

EDITOR DE TEXTOS

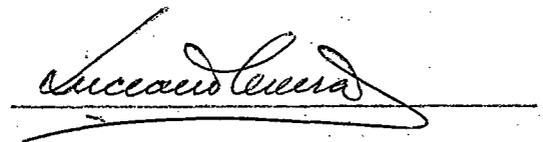
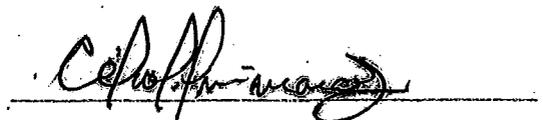
Enio Antonio Garbin

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

Aprovada por:



Presidente



RIO DE JANEIRO

ESTADO DA GUANABARA - BRASIL

OUTUBRO DE 1973

À Thereza, por seu amor e dedicação

Na elaboração deste trabalho, muitos foram os que me auxiliaram; a eles quero expressar meus agradecimentos. Agradeço, especialmente, ao Prof. Dr. Ivo Wolf, Reitor da Universidade Federal do Rio Grande do Sul; ao Prof. Dr. Eduardo Zaccaro Faraco, ex-reitor daquela Universidade; ao Prof. Eng. Laureano I. Chaffe, chefe do Departamento de Geodésia da mesma Universidade; ao Dr. Sergio Telles Ribeiro, Superintendente Geral do Departamento Geral de Computação e Sistemas da Companhia Telefônica Brasileira; ao meu orientador e professores da COPPE; ao meu sogro, José Diogo Brochado da Rocha, grande incentivador e incansável em seu auxílio; aos meus pais e aos meus filhos.

RESUMO

A Edição de Textos ou Composição de Textos é uma técnica para produzir textos impressos que tem crescido de maneira contínua nestes últimos anos.

Este crescimento é devido, principalmente, ao custo crescente da mão de obra, a facilidade de uso de computadores em tempo-partilhado e ao preço decrescente do equipamento.

Este trabalho levanta e analisa os diversos editores existentes, estabelece quais as premissas que devem ser obedecidas no projeto de tais sistemas e apresenta e implementa um novo sistema.

ABSTRACT

Text editing or text composition is a technique to produce printed texts that has been growing steadily those last years. This growing is largely due to the increasing cost of man-power, the facility of using computers through time-sharing networks and the decreasing price of hardware. This work surveys and analyzes the various text editors existing, establishes the main-points in the project of such systems and defines and implement one that follows those points, named independency, flexibility, universality, operability and safety.

INDICE

INTRODUÇÃO	1
SISTEMA PROPOSTO	8
ANÁLISE DOS SISTEMAS EXISTENTES	14
DESCRIÇÃO DO SISTEMA	31
CONCLUSÃO	94
ANEXO I - BIBLIOGRAFIA	99
ANEXO II - ALGORITMOS	102
ANEXO III - MANUAL DO USUÁRIO	109
ANEXO IV - GLOSSÁRIO	144
ANEXO V - EXEMPLOS DE APLICAÇÃO	149

CAPÍTULO 1

INTRODUÇÃO

1.1. DEFINIÇÕES

Para que o leitor do presente trabalho possa ter entendimento completo de sua exposição, três termos devem ser, previamente, definidos por não serem usuais em linguagem corrente, mas relacionados com a arte de impressão.

Estes termos são:

- a) EDIÇÃO: Define-se como edição a obtenção em um meio reconhecível pelo homem, de um texto qualquer obedecendo formato determinado.

Podemos citar como exemplo uma página de um livro; o homem pode lê-la e ela obedece formato determinado por seu conteúdo e finalidade.

- b) FORMATAÇÃO: Define-se como formatação (ou composição) o arranjo de caracteres de um texto de acordo com regras previamente estabelecidas.

O mesmo exemplo anterior serve para ilustrar esta definição; para a obtenção do texto editado, os caracteres, representados por tipos de impressão, foram arranjados em um molde de acordo com uma regra antes estabelecida.

- c) TEXTO FONTE: Define-se como texto fonte aquele que representa a idéia a ser editada e que ainda não foi formatado para esse fim.

É exemplo válido para esta definição o rascunho de uma obra qualquer não obedecendo formato definido; a forma para edição será feita posteriormente pelo editor.

1.2. PEQUENO HISTÓRICO DA EDIÇÃO

A arte da edição é velha, de mais de 1100 anos; a primeira obra editada conhecida é um livro chinês datado do ano de 868.

Este livro foi impresso a partir de moldes gravados em argila ou madeira, cada um correspondendo a uma de suas páginas.

O formato desta era, portanto, fixo e qualquer modificação em uma delas acarretaria modificações nas demais.

A primeira alteração introduzida no sistema foi a de fazerem-se moldes de tamanho tal que em cada um coubesse somente um ideograma (caráter chinês).

Dessa maneira foi possível uma maior flexibilidade na formatação (composição).

Ainda, permanecia, se bem que facilitada, a necessidade de alterar toda a composição por modificação no texto.

Em 1456, Guttemberg editava uma Bíblia composta com tipos isolados e apresentava as seguintes inovações:

- a) as linhas eram justificadas pelas margens, e
- b) os caracteres ocupavam espaço proporcional à sua

largura produzindo um texto harmônico; até aquela época os tipos tinham, todos, a mesma largura estando o caráter centrado.

Permanecia, no entanto, o problema da modificação do texto já composto em virtude de modificações no texto fonte.

Durante 400 anos, isto é, até 1885 o sistema de composição de textos não sofreu qualquer evolução.

Nesse ano, Merghentaler mecanizou o processo de composição, pela introdução do "linotipo" que gerava os tipos necessários para a composição, eliminando, com isso, a necessidade de seleção manual dos tipos.

Houve a aceleração do processo de composição, mas não foi suprimido o problema de recomposição decorrente de modificação do texto fonte.

Este processo é ainda utilizado; sofreu evolução no sentido da automação, sendo hoje controlado por fita de papel perfurada na qual estão contidos não só os caracteres bem como os controles necessários.

Em 1948 surgiu novo processo de composição, denominado de "foto-composição", que consta da sensibilização de película fotográfica.

Este processo acelerou, ainda mais, a composição de matrizes para a edição, podendo atingir até 10.000 caracteres por minuto.

Continuavam, porém, pendentes de recomposição as modificações que fossem introduzidas no texto fonte.

1.3. PROBLEMA DA EDIÇÃO CONVENCIONAL

Pelo pequeno e sucinto histórico percebe-se que o processo de composição de textos sofreu grande evolução, nos métodos de composição, mas o sistema é, ainda, o mesmo.

A composição é, ainda, feita pela colocação lado a lado dos caracteres a serem editados; se o texto sofre qualquer modificação este arranjo deve ser, todo, modificado, para atender às novas exigências.

Isto é, nova composição deve ser efetuada a cada modificação que é feita no texto fonte.

O ciclo de edição de uma obra tem o seguinte desdobramento:

- a) o autor envia os originais ao editor;
- b) estes os compõe, edita um rascunho e devolve ao autor para verificação e correção;
- c) o autor efetua as correções e modificações que crê necessárias e devolve ao editor.

Os passos b e c são repetidos até que o autor esteja satisfeito e a obra possa ser, finalmente, editada.

Toda a vez que a obra retorna ao editor este é obrigado a recompor toda ou parte dela.

Este processo, além de ter custo elevado, é demorado.

Se a obra necessitar atualização em edições posteriores, todo o processo se repete.

Qual a solução para este problema ?

"A solução proposta é manter o texto fonte arquivado em dispositivo de um sistema de processamento de informações de modo que ele possa ser modificado e atualizado veloz e eficientemente. Além disso, o sistema deve permitir sua composição total ou parcial e prever meios de possibilitar a edição, quer pela utilização de dispositivos do sistema, quer por utilização de dispositivos específicos de edição.

Os programas de sistemas de processamento de informações que permitem a execução dos objetivos acima descritos são denominados de "EDITORES DE TEXTO".

1.4: PROGRAMAS EDITORES DE TEXTO

Os programas denominados "editores de texto" surgiram em grandes Universidades que tinham excessivo número de programas usuais submetidos a seus computadores.

Se esses programas fossem submetidos uma só vez ao computador não haveria problemas.

Estes decorrem do fato de os programas serem submetidos várias vezes até estarem depurados e em condições de execução.

Isso obrigava a leitura dos "decks" de cartões um sem número de vezes, estrangulando ainda mais pontos já deficientes nos computadores: entrada e saída.

Para minorar essa dificuldade foram criados programas especiais que permitiam incluir, retirar ou modi-

ficar cartões de programa sem ter que resubmeter todo o "deck" de cartões; o "deck" original era gravado em unidade de arquivamento de acesso direto.

Esses programas foram denominados "editores de programa" e hoje são, quase sem exceção, parte dos programas utilitários dos sistemas de processamento de informações.

A filosofia que norteou o projeto desses editores é a de que o texto era composto de linhas e todas as operações eram feitas sobre elas.

Este conceito sofreu evolução no sentido de considerar um texto não composto de linhas mas sim de caracteres.

Este novo conceito, aliado à inclusão no texto fonte de caracteres de controle de edição veio a permitir o surgimento de programas que foram denominados de "editores de texto livre".

O objeto do presente trabalho é o projeto de um sistema editor de texto livre denominado EDITEXT (Editor de TEXTo).

Os editores de texto, tanto de um tipo, quanto de outro, são a seguir enumerados:

1.4.1. EDITORES DE PROGRAMA

CONVERSATIONAL CONTEXT-DIRECTED EDITOR (1)

MYLBUR (2)

QUICK-EDITOR (3)

TEXT EDITOR AND CORRECTOR (4)

EDIT-10 (5)

TVEDIT (6)

INTERACTIVE PROGRAMMING SUPPORT SYSTEM (6)

EMILY (6)

1.4.2. EDITORES DE TEXTO LIVRE

MAGNETIC-TAPE SELECTRIC TYPEWRITER (6,7)

ASTROTYPE (6,7)

SYSTEM/360-ADMINISTRATIVE TERMINAL SYSTEM (6,7)

VIPCOM (6,7)

HYPertext EDITING SYSTEM (6,7)

FILE RETRIEVAL AND EDITING SYSTEM (8,9)

SCRIPT (10)

EDIT/360 (11)

EDITPAK (12)

CAPÍTULO 2

SISTEMA PROPOSTO

O estudo dos sistemas atualmente existentes leva à conclusão de que um sistema eficiente de edição de textos livres deve ser projetado tendo em vista as seguintes premissas:

- a) Independência do Usuário
- b) Universalidade
- c) Flexibilidade
- d) Simplicidade de Operação
- e) Segurança

O sistema projetado é totalmente baseado nessas premissas e resulta do estudo pormenorizado e comparativo dos sistemas existentes.

O sistema proposto não é um sistema inédito; é o resultado da união de características existentes em outros sistemas, às quais novas foram introduzidas.

Algumas das características do sistema proposto são encontradas em outros sistemas, outras em alguns e, finalmente algumas delas em nenhum outro.

2.1. PREMISSAS

2.1.1. INDEPENDÊNCIA DO USUÁRIO

Entende-se como "independência do usuário" o seu completo desvinculamento da forma com que o sistema armazena o texto fonte.

Para o usuário, o texto fonte não está dividido em linhas, páginas ou registros.

Por ele o texto é tratado como uma longa cadeia de caracteres, uma "hipercadeia", dinâmica, que pode diminuir, aumentar ou ser modificada.

Essa cadeia pode ser tão pequena quanto um só caracter ou tão longa quanto uma enciclopédia.

A alocação de espaço de arquivamento é feita dinâmica e automaticamente pelo sistema.

2.1.2. UNIVERSALIDADE

A "universalidade" do sistema é definida como sendo sua capacidade de utilização de todos os recursos de entrada e saída do Sistema de Processamento de Dados para o qual foi projetado.

O sistema deve permitir entrada e saída de texto, fonte ou formatado, através de qualquer dispositivo, efetuando todas as conversões necessárias.

Cabe ao usuário informar quais os dispositivos que utilizará, bem como, em função destes, inse

rir no texto fonte os caracteres especiais que promoverão as conversões necessárias.

2.1.3. FLEXIBILIDADE

A "flexibilidade" é entendida como a capacidade de poder o sistema editar tanto textos livres, quanto programas de computador.

O usuário, editor de textos livres, pode ter necessidades em processamento de dados que não e são somente daquele tipo; pode, frequentemente, ter necessidade de resolver programas comuns de computação.

O sistema proposto provê um subconjunto de comandos para a edição de programas.

Neste subconjunto é abandonada a filosofia de "hipercadeia" e adotada a de "linhas de programa" numa simulação de cartões, método usual de entrada de programas.

2.1.4. SIMPLICIDADE DE OPERAÇÃO

A "simplicidade de operação" deve ser premissa básica em todo o sistema que se destine a um usuário não especialista em processamento de dados.

A sofisticação do sistema exige sua operação por pessoal especializado, elevando, assim seu custo.

Dois são os aspectos a examinar no estudo da simplicidade de operação: o primeiro relacionado com o texto fonte e suas modificações; o segundo com a formatação daquele mesmo texto.

Os "comandos" encarregam-se de manusear o texto fonte e cabe aos "controles de edição" comandar a formatação.

COMANDOS: As operações que podem ser efetuadas sobre o texto fonte são:

- . dar entrada
- . substituir
- . mover
- . copiar
- . apagar
- . formatar
- . editar

A cada uma destas funções deve corresponder um e só um comando; que deve ter capacidade ilimitada de operação.

CONTROLES DE EDIÇÃO: O formato que o texto terá ao final é dado pelos "controles de edição", inseridos no texto fonte e da mesma maneira que nos comandos, devem ser únicos e potentes. Eles indicam ao módulo de formatação ocorrências do tipo: nova página, saltar linhas, novo parágrafo, centrar ou justificar cadeias, criar tabelas, etc..

2.1.5. SEGURANÇA

Em qualquer sistema, a entrada de dados é a etapa mais lenta, que maior carga de trabalho traz ao usuário e de mais difícil recuperação.

As demais funções têm mais fácil recuperação e são de rápida execução.

A segurança do "arquivo de texto fonte" é vital ao sistema e foi objeto de cuidado especial.

Para evitar sua perda foi criado um sistema de "back-up" automático e permanente.

O texto fonte reside, simultaneamente, em dois arquivos:

- . o arquivo fonte propriamente dito e
- . o arquivo de segurança do fonte

Para que a função de segurança seja efetiva estes dois arquivos devem residir em dispositivos físicos diferentes.

Ambos arquivos são atualizados simultaneamente e pode ser recuperado um a partir do outro.

2.2. SISTEMA EDITEXT

O sistema EDITEXT obedece, em seu projeto, todas as premissas estabelecidas, sendo voltado ao conceito de "hipercadeia" quando em modo de edição de texto livre e de "linhas" quando em modo de edição de programa.

Nenhuma das premissas recebeu maior ênfase que outra; todas foram consideradas como tendo igual importância.

Algumas características não presentes em outros sistemas apresenta:

- a) permite a edição de textos com acentos e cedilha;
- b) é o único editor de texto livre que permite a edição de programa;
- c) facilita a criação de tabela de referências, permitindo o selecionamento de palavras-chave e fazer uma tabela situando-as dentro do texto, e
- d) é o único editor que prevê "back-up" automático do arquivo fonte e respectiva rotina de recuperação.

Nos capítulos 3 e 4 onde serão, respectivamente, analisados os sistemas existentes e descrito o sistema proposto estas características serão evidenciadas e facilmente visualizadas.

CAPÍTULO 3

ANÁLISE DOS SISTEMAS EXISTENTES

No presente capítulo será feita uma análise dos sistemas editores de texto livre, ao mesmo tempo, também, que uma referência mais sucinta sobre os editores de programa.

A maior ou menor profundidade da análise que será efetuada é função da bibliografia obtida e do nível de informação que esta contém.

3.1. MÉTODO DE ANÁLISE ADOTADO

No capítulo anterior foram definidas premissas que, no nosso entender, devem ser obedecidas para o projeto de editores de texto livre para serem eficientes.

A análise dos sistemas existentes será feita observando-os sob o prisma daquelas.

O método comparativo que, também, poderia ser utilizado não é o mais indicado devido à grande variação existente entre os sistemas.

Preferiu-se adotar, então, uma aproximação mais prática, examinando-os sob a luz daquelas premissas.

3.2. EDITORES DE TEXTO LIVRE

Os editores de texto livre que serão analisados são:

- a) EDITPAK
- b) SCRIPT
- c) FILE RETRIEVAL AND EDITING SYSTEM (FRESS)
- d) EDIT/360

A análise destes sistemas é possível pois deles obteve-se bibliografia suficiente.

Sob o título de "outros", faremos uma pequena análise, por insuficiência de literatura, dos seguintes editores:

- a) MAGNETIC TAPE SELECTRIC TYPEWRITER
- b) ASTROTYPE
- c) SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM
- d) VIPCOM
- e) HYPERTEXT EDITING SYSTEM (HES)

3.2.1. EDITPAK

O sistema EDITPAK é um projeto da Scientific Time Sharing Corp., e foi adotada como linguagem de programação o APL/360.

INDEPENDÊNCIA DO USUÁRIO

O usuário do sistema está totalmente desvinculado do mesmo; a forma com que trata e armazena o texto lhe é completamente invisível e o

acesso é feito por meio de 'cadeias de reconhecimento'.

UNIVERSALIDADE

As entradas e saídas de texto fonte ou editado só podem ser feitas através de um terminal IBM-2741.

FLEXIBILIDADE

O sistema edita somente textos livres; a edição de programas é inviável.

OPERAÇÃO

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

O número de comandos é amplo mas insuficiente; há comandos executando as mesmas funções a níveis diferentes.

O número de controles de edição é restrito, permitindo a edição de textos bastante simples. Estes controles, além de insuficientes, são pouco potentes; a repetição de controle é necessária sempre que a função deva ser repetida.

SEGURANÇA

Não há procedimentos previstos para a recuperação de erros e arquivos.

3.2.2. SCRIPT

O sistema SCRIPT é um editor de texto livre, propriedade da Brent Ass. e para uso no ISC Advanced/10 System, um computador PDP-10.

INDEPENDÊNCIA DO USUÁRIO

O usuário é dependente do sistema no sentido em que o mesmo, sendo orientado para uma página de n linhas, exige do usuário que este traga a página para um "buffer" para poder manuseá-la.

O sistema é orientado para a linha e não para a cadeia de caracteres.

A única exceção a isto é um comando de substituição que permite substituir uma cadeia de caracteres contida numa linha desde que a alteração não ultrapasse os limites dessa linha.

UNIVERSALIDADE

O sistema foi projetado para utilizar somente terminais do sistema PDP-10; não pode utilizar outros recursos.

FLEXIBILIDADE

O sistema não tem facilidade para a edição de programas.

OPERAÇÃO

A operação do sistema é dificultada pela necessidade que há de estar a linha a ser alterada no 'buffer' existente; um mínimo de dois comandos é necessário: o primeiro traz a linha e o segundo a modifica.

Os comandos existentes são em número suficiente para todas as funções necessárias na atualização do texto fonte.

A gama de controles de edição é completa, permitindo edições, das mais simples, às mais complexas.

Cada controle de edição ocupa uma linha de entrada, antecedendo a(s) linha(s) de texto a que se refere.

Cada controle de edição efetua somente uma função; não há controles com funções múltiplas.

A silabação é possível; não é, todavia, feita integralmente pelo sistema; necessita de auxílio do usuário.

SEGURANÇA

Não há comandos de recuperação de arquivos; o usuário pode dividir seu texto em pequenos arquivos e, ao final, editá-los em conjunto.

3.2.3. FILE RETRIEVAL AND EDITING SYSTEM (FRESS)

O sistema FRESS foi projetado pela Text Systems

Inc. para ser operado através de terminais IBM 2260 ou IBM-2741, respectivamente, vídeo e máquina de escrever.

INDEPENDÊNCIA DO USUÁRIO

O sistema prevê a completa independência do usuário em relação ao método com que manipula o texto; para ele, o texto é composto de uma única cadeia de caracteres, um ('hipertexto'), não estando amarrado a linhas, páginas ou registros.

O acesso do usuário é feito por meio de "cadeias de reconhecimento".

UNIVERSALIDADE

O sistema é universal para o sistema de processamento para o qual foi projetado; pode utilizar-se de todos os dispositivos de entrada e saída.

Com o uso de caracteres especiais pode utilizar-se de dispositivos diferentes de entrada e saída (isto é, que não têm o mesmo conjunto de caracteres).

FLEXIBILIDADE

O sistema não tem facilidades para a edição de programas.

OPERAÇÃO

A operação do sistema é bastante simples se o

texto a editar não é muito complexo e não se deseja utilizar as características sofisticadas do sistema.

Os comandos são únicos para cada função e muito poderosos; sua gama é completa.

Os controles de edição, da mesma forma que os comandos, são únicos, completos e poderosos.

Uma característica que possui e que facilita sua operação é a capacidade de introduzir marcos ("labels") no texto fonte, permitindo rápidos deslocamentos ao longo deste.

Este sistema permite qualquer tipo de edição inclusive edições com mais de uma coluna (tipo jornal).

Outra característica é a possibilidade de criação de "functions" que armazenam diversos controles e que, quando chamadas, inserem no texto este conjunto de controles.

O sistema admite quatro tipos de justificação de linhas: ambas as margens, margem direita, margem esquerda e tipo "sino" (centrando as linhas).

O sistema não tem silabação, duplicação de cadeias ou criação de tabelas de referência.

SEGURANÇA

A segurança dos arquivos não é automática,

deve ser providenciada pelo usuário que pode criar arquivos de 'back-up' e após editá-los em conjunto.

Pode armazenar a última alteração feita e em caso de incorreta o próprio sistema a desfaz retornando ao estado anterior.

3.2.4. EDIT/360

O sistema EDIT/360 é um programa produto da International Business Machine e tem como finalidade produzir entrada para dispositivos de foto-composição ou linotipos.

INDEPENDÊNCIA DO USUÁRIO

O usuário é totalmente independente da forma com que o sistema trata e armazena o texto fonte.

O acesso é feito por meio de cadeias de reconhecimento.

UNIVERSALIDADE

O sistema não é universal; a entrada é efetuada por meio de terminais do tipo IBM-2741 e a saída é entrada para dispositivos de foto-composição e linotipos. Pode utilizar impressora de linha para "display" do texto fonte para sua correção.

FLEXIBILIDADE

O sistema não tem facilidades para a edição de programas.

OPERAÇÃO

A operação do sistema é bastante complexa para um leigo em edição em que as unidades de medida "paica" e "linha" não sejam comumente utilizadas.

O conjunto de comandos é completo e poderoso, o mesmo podendo ser dito dos controles de edição.

Estes são mais complexos pois utilizam unidades de medida para a justificação das linhas.

Quando utilizados dispositivos do tipo 'foto-composição' e 'linotipo' devem ser consideradas as larguras e natureza dos tipos de impressão que serão utilizados.

Nestes dispositivos os caracteres ocupam espaços proporcionais ao seu formato.

SEGURANÇA

A segurança dos arquivos deve ser efetuada pelo usuário utilizando-se de rotinas de "checkpoint-restart" do sistema/360; o sistema não tem comandos específicos de recuperação.

3.2.5. OUTROS.

MAGNETIC TAPE SELECTRIC TYPEWRITER

ASTROTYPE

O primeiro foi desenvolvido pela IBM e o segundo pela Information Control Systems.

São sistemas isolados de computador (off-line) e constam essencialmente de uma máquina de escrever ligada por uma unidade de controle a uma unidade de fita magnética.

O texto é entrada pela máquina de escrever já formatado, sendo gravado na fita magnética com controles do tipo 'fim de linha', 'fim de página', 'tabulação', etc.

Caracteres isolados podem ser substituídos; cadeias sō podem ser substituídas por outras de mesmo tamanho.

Linhas podem ser incluídas, apagadas ou substituídas integralmente.

A edição consiste em descarregar a fita magnética onde foi gravado o texto com os caracteres de controle.

SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM

VIPCOMM

O primeiro sistema foi desenvolvido pela IBM e o segundo pela VIP Systems e diferem apenas no dispositivo de saída.

No ATS a saída é feita através de uma impressora de linhas do sistema/360.

No sistema VIPCOMM o dispositivo de saída é um equipamento de foto-composição.

As funções que ambos executam são as mesmas dos dois sistemas anteriores.

Aceitam textos formatados ou com pequenos controles de formatação.

HYPertext EDITING SYSTEM (HES)

O HES é um sistema editor de textos livres que se baseia num terminal 'vídeo' especial que contém teclas especiais, uma para cada função a realizar sobre o texto.

As funções são executadas com o auxílio de um lápis eletrónico que indica os limites de atuação.

A saída editada é uma impressora de linhas.

Este sistema foi o que introduziu o conceito de 'hipertexto', isto é, da independência do usuário quanto a forma de arquivamento do texto fonte.

3.3. EDITORES DE PROGRAMA

Os editores de programa que serão analisados são:

- a) WYLBUR
- b) QUICK-EDITOR
- c) EDIT-10
- d) TEXT EDITOR AND CORRECTOR

Sobre os sistemas acima foi obtida bibliografia suficiente para que se pudesse analisá-los.

Sob o título de 'outros' serão feitas referências a sistemas sobre os quais a informação obtida foi insuficiente.

Estes sistemas são:

- a) CONVERSATIONAL CONTEXT-DIRECTED EDITOR
- b) TVEDIT
- c) INTERACTIVE PROGRAMMING SUPPORT SYSTEM

3.3.1. WYLBUR

O sistema WYLBUR foi projetado pela Universidade de Stanford para atender a edição de programas em sua rede de time-sharing, sendo operado através de terminal IBM-2741.

O sistema é orientado para a 'linha' equivalente a um cartão de programa, com tamanho fixo.

Além do terminal, pode ser utilizado o dispositivo de perfuração de cartões.

É um sistema de fácil operação, tem um conjunto completo de comandos para a alteração do texto.

Pode editar textos simples suprimindo a numeração das linhas; o formato deve ser dado pelo usuário.

A edição pode ser feita em letras maiúsculas e/ou minúsculas.

A procura de 'cadeias de reconhecimento' foi introduzida no sistema.

3.3.2. QUICK-EDITOR

O sistema QED foi projetado na Universidade da Califórnia (Berkeley) para seu sistema de 'time-sharing', sendo ligado a um computador SDS-930.

O sistema é orientado para a linha de oitenta caracteres, otimizando a utilização da área de armazenamento do texto, gravando apenas os caracteres entrados e separando as linhas pelo caráter "carriage return".

A entrada e saída do sistema é feita, somente, através de teletipo.

O sistema pode ser utilizado para edições simples em que o formato é dado pelo usuário.

O conjunto de comandos, mesmo para edição de programas, não é completo; não há comando para a movimentação de linhas.

Os comandos agem sobre linhas e não sobre cadeias; há facilidades para a alteração de caracteres em uma linha presente na memória.

3.3.3. EDIT-10

O sistema EDIT-10 é um editor de programas da Interactive Sciences Corp. para uso num computador PDP-10.

O sistema é orientado para a linha de tamanho fixo, a entrada é feita através de teletipo e a saída pode ser pelo teletipo ou impressora de linhas.

A edição pode ser feita em letras maiúsculas ou minúsculas.

A edição de textos simples pode ser feita desde que o formato seja dado pelo usuário.

O sistema é de simples operação, tem comandos completos para atualização do texto, poderosos e que podem utilizar-se do recur

so de "cadeias de reconhecimento" para localizar pontos de seu programa.

A alteração de caracteres dentro de uma linha é permitida.

Podem ser criados "labels" para, mais facilmente, poder o usuário deslocar-se sobre o texto.

3.3.4. TEXT EDITOR AND CORRECTOR

O sistema TECO é um editor de programas projetado para uso num computador PDP-10.

O sistema é orientado para uma página de n linhas que deve estar contida num "buffer" da memória para poder ser alterada.

O sistema aceita entrada e saída de qualquer dispositivo do sistema PDP-10.

A edição de textos livres é permitida desde que a formatação seja feita pelo usuário.

É um sistema de difícil manuseio pois todas as operações devem ser feitas a nível de um ponteiro que se situa entre palavras e executadas à esquerda ou à direita da posição indicada pelo ponteiro e sobre um número de caracteres estabelecido.

O conjunto de comandos de atualização do texto é completo; há a característica de

'cadeia de reconhecimento' com apenas uma restrição: a busca só pode ser efetuada no sentido do fim do arquivo.

O conceito de "macros" ou "functions", armazenando sequência de comandos que devem ser repetidos regularmente, foi introduzido.

3.3.5. OUTROS

CONVERSATIONAL CONTEXT-DIRECTED EDITOR

Desenvolvido pelo IBM Cambridge Scientific Center é um editor de programas que permite a edição de textos livres já formatados pelo usuário.

Dentro de uma linha é permitida a substituição de uma cadeia por outra igual ou menor.

Tem a facilidade de procura de "cadeia de reconhecimento" com a restrição de que a cadeia deve estar contida integralmente na linha.

Permite avançar ou retroceder n linhas e, também, a tabulação da linha para a edição de programas com formato fixo de entrada.

TVEDIT

Desenvolvido pela Universidade de Stanford

É um dos mais antigos editores utilizando terminal "vídeo".

É um editor orientado para a linha e tem ponteiros que podem ser deslocados em qualquer direção sobre a tela.

A substituição de caracteres é feita pela superimpressão do novo caráter sobre o antigo.

O texto deve ser dividido em páginas com o tamanho da tela do terminal.

Os comandos são simples e as operações diretamente efetuadas sobre o texto.

Não tem facilidades para formatação.

INTERACTIVE PROGRAMMING SUPPORT SYSTEM

Desenvolvido pela System Development Corp. É um sistema orientado para a linha e utiliza como meio de comunicação um terminal "vídeo".

É um sistema bastante mais poderoso que o TVEDIT pois permite inserir, apagar, substituir, mover e copiar linhas.

As linhas são numeradas e a referência deve ser feita ao número da linha.

CAPÍTULO 4

DESCRIÇÃO DO SISTEMA

4.1. INTRODUÇÃO

O sistema EDITEXT é uma linguagem de comunicação homem-máquina com a finalidade de criar, atualizar, manter, formatar e editar textos livres.

Tem como atividade secundária e complementar o exercício dessas mesmas funções na obtenção de textos de programa.

A comunicação usuário-computador se faz através do uso de "comandos" que obedecem a uma sintaxe definida.

Para auxílio dos comandos na execução da função de formatação do texto, caracteres especiais, denominados "caracteres de controle de edição" são inseridos no texto fonte.

Da mesma maneira que os comandos, os caracteres de controle de edição têm, também, sintaxe precisa e perfeitamente definida.

A combinação de comandos e caracteres de controle dá edição permitem a interação proposta e a obtenção dos resultados desejados.

O sistema é todo modular para permitir, de um lado, facilidade em modificá-lo para aprimorá-lo, de outro, obter uma estrutura de "overlay" que admita sua utilização em computadores de menor porte.

Suas características distintivas são:

- a) simplicidade de uso;
- b) "back-up" automático do arquivo fonte;
- c) a utilização de todos os dispositivos de entrada e saída do sistema, e
- d) uma linguagem, poderosa e precisa, de comunicação usuário:computador.

Nos itens que se seguem serão abordados com mais precisão e profundidade os conceitos e características aqui enumeradas.

Os componentes do sistema que serão analisados são os seguintes:

- a) o programa principal e suas funções;
- b) o analisador sintático e suas funções;
- c) os arquivos utilizados;
- d) os comandos e suas funções;
- e) os controles de edição e suas funções, e
- f) a recuperação de erros.

Ao longo da abordagem destes segmentos, as características definidas como "premissas" para um bom sistema editor de texto tornar-se-ão visíveis ao leitor deste trabalho.

4.2. O PROGRAMA PRINCIPAL

O programa principal do sistema é a raiz da estrutura de "overlay" do mesmo e deve, portanto, residir permanentemente na memória.

Ao projetá-lo uma preocupação foi presente; ocupar o mínimo de memória sem prejuízo de suas funções.

As funções que são privativas do programa principal são:

- a) definir e dar valor a algumas variáveis necessárias ao analisador sintático;
- b) definir os arquivos que serão utilizados pelo sistema;
- c) definir ao sistema que tipo de texto será, por ele, manipulado, e
- d) definir os padrões ("defaults") do sistema.

4.2.1. DEFINIÇÕES DO USUÁRIO

O sistema solicita ao usuário, ao início da sessão, que defina:

- a) se irá manipular texto livre ou programa;
- b) qual o dispositivo de entrada de texto fonte;
- c) qual o dispositivo de saída do texto fonte;

d) qual o dispositivo de saída de texto editado.

Estas definições do usuário são obrigatórias ao início da sessão independente das atividades que serão solicitadas ao editor.

Como resultado dessas definições, ficarão definidos os tamanhos dos vetores ("arrays") de entrada e saída.

4.2.2. ARQUIVOS DO SISTEMA

Todos os arquivos do sistema são declarados no programa principal por serem globais ao sistema.

A entrada de comandos e saída de mensagens é feita por dispositivo alocado pelo sistema e não pelo usuário.

Este dispositivo é denominado de "terminal de trabalho".

A geração do sistema determinará qual ou quais os dispositivos alocados para este "terminal de trabalho".

Na descrição dos comandos são definidos os dispositivos de entrada e saída que utilizam, podendo ser o terminal de trabalho ou os dispositivos alocados pelo usuário.

4.2.3. PADRÕES DO SISTEMA

O programa principal define para o sistema

alguns padrões que poderão ser ou não observados pelo usuário.

Este pode, utilizando comandos, alterar esses valores.

Os padrões definidos no programa principal são:

a) formato de uma página de edição por atribuição de valores a:

- . margens direita e esquerda;
- . posição da primeira e última linhas;
- . posição do título, subtítulo e rodapé da página.

b) formato da "janela" de acesso ao texto fonte;

essa janela permite ao usuário visão de parcela do texto fonte, permitindo-lhe localizar com exatidão a parcela que deseja manipular e verificar se esta está correta; são dados valores a:

- . número de linhas a exibir;
- . número de caracteres que cada linha conterá.

Cabe, finalmente, ao programa principal, após efetuadas as funções descritas, ativar o analisador sintático que assumirá o controle do sistema.

O programa principal dá ao usuário as seguintes informações:

- a) data da sessão;
- b) hora de início da sessão;
- c) hora do término da sessão.

4.3. O ANALISADOR SINTÁTICO

A programa que denominamos de "analisador sintático" não efetua todas as funções de análise sintática dos comandos.

Sua função, entre outras, é a de reconhecer comandos válidos e ativar as rotinas que analisarão os operandos e executarão as funções pretendidas.

Este programa é o pivô do sistema já que este atua segundo comandos emitidos.

O analisador sintático ativa o comando e este, após cumprir sua função, retorna o controle àquele.

Outra função deste programa é definir, aos que lhe estão subordinados hierarquicamente, variáveis globais que devem manter valor e dimensão constante ao passarem de um para outro.

Além disso, o programa emite mensagem solicitando entrada de comandos e caso o comando fornecido seja inválido acusa o erro e lista os válidos.

4.3.1. ATIVAÇÃO DOS COMANDOS

Os comandos válidos são reconhecidos por este programa que analisa a entrada, caráter a caráter, até reconhecer um "mnemônico" válido.

Reconhecendo, ativa a função; caso contrário, emite a mensagem de erro já citada.

4.3.2. VARIÁVEIS GLOBAIS

Algumas variáveis de trabalho são globais ao sistema e devem manter seu valor constante independente da rotina que esteja sendo executada.

Qualquer alteração de valor que estas variáveis sofram, este novo valor deve ser válido para todo o sistema.

Estas variáveis são:

- a) apontadores de registros;
- b) apontadores de caracteres;
- c) contadores especiais e
- d) chaves (switches) de controle.

4.3.3. VETORES DE ENTRADA E SAÍDA

Os vetores de entrada e saída têm sua dimensão definida no programa principal sendo, pois, de tamanho variável.

Cabe ao analisador sintático a função de, usando o tamanho definido no programa principal, fixar-lhe a dimensão para uso dos módulos executivos do sistema.

Estes vetores são:

- a) entrada de texto fonte;

- b) saída de texto fonte e
- c) saída de texto editado.

4.3.4. MENSAGENS

No início de uma sessão ou após a execução de um comando o analisador solicita novo comando, emitindo a seguinte mensagem:

? COMANDO

Se o mnemônico do comando fornecido ao analisador for inválido, este emitirá a seguinte mensagem:

** COMANDO INVALIDO ** VALIDOS = XXX,YYY,
ZZZ ...

4.4. ARQUIVOS DO SISTEMA

O sistema utiliza-se de dois tipos de arquivos:

a) arquivos externos e

b) arquivos internos.

São denominados arquivos externos aqueles utilizados para a comunicação usuário-sistema, estando alguns deles sob controle do sistema e outros, do usuário.

São arquivos internos aqueles de uso exclusivo do sistema aos quais o usuário não tem acesso.

4.4.1. ARQUIVOS EXTERNOS

Os arquivos externos, como foi dito, são aqueles que servem para a comunicação entre o usuário e o sistema.

É através destes que o usuário fornece comandos e texto e recebe mensagens, texto fonte ou texto editado.

Utilizam-se, estes arquivos, de dispositivos de baixa velocidade de transmissão e são todos do tipo "registro-unitário" ("unit-record").

4.4.1.1. ARQUIVO DE ENTRADA FONTE

Através deste arquivo o usuário fornece o texto fonte ao sistema.

Este arquivo é utilizado apenas pelos comandos que dão entrada de grandes volumes de texto.

Os dispositivos que podem ser alocados para este arquivo são:

- a) leitora de cartões perfurados;
- b) leitora de fita de papel perfurada;
- c) terminal tipo máquina de escrever;
- d) terminal tipo vídeo e
- e) teletipo

O tamanho do registro ("logical record length") definido para este arquivo é o máximo aceito pelo dispositivo.

4.4.1.2. ARQUIVO DE SAÍDA FONTE

Da mesma forma que o arquivo de entrada fonte, este arquivo é utilizado somente para a saída de grande volume de texto fonte.

Os dispositivos que podem ser alocados para este arquivo são:

- a) impressora de linhas;

b) terminal tipo máquina de escrever e

c) teletipo.

O tamanho do registro definido para este arquivo é o máximo aceito pelo dispositivo.

4.4.1.3. ARQUIVO DE SAÍDA EDITADA

Este arquivo é utilizado para dar saída ao texto já em sua forma final, acabada.

Os dispositivos aceitos para este arquivo são:

a) impressora de linhas;

b) terminal tipo máquina de escrever;

c) teletipo e

d) perfuradora de cartões, somente em modo de edição de programa.

O tamanho do registro definido para este arquivo é o máximo aceito pelo dispositivo, como os anteriores.

4.4.1.4. ARQUIVO DE TRABALHO

O arquivo de trabalho é utilizado

para a entrada de comandos, saída de mensagens e pequenas entradas e saídas de texto fonte.

O dispositivo utilizado deve ser de entrada e saída, podendo ser alocados:

- a) terminal tipo máquina de escrever;
- b) terminal tipo vídeo e
- c) teletipo.

Quando o sistema não tem um dos dispositivos acima assinalados, pode o terminal de trabalho simulado por um conjunto "leitora de cartões/impressora de linhas".

O tamanho do registro definido para este arquivo é máximo aceito pelo dispositivo quando utilizado um terminal, ou os máximos quando simulado.

4.4.2. ARQUIVOS INTERNOS

Os arquivos internos são transparentes ao usuário e sua definição é feita pelo sistema.

Estes arquivos destinam-se a armazenar os textos fonte, formatado e auxiliares.

Utilizam-se estes arquivos de dispositivos de alta velocidade de transmissão e que permitam blocar registros.

4.4.2.1. ARQUIVO DE TEXTO FONTE

O arquivo de texto fonte destina-se a armazenar o texto em sua forma primitiva, sem formato e no qual es tão embebidos os caracteres de controle de edição.

Os registros deste arquivo têm tamanho fixo e, esteja o sistema em modo de edição de texto livre ou programa, têm o seguinte formato:

1	2	3	4
---	---	---	---

campo 1 - ponteiro para o registro sucessor;

campo 2 - ponteiro para o registro antecessor;

campo 3 - número de caracteres con tidos no registro, e

campo 4 - caracteres do texto fonte

O arquivo é de acesso direto e manipulado como uma "lista duplamente ligada" ("doubled linked list").

Esta lista é circular, isto é, o último registro da lista aponta como sucessor o primeiro e, reciprocamente, o primeiro registro da lista aponta como antecessor o último.

Desse modo podemos percorrer a lista em qualquer sentido, facilitando sobremaneira inserir ou apagar qualquer registro.

Em determinado momento o arquivo que contém T registros, terá este desdobramento.

$$T = m + n$$

onde m = número de registros em uso
e n = número de registros inativos.

O conjunto de registros n forma o "pool" de registros em disponibilidade.

No início $T = n$ e $m = \emptyset$; se o arquivo estiver com todos os seus registros em uso, $T = m$ e $n = \emptyset$.

O registro é reconhecido como estando em disponibilidade se o campo

"número de caracteres contido no registro" for igual a 0 (zero); se este número for $\neq 0$ o registro está em uso e o número indica quantos caracteres ali estão contidos.

Este arquivo deve residir em dispositivo que permite acesso direto, isto é, em disco ou tambor magnético.

4.4.2.2. ARQUIVO DE SEGURANÇA

O arquivo de segurança é organizado e manipulado da mesma maneira que o arquivo de texto fonte.

Seus registros têm o mesmo formato e são atualizados ao mesmo tempo que os daquele.

Esta característica permite-nos um "back-up" constante do texto fonte.

Para que este "back-up" seja efetivo esses dois arquivos devem residir em dispositivos físicos diversos.

4.4.2.3. ARQUIVO DE TEXTO FORMATADO

O arquivo de texto formatado conterá o texto fonte já manipulado pelo módulo de formatação.

Cada registro do arquivo corresponde a uma linha de texto, pronta para editar.

Somente as linhas que devem ser editadas são gravadas.

Para que isto seja possível os registros deste arquivo têm o seguinte formato.

1	2	3
---	---	---

campo 1 - número de página

campo 2 - número da linha formatada

campo 3 - caracteres da linha (normais, acentos e sublinhado).

O campo número da página permite a edição parcial do texto.

O campo número da linha é que nos possibilita que sejam gravados apenas os registros correspondentes a linhas que efetivamente contenham caracteres.

Este arquivo é organizado sequencialmente e pode residir num dos dispositivos seguintes:

a) disco magnético

b) fita magnética

A opção de escolha do dispositivo é de geração do sistema; será função das unidades disponíveis.

4.4.3. ARQUIVOS AUXILIARES

O sistema utiliza diversos arquivos auxiliares da edição.

A organização destes arquivos depende da finalidade a que se destinam.

4.4.3.1. ARQUIVO DA TABELA DE CONTEÚDO

Este arquivo é gerado durante a formatação do texto e permite ao usuário editar um índice das matérias contidas no texto.

É um arquivo de acesso direto e tem seus registros o seguinte formato:

1	2	3
---	---	---

campo 1 - número da página

campo 2 - nível do texto

campo 3 - caracteres do texto (normais e acentos).

O campo de nível tem função na edição da tabela.

4.4.3.2. AROUIVO DA TABELA DE REFERÊNCIA CRUZADA

Se o usuário desejar editar uma tabela em que constem páginas de ocorrência de determinadas "palavras-chave", este arquivo é gerado durante a formatação do texto e, posteriormente, é a tabela editada.

Este arquivo é de acesso sequencialmente indexado e seus registros têm o seguinte formato:

2	1
---	---

campo 1 - página da ocorrência

campo 2 - texto da palavra-chave (caracteres normais e acentos).

4.4.3.3. ARQUIVO DE NOTAS AO PÉ DA PÁGINA

Este arquivo permite a inserção de notas ao pé da página.

Toda a vez que o módulo de formatação encontra no texto fonte o caráter de controle que indica a existência de uma nota esta é gravada nesse arquivo que tem em seus registros somente o texto (caracteres normais e acentoa) da nota.

Sempre que o número de linhas já formatadas acrescido do número de linhas de notas preencher uma página neste arquivo é descarregado no arquivo de texto formatado; o mesmo ocorre se for forçado um salto para nova página. É de acesso direto.

4.5. COMANDOS

Os comandos são os módulos executivos do sistema, formando seu conjunto a linguagem de comunicação entre o usuário e o sistema.

Além de executarem as funções para as quais foram projetados, fazem a análise de seus operandos para verificar se estão corretos.

Sinteticamente, os comandos executam:

- a) entrada e saída de texto fonte;
- b) modificação neste mesmo texto;
- c) sua formatação e
- d) a edição do texto formatado.

São necessários, ainda, comandos de serviço que auxiliem aqueles e permitam a recuperação do arquivo texto fonte.

De acordo com a semelhança de funções executadas, os comandos podem ser divididos em:

- a) Comandos de Entrada e Saída;
- b) Comandos de Alteração;
- c) Comandos de Formatação;
- d) Comandos de Edição e

e) Comandos de Serviço.

Neste capítulo serão estudados sua sintaxe, descrição, detecção de erros e rotinas auxiliares.

4.5.1. SINTAXE DOS COMANDOS

Os comandos do sistema têm sua sintaxe representada, em metalinguagem, na forma abaixo:

```
(COMANDO)::=(MNEMONICO)(ESPAÇO)(CORPO DO
                COMANDO)(ESPAÇO)(COMENTÁRIO)
(MNEMONICO)::=(LETRA)|(MNEMONICO)(CARATER)
(ESPAÇO)::=(CARĀTER BRANCO)|(ESPAÇO)(CARĀ
                TER BRANCO)
(CORPO DO COMANDO)::=(OPERANDO)|(CORPO DO
                COMANDO),(OPERANDO)
(COMENTÁRIO)::=(CADEIA)
(LETRA)::=A|B|C|.....|x|y|z
(CARĀTER)::=/qualquer carĀter do EBCDIC/
(CARĀTER BRANCO)::=␣
(OPERANDO)::=(OPERANDO NUMÉRICO)|(OPERANDO
                ALFABÉTICO)|(OPERANDO CADEIA)|
                (VAZIO)|;
(CADEIA)::=(CARĀTER)|(CADEIA)(CARĀTER)
(OPERANDO NUMÉRICO)::=(NÚMERO)|(OPERANDO
                NUMÉRICO)(NÚMERO)
(OPERANDO ALFABÉTICO)::=(MNEMONICO)
(OPERANDO CADEIA)::=/(CADEIA)/
(NÚMERO)::=1|2|3|4|5|6|7|8|9|0
```

4.5.1.1. MNEMÔNICO

O mnemônico de um comando é um conjunto de caracteres de tamanho ilimitado e que não pode conter o caráter "branco".

Para reconhecimento do comando são necessários, apenas, de um a quatro caracteres; os que excedem esse número são ignorados pelo sistema e podem ser assemelhados a comentários.

Exemplificando o comando "apagar", todas as formas a seguir são válidas:

A

APAG

APAGAR

APAGAR/TEXTO-FONTE+DO*ARQUIVO

Se a última forma fosse apresentada sem os caracteres especiais, na análise do comando as cadeias TEXTO, FONTE, DO e ARQUIVO seriam entendidas como operandos.

4.5.1.2. CORPO DO COMANDO

O corpo do comando é o conjunto dos operandos deste, devendo estar separados entre si pelo caráter vírgula (",").

Os limites de uma cadeia são estabelecidos pelo caráter barra ("/"); este caráter pode ser incluído na cadeia desde que em duplicata (exemplo: a cadeia/A=B//C/ será entendida como A=B/C).

Se um dos operandos que não seja o último for omitido, a vírgula correspondente deverá permanecer.

4.5.1.3. COMENTÁRIOS

O caráter branco (" ") após um operando indica ser este o último deles.

Todos os caracteres que se seguirem serão ignorados pelo sistema.

Esta facilidade deve ser utilizada para documentação da sessão.

O comentário não pode exceder o fim do registro no qual está contido o último operando; no momento em que esse registro é entregue ao sistema a rotina correspondente é ativada.

4.5.1.4. ATUAÇÃO DOS COMANDOS

Conforme esteja o sistema em modo de edição de texto livre ou edição de programa o reconhecimento do início e fim da ação dos comandos é diferenciada.

EDIÇÃO DE TEXTO LIVRE

Em edição de texto livre, os comandos reconhecem o início e fim de sua ação por meio de "cadeias de reconhecimento", ou seja, com a utilização de "operandos cadeia".

EDIÇÃO DE PROGRAMA

Em modo de edição de programa, o sistema não reconhece como operandos válidos as "cadeias de reconhecimento", admitindo somente operandos numéricos que representam "linhas de programa".

EXCEÇÃO

O comando "localizar" aceita tanto operando numérico quanto cadeias de reconhecimento.

4.5.2. DESCRIÇÃO DOS COMANDOS

Os comandos têm, usualmente, dois formatos: um para a edição de texto livre e outro para a edição de programa.

Alguns comandos tem um terceiro formato, válido para textos livres e programas, e que quando aplicados referem-se a todo o arquivo, fonte ou formatado.

A indicação do mnemônico de comando é feita sublinhando-se os caracteres que o compõe.

A notação "CADn" indica "cadeias de reconhecimento" enquanto que "REGn" indica "linha de programa".

4.5.2.1. COMANDOS DE ENTRADA E SAÍDA

Os comandos de entrada e saída manipulam o arquivo fonte e utilizam-se dos dispositivos definidos no programa principal.

Estes comandos são:

COMEÇAR

Este comando começa a entrada de texto fonte.

A inicialização dos arquivos consiste em zerar o campo "número de caracteres contidos" de todos os registros.

O último registro do arquivo, em atividade, é o registro cujo "ponteiro do registro sucessor" é igual a 1 (um).

EXIBIR $\left\{ \begin{array}{l} ; \\ (\text{REG1}), (\text{REG2}) \\ (\text{CAD1}), (\text{CAD2}) \end{array} \right\}$

Este comando exibe o texto fonte delimitado pelos operandos ou todo o arquivo no caso do formato 1.

A linha de exibição tem o tamanho definido no comando "quadrar" ou o padrão do sistema.

O padrão do sistema é determinado no programa principal ao ser alocado o dispositivo de saída de texto fonte.

A função deste comando é a de fornecer ao usuário uma cópia do texto fonte, para trabalho, verificação e correção.

4.5.2.2. COMANDOS DE ALTERAÇÃO

Os comandos de alteração atuam sobre o texto fonte, inserindo, movendo, substituindo, copiando ou apagando parte dele.

As entradas e saídas efetuadas

por este comando são feitas através do "terminal de trabalho" pois pressupõem pequenos volumes de dados.

APAGAR $\left\{ \begin{array}{l} (\text{REG1}), (\text{REG2}) \\ (\text{CAD1}), (\text{CAD2}) \end{array} \right\}$

Este comando apaga o texto compreendido entre os operandos, inclusive.

Os registros que tiverem todos os seus caracteres apagados serão retirados da lista e tornado inativos.

COPIAR $\left\{ \begin{array}{l} (\text{REG1}), (\text{REG2}), (\text{REG3}) \\ (\text{CAD1}), (\text{CAD2}), (\text{CAD3}) \end{array} \right\}$

Este comando copia após o primeiro operando todo o texto compreendido entre o segundo e o terceiro operandos.

O texto copiado não é destruído (apagado).

INSERIR $\left\{ \begin{array}{l} (\text{REG1}) \\ (\text{CAD1}) \end{array} \right\}$

Este comando insere após o operando um texto de tamanho ilimitado e delimitado ao seu final pelos caracteres //.

MOVER $\left\{ \begin{array}{l} (\text{REG1}), (\text{REG2}), (\text{REG3}) \\ (\text{CAD1}), (\text{CAD2}), (\text{CAD3}) \end{array} \right\}$

Este comando copia após o primeiro operando todo o texto compreendido entre o segundo e terceiro operandos, destruindo (apagando) após o texto copiado em sua posição original.

SUBSTITUIR $\left\{ \begin{array}{l} (\text{REG1}), (\text{REG2}) \\ (\text{CAD1}), (\text{CAD2}) \end{array} \right\}$

Este comando substitui o texto compreendido entre os operandos, inclusive, por um texto de tamanho ilimitado, delimitado pelos caracteres //.

4.5.2.3. COMANDOS DE FORMATAÇÃO

Os comandos de formatação têm como função a conversão do texto fonte em texto com formato, pronto para ser editado.

Estes comandos são:

FORMATAR $\left\{ \begin{array}{l} (\text{REG1}), (\text{REG2}) \\ (\text{CAD1}), (\text{CAD2}) \end{array} \right\}$

Este comando formata o texto fonte em sua totalidade e iniciando a numeração de página pelo número 1

(um) quando adotado o primeiro formato.

Nos outros dois, formato o texto compreendido entre os operandos, inclusive, iniciando a numeração também pelo número 1 (um).

Este comando detecta erros nos caracteres de controle da edição, listando-os no terminal de trabalho, informando qual e onde ocorreu o erro.

A localização do erro é dada pelo número do registro do arquivo fonte onde foi detectada a ocorrência.

De posse deste número e utilizando o comando "localizar" tem-se acesso ao trecho do texto onde foi localizado o erro e, pela utilização dos comandos de alteração, efetuar a correção.

PAGINAR { (LINHA DO TÍTULO), LINHA DO SUBTÍTULO), (PRIMEIRA LINHA), (ÚLTIMA LINHA), (LINHA DO RODAPÉ), (MARGEM ESQUERDA), (MARGEM DIREITA) }

Este comando define para o comando "formatar" o "lay-out" da página que será editada.

É possível utilizar o padrão definido no programa principal e alterar um ou todos parâmetros que definem a página.

4.5.2.4. COMANDOS DE EDIÇÃO

O comando de edição transfere do arquivo formatado para o arquivo de saída de texto editado, definido no programa principal, todo ou em parte.

$$\text{EDITAR} \left\{ \begin{array}{l} ; \\ (\text{REG1}), (\text{REG2}) \\ (\text{CAD1}), (\text{CAD2}) \end{array} \right\}$$

Este comando edita, no primeiro formato, todo o arquivo formatado.

Nos demais formatos, edita o texto compreendido entre os operandos, inclusive.

4.5.2.5. COMANDOS DE SERVIÇO

Os comandos de serviço auxiliam os executivos, definindo quadros, recuperando arquivos, localizando cadeias ou registros, compactando arquivos e encerrando a sessão do sistema.

COMPACTAR

Após um número indeterminado de alterações do texto fonte, dois fatos podem ocorrer:

- a) não haver mais registros disponíveis e
- b) haver diminuição da eficiência em virtude da completa desorganização do arquivo.

A ocorrência de qualquer um deles determina a necessidade de compactação do arquivo.

Essa compactação elimina todos os espaços desnecessários e reorganiza o arquivo colocando os registros em sequência.

Concluída a compactação o sistema informa ao usuário quantos caracteres (e registros) há em disponibilidade no arquivo.

A solicitação de compactação do arquivo é feita pelo sistema quando ocorre o fato a e pelo usuário quando ocorre b.

FIM

Este comando encerra uma sessão do

sistema, fecha os arquivos e devolve o controle ao programa principal que emite as mensagens de fim de sessão.

LOCALIZAR $\left. \begin{array}{l} (REG1) \\ (CAD1) \end{array} \right\}$

Este comando localiza o último registro gravado e exibe-o no arquivo de trabalho quando utilizado o formato 1 (um).

Nos demais formatos, exibe a cadeia que inicia no primeiro caráter do operando.

Em qualquer dos três formatos o número de caracteres exibidos é aquele definido pelo comando "quadrar" ou segundo o padrão do sistema.

QUADRAR $\left. \begin{array}{l} (LINHAS), (CARACTERES/LI \\ NHA) \end{array} \right\}$

Este comando define um quadro para exibição de caracteres do texto fonte, na forma de m linhas de n caracteres.

Este quadro é utilizado para exibir texto fonte localizado pelo comando "localizar".

REPRODUZIR { (OPÇÃO) }

Este comando é utilizado para a recuperação do arquivo de texto fonte propriamente dito ou arquivo de segurança daquele de acordo com a opção dada.

Na recuperação de um daqueles arquivos o processo utilizado é o de copiar o arquivo não danificado.

Após a conclusão da reprodução, que o sistema informa ao usuário, deve ser localizado o último registro para verificar se houve perda da informação.

4.5.3. DETEÇÃO DE ERROS

Uma das funções exercidas pelos comandos é a de analisar seus operandos.

Se um erro for detetado haverá a emissão de uma mensagem indicando o erro cometido e o comando é descontinuado.

Os erros que podem ser detetados por informações errôneas fornecidas pelos usuários são:

** NÚMERO DO REGISTRO MAIOR QUE NÚMERO TOTAL DE REGISTROS**

** REGISTRO XXXX FORA DE USO **

** CADEIA VAZIA **

** CADEIA NÃO ENCONTRADA NO TEXTO **

** CADEIA NÃO INICIADA COM ' **

** CARÁTER INVÁLIDO APÓS CADEIA **

** CADEIA COM CARACTERES EM EXCESSO **

** CARÁTER ILEGAL EM OPERANDO NUMÉRICO **

** OPCÃO INVÁLIDA **

Existem erros que são detetados por insuficiente informação do usuário.

Estes erros ocorrem sempre que:

- a) alguma operação deve ser feita após um registro ou cadeia e este operando não é fornecido.

Exemplo: se no comando COPIAR (CAD1), (CAD2), (CAD3) o operando (CAD1) não é fornecido o sistema fica sem saber após o que o texto deve ser copiado.

- b) numa operação semelhante a anterior não é indicado o texto a processar.

Exemplo: no mesmo comando, se os operandos (CAD2) e (CAD3) não forem fornecidos o sistema ignora o que copiar.

- c) em existindo dois operandos delimitando o início e fim, é omitido o que indica o início.

Exemplo: ainda no mesmo comando, a omissão do operando (CAD2) deixa de informar ao sistema onde se inicia a operação.

Estes tipos de erros geram mensagens com o seguinte formato:

? APÓS O QUE xxxxxxxxxxxx

? O QUE xxxxxxxxxxxx

? ONDE INICIA xxxxxxxxxxxx

A cadeia "xxxxxxxxxx" é substituída pela função a ser executada pelo comando.

Quando existem dois parâmetros delimitando a ação de um comando, a emissão do segundo delimitador indica que a ação é efetuada somente sobre o texto delimitado pelo primeiro operando.

Exemplo: no comando já visto, é válida a omissão do operando (CAD3), indicando-se com isto que será copiada somente a (CAD2).

4.5.4. ROTINAS AUXILIARES

Os comandos utilizam rotinas auxiliares para a execução de suas funções.

Estas rotinas efetuam cálculos, tratam cadeias, atualizam a lista de registros e fazem entrada e saída.

4.5.4.1. ROTINAS DE CÁLCULO

CALCULA OPERANDO

Esta rotina recebe como entrada uma cadeia de caracteres e a converte num número.

Se na cadeia houver um caráter não numérico, este é detetado e mensagem de erro emitida.

4.5.4.2. TRATAMENTO DE CADEIAS

EXAMINA CADEIAS

Esta rotina examina cadeias de reconhecimento e informa ao sistema quais os caracteres que a compõem e quantos.

Os seguintes erros são detetados por esta:

- a) se o número de caracteres excede o número fixado pelo sistema;

- b) se a cadeia não inicia pelo delimitador ';
- c) se a cadeia não se encerra pelo delimitador ';
- d) se há caracteres inválidos após a cadeia; são válidos , e e
- e) se a cadeia é vazia.

PROCURA NO TEXTO

Esta rotina procura uma cadeia de reconhecimento dada e informa ao sistema onde ela inicia e onde conclui.

Se a cadeia não estiver contida no texto o sistema é informado e a mensagem de erro correspondente é fornecida.

4.5.4.3. ATUALIZAÇÃO DE LISTA DE REGISTROS

PROCURA REGISTRO

Esta rotina procura, no arquivo fonte, um registro em disponibilidade ("avail").

Se o registro é encontrado, o sistema é informado do número de seu ponteiro.

Se o registro não é encontrado, uma compactação do arquivo é solicitada.

INSERE REGISTRO

Esta rotina insere na lista de registros ativos um que tenha sido ativado.

A inserção é feita pela mudança do ponteiro de registro sucessor daquele que o antecederá e o ponteiro de registro antecessor daquele que o sucederá; seus ponteiros são também ajustados.

RETIRA REGISTRO

Esta rotina retira da lista de registros ativos um que foi tornado inativo.

As operações feitas sobre os ponteiros dos registros antecessor e sucessor deste são as inversas daquelas que foram efetuadas numa inserção.

O campo de "número de caracteres contidos" é zerado.

4.5.4.4. ENTRADA E SAÍDA

ENTRADA EXTERNA

Esta rotina fornece ao sistema entradas de origem externa, isto é, comandos e texto fonte.

Os caracteres entrados são fornecidos um a um e quando o vetor de entrada é esgotado nova entrada é feita.

ENTRADA INTERNA

Esta rotina lê registros do arquivo fonte e fornece-os ao sistema para que sejam manipulados.

Esta rotina detecta dois erros:

- a) se o registro não está em uso, e
- b) se o número do registro excede o número total de registros do arquivo.

GRAVA ARQUIVO FONTE

Esta rotina grava, simultaneamente, nos arquivos fonte e segurança um registro.

Cabe, ainda, a esta rotina a função de compactar o registro que teve parte de seus caracteres apagados.

O caráter apagado é representado

pelo zero binário (em hexadecimal
= X'00').

4.6. CONTROLES DE EDIÇÃO

Os controles de edição, em conjunto com o "lay-out" da página de edição definida, fornecem meios aos módulos de formatação e de edição para a obtenção do arquivo formatado e da edição final.

Estes controles fazem parte integrante do texto fonte, vêm nele embebidos e podem ser manipulados pelos comandos de alteração.

As funções requeridas dos controles de edição são as de iniciar novo parágrafo, saltar linhas em branco, duplicar cadeias, criar índices e tabelas, modificar caracteres, etc..

Neste capítulo serão estudados sua sintaxe, sua descrição, detecção de erros e rotinas auxiliares.

4.6.1. SINTAXE DE CONTROLE DE EDIÇÃO

A definição sintática de um controle de edição pode ser, em metalinguagem, assim feita:

(CONTROLE DE EDIÇÃO)::=(CONTROLE)|(CARÁTER ESPECIAL)

(CONTROLE)::=\$(MNEMÔNICO)(CORPO DO CONTROLE)
\$

(MNEMÔNICO)::=(LETRA)|(LETRA)(LETRA)

(CORPO DO CONTROLE)::=(VAZIO)|(OPERANDO NUMÉRICO)|(OPERANDO ALFABÉTICO)|(OPERANDO MISTO)

(OPERANDO NUMÉRICO)::=(NÚMERO)|(OPERANDO NUMÉRICO)(NÚMERO)

(OPERANDO ALFABÉTICO)::=(CADEIA)

(OPERANDO MISTO)::=(OPERANDO NUMÉRICO)|(MNEMÔNICO)|(OPERANDO MISTO),
(OPERANDO NUMÉRICO)|(OPERANDO MISTO), (OPERANDO ALFABÉTICO)

(CARÁTER ESPECIAL)::=(CARÁTER MODIFICADOR)|(CARÁTER SIMULADOR)|
{CARÁTER BRANCO ESPECIAL)

(CARÁTER BRANCO ESPECIAL)::=&

(CARÁTER MODIFICADOR)::=%

(CARÁTER SIMULADOR)::='

A definição sintática de (LETRA), (NÚMERO) e (CADEIA) é a mesma já dada no estudo de comandos.

As funções dos "caracteres especiais" serão descritas em conjunto com os demais controles.

4.6.1.1. MNEMÔNICO

Nos controles de edição, o mnemônico tem formato fixo e é composto de uma ou duas letras.

Isto é devido ao fato de que o (corpo do controle) segue-se imediatamente após o mnemônico, não havendo caráter delimitador entre ambos.

Exemplo: \$Ccomando\$ é um controle válido para centrar (mnemônico=C) a cadeia 'comando' que é um operando alfabético.

4.6.2. DESCRIÇÃO DOS CONTROLES DE EDIÇÃO

4.6.2.1. CONTROLES

ALTERA NUMERAÇÃO DE PÁGINA

`$A(novo número)$`

Este controle faz com que o contador de número de páginas seja alterado para o (novo número), a partir da corrente página.

Se seu valor for igual a zero isto equivale a impedir a numeração da página corrente; a página seguinte terá o número dois.

BLOCO DE LINHAS

`$B(número de linhas),(texto do bloco)$`

Este controle define um bloco de (número de linhas) linhas em branco que devem estar contidas na mesma página.

O sistema reserva duas destas linhas: a primeira para o (texto do bloco) e a segunda separando esse do restante texto.

Se o bloco não couber na página corrente, será jogado na página

seguinte.

CENTRA TEXTO

\$C(texto a centrar)\$

Este controle centra, entre as margens esquerda e direita, o (texto a centrar).

DUPLICA TEXTO

\$D(número),(texto)\$

Este controle provoca a duplicação do (texto) o (número) de vezes dado.

Se o(número) não é dado, o (texto) é duplicado uma vez.

Se o (texto) não é dado, um (número) de linhas é preenchido com o caráter . (ponto).

Se nenhum dos operandos é dado a linha em montagem é enchida de . (ponto).

NOTA AO PÉ DA PÁGINA

\$F(texto)\$

Este controle grava o (texto), com tamanho máximo igual à largura da

linha, no arquivo de "notas ao pé da página".

Quando o número de notas gravadas acrescido do número de linhas formatadas preencher uma página, o arquivo de notas é descarregado no arquivo formatado; o mesmo ocorre se nova página é forçada.

CABEÇALHOS

\$H(tipo),(texto)\$

Este controle define o (texto) do título ou subtítulo de acordo com o (tipo) dado.

O título será sempre centrado na linha e o subtítulo justificado pela margem esquerda.

O texto não poderá ter dimensão maior que a largura da página.

Se for fornecido somente o (tipo) o sistema suprime este na seguinte página.

JUSTIFICAR TEXTO

\$JE(coluna),(texto)\$

\$JD(coluna),(texto)\$

Este controle justifica a esquerda ((JE) ou a direita (JD), na (coluna) dada, o (texto).

Se a (coluna) não é dada é assumida a margem esquerda ou direita respectivamente.

Ele somente posiciona o (texto); deve ser associado com outro para que seja gravado no arquivo formatado.

REDEFINE MARGENS

\$R(margem esquerda),(margem direita)\$

Este controle é utilizado para redefinir as margens durante o processo de formatação.

Podem ser redefinidas, uma, outra ou ambas as margens.

Se o controle não tiver operandos as margens retornam ao "set-up" inicial.

NOVA PÁGINA

\$N

Este controle força a gravação da linha em montagem e posiciona o apontador de início de linha na

(coluna) dada.

Se aquele valor for igual a zero, assume o valor anterior, isto é, do último controle de parágrafo emitido.

Não é permitido iniciar parágrafos fora das margens.

SALTAR LINHAS

\$S(número)\$

Este controle força a gravação da linha em montagem e o (texto) é formatado sem justificação.

O (texto) é de tamanho ilimitado e seu final força a gravação da linha.

SUBLINHAR

\$U(texto)\$

Este controle faz com que o (texto) dado seja sublinhado.

O (texto) pode ter tamanho ilimitado; os caracteres "branco" normais não são sublinhados.

TABELA DE REFERÊNCIA CRUZADA

\$X(texto)\$

Este controle faz com que o (texto) seja formatado e simultaneamente gravado numa arquivo especial para posterior edição de uma tabela de referências.

O tamanho do (texto) é limitado em trinta (30) caracteres.

TABELA DE CONTEUDOS

\$Y(nível),(texto)\$

Este controle não só formata o (texto) bem como grava-o juntamente com o (nível) e número da página em arquivo especial que poderá ser posteriormente editado.

O (nível) indica um deslocamento do (texto) em relação à margem esquerda.

O número máximo de caracteres do (texto) é de 40 (quarenta).

PAUSA

\$WS\$

Este controle determina a gravação no arquivo de texto formatado de uma linha "dummy" em que o campo "caracteres contidos" terá apenas o caráter "BELL" gravado.

Quando em modo de edição esta li nha for encontrada, haverá uma pausa na edição e o usuário intervirá para efetuar uma ação que previamente planejou.

MAIUSCULA

\$M(texto)\$

Este controle faz com que o (texto) seja convertido para maiúsculas.

4.6.2.2. CARACTERES ESPECIAIS

CARÃTER BRANCO ESPECIAL

&

O caráter branco especial é um caráter de enchimento de vazios e é utilizado para:

- a) impedir, quando é feita a justificação de uma linha, que entre duas cadeias sejam inseridos brancos de justificação.
- b) permitir uma cadeia de brancos na linha;
- c) permitir que o caráter branco seja sublinhado.

O caráter branco especial é editado como se fosse um branco normal.

CARÁTER MODIFICADOR

%

Este caráter modifica o que imediatamente lhe segue e é utilizado:

- a) para a representação dos próprios caracteres especiais (&, %, \$ e ').
- b) para a representação de caracteres não constantes no dispositivo de entrada e constantes no de saída.
- c) tornar o caráter que o segue "maiusculo".

CARÁTER SIMULADOR

Quando, nem na entrada, nem na saída, existirem os caracteres de representação de acentos e cedilha, este caráter é utilizado para sua simulação.

A técnica utilizada é a de superimpressão de um caráter qual quer simulando os acentos e da vírgula para o cedilha.

4.6.3. DETEÇÃO DE ERROS

O módulo de formatação deteta erros decorrentes do mau emprego que deles é feito.

Esses erros são listados no "terminal de trabalho" e informam:

- a) registro onde ocorreu, e
- b) qual o erro ocorrido.

A detecção de erros não descontinua a formatação, esta prossegue até o final determinado pelo usuário.

É aconselhável a formatação de pequenos trechos do texto fonte, corrigindo-os e ao final, formatar todo ele.

Os erros que podem ser detetados são:

** CARÁTER NÃO NUMÉRICO EM OPERANDO NUMÉRICO **

** OPÇÃO INVÁLIDA **

** TEXTO MUITO LONGO **

** MARGENS ULTRAPASSADAS **

** TEXTO VAZIO **

As causas principais de ocorrência de erros são:

- a) falta do segundo delimitador \$
- b) falta de delimitador entre operandos.

4.6.4. ROTINAS AUXILIARES

Existem três tipos de rotinas auxiliares do módulo de formatação.

Estas rotinas são:

- a) rotinas de cálculo
- b) rotinas de entrada e saída
- c) rotinas de tratamento de caráter especial.

4.6.4.1. ROTINAS DE CÁLCULO

CALCULA OPERANDOS NUMÉRICOS

Esta rotina é invocada sempre que um operando numérico deve ser calculado.

Ele deteta caracteres não numéricos e causa a emissão de mensagem de erro.

4.6.4.2. ROTINAS DE ENTRADA E SAÍDA

OBTEM CARÁTER

Esta rotina obtem do arquivo fonte caracteres para o módulo de formatação.

FORMATATA JUSTIFICANDO

Esta rotina formata uma linha justificando-a entre a margem esquerda ou coluna de início de parágrafo e a margem direita.

A justificação é feita pela inserção de brancos entre os brancos que separam as palavras do texto.

A distribuição de brancos é feita de maneira uniforme ao longo da linha, de acordo com o algoritmo descrito no apêndice II.

FORMATATA SEM JUSTIFICAR

Esta rotina grava, simplesmente, uma linha no arquivo de texto formatado, sem justificar.

É necessária para: linhas textuais, linhas centradas, programa, etc..

FORMATATA NOTAS AO PÉ DA PÁGINA

Esta rotina é semelhante à anterior; a única diferença é a origem do texto, que neste caso origina-se do arquivo de notas ao pé da página.

4.6.4.3. TRATA CARÁTER ESPECIAL

CARÁTER ESPECIAL

Esta rotina fazo o tratamento de caracteres especiais definidos para o sistema.

Ela efetua a conversão de caracteres quando acionada pelo "carãter modificador" ou a superimpresão de caracteres quando acionada pelo "carãter simulador".

O "carãter branco especial" sempre que encontrado é convertido no carãter hexadecimal...X'FF'.

4.7. RECUPERAÇÃO DE ERROS

A recuperação de erros é capítulo obrigatório em qualquer sistema de processamento de informações.

Em sistemas no qual o número de dados a processar é muito grande, caso de edição de texto, torna-se capítulo muito importante.

Os erros que podem ocorrer são decorrentes de três tipos de falhas:

- a) humanas
- b) de equipamento
- c) externas.

Nas seções seguintes serão estudados esses tipos e recomendados métodos de recuperação.

4.7.1. FALHAS HUMANAS

As falhas humanas são as mais frequentes e as de mais fácil detecção e recuperação.

Os erros daí decorrentes podem ser detetados antes ou depois da emissão dos comandos.

Quando detetados antes da emissão do comando, podem ser cancelados pela digitação do caráter ? (ponto de interrogação).

Quando detetados após, a detecção pode ser feita pelo sistema ou pelo usuário.

Em ambos os casos o usuário deverá emitir novo comando, corrigindo o erro cometido, normalmente devido a emissão de operando incorreto.

4.7.2. FALHAS DE EQUIPAMENTO

Os equipamentos podem ser divididos em lentos e velozes e a ação de recuperação será diferente num e noutro tipo.

4.7.2.1. EQUIPAMENTOS LENTOS

Situam-se nesta categoria os seguintes:

- a) leitoras e perfuradoras de cartões
- b) leitoras de fita de papel perfurada
- c) impressoras de linhas
- d) terminais de qualquer espécie.

A recuperação devida a falhas nestes dispositivos será feita em função do estado do sistema.

- a) se em estado de entrada ou alteração: verificar a última operação sucedida e, a partir desta, reiniciar.

- b) se em estado de edição: re-iniciar a partir da última página editada.

4.7.2.2. EQUIPAMENTOS VELOZES

Situam-se nesta categoria os seguintes dispositivos:

- a) unidades de fitas magnéticas
- b) unidades de discos magnéticos
- c) unidades de tambores magnéticos
- d) célula de dados.

A recuperação devida a falhas nestes dispositivos faz-se da seguinte maneira:

- a) se em estado de formatação: reiniciá-la
- b) se a falha ocorrer em dispositivo que contenha o arquivo fonte ou o arquivo de segurança: recuperar o arquivo perdido pela utilização do comando "reproduzir"; verificar se houve perda de informação; reentrar a informação perdida e após reiniciar.

4.7.3. FALHA EXTERNA

A falha externa é aquela decorrente da falta de energia ao sistema de processamento de informações.

Nestes casos o programa é descontinuado e deve ser reinicializado.

Procede-se após a reinicialização como se houvesse ocorrido uma falha humana ou de equipamento.

CAPÍTULO 5

CONCLUSÃO

Para a elaboração do presente trabalho foi consultada a mais extensa bibliografia que se pode obter.

Obteve-se literatura suficiente para estudar e pesquisar Editores de Programa e Editores de Texto Livre.

O objetivo era mais especificamente, o estudo de Editores de Texto Livre, já que os de Programa eram limitados por uma filosofia conceitual representada pela "linha de programa".

Dos Editores de Texto Livre a bibliografia obtida permitiu uma análise mais profunda de quatro sistemas, Editpak, Fress, Script e Edit/350 e de sua análise resultou o Quadro Comparativo de fls. 98.

Pelo exame deste pode-se, de imediato, dizer que o sistema FRESS (File Retrieval and Editing System) é o melhor dos quatro estudados.

Das cinco premissas propostas, este sistema atende três delas com total eficiência.

Não tem a característica de "Flexibilidade", isto é, não está capacitado a editar programas.

Finalmente, a característica de "Segurança" não é automática, deve ser providenciada pelo Usuário.

O sistema apresentado nos termos gerais propostos no Capítulo II, atende todas as premissas estabelecidas.

Preenche, portanto, lacunas deixadas pelos outros sistemas e introduz algumas facilidades nelas não propostas, sem deixar de lado a simplicidade da operação.

As facilidades introduzidas, isto é, não constantes em outros sistemas são:

- a) edição de textos com acentos e cedilha,
- b) edição de tabela de referências cruzadas.

A primeira delas muito importante se considerarmos a edição em língua portuguesa, hoje inviável com qualquer outro sistema editor.

O sistema EDITEXT implantado pode rodar em uma região de 80 K em MVT ou MFT sem "overlay"; com estrutura de "overlay" exigirá uma partição de 32 K.

Foram efetuados testes para determinar o tempo médio de formatação e edição de uma página.

A página formatada e editada continha 53 linhas de 65 caracteres.

O tempo médio obtido foi de 2 segundos de CPU, num computador IBM/370 - M155, para formatar e editar um texto com três páginas, contando todos os caracteres de controle propostos.

Na elaboração do projeto do sistema foi estabelecido um compromisso de dar ênfase a simplicidade de operação em pre-

juízo, evidente, da criação de recursos sofisticados.

Como está ele proposto, atende a mais de 90% das necessidades de edição, os demais 10% podem, também, ser atendidos pela utilização dos controles existentes de uma forma mais complexa.

Evidentemente, a sofisticação demandaria maiores recursos de memória com discutível ganho em velocidade de codificação do texto fonte pela baixa demanda que estas novas características teriam.

Seria possível, todavia, implementar novas características tais como:

- a) Possibilidade de introdução de rótulos ('labels') no texto fonte para facilitar percorrer mais rapidamente aquele quando na procura de uma cadeia de reconhecimento; esta característica é válida num texto que devesse sofrer muita modificação.
- b) Criação de um controle de edição com função de Gerador de Tabelas; característica de uso bastante específico.
- c) Criação de um controle de tipo "function" que contivesse um conjunto de controles de edição que seriam introduzidos no texto fonte no momento da formatação.
- d) Permitir a silabação, isto é, a divisão em sílabas, das palavras evitando, com isto, uma distribuição muito grande de brancos entre as palavras de uma linha e compactando mais o texto editado.
- e) Outras do tipo:

numeração romana das páginas

- . datador de página
- . títulos flutuantes
- . diversas justificações de linhas, etc..

Como vemos, o trabalho apresentado não é conclusivo, é um passo dado no sentido da obtenção de um bom sistema editor de textos.

Uma sequência natural do trabalho seria a implementação das características anteriormente citadas.

Outra e talvez a mais importante seria a de criar um ambiente de tempo-partilhado ('time-sharing') para uso simultâneo do sistema, através de uma rede de teleprocessamento, por vários terminais.

QUADRO COMPARATIVO DOS SISTEMAS

EDITORES DE TEXTO LIVRE

SISTEMAS		EDITPAK	SCRIPT	FRESS	EDIT/360	EDITEXT	
							PREMISSAS
INDEPENDÊNCIA DO USUÁRIO		S	N	S	S	S	
UNIVERSALIDADE		N	N	S	N	S	
FLEXIBILIDADE		N	N	N	N	S	
OPERAÇÃO	SIMPLICIDADE	M	M	S	N	S	
	COMANDOS	NÚMERO	I	C	C	C	C
		EFICIÊNCIA	M	M	T	T	T
	CONTROLES	NÚMERO	I	C	C	C	C
		EFICIÊNCIA	M	T	T	T	T
SEGURANÇA		N	U	U	U	A	

S - sim

M - média

C - completo

A - automática

N - não

I - incompleto

T - total

U - usuário

ANEXO I

BIBLIOGRAFIA

- 1) VAN DAM, Andries & RICE, David E. - On-line text editing: a survey. Computing Surveys, 3(3):93-114, sept. 1971.
- 2) WALTER, Gerard O. - Typesetting. Scientific American, 220(5):61-9, may 1969.
- 3) DEUTSCH, L. Peter & LAMPSON, Butter W. - An online editor. Comm. of the ACM, 10(12):793-803, dec. 1967.
- 4) LESSING, Lawrence - The printed word goes electronic. Fortune, 83(9):116-19, sept. 1969.
- 5) FARBER, D. J., GRISWOLD, R. E. , POLONSKY, I. P. - The SNOBOL 3 programming language. Bell system technical journal - july/aug. 1966.
- 6) VAN DAM, Andries & RICE, David E.. Computers and publishing: writing, editing and printing. In: Advances in Computers. New York, Academic Press. 1970 p.145-74.
- 7) ELLIOT, W. D., POTAS, W. A., VAN DAM, A. - Computer assisted tracing of text evolution. In: Fall Joint Computer Conference, 1971 v.37 p.533-40.
- 8) LAMPSON, B. W. & LICHTENBERGER, W. W. - A user machine in a time-sharing system. In: Proc. of IEEE, 1966 v.54 n.12 p.1766-79.
- 9) KNOTT, G. O., ANDERSON, W. J., MCKAY, J. A. - A table making language. In: Proc. of IEEE, 1966 v.54, n.12 p.1779-86.

- 10) KHUTH, Donald E. - Fundamental algorithms. Reading Mass, Addison-Wesley, 1969.
- 11) APL plus program library users guide. Washington, Scientific time-sharing, 1969. 10p. STSC publ. library-10.
- 12) MVT/WILBUR Manual. Stanford, Stanford University. Computation Center, 1971. 158p.
- 13) WILBUR reference manual 4th. ed. Stanford, Stanford University, Computation Center, 1971. 76p.
- 14) SYSTEM/360 text processor, EDIT/360, program description manual. New York, IBM, 1969. 312p. SH20-0712
- 15) FRESS, concepts and facilities for the layman. Barrington R.I., Text Systems 23p.
- 16) FRESS, user's guide. Barrington, R.I. Text Systems, 1972. 85p.
- 17) TECO, text editor and corrector, reference manual. Braintree, Mass, Interactive Sciences Corporation, 1971. 30p.
- 18) EDIT 10 reference manual. Braintree, Mass, Interactive Sciences Corporation, 1971. 60p.
- 19) SCRIPT Braintree, Mass, Interactive Corporation, CSdJ 36p.

ANEXO II

ALGORITMOS

Neste apêndice serão apresentados alguns algoritmos utilizados pelo sistema.

Estes algoritmos serão descritos na forma proposta por KNUTH em "Fundamental Algorithms".

Os algoritmos que serão descritos são os que executam as seguintes funções:

- a) procuram registros disponíveis;
- b) inserem registros na lista;
- c) retiram registros da lista, e
- d) justificam as linhas.

II.1. PROCURAR REGISTRO DISPONÍVEL

Este algoritmo pesquisa o arquivo fonte em busca de um registro disponível.

Um registro disponível foi definido como sendo aquele que tem o 'campo de caracteres contidos' com o valor igual a zero.

Sejam: NREG o apontador do registro corrente
QREG o apontador do registro procurado

NCH o número de caracteres contido no registro

ERR a chave de erro; \emptyset = não há erro.

P1 inicializa Fazer $QREG = NREG + 1$ e $ERR = \emptyset$.

P2 arquivo cheio ? Se $QREG = NREG$ então o arquivo está com todos os registros ativos, emite mensagem de "arquivo cheio", faz $ERR = 1$ e fim.

P3 $NCH = \emptyset$? Lê o registro apontado por $QREG$, se $NCH \neq \emptyset$ então faça $QREG = QREG + 1$ e vá para P2.

P4 registro encontrado Caso contrário $QREG$ é o apontador do primeiro registro disponível encontrado e fim.

Assim como foi proposto o algoritmo efetua a busca a partir do registro correntemente apontado.

Outra forma de propor o algoritmo seria inicializar $QREG$ fazendo-o igual a 1 (um), isto é, iniciando a busca a partir do primeiro registro do arquivo; o teste de fim de arquivo seria indagar se $QREG$ é maior que 'número total de registros do arquivo'.

O primeiro método tem como principal vantagem maior rapidez na busca e como desvantagem a distribuição, ao longo de todo o arquivo, dos registros.

O segundo método tem como vantagem a compactação do arquivo, situando todos os registros nas posições baixas daquele; tem como desvantagem a lentidão, é menos rápido.

O método adotado no sistema foi o primeiro.

II.2. INSERÇÃO DE REGISTRO

Este algoritmo permite a inserção, na lista de registros ativos, de um registro que assim foi tornado e dos quais se conhecem os apontadores do registro a inserir e o do que o antecederá.

Graficamente, a inserção pode ser representada como na figura II.1, em que as linhas cheias indicam apontadores atuais e as tracejadas os apontadores após a inserção.

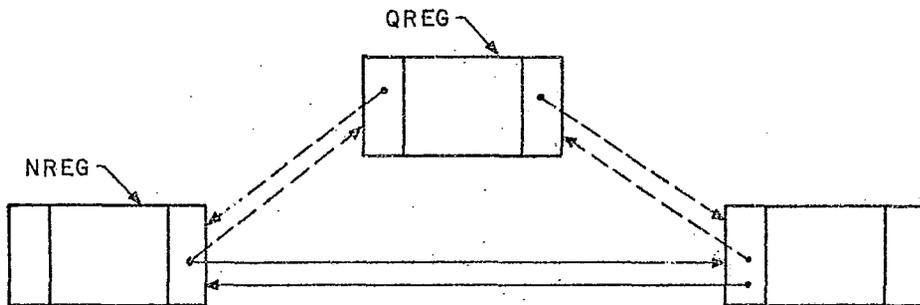


Fig: II - 1

O algoritmo descritivo é o seguinte:

Sejam: NREG o apontador do registro após o qual será feita a inserção;

QREG o apontador do registro a inserir;

NFR o apontador do registro sucessor;
 NBR o apontador do registro antecessor, e
 MFR e MBR, apontadores auxiliares.

I1 registro atual Lê registro apontado por NREG, faz
 MFR = NFR, faz NFR = QREG, grava registro apontado
 por NREG.

I2 registro inserido Faz NBR = NREG, faz NREG = QREG,
 faz QREG = MFR, faz NFR = MFR, grava registro aponta
 do por NREG.

I3 registro sucessor Faz MBR = NREG, faz NREG = QREG,
 lê registro apontado por NREG, faz NBR = MBR, grava
 registro apontado por NREG, faz NREG = MBR e fim.

O ponteiro NREG que, permanentemente, acompanha a lis-
 ta, fica apontando para o registro inserido.

II.3. RETIRA REGISTRO

Este algoritmo permite que um registro qualquer, do
 qual se conhece o apontador, seja retirado da lista e
 tornado inativo.

A representação gráfica do processo é a mesma da figu-
 ra II.1 em que as linhas tracejadas indicam os pontei-
 ros atuais e as cheias os ponteiros após a retirada.

Na figura, o registro a retirar é o apontado por QREG.

Quando se apaga um registro é necessário fazer-se com
 que o campo 'número de caracteres contidos' seja igual
 a zero; se assim não fosse, o algoritmo que "procura"
 registros não teria condições de reconhecer um regis-
 tro inativo.

A existência do campo 'ponteiro do registro antecessor' é justificada por este algoritmo.

Caso tal campo não existisse, para retirar um registro da lista, seria necessário percorrê-la a cada ocorrência e a eficiência do sistema seria prejudicada.

Sejam: NREG o apontador do registro a retirar;
 NFR o apontador do registro sucessor;
 NBR o apontador do registro antecessor;
 NCH o número de caracteres contidos, e
 MFR e MBR apontadores auxiliares.

R1 registro a retirar Lê registro apontado por NREG, faz MFR = NFR, faz MBR = NBR, faz NCH = \emptyset e grava o registro apontado por NREG.

R2 registro antecessor Faz NREG = MBR, lê registro apontado por NREG, faz NFR = MFR, regrava o registro.

R3 registro sucessor Faz NREG = MFR, lê registro apontado por NREG, faz NBR = MBR, regrava o registro e fim.

O ponteiro NREG que acompanha a lista fica apontando para o registro sucessor do registro retirado.

II.4. JUSTIFICA LINHA

A justificação de linhas é fazer com que elas iniciem na margem esquerda ou coluna onde inicia um parágrafo e concluam na margem direita (a primeira linha deste parágrafo é uma linha justificada).

Duas são as maneiras de justificar uma linha:

- a) deslocando a primeira palavra para a margem esquerda e a última para a margem direita, e
- b) distribuindo os caracteres "branco" em excesso, pelos caracteres brancos existentes entre as palavras da linha.

Em ambos os casos, por não haver silabação, as palavras incompletas ao fim da linha são armazenadas em área intermediária e jogadas para a linha seguinte.

A segunda forma proporciona uma distribuição mais harmônica do que a primeira.

A descrição deste algoritmo é muito extensa e muito complexa; daremos apenas o método utilizado para determinar a posição de inserção dos "brancos" em excesso.

Sejam: $NBD = \text{número de "brancos" a distribuir} + 1$

$NPL = \text{número de brancos entre palavras.}$

A representação gráfica da determinação das posições de distribuição é a seguinte:

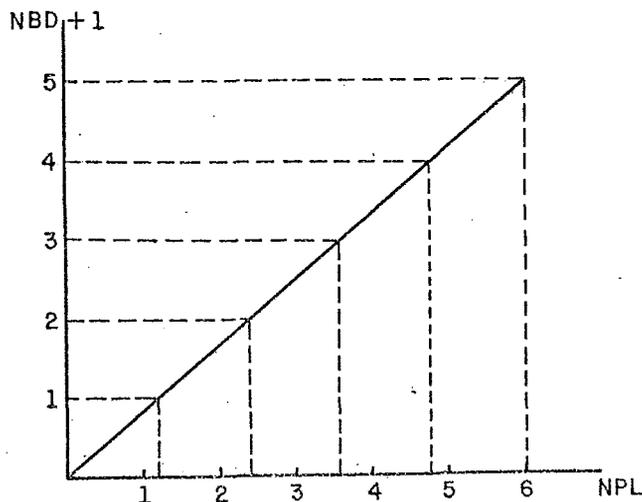


Fig: II - 2

No exemplo da figura II.2 temos 4 (quatro) brancos a distribuir entre 6 (seis) brancos entre palavras.

A distribuição mais harmônica é a de se inserir mais um branco aos brancos 2, 3, 4 e 5 existentes; a distribuição foi uniforme.

Matematicamente, esta distribuição, pode ser representada da seguinte forma:

Sendo POS_i a posição em que o branco i em excesso será inserido, temos:

$$POS_i = i \times NBL \div NBD$$

Se o número de brancos a distribuir for superior ao número de brancos entre palavras da linha, ocorrerá a distribuição de mais de um branco entre duas palavras; ainda assim a distribuição será uniformemente distribuída ao longo da linha.

ANEXO III

MANUAL DO USUÁRIO

O objetivo do presente é o de dar ao usuário, condições para que possa utilizar o sistema EDITEXT com segurança e eficiência.

A finalidade do sistema é permitir a edição de textos livres ou de programa, bem como permitir sua atualização de maneira fácil e simples.

Para isto, duas ferramentas de trabalho estão disponíveis: os comandos e os controles de edição.

Os comandos permitem que o usuário possa:

- a) criar o arquivo de texto fonte;
- b) mantê-lo atualizado;
- c) formatar o texto, e
- d) editá-lo.

Os controles de edição são auxiliares da formatação do texto, vindo nele inseridos e permitindo:

- a) dar títulos ao texto;
- b) criar tabelas;
- c) iniciar novo parágrafo ou página;
- d) modificar o formato da página;

- e) editar notas ao pé da página;
- f) duplicar cadeias de caracteres;
- g) numerar páginas, etc.

A edição permitida pelo sistema é de uma só coluna de n caracteres, com cabeçalho composto de título e subtítulo.

A numeração das páginas é feita, em algarismos arábicos, no rodapé e será centrada.

O formato normal de uma página editada, de texto livre, será a da figura III.1.

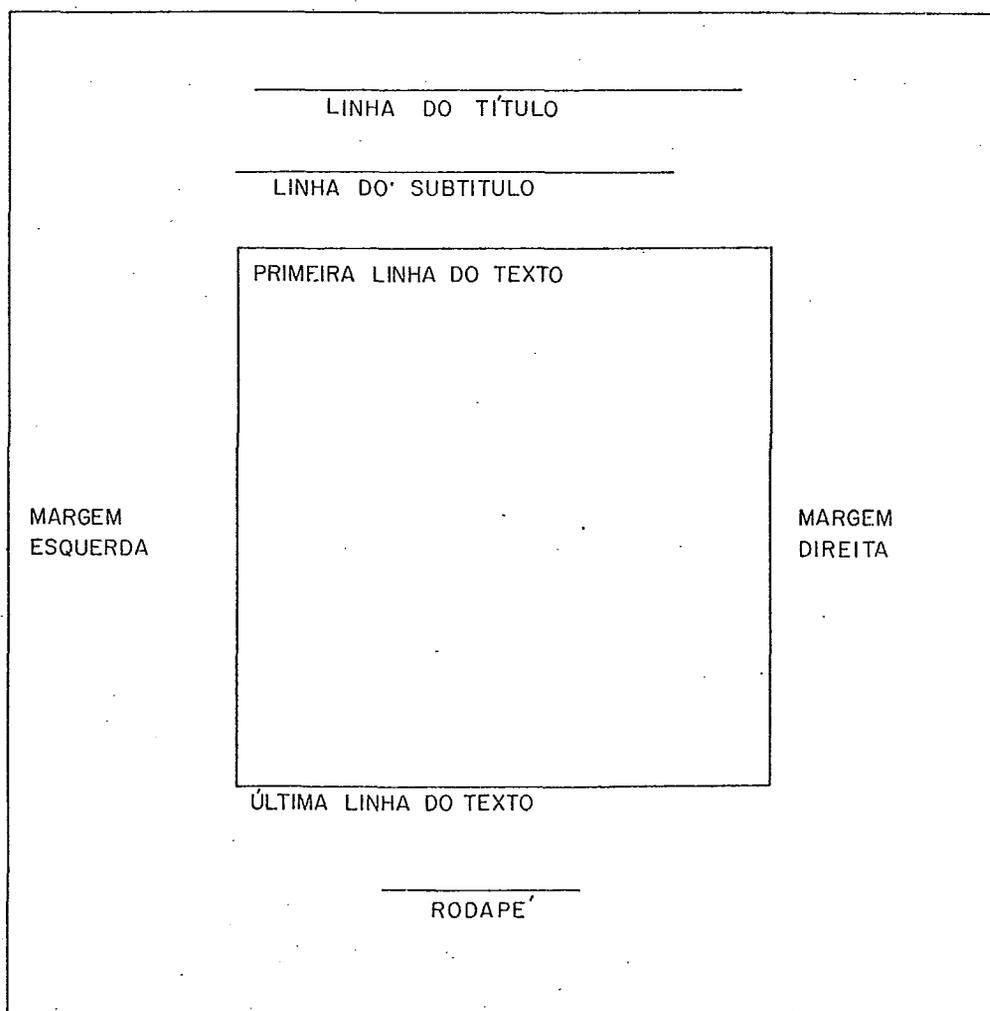


FIGURA III - 1

Na edição de programas, o formato da página editada tem, em relação à página da figura III.1, as seguintes modificações:

- a) há somente título e
- b) a numeração é feita na linha do título.

III.1. EDITEXT - VERSÃO 1

O presente manual instrui como proceder para a utilização da Versão 1 do sistema EDITEXT.

Em função do sistema de processamento de dados disponíveis, a versão do sistema que foi implantada não é completa de acordo com sua definição, estando com as seguintes restrições:

III.1.1. ENTRADA E SAÍDA

ENTRADA DE TEXTO FONTE

- a) somente em cartão perfurado.

SAÍDA DE TEXTO FONTE

- a) somente em impressora de linhas.

SAÍDA DE TEXTO EDITADO

- a) somente em impressora de linhas;

III.1.2. CONJUNTO DE CARACTERES

(LETRAS)::=A|B|C|D|.....|X|Y|Z

(NÚMEROS)::=0|1|2|3|4|5|6|7|8|9

(ESPECIAIS)::=,|.|=|+|-|/|\$|&|()|'|!*

Em consequência do conjunto de caracteres disponíveis, as funções de 'acentuação', 'sublinhação' e 'tornar ma*í*úsculas' não poderão ser efetuadas, serão simuladas.

A simulação das funções acima descritas será tratada no capítulo referente a 'controles de edição'.

As funções exercidas pelo 'terminal de trabalho' terão que ser simuladas pelo conjunto 'leitora de cartões/impressora de linhas'.

O processamento não é mais interativo, passa a ser feito em modo 'batch'.

III.2. INÍCIO E FIM DE SESSÃO

III.2.1. INÍCIO DE SESSÃO

O sistema EDITEXT é invocado pelo seguinte cartão de controle do Job Control Language:

```
// EXEC PROC=EDITEXT1
```

Ativado, ocorre a seguinte interação (as mensagens do sistema são impressas em letras maiúsculas precedidas de S. e as do usuário, em minúsculas precedidas de U.).

S. ** SESSÃO DE EDITEXT EM DD/MM/AA

S. ** HORA DE INÍCIO HH/MM/SS

S. * TEXTO LIVRE (L) OU PROGRAMA (P)

livre

U.

programa

Neste momento o programa ativa a rotina de controle (analisador sintático) que solicita um comando.

S. * COMANDO

A partir deste momento o usuário começa a fornecer os comandos e texto que necessita para as funções que deseja executar.

III.2.2. FIM DE SESSÃO

A sessão pode ser encerrada sempre que o sistema solicitar um comando; isto é feito fornecendo-se o comando.

U. fim

e o Sistema encerrarã, enviando as seguintes mensagens:

S.** SESSÃO DE EDITEXT EM DD/MM/AA

S.** HORA DE INÍCIO HH/MM/SS

S.** HORA DE FIM HH/MM/SS

III.3. EDIÇÃO DE TEXTO LIVRE

Se o usuário ao ser indagado pelo sistema que modo de edição desejava, respondeu 'livre', as facilidades para este tipo de edição são objeto do estudo das seções seguintes.

III.3.1. COMANDOS

Os comandos disponíveis para a edição de texto livre, tem o formato geral

(mnemônico) (oper1), (oper2),..., (oper_n) (comentários)

em que as seguintes regras devem ser obedecidas em sua emissão:

- a) o (mnemônico) pode ter mais caracteres do que os sublinhados (na descrição), sem limite e não permitido o \emptyset ;
- b) entre o (mnemônico) e o primeiro (operando) pode haver um número ilimitado de \emptyset ;
- c) os (operandos) são separados entre si por ',' podendo, após uma delas, haver um ou mais \emptyset ;
- d) o \emptyset após um (operando) indica o fim da lista de operandos, e
- e) o (comentário), que é ignorado pelo sistema e serve somente para documentar a sessão, não pode ultrapassar o fim do cartão que contem o último (operando).

Em modo de edição de textos livres os delimitadores da ação dos comandos são as 'cadeias de reconhecimento'.

As cadeias de reconhecimento são cadeias de caracteres delimitadas ao início e fim pelo caráter / (barra), contendo um máximo de 80 (oitenta) caracteres, não podendo ser vazias.

Se o caráter / fizer parte da cadeia, este para evitar ambiguidade, deve aparecer em duplicata (exemplo: a cadeia 250/20 deve ser fornecida ao sistema /250//20/ caso contrário a segunda barra seria entendida como o segundo delimitador da cadeia).

COMANDOS DE ENTRADA E SAÍDA

começar

Este comando inicializa o arquivo fonte apagando todo o seu conteúdo, sendo utilizado para a criação de novo texto fonte.

O texto que pode ser entrada é de tamanho ilimitado e seu fim é indicado pelos caracteres && que encerram o 'stream'.

A entrada do texto fonte é feita através de cartões perfurados; é possível a utilização parcial de um cartão, indicando o fim do 'stream' naquele cartão pela perfuração de caráter "record mark" (#, perfurações 0-2-8).

exibir {
cadeia1, cadeia2 }

No formato 1, todo o arquivo fonte é impresso.

No formato 2, o texto fonte compreendido entre cadeia1 e cadeia2, inclusive, é impresso.

Se cadeia é omitida, o sistema não sabe onde deve iniciar; se cadeia2 é omitida, o sistema imprime apenas a cadeia1.

A largura da impressão é de 80 (oitenta) caracteres por linha, dando uma imagem de cartão.

COMANDOS DE ALTERAÇÃO

apagar { cadeia1, cadeia2 }

O texto compreendido entre cadeia1 e cadeia2 é apagado do arquivo de texto fonte.

Se cadeia1 é omitida, o sistema não sabe onde começar; se cadeia2 é omitida, o sistema apaga apenas a cadeia1.

copiar { cadeia1, cadeia2, cadeia3 }

O texto compreendido entre cadeia2 e cadeia3, inclusive, é copiado após a cadeia1.

Se cadeia1 é omitida, o sistema não sabe onde copiar; se cadeia2 é omitida, o sistema não sabe onde inicia o texto a copiar; se cadeia3 é omitida, apenas a cadeia2 é copiada.

inserir { cadeia1 }

No formato 1 a inserção é feita após o último caráter do texto existente.

No formato 2 a inserção é feita após o último caráter da cadeia de reconhecimento fornecida.

O texto a inserir pode ter tamanho ilimitado, sendo delimitado pelos caracte-

res &&.

mover { cadeia1, cadeia2, cadeia3 }

O texto compreendido entre a cadeia2 e a cadeia3 é copiado após a cadeia1; este texto, em sua posição original, é apagado.

Se cadeia1 é omitida, o sistema não sabe para onde mover; se cadeia2 é omitida, o sistema não sabe onde o texto mover inicia; se cadeia3 é omitida, apenas a cadeia2 é movida.

substituir { cadeia1, cadeia2 }

O texto compreendido entre cadeia1 e cadeia2 é apagado e substituído por um texto, de tamanho ilimitado e delimitado por &&, dado.

Se cadeia1 é omitida, o sistema não sabe onde iniciar a substituição; se cadeia2 é omitida, apenas a cadeia1 é substituída.

COMANDOS DE FORMATAÇÃO

paginar { linha do título, linha do subtítulo, primeira linha, última linha, linha do rodapé, margem esquerda, margem direita. }

Este comando define para o comando 'formatar' a forma da página que será editada.

Qualquer um dos operandos pode ser suprimido; neste caso o operando suprimido fica com o padrão do sistema, que é:

linha do título	=	linha 8
linha do subtítulo	=	linha 10
primeira linha	=	linha 12
última linha	=	linha 61
linha do rodapé	=	linha 63
margem esquerda	=	coluna 13
margem direita	=	coluna 77

A supressão do operando não suprime a vírgula que lhe corresponde, os operandos são posicionais. A excessão se dá quando é suprimido o último operando.

O comando deve ter, no mínimo, um operando. A margem direita não pode ultrapassar a coluna 120.

formatar { cadeia1, cadeia2 }

O texto compreendido entre a cadeia1 e a cadeia2 é formatado e suas linhas são numeradas a partir de um número dado; no formato 1, todo o arquivo fonte é formatado.

Se cadeia1 é omitida, o sistema não sabe onde iniciar a formatação; se cadeia2 é omitida, apenas a cadeia1 é formatada.

O número da primeira página deve ser dado através do caráter de controle "a".

COMANDOS DE EDIÇÃO

editar {
página1,página2 }

No formato 1, todo o arquivo formatado é editado; no 2, é feita a edição da página1 à página2.

Se página1 é omitida, o sistema não sabe onde iniciar a edição; se página2 é omitida, é editada somente a página1.

COMANDOS DE SERVIÇO

compactar

Após um número elevado de modificações no texto fonte, dois fatos podem ocorrer:

- a) o sistema sente que precisa reorganizar o arquivo fonte e invoca a sua compactação, ou
- b) o usuário deteta a queda da eficiência do sistema e ele próprio solicita esta compactação.

A compactação elimina os espaços desnecessários e reorganiza o arquivo.

É aconselhável a reorganização do arquivo antes da formatação para tornar esta função mais eficiente.

localizar { cadeial }

Este comando localiza e imprime:

- a) no formato 1, os últimos caracteres do arquivo fonte; esta forma é utilizada para reinício de sessão ou, após a recuperação de falhas externas ao sistema, saber se houve perda de informação;

b) no formato 2, o texto que inicia pela cadeia la. O número de caracteres impresso por este comando é definido pelo comando 'quadrar' ou o padrão do sistema.

quadrar { linhas }

Este comando define um quadro impresso de m linhas de 80 (oitenta) caracteres que será utilizado pelo comando .. 'localizar'.

O padrão do sistema é 12 linhas de 80 caracteres.

reproduzir { f }
 { s }

Este comando recupera, quando da ocorrência de falhas externas ao sistema, um dos arquivos de texto fonte.

A opção 'f' recupera o arquivo fonte propriamente dito e a opção 's' o de segurança daquele.

III.3.2. CONTROLES DE EDIÇÃO

O texto fonte é levado a seu arquivo sem formato.

A edição exige uma formatação do texto, o que é executado pelo comando 'formatar'.

Este comando necessita de informação para que possa executar sua função; esta informação é dada pelos 'controles de edição'.

Estes, indicarão ao formatador, quando iniciar nova linha ou página, criar tabelas, etc

É necessário que esta informação esteja contida no texto fonte e que o formatador ao manipulá-lo possa distinguir entre caracteres de texto fonte e controles de edição.

Os controles de edição são de dois tipos: os controles propriamente ditos e os caracteres especiais.

Os controles propriamente ditos, têm o seguinte formato geral:

```
$(mnemônico)(operando1),...(operandoN)$
```

sendo o (mnemônico) composto de uma ou duas letras e os (operandos) numéricos, alfabéticos ou alfanuméricos, separados, entre si, por vírgulas; o caráter \$ delimita o controle.

Os caracteres especiais tem o formato geral:

(carãter especial)(operando)

em que o (operando) pode ser um carãter qualquer.

O texto a seguir é demonstrativo de como fica este com os controles de edição embebidos; estes estarão sublinhados para destaque.

\$CUMA HISTORIA QUE JÃ FOI ESQUECIDA\$\$\$TRA-
BALHADOR E DISCIPLINADOR, MUITOS ATÉ PREFERI
AM DEFINI-LO COMO UM HOMEM DURO.\$P\$ZM, AO
LONGO DOS SEUS 28 ANOS DE CARREIRA FOI O TÉC
NICO QUE MAIS INOVOU NO FUTEBOL BRASILEIRO.
\$S3\$A MARCAÇÃO POR ZONA, O ZAGUEIRO LIBERO,
O PONTEIRO RECUADO FAZE

CONTROLES

\$a(novo número)\$

Este controle altera o número de página que está sendo formatada.

Se (novo número) = Ø a página corrente não será numerada, o contador de páginas é incrementado de 1 (um).

O (novo número) deve ser igual ou menor que 9999.

Este controle é usado para indicar o número da página inicial a ser formatada.

\$b(número),(texto)\$

Este controle permite a criação de um bloco com (número) linhas em branco, contidas numa mesma página e permite, também, dar (texto) ao bloco.

O (número) de linhas dado deve ser igual ao número necessário para a figura a ser incluída e mais duas; uma para o texto e outra para separá-lo do restante da edição.

O (texto) do bloco é justificado pela esquerda e não pode conter mais caracteres do que uma linha.

Se (número) igual a Ø, assume-se (número) = 10.

Se o bloco não couber na página corrente é jogado para a página seguinte.

A linha anterior é formatada.

\$c(texto)\$

O (texto), igual ou menor que a largura da edição, e centrado na linha.

A linha anterior é formatada, bem como a linha a centrar.

\$d(número),(texto)\$

A linha anterior é formatada e o (texto) é duplicado o (número) de vezes dado.

Se (número) = 0 então o (texto) é duplicado uma vez.

Se (texto) é vazio, são formatadas um (número) de linhas contendo o caráter "." (ponto).

\$f(texto)\$

O (texto), igual ou menor que a linha editada, é inserido ao pé da página.

Entre o texto que está sendo editado e as notas ao pé é formatada uma linha de separação.

Esta linha contém dez caracteres '.' (ponto).

O número de notas ao pé da página é restrito ao espaço disponível.

O texto é justificado pela margem esquerda e não pode ser vazio.

A linha anterior é formatada.

$\$h(\text{tipo}),(\text{texto})\$$

Este controle define o (texto) do título ou subtítulo da página de edição, de acordo com o (tipo) dado.

Se (tipo) = t, está sendo definido o título.

Se (tipo) = s, está sendo definido o subtítulo.

Tanto o título quanto o subtítulo são justificados pela margem esquerda.

Se (texto) é vazio, o (tipo) indicado é suprimido na página seguinte.

O (texto) deve ser igual ou menor que a linha de edição.

$\$j(\text{coluna}),(\text{texto})$

O (texto) é justificado à esquerda na (coluna) indicada.

Se (coluna) = \emptyset , o texto é justificado pela margem esquerda.

O (texto) justificado não pode exceder as margens.

Este controle não formata a linha anterior e nem a que contem os caracteres justificados; deve ser usada em conjunto com um que provoque esta ação.

As letras contidas no (texto) são convertidas em maiúsculas.

Esta conversão é feita por simulação; a própria letra é superimpressa na edição.

\$n

Este controle força nova página, efetuando as seguintes operações: formata a linha anterior, formata as notas ao pé da página, incrementa contador de páginas, formata título, formata bloco (se houver) e posiciona ponteiro da primeira linha.

Este controle é uma exceção à regra, não tendo o seguinte delimitador "\$".

\$p(coluna)\$

A linha anterior é formatada e o ponteiro de início da próxima linha toma o valor da (coluna) dada.

Se (coluna) = Ø então o valor anterior é assumido; ou o padrão do sistema que é o mesmo da página esquerda ou o do último controle \$p\$.

O início de parágrafo não pode dar fora das margens.

`$r(margem esquerda),(margem direita)$`

Este controle permite alterar, uma ou outra ou ambas as margens e é efetivo após seu encontro.

Se ambos os valores forem zero, há retorno às definidas pelo comando 'paginar' ou aos padrões do sistema.

A linha anterior é formatada.

`$s(número)$`

A linha em montagem é formatada e o ponteiro da próxima linha é avançado no (número) dado.

Se (número) = 0 é assumido (número) = 1.

Se o ponteiro é forçado além da última linha disponível, nova página é forçada.

`$t(texto)$`

A linha em montagem é formatada e o (texto) é formatado sem justificação.

O fim do (texto) força a formatação da linha.

`$u(texto)$`

O (texto) dado é sublinhado; os caracteres normais não o são.

Esta operação pode ser simulada pela superimpressão de um caráter '/'.

$\$x(\text{texto})\$$

O (texto) é formatado e simultaneamente gravado num arquivo especial que permitirá a edição de uma tabela de referências cruzadas.

O (texto) poderá conter um máximo de 30 (trinta) caracteres.

$\$y(\text{nível}),(\text{texto})\$$

O (texto) é formatado e, juntamente com o (nível), gravado em arquivo especial que permitirá a edição de um índice de matérias.

O (nível) indica um deslocamento, de $((\text{nível}) * 2 - 2)$ caracteres, do (texto) em relação à margem esquerda.

O (texto) não poderá conter mais do que 40 (quarenta) caracteres.

CARACTERES ESPECIAIS

&

O 'caráter branco especial' é um caráter de enchimento de vazios e é utilizado para:

- a) impedir, quando é formatada uma linha justificada, que entre duas palavras sejam inseridos ¶ de justificação;
- b) permitir uma cadeia de brancos na linha;
- c) permitir que o ¶ seja sublinhado.

Este caráter é editado como um ¶ normal.

/

O 'caráter modificador' é utilizado para modificar o caráter que o segue, sendo utilizado para:

- a) representação dos próprios caracteres especiais (& / \$ *);
- b) tornar a letra maiúscula (a simulação é a mesma do controle m).

O 'carãter simulãdor' ã utilizado para si
mular os acentos (supermipressãõ do carã-
ter ') e criar o cedilha (superimpressãõ
da vírgula).

III.4. EDIÇÃO DE PROGRAMAS

Na edição de programas é abandonado o conceito de que o texto é uma "hipercadeia" de caracteres.

O conceito adotado é o de que o texto é composto de linhas de programa, com tamanho fixo, sem acentos e não sublinhadas.

Os procedimentos de início e fim de sessão do editor são os mesmos.

O conjunto de caracteres existente permite a edição de programas de linguagens em que 48 (quarenta e oito) caracteres são suficientes, tais como: FORTRAN, COBOL e PL/I.

III.4.1. COMANDOS

Os comandos utilizados para a edição de programas têm o mesmo formato daqueles utilizados para a edição de textos livres.

As regras que regem os comandos de edição de texto livre são integralmente aplicáveis na edição de programas.

Neste modo de edição, os delimitadores da cadeia não são mais as "cadeias de reconhecimento", mas "registros", correspondendo cada um a uma linha de programa.

A cadeia de reconhecimento só pode ser utilizada no comando "localizar", com o tamanho máximo de 80 (oitenta) caracteres.

Os "controles de edição" não são aplicáveis na edição de programa.

Os operandos que indicam início de ação são, como na edição de textos livres, obrigatórios; sua omissão causa erro.

Tudo o que foi dito a respeito dos comandos, em sua descrição na edição de texto livre, é válido para a edição de programas; somente serão feitos comentários sobre aqueles que neste modo de edição tenham características diferentes.

COMANDOS DE ENTRADA E SAÍDA

começar

exibir {registro1,registro2}

O formato de impressão é fixo e tem nas:

coluna 1 à coluna 80 - texto

coluna 83 à coluna 90 - número de registro

COMANDOS DE ALTERAÇÃO

apagar {registro1,registro2}

copiar {registro1,registro2,registro3}

inserir {registro1}

mover {registro1,registro2,registro3}

substituir {registro1,registro2}

COMANDOS DE FORMATAÇÃO

paginar { linha do título, primeira linha,
última linha }

Neste comando os únicos parâmetros que podem ser modificados são os acima.

A linha do subtítulo não tem sentido pois não há subtítulo, as demais tem valores fixos que não podem ser modificados.

Estes valores são:

linha do rodapé = linha 63

margem esquerda = coluna 1

margem direita = coluna 80

formatar

Este comando prepara o programa para ser analisado pelos "montadores" ou "compiladores".

COMANDO DE EDIÇÃO

editar

Este comando reorganiza o arquivo de texto fonte e edita integralmente o programa.

Antes da edição solicita ao usuário o título do programa com a mensagem:

* TITULO DO PROGRAMA (MÁXIMO 72 CARACTERES)

O usuário fornece o título do programa e o comando edita-o com o seguinte formato:

coluna 1 à coluna 72 - texto de programa;

coluna 73 à coluna 76 - primeiros quatro caracteres do título;

coluna 77 à coluna 80 - número da linha (ou registro).

Imprime em todas as páginas, na linha do título, este e o número da página.

COMANDOS DE SERVIÇO

compactar

<u>localizar</u>	{	registros	}
		cadeias	}
<u>quadrar</u>	{	linhas	}
<u>reproduzir</u>	{	f	}
		s	}

III.5. DETECÇÃO E RECUPERAÇÃO DE ERROS

A detecção de erros pode ser feita pelo sistema ou pelo usuário.

A ação corretiva será, sempre, feita pelo usuário, o único competente para julgar qual deva esta ser.

Os erros decorrem de:

- a) texto incorreto;
- b) comandos incorretos;
- c) controles de edição incorretos, e
- d) falhas externas ao sistema.

De acordo com o tipo de erro detetado, ações devem ser tomadas pelo sistema ou pelo usuário.

III.5.1. TEXTO INCORRETO

As incorreções no texto são detetados pelo usuário de posse de listagens do texto fonte e da edição.

A ação corretiva é feita pela utilização de comando conveniente que altere o texto e elimine o erro.

III.5.2. COMANDOS INCORRETOS

Os erros detetados na emissão de comandos podem ser detetados pelo sistema ou pelo usuário.

Os erros detetados pelo sistema são os de sintaxe ou semântica.

Os erros detetados pelo usuário são aqueles resultantes de fornecimento de informação incorreta nos operandos, como por exemplo, uma cadeia errada (com seus caracteres não representando o que o usuário deseja realmente).

Estes erros devem ser corrigidos pelo próprio usuário na forma descrita em III.5.1.

Os erros de sintaxe detetados pelo sistema são os que resultam em mensagens auto-explicativas do erro ocorrido.

Estes erros são:

- ** NUMERO DE REGISTRO MAIOR QUE O NUMERO TOTAL DE REGISTROS
- ** REGISTRO XXXX FORA DE USO
- ** CARATER ILEGAL EM OPERANDO NUMERICO
- ** OPCAO INVALIDA
- ** CADEIA VAZIA
- ** CADEIA NAO ENCONTRADA NO TEXTO
- ** CADEIA NAO INICIA COM
- ** CARATER INVALIDO APOS CADEIA

**** CADEIA MAIOR QUE 80 CARACTERES**

Os erros de semântica que ocorrem são resultantes de:

- a) a omissão do operando indicando onde deve iniciar a ação;
- b) a omissão do operando que indica o início do texto ou o primeiro registro a ser modificado, e
- c) a omissão dos operandos que indicam início e fim do texto ou primeiro e último registros a serem modificados.

Estes erros geram mensagem do tipo:

**** ONDE INICIA A AÇÃO**

Se o sistema estivesse sendo processado a-través de um terminal, o comando seria des-continuado e novo comando seria solicitado.

Em modo de 'batch' estes erros descontinuam o sistema pois de outro modo os cartões que seguissem o comando errado seriam entendidos como outros comandos e o resultado disto seria imprevisível.

III.5.3. CONTROLES INCORRETOS

Da mesma forma que nos comandos os erros resultantes de informação incorreta são dete-tados pelo usuário que obterá uma edição

deformada; a ação corretiva é a mesma sugerida em III.5.1.

III.5.4. FALHAS EXTERNAS AO SISTEMA

As falhas externas ao sistema que podem ocorrer são:

- a) falhas nos dispositivos de entrada e saída;
- b) falhas nos dispositivos de arquivamento, e
- c) falta de energia

FALHAS NOS DISPOSITIVOS DE ENTRADA E SAÍDA

A recuperação pode ser feita de uma das maneiras a seguir descritas:

- a) se no sistema de processamento de dados houver disponível outra unidade idêntica, tentar prosseguir através dela, verificando antes se não houve perda de informação; se houve adotar sistemática sugerida em III.5.1.
- b) não havendo outra unidade disponível, descontinuar o sistema, aguardar conserto da unidade, reiniciar verificando se houve perda de informação; se houve adotar sistemática sugerida em III.5.1.; agir da mesma maneira se não for possível prosseguir através de

outra unidade do sistema, sem aguardar a etapa de conserto.

FALHAS NOS DISPOSITIVOS DE ARQUIVAMENTO

A falha tendo ocorrido no dispositivo que contem o arquivo de 'texto formatado', reiniciar a formatação.

Se a falha ocorreu num dos dispositivos que contem o arquivo de 'texto fonte' ou de 'segurança', recuperar o arquivo afetado utilizando o comando 'reproduzir'.

FALTA DE ENERGIA

A falta de energia descontinua o sistema.

A recuperação é feita da seguinte forma:

- a) reinicializar o sistema;
- b) verificar se a falta de energia resultou em perda de informação; se houve perda, recuperar a informação e
- c) prosseguir normalmente.

ANEXO IV

GLOSSÁRIO DE TERMOS UTILIZADOS

ALGORITMO

Forma utilizada para descrever os passos para a resolução de um problema.

ALTURA

De uma página editada, é o número total de linhas de texto que nela podem ser contidas; não são computadas as linhas do cabeçalho (título e subtítulo) e do rodapé.

ARQUIVO

Um conjunto, ordenado ou não, de registros que contenham a mesma informação.

BACK-UP

Um procedimento utilizado para evitar a perda de informação por defeitos dos dispositivos de um computador.

BUFFER

Uma área na memória central de um computador utilizada para área de trabalho.

CADEIA DE RECONHECIMENTO

Um conjunto de caracteres utilizado para informar ao sistema o ponto inicial e/ou final de uma operação sobre o texto fonte.

CAMPO

Subdivisão de um registro; o conjunto dos campos constitui o registro.

CARÁTER

Um símbolo gráfico que representa uma letra, um número ou um símbolo especial.

CHAVE DE CONTROLE

Uma variável que pode assumir valores diversos e utilizada para alterar o fluxo de procedimentos de uma rotina.

COMANDO

Uma forma utilizada para determinar ao sistema a execução de uma operação.

CONTROLE DE EDIÇÃO

Conjunto de caracteres utilizado para informar o sistema como deve formatar o texto fonte.

DISPOSITIVO

Um equipamento físico de um computador, que utiliza um meio para armazenar informação.

EDITAR

Transferir um conjunto de linhas formatadas para um meio de saída.

EXIBIR

Imprimir, em todo ou em parte, o texto fonte num meio de saída.

FORMATAR

A nível de texto, transformar texto fonte em texto formatado; a nível de linha, arranjar-la de acordo com os controles de edição e gravá-la no arquivo de texto formatado.

JUSTIFICAR

Alinhar um texto por um ponto de referência, à esquerda ou à direita do ponto.

LARGURA

De uma página editada, é o número total de caracteres que pode conter uma linha, excluídas as margens.

MEIO

Um corpo físico destinado a armazenar informação, podendo ser: formulário contínuo, cartão, fita de papel, fita ou disco magnético.

MNEMÔNICO

Forma abreviada de uma palavra permitindo seu entendimento.

OPERANDO

Parâmetro de um controle de edição ou um comando.

OVERLAY

Método utilizado para subdividir programas de computador com a

finalidade de permitir menor emprego de memória.

PONTEIRO

Um valor que indica a posição de um caráter, uma linha ou um registro.

REGISTRO

A nível de arquivo, um conjunto ordenado e interrelacionado de informações; a nível de edição de programas, uma linha de programa.

ROTINA

Ou programa, é um conjunto de procedimentos, devidamente codificados, que o computador deve executar para a obtenção do resultado desejado.

TEXTOS

Um conjunto de caracteres cuja largura é determinada pelo número deles nele contido.

TEXTOS EDITADOS

O texto fonte já formatado, descarregado em um meio de saída (formulário contínuo, folhas isoladas ou cartão).

TEXTOS FONTE

O conjunto de caracteres que será formatado e posteriormente editado e no qual estão inseridos os controles de edição.

TEXTO FORMATADO

O conjunto de caracteres que compõem o arquivo fonte já arranjados de acordo com os controles de edição e em condições de ser editado.

ANEXO V

EXEMPLOS DE APLICAÇÃO

V.1 USO DOS COMANDOS

** SESSAO DE EDITEXT EM 04.10.73
 ** HORA DE INICIO 09.18.28

* TEXTO LIVRE(L) OU PROGRAMA(P)

TEXTO LIVRE

* COMANDO

COMEÇAR ENTRADA DE TEXTO

* AGUARDANDO TEXTO

A SOLUCAO PROPOSTA E MANTER O TEXTO FONTE ARQUIVADO EM DISPOSITIVO DE UM SISTEMA DE PROCESSAMENTO DE INFORMACOES.@@

* COMANDO

APAGAR /MANTER/,DISPOSITIVO DE /

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E UM SISTEMA DE PROCESSAMENTO DE INFORMACOES.

* COMANDO

SUBSTITUIR /PROCES/,/MACDES/

COMPLTACAO DE DADOS QUE PERMITA ACESSO DIRETO@@

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E UM SISTEMA DE COMPUTACAO DE DADOS QUE PERMITA ACESSO DIRETO.

* COMANDO

INSERIR /PROPOSTA E /

TER A INFORMACAO (TEXTO FONTE) ARQUIVADA EM DISPOSITIVO DE@@

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E TER A INFORMACAO (TEXTO FONTE) ARQUIVADA EM DISPOSITIVO DE UM SISTEMA DE COMPUTACAO DE DADOS QUE PERMITA ACESSO

O DIRETO.

* COMANDO

EXIBIR /INFORM/, /COMPU/

INFORMACAO (TEXTO FONTE) ARQUIVADA EM DISPOSITIVO DE UM SISTEMA D
E COMPU

* COMANDO

COPIAR /DISPOSITIVO DE /, /ACES/, /RETO/

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E TER A INFORMACAO (TEXTO FONTE) ARQUIVADA EM
DISPOSITIVO DE ACESSO DIRETO UM SISTEMA DE COMPUTACAO DE DADOS QUE
PERMITA ACESSO DIRETO.

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E TER A INFORMACAO ARQUIVADA EM DISPOSITIVO DE
ACCESSO DIRETO (TEXTO FONTE) UM SISTEMA DE COMPUTACAO DE DADOS QUE
PERMITA ACESSO DIRETO.

* COMANDO

QUADRAR 1

LOCALIZAR /PROPOSTA/

PROPOSTA E TER A INFORMACAO ARQUIVADA EM DISPOSITIVO DE ACESSO DI

* COMANDO

INSERIR.

A FUNCAO DE INSERIR AO FIM DO TEXTO E COMANDADA POR I. .aa

* COMANDO

EXIBIR.

A SOLUCAO PROPOSTA E TER A INFORMACAO ARQUIVADA EM DISPOSITIVO DE
ACCESSO DIRETO (TEXTO FONTE) UM SISTEMA DE COMPUTACAO DE DADOS QUE
PERMITA ACESSO DIRETO. A FUNCAO DE INSERIR AO FIM DO TEXTO E COM
ANDADA POR I. .

* COMANDO

FORMATAR.

* COMANDO

EDITAR.

** FIM DE EDICAO CU ARQUIVO

* COMANDO

REPRODUZIR F

* COMANDO

COMPACTAR

* COMANDO

FIM DE TESTE COM TEXTO LIVRE

** SESSAO DE EDITEXT EM 04.10.73

** HORA DE INICIO 09.18.28

** HORA DE TERMINO 09.18.48

V.2 USO DOS CONTROLES

Texto fonte que gera o texto editado das páginas seguintes.

À esquerda do texto, os ponteiros do registro seguinte, do anterior e o número de caracteres contidos (No registro 1, são: 0002, 0042 e 80).

```

00020042 80$HT SISTEMA EDITEXT;$HSEXEMPLOS DE USO DOS CONTROLES;$A;$N;$P$NUMERACAO DE PAGINA
00030001 80$P$/;$A(NOVO NUMERO)/;$P$/;O NUMERO DESTA PAGINA FOI GERADO PELO CONTROLE /;$A/$. SE
00040002 80 O NUMERO DESEJADO PARA ESTA PAGINA FOSSE = 291, O CONTROLE SERIA /;$A290/$.;$P$BL
00050003 80$C$O DE LINHAS CONTIGUAS EM BRANCO;$P$/;$B(NUMERO), (TEXTO)/;$P$/A SEGUIR SERA GERADO
00060004 80 UM BLOCO COM 10 LINHAS E COM TEXTO=TEXTO DE BLOCO DE 10 LINHAS, PELO CONTROLE /
00070005 80;$B10, TEXTO DE BLOCO DE 10 LINHAS/$.;$B10, TEXTO DE BLOCO DE 10 LINHAS$SE O BLOCO N
00080006 80$A$O COBER NESTA PAGINA SERA COLOCADO NA SEGUINTE.;$P$/CENTRAR TEXTO;$P$/;$C (TEXTO)/$
00090007 80;$P$/A LINHA SEGUINTE SERA CENTRADA NO TEXTO PELO CONTROLE /;$C$TEXTO CENTRADO/$.;$C$T
00100008 80$EXT$O CENTRADO;$P$/DUPLICAR TEXTO;$P$/;$D(NUMERO), (TEXTO)/;$P$/O CONTROLE /;$D/;$ GERAR
00110009 80$A$ UNA LINHA COMPLETA DE PONTOS;$C$O CONTROLE /;$D4, (ABCDEFGHIJGFEDCBA)/;$ REPETIRA 4 V
00120010 80$EZ$ES O TEXTO (ABCDEFGHIJGFEDCBA).;$D4, (ABCDEFGHIJGFEDCBA)/;$P$/NOTAS AO PE DA PAGINA;$P$/;$
00130011 80$F(TEXTO)/;$P$/A NOTA AO PE DA PAGINA COM TEXTO = NOTA DE PE DE PAGINA SERA GERADA
00140012 80 PELO CONTROLE /;$F$NOTA DE PE DE PAGINA/$.;$F$NOTA DE PE DE PAGINA;$P$/TITULO E SUBT
00150013 80$ITULO;$P$/;$H(TIPO)(TEXTO)/;$P$/O TITULO DESTA PAGINA FOI CRIADO COM O CONTROLE /;$H
00160014 80$T$SISTEMA EDITEXT/;$ E O SUBTITULO O FOI COM O CONTROLE /;$HSEXEMPLOS DE USO DOS CO
00170015 80$N$T$ROLES/$.;$P$/JUSTIFICAR;$P$/;$J(COLUNA), (TEXTO)/;$P$/O TEXTO /;$P$/TEXTO JUSTIFICADO NA
00180016 80 $COLUNA 20/;$ FOI POSICIONADO PELO CONTROLE /;$J20, TEXTO JUSTIFICADO NA COLUNA 20/
00190017 80;$ E IMPRESSO PELO CONTROLE /;$S/$.;$S;$J20, TEXTO JUSTIFICADO NA COLUNA 20;$S;$P$/TE
00200018 80$X$T$O EM MAIUSCULAS;$P$/;$M(TEXTO)/;$P$/O TEXTO /;$P$/TEXTO EM MAIUSCULAS/;$ SERA GERADO P
00210019 80$E$L$O CONTROLE /;$M$TEXTO EM MAIUSCULAS/$.;$S;$M$TEXTO EM MAIUSCULAS;$P$/NOVA PAGINA;$P$/
00220020 80/;$A;$P$/O CONTROLE /;$N FORÇA NOVA PAGINA IMPRIMINDO A ULTIMA LINHA EM FORMATAÇÃO,
00230021 80 $P$/CAPE, NOTAS AO PE E INICIA NOVA PAGINA.;$P$/NOVAS MARGENS;$P$/;$X(MARGEM ESQUERDA
00240022 80), (MARGEM DIREITA)/;$S2;$X20,50$TUDO O TEXTO QUE SERA IMPRESSO A SEGUIR SERA JUS
00250023 80$TIFICADO ENTRE AS MARGEM ESQUERDA = 20 E DIREITA = 50 POR EMISSAO DO CONTROLE /;$
00260024 80$R20,50/;$ E AC FINAL AS MARGENS ORIGINAIS SERAO REESTABELECIDAS PELO CONTROLE /;$R/
00270025 80$.;$R;$P$/NOVO PARAGRAFOS;$P$/;$P(COLUNA)/;$P30$ESTE PARAGRAFO INICIA NA COLUNA 30 PO
00280026 80$IS FOI EMITIDO O CONTROLE /;$P30/$. SE DESEJAMOS OS PROXIMOS PARAGRAFOS INICIANDO
00290027 80 NA MESMA COLUNA NAO E MAIS NECESSARIO INDICAR A COLUNA, E SUFICIENTE /;$P/$.;$P13
00300028 80;$SALTAR LINHAS;$P/;$S(NUMERO)/;$S5$ESTE PARAGRAFO DEVERA INICIAR-SE 5 LINHAS APOS
00310029 80 O ANTERIOR POIS FOI EMITIDO O CONTROLE /;$S5/$. CASO NAO HAJA 5 LINHAS DISPONIVE
00320030 80$IS NA PAGINA CORRENTE NOVA PAGINA SERA FORCADA.;$P$/TEXTUAL = SEM JUSTIFICACAO PEL
00330031 80$A MARGEM DIREITA;$P/;$T(TEXTO)/;$P$/TUDO ESTE TEXTO SERA IMPRESSO SEM JUSTIFICAR
00340032 80 $POIS ESTA TOTALMENTE INSERIDO NO CONTROLE /;$T$TUDO ... CONTROLE/$.;$P$/TEXTO SUBL
00350033 80$IN$F$O;$P/;$U(TEXTO)/;$P$/TUDO ESTE TEXTO SERA IMPRESSO SUBLIMADO POIS ESTA I
00360034 80$S$ERICO NO CONTROLE /;$U$TUDO ..... /$.;$P$/INDICE DE MATERIAS;$P/;$Y(NIVEL), (TEXTO)
00370035 80/;$P$/A INCLUSAO DO TEXTO /;$Y2,CAPITULO SEGUNDO/;$ EM UM INDICE DE MATERIAS FOI
00380036 80$F$EITA PELO CONTROLE /;$Y2,CAPITULO SEGUNDO/$. NA EDICAO DO INDICE ESTE TEXTO SERA
00390037 80 $EDITADO DESLOCADO DA MARGEM ESQUERDA SEGUNDO SEU NIVEL QUE NO CASO DO EXEMPLO E
00400038 80 $IGUAL A 2 (COIS).;$P$/TABELA DE REFERENCIAS;$P/;$X(TEXTO)/;$P$/A INCLUSAO DESTA /;$
00410039 80$X$PALAVRA CHAVE/;$ EM TABELA DE REFERENCIAS CRUZADAS E FEITA PELO CONTROLE /;$X$PAL
00420040 80$AVRA CHAVE/;$ QUE INSERIRA O TEXTO E O NUMERO DA PAGINA EM ARQUIVO QUE AO FINAL S
00010041 33ERA EDITADO.;$P$/CFIM DE EXEMPLOS$

```

SISTEMA EDITEXT

EXEMPLOS DE USO DOS CONTROLES

NUMERACAO DE PAGINA

\$A(NOVO NUMERO)\$

O NUMERO DESTA PAGINA FOI GERADO PELO CONTROLE \$A\$. SE O NUMERO DESEJADO PARA ESTA PAGINA FOSSE = 291, O CONTROLE SERIA \$A290\$.

BLOCO DE LINHAS CONTIGUAS EM BRANCO

\$B(NUMERO),(TEXTO)\$

A SEGUIR SERA GERADO UM BLOCO COM 10 LINHAS E COM TEXTO=TEXTO DE BLOCO DE 10 LINHAS, PELO CONTROLE \$B10,TEXTO DE BLOCO DE 10 LINHAS\$.

TEXTO DE BLOCO DE 10 LINHAS

SE O BLOCO NAO COUBER NESTA PAGINA SERA COLOCADO NA SEGUINTE.

CENTRAR TEXTO

\$C(TEXTO)\$

A LINHA SEGUINTE SERA CENTRADA NO TEXTO PELO CONTROLE \$CTEXTO CENTRADO\$.

TEXTO CENTRADO

DUPLICAR TEXTO

\$D(NUMERO),(TEXTO)\$

O CONTROLE \$D\$ GERARA UMA LINHA COMPLETA DE PONTOS

.....
O CONTROLE \$D4,(ABCDEFGGFEDCBA)\$ REPETIRA 4 VEZES O TEXTO (ABCDEFGGFEDCBA).
(ABCDEFGGFEDCBA)(ABCDEFGGFEDCBA)(ABCDEFGGFEDCBA)(ABCDEFGGFEDCBA)

NOTAS AD PE DA PAGINA

\$F(TEXTO)\$

A NOTA AD PE DA PAGINA COM TEXTO = NOTA DE PE DE PAGINA SERA GERADA PELO CONTROLE \$FNCTA DE PE DE PAGINA\$.

SISTEMA EDITEXT

EXEMPLOS DE USO DOS CONTROLES

TITULO E SUBTITULO

\$H(TIPO)(TEXTO)\$

O TITULO DESTA PAGINA FOI CRIADO COM O CONTROLE \$HTSISTEMA EDITEXT\$ E O SUBTITULO COM O CONTROLE \$HSEXEMPLOS DE USO DOS CONTROLES\$.

JUSTIFICAR

\$J(COLUNA),(TEXTO)\$

O TEXTO *TEXTO JUSTIFICADO NA COLUNA 20* FOI POSICIONADO PELO CONTROLE \$J20, TEXTO JUSTIFICADO NA COLUNA 20\$ E IMPRESSO PELO CONTROLE \$\$.

TEXTO JUSTIFICADO NA COLUNA 20

TEXTO EM MAIUSCULAS

\$M(TEXTO)\$

O TEXTO *TEXTO EM MAIUSCULAS* SERA GERADO PELO CONTROLE \$MTEXTO EM MAIUSCULAS\$.

TEXTO EM MAIUSCULAS

NOVA PAGINA

\$N

O CONTROLE \$N FORÇA NOVA PAGINA IMPRIMINDO A ULTIMA LINHA EM FORMATAÇÃO, RODAPE, NOTAS AO PE E INICIA NOVA PAGINA.

NOVAS MARGENS

\$R(MARGEM ESQUERDA),(MARGEM DIREITA)\$

TUDO O TEXTO QUE SERA IMPRESSO A SEGUIR SERA JUSTIFICADO ENTRE AS MARGEM ESQUERDA = 20 E DIREITA = 50 POR EMISSAO DO CONTROLE \$R20,50\$ E AO FINAL AS MARGENS ORIGINAIS SERAO RESTABELECIDAS PELO CONTROLE \$R\$.

NOTA DE PE DE PAGINA:

SISTEMA EDITEXT

EXEMPLOS DE USO DOS CONTROLES

NOVO PARAGRAFO

\$P(COLUNA)\$

ESTE PARAGRAFO INICIA NA COLUNA 30 POIS FOI EMITIDO O CONTROLE \$P30\$. SE DESEJAMOS OS PROXIMOS PARAGRAFOS INICIANDO NA MESMA COLUNA NAO E MAIS NECESSARIO INDICAR A COLUNA, E SUFICIENTE \$P\$.

SALTAR LINHAS

\$S(NUMERO)\$

ESTE PARAGRAFO DEVERA INICIAR-SE 5 LINHAS APOS O ANTERIOR POIS FOI EMITIDO O CONTROLE \$S5\$. CASO NAO HAJA 5 LINHAS DISPONIVEIS NA PAGINA CORRENTE NOVA PAGINA SERA FORCADA.

TEXTUAL = SEM JUSTIFICACAO PELA MARGEM DIREITA

\$T(TEXTO)\$

TODD ESTE TEXTO SERA IMPRESSO SEM JUSTIFICAR POIS ESTA TOTALMENT E INSERIDO NO CONTROLE \$TTODO ... CONTROLES\$.

TEXTO SUBLINHADO

\$U(TEXTO)\$

TODD ESTE TEXTO SERA IMPRESSO SUBLINHADO POIS ESTA INSERIDO NO CONTROLE \$UTODD \$.

INDICE DE MATERIAS

\$Y(NIVEL),(TEXTO)\$

A INCLUSAO DO TEXTO 'CAPITULO SEGUNDO' EM UM INDICE DE MATERIAS FOI FEITA PELO CONTROLE \$Y2,CAPITULO SEGUNDO\$. NA EDICAO DO INDICE ESTE TEXTO SERA EDITADO DESLOCADO DA MARGEM ESQUERDA SEGUNDO SEU NIVEL QUE NO CASO DO EXEMPLO E IGUAL A 2 (DOIS).

TABELA DE REFERENCIAS

\$X(TEXTO)\$

A INCLUSAO DESTA 'PALAVRA CHAVE' EM TABELA DE REFERENCIAS CRUZADAS E FEITA PELO CONTROLE \$XPALAVRA CHAVE\$ QUE INSERIRA O TEXTO E O NUMERO DA PAGINA EX ARQUIVO QUE AO FINAL SERA EDITADO.

V.3 USO DOS CARACTERES ESPECIAIS

SISTEMA EDITEXT

EXEMPLOS DE USO DOS CARACTERES ESPECIAIS

CARATER MODIFICADOR

/ (BARRA)

O CARATER MODIFICADOR TEM COMO FUNCOES, PERMITIR QUE OS CARACTERES ESPECIAIS \$, /, E & POSSAM SER EDITADOS E SIMULAR UM CARATER MAIUSCULO OU SUBLINHADO.

EXEMPLOS - O CARATER A FOI GERADO PELO CONTROLE /A E O CARATER & PELO CONTROLE /&.

CARATER SIMULADOR

' (APOSTROFE)

CARATER SIMULADOR PERMITE SIMULAR A IMPRESSAO DE UM ACENTO (PELA SUPERIMPRESSAO DO ') OU DE UM CEDILHA (PELA SUPERIMPRESSAO DA `).

EXEMPLOS - O TEXTO 'FOR'CADA' GERA 'FORÇADA' E 'PAR'A' GERA 'PARÁ'.

CARATER BRANCO ESPECIAL

& (E COMMERCIAL)

CARATER BRANCO ESPECIAL E UTILIZADO PARA IMPEDIR A INSERCAO DE BRANCOS DE JUSTIFICACAO ENTRE DUAS PALAVRAS CONSECUTIVAS DE UMA LINHA OU GERAR UM CAMPO BRANCO EM UMA LINHA.

EXEMPLOS - ENTRE 'A - B' NAO FORAM INSERIDOS BRANCOS DE JUSTIFICACAO POIS O TEXTO ENTRADA FOI 'A&-&B'.

FIM DE EXEMPLOS