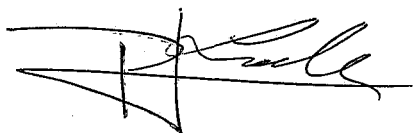


UM EDITOR CONVERSACIONAL PARA UM MINI-COMPUTADOR

Sdnei de Brito Alves

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO  
GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

Aprovada por:



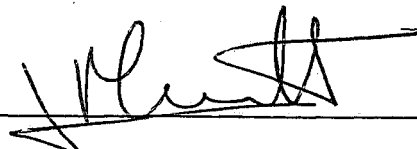
---

Prof. Pierre-Jean Lavelle  
(orientador)



---

Prof. Nelson Maculan Filho



---

Prof. Jean-Pierre Courtiat

RIO DE JANEIRO-RJ-BRASIL

DEZEMBRO DE 1976

A  
MEUS PAIS e  
IRMÃOS.

AGRADECIMENTOS

Ao Professor Pierre Jean Lavelle pela indicação do assunto e orientação deste trabalho; ao Professor Nelson Maculan Filho tanto pelo apoio como pelo incentivo constante; aos Professores Gerhard Schwartz e Jean-Pierre Courtiat; aos funcionários da Secretaria do Programa de Sistemas; aos demais professores e colegas da COPPE que de alguma forma colaboraram com o nosso projeto.

R E S U M O

O presente trabalho constitui na elaboração e implementação de um sistema Editor que permite a geração, modificação e documentação on-line de programas e eventualmente de textos livres de setenta e dois caracteres por linha.

Os algoritmos deste sistema foram programados nas linguagens Assembler e Fortran e a implementação foi feita em um mini-computador Mitra-15. É operado através do terminal de vídeo IRIS-200.

A B S T R A C T

This work consist in the design and implementation of an Editor system that permits the on-line generation, modification and documentation of programs and eventually of free texts of seven two characteres per line.

The algorithms of this system were programmed in Fortran and Assembler languages and the implementation was made in the mini-computer Mitra-15.

It is operated through the IRIS 200 video terminal.

ÍNDICE

<u>CAPÍTULO I</u>	1
1. Introdução	1
<u>CAPÍTULO II</u>	4
2. Facilidades Apresentadas por Diversos Tipos de Editores de Textos	4
2.1. Editor de Text Wylbur	5
2.2. Conversational Context-Directed Editor	5
2.3. Text Editor and Corrector (TECO)	6
2.4. Quick Editor	6
2.5. TVEDIT	7
2.6. Interactive Programming Support System	7
2.7. Edit-10	8
2.8. Magnetic Tape Selectric Type-Writes (MTST)	8
2.9. ASTROTYPE	9
2.10. EDITPAK	9
2.11. SCRIPT	10
2.12. EDIT/360	11
2.13. Hypertext Editing System (HES)	12
2.14. System/360 Administrative Terminal System (ATS)	
VIPCOM	12

<u>CAPÍTULO III</u>	13
3. Decidibilidade e Implementação	13
3.1. Introdução ao Sistema	13
3.1.1. Características Principais	14
3.1.2. Outras Características	14
3.2. Catálogo da zona DA	16
3.3. Um Sistema de Arquivo em Disco para Acesso Aleatório	17
3.3.1. Arquivo	17
3.3.1.1. Lista de Espaço-Disponível	20
3.4. Linguagem	22
3.4.1. Definição de Arquivos	22
3.4.2. Seleção de Arquivos	25
3.4.3. Remoção	28
3.4.4. Inserções	29
3.4.4.1. Overflow	31
3.4.4.2. I/O Sobre o disco	32
3.4.5. Listagem do fonte	33
3.4.5.1. Associações sobre PT	34
3.4.6. Cópia de Arquivos	35
3.4.7. Preparação para Compilação	35
3.4.7.1. Arquivo de Trabalho do Editor de Programas	36
3.5. Linguagem Usada na Implementação	38
3.6. Algumas Características Particulares	40
<u>CAPÍTULO IV</u>	42
4. Problemas da Implementação	42

4.1	Memória Real	42
4.1.1.	Memória Auxiliar	45
4.2.	Sistema de Acesso Disco	45
4.3.	Alteração no Control-block do Compilador Fortran e Fixação do Tamanho do Catálogo	46
<u>CAPÍTULO V</u>		48
5.	Conclusões	48
<u>BIBLIOGRAFIA</u>		51
<u>ANEXO I (MANUAL DO USUÁRIO)</u>		53
1.	Generalidades	54
2.	Funções das Teclas	54
3.	De Como Ativar o Editor	56
4.	Linguagem - Comandos	58
4.1.	Definição de Arquivos - Sintaxe do Comando	58
4.2.	Seleção de arquivos - Sintaxe do Comando	58
4.3.	Inserção - Sintaxe do Comando	59
4.4.	Listagem - Sintaxe do Comando	59
4.5.	Transferência de Arquivo - Sintaxe do Comando	60
4.6.	Remoção de arquivos, linha ou trechos Sintaxe do Comando	61
4.7.	Apronta para compilação - Sintaxe do Comando	61
5.	Funções dos Comandos	62
6.	Tipos de Erros	69



ANEXO II (LISTAGEM)

## CAPÍTULO I

### 1. INTRODUÇÃO

Com o surgimento de terminais permitindo a comunicação direta com um computador por usuário comum, despertou-se interesse generalizado em usar os recursos do computador em outras aplicações que não as simplesmente numéricas.

A criação, modificação e documentação on-line de programas e de textos livres, são exemplos dessas novas aplicações.

Talvez devido aos poucos recursos que são oferecidos pelos equipamentos existentes no laboratório de Sistemas da COPPE, não houve antes, interesse em se desenvolver projetos particulares de Sistemas on-line de Edição de Textos.

Além de outros motivos, a não existência de sistemas semelhantes funcionando nesse laboratório, serviu de motivação para a proposição de um projeto de Editor de Programas que permitisse aos usuários todas as operações necessárias, desde a geração, até a preparação de programas para compilação.

O sistema proposto utiliza equipamentos disponíveis no Laboratório de Sistemas e que, cujas características, estão abaixo descritas:

- 1) Mini-Computador Mitra-15 da CII-Thompson, com 16 K de memória, organizada em palavras de 16

bits + 1 bit de paridade e de 1 bit de proteção. A memória é endereçável por byte ou palavra e alterável por byte, palavra ou palavra-dupla.

2) Teletipo Modelo CII-1501, acoplado com leitora/perfuradora de fita de papel e tem as seguintes características:

a) Teletipo

Velocidade de impressão de 10 Car/seg.

72 caracteres por linha

b) Leitora/perfuradora de fita de papel

Velocidade leitura/perfuração 10 car/seg.

Largura da fita 25 mm

3) Disco Magnético (móvel), com as seguintes características:

- 2 cabeças

- Tempo Medio de posicionamento das cabeças 38ms

- Tempo Medio de acesso 50,5 ms

- Velocidade de transferência 312 Ko/s

- Número de trilhas/face 400

- Número setores por trilha 24

- Número bytes por setor 256

- Capacidade útil do disco 4.915.200 bytes.

4) Leitora de Cartões mod CII-LC 300, com capacidade de 300 cpm.

- 5) Terminal de Vídeo, modelo CII - IRISCOPE 200; tipo de transmissão assíncrona por carater; modo de transmissão em Half-duplex; velocidade de Transmissão de 110, 200 e 300 bauds; capacidade de 1280 caracteres (16x80); Velocidade máxima de 30 caracteres/segundo.

CAPÍTULO II2. FACILIDADES APRESENTADAS POR DIVERSOS TIPOS DE EDITORES DE TEXTOS

O volume de informações apresentadas neste capítulo sobre este ou aquele Editor de Textos é apenas função da bibliografia encontrada [2].

Os seguintes editores são comentados:

- a) Wylbur
- b) Conversational Context-Directed Editor
- c) Quick Editor (QED)
- e) TVEDIT
- f) Interactive Programming Support Systems.
- g) Editpak
- h) Edit - 10
- i) Magnetic Tape Selectric Typewriter
- j) Astrotype
- k) SCRIPT
- l) Edit/360
- m) Hypertext Editing System (HES)
- n) System/360 Administrative Terminal System (ATS)
- o) VIPCOMM

## 2.1. WYLBUR

O Editor de Text Wylbur foi desenvolvido pela Stanford Computation Center, implantado no IBM-360/67 e é operado através do terminal IBM-2741. Este sistema é orientado para uma linha, equivalente a um cartão e, normalmente para programas, usa no máximo 72 caracteres.

É possível a edição de Textos-livres, desde que a formatação seja feita pelo usuário. A edição pode ser feita em letras minúsculas e maiúsculas. A procura de cadeias de reconhecimento foi introduzido neste sistema, que também possui um conjunto completo de comandos que permite alteração dos textos.

## 2.2. CONVERSATIONAL CONTEXT-DIRECTED EDITOR

O Conversational Context-Directed Editor foi desenvolvido pela IBM Cambridge Scientific Center para implantação no IBM-360/67. É operado através dos terminais IBM-2741. Além de programas, onde é considerado um registro fixo de 80 caracteres. Permite a edição de textos livres. Dentro de uma linha é possível fazer a substituição de um string de caracteres por um de tamanho igual, menor ou maior.

Tem a facilidade de procura de cadeias de reconhecimento, com a restrição de que a cadeia deva estar contida inteiramente na linha. Permite também, avançar ou retroceder n linhas e

tabulação para a edição de programas com formato fixo de entrada.

### 2.3. TEXT EDITOR AND CORRECTOR (TECO)

O TECO foi desenvolvido pelo Massachusetts Institute of Technology and Project MAC para o Digital Equipment Corporation-PDP. Este sistema é voltado principalmente para a edição de programas usando como periféricos de entrada e saída, qualquer um dos dispositivos do sistema PDP-10. A edição de Textos livres é feita, desde que a formatação seja dada pelo usuário. Possui um conjunto completo de comandos que permite atualização do texto.

É orientado para uma página de n linhas que para ser alterada deve estar num buffer na memória.

O conceito de "macros" ou "functions", armazenando sequência de comandos que devem ser repetidos regularmente, foi introduzido. Pode ser considerado um sistema de difícil manuseio pois todas as operações são feitas a nível de apontadores que se situa entre palavras e executados à esquerda ou à direita da posição indicada pelo apontador e sobre um número de caracteres pré-estabelecidos.

### 2.4. Quick Editor

O sistema QED foi originalmente desenvolvido pela University of California em Berkeley, para seu sistema de Time-Sha-

ring, sendo implantado em um computador SDS-930. É orientado para uma linha de 80 caracteres. A entrada e saída do sistema é feita somente através de teletipo. Podem ser feitas edições simples em que o formato é dado pelo usuário.

Possue um conjunto de comandos que agem sobre uma linha e não sobre cadeias. Existe facilidade para alteração de caracteres em uma linha presente na memória.

## 2.5. TVEDIT

Desenvolvido pela Stanford University é um dos mais antigos editores utilizando terminal de video (1965). É orientado para uma linha de 80 caracteres e tem ponteiros que se deslocam sobre a tela, em qualquer direção. O texto é dividido em páginas de tamanho igual ao da tela do terminal. Possui comandos simples e as operações são diretamente efetuadas sobre o texto, não existindo facilidades para formatação.

## 2.6. INTERATIVE PROGRAMMING SUPPORT SYSTEM

Desenvolvido pela System Development Corporation, é operado pela IBM 2260, fazendo o uso máximo de sua limitada capacidade.

É um sistema que possui bastante recursos, incluín-



do em seu repertório inserções, deleções, substituições de linhas. As linhas são numeradas e a referência deve ser feita ao número da linha, devido haver correlação entre os números das linhas sobre a tela do vídeo e as linha do Arquivo.

## 2.7. EDIT-10

Desenvolvido pela Interactive Science Corp., o sistema EDIT-10, é um editor de programas para uso num computador PDP-10. A entrada é feita através de teletipo e a saída pode ser feita através do teletipo ou por impressora de linhas. A edição pode ser feita em letras minúsculas e maiúsculas e na edição de textos livres a formatação é dada pelo usuário.

Este sistema é de simples operação, possuindo comandos completos para atualização de textos. Pode utilizar-se do recurso de "Cadeias de reconhecimento" para localizar pontos do seu programa. A alteração de caracteres dentro de uma linha é permitido.

## 2.8. MAGNETIC TAPE SELECTRIC TYPE-WRITES (MTST) E O ASTROTYPE

O MTST foi desenvolvido pela IBM Corporation e o Astrotype pela Information Control Systems e, são sistemas similares e isolados de computador (off-line). O MTST consiste de uma u

nica máquina de escrever IBM Selestric Typewriter, conectado por uma unidade de controle e uma unidade de fita.

2.9. O Astrotype System consiste de quatro máquinas de escrever e é também ligado por uma unidade de controle a uma unidade de fita magnética (DEC-8). O texto é entrado pela máquina de escrever já formatado, sendo gravado na fita magnética com controle do tipo "fim de linha", "fim de página", "tabulação", etc.

Caracteres isolados podem ser substituídos, todavia, cadeias de caracteres só podem ser substituídos por outros do mesmo tamanho. É permitido a remoção e inserção de uma linha integral.

A edição do texto é feita simplesmente descarregando a fita magnética onde foi armazenado o texto com os caracteres de controle.

## 2.10. EDITPAK

O Sistema Editpak foi desenvolvido pela Scientific Time Sharing Corp., adotando-se para sua programação o APL/360.

Este sistema edita somente textos livres, não sendo viável a edição do programa. O usuário do sistema está totalmente desvinculado do mesmo e a forma como o sistema trata e armazena os textos lhe é totalmente invisível. O acesso é feito por meio de cadeias de reconhecimento. As entradas e saídas do Texto fonte só pode ser feitas através de um terminal IBM-2741.

A operação do sistema não exige do usuário conhecimento de processamento de dados.

Não há procedimento previsto para a recuperação de erros e arquivos.

## 2.11. SCRIPT

O Sistema SCRIPT é um editor de Textos Livre da Brent Ass., implantado num computador PDP-10.

Este sistema é orientado para uma página de n linhas, que deverá ser colocada em um buffer na memória, para que se possa manuseá-la. Neste sistema é tratado a linha e não uma cadeia de caracteres. Todavia, existe um comando que permite substituir uma cadeia de caracteres contida em uma linha, desde que a alteração não ultrapasse o limite dessa linha.

No SCRIPT, somente os terminais do sistema PDP-10 podem ser utilizados.

Para modificação ou alterações de linhas, um mínimo de dois comandos é necessário. O primeiro traz a linha e o segundo a modifica.

Possue comandos suficientes para todas as funções necessárias na atualização de Texto fonte. Permite edições as mais complexas.

Cada controle de edição ocupa uma linha de entrada, não havendo comando com funções múltiplas. A silabação é possível; não é todavia feita integralmente pelo sistema, necessita o auxí-

lio do usuário.

## 2.12. EDIT/360

O Sistema EDIT/360 foi desenvolvido pela International Business Machine para a produção de entradas para dispositivos de foto composição ou linotipos.

O usuário desconhece a forma com que os textos são tratados e armazenados. A entrada é feita somente pelos terminais do tipo IBM-2741 e a saída é entrada para dispositivos de foto composição e linotipos, podendo ainda utilizar impressora de linha para edição do texto e posterior correção.

A operação deste sistema é bastante complexa para um leigo em edição em que a unidade de medida "paica" e "linha", não seja comumente usados. O conjunto de comandos é completo e poderoso, o mesmo ocorrendo sobre os controles de edição que são por sua vez bastante complexos pois utilizam unidades de medida para a justificação da linha. Quando são utilizados os dispositivos do tipo "foto-composição" e "linotipo" devem ser consideradas as larguras e natureza dos tipos de impressão que serão utilizados. Nestes dispositivos os caracteres ocupam espaços proporcionais ao seu tamanho.

A segurança dos arquivos deve ser efetuada pelos usuários utilizando-se de rotinas de "checkpoint-restart" do sistema/360.

### 2.13. HYPertext EDITING SYSTEM (HES)

Desenvolvido pela Brow University, o HES é um sistema de edição de textos livres que se baseia num terminal de vídeo que contém teclas especiais, uma para cada função a realizar sobre o texto. As funções são executadas com o auxílio de um lápis eletrónico que indica os limites de atuação. Este sistema foi o que introduziu o conceito de "hipertexto", isto é, da independência do usuário quanto à forma de arquivamento do texto fonte.

### 2.14. SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM (ATS) VIPCOM

O Sistema VIPCOM foi desenvolvido pela VIP System e o ATS pela IBM e diferem apenas no dispositivo de saída.

No ATS a saída é feita através de uma impressora de linha do sistema/360 e no VIPCOMM o dispositivo de saída é um equipamento de fotocomposição. As funções executadas por ambos os sistemas são as mesmas que os sistemas MTST e ASTROTYPE executam.

## CAPÍTULO III

### 3. DECIDIBILIDADE E IMPLEMENTAÇÃO

Na análise dos Editores de Programas constante do Capítulo II, observa-se que nada de original foi feito de editor para editor, apenas se introduziu alguns recursos nos antes existentes. Devido a grande diversidade de terminais, não existe sistema de edição de textos universal, pois este é totalmente dependente da configuração dos terminais para os quais são projetados. Como não poderia deixar de ser, o sistema proposto é também um sistema particular, projetado para os equipamentos cujas características estão descritas no Capítulo I.

Objetivando concluir o projeto, funcionalmente, as soluções simples (desde que razoáveis) foram preferidas.

#### 3.1. Introdução ao Sistema

Fazemos a seguir uma breve explanação do funcionamento do sistema, sem entrar em detalhes técnicos, com o objetivo de dar uma visão geral do conjunto que é necessário para o melhor entendimento de especificação mais técnicas, contidas nos parágrafos e capítulos seguintes:

### 3.1.1. Características Principais

A fim de facilitar as operações sobre o texto do usuário foi criado um "buffer" na memória, imagem da tela do terminal de visualização.

Todos os caracteres digitados sobre o vídeo, são colocados no buffer, na mesma linha e coluna. Isto é possível, devido ao perfeito controle da movimentação do cursor sobre a tela do terminal.

As linhas aparecem numeradas para o usuário. A numeração começa no início de cada linha. Para cada linha inserida, uma nova linha é numerada e, todas as alterações feitas sobre o texto, são com referência a esta numeração que possui correlação direta com a ordem lógica em que os registros estão gravados em disco.

Para permitir todas as operações necessárias desde a geração até a preparação para compilação de programas, foi criado um conjunto de comandos de sintaxe simples, de fácil operação e que conserva a mesma notação dos diversos comandos de operações do Sistema Mitra 15. Além da função já citada, outras como: remoção, cópia, definição de arquivos, listagem, etc., existem.

### 3.1.2. Outras características

O Sistema é orientado para uma linha de oitenta ca

racteres, equivalente a um cartão de programa.

Pode editar textos simples de no máximo 72 caracteres por linha, desde que a formatação e silabação seja feita pelo usuário.

Esta restrição é feita devido as características do periférico, usado como impressora de linha (ver capítulo I).

Apenas um comando é necessário para trazer uma linha, trecho ou programa de um arquivo ativado, à memória, após o que, todas as operações e/ou modificações são permitidas, sem a necessidade de um segundo comando.

A substituição de linha ou caracteres é feita pela super-impressão de uma linha integral.

Permite copiar um arquivo integralmente em outro arquivo maior ou igual.

Permite remover todo um arquivo, trecho ou linha com apenas um comando.

Aceita entrada de textos através do terminal de vídeo, leitora de cartões e leitora de fita de papel perfurável.

As listagens dos programas para os usuários, são feitas sobre o teletipo, perfuradora de fita de papel e sobre o vídeo.

Prepara programas em Assembler e Fortran, para compilação.

Desta forma, procuramos inclusive, dar aos usuários vantagens sobre o uso das perfuradoras de cartões. O fato do usuário poder listar e ler o seu programa em fita de papel, no contexto atual, a nosso ver, traz o conveniente da não ocupação definiti



va da área de memória do disco destinada aos usuários, podendo o seu arquivo ser removido logo após a conclusão da tarefa executada pelo usuário.

### 3.2. Catálogo da zona DA

O catálogo da zona DA do disco, destinada aos usuários é criado a partir do console de operações (teletipo) através do comando abaixo, que é executado pelo sistema de geração de arquivos - UGF15.

a) CALL/UGF15/AFFECT/D:Y,L:Z

onde "D:Y" indica o número da unidade de disco e "L:Z" designa o comprimento do catálogo em número de setores. O tamanho do catálogo deve ser  $0 \leq Z \leq 64$ . Além da criação do catálogo do disco, sem a opção "L:Z" este comando faz a associação do UGF15 com o disco do sistema, necessária para que se possa fazer as operações de entrada e saída sobre o disco. Se a opção "L:Z" estiver presente, o catálogo anterior, se houver, é destruído e os arquivos perdidos para sempre.

No catálogo, são gravados pelo subprograma do monitor, entre outras, as seguintes informações:

- Número de arquivos existentes;
- endereço do início do arquivo;
- Nome do arquivo;

- Número da conta do usuário;
- Tamanho do arquivo em setores;
- Endereço da primeira área livre;

Cada arquivo ocupa 18 bytes sobre o catálogo e cada setor contém informações sobre 14 arquivos.

Tendo em vista evitar que o usuário manipulasse com as instruções de afetação, do Utilitário UGF15, esta associação é feita no início do programa executivo do Editor de Programas.

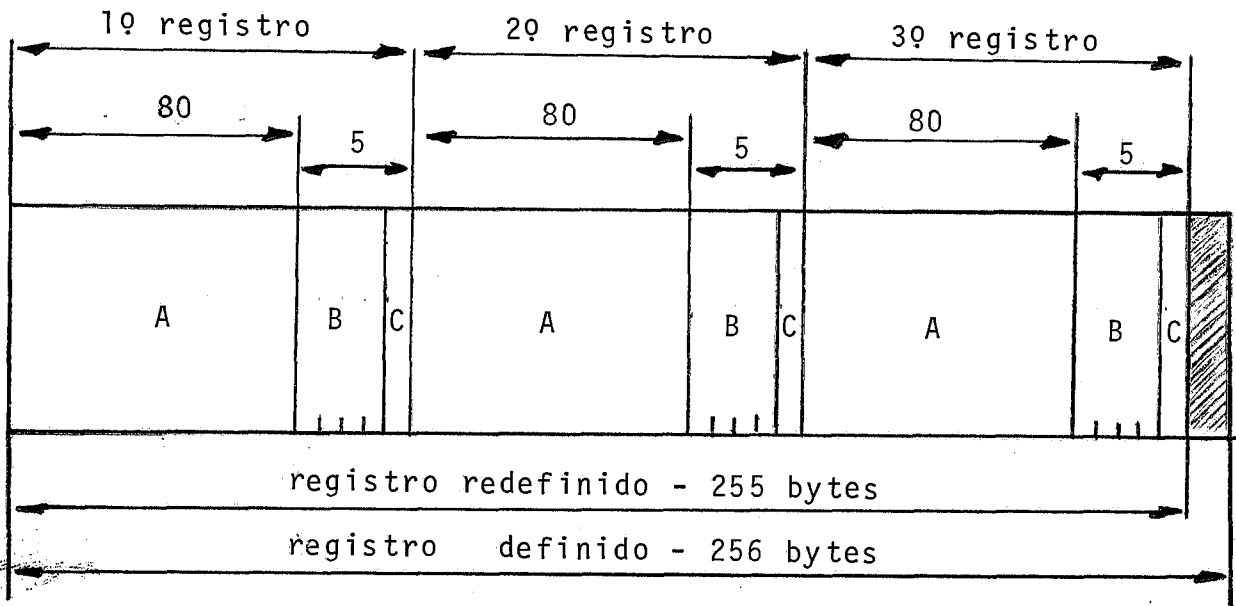
O catálogo criado é de dez setores permitindo a geração de 140 arquivos para a zona DA que possui aproximadamente 9000 setores (pode ser ampliada).

### 3.3. Um sistema de Arquivo em Disco para Acesso aleatório

A organização dos arquivos dada a seguir, foi em função das características do computador MITRA-15, onde o Editor foi implantado.

#### 3.3.1. Arquivo

A área reservada para o arquivo é constituída de registros de 256 bytes cada (1 setor), cuja estrutura é dada a seguir:



Dessa forma, tendo em vista a necessidade de maximizar a utilização da memória em disco e resolver o problema de acesso aleatório não permitido aos registros de um arquivo bloqueado, foi particularmente criado arquivos bloqueados. Assim, teremos redefinidos para os arquivos:

- Registros de 85 bytes
- Fator de grupo = 3
- Página  $3 \times 85 = 255$  bytes

Observem que do registro definido, de 256 bytes, perdemos apenas o último byte de cada registro.

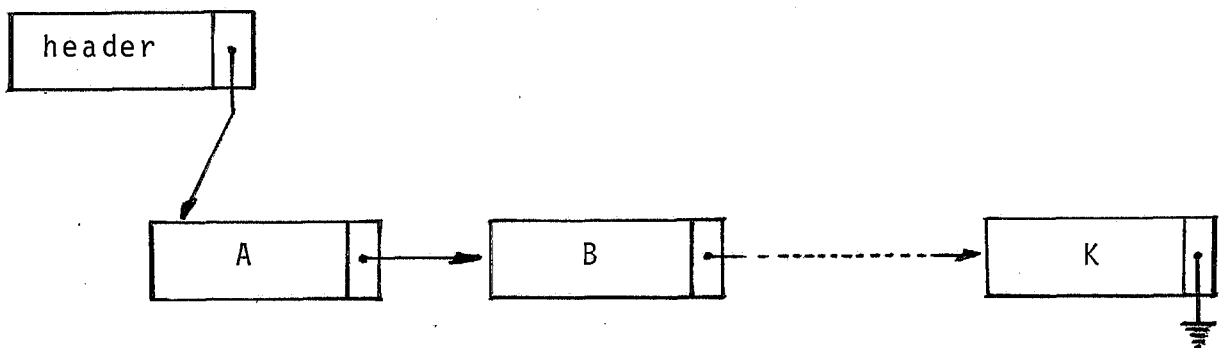
Na área "A" do registro redefinido é gravado a imagem de um cartão ou uma linha de 72 caracteres. As áreas "B" e

"C" de 4 e 1 byte respectivamente, são reservados para apontadores. A primeira contém o endereço do setor, onde está alocado o próximo registro e a segunda contém a ordem de alocação do registro no setor (1º, 2º ou 3º registro). Estes cinco últimos bytes de cada registro são chamados de "contexto".

O setor zero de cada arquivo é o "header", onde são gravadas as seguintes informações:

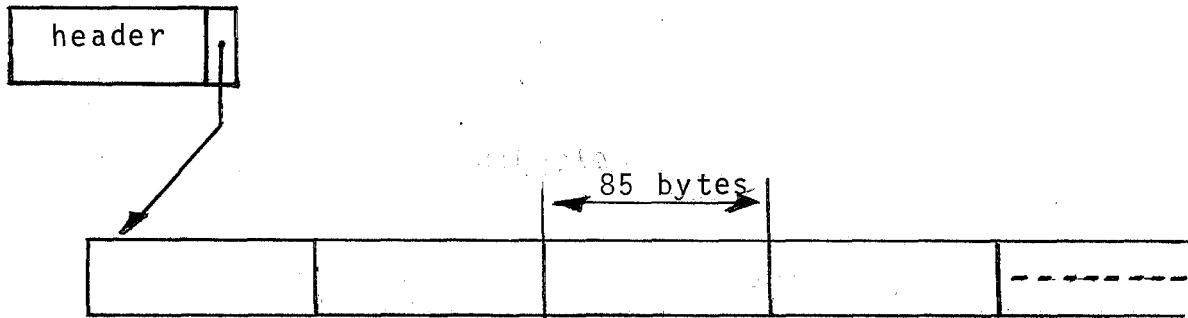
- O primeiro contexto contém o endereço e a ordem do primeiro registro do arquivo.
- O segundo contexto contém o endereço do primeiro registro disponível.
- O terceiro contexto contém o tamanho do arquivo (campo "B"). No campo "C" deste contexto é colocado um flag que informa se o arquivo está vazio ou não.

Os registros que contêm informações formam uma lista linkada, como mostra a figura abaixo:



### 3.3.1.1. Lista de Espaço-Disponível

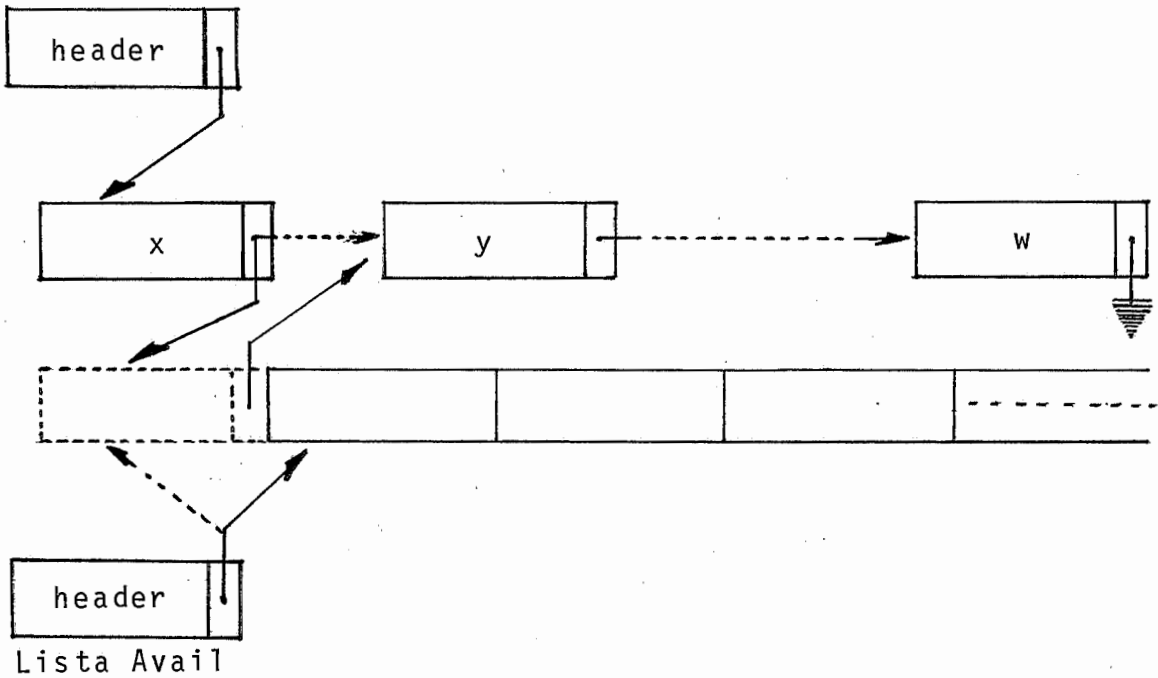
Ao ser gerado o arquivo, todos os registros, exceto o "header", forma uma lista (deque) de espaço disponível:



Um registro deixa de fazer parte de uma lista de espaço disponível quando é requisitado para inserção de uma nova linha. É importante observar que a lista de espaço disponível é sempre fisicamente sequencial. O registro requisitado, conforme mostra a figura a seguir, passa a fazer parte da lista linkada, definida antes. Este novo registro pode ser inserido em qualquer ponto da lista linkada:

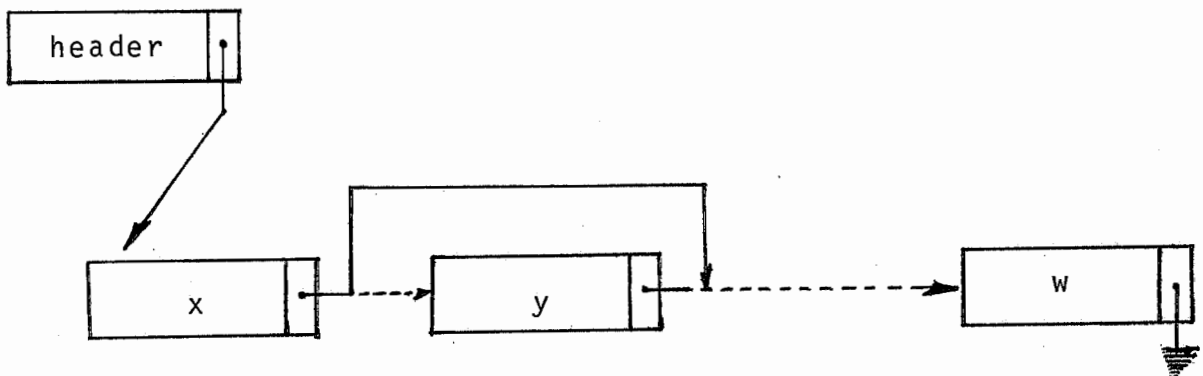
- inserção entre dois nós:

Lista Linkada



- remoção de nós:

Os nós podem ser removidos em qualquer ponto da lista linkada.  
Uma ilustração é dada pela figura abaixo:



### 3.4. LINGUAGEM

A linguagem para este sistema de edição é simples e de fácil memorização, permitindo ao usuário todas as operações necessárias as alterações do seu texto. A sintaxe dos comandos está completamente definida no Anexo I (Manual do Usuário) e, consta dos comandos de:

- a) Definição de Arquivos
- b) Seleção de Arquivos
- c) Remoção
- d) Inserção
- e) Listagem
- f) Cópia
- g) Preparação para Compilação

#### 3.4.1. Definição de Arquivos

É necessário que o usuário possua ou defina um arquivo com as características expostas no parágrafo anterior, antes de utilizar das demais operações. A definição dos arquivos é feita dinamicamente, por programa. As informações usadas pelo comando DEFINA (ver anexo I), é dado pela tabela seguinte:

0-1	NOME DO ARQUIVO	
2-3		
4-5		
6-7	FATOR DE GRUPO	TAMANHO DO REGISTRO (Palavras)
8-9	CONFIGURAÇÃO	Nº CONTA
9-10	TAMANHO DO ARQUIVO (EM SETORES)	
10-11	ZERO	Nº DO DISCO
11-12	RESERVADO	

FIG. 1

## Configuração:

BITS	0	1	2	3	4	5	6	7
0	INPUT	0	LIVRE	BINÁRIO	BLOCADO			
1	OUTPUT	0	PROTE- GIDO	ALPHANUM	NÃO BLO- CADO	N		

A fim de manter uma padronização sobre os arquivos o usuário apenas necessita fornecer, na definição do seu arquivo, os seguintes parâmetros:

- 1: Nome do arquivo
- 2: Número da conta
- 3: Número de setores



Os demais parâmetros são fixos e definidos para todos os arquivos como:

- a) Fator Grupo = 1 (para o sistema)
- b) Registro fixo de 256 bytes (1 setor)

A configuração assumida é a seguinte:

```

0 . . . . . arquivo protegido para leitura
. 0 . . . . . não importa
. . 0 . . . . . arquivo livre
. . . 1 . . . . . alfanumérico
. . . . 1 . . . . não bloqueado
. . . . . 0 0 0 sem significação

```

Os dados constantes da tabela FIG.1, vão constituir no "Control Block" de uma subrotina assembler, a "File Table".

A chamada ao Supervisor para criação do arquivo, é feita como abaixo:

```

Acumulador = endereço da Tabela do Arquivo
CSV M:FILE (chamada ao supervisor)

```

Após abertura do arquivo, o setor 0 (header), é inicializado com as seguintes informações:

- a) No primeiro contexto (campo "C") é gravado um

flag que indica fim de arquivo.

- b) O segundo contexto contém o endereço do 1º registro livre.
- c) No terceiro contexto, são gravados: tamanho do arquivo (Campo "B") e um flag que indica arquivo vazio.

### 3.4.2. Seleção de Arquivos

Para que se tenha acesso a qualquer dos arquivos existentes no Catálogo da zona DA do disco é necessário que o mesmo esteja associado a uma das etiquetas operacionais da Tabela abaixo:

NOME	NÚMERO	BINÁRIO	FUNÇÃO	MODOS
M:EI	5	& 05	ENTRADA DE ELEMENTOS	BN ou AN
M:EO	6	& 06	SAÍDA DE ELEMENTOS	BN ou AN
M:SI	10	& 0A	ENTRADA DO FONTE	AN
M:LO	7	& 07	SAÍDA DA LISTAGEM	AN

FIG. 2

Além das etiquetas acima, existem outras que, no entanto, não são utilizadas pelo Editor de Programas.

Uma mesma etiqueta não pode estar associada a mais de um arquivo ou a um arquivo e a um periférico ao mesmo tempo.

Desta forma, para que se faça I/O do arquivo em disco para um periférico, é necessário o uso de duas etiquetas operacionais.

A associação aos arquivos é feita dinamicamente através do comando ATIVE, definido no anexo I. Os dados constantes da tabela (FIG.3), são necessários para as associações.

0-1	M:XX	CONFIGURAÇÃO	} PARTE COMUM COM O PERIFÉRICO
2-3	NOME EBCDIC DO PERIFÉRICO		
4-5	ZERO	Nº PERIFÉRICO	
6-7	UEM	Nº CONTA	
8-9	NOME DO ARQUIVO		} EBCDIC
10-11			
12-13			

FIG. 3

onde M:XX - é uma das etiquetas operacionais da tabela FIG. 2

Nome EBCDIC - 2 letras características do periférico, TY (teletipo), PT(fita perfurável), DM(disco) CR(leitora)

UEM - Nº da Unidade de tratamento

A configuração é dada por:

BIT	0	1	2	3	4	5	6	7
0	INPUT	SEQUENC.	BACK ROUND	BN	PERIFÉ RICO	STOL	IN ou OUT	0
1	OUTPUT	DIRETO	TEMPO REAL	AN	ARQUIVO	ASGN	RW	0

As seguintes configurações são usadas:

a) Associação de M:EI ou M:SI sobre um arquivo

0 . . . . . 0 ou 1 não importa se o bit nº 6  
é IN ou OUT

. 1 . . . . . Acesso direto aos setores do disco

. . 0 . . . . . Usa a zona Background

. . . 1 . . . . . Alfanumérico

. . . . 1 . . . . Associação a arquivo

. . . . . 1 . . . . Associação

. . . . . 0 . . . . INPUT ou OUTPUT

. . . . . 0 . . . . não analisado

Para associação das etiquetas operacionais com periféricos é necessário apenas inverter o bit nº 4. Também, para este caso, o nome EBCDIC do Arquivo é desnecessário.

O usuário deverá guardar tanto o nome do seu arqui

vo como o número da conta, pois, sem os dois é impossível ativar qualquer dos arquivos existentes no catálogo.

A tentativa de associar, um arquivo cujo nome ou número de conta não seja o constante do catálogo, é detetado pelo sistema, carregado no acumulador um flag que é transmitido ao usuário em forma de erro pelo Editor de Programas. O comando ATIVE é o segundo comando a ser dado quando do início das operações com o Editor de Programas, caso o usuário ainda não possua arquivo definido. Caso contrário, será o primeiro (Ver anexo I).

A chamada ao supervisor para que a associação seja feita é da forma:

Acumulador = endereço da tabela de dados  
CSV M:ASGN (chamada ao Supervisor)

### 3.4.3. Remoção

As operações de remoção são feitas através do comando REMOVA, que oferece aos usuários as seguintes opções:

- a) Remoção de uma linha;
- b) Remoção entre linhas (inclusive)
- c) Remoção de todo um arquivo.

A sintaxe deste comando está definida no anexo I.

Como já dissemos, as linhas aparecem na tela do terminal, numerados para o usuário e a remoção é feita com referência a esta numeração que possui correlação direta com a ordem em que os registros estão logicamente gravados em disco. Considerando a estrutura do arquivo (ver item 3.3), os nós removidos não são devolvidos a lista avail. Para resolver o problema dos espaços perdidos em um arquivo após um grande número de remoções, foi criado através do comando COPIE (Ver item 3.4.6.) a possibilidade de compactação.

Quanto a remoção de todo um arquivo é um pouco mais complicada e consiste em apagar o nome, do catálogo da Zona DA. A exemplo do subprograma do Monitor, a atualização do endereço da área disponível no catálogo só é feita quando o arquivo removido for o último do catálogo. Em todos os casos, o contador de número de arquivos do catálogo é decrementado de um.

Deve-se observar, inclusive pela sintaxe do Comando que, o arquivo removido será sempre o ativado (Veja o item 3.4.6.).

#### 3.4.4. Inserções

As inserções de linhas são feitas através do Comando INSIRA, definido no anexo I. Este comando permite ao usuário várias opções:

- a) Inserção através do terminal de vídeo

- b) Inserção através de leitora de fita de papel
- c) Inserção através da leitora de cartões

As inserções através da leitora de fita de papel e leitora de cartões são feitas sequencialmente, registro, por registro, a partir do início do arquivo.

As inserções através do terminal de vídeo possui características especiais: elas podem ser feitas a partir de qualquer ponto do programa. Após o comando, aparecerá na tela do vídeo, a numeração da linha no ponto onde o usuário começará as inserções.

Para cada linha, a inserção é feita quando o usuário, movimentando o cursor, sair da linha onde foi digitado pelo menos um caractere (diferente dos caracteres de controle).

Os endereços das linhas que se encontram na tela do terminal, são guardados numa tabela de apontadores facilitando as alterações do texto, através do acesso direto. Para cada linha inserida, uma nova linha é numerada. Ao fim de cada inserção o usuário deverá aguardar o sinal da "campainha" antes de começar a movimentar o cursor sobre a tela.

O sucesso das alterações sobre o texto é consequência do perfeito controle da posição do cursor sobre a tela do terminal.

Para alterar o texto que se encontra sobre a tela, o usuário apenas necessita movimentar o cursor até a posição desejada e fazer as alterações necessárias. O fim das alterações é detetado do mesmo modo, com as mesmas implicações.

A inserção é um pouco demorada e notável, se inicialmente ela começar a partir de um registro "N" (grande).

Neste caso a busca do citado registro é feita sequencialmente (não fisicamente), nos seguintes passos:

- Lê o registro seguinte.
- Retira do contexto, o endereço do próximo registro.
- Verifica se é fim de arquivo. Se sim, entra em Wait a espera de novo comando.
- Compara com N. Se não for igual, volta ao primeiro passo.

As inserções a partir daí, são diretas.

#### 3.4.4.1. Overflow

Para todos os casos o overflow é detectado através da comparação do tamanho do arquivo definido e colocado no terceiro contexto, com o setor onde o próximo registro de cada arquivo irá ser inserido.

No último registro de cada arquivo não é permitido inserção. A detecção do overflow, é transmitida ao usuário em forma de ERRO.



### 3.4.4.2. I/O Sobre o disco

Além da necessidade de associarmos etiquetas operacionais ao arquivo com o qual queremos trabalhar, depois das associações e antes de fazer I/O sobre o disco é também necessário que se faça abertura ("OPEN") deste arquivo, sob a mesma etiqueta operacional usada para Entrada dos dados. A abertura é feita dinamicamente, por programa.

A chamada ao supervisor para abertura do arquivo é da forma:

Acumulador = qualquer

Registro E = M:XX (etiqueta operacional)

CSV M: OPEN (chamada ao supervisor)

Após o "OPEN", o acesso ao arquivo é tornado possível. Considerando arquivos de acesso direto não bloqueado, nesse caso, a tabela de dados para entrada/saída é dada como:

	bits				
	0	7	8	10	15
0-1				1	
2-3	ORDEM (I/O)		M:XX		
4-5	endereço do buffer				
6-7	nº de bytes				
8-9	endereço para desvio (erro)				
10-11	endereço do disco, início do Arquivo				

onde ORDEM: 80 Se INPUT e Ø se OUTPUT. Os dois primeiros bytes, indicam a existência de informações suplementares (se o bit nº 10 esta ligado). Os dois últimos bytes é a informação suplementar.

No fim de cada I/O, o fechamento do arquivo é necessário, sob a mesma etiqueta operacional.

A chamada ao supervisor para fechamento do arquivo é da forma:

Acumulador = M:XX (etiç. Operacional)

CSV M:CLOS (Chamada ao supervisor)

#### 3.4.5. Listagem do fonte

A listagem dos textos é feita pelo comando LISTE de finido no anexo I. Este comando da diversas possibilidades ao usuário tais como:

- a) listar apenas uma linha
- b) listar várias linhas
- c) listar todo o arquivo
- d) listar os nomes dos arquivos do Catálogo

As opções "a", "b" e "c" são possíveis no teletipo, terminal de vídeo e perfuradora de fita de papel. A opção "d" só é possível no terminal de vídeo.

A listagem sobre o vídeo possui características especiais. No fim de cada LISTE, é provocado um desvio para o subprograma de inserção, permitindo ao usuário movimentar o cursor até a posição desejada e fazer alterações em seu texto.

Caso o texto a listar possua mais que 15 linhas, a interrupção ocorrerá de 15 em 15 linhas. O fim da interrupção é detetado com o recebimento do carácter de controle "Control + Bell", após o que, continuará a listagem de linhas, se houver. Caso o usuário queira descontinuar a listagem sobre o vídeo, deverá aguardar a próxima interrupção e então acionar a tecla "ESCAPE" que provocará o aborto da listagem. O usuário não precisa se preocupar com o posicionamento do cursor sobre a tela no fim de uma alteração. Após ser acionada uma das teclas acima referidas, o Editor posicionará o cursor na primeira coluna a esquerda da linha inferior.

Por motivo de segurança, não é feita a listagem dos números de conta sobre a tela do vídeo, quando da listagem do catálogo.

#### 3.4.5.1. Associações sobre PT

Antes da listagem ser feita sobre a perfuradora de fita de papel, é feita associação das etiquetas M:EI ao arquivo em disco e M:LO a perfuradora de fita de papel (PT).

### 3.4.6. Cópia de Arquivos

Tendo em vista resolver os problemas de OVERFLOW em arquivos, foi criado o comando COPIE (ver anexo I), que permite ao usuário principalmente nestes casos, mover todo o seu arquivo para um arquivo maior.

A cópia do arquivo ativado para o arquivo definido no comando COPIE é feita sequencialmente, permitindo a compactação desse arquivo. Não são nos casos de overflow, como após um grande número de remoções de linhas de um arquivo qualquer, aconselhamos copiar este em um novo arquivo. Em todos os casos, após o COPIE, o arquivo velho deverá ser removido (ver item 3.2.3). A fim de facilitar esta operação (remoção), o arquivo ativado continuará sendo o velho arquivo. Desta forma, o usuário apenas necessita teclar o comando REMOVA e em seguida ativar o seu novo arquivo.

### 3.4.7. Preparação para Compilação

Como já vimos, ao usuário será permitido programar em Assembler e Fortran através do Editor de Programa. O Compilador Fortran do MITRA-15 é dirigido para leitura de cartões de 80 colunas. Todavia, no Control-block de leitura deste compilador, possui uma palavra reservada a informação "suplementar".

A informação suplementar como já vimos, é o endereço do setor, início do arquivo, a partir de onde, após as associa-

ções devidas, o compilador começará a leitura dos registros. O fim da leitura é detetado após encontrar um "%EOD". Neste arquivo é gravado apenas um registro de 80 caracteres por setor.

Nenhuma alteração é necessária para que seja possível ao Compilador Assembler ler arquivo em disco (zona DA). Neste caso, apenas são feitas associações utilizando as informações constantes da tabela da FIG. 3 (item 3.4.2).

A preparação para compilação é feita após o Comando EXEC e consiste em copiar sequencialmente o arquivo do usuário no arquivo de trabalho do Editor.

É interessante observar, que a alteração feita no "Control-Block" do Compilador Fortran, em nada afetou as operações que este executava antes, ou seja, a leitura de cartões, obviamente, continuam sendo feitas.

As instruções para execução do seu programa, após o comando EXEC, estão contidas na penúltima página do Anexo I.

#### 3.4.7.1. Arquivo de Trabalho do Editor de Programas

Devido a informação suplementar colocada no Control-block do Compilador Fortran, para permitir a leitura de programas a partir do disco do sistema, o arquivo de trabalho do Editor de Programa, ARQEDI, onde, como já vimos, é feita a cópia dos arquivos em formato aceitável pelo Compiladores Fortran e Assembler, será sempre o primeiro arquivo sobre a zona DA. A definição desse

arquivo é feita a partir do Teletipo de controle usando o comando de definição de arquivos do subprograma do Monitor e, será criado sempre que um novo catálogo for aberto:

- % DFILE/N: ARQEDI,K:Ø , AN, L:1250, R:80

onde:      K:Ø      - número da conta;  
            L:1250 - Tamanho do arquivo em setores  
            R:80     - registro de 80 caracteres  
            AN      - alpha-numérico

Devido a necessidade de se efetuar todas as operações necessárias ao acesso a disco, o nome "ARQEDI" bem como o número da conta desse arquivo, não pode sofrer alterações.

### 3.5. Linguagem Usada na Implementação

O Editor de Programa foi implementado em Fortran e Assembler. A escolha do Fortran como linguagem principal foi motivada pela facilidade com que uma linguagem de alto nível permite análises e tomadas de decisões. Além disso o Fortran do MITRA-15, permite a conexão com a linguagem assembler do sistema, na qual são feitas todas as subrotinas de Entrada/Saída. Um exemplo de passagem de parâmetros de um programa fortran para uma subrotina assembler, é dado abaixo:

```

---
---
CALL    ESDISC (FLAG, BUFFER, NSECT, EREG)
-----
-----
LDS     LDS
        :
        :
TARG    EQU    $ * TABELA ENDEREÇO ARGUMENTOS
FLAG    RES    2
BUFFER  RES    2
NSECT   RES    2
EREG    RES    2
        :
        :
        FIN

```

} endereço(relativo a G) dos parâmetros

LPS

LPS            LDS

LDE = 16 \*Nº BYTES A MOVER(passagem de endereços)

MVS TARG

LDA @NSECT \* leitura do argumento

.

.

.

.

.

RTS

END



### 3.6. Algumas Características Particulares

A fim de permitir a análise dos caracteres recebidos, transmitidos a partir do terminal de vídeo, a leitura neste terminal é feita carácter por carácter. Como nós já vimos, o Editor de Programas possui uma série de Caracteres de controle, como por exemplo, os de movimentação do cursor sobre a tela, que requerem uma tomada de decisão logo após o seu recebimento.

Seria impraticável e altamente ineficiente se a análise fosse por linha, por exemplo. Esta característica, apesar de não constar em nenhum artigo que faz comentário sobre Editores de Programas, de modo geral, estes Editores a possuem.

Outra característica particular e importante é o fato do modo de transmissão ser um Half-duplex. Sabemos que o controle da movimentação do cursor sobre a tela é fator fundamental nas inserções e alterações de textos, todavia, neste modo de transmissão ao mesmo tempo em que o processador executa uma tarefa o usuário pode desatualizar o cursor, movimentando-o. Para resolver este problema, utilizamos a câmpinha do terminal, acionando-a no fim de cada tarefa (ver anexo I).

Também a leitura em fita de papel perfurável só é possível se a fita tiver sido perfurada através do próprio Editor. Esta restrição vem do fato das características especiais que a perfuração da fita contém. No nosso caso, por exemplo, o fim dos registros é detetado através do carácter de retorno do carro. Além disso, na perfuração da fita, o Editor coloca um flag após o últi

mo registro que serve de comparação para fim de leitura.

Antes da perfuração da fita de papel pelo Editor de Programas, é feita uma filtragem dos brancos, a partir da direita permitindo a perfuração da fita, bastante compactada e de registros de tamanho variável. Esta observação é motivada pelo fato do sistema UTIL do MITRA-15, fazer a perfuração da fita em registros fixos de 80 caracteres.

Como todo sistema Editor, este é também um sistema interativo onde a ocupação do terminal pelo usuário é maior que, por exemplo, ao de uma perfuradora.

Dessa forma, a digitação de um carácter ou de um comando, não é feita mecanicamente.

Assim é que, o número de caracteres transmitidos pelo usuário é cerca de 2 por segundo, enquanto que o MITRA-15 pode executar aproximadamente 300.000 instruções por segundo. De um modo geral, o tempo de resposta é considerado bom e imperceptível para o usuário.

## CAPÍTULO IV

### 4. PROBLEMAS DA IMPLEMENTAÇÃO

Neste capítulo, abordaremos somente os principais problemas encontrados quando da implementação do Editor de Programas no Mini-Computador Mitra-15.

#### 4.1. Memória Real

A disponibilidade de memória foi um problema crítico encontrado. A fim de permitir a execução do programa executivo, foi necessária a geração de um monitor menor de ... K bytes. A geração desse monitor (MTRDA4), foi possível com os recursos existentes. A necessidade de incluir nesse monitor alguns recursos, tais como: UFG15, linhas assíncronas, etc., tornou-o suficientemente grande ao ponto de não ser possível executar um programa fortran com o citado monitor. Isto implica na necessidade de carregar sempre um novo monitor (MTRDØ3), quando for necessário executar programas escritos em fortran.

O Editor de Programa, ocupa cerca de 8 K bytes em memória e apesar da geração do novo monitor, foi necessário montar o Editor em uma estrutura de OVERLAY.

A árvore a seguir, dá uma idéia razoável dessa estrutura. A percentagem de subprogramas em Fortran é ligeiramente maior que em Assembler. No entanto, os subprogramas mais importantes e também maiores estão em Fortran.

## EDITOR

LISTE (\*240)

MOVCUR(\*22A)

ESDISC(+1B2)

MOVEL (\*124)

SEGUIR(\* FE)

ERRO (\* FS)

AGRUPA(\* AC)

CVIDEO(+ 94)

ELCIMA(\* 88)

CBNDC (+ 60)

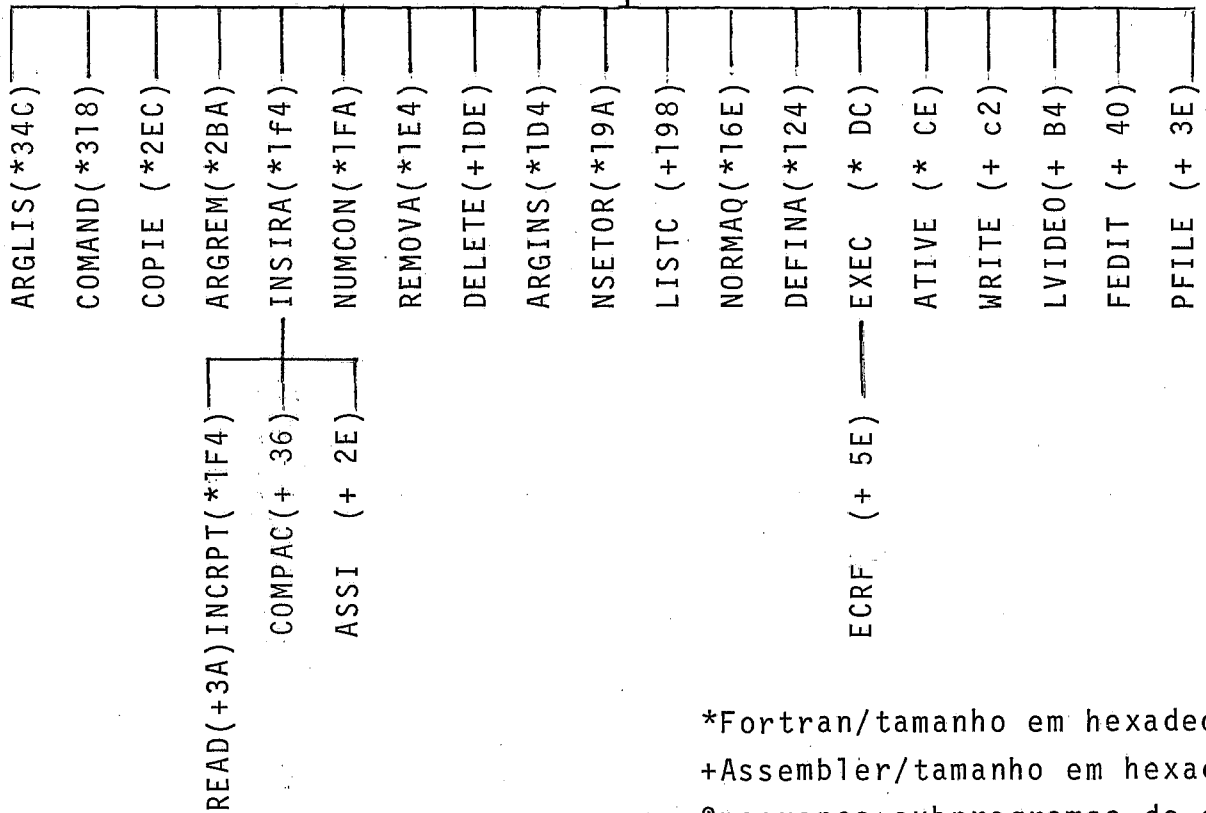
PASGN (+ 4A)

LER (+ 40)

ESCR (+ 3E)

ENDSEG(+ 3C)

⊕



\*Fortran/tamanho em hexadecimal

+Assembler/tamanho em hexadecimal

⊕pequenos subprogramas do sistema

#### 4.1.1. Memória Auxiliar

Apesar de sabermos da impossibilidade de se manter sob qualquer forma, em arquivos do próprio sistema, todos os programas de todos os usuários, a memória em disco, destinada aos usuários é também um problema crítico, pois, dispomos de apenas 9000 setores disponíveis na zona DA. Portanto, a maioria dos usuários, deverá manter seus próprios programas de alguma forma, fora do sistema.

A fita de papel perfurável é a única opção que podemos dar aos usuários, uma vez que não dispomos, por exemplo, de fita magnética.

#### 4.2. Sistema de Acesso a Disco

No sistema Mitra-15 o acesso ao disco só é permitido em dois modos:

- a) Acesso direto aos setores do disco.
- b) Acesso sequencial aos registros.

Significa que para um arquivo bloqueado o acesso é sequencial somente. Logo a necessidade de se ter arquivos bloqueados e modo de acesso direto, fez com que a estrutura do arquivo mostrada no capítulo III, fosse bastante particular, diminuindo in

clusive, a eficiência do sistema Editor. Todavia, a estrutura apresentada, em nosso sistema, ainda é preferível ao modo de acesso sequencial.

#### 4.3. Alteração no Control-block do Compilador Fortran e Fixação do Tamanho do Catálogo

A fim de permitir a leitura de um programa fortran em disco pelo subprograma de I/O desse compilador, foi necessário fazer alteração no seu bloco de leitura, gravando definitivamente em disco na palavra destinada a informação suplementar, o endereço do primeiro setor início do arquivo, a partir de onde o Compilador Fortran iniciará a leitura do programa.

Este arquivo, é o ARQEDI, definido a partir do teletipo de controle, na implantação do Editor. Acontece que a gravação da informação suplementar, implica na fixação do tamanho do catálogo, pois, o arquivo de trabalho do Editor de Programas, é sempre o primeiro arquivo do catálogo e em consequência se inicia logo após o fim deste.

Qualquer alteração a ser feita quanto ao comprimento do catálogo, implica em alterar o endereço do início do arquivo constante no Control-block do compilador.

Considerando a estrutura do arquivo do sistema Editor e o sistema de segurança que envolve a alteração em disco, de programas do sistema Mitra-15, é totalmente impossível ler diretamente dos arquivos dos usuários, que possuem obviamente, endereço

de alocação variável.



CAPÍTULO V5. CONCLUSÕES

Considerando as dificuldades encontradas principalmente pelas limitações dos recursos dos equipamentos utilizados, os objetivos do projeto do Editor de Programas foram totalmente alcançados, já estando o referido Editor em pleno funcionamento no Laboratório de Sistemas da COPPE.

Como já havíamos dito, as soluções simples e razoáveis foram preferidas visando não tornar o Editor de Programas complexo e de difícil operação. Procuramos dar aos usuários todas as facilidades possíveis, sem comprometer a performance do Programa Editor. Assim o simples fato do usuário poder, ao mesmo tempo em que executa o display do seu texto, fazer as alterações que julgarem necessárias, é um exemplo de facilidade permitida. Um outro exemplo é o conjunto de comandos que possuem os nomes das tarefas que o usuário irá executar: insira, remova, liste, define, etc, que implica, inclusive, na fácil memorização desses comandos.

O fato dos usuários poderem guardar os seus programas, fora do sistema, em fita de papel perfurável, além de ser econômico, pois evita o uso de cartões, é bastante cômodo já que o volume é bem menor em consequência, inclusive, pela maneira com

pactada com que a fita é perfurada.

A operação do Editor de Programas é feita através do terminal de vídeo IRIS-200 e não requer dos usuários outros conhecimentos que não os constantes do Manual do Usuário.

Não temos dúvida da utilidade deste trabalho para os usuários do sistema Mitra-15.

Visando tornar o Editor de Programas bem mais útil e versátil, algumas sugestões, neste ponto, podem ser dadas:

- a) Devido a sua natureza interativa, o tempo de ocupação de cada Terminal é superior ao tempo de operação de uma perfuradora pelo mesmo usuário, resultando em um investimento equipamento/ usuário elevado. No entanto, o uso de apenas um terminal ocupando o sistema, subutiliza os recursos da Máquina, tornando necessário modificar o Editor para que seja possível a diversos usuários, utilizá-lo ao mesmo tempo. Obviamente, isto implica na aquisição de mais terminais semelhantes aos existentes.
- b) Devido a disponibilidade de memória geral, seria aconselhável que a grande parte dos algoritmos programados em Fortran fossem traduzidos para o Assembler. Isto, inclusive, eliminaria as subrotinas de compactação e descompactação utilizadas, já que, a nível de Fortran, os caracteres são alocados por palavra.
- c) Prover meios para que seja possível a utilização do modo de transmissão de caracteres em Full-duplex.

- d) É possível permitir aos usuários programar, além do Fortran e Assembler, em LP15 sem que nenhuma alteração seja feita no Editor de Programas. Para tanto é necessário apenas gerar um novo monitor que possua o UFG15 e que seja menor que os existentes. A geração desse novo monitor é uma sugestão importante.
- e) Seria interessante que uma interface ou dispositivo que permitisse ao usuário perfurar em cartões o seu arquivo, fosse feita. Já pensando nesta possibilidade, foi reservado na subrotina "COMAND", que faz análise dos comandos, seis palavras destinadas a criação de uma nova instrução para perfuração de cartões e inclusive a análise dessa instrução pelo programa principal, foi considerada.

BIBLIOGRAFIA

- [1] Arthur J. Benjamin, "An Extensible Editor for a Small Machine With Disk Storage". Comm. ACM (August 1972), pp.742-747.
- [2] Andries Van Dam & David E. Rice, "On-Line Text Editing". Computing Surveys (September 1971), pp.93-115.
- [3] L. Peter Deutsch & Butler W. Lampson, "An On-Line Editor". Comm. ACM (December 1967), pp.793-803.
- [4] James Martin, "Design of Real-Time Computer Systems", Prentice-Hall, Inc.(1967): Capítulo 7 - The Man - Machine Interface (76-86); Capítulo 8 - Conversations with a Computer (91-118).
- [5] Paulo Mario Bianchi, "Sistema Preparador de Textos e Programas - Pretexto". Tese Mestrado COPPE(junho 1974), pp.5-70.
- [6] Enio Antonio Garbini, "Editor de Textos". Tese Mestrado COPPE(outubro 1973), pp.4-71.
- [7] CII - Thompson, "Manual de Referência de Linguagens do Mitra-15 : ASSEMBLER/FORTRAN". 1975. Volumes 1 e 2.

[8] CII - Thompson, "Manual de Utilização MTRD". Volume 3, pp.1-108.

ANEXO I

Os comandos entre  $\{ \}$  são necessários, todavia podem ser substituídos pelos comandos entre  $[ ]$ .

## MANUAL DO USUÁRIO

### 1. GENERALIDADES

O Editor Conversacional Mitra 15, trabalha exclusivamente com o terminal de vídeo Iriscope 200 da CII. Este vídeo possui 16 linhas de 80 caracteres e a velocidade de transmissão de caracteres está entre 110 e 300 bauds.

Devido a disponibilidade de memória, foi necessária a geração de um novo monitor, o MTRDA4, que possui, unicamente, os recursos necessários a utilização do Editor, tais como:

- Linhas assíncronas
- UGF15 - Sistemas de geração de Arquivo para o Mitra 15, etc.

OBS: Além da sintaxe dos comandos os usuários deverão ler com atenção os itens referentes as funções dos comandos.

### 2. FUNÇÕES DAS TECLAS

2.1. As teclas ↑, ↓, →, ← alteram a posição do cursor sob a tela do vídeo. O cursor não pode sair da tela. Na tentativa de sair pela

esquerda o cursor se posicionará na última coluna da linha mais acima ou da primeira linha, caso o cursor estivesse posicionado na primeira coluna da última linha. Tentando sair pela direita o cursor se posicionará na primeira coluna da linha mais abaixo ou na primeira coluna da primeira linha, caso este estivesse posicionado na última coluna da primeira linha. Neste caso, a primeira linha é movida para cima e a última linha da tela desaparece. Tentando sair pela parte superior da tela, o cursor se posicionará na primeira linha e mesma coluna em que estava. Tentando sair pela parte inferior da tela o cursor se posicionará na primeira coluna da primeira linha. Neste caso a primeira linha é movida para cima e a última linha da tela desaparece. Além dessas funções as teclas acima funcionam como caracteres de controle, após comandos LISTE e INSIRA.

- 2.2. A tecla "Clear" provoca o esvaziamento do Buffer, imagem de tela da console.
- 2.3. A tecla "Line Clear" com o "scrol" desligado não tem função, todavia, a sua utilização não implica em erro para o sistema.
- 2.4. A tecla "Return" provoca o deslocamento do curso para a primeira coluna da linha em que se encontra. Esta tecla também funciona como carácter de controle nos comandos LISTE e INSIRA.



- 2.5. As teclas "Control+Bell" possui função importante: ela provoca o "return", quando apertados após o uso dos comandos LISTE e INSIRA e a desativação do programa do usuário, quando fora destes comandos. Neste último caso, o usuário deverá ativar o seu programa, caso queira utilizá-lo novamente.
- 2.6. A tecla "Control + EOT" provoca o aborto do Editor. Ela só deverá ser utilizada no caso do usuário necessitar de executar o seu programa (ver itens 4.7 e 5.7).
- 2.7. A tecla "ESCAPE" descontinua a listagem do programa no vídeo.
- 2.8. Todas as demais teclas são analisadas como caracteres normais portanto, cuidado .

### 3. DE COMO ATIVAR O EDITOR

- 3.1. Antes de começar a manipular as teclas do console de visualização o usuário deverá tomar as seguintes providências:
- a) Desativar a tecla MA sobre o Mitra-15
  - b) Colocar a chave "zero" na posição "on"
  - c) Reinicializar o sistema, apertando as teclas RAZ, INI e MA

Obs: Neste momento aparecerá sobre o teletype, a palavra "CHARGEMENT/". O usuário deverá continuar acrescentando em seguida:

- d) ME/I:DMØ,UC/(MTRDA4) (RC)
- e) ZERAR o foreground - F0/è& Ø (RC)
- f) Desligar as teclas "Local", "scrol" e ligar a tecla "print on line" sobre o vídeo.
- g) Carregar o editor em memória dando o seguinte comando sobre o teletype: % CALL/EDITOR/

Após este último passo, aparecerá sobre a tecla do vídeo, na parte inferior esquerda, um "?": isto indica que o Editor está pronto para ser usado.

#### Importante:

A não observância da sequência de passos a ser seguida, antes do uso deste sistema, poderá acarretar problemas para o usuário, na utilização do Editor.

#### 4. LINGUAGEM - COMANDOS

##### 4.1. Definição de Arquivos - Sintaxe do comando:

{	D DE DEF . . . DEFINA	/xxxxxx.yyy,zzz	}
---	---	-----------------	---

onde: xxxxxx - é o nome do arquivo de 1 à 6 caracteres alphanuméricos

yyy - é o Nº da conta de 0 à 255

zzz - é o Nº de setores do arquivo de 1 à 450

##### 4.2. Seleção de arquivos - Sintaxe do comando:

{	A AT ATI . . . ATIVE	/xxxxxx.yyy	}
---	--	-------------	---



onde: P ou T ou V é nome do periférico (P-fita perfurável, V - vídeo, T - teletype) sobre o qual será feita a listagem (Ver item 5.4)

C - Quando a opção C estiver presente, será listado no vídeo todos os nomes dos Arquivos existentes no catálogo da zona DA.

X:Y - Inclusivem, são as linhas que serão listados em um dos periféricos.

X - Será listada a linha número x.

#### 4.5. Transferência de Arquivo - Sintaxe do Comando:

$$\left\{ \begin{array}{l} \% \\ \left[ \begin{array}{l} C \\ CO \\ COP \\ \cdot \\ \cdot \\ COPIE \end{array} \right] \\ \end{array} \right. /xxxxxx.yyy \left. \begin{array}{l} \% \\ \end{array} \right\}$$

onde: xxxxxx - é o nome do arquivo de 1 à 6 caracteres alphanumérico.

yyy - é o número da conta de 1 à 255 (Ver item 5.5).

## 4.6. Remoção de arquivos, linha ou trechos:

$$\left\{ \begin{array}{l} \% \left[ \begin{array}{l} R \\ RE \\ REM \\ \cdot \\ \cdot \\ REMOVA \end{array} \right] / \left[ \begin{array}{l} X \\ X:Y \end{array} \right] \% \end{array} \right\}$$

onde: X - é o número da linha que será removida do arquivo

X:Y - inclusivem, são as linhas que serão removidas do arquivo.

OBS: Se "X" ou "X:Y" não estiver presente será removido todo o arquivo ativado (Ver item 5.6).

## 4.7. Apronta para compilação - Sintaxe do comando:

$$\left\{ \begin{array}{l} \% \left[ \begin{array}{l} E \\ EX \\ EXE \\ \cdot \\ \cdot \\ EXEC \end{array} \right] / \left[ \begin{array}{l} F \\ A \end{array} \right] \% \end{array} \right\}$$

onde: A - é o programa Assembler

F - é o programa Fortran

OBS: Ver item 5.7.

## 5. FUNÇÃO DOS COMANDOS

### 5.1. Definição do Arquivo:

5.1.1. Cria um arquivo na zona DA do disco com o nome xxxxxx, conta yyy. O tamanho do arquivo será de zzz, com fator de grupo igual a 3. O tamanho do Registro é de 80 bytes. Os parâmetros que estiverem fora dos limites especificados em 4.1, provocarão erro de sintaxe no comando.

### 5.2. Seleção de Arquivos:

5.2.1. Após este comando, será selecionado entre os arquivos do catálogo, o arquivo de nome xxxxxx, conta yyy, que será ativado.

Importante: Este deverá ser o primeiro comando dado pelo usuário sobre o console (vídeo), caso o arquivo já exista. Caso contrário deverá ser o segundo. Qualquer tentativa de utilização de outros comandos (a menos do "defina" e "Liste/C") antes deste ser executado, ocasionará um erro do tipo 5.

5.2.2. A desativação do Arquivo é feita apertando as teclas

"Control + Bell".

5.2.3. Caso o usuário queira utilizar o arquivo novamente, torna-se necessário ativá-lo de novo.

5.3. Inserções em arquivos:

5.3.1. Opções "C" e "P":

5.3.1.1. Quando estas opções estiverem presente, o editor começará a inserir os dados lidos em cartões e fitas respectivamente, a partir do início do arquivo ativado. Para a opção "L", o fim dos dados é detetado após encontrar um "%".

Importante: O editor de programas só lê dados em fita de papel, perfuradas através do próprio editor.

5.3.2. Opções "V" e "N":

5.3.2.1. Quando a opção "V" estiver presente sem a opção "N", o editor começará a enumerar tela do Console a partir de 0001, posicionando o cursor, a cada vez na 7a. coluna, (que equivale a 1a. coluna do cartão), a partir de onde, o usuário deverá iniciar a digitação da linha. Após o fim de cada linha, o sistema numera a próxima linha, e o processo continua.



5.3.2.1.1. O fim da linha é detetado após o recebimento dos caracteres "↑", "↓" e "Return". Neste momento o sistema fará inserção da linha digitada.

Importante: O usuário deverá aguardar a numeração da próxima linha, antes de iniciar a movimentação do cursor.

5.3.2.1.1. Quando a opção "V" estiver presente com a opção "N", o sistema iniciará a numeração da linha a partir de N+1, procedendo a partir daí, como em 5.3.2.1.

### 5.3.3. Alterações de Linhas

5.3.3.1. Após a digitação da linha e inserção em disco, o usuário poderá, caso a linha ainda esteja na tela do console, alterar o conteúdo desta linha movimentando o cursor até a posição desejada, fazendo as alterações que achar necessário.

5.3.3.3. O fim das alterações de cada linha será detetada com o recebimento dos caracteres "↑", "↓", "Return", após o qual será feito um "replace da linha".

Importante: Após ser digitado um dos caracteres de controle acima, o usuário deverá aguardar o sinal da Campainha "Bell" antes de começar a movimentar o cursor.

#### 5.4. Listagem de Programas ou textos:

##### 5.4.1. Opções "P", "T" com ou sem as opções "x" ou "x:y".

5.4.1.1. Quando estas opções estiverem presentes, sem as opções "x" e "x:y" o Editor listará sobre a fita ou teletipo, respectivamente, todo o arquivo. Caso uma das opções "x" ou "x:y", esteja presente, será listada a linha "x" ou as linhas entre "x:y", inclusive, sobre um dos periféricos.

OBS: Se a listagem é sobre o teletipo, o usuário deverá antes, ligar as teclas "LIGNE e "MOTEUR" do TY nº 81.

##### 5.4.2. Opção "V e opções "x" e "x:y":

5.4.2.1. Quando esta opção estiver presente sem as opções "x" ou "x:y", todo o arquivo será listado na tela do console, de 15 em 15 linhas.

5.4.2.1.1. Caso o usuário queira fazer alteração na página listada, deverá proceder como em 5.3.3. e 5.3.3.2.

5.4.2.1.2. O fim desta interrupção é detetada após o recebimento do carácter "Control + Bell", após o qual a listagem continuará.

5.4.2.2. Quando a opção "V" estiver presente juntamente com uma das opções "X" e "X:Y", a exemplo de 5.4.1.1., serão listados na tela do console, a linha "X" ou as linhas "X:Y" inclusive.

5.4.2.2.1. Todas as alterações poderão ser feitas conforme especificado em 5.4.2.1. e 5.4.2.1.2.

5.4.2.3. O fim do comando `liste` é detetado após o recebimento do carácter de controle "Control + Bell".

5.4.3. Quando a opção "C" estiver presente, será listado no vídeo, todos os nomes dos arquivos do Catálogo da zona DA. Por segurança os números das contas não serão listados.

5.5. Transferência de arquivos:

5.5.1. Após o comando `Copie`, todo o conteúdo do arquivo ativado, será copiado no arquivo xxxxxx, conta yyy.

5.6. Remoção de arquivos, linha ou trechos:

5.6.1. Caso as opções "X" ou "X:Y" não esteja presente, todo o arquivo selecionado pelo comando `ative`, será deletado.

5.6.2. Se uma das opções, "X" ou "X:Y" estiver presente será removida a linha "X" ou as linhas entre "X:Y" (inclusive), do arquivo.

5.7. Preparação para Compilação:

5.7.1. O comando para compilação provoca uma cópia do arquivo do usuário, em um arquivo de trabalho do Editor, em modo aceitável pelo Compilador Assembler ou Fortran. Este comando só deverá ser dado no caso do usuário necessitar COMPILAR o seu programa. Então após este comando a tecla "Control + EOT" deverá ser acionada, o que provocará o aborto do Editor, aparecendo sobre o teletype, a mensagem %% FIN EDITOR.

5.7.2. No caso de Programa Assembler e após receber a mensagem %% FIN EDITOR, o usuário para executar o seu programa, dará sobre o teletype, sem recarregar o sistema, o seguinte comando:

1) %CALL/ASS2/F1, GO

5.7.2.1. A listagem do fonte, não será feita.

2) %CALL/ASS2/F1, GO, LO

5.7.2.2. A listagem do fonte será feita sobre o teletype.

5.7.3. Para executar um programa FORTRAN o usuário deverá carre

gar o sistema com o monitor "MTRD03", (ver 3.1., letras, a, b, c e d), dando em seguida os seguintes comandos sobre o teletype:

- a) %AS/M:SI, T:DM
- b) %CALL/FORTD/

5.7.3.1. Com listagem do fonte.

- a) %AS/M:SI, T:DM
- b) %AS/M:LO, T:NO
- c) %CALL/FORTD/

5.7.3.2. Sem listagem do fonte

5.7.4. A link-edição e execução do Programa, continua do modo usual:

- a) %CALL/LINKD/GO - assembler
- b) %CALL/LINKD/SL - fortran
- c) %LOAD - assembler/fortran
- d) %RUN - assembler/fortran

## 6. TIPOS DE ERROS

- ERRO 1 - Início de comando Esperado.
- ERRO 2 - Comando Inexistente ou nome de arquivo/conta não associado na definição do arquivo.
- ERRO 3 - Overflow no arquivo em uso.
- ERRO 4 - Operação INVÁLIDA.
- ERRO 5 - Primeiro comando deve ser uma definição ou seleção de arquivo.
- ERRO 6 - Carácter inválido em nome de arquivo, conta ou setores.
- ERRO 7 - Arquivo, conta ou setores definidos fora dos limites permitidos.
- ERRO 8 - Ausência do número de setores em comando de definição de arquivo.
- ERRO 9 - Argumento inválido em comando.

ANEXO II

```

C *****
C IMPLICIT INTEGER(A-Z)
C CURSOR -SIMULA O CURSOR DO VIDEO
C COM -FLAG DE COMANDO (SUBROUTINE)
C ARQ -FLAG ARQUIVO (SUBROUTINE)
C CON -FLAG NUMERO DE CONTA
C SET -FLAG SETOR (SUBROUTINE)
C LIS -FLAG DO COMANDO LISTE
C INS -FLAG DO COMANDO INSIRA
C SETOR -VARIAVEL RECEBE NUMERO SETORES
C PTR -PONTEIRO INICIO DE CADA REGISTRO NO BUFFER
C KK -VARIAVEL DE AGRUPA RECEBE NUMEROS LINHAS
C L1 E L2 -VARIAVEIS AUXILIARES
C TY -RECEBE U NUMERO PERIFERICO
C ENDREG - ENDEREÇO REGISTRO QUE ESTAO NO VIDEO
C ENDSET - ENDEREÇO SETOR LINHA NO VIDEO
C ENDLIN -NUMERO DA LINHA QUE ESTA NO VIDEO
C CURSO -TEM OS CARACTERES DE CONTROLE
C NOMA -NOME DOS ARQUIVOS
C BUFFER -AREA DE TRABALHO
C VIDEO -IMAGEM DO VIDEO
C *****
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS, SETOR,PTR
COMMON KK,L1,L2,TY
COMMON /GR1/ENDREG(16),ENDSET(16),ENDLIN(16)
COMMON /GR2/SETOR1
DIMENSION CURSO(8),NOMA(6),NOMC(6),BUFFER(43)
DATA CURSO/13,21,22,24,50,61,64,63/
PREPARACAO DO VIDEO
4 CALL ESCR(24)
CALL MOVCUR(24)
FLAG=0
CURSOR=1
1 CALL ESCR(13)
CALL MOVCUR(13)
CALL ESCR(111)
CALL MOVCUR(61)
2 I=LER(0)
108=% E 47=BELL
IF(I.EQ.108)GO TO 10
IF(I.EQ.47)GO TO 4
CONTROL+EOT - FIM DO PROGRAMA
IF(I.EQ.55)GO TO 150
DO 3 L=1,8
IF(CURSO(L).EQ.I)GO TO 6
3 CONTINUE
CALL ERRO(241)
GO TO 1
6 CALL MOVCUR(I)
GO TO 2
10 CALL MOVCUR(61)
CALL COMAND
IF(COM.EQ.0)GO TO 1
IF(LIS.EQ.-1)GO TO 105
IF(COM.NE.2.AND.COM.NE.1)GO TO 35
GO TO(15,30),COM
C INICIO DO COMANDO DEFINA *****
15 CALL NOMARQ(NOMC)
IF(ARQ.EQ.0)GO TO 1
CALL NUMCON(CONTC)
IF(CON.EQ.0)GO TO 1
CALL NSETOR
IF(SET.EQ.0)GO TO 1
CALL CONCAT(NOMC)
CALL PFILE(NOMC,CONTC,SETOR)
CALL DEFINA(NOMC,CONTC,BUFFER)
GO TO 1
C FIM DO COMANDO DEFINA
C INICIO DO COMANDO ATIVE *****

```



```

30 CALL NOMARQ(NOMA)
   IF(ARQ.EQ.0)GO TO 1
   CALL NUMCON(CONTA)
   IF(CON.EQ.0)GO TO 1
C   ABERTURA E ASSOCIACAO SOBRE M:EI
   CALL CONCAT(NOMA)
   CALL ATIVE(NOMA,CONTA,BUFFER)
   CALL ESCR(21)
   CALL MOVCUR(21)
   FLAG=1
   GO TO 1
C   FIM DO COMANDO ATIVE
C   CONTINUACAO
35 IF(FLAG.EQ.0)GO TO 125
C   IOS PODE SER MUDADA PARA OUTRO NUMERO QUANDO FOR
C   INSERIDO UM NOVO COMANDO-PUNCH
   GO TO(15,30,45,60,75,90,105,120),COM
45 CALL ARGREM
   IF(LIS.EQ.0)GO TO 1
   GO TO(62,64,61),LIS
61 CALL DELETE(NOMA)
   FLAG=0
   GO TO 68
62 K1=L1
   K2=L1
   GO TO 66
64 K1=L1
   K2=L2
66 CALL REMOVA(K1,K2,BUFFER)
68 CALL ESCR(21)
   CALL MOVCUR(21)
   GO TO 1
C   INICIO DO COMANDO COPIE      ****
60 CALL NOMARQ(NOMC)
   IF(ARQ.EQ.0)GO TO 1
   CALL NUMCON(CONTC)
   IF(CON.EQ.0)GO TO 1
   CALL CONCAT(NOMC)
   CALL COPIE(NOMC,NOMA,CONTC,CONTA,BUFFER)
C   FIM DO COMANDO COPIE
   IF(SET)73,73,74
73 CALL ERRO(244)
   GO TO 1
74 CALL ESCR(21)
   CALL MOVCUR(21)
   GO TO 1
C   INICIO DO COMANDO LISTE     ****
75 CALL ARG LIS
   IF(LIS.EQ.0)GO TO 1
   CALL ESCR(21)
   CALL MOVCUR(21)
   GO TO(86,87,88),LIS
86 K1=1
   K2=30000
   GO TO 89
87 K1=L1
   K2=L1
   GO TO 89
88 K1=L1
   K2=L2
89 CALL LISTE(K1,K2,BUFFER)
C   FIM DO COMANDO LISTE
C   INICIO DO COMANDO INSIRA    ****
90 CALL ARGINS
   IF(INS.EQ.0)GO TO 1
   GO TO(92,94),INS
92 LL=0
   GO TO 96
94 LL=KK

```

```

96 CALL INSIRA(LL,0,BUFFER)
C FIM DO COMANDO INSIRA
GO TO 1
C LISTAGEM DO CATALOGO DA ZONA DA
105 CALL LISTC
LIS=0
CALL ESCR(21)
CALL MOVCUR(21)
GO TO 1
C COPIA O ARQUIVO DE TRABALHO NO ARQEDI
120 CALL EXEC(BUFFER)
CALL ESCR(21)
CALL MOVCUR(21)
GO TO 1
125 CALL ERRO(245)
GO TO 1
C FAZ ASSOCIACOES E ESCRIVE FIN EDITOR
150 CALL FEDIT
STOP
END
C *****
C SUBROUTINE MOVCUR(W)
C ESTA SUBROUTINE SIMULA O CURSOR DO VIDEO.
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS, SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION CURS(8)
DATA CURS/13,21,22,24,50,61,63,64/
DATA BRANCO/147
DO 1 L=1,8
IF(CURS(L).EQ.W)GO TO 3
1 CONTINUE
C PARA CADA CARACTER APONTADO POR L UMA TAREFA DISTINTA.
3 GO TO(13,21,22,24,50,61,63,64),L
C RETURN
13 K=MOD(CURSOR,80)
IF(K.EQ.0)GO TO 15
CURSOR=CURSOR-K+1
RETURN
15 CURSOR=CURSOR-79
RETURN
C CURSOR PARA BAIXO
21 IF(CURSOR.LE.80)GO TO 17
CURSOR=CURSOR+80
RETURN
17 CURSOR=1
CALL MOVEL
RETURN
C CURSOR PARA ESQUERDA
22 IF(CURSOR.NE.1201)GO TO 23
CURSOR=80
RETURN
23 RESTO=MOD(CURSOR,80)
IF(RESTO.NE.1)GO TO 20
CURSOR=CURSOR+159
RETURN
20 CURSOR=CURSOR-1
RETURN
C CLEAR
24 DO 25 K=1,1280
VIDEO(K)=BRANCO
25 CONTINUE
CURSOR=1
RETURN
C LINE CLEAR. POR ENQUANTO NAO FAZ NADA
50 RETURN
C CURSOR PARA DIREITA
61 IF(CURSOR.NE.80)GO TO 62

```

```

CURSOR=1
CALL MOVEL
RETURN
62 RESTO=MOD(CURSOR,80)
IF(RESTO.NE.0)GO TO 65
CURSOR=CURSOR-159
RETURN
65 CURSOR=CURSOR+1
RETURN
63 IF(CURSOR.LT.1201)GO TO 66
CURSOR=CURSOR-1200
RETURN
66 CURSOR=CURSOR+80
RETURN
C   ESPACO
64 CURSOR=CURSOR+1
RETURN
END
C   *****
C   SUBROUTINE COMAND
C   INST -MATRIZ ONDE SAO GUARDADOS OS COMANDOS
C   ESTA SUBROUTINA LER O COMANDO DO VIDEO E VERIFICA SE
C   E UM COMANDO VALIDO.
C   *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,      SETOR,PTR
COMMON KK,LI,L2,TY
DIMENSION INST(7,8),CUR(9 )
DATA CUR/109,14,22,23,25,48,51,62,64/
DATA INST /196,197,198,201,213,193,97,193,227,201,229,197,97,64,
1      217,197,212,214,229,193,97,195,214,215,201,197,97,64,
2      211,201,226,227,197,97,64,201,213,226,201,217,193,97,
3      90,214,226,227,217,197,97,197,231,197,195,228,227,197/
COM=0
CONTI=0
CONTA=0
PTR APONTA PARA O INICIO DA LINHA
PTR=CURSOR
2 I=LER(0)
I=I+1
DO 4 L=1,9
IF(CUR(L).EQ.I)GO TO 3
C   4 CONTINUE
COLOCA O CARACTER NO VIDEO (IMAGEM)
VIDEO(CURSOR)=I-1
CALL MOVCUR(61)
CONTI=CONTI+1
IF(I.NE.0)GO TO 2
3 GO TO(1,14,22,23,25,47,51,62,64),L
1 VIDEO(CURSOR)=I-1
CALL MOVCUR(61)
C   VERIFICA A PRIMEIRA LETRA DO COMANDO
DO 5 L=1,8
IF(INST(1,L)-VIDEO(PTR))5,6,5
5 CONTINUE
CALL ERRO(242)
RETURN
6 COM=L
C   VERIFICA AS LETRAS RESTANTES
DO 8 C=2,7
PTR=PTR+1
IF(VIDEO(PTR)-97)7,9,7
7 IF(INST(C,L)-VIDEO(PTR))10,8,10
8 CONTINUE
GO TO 10
9 PTR=PTR+1
I=VIDEO(PTR)
IF(COM.EQ.5.AND.I.EQ.195)LIS=-1
RETURN

```

```

10 CALL ERRO(242)
   COM=0
   RETURN
C   RETURN
C   14 GO TO 22
C   CURSOR PARA BAIXO
C   22 CALL MOVCUR(I-1)
   CALL ERRO(244)
   RETURN
C   CURSOR PARA ESQUERDA
C   23 IF(CONTI)30,22,30
C   30 CONTI=CONTI-1
   CALL MOVCUR(I-1)
   GO TO 2
C   CLEAR
C   25 GO TO 22
C   ABANDONO DO COMANDO
C   47 RETURN
C   LINE CLEAR
C   51 RETURN
C   CURSOR PARA DIREITA
C   62 CONTI=CONTI+1
   CALL MOVCUR(I-1)
   GO TO 2
C   CURSOR PARA CIMA
C   64 IF(COM.EQ.0)GO TO 22
   RETURN
   END
C   *****
C   SUBROUTINE ARGLIS
C   ESTA SUBROTINA VERICA SE OS ARGUMENTOS DO COMANDO
C   LISTE SAO VALIDOS.CASO CONTRARIO DEVOLVE COM LIS=0
C   *****
C   IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CUN,SET,LIS,INS,      SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION LN1(4),LN2(4)
C   VARIAVEIS DO COMMON
LIS=0
L1=0
L2=0
TY=0
C   PTR SEMPRE APONTA PARA CARACTER NAO ANALIZADO
I=VIDEO(PTR)
C   229=V,227=T,215=P,107=, E 108=%,122=:
C   IF(I.EQ.229.OR.I.EQ.227.OR.I.EQ.215)GO TO 5
C   3 CALL ERRO(249)
   RETURN
C   TY INDICA EM QUAL PERIFERICO SERA A OPERACAO
C   5 TY=I-220
   PTR=PTR+1
   I=VIDEO(PTR)
   IF(I.EQ.107)GO TO 10
   IF(I.NE.108)GO TO 3
   LIS=1
   RETURN
C   10 DO 11 K=1,4
   PTR=PTR+1
   I=VIDEO(PTR)
C   VERIFICA SE E NUMERO
C   IF(I.LT.240.OR.I.GT.249)GO TO 20
   LN1(K)=I-240
C   11 CONTINUE
   CALL AGRUPA(LN1,K-1)
   L1=KK
   I=VIDEO(PTR)
   PTR=PTR+1
C   20 IF(I.EQ.122.AND.K.GT.1)GO TO 30
   IF(I.EQ.108.AND.K.GT.1)GO TO 25

```

```

CALL ERRO(249)
LIS=0
RETURN
C 25 LIS=2
FAZ AGRUPAMENTO DO NUMERO LIDO
CALL AGRUPA(LN1,K-1)
L1=KK
RETURN
30 CALL AGRUPA(LN1,K-1)
L1=KK
DO 40 J=1,4
PTR=PTR+1
I=VIDEQ(PTR)
IF(I.LT.240.OR.I.GT.249)GO TO 45
LN2(J)=I-240
40 CONTINUE
CALL AGRUPA(LN2,J-1)
L2=KK
I=VIDEQ(PTR)
PTR=PTR+1
45 IF(I.EQ.108.AND.J.GT.1)GO TO 50
CALL ERRO(249)
LIS=0
RETURN
50 LIS=3
CALL AGRUPA(LN2,J-1)
L2=KK
RETURN
END
C *****
C SUBROUTINE ARGREM
C ESTA SUBROUTINE VERIFICA OS ARGUMENTOS DE REMOVE
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEQ(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION LN1(4),LN2(4)
L1=0
L2=0
IF(VIDEQ(PTR).NE.108)GO TO 5
LIS=3
RETURN
5 DO 11 K=1,4
I=VIDEQ(PTR)
PTR=PTR+1
C VERIFICA SE E NUMERO
IF(I.LT.240.OR.I.GT.249)GO TO 20
LN1(K)=I-240
C 11 CONTINUE
FAZ AGRUPAMENTO DO NUMERO LIDO
CALL AGRUPA(LN1,K-1)
L1=KK
I=VIDEQ(PTR)
PTR=PTR+1
C 122 = : E 108 =%
20 IF(I.EQ.122.AND.K.GT.1)GO TO 30
IF(I.EQ.108.AND.K.GT.1)GO TO 25
CALL ERRO(249)
LIS=0
RETURN
25 LIS=1
CALL AGRUPA(LN1,K-1)
L1=KK
RETURN
30 CALL AGRUPA(LN1,K-1)
L1=KK
DO 40 J=1,4
I=VIDEQ(PTR)
PTR=PTR+1

```

```

IF(I.LT.240.OR.I.GT.249)GO TO 45
LN2(J)=I-240
40 CONTINUE
CALL AGRUPA(LN2,J-1)
L2=KK
I=VIDE0(PTR)
PTR=PTR+1
45 IF(I.EQ.108.AND.J.GT.1)GO TO 50
CALL ERRO(249)
LIS=0
RETURN
50 LIS=2
CALL AGRUPA(LN2,J-1)
L2=KK
RETURN
END
C *****
C SUBROUTINE ARGINS
C SUBROUTINE PARA TESTE DOS ARGUMENTOS DE LISTE.
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDE0(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION LN(5)
INS=0
C PTR APONTA PARA O CARACTER NAO ANALIZADO
I=VIDE0(PTR)
PTR=PTR+1
C VERIFICA SE INSERCAO E A PARTIR DO VIDE0
IF(I.NE.229)GO TO 20
TY=I-220
I=VIDE0(PTR)
PTR=PTR+1
C 107 = , E 108 = %
IF(I.NE.107)GO TO 23
DO 10 J=1,4
I=VIDE0(PTR)
PTR=PTR+1
C TESTA SE NUMERO
IF(I.LT.240.OR.I.GT.249)GO TO 15
LN(J)=I-240
10 CONTINUE
12 IF(I.NE.108)GO TO 30
CALL AGRUPA(LN,J-1)
INS=2
RETURN
15 IF(J.GT.1)GO TO 12
CALL ERRO(246)
RETURN
C 215 = P E 211 = L
20 IF(I.NE.215.AND.I.NE.211)GO TO 30
TY=I-210
I=VIDE0(PTR)
23 IF(I.NE.108)GO TO 25
INS=1
RETURN
25 CALL ERRO(246)
RETURN
30 CALL ERRO(249)
RETURN
END
C *****
C SUBROUTINE NSETOR
C ESTA SUBROUTINE VERIFICA A VALIDADE DO NUMERO DOS
C SETORES E DEIXA EM SETOR. SETA O FLAG SET
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDE0(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR,PTR

```

```

COMMON KK,L1,L2,TY
DIMENSION NSET(5)
SET=0
DO 1 Y=1,5
1 NSET(Y)=0
DO 6 J=1,5
  I=VIDEO(PTR)
  PTR=PTR+1
C VERIFICA SE E NUMERO
  IF(I.GE.240.AND.I.LE.249)GO TO 5
  IF(J.GT.1)GO TO 2
  CALL ERRO(248)
  RETURN
C 108 = % (FIM DE COMANDO)
2 IF(I.EQ.108)GO TO 7
  CALL ERRO(246)
  RETURN
5 I=I-240
  NSET(J)=1
6 CONTINUE
  I=VIDEO(PTR)
  IF(I.NE.108)GO TO 10
7 SET=1
C AGRUPA OS NUMEROS LIDOS
  CALL AGRUPA(NSET,J=1)
  SETOR=KK
C VERIFICA SE SETOR ESTAR NO INTERVALO PERMITIDO
  IF(SETOR.LE.0.OR.SETOR.GT.450)GO TO 10
  RETURN
10 SET=0
  CALL ERRO(247)
  RETURN
END
C *****
C SUBROUTINE NOMARQ(NOM)
C ESTA SUBROUTINE COLOCA EM NOM O NOME DO ARQUIVO E
C VERIFICA AD MESMO TEMPO SE E UM NOME VALIDO,SETA ARQ
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEOS(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS, SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION NOM(1)
DATA BREAK/2H /
ARQ=0
DO 4 K=1,6
C ZERA NOM
  NOM(K)=BREAK
4 CONTINUE
DO 8 K=1,6
  I=VIDEO(PTR)
  PTR=PTR+1
C VERIFICA SE E LETRA
  IF(I.GE.193.AND.I.LE.233)GO TO 7
  IF(K.GT.1)GO TO 6
  CALL ERRO(246)
  RETURN
C VERIFICA SE E NUMERO
6 IF(I.LT.240.OR.I.GT.249)GO TO 9
7 NOM(K)=I
8 CONTINUE
  I=VIDEO(PTR)
  PTR=PTR+1
  IF(I.NE.75)GO TO 15
  IF(I.NE.75)GO TO 11
C ACIMA TESTAMOS SE E '!'
  ARQ=1
  RETURN
11 CALL ERRO(246)
  RETURN

```

```

15 CALL ERRO(247)
   RETURN
   END

```

```

C
C
C *****
C SUBROUTINE NUMCON(CONT)
C ESTA SUBROUTINE VERIFICA A VALIDADE DO NUMERO DA
C CONTA E GUARDA EM CONT. SETA FLAG CON
C *****
C IMPLICIT INTEGER(A-Z)
C COMMON VIDEO(1280)
C COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR, PTR
C COMMON KK,L1,L2,TY
C DIMENSION NUM(3)
C CON=0
C CONT=0
C PTR APONTA SEMPRE PARA O CARACTER NAO ANALIZADO
C I=VIDEO(PTR)
C IO8=% E IO7=% DIGO =,
C IF(I.EQ.IO8)GO TO 30
C DO 5 K=1,3
C     I=VIDEO(PTR)
C     PTR=PTR+1
C VERIFICA SE NAO E NUMERO
C     IF(I.LT.240.OR.I.GT.249)GO TO 6
C     I=I-240
C     NUM(K)=I
5 CONTINUE
C I=VIDEO(PTR)
C PTR=PTR+1
C IF(I.NE.IO8.AND.I.NE.IO7)GO TO 25
6 IF(K-1)12,12,7
7 IF(I.EQ.IO8)GO TO 10
C IF(I-IO7)12,20,12
C VERIFICA SE COMANDO DEFINA OU ATIVE
10 IF(COM.EQ.4.OR.COM.EQ.2)GO TO 13
12 CALL ERRO(246)
   RETURN
13 CON=1
C FAZ AGRUPAMENTO DOS NUMEROS
C CALL AGRUPA(NUM,K-1)
C CONT=KK
C IF(CONT.LT.0.OR.CONT.GT.255)GO TO 25
C RETURN
20 IF(COM=1)12,13,12
25 CONT=0
C CALL ERRO(247)
   RETURN
30 CALL ERRO(248)
   RETURN
   END
C *****
C SUBROUTINE DEFINA(NOMC,CONTC,BUFFER)
C NOMC          - NOME DO ARQUIVO
C CONTC        - NUMERO DA CONTA DO ARQUIVO
C BUFFER       - AREA DE TRABALHO EDITOR
C *****
C IMPLICIT INTEGER(A-Z)
C COMMON VIDEO(1280)
C COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR, PTR
C COMMON KK,L1,L2,TY
C DIMENSION BUFFER(1),NOMC(1),NOMA(1)
C FAZ ASSOCIACAO
C FL=1
C CALL PASGN(NOMC,CONTC,FL)
C FL=0
C CALL PASGN(NOMC,CONTC,FL)
C POSICIONA O PRIMEIRO CONTEXTO

```



```

CALL CBNDG (BUFFER,0,-1)
CALL ESDISC (1,0,1,BUFFER)
C POSICIONA O SEGUNDO CONTEXTO
CALL CBNDG (BUFFER,1,1)
CALL ESDISC (1,0,2,BUFFER)
C POSICIONA O TERCEIRO CONTEXTO
CALL CBNDG (BUFFER,SETOR,0)
CALL ESDISC (1,0,3,BUFFER)
CALL ESCR (21)
CALL MDVCR (21)
RETURN
END
C *****
C SUBROUTINE ATIVE (NOMA,CONTA,BUFFER)
C NOMA E CONTA SAO NOME E CONTA DO ARQUIVO
C *****
C IMPLICIT INTEGER (A-Z)
COMMON VIDEO (1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS, SETOR,PTR
COMMON KK,L1,L2,TY
COMMON /GR2/SETOR1
C DIMENSION BUFFER (1),NOMC (1),NOMA (1)
C FAZ ASSOCIACAO SOBRE M:EI
FL=0
CALL PASGN (NOMA,CONTA,FL)
C IF (FL.LT.0) GO TO 130
C FAZ ASSOCIACAO SOBRE M:ED
FL=1
CALL PASGN (NOMA,CONTA,FL)
C FAZ LEITURA DO TERCEIRO CONTEXTO. ARQUIVO VAZIO?
CALL ESDISC (0,0,3,BUFFER)
CALL ENDSEG (BUFFER,SETOR1,EREG)
RETURN
130 CALL ERRO (242)
RETURN
END
C *****
C SUBROUTINE INSIRA (LL,FLIS,BUFFER)
C LL - INSERCAO A PARTIR DE LL
C FLIS - CHAMADA POR LISTE (FLAG)
C BUFFER - MESMA AREA DE TRABALHO
C CUR - CARACTERES DE CONTROLE
C FAZ INSERCAO EM DISCO A PARTIR DO VIDEO/LEITORA/FITA
C *****
C IMPLICIT INTEGER (A-Z)
COMMON VIDEO (1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,SETOR,PTR
COMMON KK,L1,L2,TY
COMMON /GR1/ENDREG (16),ENDSET (16),ENDLIN (16)
COMMON /GR2/SETOR1
DIMENSION LN (4),CUR (9)
DIMENSION BUFFER (1)
DATA CUR /13,47,63,21,61,22,24,50,39/
DATA BRANCO, LN /1H ,4*0/
C VARIAVEIS AUXILIARES
CARAC=0
JAFUI=1
FLAG=0
XX=0
LINHA=LL
M=0
PTR=0
IFINS=0
PTR1=0
C EM JK TEMOS O TAMANHO DO ARQUIVO
JK=(SETOR1-1)*3-1
IF (FLIS.NE.0) GO TO 30
IF (TY.EQ.9) GO TO 20
CALL ASSI (TY)
C ENTRADA POR CARTAO

```

```

CALL INCRPT(BUFFER)
CALL ESCR(21)
C MOVIMENTACAO DO CURSOR(ATUALIZACAO)
CALL MOVCUR(21)
RETURN
20 LINHA=LINHA+1
IF(LINHA.LE.JK)GO TO 22
CALL ERRO(243)
GO TO 80
22 IF(I.EQ.21)GO TO 25
CALL ESCR(21)
CALL MOVCUR(21)
IF(I.NE.63)GO TO 25
CALL ESCR(21)
CALL MOVCUR(21)
25 CALL ESCR(13)
CALL MOVCUR(13)
PTR1=CURSOR+1
C COLOCA NO BUFFER COMPACTADO
CALL CVIDEO(VIDEO(1),LINHA,0)
CURSOR=CURSOR+6
30 I=LER(0)
DO 35 L=1,9
IF(CUR(L).EQ.I)GO TO 40
35 CONTINUE
VIDEO(CURSOR)=I
CALL MOVCUR(61)
FLAG=1
GO TO 30
C VAI PARA CARACTER CONTROLE INDICADO POR L (CUR)
40 GO TO(101,80,101,101,105,102,102,102,115),L
C REPLACE DE UMA LINHA ***
46 DO 57 K=1,4
CARAC=VIDEO(PTR)
IF(CARAC.LT.240.OR.CARAC.GT.249)GO TO 56
LN(K)=CARAC-240
PTR=PTR+1
57 CONTINUE
CALL AGRUPA(LN,K-1)
XX=KK
C SUBSTITUE SEGUIR ***** (XX-1,BUFFER,NSECT,EREG)
DO 59 XY=1,16
IF(ENDLIN(XY).EQ.XX)GO TO 61
59 CONTINUE
GO TO 56
61 NS=ENDSET(XY)
ER=ENDREG(XY)
C FIM SEGUIR *****
CALL ESDISC(0,NS,ER,BUFFER)
C COMPACTATA PARA IMPRIMIR EM DISCO
CALL COMPAC(VIDEO(PTR+1),BUFFER(1),72)
DO 60 K=37,40
BUFFER(K)=BRANCO
60 CONTINUE
CALL ESDISC(1,NS,ER,BUFFER)
C BELL
CALL ESCR(47)
FLAG=0
CALL MOVCUR(I)
GO TO 30
C POSICIONA O CURSOR NO VIDEO NA 1 LINHA
56 IF(CURSOR.LE.80)GO TO 58
XXP=CURSOR
DO 55 XY=81,XXP,80
CALL ESCR(21)
CALL MOVCUR(21)
55 CONTINUE
CALL ESCR(13)
CALL MOVCUR(13)
58 CALL ERRO(244)

```

```

                FLAG=0
                IF (FLIS) 25, 25, 30
C   PTR=PTR1 - UMA NOVA LINHA A INSERIR ***
70  IF (LL.EQ.0) GO TO 71
C   PROCURA O REGISTRO 'IN' A PARTIR DE ONDE E INSERIDO NOVO
    CALL SEGUIR(LINHA-2,BUFFER,NSECT,EREG)
    IF (SETOR.NE.-1) GO TO 72
    CALL MOVCUR(I)
    RETURN
71  NSECT=0
    EREG=1
    LL=1
72  CALL ESDISC(0,NSECT,EREG,BUFFER)
    CALL ENDSEG(BUFFER,NSECTS,EREGS)
    CALL CBNDIC(BUFFER,NSECTF,EREGF)
    CALL ESDISC(1,NSECT,EREG,BUFFER)
    CALL COMPAC(VIDEO(PTR+5),BUFFER(1),72)
    DO 73 K=37,40
        BUFFER(K)=BRANCO
73  CONTINUE
    CALL CBNDIC(BUFFER,NSECTS,EREGS)
    CALL ESDISC(1,NSECTF,EREGF,BUFFER)
C   GUARDA ENDEREÇO DAS LINHAS SOBRE A TELA
    ENDREG(1)=EREGF
    ENDSET(1)=NSECTF
    ENDLIN(1)=LINHA
    NSECT=NSECTF
    EREG=EREGF
    EREGF=EREGF+1
    IF (EREGF.LE.3) GO TO 74
    NSECTF=NSECTF+1
    EREGF=1
74  CALL CBNDIC(BUFFER,NSECTF,EREGF)
    CALL ESDISC(1,0,2,BUFFER)
C   AVISA QUE JA INSERIU
    CALL ESCR(47)
    IFINS=1
    FLAG=0
    JAFUI=0
    CALL MOVCUR(I)
    GO TO 20
C   BELL - FIM DO COMANDO
80  IF (IFINS.EQ.0) GO TO 107
    CALL CBNDIC(BUFFER,SETOR1,1)
    CALL ESDISC(1,0,3,BUFFER)
C   COLOCA O CURSOR NA PRIMEIRA LINHA
107 IF (CURSOR.LE.80) RETURN
    XXP=CURSOR
    DO 100 L=80,XXP,80
        CALL ESCR(21)
        CALL MOVCUR(21)
100 CONTINUE
    CALL ESCR(13)
    CALL MOVCUR(13)
    RETURN
101 IF (FLAG.NE.0) GO TO 110
102 CALL MOVCUR(I)
    GO TO 30
C   APUNTA O REGISTRO ONDE FOI FEITA ALTERACOES
105 PM=MOD(CURSOR,80)
    IF (PM) 101,101,102
110 PTR=((CURSOR-1)/80)*80+2
    IF (PTR.NE.PTR1) GO TO 46
    CALL ELCIMA(BUFFER,NSECTF,EREGF,2)
    IF (JAFUI) 70,72,70
C   TECLA ESCAPE=FIM LI STAGEM
115 TY=1
    GO TO 107
    END
C   *****

```



```

C     FAZ ASSOCIACAO COM O ARQUIVO MAIOR(NOVO)
      FL=0
      CALL PASGN(NOMC,CONTC,FL)
      IF(FL.GE.0)GO TO 11
      SET=0
      RETURN
11    CONTINUE
      FL=1
      CALL PASGN(NOMC,CONTC,FL)
C     PEGA ENDERECO DO PROXIMO REGISTRO
      CALL ENDSEG(BUFFER,NSECT,EREG)
      NSCS=NSC
      NECS=NEC+1
      IF(NECS.LE.3)GO TO 20
      NSCS=NSC+1
      NECS=1
20    CONTINUE
C     FAZ CONVERSAO DO CONTEXTO
      CALL CBNDC(BUFFER,NSCS,NECS)
C     FAZ LEITURA(1) E (0) E IMPRESSAO
      CALL ESDISC(1,NSC,NEC,BUFFER)
      IF(EREG.LT.0)GO TO 30
      NSC=NSCS
      NEC=NECS
      GO TO 10
30    CONTINUE
      CALL CBNDC(BUFFER,NSCS,-1)
      CALL ESDISC(1,NSC,NEC,BUFFER)
      CALL CBNDC(BUFFER,1,1)
      CALL ESDISC(1,0,1,BUFFER)
      CALL CBNDC(BUFFER,NSCS,NECS)
      CALL ESDISC(1,0,2,BUFFER)
      CALL ESDISC(0,0,3,BUFFER)
      CALL ENDSEG(BUFFER,TS,TF)
      CALL CBNDC(BUFFER,TS,1)
      CALL ESDISC(1,0,3,BUFFER)
      FL=0
C     DEIXA O VELHO ARQUIVO ATIVADO PARA REMOCAO
      CALL PASGN(NOMA,CONTA,FL)
      FL=1
      CALL PASGN(NOMA,CONTA,FL)
      RETURN
      END
C     *****
C     SUBROUTINE ELCIMA(BUFFER,NSECT,EREG,COD)
C     COD -INDICA SE PRIMEIRO,SEGUNDO OU TERCEIRO CONTEXTO
C     ESTA SUBROUTINA RETIRA DO CONTEXTO(PRIMEIRO) O ENDERECO
C     DO PRIMEIRO REGISTRO DO ARQUIVO ATIVADO
C     *****
      IMPLICIT INTEGER(A-Z)
      COMMON VIDEO(1280)
      COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR,PTR
      COMMON KK,L1,L2,TY
      DIMENSION BUFFER(1)
      NSECT=0
      EREG=COD
C     FAZ A LEITURA DE REGISTRO DESEJADO
      CALL ESDISC(0,NSECT,EREG,BUFFER)
      CALL ENDSEG(BUFFER,NSECT,EREG)
      RETURN
      END
C     *****
C     SUBROUTINE SEGUIR(ENDLIN,BUFFER,NSECT,EREG)
C     ENDLIN -NUMERO DA LINHA A PARIR DE ONDE SERA FEITA
C     A BUSCA EM DISCO
C     NSECT EREG -SETOR E REGISTRO
C     *****
      IMPLICIT INTEGER(A-Z)
      COMMON VIDEO(1280)
      COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR,PTR

```

```

COMMON KK,L1,L2,TY
  DIMENSION BUFFER(1)
  SETOR=0
C   PEGA O ENDERECO DA PRIMEIRA LINHA
  CALL ELCIMA(BUFFER,NSECT,EREG,1)
C   IF(ENDLIN.EQ.0)RETURN
C   LOOP DA LEITURA
  MN=ENDLIN
C   PESQUISA ATE A LINHA ENDLIN
  DO 10 II=1,MN
    CALL ESDISC(O,NSECT,EREG,BUFFER)
    CALL ENDSEG(BUFFER,NSECT,EREG)
  IF(EREG.GE.0)GO TO 10
  SETOR=-1
  RETURN
10  CONTINUE
  RETURN
  END
C *****
C   SUBROUTINE MOVEL
C   ESTA SUBROUTINE MOVIMENTA AS LINHA NO VIDEO . SEMPRE QUE
C   UMA NOVA LINHA ENTRA A ULTIMA SAI SE O VIDEO ESTA CHEIO
C *****
  IMPLICIT INTEGER(A-Z)
  COMMON VIDEO(1280)
  COMMON COM,ARQ,CDN,SET,LIS,INS,SETOR,PTR
  COMMON KK,L1,L2,TY
  COMMON /GR1/ENDREG(16),ENDSET(16),ENDLIN(16)
  DATA BRANCO/IH /
  DO 10 K1=1,15
    K=16-K1+1
    M=(K-1)*80+1
    M1=M-80
    MM=M-1
    DO 5 L=M1,MM,1
      VIDEO(M)=VIDEO(L)
      M=M+1
    5  CONTINUE
  10  CONTINUE
  DO 15 K=1,80
    VIDEO(K)=BRANCO
  15  CONTINUE
  DO 20 K=1,15
    L=16-K+1
    ENDREG(L)=ENDREG(L-1)
    ENDLIN(L)=ENDLIN(L-1)
    ENDSET(L)=ENDSET(L-1)
  20  CONTINUE
C   COLOCA BRANCO NA NOVA LINHA
  ENDSET(1)=BRANCO
  ENDLIN(1)=BRANCO
  ENDREG(1)=BRANCO
  RETURN
  END
C *****
C   SUBROUTINE LISTE(K1,K2,BUFFER)
C   K1 E K2 -NUMEROS DAS LINHAS ENTRE AS QUAIS SERA LISTADA.
C   ESTA SUBROUTINA LISTA SOBRE O VIDEO, FITA DE PAPEL E
C   IMPRESSORA(TELETIPO) O ARQUIVO DO USUARIO, TODO O AR-
C   QUIVO OU APENAS UM TRECHO OU LINHA.
C *****
  IMPLICIT INTEGER (A-Z)
  COMMON VIDEO(1280)
  COMMON CURSOR,COM,ARQ,CDN,SET,LIS,INS,SETOR,PTR
  COMMON KK,L1,L2,TY
  COMMON /GR1/ENDREG(16),ENDSET(16),ENDLIN(16)
  DIMENSION BUFFER(1)
C   PEGA ENDERECO PRIMEIRO REGISTRO
  CALL ELCIMA(BUFFER,NSECT,EREG,3)
  IF(EREG.EQ.0)RETURN

```

```

NLV=0
C PROCURA ATE O REGISTRO DESEJADO
CALL SEGUIR(K1=1,BUFFER,NSECT,EREG)
IF(SETOR.LT.0.OR.EREG.LT.0)RETURN
I=K1-1
L=2
C VERIFICA SE LISTAGEM SOBRE O VIDEO
IF(TY.EQ.9)GO TO 20
IFBEG=1
L=1
20 CONTINUE
CALL ESDISC(0,NSECT,EREG,BUFFER)
NSECTL=NSECT
EREGL=EREG
IF(I.EQ.K2.OR.EREG.LT.0)GO TO 30
CALL ENDSEG(BUFFER,NSECT,EREG)
I=I+1
GO TO (40,50),L
C LISTAGEM SOBRE O VIDEO
50 CALL LVIDEO(BUFFER,I)
C COLOCA NO VIDEO AS LINHAS DO PROGRAMA
CALL CVIDEO(VIDEO,I,BUFFER)
ENDREG(1)=EREGL
ENDSET(1)=NSECTL
ENDLIN(1)=I
C FAZ O SHIFIT PARA CIMA DAS LINHAS
CALL MOVEL
NLV=NLV+1
IF(NLV.LT.15)GO TO 20
NLV=0
CURSOR=1
C CALL INSIRA(0,1,BUFFER)
TESTA SE ABONDDND LISTAGEM
IF(TY.EQ.1)RETURN
GO TO 20
30 IF(L.EQ.1)GO TO 60
CURSOR=1
CALL INSIRA(0,1,BUFFER)
RETURN
C LISTAGEM SOBRE A FITA DE PAPEL OU TELETIPO
40 CALL WRITE(BUFFER,TY,IFBEG)
IFBEG=0
GO TO 20
60 IF(TY.NE.-5)RETURN
TY=100
CALL WRITE(BUFFER,TY,IFBEG)
RETURN
END
C *****
SUBROUTINE REMOVA(K1,K2,BUFFER)
C ESTA SUBROUTINE REMOVE UM ARQUIVO DO CATALOGO,UMA LINHA
C DO TEXTO OU UM TRECHO DO PROGRAMA DO USUARIO.
C K1 E K2 =LINHAS QUE INCLUSIVE SERAO REMOVIDAS
C SE K1 OU K2 NAO ESTIVER PRESENTE O ARQUIVO SERA REMOVIDO
C *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,CDM,ARQ,CON,SET,LIS,INS,SETOR,PTR
COMMON KK,L1,L2,TY
COMMON /GR2/SETOR1
DIMENSION BUFFER(1)
IF(K1.EQ.1)GOTO 100
C PROCURA O REGISTRO DESEJADO
CALL SEGUIR(K1=2,BUFFER,NSECT,EREG)
CALL SEGUIR(K2,BUFFER,NS,ER)
CALL ESDISC(0,NSECT,EREG,BUFFER)
CALL CBND(C(BUFFER,NS,ER)
CALL ESDISC(1,NSECT,EREG,BUFFER)
RETURN
100 CONTINUE

```

```

C     FAZ A LIGACAO FINAL DOS NOS RESTANTES
CALL  ELCIMA(BUFFER,NSECT,EREG,1)
CALL  SEGUIR(K2,BUFFER,NS,ER)
CALL  CBNDNC(BUFFER,NS,ER)
CALL  ESDISC(1,0,1,BUFFER)
IF(ER.GE.0) RETURN
CALL  CBNDNC(BUFFER,SETOR1,0)
CALL  ESDISC(1,0,3,BUFFER)
CALL  CBNDNC(BUFFER,1,1)
CALL  ESDISC(1,0,2,BUFFER)
RETURN
END

C     *****
C     SUBROUTINE AGRUPA(WORK,K)
C     ESTA SUBROUTINA FAZ A CONCATENACAO DOS NUMEROS.
C     *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,      SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION WORK(1)
KK=0
Z=K
DO 5 L=1,Z
A=1
EXP=Z-L
20  IF(EXP.EQ.0) GOTO 10
A=A*10
EXP=EXP-1
GOTO 20
10  KK=KK+WORK(L)*A
5   CONTINUE
RETURN
END

C     *****
C     SUBROUTINE EXEC(BUFFER)
C     ESTA SUBROUTINE ESCREVE NO ARQUIVO DE TRABALHO DO EDITOR
C     O ARQUIVO DO USUARIO GRAVANDO UM REGISTRO POR SETOR
C     QUE E O FORMATO ACITO PELOS COMPILADORES FORTD E ASS2
C     *****
IMPLICIT INTEGER(A-Z)
COMMON VIDEO(1280)
COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,SETOR,PTR
COMMON KK,L1,L2,TY
DIMENSION BUFFER(1)
C     INICIALIZA
CALL  ECRF(0,BUFFER)
10  CALL  ELCIMA(BUFFER,NSECT,EREG,1)
CONTINUE
IF(EREG.LT.0) GOTO 20
C     ESRF FAZ A GRAVACAO
CALL  ESDISC(0,NSECT,EREG,BUFFER)
CALL  ECRF(1,BUFFER)
C     ENDEREÇO DO PROXIMO REGISTRO
CALL  ENDSEG(BUFFER,NSECT,EREG)
20  GOTO 10
CONTINUE
CALL  ECRF(2,BUFFER)
RETURN
END

C     *****
C     SUBROUTINE ERRO(TIPO)
C     ESTA SUBROUTINA LISTA AS MENSAGENS DE ERROS NO VIDEO.
C     ERRO NUMERO 1 : INICIO DE COMANDO ESPERADO.
C     ERRO NUMERO 2 : COMANDO INEXISTENTE,NAO ARQUIVO.
C     ERRO NUMERO 3 : OVERFLOW NO ARQUIVO EM USO.
C     ERRO NUMERO 4 : OPERACAO INVALIDA.
C     ERRO NUMERO 5 : PRIMEIRO COMANDO DEVE SER UMA
C     ERRO NUMERO 6 : DEFINICAO DE ARQUIVO OU ATIVACAO
C     ERRO NUMERO 6 : CARACTER INVALIDO EM NOME DE AR -

```



```

C      QUIVO,CONTA DO SETORES.
C      ERRO NUMERO 7      :   ARQUIVO,CONTA OU SETOR MAIOR OU
C      ERRO NUMERO 8      :   MENOR QUE O PERMITIDO
C      ERRO NUMERO 9      :   AUSENCIA DO NUMERO DE SETORES ,
C      ERRO NUMERO 9      :   NUMERO DE CONTA OU PERIFERICO.
C      ERRO NUMERO 9      :   ARGUMENTO INVALIDO EM COMANDO.
C      *****
C      IMPLICIT INTEGER(A-Z)
C      COMMON VIDE0(1280)
C      COMMON CURSOR,COM,ARQ,CON,SET,LIS,INS,          SETOR,PTR
C      COMMON KK,L1,L2,TY
C      DIMENSION ME(8)
C      DATA ME/3*47,197,2*217,214,64/
C      ACIMA/TRES BELL E A PALAVRA ERRO
C      T=TIPO
C      CALL ESCR(21)
C      CALL MOVCUR(21)
C          DO 5 I=1,8
C              CALL ESCR(ME(I))
C              VIDE0(CURSOR)=ME(I)
C              CALL MOVCUR(61)
5      CONTINUE
C      CALL ESCR(T)
C      VIDE0(CURSOR)=T
C      CALL ESCR(13)
C      CALL MOVCUR(13)
C      CALL ESCR(21)
C      CALL MOVCUR(21)
C      RETURN
C      END
*      *****
*      ESTA SUBROUTINE FAZ ENTRADA/SAIDA SOBRE O DISCO.
*      CHAMADA CALL ESDISC(FLAG,NSETOR,EREG,BUFFER)
*      FLAG FOR O INPUT , CASO CONTRARIO OUTPUT
*      *****
LDIS  LDS
      RES 2
TARG  EQU
FLAG  RES 2          *SE ENTRADA OU SAIDA
NSETOR RES 2          *PARAMETROS
EREG   RES 2          *ENDERECO DO REGISTRO NO SETOR
BUFFER RES 2          *ENDERECO DO BUFFER
PBUF   DATA BUF
PT     RES 1          *VARIAVEL DE INDICE
XT1    RES 1          *VARIAVEL DE INDICE
CBL    EQU           *BLOCO DE INPUT
      DATA &20      *TABELA
      DATA,1 0
      DATA,1 M:EI *ETIQUETA
      DATA BUF
      DATA 256      *TAMANHO DO REGISTRO
      DATA 0
NSECTL RES 1
CBE    EQU           *BLOCO DE OUTPUT
      DATA &20      *TABELA
      DATA,1 &80
      DATA,1 M:EO *ETIQUETA
      DATA BUF
      DATA 256
      DATA 0
NSECTE RES 1
CMEI   DATA          &8005
CMEQ   DATA          &8006
BUF    RES 128        *BUFFER BLOC/DEBLOCAGEM
      FIN
ESDISC LPS LDIS
*      *****
*      LDE =16          *MOVER BAYTES ARGUMENTOS
*      MVS TARG
*      LDA =0

```

```

STA XT1
LDA JEREG
SUB =1
MUL =85
STA PT
LDA JFLAG
BNZ SAIDA *TESTA SE FLAG EQ 0 OU 1
*
ENTRADA
LDE =M:EI *ABERTURA SOBRE M:EI
CSV M:OPEN
LDA JNSETOR
STA NSECTL
LEA CBL *FAZ LEITURA DE UM REGISTRO
CSV M:ID
CSV M:WAIT
LDA CMEI
RET1 CSV M:CLOS *FECHA SOBRE M:EI
EQU
LDX PT
LBR JPBUF,X
ICX =1
STX PT
LDX XT1
SBR JBUFFER,X
XAX
ADD =1
STA XT1
CMP =85
BNE RET1
RTS
* SAIDA
SAIDA EQU
LDA JNSETOR *PREPARACAO PARA LEITURA
STA NSECTL
STA NSECTE
LDE =M:EI *FAZ ABERTURA SOBRE M:EI
CSV M:OPEN
LEA CBL
CSV M:ID
CSV M:WAIT
LDA CMEI
RET2 CSV M:CLOS *FECHA D ARQUIVO
EQU
LDX XT1
LBR JBUFFER,X
LDX PT
SBR JPBUF,X
ICX =1
STX PT
LDA =1
ADM XT1
CMP =85
BNE RET2
LDE =M:EO *ABERTURA SOBRE M:EO
CSV M:OPEN
LBR =840
LDX =255
SBR JPBUF,X
LEA CBE *ESCREVE UM REGISTRO
CSV M:ID
CSV M:WAIT
LDA CMEO
RET2 CSV M:CLOS *FECHA D ARQUIVO
RTS
FIN
END
*****
* SUBROUTINA PARA LER UM CARTAO OU FITA
* LE REGISTRO POR REGISTRO
* CHAMADA CALL READ(BUFFER,I)

```

```

* ****
LDRE LDS
RES 2
TARG EQU 2 *TABELA DE ARGUMENTOS
BUFFER RES 2
FL RES 2 *BLOCO DE LEITURA
CBLER EQU
DATA,1 0
DATA,1 0 M:SI
BUFFER RES 1 80
TESTE TEXT [%%] *FIM - FLAG
PBUF RES 1
READ LPS LDRE
* ****
LDE =8 *BAYTES A MOVER
MVS TARG
LDA BUFFER
STA BUFFER
LEA CBLER
CSV M:ID
CSV M:WAIT
* FITA - VERIFICA SE LINE-FEED
LDA BUFFER
STA PBUF
LDX =0
RET EQU
LBR ]PBUF,X
CMP =&0D *COMPARA FIM DE REGISTRO
BE S1
XAX
CMP =79 *ULTIMO CARACTER?
BE S2
XAX
ICX =1
BRU RET
* S1 IMPORTE=COLOCA BRANCO BUFFER FINAL
EQU
LBR =&40
SBR ]PBUF,X
XAX
CMP =79
BE S2
XAX
ICX =1
BRU S1 *CONTINUA COLOCANDO BRANCOS
S2 EQU
LEA TESTE *VERIFICA SE %%
XAE
LDA BUFFER
LDX =2
CSV M:CMPS
STA JFL *RESULTADO EM FLAG
RTS
FIN
END
* ****
* SUBROTINA QUE RECEBE UM CARACTER E ESCRE
* VE NO VIDEO E FAZ AFETACAO COM O UGF15
* CHAMADA CALL ESCR(I)
* ****
LDS2 LDS * PREPARACAO DA LDS
RES 2 * ENDEREÇO DE RETORNO
TARG EQU
X RES 2
CB DATA 0,&8082
DATA WK
DATA 1

```

```

WK      RES,1 1
IFBEG  DATA,1 1
K100   DATA E100          *NUMERO NECESSARIO AFETACAO
K3B2   DATA E3B2          *ENDERECO PARA NUMERO AFETACAO
ADR     RES 1
      FIN
ESCR    LPS  LDS2
*      *****
INI     LDE =4
      MVS TARG
LBR    IFBEG
BAZ    S1
*      UGF15 AFF
      LDA =0
      STA IFBEG          *FLAG QUE INDICA OK AFETACAO
      LDR =2
      STA ADR
      LDA K3B2
      SUB ADR
      STA ADR
      LDA K100
      STA JADR
S1      EQU
      LDA JX
      SBR WK
      LEA CB
      CSV M:IO
      CSV M:WAIT
      RTS
      FIN INI
      END
*      *****
*      FUNCAO QUE LER UM CARACTER E
*      TRANSMITE A UM PROGRAMA FORTRAN
*      CHAMADA CALL LER(O)
*      *****
LDS1    LDS          * PREPARACAO DA LDS
PTARG   RES 2
TARG    DATA PTARG *PONTEIRO TAB. ARGUMENTOS
X        EQU          * TABELA DOS ARGUMENTOS
CB       RES 2
      DATA 0,80082
      DATA WK
      DATA 1
      RES 1
      DATA TWB
      DATA TWB+4
      RES 4
      FIN
      LPS  LDS1          * PREPARACAO DA LPS
*      *****
INI     LDE =4          *NUMERO ARGUMENTOS
      LDX PTARG          *ENDERECO BUFFER
      CSV M:MOVE
      OLD          JPTWB1
      DST          PRES
      OLD          JPTWB2
      DST          PRES+4
      LEA          CB
      CSV M:IO
      CSV M:WAIT
      OLD          PRES
      DST          JPTWB1
      OLD          PRES+4
      DST          JPTWB2
      LBR WK
      RTS
      FIN INI
      END
*      *****

```

```

*          ESTA SUBROUTINE ESCREVE UM CARTAO NA TY OU PT.
*          CHAMADA CALL WRITE(BUFFER, TY, IFBEG)
*          ****
LDWR      LDS          RES          2
          EQU          *ARGUMENTOS- PARAMETROS
TARG      RES          2
BUFFER    RES          2
TY        RES          2
IFBEG     EQU          *FIM DOS PARAMETROS
CBESCR    EQU          **BLOCO ESCRITURA

M1        DATA       0
BUFFW     RES          1
NCARAC    RES          1
PTAMPF    RES          1
CBLFRC    EQU          *BLOCO ESCREVE LINE-FEED E RET. CARRO.
          DATA 0
          DATA &8081
          DATA LFRC
          DATA 2
          DATA &150D
          EQU
          DATA 0
          DATA &8081
          DATA LFRC
          DATA 1
          EQU
          DATA &10
          DATA 0
          DATA 4
          DATA &8081
          DATA &9007
          EQU          *ASSOCIA M:LD COM PT
          DATA,1    M:LD
          DATA,1    &94
          TEXT       [PT]
          DATA      0,0
          RES 1
          EQU          *ESCREVE BRANCOS INICIAIS
          DATA      0
          DATA      &8007
          DATA 0
          DATA 40
          DATA FIN
          TEXT [%]
          DATA &150D
          EQU          *FIM DE LEITURA (IMPRESSAO)
          EQU          *CARACTERES LF E RC
          FIN
WRITE     LPS          LDWR
*          ****
          =12 *BAYTES A MOVER PARA OS PARAMETROS
          MVS          TARG
          LDA          BUFFER
          STA          BUFFW
          ADD          =79
          STA          PTAMPF
          LDX          =0
          EQU
          LBR          JPTAMPF, X
          CMP          =&40
          BNE          S2
          DCX          =1
          BRU          RET1
          EQU
          XAX          =80
          ADD          NCARAC
          STA          BUFFER
          STA          PF
          LDE          =2
          LEA          LFRC          *GUARDA LFRC EM PF

```

```

MVS          JPF
LDA          MFI
STA          M1
LDA          JTY
CMP          =7
BE          TYPE
CMP          =100
BE          FPT

* FITA
LDA          JIFBEG
BAZ          S1
LEA          TASSPT
CSV         M:ASGN
LEA          CBS
CSV         M:IO
CSV         M:WAIT
S1          EQU
LDA          =2
ADM          NCARAC
LEA          CBESCR
CSV         M:IO
CSV         M:WAIT
RTS

* TELETYPE
TYPE        EQU
LDA          MTY
STA          M1
LDA          JIFBEG
BAZ          S3
LEA          CBP
CSV         M:IO
CSV         M:WAIT
LEA          CBD
CSV         M:DLAY
CSV         M:WAIT
S3          EQU          *ESCREVE UM REGISTRO
LDA          CBESCR
CSV         M:IO
CSV         M:WAIT
LEA          CBLFRC
CSV         M:IO
CSV         M:WAIT
RTS

* FIN FITA
FPT         EQU          *ESCREVE %%
LDA          PFIN
STA          BUFFW
LDA          =4
STA          NCARAC
LEA          CBESCR
CSV         M:IO
CSV         M:WAIT
LEA          CBS
CSV         M:IO
CSV         M:WAIT
RTS
FIN
END

*****
*          ESTA SUBROTINA FAZ ASSOCIACAO DAS ETIQUETAS
*          M:EI E M:ED COM O ARQUIVO. CHAMADA COMO
*          CALL PASGN(NOMA,CONTA,FL). NOMA E O NOME DO
*          ARQUIVO E FL INFORMA SE M:EI OU M:ED.
*          *****
LASGN      LDS
RES        RES 2
TARG      EQU
ANOM      RES 2          *ARGUMENTOS
ACONTA    RES 2          *NUMERO DA CONTA
FLAG      RES 2          *FL 0 OU FL 1

```

```

PNF      DATA  NF
MEI      DATA  &055C  *ETIQUE E MODE SOBRE M:EI
MED      DATA  &06DC  *ETIQUETA E MODE DE USO M:ED
TASS
MEID    EQU
RES      1
TEXT    [DME]
DATA    ,1  0
DATA    ,1  0
DATA    ,1  0
CLE      RES,1  1
NF       RES,3      *NOME DO ARQUIVO
FIN
PASGN   LPS  LASGN
*       *****
LDE     =12      *BAYTES A MOVER
MVS     TARG
DLD     JANOM   *CARREGA NOME ACC E EXTENSAD
DST     NF
LDX     =4
LDA     JANOM,X *PREPARACAO DO CB
STA     JPNF,X
LDA     JACONTA
SBR     CLE
LDA     JFLAG
BAZ     +3
LDE     MED
BRU     +2
LDE     MEI
STE     MEID
LEA     TASS     *FAZ ASSOCIACAO SOBRE M:EI OU M:ED
CSV     M:ASGN
STA     JFLAG   *RESULTADO EM FLAG
RTS
FIN
END
*       *****
*       ESTA SUBROTINA FAZ ABERTURA DE UM ARQUIVO. CHAMADA:
*       CALL PFILE(NOMA,CONTA,SETOR) . NOMA E O NOME DO AR-
*       QUIVO , DE UM A SEIS CARACTERES.
*       *****
LFILE   LPS
RES      2
TARG     EQU
ANOM     RES  2      *PARAMETROS
ACONTA   RES  2      *ENDERECO DA CONTA
ASETOR   RES  2      *ENDERECO DO SETOR
FIC      DATA  FICHER
TF       EQU
FICHER   RES  3      *NOME DO ARQUIVO
DATA     128
CONT     DATA,1  &18 *ETAT
SET      RES,1  1     *CONTA
RES      1          *SETOR
DATA    0,0
FIN
PFILE   LPS  LFILE
*       *****
LDE     =12      *BAYTES A MOVER
MVS     TARG
DLD     JANOM   *PREPARACAO DA TABELA ARQUIVO
DST     FICHER
LDX     =4
LDA     JANOM,X
STA     JFIC,X
LDA     JACONTA
SBR     CONT
LDA     JASETOR
STA     SET
LEA     TF      *CARREGA ENDERECO DA TABELA
CSV     M:FILE  *ABRE O ARQUIVO

```

```

RTS
FIN
END
* *****
* ESTA SUBROTINA FAZ COMPACTACAO DO NOME DO
* ARQUIVO, LIDO PELO PROGRAMA FORTRAN.
* *****
LNDMAR  LDS
RES 2
TARG    EQU
NOMC   RES 2
XT1    RES 1
XT2    RES 1
CONCAT  LPS LNDMAR
* *****
LDE    =4
MVS    TARG
LDA    =0
STA    XT1
STA    XT2
RET
LDX    XT1
LDA    ]NOMC,X
ICX    =2
STX    XT1
LDX    XT2
SBR    ]NOMC,X
XAX
ADD    =1
STA    XT2
CMP    =6
BNE    RET
RTS
FIN
END

C
C
C
* *****
* ESTA SUBROTINA PROCURA NO PROGRAMA EM DIS-
* CO, O ENDERECO DO SETOR PARA INSERCAO DA
* PROXIMA LINHA.
* CHAMADA CALL ENDSEG(BUFFER,NSECT,EREG)
* *****
LADS   LDS
RES 2
TARG   EQU          *TABELA DE ARGUMENTOS
BUFFER RES 2
ADSECT RES 2
ADREG  RES 2          *ULTIMO ARGUMENTO
CHAIN1 RES 3          *PARA O CONTEXTO
MINUS  DATA -1      *FLAG DE FIM
FIN
ENDSEG LPS LADS
* *****
LDE    =12          *BAYTES A MOVER
MVS    TARG
LDX    =80
OLD    ]BUFFER,X
DST    CHAIN1
LEA    CHAIN1      *TRANSFORMA O CONTEXTO EM BIBARIO
CSV    M:DCBN
STE    ]ADSECT
LDX    =84
LBR    ]BUFFER,X
CMP    =&FF
BE     FCHAI
AND    =&0F
STA    ]ADREG

```



```

FCHAI   RTS
        EQU          *FECHAMENTO
        LDA MINUS
        STA JADREG
        RTS
        FIN
        END

*
* *****
*
*   ESTA SUBROTINA CONVERTE A CONFIGURACAO BI-
*   NARIA DO CONTEXTO DO REGISTRO PARA DECIMAL,
*   CHAMADA CALL CBND(CBUFFER,ASETOR,ADREG)
* *****
*
LCBND   LDS
        RES 2
TARG    EQU 2          *TABELA DE PARAMETROS
BUFFER  RES 2          *ENDERECO BUFFER
ADSECT  RES 2          *ENDERECO DO SETOR
ADREG   RES 2          *ENDERECO DO REGISTRO EM DISCO
CHAI    RES 3
PCHAIN  DATA CHAI+1
BRANCO  DATA &4040,&4040
PT      RES 1          *INDICE
XT1     RES 1          *INDICE
        FIN
CBND    LPS LCBND
*
* *****
*
LDE     =12          *BYTES ARGUMENTOS A MOVER
MVS    TARG
LDE    JADSECT      *TRANSFORMA SETOR DE BN PARA DC
LEA    CHAI
CSV    M:BNDC
LDA    BUFFER
ADD    =80
STA    PT
ULD    BRANCO
DST    IPT
LDX    =0
LBR    IPCHAIN,X
CMP    =&40          *VERIFICA SE ULTIMO CARACTER
BNE    +3
ICX    =1
BRU    -4
STX    XT1
RET    EQU
LDA    =4
SUB    XT1
XAE
LDA    XT1
ADD    PCHAIN
MVS    IPT
S1     EQU
LDX    =4
LDA    JADREG
BAN    +4
IDK    =&F0
SBR    IPT,X
RTS
LBR    =&FF
SBR    IPT,X
RTS
FIN
END

*
* *****
*
*   ESTA SUBROUTINE FAZ ASSOCIACAO DE M:LO COM O
*   ARQUIVO DE TRABALHO DO EDITOR E ESCREVE %%FIN EDITOR
* *****
*
LFEDIT  LDS
        RES 2
CBE     EQU          *ESCREVE FIN EDITOR
        DATA 0

```

```

DATA,1 &80
DATA,1 M:LO
DATA LINE
DATA 14
TASS1 EQU
DATA,1
DATA,1 &1C
TEXT [DMC]
DATA 0,0
TEXT [ARQEDIE]
LINE EQU
DATA &150D*LINE-FEED RC
TEXT [%%FIN EDITOR]
MLO EQU
DATA,1 M:LO
DATA,1 &94
TEXT [TY]
DATA 0,0
FIN
FEDIT LPS LFEDIT
* *****
LEA TASS1
CSV M:ASGN
LEA MLO
CSV M:ASGN
LEA CBE
CSV M:IO
CSV M:WAIT
FIN M:EXIT
END

* *****
* ESTA SUBROUTINE FAZ ASSIGN COM A LEITORA DE CARTOES
* E FITA DE PAPEL PERFORAVEL(LEITURA)
* *****
LASSI LDS
RES 2
TARG EQU
TY RES 2 *FLAG CR OU PT ?
TASS EQU
DATA,1 M:SI
TEXT DATA,1 &14 *ETAT
CR RES 1
PT DATA 0,0
TEXT [CR]
TEXT [PT]
FIN
ASSI LPS LASSI
LDE =4 *BAYTES A MOVER
MVS TARG
LDA JTY
CMP =1
BE CCR *ASSOCIA COM LEITORA
LDA PT
STA TEXT
BRU ASGN
CCR EQU
LDA CR
STA TEXT
ASGN EQU
LEA TASS
CSV M:ASGN
RTS
FIN
END

* *****
* ESTA SUBROUTINE FAZ A DESCOMPACTACAO DO BUFFER
* PARA GRAVACAO EM DISCO. CHAMADA CALL COMPAC(BUFFER)
* *****
LDCOMP LDS

```

```

RES 2
TARG EQU 2 *ARGUMENTOS (PARAMETRO)
BUF1 RES 2 *VARIABLES AUXILIARES
BUF2 RES 2 *COMPACTADO
NCARAC RES 2 *NUMEROS DE CARACTERES A DESCOMPACTAR
XT1 RES 1
XT2 RES 1
FIN
COMPAC LPS LD COMP
* *****
LDE =12 *NUMERO DE BAYTES A MOVER
MVS TARG
LDA =0
STA XT1 *XT1 E XT2 SAO INDICES
STA XT2
RET EQU
LDX XT1
LDA ]BUF1,X
IC X =2
STX XT1
LDX XT2
SBR ]BUF2,X
XAX
ADD =1
STA XT2
CMP ]NCARAC *VERIFICA SE ULTIMO CARACTER
BNE RET
RTS
FIN
END
* *****
LECRF LDS
* *****
RES 2
TARG EQU
CODE RES 2
BUFFER RES 2
TASS1 EQU
DATA,1 M:LD
DATA,1 &9C
TEXT [DM[
DATA 0,0
TEXT [ARQEDIC
TASS2 EQU
DATA,1 M:LD
DATA,1 &94
TEXT [TY[
DATA 0,0
CB EQU
DATA &20
DATA,1 &80
ADBUF RES
DATA,1 M:LD
DATA 1
DATA 80
DATA 0,0
FIN
LECRF
* *****
LDE =8
MVS TARG
LDA ]CODE
BAZ C1
CMP =1
BE C2
* CLOSE +ASGN M:LD T:TY
LDA =M:LD
CSV M:CLOS
LEA TASS2
CSV M:ASGN
RTS

```

```

* OPEN +ASGN
C1      EQU          TASS1
        LEA          M:ASGN
        CS          M:M:LO
        CS          M:OPEN
        RTS

* WRITE
C2      EQU          BUFFER
        LDA          ADBUF
        STA          CB
        LEA          M:IO
        CS          M:WAIT
        CS          RTS
        FIN

*
* *****
* ESTA SUBROUTINE COLOCA CARACTERES NO VIDEO(IMAGEM TELA)
* *****
LCVID   LDS          RES 2
        RES 2          *TABELA DE ARGUMENTOS
TARG    EQU          *TABELA DE ARGUMENTOS
ADVID   RES 2
NL      RES 2
ADBUF   RES 2
CHAINEDATA,1 5
        RES,1 5          *FIM DOS ARGUMENTOS
XT1     DATA,1 840
XT2     RES 1
CBVID   EQU          *BLOCO ESCREVE NO VIDEO
        DATA 0
        DATA,1 880
        DATA,1 882
PCHAIN  DATA CHAINEDATA 6
        DATA 6
        FIN
CVIDEO  LPS LCVID
* *****
        LDE =12          *BAYTES A MOVER
        MVS TARG
        LDE JNL          *NUMERO DA LINHA
        LEA CHAINEDATA
        CSV M:BNDC
        LDA =0
        STA XT1          *XT1 E XT2 INDICES
        STA XT2
RET     EQU          *LOOP COLOCA CARACTERES VIDEO
        LDX XT1
        LBR JPCCHAIN,X
        ICX =1
        STX XT1
        LDX XT2
        STA JADVID,X
        ICX =2
        STX XT2
        LDA XT2
        CMP =10
        BNE RET
        LDX =2
RET2    EQU          *ARMAZENA NO VIDEO
        LDA JADVID,X
        CMP =840
        BNE S1
        LBR =8F0
        STA JADVID,X
        ICX =2
        BRU RET2
S1      EQU
        LDX =10

```

```

LBR = &40
STA JADVID,X
LDA JADBUF
BAZ SVISU

*
LDA =0
STA XT1
LDA =12
STA XT2
RET1 EQU *LOOP TRANSFERENCIA
LDX XT1
LBR JADBUF,X
ICX =1
STX XT1
LDX XT2
STA JADVID,X
ICX =2
STX XT2
LDA XT1
CMP =72
BNE RET1
FIN EQU
SVISU EQU
LDX =1
RET3 EQU
LBR JPCCHAIN,X
CMP = &40
BNE S2
LBR = &F0
SBR JPCCHAIN,X
ICX =1
BRU RET3
S2 EQU
LEA CBVID
CSV M:IO
CSV M:WAIT
BRU FIN
FIN EQU
END

*
*****
*
*   ESTA SUBROUTINA LISTA O PROGRAMA DO USUARIO
*   NO VIDEO CHAMADA CALL LVIDEO(BUFFER,I).
*
*****
LVID EQU
RES 2 *TABELA DE PARAMETROS
TARG EQU
BUFFER RES 2
LINE RES 2
CBV EQU *ESCREVE NO VIDEO
DATA 0
DATA,1 &80
DATA,1 &82
PTAMP DATA TAMP
NCARAC RES 1 *NUMERO CARACTERES A ESCREVER
PTAMPF DATA TAMP+79
CBLFRC EQU *ESCREVE LINE FEED E RC
DATA 0
DATA,1 &80
DATA,1 &82
DATA LFRC
DATA 2
LFRC DATA &150D
TAMP RES 39
TEXT [ ]
FIN EQU
LVIDEO LPS LVID
*
*****
*
*   *BAYTES A MOVER PARA PARAMETROS
LDE =8
MVS TARG

```

```

LDE =72
LDA BUFFER
MVS TAMP+6
LDE JLINE *CONVERTE LINHA DE BN PARA DC
LEA TAMP
CSV M:BNDC
LDX =1
RET EQU *ZERA RESTANTE
LBR JPTAMP,X
CMP =&40
BNE S1
LBR =&F0
SBR JPTAMP,X
ICX =1
BRU RET
S1 EQU
LBR =&40
SBR TAMP+5
LDX =0
RET1 EQU
LBR JPTAMPF,X
CMP =&40
BNE S2
DCX =1
BRU RET1
S2 EQU
XAX
ADD =80
STA NCARAC
LEA CBV *ESCREVE NO VIDEO
CSV M:IO
CSV M:WAIT
LEA CBLFRC *FAZ LF E RC
CSV M:IO
CSV M:WAIT
RTS
FIN
END

```

```

C *****
C ESTRUTURA DE OVERLAY DO PROGRAMA
C *****
%JOB
%C/BIB/COPY/UL.GO,(EDITOR)
%C/BIB/FETCH/SL,(F:INI1)
%C/BIB/FETCH/UL,(ESCR)
%C/BIB/FETCH/UL,(MOVCUR)
%C/BIB/FETCH/UL,(LER)
%C/BIB/FETCH/UL,(ERRR)
%C/BIB/FETCH/SL,(F:GOTO)
%C/BIB/FETCH/UL,(LISTE)
%C/BIB/FETCH/SL,(F:DENR)
%C/BIB/FETCH/SL,(F:ERFD)
%C/BIB/FETCH/SL,(MOD)
%C/BIB/FETCH/UL,(MOVEL)
%C/BIB/FETCH/UL,(AGRUPA)
%C/BIB/FETCH/UL,(PASGN)
%C/BIB/FETCH/UL,(CBNDC)
%C/BIB/FETCH/UL,(ESDISC)
%C/BIB/FETCH/UL,(ENDSEG)
%C/BIB/FETCH/UL,(SEGUIR)
%C/BIB/FETCH/UL,(ELCIMA)
%C/BIB/FETCH/UL,(CVIDEO)
%C/BIB/FETCH/SL,(F:CVIR)
%C/BIB/FETCH/SL,(F:CMPR)
%C/BIB/FETCH/UL,(COMAND)
%C/BIB/FETCH/UL,(NOMARQ)
%C/BIB/FETCH/UL,(NUMCON)
%C/BIB/FETCH/UL,(NSETOR)
%C/BIB/FETCH/UL,(CONCAT)
%C/BIB/FETCH/UL,(PFILE)

```

```

% C/BIB/FETCH/UL,(DEFINA)
% C/BIB/FETCH/UL,(ATIVE)
% C/BIB/FETCH/UL,(ARGREM)
% C/BIB/FETCH/UL,(DELETE)
% C/BIB/FETCH/UL,(REMOVA)
% C/BIB/FETCH/UL,(COPIE)
% C/BIB/FETCH/UL,(ARGLIS)
% C/BIB/FETCH/UL,(ARGINS)
% C/BIB/FETCH/UL,(INSIRA)
% C/BIB/FETCH/UL,(ASSI)
% C/BIB/FETCH/UL,(INCRPT)
% C/BIB/FETCH/UL,(READ)
% C/BIB/FETCH/UL,(COMPAC)
% C/BIB/FETCH/UL,(LISTC)
% C/BIB/FETCH/UL,(EXEC)
% C/BIB/FETCH/UL,(ECRF)
% C/BIB/FETCH/UL,(FEDIT)
% C/BIB/FETCH/UL,(LVIDED)
% C/BIB/FETCH/UL,(WRITE)
% FIN
% JOB
% C/LINKD/GI,GO,OV,(EDITOR)
% TREE/EDITOR+F:INI1+F:STP+F:STPN+F:REW+F:BKSP+F:ENDF+F:RECB;
% TREE/+ESCR+MOVCUR+LER+ERRO+F:GOTO+LISTE+F:DENR+F:FENR+F:ERFO+;
% TREE/MOD+MOVE+AGRUPA+PASGN+CBNDC+ESDISC=5EMBVG+B5GUR+ELCIMA+;
% TREE/CVIDED+F:CVIR+F:CMR(COMAND,NOMARQ,NUMCON,NSETOR,;
% TREE/CONCAT,PEFILE,DEFINA,ATIVE,ARGREM,DELETE,REMOVA,COPIE,ARGLIS;
% TREE/,ARGINS,INSIRA(ASSI;
% TREE/,INCRPT+READ,COMPAC),LISTC,EXEC+ECRF,FEDIT,LVIDED,WRITE)
% C/BIB/REP/.EP
% FIN

```







```

L DA          = 0
L STA        ML
L STA        ME
L STA        CPTA
L DR         = 2
L STA        *ENDEREÇO BASE G
L DA        ADR *ENCONTRA ENDEREÇO PARA ADR
L SUB       K3B2 *ENDEREÇO PARA COLOCACAO DO FLAG
L STA        ADR
L DA        = 0
L STA        JADR *DISCONNECT DO UGF15
L LEA       TASS1
L CSV      M:ASGN
L LEA       TASS2
L CSV      M:ASGN
L LEA       CBL *LEITURA PRIMEIRO SETOR DO CATALOGO
L CSV      M:IO
L CSV      M:WAIT
L DX       = 1
L BR       JPTAMP,X
L STA      NARQ *DETERMINACAO DO NUMEROS ARQUIVOS
L DA      = 4
L STA      XT1
RET
L EQU     XT1
L DA      JPTAMP,X
L BAZ     INCR *BRANCH SE CATALOGO VAZIO
L DA      = 1
L ADM     CPTA
L LEA     TAMP
L ADD     XT1
L XAE     NOME
L DA      = 6
L DX      M:CMPS - *COMPARA O NOME DO ARQUIVO
L CSV     FDEL
INCR
L BAZ     = 18
L EQU     XT1 *INCREMENTA PARA LEITURA NOVO ARQUIVO
L DA      K100 *COMPARA COM 256
L ADM     RET
L BNE     = 1
L DA      ME
L ADM     ML
L STA     CBL *LEITURA DO SETOR SEGUINTE
L LEA     M:IO
L CSV     M:WAIT
L BRU     RET-2
FDEL
L EQU     XT1 *DELETA
L DX     = 0
L DA     JPTAMP,X
L STA     = 10
L ICX     JPTAMP,X
L DA     SETOR
L STA     * DELETE = ESCREVE DO SETOR(SUPRESSAO DO NOME)
L LEA     CBE
L CSV     M:IO
L CSV     M:WAIT
L DA     = 0 *ENDEREÇO PARA O PRIMEIRO SETOR
L STA     ML
L STA     ME
L LEA     CBL *LEITURA DO PRIMEIRO SETOR
L CSV     M:IO
L CSV     M:WAIT
L DA     NARQ
L SUB     = 1
L DX     = 1
L SBR     JPTAMP,X
L DA     CPTA *TESTA SE ARQUIVO DESTRUIDO E ULTIMO

```

```
MSO      CMP
          BNE
          LDX
          LDA
          SUB
          STA
          EQU
          LEA
          CSV
          CSV
          EQU
          LDA
          STA
          RTS
          FIN
          END

          NARQ
          MSO
          = 2
          JPTAMP, X
          SETOR
          JPTAMP, X

          CBE
          M:IO
          M:WAIT

          K100 *AFETA CDM 0 UGF15
          JADR
```

1 e