, executando retorna novamente o asterisco (*) para nova linha, até que seja tecla somente um CR.

MANUAL DO PROGRAMADOR

Ex. PIP <CR>

*Linha de comando <CR>

*Linha de comando <CR>

* <CR>

D>

MANUAL DO PROGRAMADOR

FORMATO DAS LINHAS DE COMANDO

Uma linha de comando não pode exceder 255 caracteres e tem como forma geral:

Destino = origem, origem,...

É o nome do arquivo ou periférico que receberá os dados, não pode conter referencia multipla.

São os arquivos que serão transferidos.

OBSERVAÇÃO:

Quando transferimos mais de um arquivo para um unico arquivo, as cópias são feitas a partir da esquerda para a direita e os arquivos serão assumidos como ASCII e tendo como fim de arquivo CTRL-Z (1AH).

-Arquivos com o tipo .COM serão assumidos como não ASCII.

-Um arquivo pode ser copiado no mesmo diskette, com referencia diferente.

MANUAL DO PROGRAMADOR

Quando transferimos varios arquivos de um diskette para outro, o cestino só possuirá a referencia do drive em que está o diskette e a origem será uma referencia multipla.

-O arquivo original nunca é alterado ou perdido.

MANUAL DO PROGRAMADOR

PIP A1.COB=A2.COB

Faz a cópia do arquivo A2.COB no drive corrente (A indicação de drive foi omitida), no mesmo drive (a indicação do drive destino também foi omitida) com o nome de A1.COB.

PIP B:A1.COB=A:A2.COB

Faz a cópia do arquivo A2.COB no drive A para o drive B com o nome de A1.COB.

PIP B:A1=B:A2,A:A3,A4

Faz a cópia do arquivo A2 do drive B, a seguir do arquivo A3 do drive A e do arquivo A4 do drive corrente para o drive B com o nome A1.

PIP B:=A:*.COB

Faz a cópia de todos os arquivos que tenham o tipo .COB do drive A para o drive B com os mesmos nomes.

A ordem de transferência será a ordem de gravação do disco origem.

PIP A.COB=B

Equivale a PIP A.COB=B:A.COB

MANUAL DO PROGRAMADOR

OBSERVAÇÃO:

Em operação normal o programa PIP não reescreve um arquivo que esteja na condição R/U, sem antes enviar a Mensagem DESTINO ESTÁ R/U APAGA (Y/N)?

Se não for teclado Y, a operação não será completada e será enviada a mensagem ****NOT DELETED****

Se o arquivo fonte estiver na condição SYS não será efetuada a operação.

MANUAL DO PROGRAMADOR

ASSINALAMENTOS PARA OUTROS PERIFÉRICOS

Usaremos as seguintes referencias:

Para o vídeo

TTY:

Para a impressora

LST: ou PRN:

PIP TTY:=A1.COB

Testa o conteúdo do arquivo A1.COB do drive corrente no vídeo.

PIP LST:=B:A1.COB

Lista o conteúdo do arquivo A1.COB do drive B na impressora.

PIP PRN:=B:A1.COB

Lista o conteúdo do arquivo A1.COB do drive B na impressora, numerando as linhas e saltando página a cada 60 linhas.

MANUAL DO PROGRAMADOR

PARAMETROS PARA TRANSFERÊNCIA

O programa PIP permite parametros para alteração dos dados durante a transferência.

Somente o arquivo destino será alterado pelos parametros.

Para colocar os parametros teremos o formato:

```
PIP destino = origem [ Parametros ] , origem  
[Parametros]...
```

Os Parametros são:

- Dn: apaga os caracteres que ultrapassarem a coluna n de cada linha do arquivo origem.
- F: Filtra todos os comandos de salto de página.
O Parametro P pode ser usado simultaneamente para colocar novos saltos.
- Gn: Permite que sejam transferidos arquivos de outras áreas de usuários (comando USER).
- L: Transforma caracteres alfabéticos maiusculos em minusculos.
- N: Numera as linhas do destino, zeros a esquerda são suprimidos e o número é seguido de :,
Se N2 é especificado, os zeros são incluídos.

MANUAL DO PROGRAMADOR

- O: Transferências de arquivos objetos, não interpreta CONTROL-Z (1AH) como final de arquivo; arquivos tipo .REL ou .INT.
- Pn: Salta página inicialmente, e a cada n linhas salta página.
Para n=1 será assumido 60 linhas.
- Qtexto↑Z: Termina a transferência quando o texto terminado por um CTRL-Z é encontrado.
- Stexto↑Z: Inicia a transferência quando o texto terminado por um CTRL-Z é encontrado.
- R: Lê Os arquivos na condição SYS.
As condições SYS e R/O são transferidas quando existirem.
- Tn: Expande o caracter TAB (09H) para n brancos.
- U: Transforma os caracteres alfabéticos minúsculos em maiúsculos.
- V: Verifica se o dado foi bem copiado, relendo depois da gravação (o destino deve ser um arquivo em disco.)
- W: Escreve sobre arquivos na condição R/O, sem interrogar na console.

MANUAL DO PROGRAMADOR

OBSERVAÇÕES:

Para transferir arquivo de uma área para outra é necessário que o comando PIP esteja residente nesta área.

Para obtermos isso, executamos a seguinte sequência:

```
-A>USER n1 <CR>
-A>PIP <CR>
* <CR>
A>USER n2 <CR>
A> SAVE 32 PIP.COM <CR>
A> PIP A:=A:*.COM [Gn1]<CR>
```

Na sequência acima estamos transferindo todos os arquivos tipo .COM da área n1 para a área n2 do drive drive A.

Para transferirmos somente uma parte de um arquivo, podemos usar os parametros S e Q concatenados. Se o texto definido em S ou Q estiver em minuscuro no arquivo original, deverá ser usado o segundo formato do comando PIP. O primeiro formato transforma o texto dado em maiusculo automaticamente antes de pesquisar.

MANUAL DO PROGRAMADOR

- PIP A:A1.COB=B:A2.COB[D72] Faz a cópia do arquivo A2.COB no drive B para o drive A com o nome de A1.COB.
Ignora todos os caracteres que ultrapassam a coluna 72.
- PIP LST:=A1.COB[NP] Identico a PIP PRN:=A1.COB
- PIP B:=A1.COB[SINI↑Z QFIM↑Z] Copia o arquivo A1.COB do drive corrente no drive B com o mesmo nome.
Inicia a cópia se encontrar o texto "INI" e termina ao encontrar o texto "FIM".
- PIP LST:=A2.COB[FP30] Lista o arquivo A2.COB na impressora, filtra os saltos de página anteriores e coloca um novo a cada 30 linhas.
- PIP B:=A:X.REL[OV] Tranfere todos os arquivos tipo REL do drive A para o drive B.
Verifica se os dados foram sem gravados e ignora um fim de arquivo (1AH) se encontrar; força a leitura até o final.

4.5 - ENCADEAMENTO

Introdução: Dentro do sistema é possível encadear uma série de chamadas de programas, inclusive passando os parametros que um programa necessite para continuar a execução sem a intervenção do operador.

UTILITARIO 8: SUBMIT

Função: Executa uma sequencia de comandos de um arquivo tipo SUB.

Execução: SUBMIT NOME PARAM1 PARAM2..<CR>

Após a carga do programa na memória será criado no disco um arquivo \$\$\$SUB contendo os comandos dados no arquivo NOME.SUB.

Os parametros, se existirem, substituirão as referencias \$1, \$2... dentro do arquivo SUB.

MANUAL DO PROGRAMADOR

Criação do Arquivo SUB

Exemplo:

```
A> ED X.SUB
*I
DIR
STAT$1$2
PIP $3=$1$4
*E
```

Chamada do Comando

```
A> SUBMIT X B: TESTE.COB A:
STAT.COM A execução será:
```

```
DIR
STAT B: TESTE.COB
PIP A: =B:STAT.COM
```

A substituição foi assim efetuada.

```
$1=B:           $3=A:
$2=TESTE.COB   $4=STAT.COM
```

OBSERVAÇÃO:

Para a execução correta do programa SUBMIT, o drive corrente deverá ser o A.

Os arquivos e programas podem ser acessados em outros drives inclusive o arquivo tipo SUB, porem a cada execução de uma linha de comando no SUBMIT o sistema deve retornar para o drive A.

O número máximo de parametros é 9.

MANUAL DO PROGRAMADOR

UTILITARIO 9:

XSUB

Função:

Complementa a execução do SUBMIT, proporcionando a entrada de dados automática em programas que necessitam de leitura de dados pelo console.

Execução:

Deve ser colocado com a primeira linha de comando no arquivo SUB, que será chamado pelo SUBMIT.

Exemplo:

```
A ED X.SUB
*I
XSUB
ED ARQ1.COB
I LISTAGEM 1
E
TYPE ARQ1.COB
*E
A>
```

A execução será aberto o arquivo ARQ1.COB, inserida a linha "LISTAGEM 1" e em seguida listado no vídeo.

Conclusão:

É possível usar programas como EDITOR e outros que requerem comandos da console para sua execução.

Porem é preciso tomar cuidado com o uso excessivo do programa SUBMIT, pois se há uma queda de força, o programa interrompe apenas o comando que estava executando logo após a carga do sistema, recomeça a execução.

Somente um CTRL-C ou BREAK ou um erro nas chamadas introduzidas é que interrompem o programa.

4.6 - OUTROS

UTILITARIO 10:

STAT

Função:

Fornece estatística dos discos e arquivos e coloca condições para segurança nos arquivos.

1. STAT <CR>

Retorna o espaço livre nos drives ativos (drives onde houve pelo menos um acesso para leitura ou gravação após um RESET ou CTRL-C), e o tipo de assinalamento no formato:

D:ASSIN espaço nnnk

onde:

D: Drive Ativo

ASSIN = R/W ou R/O

nnnk = numero de kbytes livres no disco.

2. STAT D:<CR>

Retorna o espaço livre do drive especificado em D: no formato:

BYTES LIVRES no D: nnnk

3. STAT ref.mult.<CR>

Retorna a estatística sobre os arquivos referenciados e o espaço livre no drive, no formato:

MANUAL DO PROGRAMADOR

```
REGIST BYTES      EXT      IND
nnnn   bbbk       ee       ASS  D:NOME.TIPO
```

BYTES LIVRES NO D: nnnk

onde:

```
nnnn      numero de registros de 128 bytes
bbb       numero de kbytes ocupado pelo arquivo
ee        numero de extensões de 16k
ASS       R/O ou R/W
D:        Drive
Nome      nome do arquivo
Tipo      tipo do arquivo.
```

4. STAT ref.mult \$S <CR> acrescenta no modo anterior a
indicação TAMANHO

TAM REGIST.. TAM é o tamanho virtual do arquivo
em registro e REGIST é a soma dos
registros em cada extensão.
Para arquivos sequencias TAM e
REGIST é idêntico.

5. STAT ref.^{mult} ~~unica~~ \$ASS <CR> assinala uma condição para o
arquivo referenciado.
ASS pode ser:
R/O - arquivo somente poderá ser
lido.
R/W - retorna a condição de
leitura e gravação.

MANUAL DO PROGRAMADOR

SYS - arquivo não pode ser lido ou gravado nem transferido.

não aparece no diretório através do comando DIR; pelo comando STAT aparece entre parenteses.

DIR - retorna a condição normal; depois da condição SYSTEM.

6. STAT USR: <CR>

indica quais as areas que contem arquivos e qual a area usada no formato:

area ativa n

arquivos ativos n

Refere-se ao comando USER.

7. STAT D:DSK: <CR>

informa a estatistica do drive referido em D: no formato:

n1 - registros de 128 bytes

n2 - capacidade de kilobytes

n3 - arquivos no diretorio

n4 - checagem de no. de arquivos.

n5 - Regist/Ext

n6 - Regist/Bloco

n7 - Setores/Trilha

n8 - Trilha Sistema

UTILITARIO 11:

DUMPAS

Função:

exibe no video o conteudo de um arquivo.

Execução:

DUMPAS referencia unica S <CR>

O programa exhibira no video, o conteudo do arquivo referenciado a partir do setor S no formato:

n n1 n2.....n16 A1.....A16

onde:

n é o numero da linha

n1.....n16 16 bytes em hexadecimal

A1.....A16 16 bytes em codigo ASCII

OBSERVAÇÃO:

quando não existe tradução do valor hexadecimal para o formato ASCII ou é um caracter não grafico, é colocado um ponto.

qualquer tecla pressionada durante a execução, interrompe o programa.

para sair na impressora use a tecla CTRL-P antes de teclar CR. Se S é omitido, é assumido 1.

MANUAL DO PROGRAMADOR

UTILITARIO 12:

DESPPOOL

Função:

Libera a consola enquanto lista o conteúdo de um arquivo ASCII na impressora.

Execução:

DESPPOOL <CR>

Após a carga o sistema emitira mensagem:

"DESPPOOL PRONTO CTL-F"

A partir desse momento o programa estará disponível na memória, bastando teclar CTL-F.

Quando teclarmos CTRL-F emite a mensagem:

*NOME DO ARQUIVO:

Devemos entrar com o nome do arquivo que iremos imprimir, em seguida basta teclar CR para retornar ao sistema ou a operação que estivermos executando.

A cada CTRL-F o programa pedirá um novo arquivo.

Para desativar o programa devemos teclar CTRL-B ou RESET.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO:

O arquivo que será impresso pode estar em qualquer drive.

Ao teclar CTRL-C ou BREAK interrompemos apenas a listagem do arquivo referenciado no ultimo CTRL-F.

5- Introdução ao Editor de Textos.

5.1- Descrição Funcional.

O Editor de Textos do Sistema Operacional permite a criação e manutenção de arquivos contendo qualquer texto em código ASCII (programas fontes em COBOL, BASIC, FORTRAN, dados que não estejam compactados, etc...). O Programa será referenciado como:

ED.COM

O formato geral para execução do editor é:

ED D:NOME.TIPO

onde:

D: é o drive do arquivo que será editado

NOME.TIPO: referencia única segundo os padrões do SOP.

OBSERVAÇÃO: qualquer nome com qualquer tipo podem ser editados, porem convem lembrar que o uso da referencia multipla (*,?) poderá destruir o diretório do disco.

Durante a execução do editor serão criados os arquivos:

NOME.TIPO - arquivo que será editado.

NOME.BAK - arquivo criado com a versão anterior do arquivo editado.

NOME.\$\$\$ - arquivo temporário usado durante a edição.

X\$\$\$\$\$\$\$.LIB - arquivo temporário auxiliar criado como biblioteca.

MANUAL DO PROGRAMADOR

Quando usamos a forma ED NOME.TIPO

Se o arquivo referenciado na chamada do editor não existir no diretório do disco desejado, será enviada a mensagem:

NOVO ARQ

retornando em seguida um asterisco (*), indicando que o arquivo está aberto para acesso.

Nesse momento é criado no mesmo disco um arquivo com o nome de NOME. \$\$\$

Se o arquivo referenciado já existir será enviado apenas o asterisco (*).

Se o arquivo referenciado estiver na condição R/O, ativada pelo comando STAT, será enviada a mensagem:

"ARQ ESTA R/O"

sendo somente possível a leitura deste arquivo.

Se o arquivo referenciado estiver na condição SYSTEM ativada pelo comando STAT será enviada a mensagem:

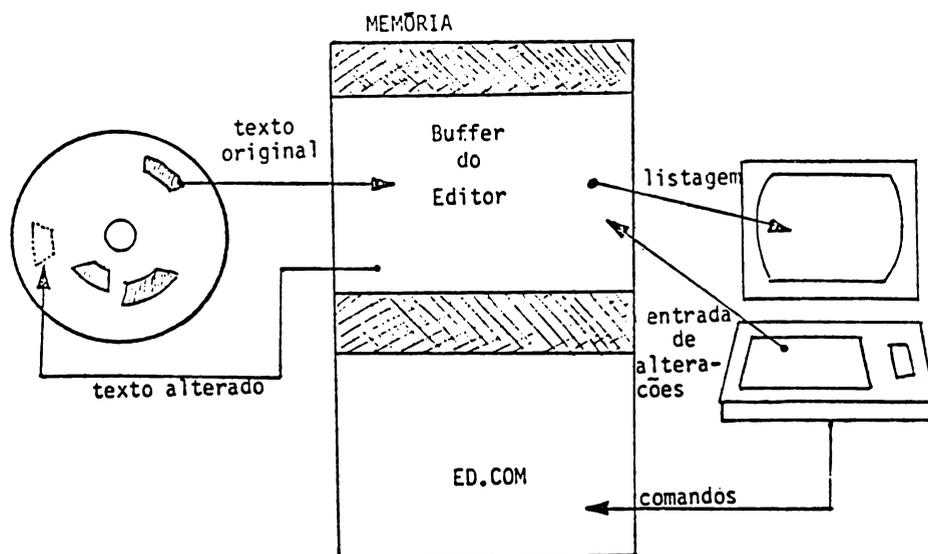
"SYSTEM" - ARQ BLOQUEADO

e o programa é abandonado.

Tanto no caso de um arquivo novo como no caso de arquivo existente, quando o editor é finalizado e não ocorre nenhum erro, o arquivo original NOME.TIPO será renomeado para NOME.BAK e o arquivo temporário NOME. \$\$\$ será renomeado para NOME.TIPO.

MÁNUAL DO PROGRAMADOR

ARQUIVOS NO DISCO



Se houver falta de espaço no disco, é possível criar o arquivo temporário em outro drive, com o arquivo formato:

ED D1:NOME.TIPO D2

onde:

D1 é o drive onde será lido o arquivo original caso exista, e onde será gravado ou renomeado para NOME.BAK

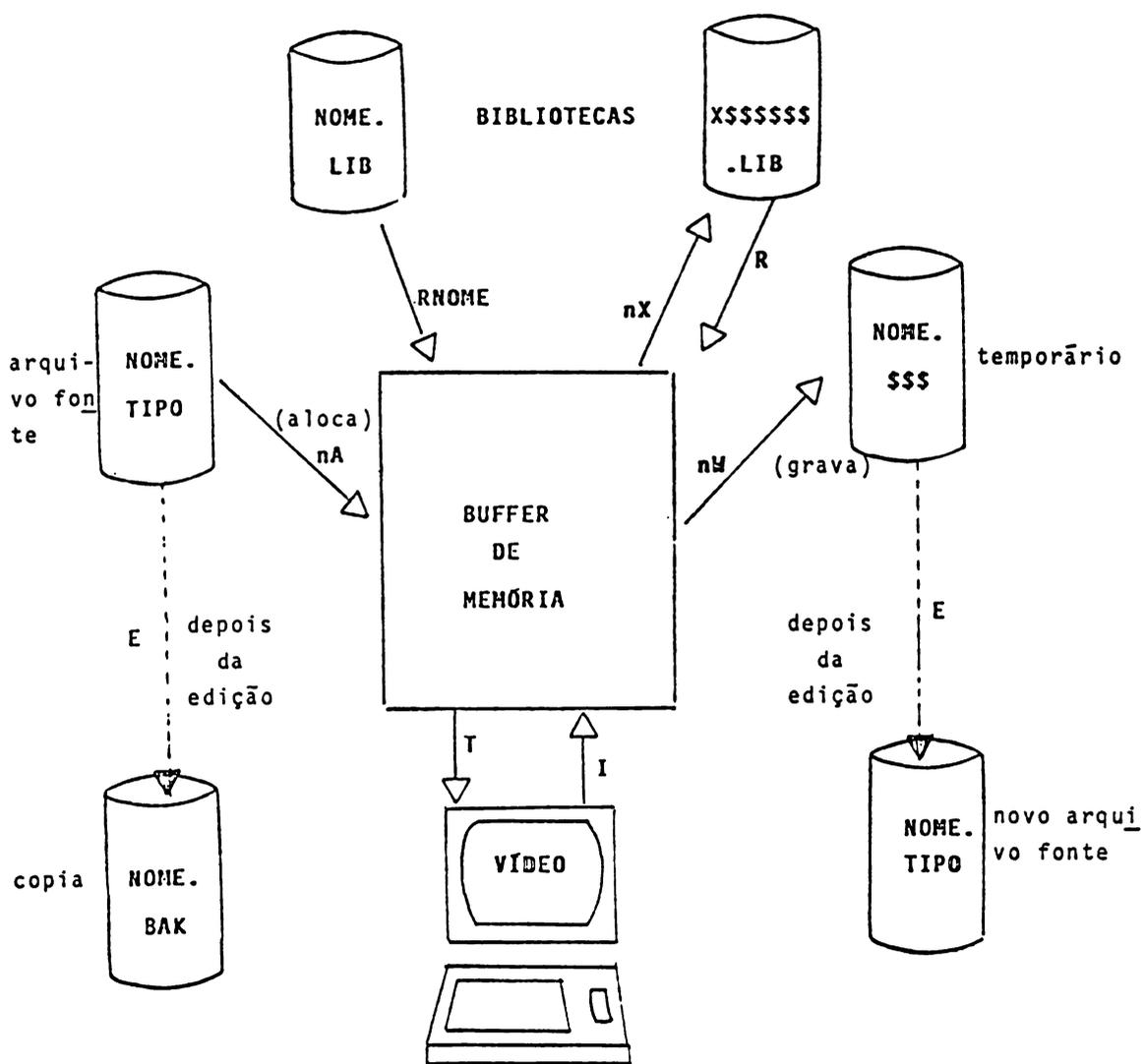
D2 é o drive onde será gravado o arquivo temporário NOME.\$\$\$ com todas as alterações executadas e depois renomeado para NOME.TIPO.

OBSERVAÇÃO: O programa editor pode ser interrompido em qualquer ponto da execução, sem que o arquivo referenciado seja destruído.

Nesse caso o arquivo tipo BAK não será criado, e todas as alterações executadas na edição serão anuladas.

(uma interrupção pode ser do tipo: queda de força, RESET, CONTROL-C, comandos próprios).

5.2 - DESCRIÇÃO GRÁFICA DO EDITOR

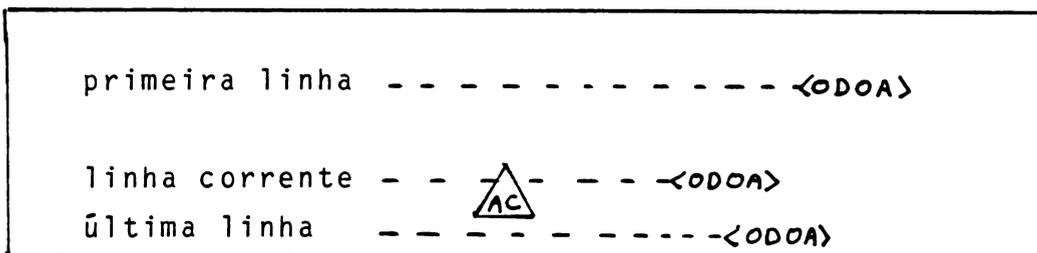


5.3 - ORGANIZAÇÃO DO BUFFER DE MEMÓRIA

O Buffer de memória pode ser considerado como uma sequência de linhas alocadas do arquivo fonte ou inseridas diretamente pela console. Associado ao Buffer de memória temos um apontador de caracteres imaginário (AC) que se move no Buffer sob o comando do operador. Notar que AC está sempre localizado adiante do primeiro caracter da primeira linha, antes do último caracter da última linha, ou entre dois caracteres. A linha corrente é a linha que contém o AC.

- AC apontador de caracter - aponta um caracter dentro de uma linha, está sempre posicionado no início da linha após cada operação.
- AF apontador do fonte - aponta a primeira linha não processada no arquivo fonte.
- AM apontador da memória - aponta a última linha alocada na memória.
- AT apontador do temporário - aponta a última linha gravada no arquivo temporário.

BUFFER DA MEMÓRIA



A linha pontilhada indica uma linha de tamanho qualquer terminada pelos delimitadores (ODOA)

MANUAL DO PROGRAMADOR

O Buffer de memória tem capacidade aproximada de 39.000 caracteres (aproximadamente 528 linhas de 72 caracteres mais OD,UA).

Todas as alterações ou inclusões são feitas diretamente no Buffer de memória, somente na finalização é que o conteúdo da edição é gravado em disco e os arquivos fechados.

Para manipulação das linhas do Buffer, dispomos de uma série de comandos, mais suas possíveis combinações; valem ainda as teclas de edição e controle disponíveis.

Alguns comandos podem ser executados várias vezes automaticamente, para isso indicaremos um número inteiro n que poderá variar de 0 até 65.535.

O caracter # substitui o valor 65.535; na omissão de n é assumido o valor 1, todos os comandos devem ser seguidos da tecla CR.

5.4 - COMANDOS DE MOVIMENTAÇÃO DO BUFFER

COMANDO 1 Alocar nA

FUNÇÃO: para tornar disponível um texto fonte, se quisermos alterá-lo, é necessário que este seja alocado do disco para o Buffer de memória. Através de comando A podemos alocar quantas linhas quisermos.
A cada execução do comando A, são alocadas as n próximas linhas do texto fonte a partir de AF para o Buffer de memória a partir de AM.
AF e AM são incrementados de n.

COMANDO 2: Gravar nW

FUNÇÃO: Nem sempre o texto fonte pode ser alocado totalmente na memória; quando isso acontece, é necessário editar em partes o texto fonte.
O comando W permite esvaziar quantas linhas quisermos da memória.
A cada execução do comando W são transferidas as n primeiras linhas do Buffer de memória para o espaço livre no arquivo temporário a partir de AT.
As linhas restantes no Buffer são deslocadas de n+1 de AM até o topo do Buffer. AT é encrementado de n.

MANUAL DO PROGRAMADOR

OBSERVAÇÃO: os comandos A e W apresentam uma particularidade:
Se usarmos OA, será preenchido o Buffer de memória até que pelo menos a metade esteja cheia.
Se usarmos OW, será transferidas as linhas até que pelo menos a metade do Buffer esteja vazia.

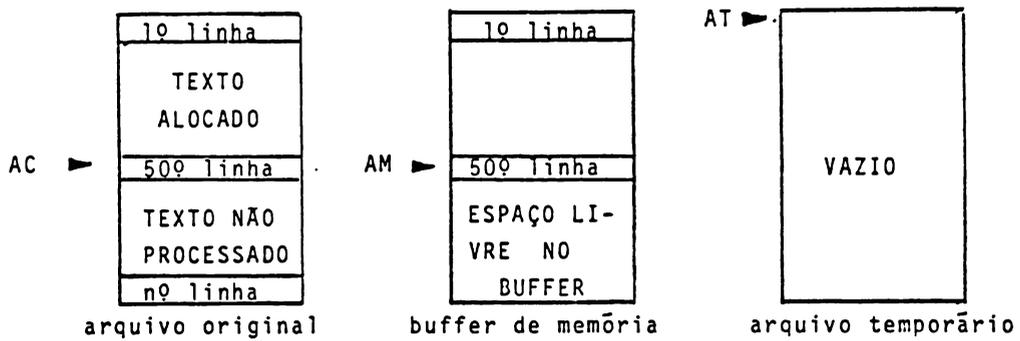
MANUAL DO PROGRAMADOR

EXEMPLO DE ALOCAÇÃO DE LINHAS NA MEMÓRIA

INÍCIO DA EDIÇÃO



DEPOIS DA EXECUÇÃO DO COMANDO 50A



DEPOIS DA EXECUÇÃO DO COMANDO 25W



MANUAL DO PROGRAMADOR

5.5 - COMANDOS DE MOVIMENTAÇÃO DO APONTADOR

COMANDO 3: Início, Fim B ou -B

FUNÇÃO: Move o apontador (AC) para a primeira linha do texto no Buffer (B) ou para a última linha (-B). Não mostra na console o conteúdo da linha em que parou.

COMANDO 4: Apagar nK ou -nK

FUNÇÃO: Apagar do Buffer de memória n linhas a partir da linha corrente (nK) ou n linhas anteriores a corrente (-nK).

COMANDO 5: Listar nT ou -nT

FUNÇÃO: Lista o conteúdo de n linhas a partir da corrente (nT) ou anteriores a corrente (-nT) sem mudar a posição do apontador (AC).
Se n=0 lista a linha corrente.

COMANDO 6: Movimentação AC n ou -n

FUNÇÃO: movimenta o apontador (AC) n linhas a frente (n) da posição atual ou n linhas para tras (-n) imprimindo a linha da nova posição atual.
Se n=0 imprime a linha corrente, sem alterar a posição do apontador. Somente a tecla CR, é assumida o valor de n=1.

MANUAL DO PROGRAMADOR

COMANDO 7: Página nP ou -nP

FUNÇÃO: Move o AC e lista n páginas de texto do Buffer a frente do apontador (n) ou atrás (-n). Uma página contém 24 linhas. OP lista a linha e a página do apontador.
O apontador é movimentado para o início da página listada.

5.6 - COMANDOS DE FINALIZAÇÃO DE TEXTOS

COMANDO 8: FIM E

FUNÇÃO: Finaliza o editor copiando o texto armazenado no Buffer mais as linhas não processadas do arquivo fonte, para o arquivo temporário. Renomeia o arquivo fonte para tipo BAK e o arquivo temporário para o tipo da fonte. Retorna para o Sistema.

MANUAL DO PROGRAMADOR

COMANDO 9: FIM-RETORNO H

FUNÇÃO Finaliza o editor copiando o texto armazenado no Buffer mais as linhas não processadas do arquivo fonte para o arquivo temporário. Renomeia o arquivo fonte para tipo BAK e o arquivo temporário para o tipo do fonte, esvazia o Buffer de memória, cria um novo arquivo temporário (reinicia o processo de edição).

COMANDO 10: Origem O

FUNÇÃO: Ignora todo o processo de edição, esvazia o Buffer e o arquivo temporário, retorna os apontadores as origens (reinicia o processo de edição).

COMANDO 11: Abandonar Q

FUNÇÃO: Ignora todo o processo de edição, apaga o arquivo temporário, não cria o arquivo BAK, retorna ao sistema.

NOTA: Os comandos O e Q quando executados emitem a mensagem:
(Y/N)? - somente a letra Y prossegue a execução do comando.

5.7 - COMANDOS DE MANIPULAÇÃO DE TEXTOS

COMANDO 12: Inserir I ou Itexto

FUNÇÃO: Aceita da console qualquer número de linhas, onde cada linha é terminada com a tecla CR, até que seja digitado um CTRL-Z (Z). O apontador AC será colocado depois do último caracter digitado.

Se o comando I for digitado em letra maiúscula, todo o texto de entrada será inserido em maiúscula.

Caso contrário o texto será inserido em maiúscula ou minúscula conforme a digitação.

No segundo formato (Itexto), o texto que segue é inserido automaticamente, não é preciso digitar CTRL-Z.

Em ambos os casos o texto é inserido acima da linha do apontador (AC)

OBSERVAÇÃO: Notar que quando o comando I é executado, todos os outros comandos são ignorados até que um CTRL-Z seja teclado, isto é, dentro de um modo de inserção, todos os caracteres são válidos, inclusive as teclas de edição e controle.

Se entrarmos no meio de um texto com o comando I, e usarmos a tecla DEL, conseguiremos apagar todo o texto acima do ponto em que está posicionado AC.

É terminada com a tecla CR, até que seja digitado um.

MANUAL DO PROGRAMADOR

COMANDO 13: Procura nFtexto

FUNÇÃO: procura a partir de AC, a enésima ocorrência do texto, se encontrar posiciona o AC no fim do texto procurado, caso não desloca AC.
Se terminarmos o comando com um CTRL-Z podemos colocar outros comandos.

COMANDO 14: Substituição nStexto1↑Ztexto2

FUNÇÃO: Substitui n vezes o texto1 pelo texto2.
texto1 e texto2 não precisam ter o mesmo tamanho, o comando ajusta automaticamente deslocando o conteúdo da linha para a direita ou esquerda.
Se o texto2 for omitido, o texto1 será apagado da linha.

5.8 - COMANDOS DE USO GERAL

COMANDO 15: Maiuscula U ou -U

FUNÇÃO: Transforma em maiúscula todos os caracteres alfabéticos que entrarem no Buffer da memória, através da console ou alocada do arquivo fonte. O comando -U desativa o comando (o que foi gravado com maiúscula permanece).

MANUAL DO PROGRAMADOR

COMANDO 16: Numeração V ou -V

FUNÇÃO: numera na console as linhas do texto do Buffer de memória para orientação do operador. No formato:

n: conteúdo da linha

onde n é um no. sempre com cinco dígitos no intervalo. 0 a 65.535.

A linha em que está posicionado o AC é apresentada também; no formato:

n: (cursor)

No início da edição o comando V já está ativado para desativar, usar -V.

Quando o fim do texto no Buffer é apresentado apenas 5 brancos e dois pontos (:).

COMANDO 17: Estatística GV

FUNÇÃO: imprime o espaço vazio do Buffer e o tamanho deste formato.

LIVRES/TOTAL (bytes).

MANUAL DO PROGRAMADOR

COMANDO 18 : Posicionamento :n, n: e n1:n2T

FUNÇÃO: quando usamos :n, AC é colocado na posição n+1 se n for maior que a atual posição do cursor ou na posição n se n é menor.
O conteúdo da linha posicionado é listado.
quando usamos n:, AC é colocado na posição n.
se usado n1:n2T, o AC é colocado na posição n1 e será listado na console as linhas da posição n1 até n2

5.9 - COMANDOS MAIS COMPLEXOS

COMANDO 19: Biblioteca nX

FUNÇÃO: Transfere para o arquivo temporário auxiliar X\$\$\$\$\$.LIB o conteúdo das n próximas linhas a partir de AC.
quando o comando X é executado mais que uma vez, o conteúdo das transferências anteriores permanecem, as novas linhas são colocadas abaixo das anteriores.
para se esvaziar o arquivo temporário auxiliar usamos n=0 (0X).

MANUAL DO PROGRAMADOR

COMANDO 23: Movimenta AC nL ou -nL

FUNÇÃO: Movimenta AC n linhas a frente (nL) ou atrás (-nL). Não lista a linha.

COMANDO 24: Procura nNtexto .

FUNÇÃO: Procura a enésima ocorrência do texto dado a partir de AC, se não encontrar, transfere Buffer para o arquivo temporário, aloca o restante do texto não processado do arquivo fonte no Buffer, repete esse processo até encontrar a ocorrência procurada. Se a ocorrência não existir o Buffer de memória ficará vazio.

COMANDO 25: Justaposição nJtexto1↑Ztexto2↑Ztexto3

FUNÇÃO: Procura a partir de AC a enésima ocorrência do texto1, encontrando insere no texto2 e move AC para o caracter seguinte ao fim de texto2, apagando os caracteres entre texto2 e texto3 (não inclui texto3). Se texto3 não existir, não será apagado nenhum caracter.

MANUAL DO PROGRAMADOR

Exemplo do Comando J

```
Linha      o caderno é AZUL < OD OA >
Comando    JO ZQUE?↑Z↑L
Resultado  O QUE? < OD OA >
```

Nota ↑L = CTRL-L que representa o par ODOA.

COMANDO 26: Repetição nMC1C2C3

Função: permite usar um grupo de comandos juntos, repetindo n vezes. C1C2C3 representa um grupo de comando do editor separado por CTRL-Z, não incluindo outro comando M. Serão executados n vezes se n # 1, se n=0 ou n=1, serão executados até que uma condição de erro seja encontrada.

Exemplo do comando M

```
MSROTINA↑ZROT2↑ZOTT < CR >
```

5.10 -MENSAGEM DE ERRO

Quando uma condição de erro ocorre o programa editor imprime o último caracter lido antes do erro. Mais um indicador de erro no formato:

```
BREAK "C" AT X
```

onde X - é o último caracter lido

C - é o último indicador da condição de erro.

C pode ser:

"?" Comando errado

">" Buffer de memória cheio

"#" o comando não pode ser executado

o número de vezes especificado

"O" não pode abrir o arquivo tipo LIB no comando R.

MANUAL DO PROGRAMADOR

APENDICE "A" - RESUMO DO EDITOR

nA	alocar ; =65535
nW	gravar (esvaziar buffer)
B, -B	Inicio, Fim
nK, -nK	apagar linhas
nI, -nI	lista
n, -n	movimenta apontador
nP, -nP	página
E	FIM
H	Fim - Retorno
U	Origem
Q	Abandonar
I ;Itexto	Inserir
nFtexto	Procura
nStexto1 Ztexto2	Substituição
U, -U	Maiusculas
V, -V	Numeração
nX	Biblioteca
R, Rnome	Ler biblioteca
ERROS:-	
?	comando errado
	Buffer de memória cheio
	no. de vezes não completado
U	não existe biblioteca.

MANUAL DO PROGRAMADOR

APENDICE "B" - TABELA DE CONVERSÃO
BINÁRIO, OCTAL, DECIMAL E HEXADECIMAL

<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>	<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>	<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>
00000000	000	000	00	00110011	063	051	33	01100110	146	102	66
00000001	001	001	01	00110100	064	052	34	01100111	147	103	67
00000010	002	002	02	00110101	065	053	35	01101000	150	104	68
00000011	003	003	03	00110110	066	054	36	01101001	151	105	69
00000100	004	004	04	00110111	067	055	37	01101010	152	106	6A
00000101	005	005	05	00111000	070	056	38	01101011	153	107	6B
00000110	006	006	06	00111001	071	057	39	01101100	154	108	6C
00000111	007	007	07	00111010	072	058	3A	01101101	155	109	6D
00001000	010	008	08	00111011	073	059	3B	01101110	156	110	6E
00001001	011	009	09	00111100	074	060	3C	01101111	157	111	6F
00001010	012	010	0A	00111101	075	061	3D	01110000	160	112	70
00001011	013	011	0B	00111110	076	062	3E	01110001	161	113	71
00001100	014	012	0C	00111111	077	063	3F	01110010	162	114	72
00001101	015	013	0D	01000000	100	064	40	01110011	163	115	73
00001110	016	014	0E	01000001	101	065	41	01110100	164	116	74
00001111	017	015	0F	01000010	102	066	42	01110101	165	117	75
00010000	020	016	10	01000011	103	067	43	01110110	166	118	76
00010001	021	017	11	01000100	104	068	44	01110111	167	119	77
00010010	022	018	12	01000101	105	069	45	01110000	170	120	78
00010011	023	019	13	01000110	106	070	46	01110001	171	121	79
00010100	024	020	14	01000111	107	071	47	01111010	172	122	7A
00010101	025	021	15	01001000	110	072	48	01111011	173	123	7B
00010110	026	022	16	01001001	111	073	49	01111100	174	124	7C
00010111	027	023	17	01001010	112	074	4A	01111101	175	125	7D
00011000	030	024	18	01001011	113	075	4B	01111110	176	126	7E
00011001	031	025	19	01001100	114	076	4C	01111111	177	127	7F
00011010	032	026	1A	01001101	115	077	4D	10000000	200	128	80
00011011	033	027	1B	01001110	116	078	4E	10000001	201	129	81
00011100	034	028	1C	01001111	117	079	4F	10000010	202	130	82
00011101	035	029	1D	01010000	120	080	50	10000011	203	131	83
00011110	036	030	1E	01010001	121	081	51	10000010	204	132	84
00011111	037	031	1F	01010010	122	082	52	10000011	205	133	85
00100000	040	032	20	01010011	123	083	53	10000110	206	134	86
00100001	041	033	21	01010100	124	084	54	10000111	207	135	87
00100010	042	034	22	01010101	125	085	55	10001000	210	136	88
00100011	043	035	23	01010110	126	086	56	10001001	211	137	89
00100100	044	036	24	01010111	127	087	57	10001010	212	138	8A
00100101	045	037	25	01011000	130	088	58	10001011	213	139	8B
00100110	046	038	26	01011001	131	089	59	10001100	214	140	8C
00100111	047	039	27	01011010	132	090	5A	10001101	215	141	8D
00101000	050	040	28	01011011	133	091	5B	10001110	216	142	8E
00101001	051	041	29	01011100	134	092	5C	10001111	217	143	8F
00101010	052	042	2A	01011101	135	093	5D	10010000	220	144	90
00101011	053	043	2B	01011110	136	094	5E	10010001	221	145	91
00101100	054	044	2C	01011111	137	095	5F	10010010	222	146	92
00101101	055	045	2D	01100000	140	096	60	10010011	223	147	93
00101110	056	046	2E	01100001	141	097	61	10010100	224	148	94
00101111	057	047	2F	01100010	142	098	62	10010101	225	149	95
00110000	060	048	30	01100011	143	099	63	10010110	226	150	96
00110001	061	049	31	01100100	144	100	64	10010111	227	151	97
00110010	062	050	32	01100101	145	101	65				

MANUAL DO PROGRAMADOR

APENDICE "C" - TABELA ASCII

CARACTER	HEXA	CARACTER	HEXA	CARACTER	HEXA
NUL	00	*	2A	T	54
SOH	01	+	2B	U	55
STX	02	,	2C	V	56
ETX	03	-	2D	W	57
EOT	04	.	2E	X	58
ENG	05	/	2F	Y	59
ACK	06	0	30	Z	5A
BEL	07	1	31	[5B
BS	08	2	32	\	5C
HT	09	3	33]	5D
LF	0A	4	34	^	5E
VT	0B	5	35	_	5F
FF	0C	6	36	`	60
CR	0D	7	37	a	61
SO	0E	8	38	b	62
SI	0F	9	39	c	63
DLE	10	:	3A	d	64
DC1	11	;	3B	e	65
DC2	12	<	3C	f	66
DC3	13	=	3D	g	67
DC4	14	>	3E	h	68
NAK	15	?	3F	i	69
SYN	16	@	40	j	6A
ETB	17	A	41	k	6B
CAN	18	B	42	l	6C
EM	19	C	43	m	6D
SUB	1A	D	44	n	6E
ESC	1B	E	45	o	6F
FS	1C	F	46	p	70
GS	1D	G	47	q	71
RS	1E	H	48	r	72
VS	1F	I	49	s	73
SP	20	J	4A	t	74
!	21	K	4B	u	75
"	22	L	4C	v	76
#	23	M	4D	w	77
\$	24	N	4E	x	78
%	25	O	4F	y	79
&	26	P	50	z	7A
'	27	Q	51	{	7B
(28	R	52		7C
)	29	S	53	}	7D
				~	7E
				DEL	7F

MANUAL DO PROGRAMADOR

<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>	<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>	<u>N₂</u>	<u>N₈</u>	<u>N₁₀</u>	<u>N₁₆</u>
10011000	230	152	98	10111010	272	186	BA	11011101	335	221	DD
10011001	231	153	99	10111011	273	187	BB	11011110	336	222	DE
10011010	232	154	9A	10111100	274	188	BC	11011111	337	223	DF
10011011	233	155	9B	10111101	275	189	BD	11100000	340	224	EA
10011100	234	156	9C	10111110	276	190	BE	11100001	341	225	E1
10011101	235	157	9D	10111111	277	191	BF	11100010	342	226	E2
10011110	236	158	9E	11000000	300	192	C0	11100011	343	227	E3
10011111	237	159	9F	11000001	301	193	C1	11100100	344	228	E4
10100000	240	160	A0	11000010	302	194	C2	11100101	345	229	E5
10100001	241	161	A1	11000011	303	195	C3	11100110	346	230	E6
10100010	242	162	A2	11000100	304	196	C4	11100111	347	231	E7
10100011	243	163	A3	11000101	305	197	C5	11101000	350	232	E8
10100100	244	164	A4	11000110	306	198	C6	11101001	351	233	E9
10100101	245	165	A5	11000111	307	199	C7	11101010	352	234	EA
10100110	246	166	A6	11001000	310	200	C8	11101011	353	235	EB
10100111	247	167	A7	11001001	311	201	C9	11101100	354	236	EC
10101000	250	168	A8	11001010	312	202	CA	11101101	355	237	ED
10101001	251	169	A9	11001011	313	203	CB	11101110	356	238	EE
10101010	252	170	AA	11001100	314	204	CC	11101111	357	239	EF
10101011	253	171	AB	11001101	315	205	CD	11110000	360	240	F0
10101100	254	172	AC	11001110	316	206	CE	11110001	361	241	F1
10101101	255	173	AD	11001111	317	207	CF	11110010	362	242	F2
10101110	256	174	AE	11010000	320	208	D0	11110011	363	243	F3
10101111	257	175	AF	11010001	321	209	D1	11110100	364	244	F4
10110000	260	176	B0	11010010	322	210	D2	11110101	365	245	F5
10110001	261	177	B1	11010011	323	211	D3	11110110	366	246	F6
10110010	262	178	B2	11010100	324	212	D4	11110111	367	247	F7
10110011	263	179	B3	11010101	325	213	D5	11111000	370	248	F8
10110100	264	180	B4	11010110	326	214	D6	11111001	371	249	F9
10110101	265	181	B5	11010111	327	215	D7	11111010	372	250	FA
10110110	266	182	B6	11011000	330	216	D8	11111011	373	251	FB
10110111	267	183	B7	11011001	331	217	D9	11111100	374	252	FC
10111000	270	184	B8	11011010	332	218	DA	11111101	375	253	FD
10111001	271	185	B9	11011011	333	219	DB	11111110	376	254	FE
				11011100	334	220	DC	11111111	377	255	FF

MANUAL DO PROGRAMADOR

APENDICE "D" TABELA DOS CONTROLES DISPONIVEIS

CONTROL	VALOR HEXA	FUNÇÃO
CTRL - A	01	
CTRL - B	02	
CTRL - C	03	BREAK
CTRL - D	04	
CTRL - E	05	
CTRL - F	06	
CTRL - G	07	BELL
CTRL - H	08	BS
CTRL - I	09	TAB
CTRL - J	0A	Salta Linha
CTRL - K	0B	
CTRL - L	0C	Limpa video
CTRL - M	0D	RETURN
CTRL - N	0E	BRILHO
CTRL - O	0F	
CTRL - P	10	Ativa Impressora
CTRL - Q	11	
CTRL - R	12	Reescreve Linha
CTRL - S	13	PAUSE
CTRL - T	14	Libera Cursor
CTRL - U	15	Apaga linha + LF
CTRL - V	16	
CTRL - W	17	
CTRL - X	18	Apaga linha
CTRL - Y	19	
CTRL - Z	1A	Sobe 1 linha
CTRL - [1B	
CTRL - \	1C	
CTRL -]	1D	HOME
CTRL - ^	1E	Limpa o fim da linha
CTRL - _	1F	Limpa da posição do curso p/ baixo.

MANUAL DO PROGRAMADOR

APENDICE "E" - POTENCIAS DE 2

$2^0 = 1$	$2^{17} = 131,072$	$2^0 = 1.0$
$2^1 = 2$	$2^{18} = 262,144$	$2^{-1} = 0.5$
$2^2 = 4$	$2^{19} = 524,288$	$2^{-2} = 0.25$
$2^3 = 8$	$2^{20} = 1,048,576$	$2^{-3} = 0.125$
$2^4 = 16$	$2^{21} = 2,097,152$	$2^{-4} = 0.0625$
$2^5 = 32$	$2^{22} = 4,194,304$	$2^{-5} = 0.03125$
$2^6 = 64$	$2^{23} = 8,388,608$	$2^{-6} = 0.015625$
$2^7 = 128$	$2^{24} = 16,777,216$	$2^{-7} = 0.0078125$
$2^8 = 256$	$2^{25} = 33,554,432$	$2^{-8} = 0.0039062$
$2^9 = 512$	$2^{26} = 67,108,864$	$2^{-9} = 0.0019531$
$2^{10} = 1024$	$2^{27} = 134,217,728$	$2^{-10} = 0.0009766$
$2^{11} = 2048$	$2^{28} = 268,435,456$	$2^{-11} = 0.0004883$
$2^{12} = 4096$	$2^{29} = 536,870,912$	$2^{-12} = 0.0002441$
$2^{13} = 8192$	$2^{30} = 1,073,741,824$	$2^{-13} = 0.0001220$
$2^{14} = 16,384$	$2^{31} = 2,147,483,648$	$2^{-14} = 0.0000610$
$2^{15} = 32,768$	$2^{32} = 4,294,967,296$	$2^{-15} = 0.0000305$
$2^{16} = 65,536$		$2^{-16} = 0.0000153$

MANUAL DO PROGRAMADOR

APENDICE "F" - RESUMO UTILITARIOS

UTILITARIO	FUNÇÃO
REFORM	Formatação de Discos
SYSGEN	Copia Sistema Operacional
SETUP	Atualiza Sistema Operacional
MODE	Atualiza densidade do Drive
COPY	Copia densidade simples
DCOPY	Copia densidade dupla
PIP	Transferencia de arquivo
ED	Editor de Textos
SUBMIT	Encadeamento de Programas
XSUB	Parametros para o SUBMIT
STAT	Estatisticas
DUMPAS	DUMP de um arquivo
DESPool	Concorre a impressora com a console

MANUAL DO PROGRAMADOR

APENDICE "G" - CONVERSÃO DE ARQUIVOS ATRAVÉS DO SORT

COBOL-80 utiliza os seguintes tipos de arquivos:

- 1 - Line sequential
- 2 - Sequential
- 3 - Relative
- 4 - Indexed

1 - Line sequential

Nesta organização os registros são delimitados pelos caracteres ASCII ODH, OAH ("Carriage Return" e "Line Feed") entre registros.

Os dados nesta organização são representados somente na forma de caracteres de Texto ASCII.

2 - Sequential

A esta classe pertencem os arquivos cujo tamanho do registro são armazenados nos primeiros dois bytes de cada registro.

Os arquivos poderão ter registros de tamanhos variáveis.

OBS.:O tamanho de cada registro é armazenado separadamente dos dados.

MANUAL DO PROGRAMADOR

1 - COMANDO INPUT-ATRIBUTES

Sua sintaxe é a seguinte:

INPUT-ATRIBUTES = record-length Tipo de organização

ou

INPUT = record-length Tipo de organização

ou

I = record-length Tipo de organização

onde:

Record-length é o tamanho do registro no arquivo variando de 1 a 4096 bytes. Note-se que se o arquivo for do tipo Line Sequential (Cobol-80) devem ser computados ao tamanho do registro 2 bytes a mais.

EX. Registro com 100 bytes de dados implica num tamanho de registro igual a 102 bytes.

Os tipos de organização podem ser:

FIXED-LENGTH Arquivos de tamanho fixo sem delimitadores entre registros.

CR-DELIMITED Line sequential

VARIABLE Sequential (COBOL-80)

RELATIVE Arquivo relativo.

MANUAL DO PROGRAMADOR

2 - Comando OUTPUT-FILE

Sua sintaxe é a seguinte:

OUTPUT-FILE =Filename /C ,REC-LEN tipos de organização

ou

OUTPUT = Filename /C ,REC-len tipos de organização

ou

O = Filename /C ,REC-len tipos de organização

onde

Filename Nome do arquivo de saída.

/C Indica que se quer a mudança do diskette antes do arquivo de saída ser gravado.

REC-len Número variando de 1 a 4096 especificando o tamanho do registro em bytes. Mesmo tratamento feito pelo record-length no INPUT-ATRIBUITES.

Tipos de organização:

FIXED-LENGTH

CR-DELIMITED

VARIABLE

RELATIVE

MANUAL DO PROGRAMADOR

Para a conversão de um arquivo procede da seguinte maneira:

- 1 - Especificar a organização do arquivo a ser convertido no comando INPUT-ATTRIBUTES.
- 2 - Especificar o arquivo convertido no comando OUTPUT-FILE.
- 3 - Realizar o SORT.

MANUAL DO PROGRAMADOR

3 - Relative

Contem registros de tamanho fixo sem delimitadores. Os primeiros 6 bytes do arquivo pertencem ao "Header" que indica o tamanho do registro e a chave do último registro gravado.

4 - Indexed

Não é possível efetuar conversão neste tipo de organização que é uma mistura de chaves e dados num mesmo arquivo.

Para realizar conversão para os tipos de organização descritos anteriormente dois comandos foram modificados no utilitário SORT, os quais são descritos a seguir:

