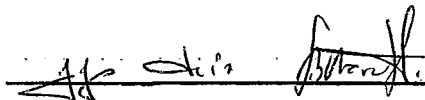


EDITOR DE TEXTOS PARA  
TERMINAL INTELIGENTE


Oswaldo Serra Alves Pereira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRA  
MAS 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:

  
Jayme Luiz Szwarcfiter  
presidente

  
João Lizardo Rodrigues Hermes de Araújo

  
Nelson Maculan Filho

PEREIRA, OSWALDO SERRA ALVES

Editor de Textos para Terminal Intelig  
gente Rio de Janeiro 1977.

XII, 91 p. 29,7 cm (COPPE-UFRJ, M.Sc,  
Engenharia de Sistemas e Computação, 1977)

Tese - Universidade Federal do Rio de  
Janeiro, Coordenação dos Programas de Pós  
Graduação de Engenharia.

I. Computadores I. COPPE/UFRJ II. Título  
(série).

A minha esposa.

Agradecimento

Aos professores JAYME LUIZ SZWARCFITER e  
GUILHERME CHAGAS RODRIGUES pela orientação recebida  
durante o desenvolvimento do trabalho.

## R E S U M O

O componente de "software" de um computador que visa eliminar a utilização de cartões perfurados (elemento custoso e não reusável) na transcrição de programas, arquivos de pequeno volume, cartões de controle ou texto livre é dito um Editor de Textos.

Esse trabalho descreve o projeto e a implementação de um Editor de Textos orientado para computadores de pequeno porte e que usa disco magnético, teclado e vídeo como unidades de entrada/saída. A descrição aborda os meios utilizados para racionalizar as tarefas de criação e atualização dos citados arquivos imagem de cartão e a metodologia usada no desenvolvimento do "software".

## A B S T R A C T

A computer software component which aims to eliminate the use of punched cards (an expensive and not re-usable element) in the transcription of programs, small sized files, control cards or free text is said to be a Text Editor.

This paper describes the project and implementation of a Text Editor oriented for small sized computers which uses magnetic disk, keyboard and video as input/output units. The discription relates the means used to rationalize the tasks of generating and updating card image files as well as the methodology used in software development.

## VII

### I N D I C E

1. Introdução .....	1
1.1 - O que é o Editor de Textos .....	2
1.2 - Comunicação Homem/Máquina .....	3
1.3 - Funções do Editor de Textos .....	8
2. Unidade de Vídeo .....	11
2.1 - Teclas de Controle .....	13
2.2 - Caracteres com função especial .....	14
3. Manipulação de Abreviaturas .....	15
4. Comandos .....	19
4.1 - Relação Alfabética dos Comandos .....	20
4.2 - Finalidade dos comandos .....	21
4.3 - Descrição sintática .....	22
4.4 - Descrição semântica .....	25
5. Organização dos Arquivos .....	31
5.1 - Descrição dos arquivos .....	32
5.2 - Descrição dos registros .....	36
6. Organização do Programa .....	56
6.1 - Descrição dos módulos .....	58
6.2 - Possibilidades de "Overlay" .....	64
6.3 - Comunicação entre os módulos .....	66
7. Implementação .....	70
7.1 - Características do PLTI sem similar no PL/I....	72
7.2 - Comandos PLTI escritos em PL/I .....	73
7.3 - Metodologia de conversão .....	82
7.4 - Metodologia de Programação.....	84
8. Conclusões .....	89

## 1 - INTRODUÇÃO

Esse trabalho, juntamente com vários outros desenvolvidos no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE/UFRJ), trata do desenvolvimento de "software" para seu Terminal Inteligente (TI/NCE).

No trabalho procura-se desenvolver um produto ao máximo independente do "hardware" onde ele possa vir a ser instalado, e em particular, atendendo às diretrizes estabelecidas para a produção do "software" para o TI/NCE.



## 1.1 - O QUE É O EDITOR DE TEXTOS

O Editor de Textos - EDTEX - é um programa de computador que permite em modo conversacional a criação e a atualização de um arquivo imagem de cartões. Esse arquivo imagem de cartões é chamado pelo Editor de Textos de Módulo e pode representar um programa, dados, cartões de controle e/ou texto livre. Cada registro de arquivo é dito uma Linha do Módulo.

O EDTEX usa um arquivo de acesso direto em disco para gravar o módulo, um teclado provido de teclas alfanuméricas e de controle para criá-lo ou atualizá-lo, e uma tela de vídeo para exibir o módulo, fazer eco dos dados digitados e exibir mensagens.

Na criação ou atualização de um módulo pode-se utilizar de qualquer das operações seguintes:

- incluir novas linhas
- excluir ou alterar (toda ou parte) linhas anteriormente incluídas
- rolar o módulo na tela para baixo ou para cima
- rearrumar as linhas do módulo
- criar e usar formas abreviadas
- sequenciar ou limpar as colunas finais do módulo
- juntar ao módulo processado um outro módulo
- estabelecer posições de tabulação

## 1.2 - COMUNICAÇÃO HOMEM/MÁQUINA

O EDTEX tem três modos de comunicação:

- Modo Seleção do "Menu"
- Modo Comando
- Modo Socorro

### MODO SELEÇÃO DO "MENU"

Nesse modo o EDTEX relaciona as tarefas ou opções que podem ser realizadas ou utilizadas e solicita ao usuário que faça a sua escolha. No local da resposta é colocada a resposta padrão ("default"), permitindo que se dê entrada nela ou se digite outra resposta.

Exemplo :

" E D T E X "    EDITOR DE TEXTOS

INDIQUE O NUMERO DA FUNCAO REQUERIDA :

- 1 - ESTABELECEER OPCOES DE PROCESSAMENTO
- 2 - CRIAR OU ATUALIZAR UM MODULO
- 3 - CONTINUAR UM SERVICO ANTERIORMENTE  
  TERMINADO NORMAL OU ANORMALMENTE
- 4 - TERMINAR O SERVICO

RESPOSTA : 1

## MODO COMANDO

Nesse modo o EDTEX recebe as ordens do usuário através de comandos e realiza todas as tarefas de criação ou atualização do módulo. É, portanto, o modo de comunicação que será mais usado em condições normais.

A tela do vídeo é subdividida em:

- Tela Rolante
- Linha de Comando ou de Trabalho
- Linha Mapa
- Linha de Tabulação
- Linha de Mensagem

### Tela Rolante

A tela rolante compreende as vinte linhas superiores da tela do vídeo e funciona como se fosse uma janela que pode exibir vinte linhas consecutivas quaisquer do módulo, isto é, uma tela na qual o módulo pode ser rolado para cima (do início para o fim) ou para baixo (do fim para o início). A linha inferior da tela rolante é dita base da tela rolante.

### Linha de Comando ou de Trabalho

Situa-se imediatamente abaixo da tela rolante e é a linha onde os comandos e novas linhas do módulo são escritos (entrada de dados pelo teclado).

### Linha Mapa

Situa-se imediatamente abaixo da linha de comando ou de trabalho e contém a numeração das colunas da tela (1 a 80), permitindo a visualização da posição horizontal do cursor e dos caracteres nas linhas do módulo.

### Linha de Tabulação

Situa-se imediatamente abaixo da linha mapa e exibe os campos de tabulação que compõem a linha de trabalho. Isso permite que se saiba para onde será deslocado o cursor quando se pressionar a tecla de tabulação horizontal.

### Linha de Mensagem

É a linha inferior da tela e é usada para exibir as mensagens de erro ou informativas e também o último comando executado pelo EDTEX.

A utilização da tela no modo indicado é consequência das seguintes diretrizes:

- aproximar as linhas de maior probabilidade de serem examinadas, do teclado e da mesa sobre a qual se coloca o vídeo e o teclado.
- prover meio de visualização da posição horizontal dos caracteres e do cursor.
- prover meio de visualização das posições de tabulação.
- aproximar a linha de comando ou trabalho da linha base da tela rolante.



## MODO SOCORRO

Nesse modo o EDTEX exhibe ao usuário explicações sobre como utilizá-lo. A cada página de explicações exibida, é perguntado se se deseja continuar recebendo explicações ou se se deseja voltar ao ponto onde foi solicitado o socorro.

O modo socorro é invocado pelo usuário ao digitar como primeiro caráter de uma resposta ou de um comando um ponto de interrogação. Nesses casos o EDTEX salva seu status corrente e exhibe a primeira página de explicações sobre sua utilização. As páginas subsequentes de explicações podem ser exibidas, a critério do usuário. Após a exibição da última página, se for solicitado que o EDTEX continue fornecendo socorro, volta a ser exibida a primeira página de explicações.

## OUTRAS FORMAS DE COMUNICAÇÃO

Os comandos e respostas tão logo dêem entrada são analisados pelo EDTEX. Se estiverem corretos provocam a emissão do som de acerto (menos potente) pela buzina ou alarme da unidade de vídeo. Se estiverem errados provocam a emissão do som de erro (mais potente).

Durante a execução das tarefas mais duradouras do Editor de Textos, o usuário é cientificado pela colocação de uma mensagem informativa na linha de mensagem da tela.

### 1.3 - FUNÇÕES DO EDITOR DE TEXTOS

A divisão do EDTEX em funções tem dois objetivos principais:

- 1 - permitir que se execute um "restart" e se atualize o módulo em uma única chamada de programa.
- 2 - permitir que se trate como um módulo um "job" completo tipo "compile & go" usando-se alternadamente facilidades do programa para manipular cartões de controle, linguagens de programação e dados de teste.

O EDTEX foi dividido em quatro funções principais:

- 1 - Estabelecimento de opções de processamento
- 2 - Criação ou atualização de um módulo
- 3 - Continuação de um serviço anteriormente terminado normal ou anormalmente
- 4 - Término do serviço.

No início do processamento e após o término de qualquer função, exceto da de número 4 que devolve o controle ao sistema operacional, é solicitado ao usuário informar a função que deseja executar. A opção do usuário de usar e reusar funções em diferentes sequencias permite que se atinja os dois objetivos anteriormente citados.

#### DESCRIÇÃO DAS FUNÇÕES

##### 1 - Estabelecimento das Opções de Processamento

Permite que se escolha opções, que permanecem válidas até o término do serviço ou então até que se escolha novas opções.

Existem opções para:

- modificar o delimitador de início de comando

- modificar o delimitador de início de abreviatura
- selecionar uma linguagem juntamente com seu conjunto de abreviaturas padrão
- limpeza das colunas 73 a 80 do módulo no início da atualização do mesmo
- sequenciação das colunas 73 a 80 do módulo no término do serviço.

## 2 - Criação ou Atualização de um Módulo

Permite que se crie ou atualize um módulo através das seguintes operações:

- inclusão de novas linhas
- exclusão ou alteração (total ou parcial) de linhas anteriormente incluídas
- rolamento do módulo na tela para baixo ou para cima
- rearrumação das linhas do módulo em outra sequencia
- criação e uso de abreviaturas temporárias
- sequenciação ou limpeza das colunas finais do módulo
- junção ao módulo processado de um outro módulo
- estabelecimento de posições de tabulação.

## 3 - Continuação de um Serviço Previamente Terminado Normal ou Anormalmente

Permite "restart" automático. Inicialmente o EDTEX verifica se o serviço anterior terminou normalmente. Em caso afirmativo é aproveitado diretamente o conteúdo do arquivo de trabalho e evita-se os retardos necessários a sua formatação e transferência de dados do arquivo de entrada que contém o módulo a ser atualizado.

Quando o serviço anterior terminou anormalmente, o Editor de Textos tenta recuperar o arquivo de trabalho e informa



ao usuário o resultado dessa tentativa.

#### 4 - Término do Serviço

Transfere para um arquivo sequencial de saída o módulo contido no arquivo de trabalho. Caso este arquivo contenha um módulo que o EDTEX não conseguiu recuperar, é transferida para o arquivo de saída a porção recuperável do mesmo.

## 2 - UNIDADE DE VÍDEO

O teclado é o meio que o EDTEX usa para criar ou atualizar o módulo. Entretanto não se pode considerar o teclado como um meio de entrada independente do meio de saída tela. O EDTEX trata teclado e tela da unidade de vídeo como um dispositivo único através do qual homem e máquina dialogam.

Outro fator importante a se considerar são as acentuações das diferenças no "hardware" de unidades de vídeo. Existem as mais simples que transmitem os dados do teclado caracter a caracter, desprovidas de "buffer" para a tela e que fazem o refresh do TRC a partir da memória do computador. Existem também as unidades de vídeo que transmitem um bloco inteiro de informação, são providos de diversas teclas de controle que atuam diretamente sobre a tela, e que "entendem" um número razoável de comandos enviados pelo computador.

O EDTEX procura fazer que a maior parte de seu processamento seja independente da unidade de vídeo utilizada.

Considerou-se que haveria uma área de memória única, do exato tamanho da tela cujo conteúdo seria sempre idêntico ao exibido na tela no momento atual ou imediatamente seguinte. A transmissão de dados dessa área de memória para a tela, se necessária, seria feita por uma única subrotina que transmite toda a sua extensão.

Considerou-se também que a entrada de dados do teclado seria feita por uma única subrotina, que, se necessário, executaria as seguintes funções:

- colocar o campo de entrada em uma área predeterminada
- informar o tamanho em bytes do campo de entrada
- processar os caracteres de controle
- fazer eco dos dados digitados na tela
- atualizar a variável que contém a posição do cursor na forma linha, coluna

- não permitir que se escreva fora da delimitação do campo.

Um outro aspecto que devemos considerar são as teclas de controle. Sua presença é de grande importância para que o trabalho de criação e atualização do módulo possa ser realizado eficientemente. É bom frisar, entretanto, que a ausência de alguns controles não se constitui em fator impeditivo para o trabalho de digitação.

Os processamentos associados às teclas de controle podem estar no "hardware" da unidade de vídeo, na subrotina que faz a entrada de dados do teclado ou parte no "hardware" e parte na subrotina.

No item seguinte são descritas as teclas de controle consideradas relevantes para o Editor de Textos apresentado.

## 2.1 - TECLAS DE CONTROLE

- 1) SPACE - Coloca um carater branco na posição do cursor e a vança este uma posição para frente.
- 2) DUP - Substitui o carater da posição do cursor pelo situado na mesma coluna da linha imediatamente superior e a vança o cursor uma posição para frente.
- 3) ADVANCE - Avança o cursor uma posição para frente.
- 4) TAB - Avança o cursor para a próxima posição de tabulação a frente. Se não houver mais posições de tabulação adi ante no campo de entrada, o cursor vai para a últi ma posição do campo.
- 5) BACKSPACE - Movimenta o cursor uma posição para trás.
- 6) CARRIAGE RETURN - Movimenta o cursor para a primeira posi ção do campo de entrada.
- 7) CANCEL - Movimenta o cursor para a primeira posição do cam po de entrada e apaga todos os caracteres do mes mo.
- 8) INSERT - Desloca as posições do campo de entrada a direita do cursor, inclusive esta, uma posição para a di reita, desprezando o último carater do campo de en trada e incluindo um branco na posição do cursor. O cursor não muda de posição.
- 9) DELETE - Suprime o carater da posição do cursor, desloca a porção do campo de entrada à direita do cursor uma posição para a esquerda e insere um branco na últi ma posição do campo de entrada. O cursor não muda de posição.
- 10) ENTER - Transmite o campo de entrada para que seja processa do.

## 2.2 - CARACTERES COM FUNÇÃO ESPECIAL

Os caracteres com função especial são caracteres que modificam o significado do carater ou caracteres que o seguem, fazendo com que o EDTEX lhes dê um tratamento diferenciado.

### Delimitador de Início de Comando (=)

Quando identificado na primeira coluna da linha de comando ou trabalho, faz com que a mesma seja tratada como um comando. Quando o primeiro carater da linha é um outro carater qualquer a linha é considerada uma linha de dados.

Se houver necessidade de se criar linhas de dados com o sinal de igual na primeira coluna, previamente, ao se estabelecer as opções de processamento, deve-se substituir esse delimitador por um carater qualquer que não apareça na primeira coluna da linha de dados.

### Delimitador de Fim de Comando (.)

É o sinal pesquisado pelo "scanner" para detectar o fim do comando.

### Delimitador de Início de Abreviatura (!)

É o carater pesquisado nas linhas de dados que dão entrada para verificar-se se foram utilizadas formas abreviadas. Se for encontrado e houver uma forma extensa associada à abreviatura, esta é substituída por aquela.

Se nas linhas de dados houver o ponto de exclamação como carater do texto, o delimitador deve ser previamente trocado para outro carater que não apareça nas linhas de dados.

## 3 - MANIPULAÇÃO DE ABREVIATURAS

Ao executar-se a função de criação ou atualização de um módulo, cadeias de caracteres frequentemente usadas nas linhas de dados, podem ser escritas em uma forma abreviada, para serem expandidas pelo EDTEX. Para tal fim existem conjuntos de abreviaturas padrão que podem ser usados automaticamente. Tais conjuntos foram definidos em função das palavras chave de maior ocorrência nas linguagens de programação ou controle de "jobs" (ALGOL, COBOL, FORTRAN, PL/I, PLTI, IBM-OS, IBM-DOS, B-WFL). Ao indicarmos para o EDTEX a linguagem do módulo, automaticamente selecionamos e colocamos disponível para uso seu conjunto de abreviaturas padrão. Por sua vez, esse conjunto pode ser expandido com até 10 novas abreviaturas temporárias, mutáveis ao longo do tempo.

As abreviaturas são códigos de 2 caracteres, o primeiro ! (ponto de exclamação) e o segundo um caráter alfabético, nas abreviaturas padrão, e numérico nas abreviaturas temporárias. O caráter alfabético da abreviatura padrão coincide sempre com a letra inicial da forma extensa, que por sua vez pode ter no máximo 20 caracteres.

Embora o EDTEX suporte até 18 conjuntos de 26 abreviaturas padrão sua versão primeira dispõe dos seguintes:

## ALGOL

! A	alpha	! P	procedure
! B	begin	! R	real
! C	call	! S	skip
! D	data	! T	then
! E	else	! U	until
! F	format	! V	value
! G	go to	! W	while
! I	integer	! Z	zerodivide
! L	line		

## COBOL

! B	blank	! M	multiply
! C	compute	! N	numeric
! D	divide	! O	otherwise
! E	examine	! P	perform
! F	filler	! R	read
! G	greather	! S	subtract
! H	high-value	! V	value
! I	identification	! W	write
! J	justified	! Z	zero
! L	low-value		

## FORTRAN

! A	assign	! I	integer
! B	backspace	! L	logical
! C	continue	! R	read
! D	double precision	! S	subroutine
! F	format	! W	write
! G	go to		

## PL/I

! A	allocate	! L	locate
! B	begin	! O	output
! D	delete	! P	page
! E	else	! R	read
! F	format	! S	skip
! G	go to	! T	then
! H	high	! W	write
! K	keyfrom		

## PLTI

! A	address	! L	literally
! B	byte	! M	move
! C	case	! P	procedure
! D	declare	! R	read
! E	else	! S	seek
! F	file	! T	then
! G	go to	! W	write
! I	initial		

## IBM-OS

! B	blksize	! L	lrecl
! C	catlg	! M	msglevel
! D	delete	! P	pass
! E	exec	! R	recfm
! H	hold	! S	sysout
! I	indexed	! T	time
! J	joblib	! U	uncatlg
! K	keep		

## IBM-DOS

! A	assgn	! O	option
! D	dlbl	! P	pause
! E	extent	! R	reset
! I	include	! S	sysin
! L	lbltyp	! U	upsi



## B-WFL

! A	algol	! L	library
! B	begin	! N	name
! C	compile	! P	printer
! D	data	! R	reader
! E	end job	! S	stacklimit
! F	fortran	! T	title
! G	go to	! U	user
! I	iotime	! X	xalgol
! K	kind	! W	wait

## 4 - COMANDOS

Comandos são cadeias de até 80 caracteres iniciados por um delimitador de início de comando na coluna 1 e opcionalmente terminados por um delimitador de fim de comando. Brancos são ignorados, exceto quando pertençam a literais contidas no comando. Cada comando é identificado pelo primeiro caráter não branco que sucede o delimitador de início, este obrigatoriamente na coluna 1. O delimitador de início é o sinal de igual (=) e o delimitador de fim é o ponto (.).

Caso o sinal de = apareça na coluna 1 de uma linha do texto editado, o delimitador de início de comando pode ser trocado temporariamente. Quando o comando não contiver o delimitador de fim, será colocado um ponto implícito na posição 81 do mesmo.

Se a análise do comando indicar erro, seus caracteres são preservados, o cursor é colocado na 1.<sup>a</sup> posição de sua linha e a mensagem de erro apropriada é colocada na linha de mensagem.

O último comando executado é preservado internamente pelo editor para permitir sua reexecução, enquanto o fluxo de trabalho do programa permanecer compatível com o comando preservado. Tal fato é indicado pela transferência do comando, da linha de comando para a linha de mensagem, possibilitando, sempre, uma visualização do que ocorre.

## 4.1 - RELAÇÃO ALFABÉTICA DOS COMANDOS

- = A Altera a linha da base da tela rolante
- = B Rola o módulo para Baixo (no sentido do seu fim para seu início) uma ou mais linhas
- = C Rola o módulo para Cima (no sentido do seu início para seu fim) uma ou mais linhas
- = D Desloca a linha da base da tela rolante para o final do bloco de linhas deslocadas
- = E Exclui a linha da base da tela rolante
- = F Finaliza a criação ou atualização do módulo
- = G Gera uma nova abreviatura
- = I Inclui uma linha na base da tela rolante
- = J Junta ao módulo em processamento, imediatamente após a linha da base da tela rolante, o módulo contido no arquivo MODAUX total ou parcialmente
- = L Limpa (colocando brancos) as colunas 73 a 80 do módulo
- = M Modifica o módulo pela troca das cadeias de caracteres indicada
- = N Não executa a troca especificada no comando = M
- = R Reinclui, após a linha da base da tela rolante, o bloco de linhas deslocadas, preservando-o para inclusão em outro ponto do módulo, ou não
- = S Sequencia o módulo em suas colunas 73 a 80
- = T Estabelece posições de Tabulação
- = ? Fornece socorro operacional

## 4.2 - FINALIDADE DOS COMANDOS

- = A, = I, = E      permitem a alteração, inclusão ou exclusão de linhas do módulo
- = B, = C            permitem o rolamento do módulo para Baixo (do fim para o início) ou para Cima (do início para o fim)
- = D, = R            permitem a rearrumação das linhas do módulo em outra sequencia
- = G                  permite que seja estabelecida uma nova abreviatura temporária
- = J                  permite que seja anexado ao módulo um outro total ou parcialmente
- = L, = S            permitem que as colunas 73 a 80 do módulo sejam limpas ou sequenciadas
- = M, = N            permitem a pesquisa, com opcional modificação, de cadeias de caracteres em todo ou parte do módulo
- = T                  estabelece posições de tabulação
- = ?                  fornece socorro operacional

## 4.3 - DESCRIÇÃO SINTÁTICA

## SIMBOLOGIA :

$\{a \mid b\}$	- "a" ou "b"
$[a]$	- "a" opcional
,a ...	- sequência de "a" separados por vírgula
:=	- é definido por
maiúsculas e os <u>ca</u> racteres especiais	
= . ( ) : , ?	- símbolos terminais
minúsculas	- símbolos definidos no anexo

## COMANDOS :

- 1) Comando Altera := = A [texto antigo texto novo  
[(coluna inicial : coluna final)]]  
[.]
- 2) Comando Rola para Baixo := = B [{nº de linhas | I}] [.]
- 3) Comando Rola para Cima := = C [{nº de linhas | F}] [.]
- 4) Comando Desloca := = D [.]
- 5) Comando Exclui := = E [.]
- 6) Comando Finaliza := = F [.]
- 7) Comando Gera Nova Abreviatura := = G nº da abreviatura forma  
extensa [.]
- 8) Comando Inclui := = I [.]
- 9) Comando Junta := = J [(linha inicial : linha final)]  
[.]
- 10) Comando Limpa := = L [.]
- 11) Comando Modifica := = M texto antigo texto novo  
[A] [(coluna inicial : coluna final)]  
[, (linha inicial : linha final)] [.]
- 12) Comando Não Executa := = N [.]
- 13) Comando Reinclui := = R [P] [.]
- 14) Comando Sequencia := = S [.]
- 15) Comando Tabula := = T coluna inicial [, coluna inici  
al ...] [.]
- 16) Comando Socorro := = ? [.]

## ANEXO :

Texto antigo :=	texto de tamanho n com $1 \leq n \leq 30$
Texto novo :=	texto de tamanho n com $0 \leq n \leq 30$
Forma extensa	texto de tamanho n com $1 \leq n \leq 20$
Texto de tamanho n :=	cadeia de n caracteres precedida e sucedida por um delimitador. O delimitador é um qualquer dos caracteres especiais, ' " / + - * % > < {   , que não esteja presente na cadeia.
Coluna inicial :=	$1 \leq N^{\circ} \text{ inteiro} \leq 80$
Coluna final :=	$1 \leq N^{\circ} \text{ inteiro} \leq 80$ e $\geq$ coluna inicial
N <sup>o</sup> de linhas :=	$1 \leq N^{\circ} \text{ inteiro} \leq 32767$
N <sup>o</sup> da abreviatura :=	$0 \leq N^{\circ} \text{ inteiro} \leq 9$
Linha inicial :=	$0 \leq N^{\circ} \text{ inteiro} \leq 32767$
Linha final :=	$0 \leq N^{\circ} \text{ inteiro} \leq 32767$ e $\geq$ linha inicial

## 4.4 - DESCRIÇÃO SEMÂNTICA

= A [ texto antigo    texto novo [(coluna inicial:coluna final)]] [.]

Altera a linha da base da tela, manual ou automaticamente.

O modo manual é identificado pela ausência de parâmetros no comando e permite que o usuário reescreva na linha de comando a linha que substituirá a linha da base da tela quando for pressionado ENTER.

O modo automático é identificado pela presença obrigatória dos textos antigo e novo e opcionalmente pela presença das colunas inicial e final. Nesse modo, na linha da base da tela rolante, o texto antigo é pesquisado e, se encontrado, substituído pelo texto novo. Se forem especificadas colunas inicial e final, a pesquisa e troca não ultrapassam os limites especificados, enquanto que, quando não especificados, a pesquisa e troca abrangem a linha inteira.

Após a execução do comando a linha alterada permanece na base da tela.

= B [ {nº de linhas | I} ] [.]

Rola o módulo para baixo, isto é, no sentido de seu fim para seu início. O rolamento pode ser de uma ou mais linhas, ou então diretamente para o início do módulo, quando especificado o parâmetro I.

Se o número de linhas não for especificado é executado um rolamento de uma linha.

Se o número de linhas especificado ultrapassar o início do módulo, o rolamento cessa com a primeira linha do mesmo na base da tela, tal como ocorre quando especificado o parâmetro I.



= C  $\left[ \left\{ \text{nº de linhas} \mid F \right\} \right] \left[ \cdot \right]$

Rola o módulo para cima, isto é, no sentido do seu início para o seu fim. O rolamento pode ser de uma ou mais linhas, ou então diretamente para o final do módulo, quando especificado o parâmetro F.

Se o número de linhas não for especificado, é executado um rolamento de uma linha. Se o número de linhas especificado ultrapassar o final do módulo, o rolamento cessa com a última linha do mesmo na base da tela, tal como ocorre quando especificado o parâmetro F.

= D  $\left[ \cdot \right]$

Desloca a linha da base da tela para o final do Bloco de Linhas Deslocadas.

O Bloco de Linhas Deslocadas é formado por um conjunto de uma ou mais linhas, sucessivamente deslocadas, e que pode ser incluído em qualquer ponto do módulo.

O Bloco de Linhas Deslocadas foi definido para ser utilizado na rearrumação de um módulo. A linha seguinte à linha deslocada ocupa seu lugar na tela rolante, após a execução do comando. Se for deslocada a última linha do módulo a tela rolante gira uma posição para baixo.

= E  $\left[ \cdot \right]$

Exclui do módulo a linha da base da tela rolante.

A linha seguinte à linha excluída ocupa seu lugar na tela rolante, após a execução do comando. Se for excluída a última linha do módulo a tela rolante gira uma posição para baixo.

= F [.]

Finaliza a criação ou atualização do módulo, então novamente o EDTEX apresenta sua Tela Inicial para que o usuário indique ou tra função ou termine o serviço.

= G nº da abreviatura forma extensa [.]

Gera uma abreviatura nova ou substitui outra de mesmo número já existente.

Após sua geração, o aparecimento desse número, precedido do delimitador de início de abreviatura (!) faz com que haja substituição pela forma extensa em dois casos:

- na inclusão de uma linha
- na alteração de uma linha do modo manual.

= I [.]

Inclui uma nova linha imediatamente após a linha da base da tela rolante. A tela rolante gira uma linha para cima, ficando a linha incluída na base da tela.

= J [(linha inicial : linha final)] [.]

Junta ao módulo em processamento, imediatamente após a linha da base da tela, as linhas contidas no arquivo MODAUX, da primeira à última ou no intervalo especificado.

Após a execução do comando, a tela rolante permanece exibindo as mesmas linhas que eram exibidas antes da junção dos módulos. Se tiver sido especificado que as colunas 73 a 80 do módulo devem ser branqueadas, estas colunas também o serão nas

linhas de MODAUX juntadas ao módulo.

= L [.]

Limpa (coloca brancos) as colunas 73 a 80 do módulo, iniciando pela primeira linha e terminando na última.

= M texto antigo texto novo [A] [(coluna inicial : coluna final)] [, (linha inicial : linha final)] [.]

Modifica o texto antigo, substituindo-o pelo texto novo, em modo automático ou manual.

Se forem especificadas as colunas inicial e final, a pesquisa e troca não ultrapassam os limites especificados, enquanto que, quando não especificados, a pesquisa e troca abrangem a linha inteira. A pesquisa do texto antigo se faz girando continuamente para cima o módulo, a partir do seu início ou da linha inicial indicada.

O modo automático é identificado pelo parâmetro A e o modo manual pela ausência do mesmo. No modo automático a rotação para quando é detectado o fim do módulo ou a linha final. No modo manual, além da parada final, ocorre parada quando é encontrado o texto antigo. Nesse caso dispõem-se de três alternativas:

- realizar a modificação (ENTER)
- não realizar a modificação (= N. ENTER)
- não realizar a modificação e terminar a execução do comando = M (digitando-se outro comando).

No modo automático ou manual, quando ocorre uma parada pelo encontro do fim do módulo ou da linha final, existem duas alternativas:

- terminar a execução do comando = M (digitando-se outro comando válido)
- executar o comando = M novamente, a partir do início do módulo ou da linha inicial (ENTER)

= N [.]

Não executa a substituição do texto antigo pelo texto novo, quando o primeiro é encontrado durante a execução do comando = M.

= R [P] [.]

Reinclui, imediatamente após a linha da base da tela rolante, o Bloco de Linhas Deslocadas (BLD), preservando-o para inclusão em outro ponto do módulo, ou não.

O parâmetro P indica que o Bloco de Linhas Deslocadas deve ser preservado e sua ausência implica na destruição do BLD. Após a execução do comando, a tela rolante permanece exibindo as mesmas linhas que eram exibidas antes da reinclusão das linhas deslocadas.

= S [.]

Sequencia o módulo nas colunas 73 a 80 iniciando em 1 com incremento unitário.

O algarismo de mais baixa ordem da sequencia fica na coluna 80 e zeros não significativos não são suprimidos,

= T coluna inicial [, coluna inicial ...] [.]

Estabelece até 9 posições de tabulação para serem usadas associadamente com a tecla de controle TAB (horizontal tabulate).

A linha de tabulação é preenchida com os algarismos de 1 a 9 para indicar a divisão em campos da linha de comando.

Exemplo:

O comando = T 6, 7, 70, 73 faz com que a linha de comando fique dividida em 5 campos:

- 1.<sup>a</sup> - colunas 1 a 5
- 2.<sup>a</sup> - coluna 6
- 3.<sup>a</sup> - colunas 7 a 69
- 4.<sup>a</sup> - colunas 70 a 72
- 5.<sup>a</sup> - colunas 73 a 80

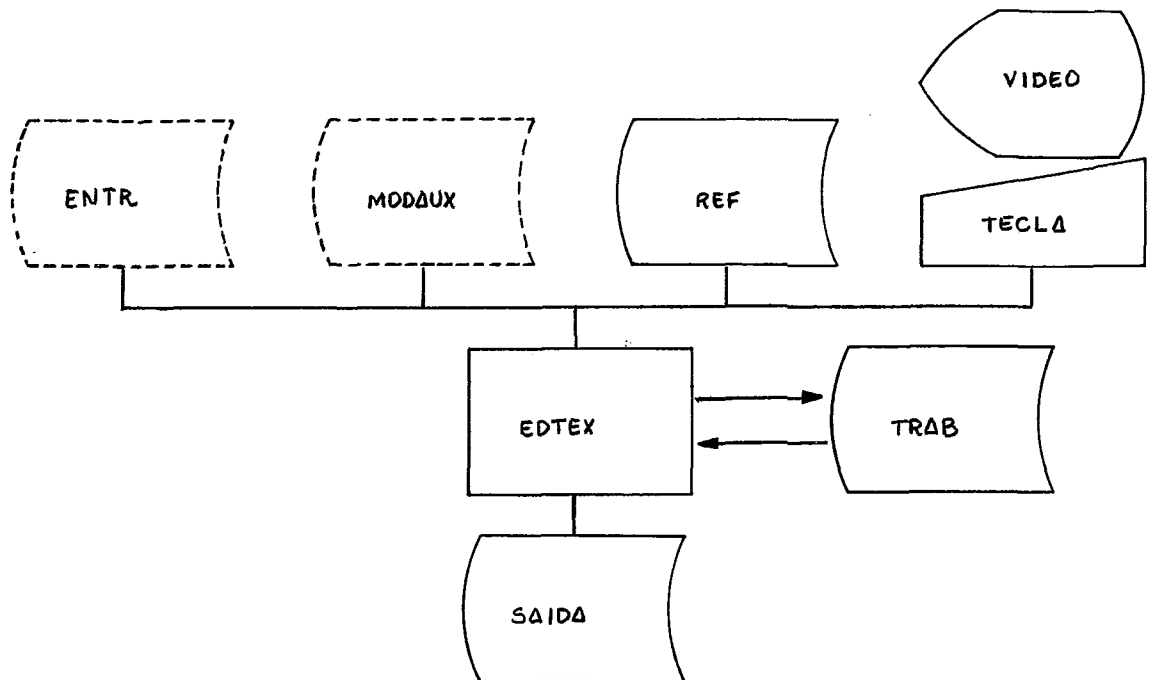
Nas linhas mapa e de tabulação ficará escrito:

1...5...10	}	..70....5...80
1111123333	}	33344455555555

= ? [.]

Salva o conteúdo da tela e exhibe a primeira página de explicação ao usuário. Ao final da página e perguntando se o usuário deseja continuar recebendo explicações ou quer retornar ao ponto onde foi solicitado socorro.

## 5 - ORGANIZAÇÃO DOS ARQUIVOS



## 5.1 - DESCRIÇÃO DOS ARQUIVOS

## ENTR - ARQUIVO DE ENTRADA

Arquivo sequencial, imagem de cartão, e que contém o módulo que vai ser atualizado.

É opcional pois não é utilizado quando vai se criar um módulo (não existe módulo para ser atualizado) ou quando vai-se continuar um serviço previamente terminado (é utilizado o módulo contido no arquivo TRAB).

## MODAUX - ARQUIVO MÓDULO AUXILIAR

Arquivo sequencial, imagem de cartão e que contém um módulo que pode ser anexado ao módulo que está sendo criado ou atualizado.

É opcional, pois pode-se criar ou atualizar um módulo sem anexar a ele um outro.

## REF - ARQUIVO DE REFERÊNCIAS

Arquivo de acesso direto e que contém 4 diferentes tipos de registros:

- 1 - Registros de abreviaturas, contendo as abreviaturas padrão e as formas extensas de todas as linguagens manipuladas pelo EDTEX.
- 2 - Registros contendo todas as mensagens emitidas pelo EDTEX.
- 3 - Registros contendo imagens das telas que são exibidas quando é solicitado socorro operacional.
- 4 - Um registro de controle que contém informações necessárias à manipulação dos demais registros.

Nota : É um arquivo que é distribuído pronto, juntamente

te com o programa.

#### TECLA - ARQUIVO TECLA

Arquivo em teclado, através do qual o usuário informa ao programa o que ele deseja que seja executado.

#### VIDEO - ARQUIVO VIDEO

Arquivo que o programa usa para informar ao usuário, as ações que são executadas.

#### TRAB - ARQUIVO DE TRABALHO

Arquivo de acesso direto onde são executadas todas as operações do EDTEX.

Esse arquivo é formatado no início do processamento, exceto quando se vai continuar um serviço anteriormente terminado (nesse caso, o arquivo foi formatado anteriormente).

O arquivo contém 7 diferentes tipos de registros:

1 - Registros de Controle (2) - contém apontadores, indicadores e delimitadores necessários à manipulação do arquivo.

Registro de Controle nº 1

- contém as informações que podem ser atualizadas durante a criação ou correção de um módulo (apontadores das listas simplesmente encadeadas).

Registro de Controle nº 2

- contém as informações que podem ser atualizadas apenas no início ou fim das funções principais do editor (equivalentes aos módulos secundários, citados no capítulo 5 - Organização do Programa).



- 2 - Registros de Abreviatura (9) - contém as abreviaturas padrão da linguagem selecionada e até 10 a breviaturas temporárias definidas pelo usuário.
- 3 - Registros de Linhas da Tela (24) - contém espaço para se salvar o conteúdo da tela quando é solicitado socorro operacional, permitindo a sua regeneração quando findo o socorro.
- 4 - Registros de Linhas do Módulo - formam uma lista duplamente encadeada provida de apontadores para seu início e fim. Essa lista pode ser percorrida para frente ou para trás.  
Qualquer elemento da lista pode ser alterado ou excluído. Após qualquer das linhas, podem ser incluídas novas linhas.
- 5 - Registros de Blocos de Linhas Deslocadas (BLD) - formam uma lista simplesmente encadeada provida de apontadores para seu início e final. É uma lista estruturada como uma fila:
  - as linhas são deslocadas do módulo para seu final
  - o BLD é reincluído no módulo a partir de seu início até o seu fim.
- 6 - Registros da Pilha de Registros Devolvidos (PRD) - formam uma lista simplesmente encadeada provida de um apontador para seu início. É uma lista estruturada como uma pilha:
  - um registro devolvido ao espaço disponível é incluído no início da PRD
  - quando solicitado um registro ao espaço disponível é fornecido o registro inicial da PRD.
- 7 - Registros não Utilizados - são os registros finais do arquivo, que se situam após o último registro utilizado pelo módulo ou pelo BLD.

## SAIDA - ARQUIVO DE SAIDA

Arquivo sequencial, imagem de cartão, para onde, ao fi  
nal de processamento é copiado o módulo contido no ar  
quivo TRAB.

## 5.2 - DESCRIÇÃO DOS REGISTROS

## ARQUIVO DE REFERÊNCIAS

Registro de Controle

É o único registro do arquivo que é acessado através de uma chave fixa (nº 1) e provê as informações necessárias ao acesso e manipulação dos demais registros do arquivo, que se colocam nas posições sequenciais seguintes.

## Tipo de registro

posição: 1

valor: '1' em codificação de caráter

descrição: identifica o registro de controle

## Número do registro de controle

posição: 2 a 3

valor: '01' em codificação de caráter

descrição: identifica o primeiro e único, no momento, registro de controle.

## Apontador do primeiro registro de mensagem

posição: 4 a 5

valor: a chave de acesso ao registro em codificação binária

descrição: contém a chave de acesso (número de ordem do registro) do primeiro registro de mensagem do arquivo.

## Carater delimitador

posição: 6

valor: o caráter delimitador em codificação caráter

descrição : identifica o carater delimitador de fim de forma extensa usado nos registros de abreviaturas. Esse delimitador é único para todas as linguagens e portanto não pode estar presente em qualquer forma extensa de qualquer linguagem.

Apontadores do primeiro registro de abreviatura das linguagens

posição : 7 a 42, sendo que cada apontador ocupa 2 posições consecutivas.

valor : a chave de acesso ao registro em codificação binária.

descrição : contém até 18 chaves de acesso (número de ordem do registro) do primeiro registro de abreviatura de 18 linguagens.

Os elementos do vetor que não apontam para outro registro contém o valor zero. Na  $i^{\text{ésima}}$  posição do vetor encontra-se o apontador da linguagem  $i$ .

Apontadores da primeira linha das páginas de socorro operacional

posição : 43 a 80, sendo que cada apontador ocupa 2 posições consecutivas

valor : a chave de acesso ao registro em codificação binária

descrição : contém até 19 chaves de acesso (número de ordem do registro) da primeira linha de cada uma das 19 páginas de socorro operacional.

Os elementos do vetor que não apontam para outro registro contém o valor zero. Na  $i^{\text{ésima}}$  posição do vetor encontra-se o apontador da página de número  $i$ .

Registros de Abreviaturas

Contém todas as abreviaturas e formas extensas pa  
drão de todas as linguagens.

## Tipo de registro

posição : 1

valor : '2' em codificação de caráter

descrição : identifica os registros de abreviaturas

## Código da linguagem

posição : 3 a 4

valor : o número da linguagem 01 a 18 em codificação de  
caráter

descrição : contém o número da linguagem, número esse que  
se mantém constante em todos os registros des  
sa linguagem. Coincide com a posição de seu a  
pontador no vetor de apontadores de primeiro  
registro de mensagem do Registro de Controle.

## Abreviaturas e formas extensas

posição : 6 a 80

valor : abreviaturas e formas extensas em codificação de  
caráter

descrição : define um conjunto de abreviaturas padrão, po  
dendo estas serem definidas em qualquer sequên  
cia dentro de cada linguagem.

Cada abreviatura é definida pela forma abrevi  
ada (uma letra) imediatamente seguida do sinal  
de igual e da forma extensa. Esta por sua vez  
é terminada pelo delimitador de fim de forma  
extensa.

São permitidos brancos (espaços) entre a defi  
nição das abreviaturas, porém não é permitido

que uma abreviatura comece a ser definida em um registro e termine no seguinte nem que um único registro contenha abreviaturas de duas linguagens.

### Registros de Mensagens

Contém todas as mensagens emitidas pelo EDTEX, sejam elas mensagens de erro ou informativas e advertenciais.

#### Tipo de registro

posição : 1

valor : '3' em codificação de caráter

descrição : identifica os registros de mensagem

#### Número da mensagem

posição : 3 a 4

valor : o número da mensagem em codificação de caráter

descrição : contém o número de ordem da mensagem

#### Código da mensagem

posição : 4 a 9

valor : o código identificador da mensagem ou codificação  
caráter

descrição : nas 3 primeiras posições identifica o tipo de mensagem com um dos vocábulos ERR, INF, ADV . Nas 3 posições seguintes identifica a mensagem por seu código numérico.

#### Texto da mensagem

posição : 6 a 80

valor: o texto explicativo da mensagem em codificação de carater

descrição: contém o texto explicativo da mensagem na forma mais explícita que o espaço permitir. Quando for o caso, esse texto juntamente com o código da mensagem são colocados na linha de mensagem da tela.

### Registros de Socorro Operacional

Contém todas as linhas de informação associadas ao socorro operacional que o EDTEX oferece.

Tipo de registro

posição: 1

valor: '4' em codificação de carater

descrição: identifica os registros de socorro operacional

Número da página

posição: 2 a 3

valor: número da página em codificação de carater

descrição: contém o número da página de socorro operacional, iniciando-se em '01' e podendo crescer até 19.

Número da linha

posição: 4 a 5

valor: número da linha em codificação de carater

descrição: contém o número da linha da tela para onde o texto informativo é levado. A numeração das linhas é feito de cima para baixo na tela, sendo sua linha do topo a de número 01. Se se quiser no texto explicativo deixar linhas em branco,

separando parágrafos do texto, basta que se in  
troduza quebra na sequência das linhas.

#### Texto informativo

posição : 6 a 80

valor : linha de texto informativo em codificação de car  
ter

descrição : contém uma linha de texto informativo que será  
colocada na linha da tela indicada no campo an  
terior quando exibida a página de socorro ope  
racional indicada no campo Número da Página.



# FORMATO DE ARQUIVO

FORMATO

Z	P	B
---	---	---

MEIO FÍSICO

C	F	D
---	---	---

CIC

F.B.
------

NOME DO ARQUIVO

REF

NOME DO REGISTRO

ZONTOLE

CIC

NOME DO REGISTRO

ABREVIATURAS

CIC

NOME DO REGISTRO

mensagem

CIC

NOME DO REGISTRO

SOCORRO OPERACIONAL

CIC

NOME DO REGISTRO

CIC

APONTADORES DO 1º REGISTRO DE ABREVIATURA DAS LINGUAGENS APONTADORES DA 1ª LINHA DAS PÁGINAS DE SOCORRO OPERACIONAL

1	2	...	18	1	2	...	19
---	---	-----	----	---	---	-----	----

ABREVIATURAS E FORMAS EXTENSAS

( ABREVIATURA = FORMA EXTENSA DELIMITADOR )

A = ASSIGN | B = BACKSPACE |

CÓDIGO DA MENSAGEM

TEXT O DA MENSAGEM

Nº DA LINHA

TEXT O INFORMATIVO

NOME DO REGISTRO

CIC

CIC	POSICÃO	COD.	SIGNIFICADOS	CIC	POSICÃO	COD.	SIGNIFICADOS	CIC	POSICÃO	COD.	SIGNIFICADOS	CONVENÇÃO
												FB-FATOR DE BLOCO
												B - BINARIO
												C - CARTÃO
												F - FITA
												D - DISCO
												Z - ZONADO
												P - COMPACTADO

---CAMPOS CODIFICADOS/SIGNIFICAÇÃO

## ARQUIVO DE TRABALHO

Primeiro Registro de Controle

Situa-se na primeira posição do arquivo e é acessado pela chave de número 1. Contém os apontadores das listas simplesmente encadeadas contidas no arquivo, qual sejam o Bloco de Linhas Deslocadas (BLD) e a Pilha de Registros Devolvidos (PRD). Esses apontadores podem ser constantemente atualizados durante o processo de atualização do módulo.

Nome do registro

posição : 1 a 2

valor : 1 em codificação binária

descrição : número de registro de controle

Tipo de registro

posição : 5

valor : '1' em codificação de caráter

descrição : identifica um registro de controle

Apontador do início do Bloco de Linhas Deslocadas (PIBLD)

posição : 6 a 7

valor : apontador de início de BLD em codificação binária

descrição : contém a chave do registro inicial do BLD. Se o BLD é um bloco vazio, conterá o valor 0. É atualizado quando incluído o primeiro registro no BLD e quando o BLD é devolvido ao espaço disponível.

Apontador do tipo da Pilha de Registros Devolvidos (PTPPRD)

posição : 8 a 9

valor : apontador do topo da PRD em codificação binária  
 descrição : contém a chave do registro topo da PRD. Se a PRD é uma pilha vazia, conterá o valor 0. É atualizado quando devolvido um registro ao espaço disponível e quando solicitado registro a espaço disponível e a PRD não está vazia.

OBS : As posições 3 a 4 e 10 a 80 não são utilizadas.

### Segundo Registro de Controle

Situa-se na segunda posição do arquivo e é acessado pela chave de número 2. Contém os apontadores e indicadores que são atualizados unicamente no início e fim da função de criação ou atualização do módulo (EDATU).

Número do registro

posição : 1 a 2

valor : 2 em codificação binária

descrição : número do registro de controle

Tipo de registro

posição : 5

valor : '1' em codificação de carater

descrição : identifica um registro de controle

Indicador de conteúdo do Arquivo de Trabalho (CNTRAB)

posição : 6

valor : '0', '1' ou '2' em codificação de carater

descrição : o valor '0' indica que o arquivo de trabalho não contém um módulo.

O valor '1' indica que o arquivo de trabalho contém um módulo que está sendo criado ou atualizado.

O valor '2' indica que o arquivo de trabalho contém um módulo completo, que já foi ou vai ser atualizado.

Apontador do início do módulo (PIMOD)

posição : 7 a 8

valor : apontador do registro inicial do módulo em codificação binária

descrição : contém a chave do registro inicial do módulo. Se o módulo é um conjunto vazio, conterá o valor 0. O apontador só estará atualizado ao iniciar-se e ao findar-se o processo de atualização do módulo (subrotina EDATU).

Apontador do fim do módulo (PFMOD)

posição : 9 a 10

valor : apontador do registro final do módulo de codificação binária

descrição : contém a chave do registro final do módulo. Se o módulo é um conjunto vazio, conterá o valor 0. O apontador só estará atualizado ao iniciar-se e ao findar-se o processo de atualização do módulo (subrotina EDATU).

Apontador do fim do Bloco de Linhas Deslocadas (PFBLD)

posição : 11 a 12

valor : apontador do final do Bloco de Linhas Deslocadas em codificação binária

descrição : contém a chave do registro final do Bloco de Linhas Deslocadas. Se o BLD é um bloco vazio, conterá o valor 0. O apontador só estará atualizado ao iniciar-se e ao findar-se o processo de atualização do módulo (subrotina EDATU).

Apontador do início da área não utilizada (INUTIL)

posição : 13 a 14

valor : apontador do início da área não utilizada do arqui  
vo de trabalho

descrição : contém a chave do primeiro registro não utili  
zado do arquivo. Note que os registros não uti  
lizados se situam na parte final do arquivo. O  
apontador só estará atualizado ao iniciar-se e  
ao findar-se o processo de atualização do môdu  
lo (subrotina EDATU).

Apontador do último registro do arquivo (PULTRG)

posição : 15 a 16

valor : apontador do último registro do arquivo

descrição : contém a chave do último registro físico do ar  
quivo. Seu valor é determinado nos procedimen-  
tos iniciais do EDTEX, precisamente na formataç  
ão do arquivo de trabalho, permanecendo imutá  
vel enquanto se atualiza esse arquivo.

Delimitador de início de comando (DEICOM)

posição : 17

valor : o carater delimitador de início de comando em codi  
ficação de carater

descrição : contém o carater delimitador de início de co  
mando. Seu valor é determinado na função de Es  
tabelecimento das Opções de Processamento. Ca  
so ela não tenha sido referenciada é tomado o  
valor padrão (default) = .

Delimitador de início de abreviatura (DEIABR)

posição : 18

valor : o caráter delimitador de início de abreviatura em formato de caráter

descrição : contém o caráter delimitador de início de abreviatura. Seu valor é determinado na função de Estabelecimento das Opções de Processamento. Caso ela não tenha sido referenciada é tomado o valor padrão (default) ! .

Número da linguagem (NLING)

posição : 19 a 20

valor : o número da linguagem do módulo em codificação binária

descrição : contém o número identificador da linguagem do módulo. É associado a uma lista de linguagens, que é apresentada ao usuário na função de Estabelecimento das Opções de Processamento. Caso essa função não tenha sido referenciada é tomado o valor padrão (default) 4, que corresponde à linguagem PLTI. O valor indicado ou padrão é utilizado para indexação do Vetor de Apontadores do Primeiro Registro de Abreviaturas das Linguagens, situado no Arquivo de Referências.

Indicador de limpeza re-sequenciação (CLEARR)

posição : 21

valor : '0' ou '1' em codificação de caráter

descrição : contém o caráter indicador de limpeza e re-sequenciação das colunas 73 a 80 do módulo. Se igual a '1', quando o módulo é trazido para o arquivo de trabalho, suas colunas 73 a 80 são limpas. Ao terminar-se o serviço, quando o módulo é transferido do arquivo de trabalho para o arquivo de saída, nessas mesmas colunas é colocada sequência identificadora das linhas, iniciando-se em 1 e com incrementos unitários. Se '0', não se realiza qualquer operação nas

colunas 73 a 80.

O valor desse identificador é determinado na função de Estabelecimento das Opções de Processamento. Caso essa função não tenha sido referenciada é tomado o valor padrão (default) '1'.

OBS : As posições 3 a 4 e 22 a 80 não são utilizadas.

### Registros de Abreviaturas

São nove registros que se situam nas posições 3 a 11 do arquivo. Contém as abreviaturas padrão e temporárias da linguagem do módulo, até 4 em cada registro e as padrão antecedendo as temporárias. As 36 abreviaturas que esses registros podem conter (26 letras e 10 algarismos) estão localizadas de acordo com a ordenação das letras e dos números. Assim, a forma extensa correspondente à letra E, estará no primeiro campo do segundo registro de abreviaturas. Do mesmo modo a forma extensa correspondente à abreviatura de número 7, estará no segundo campo do último registro de abreviaturas.

Número de registro

posição : 1 a 2

valor : número de registro (1 a 9) em codificação binária

descrição : número sequencial do registro de abreviatura

Tipo de registro

posição : 5

valor : '2' em codificação de caráter

descrição : identifica um registro de abreviatura

Forma extensa

posição : 6 a 86

valor : até 4 formas extensas, associadas a 4 abreviaturas  
 descrição : cada 20 posições consecutivas pode conter uma  
 forma extensa ou ser um espaço vazio desassociado a qualquer abreviatura. A forma extensa, se  
 houver, estará alinhada à esquerda do campo.

OBS : As posições 3 a 4 não são utilizadas.

### Registros de Linhas da Tela

São 24 registros consecutivos utilizados para salvar o conteúdo corrente da tela quando é solicitado socorro operacional. Isso permite a regeneração da mesma quando findo o socorro.

#### Número da linha

posição : 1 a 2

valor : o número da linha em codificação binária

descrição : contém o número sequencial da linha da tela (1 a 24), gravadas no sentido do tampo para a base.

#### Tipo de registro

posição : 5

valor : '3' em codificação de caráter

descrição : identifica um registro de Linha da Tela

#### Cópia da linha da tela

posição : 6 a 85

valor : cópia da linha em codificação de caráter

descrição : contém uma cópia exata da linha da tela exibida antes de solicitado o socorro operacional.

OBS : As posições 3 a 4 não são utilizadas.



Registros de Linha do Módulo

Formam uma lista duplamente encadeada providas de a pontadores para o seu início e fim, onde cada registro representa uma linha do módulo. Existirão tantos registros desse tipo quantas forem as linhas do módulo.

Elo à esquerda

posição : 1 a 2

valor : apontador do registro antecessor em codificação bi  
nária

descrição : contém a chave de acesso ao registro que con  
tém a linha antecessora (que precede) do módu  
lo. No registro da primeira linha, contém o va  
lor zero, indicando que não existe linha ante  
cessora.

Elo à direita

posição : 3 a 4

valor : apontador do registro sucessor em codificação binâ  
ria

descrição : contém a chave de acesso ao registro que con  
tém a linha sucessora (imediatamente seguinte)  
do módulo. No registro da última linha, contém  
o valor zero, indicando que não existe linha  
seguinte.

Tipo de registro

posição : 5

valor : '4' em codificação de carater

descrição : identifica um registro de linha do módulo

Linha do módulo

posição : 6 a 85

valor : a linha do módulo em codificação de carater

descrição : contém uma linha do módulo.

#### Registros do Bloco de Linhas Deslocadas (BLD)

Formam uma lista simplesmente encadeada, provida de apontadores para o seu início e fim, onde cada registro representa uma linha deslocada do módulo. Uma nova linha deslocada é incluída no final do BLD. Se for solicitada a reinclusão do BLD no módulo, este é reincluído a partir do seu início.

Elo à direita

posição : 3 a 4

valor : apontador do registro sucessor em codificação binária

descrição : contém a chave de acesso ao registro que contém a linha sucessora (seguinte) do BLD. No último registro do BLD, contém o valor zero, indicando que não existe linha sucessora.

Tipo de registro

posição : 5

valor : '5' em codificação de carater

descrição: identifica um registro do Bloco de Linhas Deslocadas (BLD)

Linha do Bloco de Linhas Deslocadas (BLD)

posição : 6 a 80

valor : a linha do BLD em codificação de carater

descrição : contém uma linha do BLD. Note que uma linha do BLD será sempre uma linha que anteriormente pertenceu ao módulo.

OBS.: As posições 1 e 2 não são utilizadas.

Registros da Pilha de Registros Devolvidos (PRD)

Formam uma lista simplesmente encadeada provida de apontador para seu início, onde cada registro apresenta uma linha devolvida no espaço disponível. Um novo registro devolvido é colocado no início da lista. Se solicitado um registro é fornecido o registro do início da lista.

Elo à Direita

posição: 3 a 4

valor: Apontador de registro sucessor em codificação binária

descrição: Contém a chave de acesso ao registro seguinte da PRD. No último registro da PRD, contém o valor zero, indicando que não existe registro sucessor.

Tipo de registro

posição: 5

valor: '6' em codificação de caráter

descrição: Identifica um registro da Pilha de Registros Devolvidos (PRD)

Linha da Pilha de Registros Devolvidos (PRD)

posição: 6 a 85

valor: A linha devolvida à PRD

descrição: Contém informação não utilizada (lixo). Note que uma linha da PRD anteriormente pertenceu ao módulo.

OBS.: As posições 1 e 2 não são utilizadas.

### Registros Não Utilizados

Formam um conjunto sequencial dos registros finais do arquivo de trabalho. Sempre que solicitado um registro ao espaço disponível e a PRD estiver vazia, é fornecido o primeiro registro não utilizado. O conjunto de registros não utilizados é provido de apontadores para seu início e fim. Seu fim coincide sempre com o último registro físico do arquivo.

#### Tipo de registro

posição: 5

valor: '7' em codificação de caráter

descrição: Identifica um Registro não Utilizado

#### Linha não Utilizada

posição: 6 a 85

valor: Espaços (brancos) em codificação de caráter

descrição: Contém espaço previamente alocado para ser utilizado por novas linhas do módulo ou do BLD.

OBS.: As posições 1 a 4 não são utilizadas.

# FORMATO DE ARQUIVO

NOME DO ARQUIVO		CIC	F.B.	MEIO FÍSICO		FORMATO	
TRAB				C	F	D	B
NOME DO REGISTRO	CONTROLE 1	CIC					
NOME DO REGISTRO	CONTROLE 2	CIC					
NOME DO REGISTRO	ABREVIATURAS	CIC					
NOME DO REGISTRO	LINHA DA TELA	CIC					
NOME DO REGISTRO	LINHA DO MÓDULO	CIC					

Nº DO REGISTRO	TIPO REG	PIBLD	PTPRD	PIMOD	FEMOD	PFLD	PNUTIL	PULTRG	DEICOM	DEIABR	NLINS	CLEAR
1												
2												

Nº DA LINHA	FORMA - EXTENSA (ALINHADA A ESQUERDA)	IDEN	IDEN	IDEN
1				
2				

Nº DA LINHA	CÓPIA DE LINHA DA TELA
1	
2	

Nº DA LINHA	LINHA DO MÓDULO
1	
2	

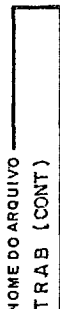
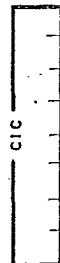
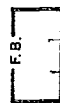
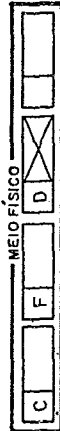
CIC	POSICÃO	COD.	SIGNIFICADOS	CIC	POSICÃO	COD.	SIGNIFICADOS	CIC	POSICÃO	COD.	SIGNIFICADOS

CONVENÇÃO	
FB-FATOR DE BLOCO	B - BINÁRIO
C-CARTÃO	
F-FITA	
D-DISCO	
Z-ZONADO	
P-COMPACTADO	

---CAMPOS CODIFICADOS/SIGNIFICAÇÃO

# FORMATO DE ARQUIVO



5

EL O A DIREITA

LINHA DO BLD (BLOCO DE LINHAS DESLOCADAS)

05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

6

EL O A DIREITA

LINHA DO PRD (PILHA DE REGISTROS DEVOLVIDOS)

05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

7

EL O A DIREITA

LINHA NÃO UTILIZADA

05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

CIC	POSICÃO COD.	SIGNIFICADOS	CIC	POSICÃO COD.	SIGNIFICADOS	CIC	POSICÃO COD.	SIGNIFICADOS	CONVENÇÃO
									FB-FATOR DE BLOCO
									B-BINÁRIO
									C-CARTÃO
									F-FITA
									D-DISCO
									Z-ZONADO
									P-COMPACTADO

\_\_\_\_CAMPOS CODIFICADOS/SIGNIFICAÇÃO

## 6 - ORGANIZAÇÃO DO PROGRAMA

O Editor de Textos (EDTEX) é composto por um programa principal e trinta e oito subrotinas, todos eles externos, isto é, podem ser compilados separadamente. Essa divisão baseou-se nas seguintes diretrizes:

- produzir um "software" que tenha os trabalhos de manutenção e extensão facilitados.
- produzir um "software" dotado de razoável grau de portabilidade e adaptabilidade.
- evitar a construção de módulos externos que trariam problemas de compilação em máquinas de pequeno porte.
- permitir a construção de estruturas de overlay com elevado índice de utilização de memória e pouca atividade de carga e recarga de módulos.

Cada módulo contém como comentário um prólogo do qual consta:

- descrição do módulo
- descrição dos parâmetros recebidos ou de retorno
- variáveis externas (globais) referenciadas
- subrotinas externas chamadas
- arquivos manipulados diretamente (pelos verbos READ,  
WRITE, REWRITE, SEEK)
- algoritmo utilizado

## EXEMPLO:

```

* PROCESS A,X,AC,NEST,GS,MAR(2,72,1),LC(59);                                00000010
/* */ EDEXCA : PROCEDURE(CAR1,TAM1,PT1,TAM2,PT2,N1,N2);/*ALTERA : =A*/00000020
/* EDEXCA *****/00000030
/* */00000040
/* DESCRICAO : */00000050
/* A SUBRTINA ALTERA A LINHA DA BASE DA TELA RCLANTE, ISTO E', */00000060
/* EXECUTA O COMANDO =A. */00000070
/* */00000080
/* DATA : 5 DEZ 1977. */00000090
/* */00000100
/* PROGRAMADOR : OSWALDC SERRA ALVES PEREIRA */00000110
/* */00000120
/* PARAMETROS : */00000130
/* CAR1 - CHAVE INDICADORA DA OPCAO DO COMANDO =A UTILIZADA */00000140
/* = '.' - OPCAO MANUAL, ISTO E', A ALTERACAO SE DARA PE-*/00000150
/* LA LETTURA DE NOVA LINHA. */00000160
/* = ' ' - OPCAO AUTOMATICA, ISTO E', A ALTERACAO SE DARA*/00000170
/* AUTOMATICAMENTE PELA SUBSTITUICAC DO TEXTO AN-*/00000180
/* TIGO PELO TEXTO NOVO. */00000190
/* TAM1 - TAMANHO EM BYTES DO TEXTO ANTIGO (TEXTO QUE E' PESQUI-*/00000200
/* SADO NA LINHA). */00000210
/* PT1 - APONTADOR DO TEXTO ANTIGO. */00000220
/* TAM2 - TAMANHO EM BYTES DO TEXTO NOVO (TEXTO QUE SUBSTITUI O */00000230
/* TEXTO ANTIGO, QD O MESMO E' ENCONTRADO NA LINHA). */00000240
/* PT2 - APONTADOR DO TEXTO NOVO. */00000250
/* N1 - COLUNA INICIAL DO CAMPO DE ALTERACAO (SE = 0 INDICA QUE */00000260
/* O CAMPO DE ALTERACAO E' A LINHA TODA) */00000270
/* N2 - COLUNA FINAL DO CAMPO DE ALTERACAO. */00000280
/* */00000290
/* VARIAVEIS EXTERNAS : */00000300
/* TL(*),PLSUP,ULIAR,CLIN,CCOL,PCCL,ENDT(*),ELOET(*),ELODT(*), */00000310
/* AREA(*),ERRO,DEICOM,CHVLE */00000320
/* */00000330
/* SUBROTINAS EXTERNAS : */00000340
/* EDMEMO,EDLMPT,EDMOV,EDLETC,EDIJ,EDABR3,EDEXBT,EDEST */00000350
/* */00000360
/* ALGORITMO : */00000370
/* 1 - TESTA SE NA BASE DA TELA RCLANTE EXISTE UMA LINHA DO */00000380
/* MODULO. */00000390
/* 2 - ALCCA AREA DE E/S */00000400
/* 3 - SE FOI UTILIZADA A OPCAO MANUAL DO COMANDO LE E PROCESSA */00000410
/* A NOVA LINHA. */00000420
/* 3.1 - MOVE COMANDO PARA LINHA DE MENSAGEM, LIMPA AREA DE EN- */00000430
/* TRADA, LINHA DE COMANDO (ONDE HA ECO DA ENTRADA), */00000440
/* EXIBE TELA E LE NOVA LINHA. */00000450
/* 3.2 - SE NAO FOI DIGITADA NOVA LINHA, INDICA EM ERRO E RETOR- */00000460
/* NA. */00000470
/* 3.3 - SE LINHA DIGITADA E' UM NOVO COMANDO, RETORNA PARA */00000480
/* EXECUTA-LC. */00000490
/* 3.4 - PROCESSA ABREVIATURAS NA NOVA LINHA. */00000500
/* 3.5 - MOVE INFORMACAO LIDA PARA AREA DE E/S. */00000510
/* 3.6 - LIMPA LINHA DE COMANDO E AREA DE ENTRADA DE DADOS, */00000520
/* NELAS COLOCANDO O COMANDO =A, */00000530
/* 4 - SE FOI UTILIZADA A OPCAO AUTOMATICA DO COMANDO, PESQUISA */00000540
/* O TEXTO ANTIGO, PARA SUBSTITUI-LO PELO TEXTO NOVO. */00000550
/* 4.1 - MARCA OS LIMITES DE PESQUISA. */00000560
/* 4.2 - MOVE LINHA DO MODULO P/ AREA DE E/S. */00000570
/* 4.3 - PESQUISA TEXTO ANTIGO, ENCONTRANDO-O, SUBSTITUI PELO */00000580
/* TEXTO NOVO. */00000590
/* 5 - ATUALIZA LINHA DO MODULO. */00000600
/* 6 - MOVE LINHA ATUALIZADA DO MODULO PARA BASE DA TELA. */00000610
/* 7 - LIBERA AREA DE E/S ALOCADA. */00000620
/* *****/00000630

```



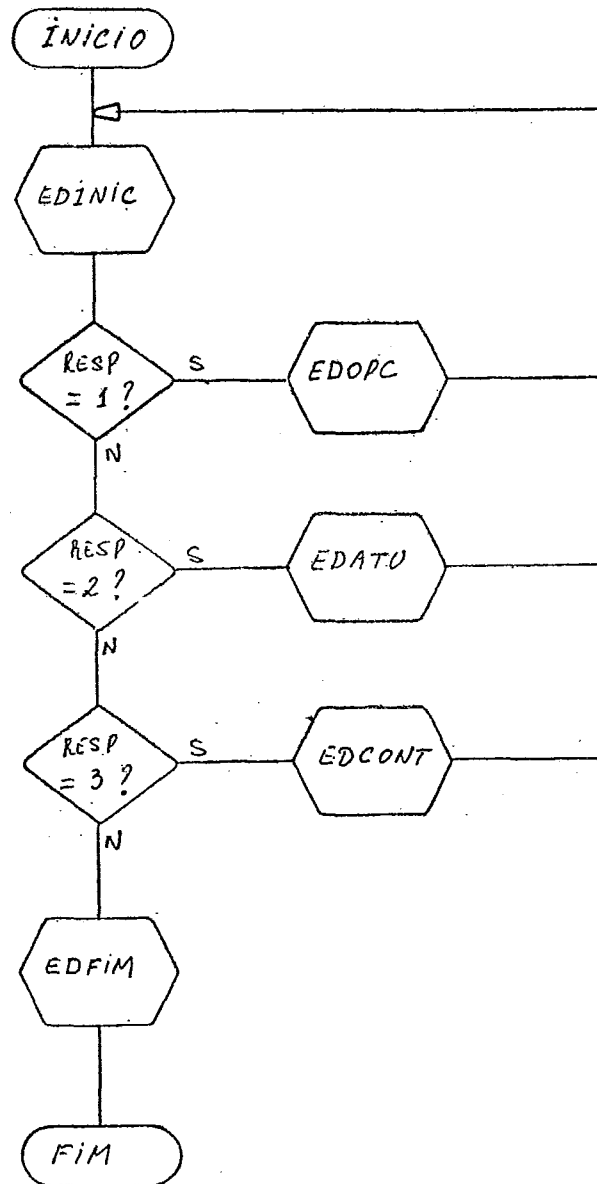
## 6.1 - DESCRIÇÃO DOS MÓDULOS

- 1) EDTEX - É o módulo raiz do programa e tem como funções receber o controle do sistema operacional, passar e receber o controle dos módulos secundários e devolver o controle ao sistema operacional após o término do serviço.

Os módulos secundários são cinco, e tem sua descrição a seguir:

- 2) EDINIC - Executa os procedimentos iniciais do Editor de Textos, tais como perguntar ao usuário a função que ele deseja executar, dele obter uma resposta válida e, se for o caso, formatar o arquivo de trabalho e transferir do arquivo de entrada para ele o módulo.
- 3) EDOPC - Estabelece as opções de processamento através de um diálogo mantido com o usuário.
- 4) EDATU - Atualiza o módulo, de acordo com os comandos emitidos pelo usuário até que seja digitado o comando Finalize (=F).
- 5) EDCONT - Executa os procedimentos que permitem a continuação de um serviço anteriormente terminado normal ou anormalmente.
- 6) EDFIM - Finaliza o serviço, transferindo o módulo contido no arquivo de trabalho para o arquivo de saída.

## EDTEX - FLUXO PRINCIPAL



Os demais módulos são ditos módulos de serviço e são descritos a seguir em ordem alfabética:

- 7) EDABR1 - A subrotina cria as abreviaturas padrão da linguagem indicada pela variável externa "NLING".
- 8) EDABR2 - Cria uma nova abreviatura, gravando opcionalmente a forma extensa no arquivo TRAB.
- 9) EDABR3 - A subrotina pesquisa a existência de abreviaturas ' na área de entrada. Encontrando, a substitui pela forma extensa. São atualizadas a área de entrada "AREA" e a linha da tela apontada pelo cursor.
- 10) EDATUC - Analisa, interpreta e executa os comandos do EDTEX.
- 11) EDATUI - Executa os procedimentos iniciais da atualização ' (criação ou correção) do módulo.
- 12) EDCVCN - A subrotina converte uma cadeia de caracteres numé<sup>u</sup>ricos sem sinal para um número binário de 15 bits.
- 13) EDCVNC - A subrotina converte um número binário positivo de 15 bits em uma cadeia de caracteres numéricos.
- 14) EDDEVE - A subrotina inclui no topo da pilha de registros devolvidos (PRD) o registro cujo endereço está con<sup>u</sup>tido na estrutura "REG" apontada por PT.
- 15) EDEBLD - A subrotina devolve o registro inicial do bloco de linhas deslocadas (BLD) ao espaço disponível.
- 16) EDEMOD - A subrotina exclui do módulo o registro contido na estrutura apontada por PT1.  
Se as estruturas apontadas por PT2 e/ou PT3 já contiverem os registros evita-se a leitura de 1 ou 2 registros do módulo.

- 17) EDEST - A subrotina lê e atualiza um registro do arquivo TRAB.
- 18) EDEXBT - Exibe o conteúdo e a área de memória denominada te la.
- 19) EDEXCA - A subrotina altera a linha da base da tela rolante, isto é, executa o comando =A.
- 20) EDEXCD - A subrotina desloca a linha da base da tela para o final do Bloco de Linhas Deslocadas (BLD), isto é, executa o comando =D.
- 21) EDEXCE - A subrotina exclui a linha do módulo situada na ba se da tela rolante, isto é, executa o comando =E.
- 22) EDEXCG - A subrotina gera uma abreviatura temporária.
- 23) EDEXCI - A subrotina inclui no módulo as linhas digitadas. A inclusão cessa quando digitado um delimitador de início de comando ou quando detectado algum erro.
- 24) EDEXCJ - A subrotina executa o comando =J.
- 25) EDEXCM - A subrotina executa o comando =M.
- 26) EDEXCR - A subrotina executa o comando =R.
- 27) EDHELP - A subrotina fornece socorro operacional ao usuário, salvando o status atual, exibindo as páginas de explicação e restaurando o status primitivo.
- 28) EDIJ - A função retorna o índice linear correspondente a te la (linha,coluna). Se passados valores inválidos, é retornado o índice linear equivalente a última colu na da última linha.

- 29) EDINCB - A subrotina inclui no final do Bloco de Linhas Deslocadas (BLD) o registro contido na estrutura "REG" apontada por PT1.
- 30) EDINCM - A subrotina inclui no módulo o registro contido na estrutura apontada por PT1.  
Se as estruturas apontadas por PT2 e/ou PT3 já contiverem os registros evita-se a leitura de 1 ou 2 registros do módulo.
- 31) EDLETC - A subrotina lê um campo, colocando-o na area "AREA", fazendo eco na tela e informando seu tamanho.
- 32) EDLMPT - A subrotina limpa a porção da tela situada entre as linhas indicadas (inclusive estas).
- 33) EDMEMO - A subrotina aloca ou libera areas em um "POOL" de memória. A area alocada sempre se inicia em limite de palavra dupla.
- 34) EDMOV - A subrotina move N bytes do endereço apontado por PTF p/ o endereço apontado por PTD.
- 35) EDMSG - A subrotina coloca na linha de mensagem da tela a mensagem cujo número está contido no iesimo elemento do vetor NUMMSG.
- 36) EDREQE - A subrotina fornece o endereço de um registro do arquivo TRAB disponível para uso. Se não houver registro disponível é fornecido o endereço zero.
- 37) EDSOM- A subrotina aciona a buzina da unidade de vídeo.
- 38) EDXBC - A subrotina rola o módulo para cima (do inicio para o fim) ou para baixo (do fim para o inicio) "N" linhas ou então o rola diretamente para seu inicio ou fim.
- 39) EDXLS - A subrotina limpa (coloca brancos) ou sequencia as colunas 73 a 80 do módulo.

Na implementação do programa TI/NCE, são utilizadas diretamente pelo programa 3 subrotinas constantes do "software" básico do mesmo: CONSOL, VIDEO e LOAD, utilizadas respectivamente para ler caracteres do teclado, manipular a tela da unidade de vídeo e carregar módulos em "overlay".

Além dos trinta e nove módulos componentes do Editor de Textos, foram desenvolvidos dois programas independentes:

EDCREF - Cria e lista o Arquivo de Referências (EDREF) do Editor de Textos (EDTEX). A criação se dá pela leitura de cartões e gravação em disco, executando-se as conversões necessárias de caráter para binário.

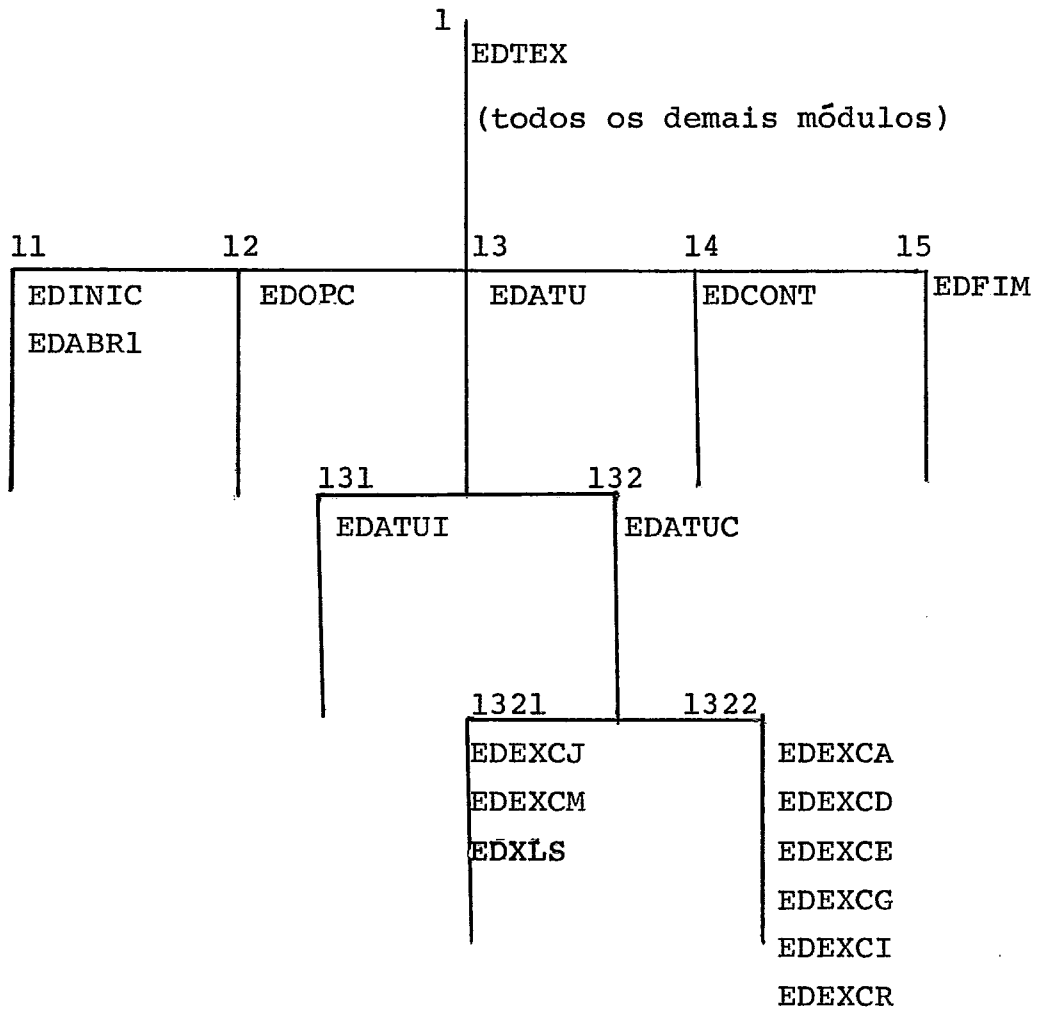
EDLTRAB - O programa, se solicitado, lista sequencialmente o Arquivo de Trabalho (EDTRAB) do Editor de Textos (EDTEX), identificando os registros e seus campos.

## 6.2 - POSSIBILIDADES DE "OVERLAY"

O EDTEX pode ter seus módulos organizados de modo a permitir que porções de memória possam ser utilizadas em tempos distintos com finalidades distintas, reduzindo assim o espaço total que o programa necessita para ser executado e sem que o mesmo perca parte significativa de seu tempo com sucessivas carga e recarga de seus módulos.

A estrutura em árvore seguinte indica uma alternativa que reduz o espaço de memória necessário para o EDTEX rodar sem gerar excessiva atividade de carga e recarga dos módulos. Fica implícito também, que o programa pode ser executado com outras estruturas em árvore com mais ou menos nós, ou com os módulos arranjados linearmente. Entretanto, a consequência será maior utilização de memória ou maior lentidão do programa.

## Estrutura de "Overlay"



## NOTAS:

- 1 - O programa será mais extenso quando estiverem na memória os nós 1, 13, 132, 1322
- 2 - O nó 1322 reúne os módulos que executam os comandos rápidos.
- 3 - O nó 1321 reúne os módulos de execução dos comandos lentos.



### 6.3 - COMUNICAÇÃO ENTRE OS MÓDULOS

A comunicação entre os procedimentos externos do programa é feito por 3 meios:

- passagem de argumentos
- variáveis externas (globais)
- arquivo de trabalho (TRAB)

O conteúdo do arquivo de trabalho foi detalhado no capítulo 5 - Organização dos Arquivos. As variáveis globais foram definidas com a finalidade principal de evitar-se comunicação através de longas listas de argumentos. Os argumentos são utilizados nas comunicações onde é exigida uma maior independência entre o módulo e os dados manipulados.

As variáveis globais são variáveis estáticas definidas no procedimento raiz do programa. Elas estão logicamente organizadas em 5 grupos e são relacionadas adiante:

- variáveis associadas à manipulação da tela;
- variáveis de controle geral;
- variáveis associadas ao módulo;
- variáveis associadas à manipulação de comandos;
- variáveis associadas à manipulação de abreviaturas.



Cont. (1)

```

/*****
/*  VARIÁVEIS DE CONTROLE GERAL  */
*****/
DECLARE
DEICOM CHAR(1) EXT      INITIAL(' '), /*DELIMITADOR DE INICIO */
/*DE COMANDO           */
DEIABR CHAR(1) EXT      INITIAL(') '), /*DELIMITADOR DE INICIO */
/*DE ABREVIATURA       */
NLING BIN FIXED(15)EXT  INITIAL(4), /*NO. DA LINGUAGEM DO */
/*MODULO.O VALOR DEFAULT*/
/*CORRESPONDE AO PLTI  */
CLEARR CHAR(1) EXT      INITIAL('1'), /*INDICADOR CLEAR/RESEQ */
/*PARA AS COLUNAS 73 A */
/*80 DO MODULO. O VALOR */
/*DEFAULT CORRESPONDE A */
/*VERDADEIRO.          */
CNTRAB CHAR(1) EXT      INITIAL('0'), /*INDICADOR DO CONTEUDO */
/*DO ARQUIVO TRAB      */
/*='0' - NAO CONTEM UM */
/*MODULO               */
/*='1' - CONTEM UM MODU-*/
/*QUE ESTA SENDO      */
/*ATUALIZADO          */
/*='2' - CONTEM UM MODU-*/
/*LO COMPLETO         */
RESP CHAR(1) EXT        INITIAL('1'), /*CONTEM A RESPOSTA DADA*/
/*A PERGUNTA FORMULADA */
/*NA TELA INICIAL DO  */
/*EDITOR DE TEXTOS    */
REF1(0:79) CHAR(1) EXT , /*AREA ONDE E' LIDO, NO */
/*INICIO DO PROG, O PRI-*/
/*MEIRO REG DO ARQ "REF"*/
PVEZ CHAR(1) EXT        INITIAL('1'), /*CHAVE INDICADORA DA */
/*PRIMEIRA VEZ QUE A  */
/*SUBROTINA EDINIC E' */
/*CHAMADA             */
/* VETOR CUJOS ELEMENTOS CONTEM OS NUMERQS DAS MENSAGENS EMITI */
/* DAS PELO PROGRAMA. COM SEU AUXILIO AS MENSAGENS PODEM SER */
/* DEFINIDAS NA MEDICA QUE A PROGRAMAÇÃO EXIGE E POSTERIORMEN- */
/* TE NUMERADAS EM UMA SEQUENCIA LOGICA.                        */
NUMMSG(0:030) BIN FIXED(15) EXT  INITIAL(
0, /* 0- ELEMENTO NAO UTILIZADO */
1, /* 1- ERR ESPACO INSUFICIENTE PARA A FORMATAÇAO DO */
/* ( ARQUIVO TRAB */
2, /* 2- ERR A FORMATAÇAO DO ARQUIVO TRAB NAO SE COMP */
/* LETOU */
3, /* 3- ERR C ARQ TRAB NAO CONTEM ESPACC SUFICIENTE */
/* PARA CONTER O MODULO */
4, /* 4- ERR RESPOSTA INVALIDA PARA A PERGUNTA FORMUL */
/* ADA */
5, /* 5- INF C EDTEX ESTA EXECUTANDO FUNCOES INTERNAS */
6, /* 6- ERR PARAMETRO INVALIDO DO COMANDO */
7, /* 7- ERRO PARAMETROS EM EXCESSO NO COMANDO */
8, /* 8- ERR NUMERO INTEIRO INVALIDO */
9, /* 9- ERR LITERAL MUITO LONGA */
10, /* 10- ERR COMANDO NAO IDENTIFICADO */
11, /* 11- ERR COMANDC CCM ERRO DE SINTAXE */
12, /* 12- ERR NAO FOI DIGITADO UM COMANDO QUANDO DEVERIA */
/* SE-LO */
13, /* 13- ERR CCMANDO DIGITADO EM MOMENTO IMPRGRPIO */
14, /* 14- INF DE ENTRADA NO COMANDO OU DIGITE OUTRO */
15, /* 15- ERR NAO FOI DIGITADA NOVA LINHA QD DEVERIA SE-LO */
0);

```

```

/*****
/* - VARIÁVEIS ASSOCIADAS AO MÓDULO */
/*****
DECLARE
PIMOD  BIN FIXED(15) EXT      INITIAL(0), /*APONTADOR DO INÍCIO */
                                           /*DO MÓDULO */
PFMOD  BIN FIXED(15) EXT      INITIAL(0), /*APONTADOR DO FIM DO */
                                           /*MÓDULO */
PIBLD  BIN FIXED(15) EXT      INITIAL(0), /*APONTADOR DO INÍCIO */
                                           /*DO BLOCO DE LINHAS */
                                           /*DESLOCADAS (BLD) */
PFBLD  BIN FIXED(15) EXT      INITIAL(0), /*APONTADOR DO FIM DO */
                                           /*BLD */
PTPPRO BIN FIXED(15) EXT      INITIAL(0), /*APONTADOR DO TOPO DA */
                                           /*PILHA DE REGISTROS */
                                           /*DESVOLVIDOS (PRD) */
PNUTIL BIN FIXED(15) EXT      ,          /*APONTADOR DO PRIMEIRO*/
                                           /*REGISTRO DA ÁREA NÃO */
                                           /*UTILIZADA DO ARQUIVO */
                                           /*TRAB */
PULTRG BIN FIXED(15) EXT      ;          /*APONTADOR DO ÚLTIMO */
                                           /*REGISTRO DO ARQUIVO */
                                           /*TRAB */
/*****
/* VARIÁVEIS ASSOCIADAS A MANIPULAÇÃO DOS COMANDOS */
/*****
DECLARE
RET CHAR(1) EXT      ,          /*CÓDIGO DE RETORNO DA EXECUÇÃO DE*/
                                           /*COMANDOS */
ERRO  BIN FIXED(15) EXT      ,          /*VARIÁVEL QUE CONTEM O VALOR O QD*/
                                           /*O COMANDO NÃO CONTEM ERRO. QD HA*/
                                           /*ERRO, CONTEM O ÍNDICE DE "NUMMSG*/
                                           /*ASSOCIADO AO ERRO */
CHVLE CHAR(1) EXT      ,          /*CHAVE PARA CONTROLE DA LEITURA */
                                           /*DE DADOS */
                                           /*'S' PARA LER-SE NOVO COMANDO */
                                           /*'N' PARA NÃO LER-SE NOVO COMANDO*/
AREA(0:79) CHAR(1) EXT      ,          /*ÁREA DE LEITURA DE COMANDOS E */
                                           /*DADOS */
TABPOS(0:9)  BIN FIXED(15) EXT
              INITIAL(-1,79,-2,-2,-2,-2,-2,-2,-2);
/*****
/* CONTEM A POSIÇÃO FINAL DOS CAMPOS DE TABULAÇÃO, RE- */
/* LATIVA A UMA ORIGEM ZERO (COMO SE A LINHA VARIASSE */
/* DE 0 A 79). */
/* PODEM EXISTIR DE 1 A 9 CAMPOS DE TABULAÇÃO, NUMERA- */
/* DOS DE 1 A N. */
/* SE EXISTIREM MENOS DE 9 CAMPOS DE TABULAÇÃO, AS PO- */
/* SICOES FINAIS DE "TABPOS" CONTEM O VALOR -2. */
/* "TABPOS(C)" CONTEM O VALOR -1, "TABPOS(ÚLTIMO CAMPO)" */
/* CONTEM O VALOR 79 E "TABPOS(I)" CONTEM A ÚLTIMA PO- */
/* SICAÇÃO DO I-ÉSIMO CAMPO. */
/*****
/* VARIÁVEIS ASSOCIADAS A MANIPULAÇÃO DE ABREVIATURAS */
/*****
DECLARE
TAMEXT(C:35)
      BIN FIXED(15) EXT      ; /*CONTEM O TAMANHO DAS */
                                           /*FORMAS EXTENSAS O */
                                           /*Í-ESIMO ELEMENTO COR- */
                                           /*RESPONDE A Í-ESIMA */
                                           /*ABREVIATURA. QD O TA- */
                                           /*MANHO É ZERO INDICA- */
                                           /*SE QUE NÃO EXISTE */
                                           /*FOLHA EXTENSA */
DECLARE
ALFA(0:35)
      CHAR(1) EXT      INITIAL /*CONTEM A 26 LETRAS E OS 10 */
('A','B','C','D','E','F', /*ALGARISMOS DECIMAIS. É UTI*/
'G','H','I','J','K','L', /*LIZADO NA IDENTIFICAÇÃO DE */
'M','N','O','P','Q','R', /*ABREVIATURAS */
'S','T','U','V','W','X',
'Y','Z','0','1','2','3',
'4','5','6','7','8','9');

```

## 7. - IMPLEMENTAÇÃO

Como foi dito no Capítulo I, o EDTEX devia ser implementado obedecendo às diretrizes estabelecidas para produção de "software" para o TI/NCE. Uma delas prescreve o uso do PLTI como linguagem de programação.

O primeiro problema que se defrontou ao planejar-se a implementação do EDTEX foi a indisponibilidade de um compilador/interpretador PLTI. Na época existia apenas uma Versão Ø, em fase de testes no computador Burroughs B6700 do Núcleo de Computação Eletrônica da UFRJ.

A partir dessa situação projetou-se, devido ao grau de similaridade entre as duas linguagens, emular o PLTI com o PL/I, utilizando-se um equipamento qualquer que tivesse um compilador PL/I implementado. Posteriormente e oportunamente o Editor de Textos seria convertido para PLTI e implementado no Terminal Inteligente.

Foi selecionado um equipamento IBM da série 370, provido de Sistema Operacional OS e de um compilador PL/I Optimizing. Tal máquina não dispunha de terminais de vídeo e era operada unicamente em modalidade "batch" com multiprogramação. Conseqüentemente teve-se também que emular o teclado e a tela da unidade de vídeo por leitora de cartões e impressora durante os testes do programa.

A partir das decisões tomadas procurou-se:

- Determinar as características da linguagem emulada que não estão presentes na linguagem emuladora.
- Descrever na linguagem emuladora como as características da linguagem emulada são reproduzidas.

- Estabelecer uma metodologia de conversão de programas escritos na linguagem emuladora , ao máximo automática.
- Criar a metodologia de programação na linguagem emuladora que permitirá a produção de programas possíveis de conversão para a linguagem emulada.

### 7.1 - CARACTERÍSTICAS DO PLTI SEM SIMILAR NO PL/I.

O PLTI permite que variáveis do tipo apontador sejam manipuladas como variáveis aritméticas, na realidade ele não distingue apontadores de variáveis aritméticas. Esse fato, na maioria dos casos, não acarreta maior dificuldade de programação, pois a atribuição de endereços à posterior pode ser feita pela função (builtin) de nome ADDR no PL/I e similarmente no PLTI pelo operador de endereço @ . Convém ressaltar que tal prática geralmente resulta em programas mais documentados e mais facilmente entendíveis, bem como um código objeto eficiente na linguagem PLTI.

O PLTI permite que variáveis do tipo "byte" possam ser utilizadas como variáveis aritméticas. Como isso não era permitido no PL/I usado, utilizam-se apenas variáveis de 2 bytes para conter variáveis aritméticas. É evidente que essa prática aumenta o tamanho e tempo de execução dos programas, podendo em muitos casos atingir limites críticos.

NO PLTI foi implementado o comando CASE, operadores ' de deslocamento (shift) e o operador lógico XOR, tais características de uso bastante eventual, devem ser substituídas respectivamente por:

- ninhos de comandos IF;
- operações de multiplicação ou divisão por potências de 2;
- operações lógicas AND, OR e NOT.

No PLTI foi implementado o comando ABORT, que permite o retorno de uma subrotina diretamente para um rótulo de um procedimento mais externo. Esse comando, de utilização eventualíssima, inclusive porque quebra o nível estrutural do programa, não tem similar no PL/I.

Todas as demais características do PLTI tem uma similar correspondente no PL/I.

## 7.2 - COMANDOS PLTI ESCRITOS EM PL/I

Nesse item são descritos, por meio de exemplos, como os comandos PLTI podem ser escritos usando-se a sintaxe PL/I. Para efeito de simplificação, são omitidos os comandos onde existe igualdade sintática e semântica. Tal prática evidentemente não esgota o assunto, sendo inclusive necessário, para melhor compreensão desse item prévio conhecimento das linguagens PLTI e PL/I.

Programa Principal Sem Variáveis Externas (Globais)

```

PLTI          PROG: DO;
                .
                .
                .
                .
                .
                .
                .
                END PROG;

```

```

PL/I          PROG: PROC OPTIONS (MAIN);
                .
                .
                .
                .
                .
                END PROG;

```

Programa Principal Com Variáveis Externas (Globais)

```

PLTI          PROG: DO;
                DECLARE (A,B) BYTE GLOBAL;
                .
                .
                .
                .
                END;

```



```

PL/I      PROG:  PROC OPTIONS (MAIN);
          DECLARE (A,B) CHAR (1) EXT;
          .
          .
          .
          .
          END;

```

Subrotina Interna ou Externa Com Argumentos de Retorno

```

PLTI      SUB2:  PROCEDURE (ARG1, ARG2, ARG3)
          RESULT (ARG1, ARG2);
          .
          .
          .
          .
          END SUB2;

```

```

PL/I      SUB2:  PROCEDURE (ARG1,ARG2,ARG3);
          .
          .
          .
          .
          END SUB2;

```

Subrotina Externa Com Variáveis Externas

```

PLTI      [ V1 BYTE, V2 ADDRESS; ]
          SUB3:  PROCEDURE;
          .
          .
          .
          END SUB3;

PL/I      SUB3:  PROCEDURE;
          DECLARE V1 CHAR(1) EXTERNAL,
                   V2 BIN FIXED (15)EXTERNAL;
          .
          .
          .
          END SUB3;

```

Função (interna ou externa)

```

PLTI          FUNC:  PROCEDURE (ARG1,ARG2) BYTE;
               .
               .
               .
               .
               END FUNC;

```

```

PL/I          FUNC:  PROCEDURE (ARG1, ARG2)
               RETURNS (CHAR (1));
               .
               .
               .
               .
               END FUNC;

```

Se a função tiver argumentos de retorno ou variáveis externas, declara-se do mesmo modo que para subrotinas.

Declaração de Variáveis Internas

```

PLTI          DECLARE
               A  BYTE;
               B  ADDRESS;
               C  ADDRESS;
               D(9) BYTE;
               1  E,
               2  F  BYTE  INITIAL ('1'),
               2  G(5) ADDRESS INITIAL (6*0),
               1  H  BASED B,
               2  I  ADDRESS
               2  J  BYTE;

```

```

PL/I          DECLARE
               A  CHAR(1),
               B  POINTER,
               C  BIN FIXED (15),
               D (0:9)CHAR(1),

```

```

1 E,
  2 F CHAR(1) INITIAL ('1'),
  2 G(0:5) BIN FIXED(15) INITIAL ( (6)0),
1 H BASED (B),
  2 I BIN FIXED (15),
  2 J CHAR (1);

```

### Declaração de Rótulos

PLTI            DECLARE ROT LABEL;

PL/I            Não deve haver declaração.

### Atribuição de Valor Inicial

PLTI            DECLARE

```

A    BYTE INITIAL ('X'),
B    ADDRESS INITIAL (0),
C    ADDRESS INITIAL (0),
D (3)BYTE INITIAL ( 4 * '0'),
E (2)ADDRESS INITIAL (0,1,2),
F (15)BYTE INITIAL ('EDITOR DE TEXTOS');
```

PL/I            DECLARE

```

A    CHAR(1)    INITIAL('X'),
B    BIN FIXED(15) INITIAL (0),
C    POINTER    INITIAL (NULL),
D(0:3) CHAR(1) INITIAL((4)'0'),
E(0:2) BIN FIXED(15) INITIAL (0,1,2),
F(0:15) CHAR(1) INITIAL ('E','D',...,'S'),
ou
F    CHAR(16) INITIAL ('EDITOR DE TEXTOS');
```

NOTA: Deve-se observar que as variáveis no PLTI tem alocação es  
tática enquanto no PL/I tem alocação dinâmica.

Constante Hexadecimal

```
PLTI      DECLARE
          LINEFEED  BYTE  INITIAL (#0A);
```

```
PL/I     DECLARE
          LINEFEED CHAR(1)  DEFINED  LF,
          LF BIT(8)  INITIAL ('00001010'B),
```

Declaração de Subrotina Externa

```
PLTI     SUB:  PROCEDURE (ARG1,ARG2,ARG3) PLTI
          RESULT (ARG1);
          END;
```

```
PL/I     DECLARE SUB  ENTRY  (CHAR(1), CHAR(1) ,
          BIN FIXED (15));
```

Declaração de Função Externa

```
PLTI     FUNC: PROCEDURE (ARG1,ARG2) ADDRESS PLTI
          RESULT (ARG1);
          END;
```

```
PL/I     DECLARE  FUNC  ENTRY  (CHAR(1),CHAR(1))
          RETURNS (BIN FIXED(15));
```

Declaração de arquivo

```
PLTI     DECLARE ARQ FILE 'EXTARQ' ON 'DISCO'
          RECORD 80;  (não é válido em subrotina)

          DECLARE ARQ FILE;
```

```
PL/I     DECLARE ARQ FILE;  (para arquivos sequenciais)

          DECLARE ARQ1 FILE RECORD KEYED
          ENVIRONMENT (F REGIONAL (1));
          (para arquivos diretos)
```

Leitura de Arquivos

```

PLTI          READ  ARQ1  INTO AREA1 EOF ROT1
                                   ERROR ROT2;

              SEEK ARQ2  AT  CHAVE;

              READ ARQ2  INTO AREA2 EOF ROT3
                                   ERROR ROT4;

PL/I          ON ENDFILE (ARQ1) GO TO ROT1;
              ON TRANSMIT (ARQ1) GO TO ROT2;
              .
              .
              .
              READ FILE (ARQ1) INTO (AREA1);

              ON KEY (ARQ2) GO TO ROT3;
              ON TRANSMIT (ARQ2) GO TO ROT5;
              .
              .
              .
              READ FILE (ARQ2) INTO (AREA2) REY (CHAVE);

```

Gravação de Arquivos

```

PLTI          WRITE ARQ1 FROM AREA1  ERROR ROT1;

              SEEK ARQ2  AT  CHAVE;

              WRITE ARQ2 FROM AREA2  ERROR ROT2;

PL/I          ON TRANSMIT (ARQ1) GO TO ROT1;
              .
              .
              .
              WRITE FILE (ARQ1) FROM (AREA1);

```

```
ON KEY (ARQ2) GO TO ROT2;
WRITE FILE (ARQ2) FROM (AREA2) KEY FROM (CHAVE);
```

### Atualização de Arquivos

```
PLTI      REWRITE ARQ1 FROM AREA1 ERROR ROT1;
          SEEK ARQ2 AT CHAVE;
          REWRITE ARQ2 FROM AREA2 ERROR ROT2;
```

```
PL/I      ON KEY (ARQ1) GO TO ROT1;
          .
          .
          .
          REWRITE FILE (ARQ1) FROM (AREA1);

          ON KEY (ARQ2) GO TO ROT2;
          .
          .
          .
          REWRITE FILE (ARQ2) FROM (AREA2) KEY (CHAVE);
```

### Gravação de Marca de Fim

```
PLTI      ENDFILE ARQ1;

PL/I      CLOSE FILE (ARQ1); (opcional)
```

### Atribuição e Movimentação de Dados

```
PLTI      DECLARE
          (A1, A2) BYTE,
          B(10) ADDRESS,
          1 C,
          2 D(10) BYTE,
          2 E(10) ADDRESS,
```

```

1 F,
  2 G(10) BYTE,
  2 H(10) ADDRESS;

```

```

A1 = A2;
B(5) = E(9);
MOVE B(0) TO E(0) FOR 22;
MOVE C TO F FOR 33;

```

PL/I

```

DECLARE
  (A1,A2) CHAR(1),
  B(0:10) BIN FIXED(15),
  1 C,
  2 D (0:10) CHAR(1),
  2 E (0:10) BIN FIXED(15),

  1 F,
  2 G (0:10) CHAR(1),
  2 H (0:10) BIN FIXED(15);

A1 = A2;
B(5) = E(9);
E(*) = B(*);
F = C;

```

Operadores

PLTI	PL/I
<>	← =
NOT	┌
AND	ε
OR	
ω	ADDR

Comentários

PLTI           É o texto do registro de entrada (cartão) que sucede o símbolo %. Se um texto de comentário se estender por vários cartões, em cada cartão ele deve ser precedido pelo símbolo %.

PL/I           É o texto compreendido entre os símbolos /\* e \*/, podendo se estender por um ou mais registros de entrada (cartões).

Depuração e Estatística

PLTI           TRACE (A, ROT, PROC);  
                   SET 5;  
                   RESET 5;

PLTI           \*PROCESS COUNT, FLOW;  
                   PUT DATA;

Pré-processamento ou Processamento de Macros

PLTI           DECLARE  
                   DP LITERALLY 'DESVIOPADRAO';  
                   Ao encontrar o símbolo DP, o pré-processador trans  
                   forma-o na cadeia DESVIOPADRÃO.

PL/I           %DCL DP CHAR;  
                   %DP = 'DESVIOPADRAO';  
                   Ao encontrar o símbolo DP, o pré-processador trans  
                   forma-o na cadeia DESVIOPADRAO.



## 7.3 - METODOLOGIA DE CONVERSÃO

Uma vez que optou-se por desenvolver o software inteiramente em PL/I, optou-se também por realizar uma conversão para PLTI em modo inteiramente automático. Retirando-se o trabalho humano da atividade, não só elimina-se o erro de conversão como também permite que se trabalhe com um único texto fonte (em PL/I), conversível a qualquer momento em texto fonte PLTI pelo próprio computador.

A metodologia de conversão baseou-se unicamente na pesquisa de determinadas cadeias de caracteres com substituição por outra de igual tamanho. Sua escolha baseou-se, principalmente, na facilidade com que ela pode ser feita, inclusive dispensando qualquer tipo de análise sintática do texto. O equipamento utilizado dispunha inclusive de um programa utilitário que realizava a conversão.

A relação seguinte indica as cadeias pesquisadas e as cadeias que as substituem na transformação do programa PL/I para PLTI. Nela o símbolo b indica o espaço em branco.

<u>Cadeia Pesquisada</u>	<u>Cadeia Substituida</u>
*bPROCESS	%bPROCESS
/*	%b
*/	%b
CHAR(1)	BYTEbbb
BINbFIXED(15)	ADDRESSbbbbbb
POINTER	ADDRESS
bEXTbbb	bGLOBAL
EXTERNAL	bbbbbbbbb
UNALIGNED	bbbbbbbbb
STATIC	bbbbbb
ADDR(	bbb(Ⓐ)
Ø:	bb
⌈ =	<>
b Ⓔ b	AND

b   b	ORb
b - b	NOT
BASED (PT)	BASEDbPTb
BASED (PT1)	BASEDbPT1b
.	.
.	.
.	.

todas as variáveis "based" utilizadas.

#### 7.4 - METODOLOGIA DE PROGRAMAÇÃO

Nesse item são abordados apenas os aspectos da programação na linguagem PL/I de programas possíveis de conversão para PLTI, pela simples aplicação dos métodos descritos em 7.3. É portanto, mera decorrência das diferenças sintáticas e semânticas entre as duas linguagens e dos citados métodos.

Cinco princípios básicos nortearam a programação:

1. As características do PLTI sem correspondente no PL/I não foram utilizadas.
2. As características do PL/I sem correspondente no PLTI não foram utilizadas.
3. O texto sujeito a conversão foi sempre escrito de modo rigorosamente igual a cadeia de pesquisa.
4. Os textos onde não era permitida a conversão pela pesquisa e troca de cadeias foi escrito duas vezes: uma como comando PL/I e como comentário PLTI e outra como comando PLTI e como comentário PL/I.
5. Os textos não possíveis de conversão automática usados repetidas vezes foram localizados em subrotinas.

A apresentação de um módulo antes e após a conversão exemplifica os princípios adotados.

## PROGRAMA PL/I

```

* PROCESS A,X,AG,NEST,GS,MAR(2,72,1),LC(59); 0000010
/* */ EDEXCE : PROCEDURE; /* EXCLUI LINHA BASE TELA (COMANDO =E.) */ 0000020
/* EDEXCE ***** */ 0000030
/* 0000040
/* DESCRICAO : 0000050
/* A SUBROTINA EXCLUI A LINHA DO MODULO SITUADA NA BASE DA TELA 0000060
/* RULANTE, ISTO E', EXECUTA O COMANDO =E. 0000070
/* 0000080
/* DATA : 25 NOV 1977 0000090
/* 0000100
/* PROGRAMADJR : OSWALDO SERRA ALVES PEREIRA 0000110
/* 0000120
/* PARAMETROS : NAO_HA 0000130
/* 0000140
/* VARIAVEIS EXTERNAS : PLSUP,ULINR,PCOL,ENDT(*),ELOET(*),ELODT(*), 0000150
/* TL(*) 0000160
/* 0000170
/* SUBROTINAS EXTERNAS : EDMEMO,EDXBC,EDMOV,EDIJ,EDMOD,EDDEVE 0000180
/* 0000190
/* ALGORITMO : 0000200
/* 1- ALUCA 3 AREAS DE 80 BYTES PARA CCNTER O REGISTRO EXCLUIDO, 0000210
/* SEU ANTECESSOR E SUCESSOR RESPECTIVAMENTE 0000220
/* 2- VERIFICA SE A LINHA DA BASE DA TELA E' UMA LINHA DO MODULO 0000230
/* 3- MOVE LINHA DA BASE DA TELA PARA ESTRUTURA "REG1" 0000240
/* 4- SE A PENULTIMA LINHA DA TELA CONTIVER UMA LINHA DO MODULO, 0000250
/* ELA E' MUVIDA PARA ESTRUTURA "REG2" 0000260
/* 5- EXCLUI O REGISTRO DO MODULO E O DEVOLVE AO ESPACO DISPO- 0000270
/* NIVEL 0000280
/* 6- SE HOUVER UM REGISTRO A DIREITA DO REGISTRO EXCLUIDO, 0000290
/* ESTE E' COLCOADO NA BASE DA TELA 0000300
/* 7- SE NAO HOUVER UM REGISTRO A ESQUERDA DO REGISTRO EXCLUIDO, 0000310
/* ROLA-SE A TELA 1 LINHA PARA BAIXO 0000320
/* 8- LIBERA AS 3 AREAS DE 80 BYTES ALOCADAS 0000330
/* 0000340
/* ***** */ 0000350
/* ===== VARIAVEIS EXTERNAS ===== */ 0000360
/* 0000370
/* 0000380
/* 0000390
/* */ DECLARE 0000400
/* (PLSUP,ULINR,PCOL) BIN FIXED(15) EXTERNAL, 0000410
/* ENDT(0:19) BIN FIXED(15) EXTERNAL, 0000420
/* ELOET(0:19) BIN FIXED(15) EXTERNAL, 0000430
/* ELODT(0:19) BIN FIXED(15) EXTERNAL, 0000440
/*
/* TL(0:19) CHAR(1) EXTERNAL; 0000450
/* 0000460
/* 0000470
/* EDEXCE : PROCEDURE; 0000480
/* 0000490
/* ===== SUBROTINAS EXTERNAS ===== */ 0000500
/* =====DECLARACAO PL/I===== */ 0000510
/* */ DECLARE EDDEVE ENTRY(POINTER), 0000520
/* */ EDMOD ENTRY(POINTER,POINTER,POINTER), 0000530
/* */ EDMEMO ENTRY(CHAR(1),POINTER,CHAR(1),BIN FIXED(15)), 0000540
/* */ EDMOV ENTRY(POINTER,POINTER;BIN FIXED(15)), 0000550
/* */ EDIJ ENTRY(BIN FIXED(15),BIN FIXED(15)) RETURNS(BIN FIXED(15)), 0000560
/* */ EDXBC ENTRY(CHAR(1),BIN FIXED(15),CHAR(1)); 0000570
/* =====DECLARACAO PLTI===== */ 0000580
/* 0000590
/* EDMEMO : PROCEDURE(RET,PTA,OP,TAM) PLTI RESULT(RET,PTA); END; 0000600
/* EDMOV : PROCEDURE(PTO,PTF,TAM) PLTI; END; 0000610
/* EDXBC : PROCEDURE(OP,NLIN,EXBT) PLTI; END; 0000620
/* EDIJ : PROCEDURE(LINHA,COLUNA) ADDRESS PLTI; END; 0000630
/* EDMEMO : PROCEDURE(PT1,PT2,PT3) PLTI; END; 0000640
/* EDDEVE : PROCEDURE(PT) PLTI; END; 0000650
/* 0000660
/* ===== VARIAVEIS INTERNAS ===== */ 0000670
/* DECLARE (PT1,PT2,PT3,PTF,PTD) POINTER, 0000680
/* 1 REG1 BASED(PT1), /*CONTERA O REG */ 0000690
/* 2 R1ENDEREÇO BIN FIXED(15), /*EXCLUIDO */ 0000700
/* 2 R1, 0000710
/* 3 R1ELOESQ BIN FIXED(15), 0000720
/* 3 R1ELQDIR BIN FIXED(15), 0000730
/* 3 R1TIPO CHAR(1), 0000740
/* 3 R1DADOS, 0000750
/* 4 R1D(0:79) CHAR(1), 0000760
/* 1 REG2 BASED(PT2), /*CONTERA O REG A */ 0000770
/* 2 R2ENDEREÇO BIN FIXED(15), /*ESQ DO REGISTRO */ 0000780
/* 2 R2, /*EXCLUIDO */ 0000790
/* 3 R2ELOESQ BIN FIXED(15), 0000800
/* 3 R2ELQDIR BIN FIXED(15), 0000810

```

Cont. (1)

3 R2TIPO	CHAR(1),	00000820
3 R2DADOS,		00000830
4 R2D(0:79)	CHAR(1),	00000840
1 REG3	BASED(PT3),	/*CONTERA O REG A */ 00000850
2 R3ENDERECO	BIN FIXED(15),	/*DIREITA DO REG */ 00000860
2 R3,		/*EXCLUIDO */ 00000870
3 R3ELOESQ	BIN FIXED(15),	00000880
3 R3ELODIR	BIN FIXED(15),	00000890
3 R3TIPO	CHAR(1),	00000900
3 R3DADOS,		00000910
4 P3D(0:79)	CHAR(1),	00000920
(RET1,RET2,RET3)	CHAR(1),	00000930
ULINR	BIN FIXED(15);	00000940
/*		00000950
DECLARE	RETORNO LABEL; */	00000960
/*	1- ALOCA 3 AREAS DE 88 BYTES PARA CONTER O REGISTRO EXCLUI-	00000970
/*	DO, SEU ANTECESSOR E SUCESSOR RESPECTIVAMENTE */	00000980
CALL	EDMEMO(RET1,PT1,'A',88);	00000990
CALL	EDMEMO(RET2,PT2,'A',88);	00001000
CALL	EDMEMO(RET3,PT3,'A',88);	00001010
IF	RET1 = '1'   RET2 = '1'   RET3 = '1'	00001020
THEN	GO TO RETORNO;	00001030
/*	2- VERIFICA SE A LINHA DA BASE DA TELA E' UMA LINHA DO */	00001040
/*	MODULO */	00001050
IF	PLSUP > ULINR	00001060
THEN	GO TO RETORNO;	00001070
/*	3- MOVE LINHA DA BASE DA TELA PARA ESTRUTURA "REG1" */	00001080
R1ENDERECO	= ENDT(ULINR);	00001090
R1ELOESQ	= ELOET(ULINR);	00001100
R1ELODIR	= ELODT(ULINR);	00001110
R1TIPO	= '4';	00001120
PTF	= ADDR(TL(EDIJ(ULINR,PCOL)));	00001130
PTD	= ADDR(R1DADOS);	00001140
CALL	EDMOV(PTD,PTF,80);	00001150
/*	4- SE A PENULTIMA LINHA DA TELA CONTIVER UMA LINHA DO MODU-	00001160
/*	LO, ELA E' MOVIDA PARA ESTRUTURA "REG2" */	00001170
IF	PLSUP < ULINR	00001180
THEN	DO;	00001190
ULINR1	= ULINR - 1;	00001200
R2ENDERECO	= ENDT(ULINR1);	00001210
R2ELOESQ	= ELOET(ULINR1);	00001220
R2ELODIR	= ELODT(ULINR1);	00001230
R2TIPO	= '4';	00001240
PTF	= ADDR(TL(EDIJ(ULINR1,PCOL)));	00001250
PTD	= ADDR(R2DADOS);	00001260
CALL	EDMOV(PTD,PTF,80);	00001270
END;		00001280
ELSE	R2ENDERECO = 0; /*INDICA A EDEMODO QUE NAO EXISTE */	00001290
	/*REGISTRO A ESQUERDA */	00001300
/*	5- EXCLUI O REGISTRO DO MODULO E O DEVOLVE AO ESPACO */	00001310
/*	DISPONIVEL */	00001320
R3ENDERECO	= 0;	00001330
CALL	EDEMODO(PT1,PT2,PT3);	00001340
CALL	EDDEVE(PT1);	00001350
/*	6- SE HOUVER UM REGISTRO A DIREITA DO REGISTRO EXCLUIDO, */	00001360
/*	ESTE E' COLOCADO NA BASE DA TELA */	00001370
IF	R3ENDERECO > 0	00001380
THEN	DO;	00001390
ENDT(ULINR)	= R3ENDERECO;	00001400
ELOET(ULINR)	= R3ELOESQ;	00001410
ELODT(ULINR)	= R3ELODIR;	00001420
PTF	= ADDR(R3DADOS);	00001430
PTD	= ADDR(TL(EDIJ(ULINR,PCOL)));	00001440
CALL	EDMOV(PTD,PTF,80);	00001450
IF	PLSUP < ULINR	00001460
THEN	ELODT(ULINR - 1) = R3ENDERECO;	00001470
END;		00001480
/*	7- SE NAO HOUVER UM REGISTRO A ESQUERDA DO REGISTRO EXCLUI-	00001490
/*	DO, ROLA-SE A TELA 1 LINHA PARA BAIXO */	00001500
ELSE	DO;	00001510
CALL	EDX8C('B',1,'C');	00001520
ELODT(ULINR)	= 0;	00001530
END;		00001540
/*	8- LIBERA AS 3 AREAS DE 88 BYTES ALOCADAS */	00001550
RETORNO :		00001560
CALL	EDMEMO(RET3,PT3,'L',88);	00001570
CALL	EDMEMO(RET2,PT2,'L',88);	00001580
CALL	EDMEMO(RET1,PT1,'L',88);	00001590
END	EEXCE;	00001600

## PROGRAMA PLTI

```

% PROCESS A, X, AG, NEST, GS, MAR(2,72,1), LC(59); 00000010
% EDEXCE : PROCEDURE; % EXCLUI LINHA BASE TELA (COMANDO =E.) % 00000020
EDEXCE. ***** 00000030
% DESCRICAO : 00000050
% A SUBROTINA EXCLUI A LINHA DO MODULO SITUADA NA BASE DA TELA 00000060
% ROLANTE, ISTO E', EXECUTA O COMANDO =E. 00000070
% DATA : 25 NOV 1977 00000080
% PROGRAMADOR : OSWALDO SERRA ALVES PEREIRA 00000110
% PARAMETRUS : NAU HA 00000120
% VARIAVEIS EXTERNAS : PLSUP, ULINR, PCCL, ENDT(*), ELOET(*), ELODT(*), 00000150
% TL(*) 00000160
% SUBROTINAS EXTERNAS : EDMEMO, EDXBC, EDMOV, EDIJ, EDMOD, EDDEVE 00000180
% ALGORITMO : 00000200
% 1- ALOCA 3 AREAS DE 80 BYTES PARA CONTER O REGISTRO EXCLUIDO, 00000210
% SEU ANTECESSOR E SUCESSOR RESPECTIVAMENTE 00000220
% 2- VERIFICA SE A LINHA DA BASE DA TELA E' UMA LINHA DO MODULO 00000230
% 3- MOVE LINHA DA BASE DA TELA PARA ESTRUTURA "REG1" 00000240
% 4- SE A PENULTIMA LINHA DA TELA CONTIVER UMA LINHA DO MODULO, 00000250
% ELA E' MOVIDA PARA ESTRUTURA "REG2" 00000260
% 5- EXCLUI O REGISTRO DO MODULO E O DEVOLVE AO ESPACO DISPO- 00000270
% NIVEL 00000280
% 6- SE HOUVER UM REGISTRO A DIREITA DO REGISTRO EXCLUIDO, 00000290
% ESTE E' COLOCADO NA BASE DA TELA 00000300
% 7- SE NAO HOUVER UM REGISTRO A ESQUERDA DO REGISTRO EXCLUIDO, 00000310
% ROLA-SE A TELA 1 LINHA PARA BAIXO 00000320
% 8- LIBERA AS 3 AREAS DE 80 BYTES ALOCADAS 00000330
% ***** 00000350
% ===== VARIAVEIS EXTERNAS ===== 00000360
% [ 00000370
% % DECLARE 00000380
% (PLSUP, ULINR, PCCL) ADDRESS 00000410
% ENDT( 19) ADDRESS , 00000420
% ELOET( 19) ADDRESS , 00000430
% ELODT( 19) ADDRESS , 00000440
%
% TLI( 1919) BYTE ; 00000450
% ] 00000460
% EDEXCE : PROCEDURE; 00000470
% 00000480
% ===== SUBROTINAS EXTERNAS ===== 00000490
% ==DECLARACAO PL/I== 00000500
% % DECLARE EDDEVE ENTRY(ADDRESS), 00000520
% % EDMOD ENTRY(ADDRESS, ADDRESS, ADDRESS), 00000530
% % EDMEMO ENTRY(BYTE , ADDRESS, BYTE , ADDRESS ), 00000540
% % EDMOV ENTRY(ADDRESS, ADDRESS, ADDRESS ), 00000550
% % EDIJ ENTRY(ADDRESS , ADDRESS ) RETURNS(ADDRESS ), 00000560
% % EDXBC ENTRY(BYTE , ADDRESS , BYTE ); 00000570
% ==DECLARACAO PLTI== 00000580
% EDMEMO % PROCEDURE(RET, PTA, OP, TAM) PLTI RESULT(RET, PTA); END; 00000590
% EDMOV : PROCEDURE(PTD, PTF, TAM) PLTI; END; 00000600
% EDXBC : PROCEDURE(OP, NLIN, EXBT) PLTI; END; 00000620
% EDIJ : PROCEDURE(LINHA, COLLNA) ADDRESS PLTI; END; 00000630
% EDMOD : PROCEDURE(PT1, PT2, PT3) PLTI; END; 00000640
% EDDEVE : PROCEDURE(PT) PLTI; END; 00000650
%
% ===== VARIAVEIS INTERNAS ===== 00000660
% DECLARE (PT1, PT2, PT3, PTF, PTD) ADDRESS, 00000680
% 1 REG1 BASED PT1 , % CONTERA O REG % 00000690
% 2 R1ENDereco ADDRESS , % EXCLUIDO % 00000700
% 2 R1, 00000710
% 3 R1ELOESQ ADDRESS , 00000720
% 3 R1ELODIR ADDRESS , 00000730
% 3 R1TIPO BYTE , 00000740
% 3 R1DADOS, 00000750
% 4 R1D( 79) BYTE , 00000760
% 1 REG2 BASED PT2 , % CONTERA O REG A % 00000770
% 2 R2ENDereco ADDRESS , % ESQ DO REGISTRO % 00000780
% 2 R2, % EXCLUIDO % 00000790
% 3 R2ELOESQ ADDRESS , 00000800
% 3 R2ELODIR ADDRESS , 00000810

```

Cont. (1)

```

3 R2TIPO, BYTE, 00000820
3 R2DADOS, 00000830
4 R2D( 79) BYTE, 00000840
1 REG3 BASED PT3, % CONTERA O REG A % 00000850
2 R3ENDERECO ADDRESS, % DIREITA DO REG % 00000860
2 R3, % EXCLUIDO % 00000870
3 R3ELOESQ ADDRESS, 00000880
3 R3ELODIR ADDRESS, 00000890
3 R3TIPO BYTE, 00000900
3 R3DADOS, 00000910
4 R3D( 79) BYTE, 00000920
(R2T1,R2T2,R2T3) BYTE, 00000930
ULINR1 ADDRESS, 00000940
% 00000950
DECLARE RETORNO LABEL; % 00000960
% 1- ALUCA 3 AREAS DE 88 BYTES PARA CONTER O REGISTRO EXCLUI-? 00000970
% DO, SEU ANTECESSOR E SUCESSOR RESPECTIVAMENTE % 00000980
CALL EDMEMO(R2T1,PT1,'A',88); 00000990
CALL EDMEMO(R2T2,PT2,'A',88); 00001000
CALL EDMEMO(R2T3,PT3,'A',88); 00001010

IF RET1 = '1' OR RET2 = '1' OR RET3 = '1' 00001020
THEN GO TO RETORNO; 00001030
% 2- VERIFICA SE A LINHA DA BASE DA TELA E' UMA LINHA DO % 00001040
% MODULO % 00001050
IF PLSUP > ULINR 00001060
THEN GO TO RETORNO; 00001070
% 3- MOVE LINHA DA BASE DA TELA PARA ESTRUTURA "REG1" % 00001080
R1ENDERECO = ENDT(ULINR); 00001090
R1ELOESQ = ELOET(ULINR); 00001100
R1ELODIR = ELODT(ULINR); 00001110
R1TIPO = '4'; 00001120
PTF = (ATL(EDIJ(ULINR,PCOL))); 00001130
PTD = (AR2DADOS); 00001140
CALL EDMOV(PTD,PTF,80); 00001150
% 4- SE A PENULTIMA LINHA DA TELA CONTIVER UMA LINHA DO MODU-? 00001160
% LO, ELA E' MOVIDA PARA ESTRUTURA "REG2" % 00001170
IF PLSUP < ULINR 00001180
THEN DO; 00001190
ULINR1 = ULINR - 1; 00001200
R2ENDERECO = ENDT(ULINR1); 00001210
R2ELOESQ = ELOET(ULINR1); 00001220
R2ELODIR = ELODT(ULINR1); 00001230
R2TIPO = '4'; 00001240
PTF = (ATL(EDIJ(ULINR1,PCOL))); 00001250
PTD = (AR2DADOS); 00001260
CALL EDMOV(PTD,PTF,80); 00001270
END; 00001280
ELSE R2ENDERECO = 0; % INDICA A EDEMODO QUE NAO EXISTE % 00001290
% REGISTRO A ESQUERDA % 00001300
% 5- EXCLUI O REGISTRO DO MODULO E O DEVOLVE AO ESPACO % 00001310
% DISPONIVEL % 00001320
R3ENDERECO = 0; 00001330
CALL EDMEMO(PT1,PT2,PT3); 00001340
CALL EDDEVE(PT1); 00001350
% 6- SE HOUVER UM REGISTRO A DIREITA DO REGISTRO EXCLUIDO, % 00001360
% ESTE E' COLOCADO NA BASE DA TELA % 00001370
IF R3ENDERECO > 0, 00001380
THEN DO; 00001390
ENDT(ULINR) = R3ENDERECO; 00001400
ELOET(ULINR) = R3ELOESQ; 00001410
ELODT(ULINR) = R3ELODIR; 00001420
PTF = (AR3DADOS); 00001430
PTD = (ATL(EDIJ(ULINR,PCOL))); 00001440
CALL EDMOV(PTD,PTF,80); 00001450
IF PLSUP < ULINR 00001460
THEN ELODT(ULINR - 1) = R3ENDERECO; 00001470
END; 00001480
% 7- SE NAO HOUVER UM REGISTRO A ESQUERDA DO REGISTRO EXCLUI-? 00001490
% DO, ROLA-SE A TELA 1 LINHA PARA BAIXO % 00001500
ELSE DO; 00001510
CALL EDXBC('B',1,'C'); 00001520
ELODT(ULINR) = 0; 00001530
END; 00001540
% 8- LIBERA AS 3 AREAS DE 88 BYTES ALOCADAS % 00001550
RETORNO : 00001560
CALL EDMEMO(R2T2,PT2,'L',88); 00001570
CALL EDMEMO(R2T3,PT3,'L',88); 00001580

CALL EDMEMO(R2T1,PT1,'L',88); 00001590
END EDEXCE; 00001600

```

NOTA: O símbolo Å significa Å

## 8. CONCLUSÕES

É possível, para uma máquina de pequeno porte, provida de uma linguagem de alto nível que apresente similaridade com alguma linguagem de um computador de maior porte, estabelecer um conjunto de regras que permitem o desenvolvimento de software para a máquina de menor porte no computador de maior porte.

Tal fato se reveste de importância pela possibilidade de:

- explorar-se as facilidades de depuração e documentação que um compilador projetado para máquinas de maior porte oferece.
- utilizar-se das facilidades que o sistema operacional e os utilitários do computador maior oferecem.
- valer-se da velocidade da máquina maior e de seus dispositivos de entrada-saída para compilar e testar os programas.
- usar-se uma memória mais extensa para criar armadilhas de teste temporárias.
- produzir-se um software com acentuado grau de portabilidade entre as duas máquinas.

Usando-se tal recurso consegue-se migrar o software para o mini em um estágio razoavelmente avançado de depuração. Em contrapartida pode-se observar mais tardiamente efeitos indesejáveis associados ao tempo de execução e às necessidades de memória. Alguns programas poderão exigir que se aproveite, em pontos críticos, potencialidade não exploradas da linguagem de alto nível do minicomputador ou até de seu "assembler".



As possibilidades de desenvolvimento de "software" em locais diversos e distantes, em máquinas de uso difundido e sem a utilização de "cross-software", inclusive em paralelo com a própria fabricação dos protótipos e do "software" básico, permitem-me concluir que a idéia deve ser explorada com mais profundidade e com mais direções, principalmente no que tange aos problemas, da maior similaridade, da metodologia de conversão e de novas linguagens de minicomputadores (ou micro) similares ao PL/I, ALGOL, COBOL, FORTRAN e APL.

A similaridade do PLTI ao PL/I permitiu o desenvolvimento do Editor de Textos (EDTEX) usando-se as vantagens de uma máquina PL/I e chegando-se a um resultado positivo no computo das vantagens e desvantagens.

B I B L I O G R A F I A

- 1 - MARTIN, James  
"Design of Man-Computer Dialogue"  
Prentice-Hall, Inc.
  
- 2 - DATAPOINT CORPORATION  
"Cassete and DOS Editor - Program Users Guide"  
Datapoint Corporation - 1973
  
- 3 - FRANÇA, Paulo Mario Bianchi e PAIVA, Paulo Fernando Molo  
"Sistema Preparador de Programas"  
NCE/UFRJ - 1975
  
- 4 - RODRIGUES, Guilherme Chagas  
"Sistema Operacional para Terminal Inteligente"  
NCE/UFRJ
  
- 5 - MELO, Paulo Cesar Moraes  
"Uma linguagem de Alto Nível para Microcomputadores-PLTI"  
NCE/UFRJ
  
- 6 - GRIES, David  
"Compiler Construction for Digital Computers"  
John Wiley & Sons, Inc. 1971
  
- 7 - BURROUGHS CORPORATION  
"B6700 - B7700 Command and Edit (Cande)  
Language. Information Manual"  
Burroughs Corporation, Detroit, Michigan 48232