

UM EDITOR INTERATIVO

PARA O TERMINAL INTELIGENTE

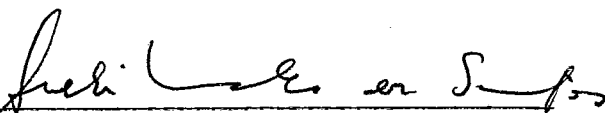
William Barbosa de Araujo

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

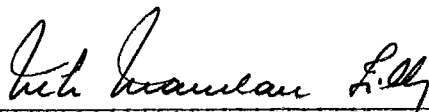
Aprovada por:



PAULO MARIO BIANCHI FRANÇA



SUELI MENDES DOS SANTOS



NELSON MACULAN FILHO

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 1981

ARAUJO, WILLIAM BARBOSA DE

Um Editor Interativo Para o Terminal
Inteligente Rio de Janeiro 1981.

VII, 68 p. 29,7 cm (COPPE-UFRJ, M.Sc,
Engenharia de Sistemas e Computação, 1981)

Tese - Univ. Fed. Rio de Janeiro. Coor-
denação dos Programas de Pós-Graduação em
Engenharia

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

Aos meus avôs:

Arnaud,
Maria do Carmo,
Antônio e
Petronilha.

AGRADECIMENTOS

Ao professor GUILHERME CHAGAS RODRIGUES, pela orientação e incentivo recebidos durante a execução deste trabalho.

A Paulo César Moraes Melo e José Antônio dos Santos Borges pelo auxílio na utilização do terminal e pelas críticas e sugestões.

Ao NCE/UFRJ pelo equipamento utilizado.

RESUMO

Descreve o projeto e a implementação de um editor de textos interativo para o terminal inteligente desenvolvido pelo Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro.

São apresentados os critérios para a escolha das funções de edição, do modo de operação e da estrutura do arquivo empregado para armazenar o texto em disco magnético.

O editor procura fornecer ao usuário as facilidades permitidas pela associação de um teclado dotado de teclas de controle a um terminal de vídeo baseado em tubo de raios catódicos.

Na tela do terminal, é exibida uma janela bi-dimensional através da qual o operador pode visualizar e modificar o conteúdo do texto. Um cursor luminoso indica a posição em que o texto vai ser modificado. A janela pode ser deslocada sobre o texto de forma que qualquer porção do mesmo possa ser exibida.

Embora orientado para edição de programas escritos em linguagem de alto nível, o editor permite também a edição de textos em formato livre.

ABSTRACT

Describes the design and implementation of an interactive text editor for the intelligent terminal developed by Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro.

The criterion for selecting the editing functions, the mode of operation and the structure of the file employed to store the text are presented.

The editor tries to give to the user the power generated by the association of a keyboard with control keys and a CRT display.

A two-dimensional window into the text is displayed on the screen. A cursor points the place where the text will be modified. The window can be moved up and down the file in order to edit different parts of it.

Although oriented to program editing it can be also used to edit free form texts.

INDICE

I. INTRODUÇÃO

Antecedentes	01
Proposição	02
Equipamento	03
Sistema operacional	04
Aplicações	05
Usuários	06
Diretrizes	07
Objetivos	08

II. PROJETO

Perfil do usuário	09
Princípios de projeto	10
Características do editor	13
Modelo do editor	17

III. DESCRIÇÃO DO EDITOR

Utilização	18
Formato da tela	19
Funções do editor	20
Funções de controle	21
Funções de movimento do cursor	22
Funções de movimento do texto	23
Funções de modificação do texto	24
Mensagens de erro	25
Comandos do SOCO	27
Descrição das funções de controle	29
Descrição das funções de movimento do cursor	32
Descrição das funções de movimento do texto	34
Descrição das funções de modificação do texto	36

IV. IMPLEMENTAÇÃO

Definição do programa	39
Estrutura do arquivo de texto	40
Níveis de implementação.	42
Módulos	43
Descrição dos módulos	44
Tabelas	54
Descrição das tabelas	55

V. CONCLUSÃO

Avaliação	63
Extensões	65

VI. BIBLIOGRAFIA	67
----------------------------	----

I. INTRODUÇÃO

Antecedentes

O Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE) desenvolveu, em um projeto integrado de 'hardware' e 'software', um terminal inteligente baseado no microprocessador INTEL-8080. Este terminal pode ser ligado a um computador de maior porte ou ser utilizado de modo independente para executar aplicações típicas dos pequenos computadores.

Para possibilitar a compilação e o teste de programas para o terminal, em paralelo com a construção do primeiro protótipo, o Sistema Operacional de Simulação (SOS) foi implementado no computador BURROUGHS 6700 disponível no NCE. Graças ao emprego da técnica da compilação cruzada, que consiste na compilação de um programa para um novo computador utilizando um outro computador já existente, o sistema SOS permitiu o desenvolvimento de uma versão mínima do sistema operacional projetado para o terminal, o Sistema Operacional de Disco (SOCO).

Após a implementação da versão mínima do sistema SOCO no primeiro protótipo, o terminal estava capacitado a compilar e executar seus próprios programas, inclusive a finalizar o desenvolvimento do sistema SOCO. Entretanto, algumas dificuldades tornavam a tarefa de desenvolver programas no terminal excessivamente trabalhosa.

Os programas fonte eram perfurados em cartões e alimentados no terminal por meio de uma leitora de cartões. Como a leitora apresentasse um comportamento errático, todo programa lido tinha que ser inspecionado visualmente na tela do terminal de vídeo. Sucessivas leituras eram feitas até que se obtivesse uma leitura correta. Caso algum erro fosse encontrado durante a compilação ou durante o teste do programa, o usuário deveria deslocar até a sala das perfuradoras, perfurar um novo cartão e substituir o cartão errado. A partir daí, todo o processo tinha que ser repetido desde o início.

Proposição

Considerando a utilidade de um programa que permitis_ se a criação de novos programas e a modificação de programas já existentes, este trabalho se propõe a definir e implementar um e_ditor de textos interativos para o terminal inteligente desenvol_vido pelo NCE.

Embora orientado para a edição de programas, o edi_tor poderá também editar qualquer outro tipo de texto, tais como: manuais, instruções, notas de aula, artigos, correspondências , etc...

As modificações realizadas no texto deverão ser mos_tradas à medida que forem comandadas pelo operador, aproveitando as potencialidades da associação de um teclado com teclas de ca_racteres e contrôle à uma unidade de vídeo baseada em tubo de raios catódicos.

Por meio do editor, poderemos:

- 1 - criar novos textos;
- 2 - modificar textos existentes;
- 3 - retirar linhas do texto;
- 4 - inserir novas linhas no texto;
- 5 - copiar linhas de um outro texto.

Equipamento

O terminal inteligente é um microcomputador que pode se comunicar com outros computadores através de uma linha de comunicações. A ele podem ser ligados quaisquer dispositivos de entrada e saída, através de uma interface adequada.

A configuração mínima prevista para o terminal consta de:

- 1 - CPU baseada no microprocessador INTEL-8080, com um tempo de execução por instrução, de 2 us a 9 us;
- 2 - Memória ROM para o programa de carga inicial;
- 3 - Memória RAM, 16 K bytes, para armazenamento de dados e instruções. Tem uma largura de 8 bits e tempo de acesso da ordem de 480 ns;
- 4 - Teclado, MICROSWITCH, com teclas de caracteres e de controle, em código ASCII;
- 5 - Unidade de vídeo, baseada em tubo de raios catódicos, com 24 linhas de 80 caracteres, mapeados diretamente da memória interna do terminal a partir do endereço 1000_{16} ;
- 6 - Uma unidade de disco, DIGITAL RK-05, removível com capacidade de 2,4 M bytes. Com 203 trilhas por superfície, formatadas em 12 setores de 256 bytes cada. O tempo de posicionamento é de 10 ms para uma trilha e de 50 ms em média. O tempo de estabilização do braço da leitura é de 20 ms. A velocidade de transferência é de 11,1 us por palavra (16 bits).

Sistema Operacional

O sistema operacional SOCO proporciona ao terminal a capacidade de compilar e executar programas. Apresenta uma concepção bastante simples, mas suficientemente poderosa para possibilitar a plena utilização dos recursos do terminal por usuários com diferentes graus de treinamento e diferentes necessidades. O sistema SOCO é formado pelos seguintes módulos:

- 1 - programa de carga inicial armazenando em memória ROM;
- 2 - núcleo residente;
- 3 - carregador de programas;
- 4 - rotinas de entrada e saída;
- 5 - editor de referências externas;
- 6 - compilador e interpretador da linguagem PLTI, uma linguagem semelhante ao PL/M da INTEL, desenvolvida especialmente para o terminal;
- 7 - rotinas de uso geral;
- 8 - programas utilitários;
- 9 - editor de textos.

Aplicações

O terminal é utilizado para executar três principais tipos de aplicações:

- 1 - desenvolvimento da programação básica (sistema operacional) do terminal;
- 2 - desenvolvimento de pacotes de programas de aplicação;
- 3 - execução dos programas de aplicação.

Usuários

A cada uma das principais aplicações do terminal, não de uma maneira muito rígida, podemos associar um tipo de usuário, com características e interesses próprios:

- 1 - profissionais de processamento de dados;
- 2 - alunos de cursos de processamento de dados;
- 3 - usuários sem formação em processamento de dados.

Diretrizes

Sendo o editor de textos um módulo do sistema operacional SOCO, devemos seguir as mesmas diretrizes adotadas para o desenvolvimento dos demais módulos:

- 1 - utilizar a configuração mínima do terminal inteligente;
- 2 - ser compatível com os demais módulos do sistema operacional SOCO;
- 3 - ser programado em linguagem PLTI;
- 4 - operar de modo interativo;
- 5 - ser fácil de aprender;
- 6 - ser fácil de utilizar;
- 7 - ser fácil de implementar;
- 8 - ser fácil de modificar.

Além de atender a essas diretrizes gerais, ênfase deve ser colocada em fornecer ao usuário o máximo de potencialidade permitido pelos recursos disponíveis no terminal. A comodidade do usuário deve ser considerada mais importante do que uma eficiência operacional ótima. As modificações devem ser permitidas em qualquer parte do texto e na sequência que o usuário desejar.

Objetivos

Os objetivos de um editor de textos, do ponto de vista do usuário, são:

- 1 - resposta rápida;
- 2 - linguagem de comandos concisa e mnemônica;
- 3 - comandos poderosos, com poucas excessões e poucas restrições.
- 4 - comandos que aproveitem as capacidades do computador. Por exemplo, procurar a primeira ocorrência de uma sequência de caracteres.

Estes objetivos não são independentes. Em alguns casos, são até mesmo conflitantes. Um repertório de instruções poderosas implica na utilização de uma maior quantidade de recursos, acarretando um maior tempo de resposta. Deve-se buscar o equilíbrio no atendimento destes objetivos, de modo a produzir uma solução de compromisso satisfatória.

II . PROJETO

Perfil do usuário

Os usuários do editor de textos são os mesmos do terminal: profissionais de processamento de dados, alunos de cursos de processamento de dados e usuários sem formação em processamento de dados. Devido às grandes variações existentes nas formações, nas experiências, nos interesses e nas habilidades destes usuários, não podemos estabelecer um único perfil para o usuário. Consideramos, para efeito de projeto, que o usuário típico do editor é um aluno de curso de processamento de dados desenvolvendo programas de aplicação escritos em linguagem de alto nível (PLTI). Esta simplificação, entretanto, não deve implicar em restrições importantes à utilização do editor por qualquer dos tipos de usuários previstos.

Além do grau de instrução do usuário, o projeto deve também tomar em consideração diferentes frequências de utilização por parte dos mesmos, sendo capaz de atender tanto ao usuário eventual que deseja modificar uma única linha do texto como ao que trabalha em tempo integral criando e modificando programas e textos.

Outra importante característica a ser considerada é a habilidade do usuário como datilógrafo. O editor não deve exigir mais do que a mínima capacidade de operar o teclado e deve ser capaz de atender até a exímios datilógrafos que não olham para o teclado à medida que digitam e acompanham o aparecimento dos caracteres na tela.

Princípios de projeto

Apesar das diferenças entre os usuários, eles compartilham duas importantes características: esquecem e cometem erros.

Durante a operação de um sistema interativo, o usuário pode esquecer como executar determinada função, o conteúdo do arquivo e, se interrompido, até mesmo o que estava fazendo. O trabalho sob forte pressão de tempo e em ambientes dispersivos, frequentemente ocasiona a ocorrência de muitos erros.

Para minimizar o impacto negativo destes fatores, o editor deve ser projetado como uma ferramenta de auxílio ao desenvolvimento de programas, atuando como um amplificador das capacidades do operador. Isto pode ser atingido pelo emprego de três princípios de projeto:

- 1 - Minimizar as necessidades de memorização;
- 2 - Otimizar a execução das operações;
- 3 - Minimizar possibilidades de erro.

1 - Minimizar necessidade de memorização

a - Selecionar entre opções

Deve ser apresentado ao usuário um repertório de escolhas ("menu") a partir do qual ele seleciona, a cada passo, a opção desejada, evitando que tenha de decorar uma série de códigos de operação.

b - Evitar a utilização de números

As escolhas devem ser feitas a partir de nomes e não de números.

c - Apresentar comportamento previsível

O usuário deve dispor de um modelo de funcionamento do editor que lhe permita prever as consequências da execução dos comandos fornecidos.

d - Fornecer acesso às informações internas

O usuário deve ter acesso às informações internas que indicam o estado corrente do editor.

2 - Otimizar operações

Este princípio está ligado à aparência externa do editor e se baseia em evitar que o programa interfira no processo consciente da edição de textos, não criando obstáculos à execução das operações e permitindo que o usuário se concentre nas modificações que deseja introduzir no texto.

a - Executar operações comuns rapidamente

As operações mais comuns (mais frequentes) devem ser executadas rapidamente.

b - Manter inércia de exibição

A informação apresentada na tela deve ser modificada o mínimo necessário para cumprir a execução de cada comando.

c - Aproveitar a memória muscular

Operações muito repetitivas podem ser executadas sem o controle contínuo da consciência. Para aproveitar esta característica, o editor deve ser capaz de aceitar rápidas rajadas de comando e utilizar cada tecla com apenas um significado, independente do modo de operação.

3 - Minimizar possibilidades de erro

O editor deve ser projetado de modo a tornar difícil a ocorrência de erros catastróficos e permitir a recuperação da maior quantidade de erros possíveis. Isto pode ser obtido a partir de :

a - Fornecer boas mensagens de erro

A disponibilidade de uma unidade de vídeo permite a emissão de mensagens indicando a causa do erro com um mínimo de perda de tempo.

b - Impedir erros comuns

O editor deve ser projetado de modo a dificultar a execução acidental de operações que possam modificar substancialmente o conteúdo do texto.

c - Prover comandos reversíveis

Para facilitar a recuperação de erros acidentais, o editor deve prover meios para que toda modificação introduzida no texto possa ser revertida, retornando o texto à sua forma original.

d - Prover integridade dos dados

A estrutura dos dados armazenados dever ser tal que apesar dos erros de equipamento e do operador, alguma versão do texto seja sempre preservada.

Características do editor

As características de um editor de textos são fortemente influenciadas pelas características do equipamento utilizado e pelo tipo de texto a que o editor se destina. Embora seja possível o projeto de um editor que possa tratar adequadamente tanto programas escritos em linguagem formal quanto textos de formato livre, considerações de ordem prática nos levaram a dirigir o projeto para um editor de programas capaz de editar, ainda que com algumas restrições, textos em formato livre. A partir desta orientação, relacionamos as seguintes características para o editor:

1 - Tipo de edição

Os editores de texto livre são orientados para modificar, retirar ou inserir uma sequência de caracteres (palavra) de cada vez. A largura física do meio utilizado para representar o texto (cartão, tela, formulário) não está relacionada com a estrutura do texto, que é organizado em parágrafos, capítulos, etc. Os editores de programa, por outro lado, são orientados para modificar um caracter de cada vez e as declarações do programa, geralmente, estão associadas a uma linha. Considerando que a utilização mais frequente definida para o editor é a edição de programas, e que um editor de programas é mais simples e mais adequado ao pequeno porte do terminal, optamos por desenvolver um editor orientado para programas, capaz de manipular também, ainda que com menor facilidade, textos em formato livre.

2 - Repertório de funções

O repertório de funções está também associado ao tipo de texto a ser editado. A opção por um editor de programas conduz a escolha de um repertório de funções poderosas (que permitam ao usuário realizar com facilidade modificações que deseja) e orientadas para a modificação de caracteres e linhas. Para maior comodidade do usuário, algumas funções redundantes (que poderiam ser executadas por combinação de outras funções) foram definidas.

3 - Linguagem de comandos

Um comando para um editor de textos deve especificar a operação a ser executada e o ponto do texto a ser modificado.

A operação pode ser especificada pela utilização de um código de operação (geralmente abreviado e mnemônico) ou pelo acionamento de uma tecla de função. Optamos por utilizar as teclas de função por apresentarem as seguintes vantagens sobre os códigos de operação:

a - Minimiza memorização

Se as teclas de função são identificadas, o usuário dispõe, em sua frente, de um "menu", representado pelo teclado, a partir do qual pode selecionar a função desejada.

b - Otimiza operações

A utilização de teclas de função minimiza o número de teclagens necessárias para especificar um comando. Além de permitir que o usuário seja mais rápido, permite uma execução mais rápida, uma vez que não é necessário interpretar um comando digitado, caracter por caracter.

c - Minimiza erros

Diminuindo o número de digitações necessárias a especificar o comando, diminui, conseqüentemente, as possibilidades de ocorrência de enganos.

A indicação da posição do texto a ser modificado pode ser feita a partir do número da linha em relação ao início do texto ou por um cursor luminoso na tela. Como o terminal de vídeo dispõe da facilidade de representar o cursor, optamos por utilizar o mesmo, tendo em vista que apresenta as seguintes vantagens:

a - Minimiza memorização

O usuário não precisa de lembrar do número da linha a ser modificada.

b - Otimiza operações

O número da linha muda dinamicamente durante a edição, como consequência de inserções e retiradas de linhas. O uso do cursor cria uma inércia na representação da localização da linha.

4 - Armazenamento do texto

A forma pela qual o texto é armazenado na memória interna é dependente, principalmente, do tamanho da memória e do tipo do texto. Os textos em formato livre são melhor armazenados em uma forma comprimida onde os espaços são suprimidos e indicadores de fim de linha, fim de página, início de parágrafo e outros controles de edição são armazenados embebidos no próprio texto. Os textos de programa, dado sua simplicidade, caso não haja problemas com a memória, podem ser armazenados em formato: imagem, linha a linha.

5 - Apresentação do texto

A forma pela qual o texto é apresentado durante a edição depende do dispositivo empregado para apresentação, da velocidade de processamento e do modo em que o texto está armazenado. Impressoras lentas permitem a apresentação de uma tecla de cada vez e sob solicitação do usuário. Telas de tubos de raios catódicos apresentam grande quantidade de informação em alta velocidade e são ideais para a edição interativa de textos.

Se o processador é bastante rápido o texto pode estar comprimido ou não. Para processadores mais lentos, a sobrecarga para reformatar os dados antes de apresentá-los na tela torna mais indicado o uso de dados sem compressão.

Os dispositivos impressores, embora não sendo os mais indicados para a edição interativa, permitem o registro permanente da sessão de edição. Vale lembrar, também, que uma impressora de boa qualidade, dispondo de tipos maiúsculos e minúsculos, é de grande valia para o processamento de textos em formato livre.

Modelo do editor

A partir do levantamento do perfil do usuário e das suas necessidades, considerando os princípios de projeto e das características dos editores de texto, procuramos conciliar as características do equipamento disponível para elaborar um modelo do editor a ser implementado.

Este modelo considera a edição de programas e prevê a edição de um arquivo contendo as declarações em um formato imagem de cartão. O editor deve ser capaz de permitir as modificações que usualmente são feitas em uma massa de cartões com auxílio de uma máquina perfuradora.

O texto é formado por um conjunto de linhas de 80 caracteres. A tela do terminal do vídeo é utilizada como uma janela bi-dimensional através da qual as linhas podem ser visualizadas e modificadas, um cursor luminoso sobre a tela indica o ponto do texto a ser modificado. As modificações são feitas a medida que são comandadas e o usuário vê o resultado podendo acompanhar a edição de modo interativo. Os comandos são fornecidos ao editor por meio de teclas de função e controle do teclado. O cursor não sai dos limites da tela, ficando retido na margem correspondente ao sentido do movimento. A primeira linha da tela só pode ser rolada por solicitação explícita do usuário. Só podem ser modificadas linhas que estejam na tela e só podem ser retiradas ou inseridas linhas que estejam na tela. A atualização do arquivo em disco se faz a medida que as linhas modificadas ou inseridas são roladas para fora da tela. As linhas retiradas devem ser mantidas no arquivo até o final da edição. Por motivo de segurança, o usuário deve editar uma cópia do arquivo original.

III. DESCRIÇÃO DO EDITOR

Utilização

O editor de textos é um módulo do sistema operacional do terminal inteligente que permite a criação e modificação interativas de programas e textos em formato livre armazenados em arquivos de disco magnético. Os arquivos contêm registro de 80 caracteres, em formato imagem de cartão.

A unidade de vídeo fornece ao usuário uma janela bidimensional através da qual ele pode visualizar e modificar o texto que está editando. A posição do texto em que será efetuada a modificação é indicada na tela por um cursor. O cursor é representado por um caracter exibido na forma negativa, isto é, caracter escuro sobre fundo claro. A medida que o operador digita caracteres no teclado, estes substituem o caracter que se encontra sobre o cursor e o cursor avança para o caracter seguinte, em direção ao final da linha. Erros de datilografia podem ser facilmente corrigidos, retrocedendo o cursor e digitando o caracter correto sobre o errado.

Utilizando as teclas de controle do teclado, o operador pode apagar, copiar, inserir e retirar linhas ou caracteres do texto, movimentar a janela sobre o arquivo e posicionar o cursor sobre a tela. Pode também fornecer um argumento numérico que determina múltiplas execuções de uma mesma função com um único comando. Para facilitar o posicionamento de um determinado trecho do arquivo na tela do terminal, existe uma função que procura a primeira ocorrência de uma sequência de caracteres.

Um segundo arquivo, já existente, pode ser utilizado na edição e dele se podem visualizar e copiar linhas. Não são permitidas modificações no arquivo auxiliar. Mais de um arquivo auxiliar pode ser utilizado em uma única sessão de edição.

Formato da Tela

O editor utiliza a tela da unidade de vídeo como veículo para comunicação com operador. As vinte e quatro linhas da tela foram formatadas da seguinte maneira:

- 1 - As vinte primeiras linhas da tela foram reservadas para exibir continuamente uma página do texto em edição, formada por vinte linhas consecutivas do texto. A primeira linha da página denomina-se topo da página e a última linha denomina-se base da página.
- 2 - a vigésima primeira linha da tela foi deixada em branco, de modo a permanecer apagada e prover uma separação entre o texto e as demais linhas da tela.
- 3 - a vigésima segunda linha da tela foi destinada à apresentação do texto do arquivo auxiliar da edição, uma linha de cada vez
- 4 - a vigésima terceira linha da tela foi reservada para comunicações do editor com o operador, solicitando informações ou indicando uma condição de erro encontrada ao tentar executar um comando.
- 5 - a vigésima quarta linha da tela, a mais próxima do teclado, foi destinada a reproduzir as respostas do operador às solicitações do editor e os argumentos fornecidos como parâmetros para modificar a execução da função subsequente.

Funções do Editor

As funções do editor de textos podem ser classificadas, de acordo com sua finalidade principal, em quatro categorias:

- 1 - funções de controle da edição;
- 2 - funções de movimento do cursor;
- 3 - funções de movimento do texto;
- 4 - funções de modificação do texto.

Funções de Contrôle

As funções de controle permitem o controle de processo de edição e a passagem de argumentos para a execução das demais funções. São elas:

- 1 - Entrar com argumento;
- 2 - Indicar arquivo auxiliar;
- 3 - Posicionar Arquivo Auxiliar;
- 4 - Guardar Linhas do Texto;
- 5 - Exibir Estado do Editor;
- 6 - Terminar Sessão de Edição.

Funções de Movimento do Cursor

As funções de movimento do cursor permitem o posicionamento do cursor sobre qualquer ponto da página.

- 1 - Mover cursor para direita;
- 2 - Mover cursor para esquerda;
- 3 - Mover cursos para baixo;
- 4 - Mover cursor para cima;
- 5 - Mover cursor para início da linha corrente;
- 6 - Mover cursor para início da linha seguinte;
- 7 - Mover cursor para início da tela;
- 8 - Mover cursor para tabulação à direita;
- 9 - Mover cursor para tabulação à esquerda.

Funções de Movimento do Texto

Rolam o texto para cima e para baixo, de modo que qualquer porção do mesmo possa ser exibida na página da tela do terminal do vídeo.

- 1 - Rolar uma linha para frente;
- 2 - Rolar uma linha para trás;
- 3 - Rolar uma página para frente;
- 4 - Rolar uma página para trás;
- 5 - Rolar para frente até encontrar uma cadeia;
- 6 - Rolar para trás até encontrar uma cadeia;
- 7 - Rolar para última página do texto.
- 8 - Rolar para primeira página do texto.

Funções de Modificação do Texto

Modificam a porção do texto exibida na tela do terminal de vídeo.

- 1 - Substituir o caracter sob o cursor;
- 2 - Copiar o caracter acima do cursor;
- 3 - Retirar o caracter sob o cursor;
- 4 - Inserir um espaço sob o cursor;
- 5 - Apagar o resto da linha a partir do cursor;
- 6 - Copiar linha da memória;
- 7 - Copiar linha do arquivo auxiliar;
- 8 - Trocar duas linhas;
- 9 - Retirar a linha sob o cursor;
- 10 - Inserir uma linha sob o cursor.

Mensagens de Erro

Toda vez que o editor se encontra na impossibilidade de executar um comando, uma mensagem é movida para a linha de mensagens da tela e o alarme sonoro do teclado é acionado de modo a alertar o operador. O editor emite as seguintes mensagens:

1- .NOME INVALIDO

Motivo: O nome de arquivo fornecido ao editor para utilização como arquivo auxiliar não obedece as regras de sintaxe para formação de nomes de arquivo. Lembrar que o nome de arquivo deve começar obrigatoriamente com uma letra e ser formado de até seis letras ou algarismos.

Ação: Fornecer um nome válido.

2- .ARQUIVO INEXISTENTE

Motivo: O arquivo com o nome fornecido não foi encontrado no diretório do disco.

Ação: Fornecer o nome de um arquivo existente.

3- .ERRO NO DISCO

Motivo: O terminal detectou um erro durante uma operação de leitura ou escrita no disco.

Ação: Este erro não é recuperável e causa uma terminação anormal da sessão de edição. Verifique se a unidade de disco está ligada e se não se encontra protegida contra escrita.

4- .ESPAÇO ESGOTADO

Motivo: Não há mais espaço em disco para inclusão de novas linhas ou a tabela de linhas deslocadas está cheia.

Ação: Encerre a sessão de edição de modo a forçar a reorganização do arquivo. Inicie uma no-

va sessão e prossiga. Caso o erro persista, crie um arquivo maior e copie o texto para ele.

5- .CURSOR RETIDO

Motivo: Foi solicitado um deslocamento do cursor para fora dos limites da tela. O cursor ficou retido no limite da tela correspondente ao sentido do movimento.

Ação: Nenhuma, prossiga normalmente.

6- .LIMITE TEXTO

Motivo: Foi solicitado um rolamento do texto além de seus limites. Pelo menos uma linha do arquivo deve permanecer na tela.

Ação: Nenhuma, prossiga normalmente.

7- .FUNÇÃO INEXISTENTE

Motivo: Foi digitada uma tela de controle para a qual não existe nenhuma função do editor implementada.

Ação: Digite outra tecla.

8- .EXECUÇÃO INCOMPLETA

Motivo: Impossibilidade total ou parcial, de executar a função comandada.

Ação: Nenhuma, prossiga normalmente.

9- .CADEIA INEXISTENTE

Motivo: Durante uma operação de busca de cadeia no texto, a cadeia não foi encontrada.

Ação: Nenhuma, prossiga normalmente.

Comandos do SOCO

Apresentamos aqui alguns comandos do sistema operacional SOCO, úteis para a edição de textos:

- 1 - Criação de um arquivo denominado TEXTO, formado por 200 registros de 80 caracteres.

CRIE 'TEXTO COM 200 DE 80'.

- 2 - Copiar o arquivo texto para um arquivo denominado FONTE.

COPIE 'TEXTO PARA FONTE'.

- 3 - Editar o arquivo chamado FONTE.

EDITE 'FONTE'.

- 4 - Para compilar um programa escrito em linguagem PLTI, armazenado no arquivo FONTE.

PLTI.

- 5 - Para resolver as referencias externas de um programa chamado PROG, já compilado.

MONTE 'PROG'.

- 6 - Para executar o programa PROG.

PROG.

- 7 - Para obter uma cópia impressa do arquivo FONTE.

COPIE 'FONTE PARA I'.

- 8 - Para listar o arquivo TEXTO no terminal de vídeo:

COPIE 'TEXTO PARA V'.

- 9 - Para obter uma cópia impressa do arquivo TEXTO, utilizando os caracteres de controle embebidos' no texto.

LISTE 'TEXTO'.

- 10 - Para retirar o arquivo do disco:

APAGUE 'TEXTO'.

Descrição das funções de controle

Entrar com argumento

Permite ao usuário fornecer uma cadeia formada por até 80 caracteres e que será utilizada como parâmetro para a execução da próxima função solicitada. Caso a função não utilize ' argumento, ele será ignorado.

A cada argumento é associado um valor numérico que é igual a 1 (um) para as cadeias iniciadas por caracteres alfabéticos e igual ao número formado pelos primeiros algarismos da cadeia, até um máximo de quatro, para as cadeias iniciadas por algarismos.

Um cursor é colocado na primeira coluna da linha de argumento. À medida que as teclas são digitadas, o caracter correspondente substitui o cursor e este avança para direita. Utilizando-se teclas de controle, o cursor pode ser movido para direita e para esquerda. Quando uma tecla de função é digitada, os caracteres do início da linha até o cursor, inclusive, são utilizados como argumento. Os caracteres à direita do cursor são ignorados. Se a tecla de função é digitada imediatamente após a tecla de entrar com argumento, o último argumento utilizado é repetido.

Indicar arquivo auxiliar

Por meio desta função o usuário pode indicar um segundo arquivo do qual podem ser copiadas linhas para o texto que está sendo editado. Embora só se possa utilizar um arquivo auxiliar de cada vez, mais de um arquivo pode ser consultado durante uma única sessão de edição. Esta função utiliza o argumento para obter o nome do arquivo. O nome do arquivo deve iniciar por uma letra, ser formado por letras e algarismos e conter, no máximo, seis letras ou algarismos. Caso o nome fornecido não satisfaça as regras de sintaxe ou não seja encontrado um arquivo com este nome no disco, o operador é advertido. Caso contrário, a primeira linha do arquivo é exibida na linha do arquivo auxiliar da tela.

Posicionar arquivo auxiliar

Permite a exibição, na tela do terminal de vídeo, de qualquer linha do arquivo auxiliar.

Se um arquivo auxiliar não foi anteriormente indicado com sucesso, a função retorna com uma mensagem de erro. Se não é fornecido um argumento para esta função, a linha seguinte à que está sendo apresentada na tela é exibida. Caso seja fornecido um argumento, ele é interpretado do seguinte modo:

- a - Se o argumento inicia com o caracter '=', o número a seguir é interpretado como número da linha do arquivo auxiliar que se deseja exibir;
- b - Se o argumento inicia com o caracter '+', o número a seguir indica quantas linhas à frente da linha atualmente exibida se encontra a linha que se deseja exibir;
- c - Se o argumento inicia com o caracter '-', o número a seguir indica quantas linhas atrás da linha atualmente exibida se encontra a linha que se deseja exibir.

Caso os limites do arquivo sejam ultrapassados, a exibição se limita à primeira ou à última linha do arquivo auxiliar, conforme o sentido da ultrapassagem.

Guardar linhas do texto

Por meio desta função, até vinte linhas do texto que estejam sendo exibidas na tela do terminal de vídeo, podem ser armazenadas em uma área interna de memória do editor. Se a função é solicitada sem a utilização do argumento, a linha que contém o cursor é armazenada. Caso se utilize o argumento, este indica quantas linhas, a partir da que contém o cursor, inclusive, devem ser armazenadas. Somente linhas que estejam sendo exibidas na tela podem ser armazenadas. Esta função é útil para mover um trecho do texto de um ponto para outro do mesmo.

Exibir estado do editor

Esta função coloca na linha de mensagens da tela do terminal de vídeo os valores correntes das seguintes variáveis internas, indicativas do estado do editor:

- a - número da coluna em que se encontra o cursor;
- b - número da linha da tela em que se encontra o cursor;
- c - número da linha do texto em que se encontra o cursor;
- d - número da linha do texto que se encontra no topo da página;
- e - número total de linhas no texto;
- f - número da linha do arquivo auxiliar que está sendo exibida;
- g - número total de linhas no arquivo auxiliar;
- h - número de linhas disponíveis para inserções;
- i - número de entradas disponíveis na tabela de linhas deslocadas.

Terminar sessão de edição

Esta função termina a sessão de edição, causando a reorganização do arquivo de texto, isto é, as linhas deslocadas são colocadas nas posições correspondentes no arquivo de disco.

Descrição das funções de movimento do cursor

Mover cursor para direita

O cursor é movido para a próxima coluna, à direita, em direção ao final da linha. Quando atinge o final da linha, o cursor não mais se desloca, ficando retido na margem.

Mover cursor para esquerda

O cursor é movido para a coluna anterior, à esquerda, em direção ao início da linha. Quando atinge o início da linha, o cursor não mais se desloca, ficando retido na margem.

Mover cursor para baixo

O cursor é movido para linha imediatamente abaixo da que se encontra, em direção à base da tela, permanecendo na mesma coluna. Quando atinge a base da tela, o cursor não mais se desloca, ficando retido.

Mover cursor para cima

O cursor é movido para linha imediatamente acima da que se encontra, em direção ao topo da tela, permanecendo na mesma coluna. Quando atinge o topo da tela, o cursor não mais se desloca, ficando retido.

Mover cursor para início da linha corrente

O cursor é movido para primeira coluna da linha em que se encontra.

Mover cursor para início da linha seguinte

O cursor é movido para primeira coluna da linha imediatamente abaixo da linha em que se encontra. Caso o cursor esteja na linha de base, a função não é executada.

Mover cursor para início da página

O cursor é movido para primeira coluna da linha de topo da tela, no canto superior esquerdo da mesma.

Mover cursor para tabulação à direita

O cursor é movido para próxima coluna tabulada à direita. São consideradas como tabuladas a primeira coluna da linha, as colunas 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 69, 73, 77 e a última coluna da linha. Caso atinja a última coluna da linha, o cursor fica retido na margem.

Mover cursor para tabulação à esquerda

O cursor é movido para coluna anterior, à esquerda, tabulada. Caso atinja a primeira coluna da linha, o cursor fica retido na margem.

Descrição das funções de movimento do texto

Rolar uma linha para frente

A linha do texto que se encontra na linha de topo da tela é retirada, as demais linhas são deslocadas para cima e uma nova linha do texto é exibida na linha de base, causando o deslocamento da tela em direção ao final do texto. Com a utilização de um argumento numérico, pode-se rolar mais de uma linha de cada vez, por meio de um único comando. Caso a execução do comando cause a saída do texto dos limites da tela, a linha indicativa de fim de texto é exibida na linha de topo da tela.

Rolar uma linha para trás

A linha do texto que se encontra na linha de base da tela é retirada, as demais linhas são deslocadas para baixo e uma nova linha do texto é exibida na linha de topo, causando o deslocamento da tela em direção ao início do texto. Com a utilização de um argumento numérico, pode-se rolar mais de uma linha de cada vez por meio de um único comando. Caso a execução do comando cause a saída do texto dos limites da tela, a primeira linha do texto é exibida na linha de topo da tela.

Rolar uma página para frente

Esta função corresponde ao rolamento de vinte linhas consecutivas para frente. Utiliza o argumento numérico para indicar o número de páginas a rolar.

Rolar uma página para trás

Esta função corresponde ao rolamento de vinte linhas consecutivas para trás. Utiliza o argumento numérico para indicar o número de páginas a rolar.

Rolar para frente até encontrar uma cadeia

Esta função causa o rolamento do texto para frente ,

até que a primeira ocorrência de uma cadeia no texto igual a uma cadeia de caracteres fornecidas como argumento ocupe a linha de topo da tela. O cursor é colocado sobre o primeiro caracter da cadeia encontrada. Caso a cadeia não seja encontrada, o rolamento não é executado e o cursor permanece na mesma posição.

Rolar para trás até encontrar uma cadeia

Esta função causa um rolamento do texto para trás até que a primeira ocorrência no texto de uma cadeia igual à cadeia de caracteres fornecida como argumento ocupe a linha de topo da tela. O cursor é posicionado sobre o primeiro caracter da cadeia encontrada. Caso a cadeia não seja encontrada, o rolamento não é executado e o cursor permanece na mesma posição.

Rolar para última página do texto

As últimas vinte linhas do texto são exibidas na tela do terminal de vídeo.

Rolar para primeira página do texto

As primeiras vinte linhas do texto são exibidas na tela do terminal de vídeo.

Descrição das funções de modificação do texto

Substituir o caracter sob o cursor

Esta função é executada toda vez que um caracter é digitado no teclado. O caracter digitado substitui o caracter que está sob o cursor e o cursor se desloca para a próxima coluna à direita, em direção ao fim da linha. Quando o cursor atinge o final da linha, as substituições continuam a ser efetuadas, mas o cursor não é deslocado e o operador é advertido.

Copiar o caracter acima do cursor

Substitui o caracter que está sob o cursor pelo caracter que se encontra na mesma coluna, na linha imediatamente superior à que se encontra o cursor. Após a substituição o cursor se desloca para a próxima coluna à direita, em direção ao final da linha. Quando o cursor atinge o final da linha, a cópia é efetuada, mas o cursor não é deslocado e o operador é advertido. Quando utilizada com o cursor na primeira linha da tela (linha de topo) esta função substitui o caracter sob o cursor por um espaço (caracter branco).

Retirar o caracter sob o cursor

O caracter que está sob o cursor é substituído pelo caracter que lhe está à direita, o mesmo sucedendo aos demais caracteres a partir do cursor até o fim da linha. A última coluna da linha é preenchida por um caracter em branco. A posição do cursor não é modificada.

Inserir um espaço sob o cursor

Um caracter em branco (espaço) é colocado na coluna em que se encontra o cursor. Os caracteres a partir do cursor são deslocados uma coluna para direita. O último caracter da linha é perdido. A posição do cursor não é modificada.

Apagar o resto da linha a partir do cursor

Os caracteres da linha, a partir da posição do cursor, inclusive, até o final da linha são substituídos por caracteres brancos (espaços).

Copiar linha da memória

Por meio desta função, até vinte linhas anteriormente guardadas na memória do editor podem ser transferidas para tela a partir da linha em que se encontra o cursor. A função aceita um argumento numérico que indica o número de linhas a copiar. O número de linhas a copiar deve ser menor ou igual ao número de linhas anteriormente guardado e todas as linhas copiadas devem poder ser exibidas na tela. Caso estas condições não sejam satisfeitas, a função é truncada para o maior número que satisfaça a estas duas condições.

Copiar linha do arquivo auxiliar

Por meio desta função, até vinte linhas do arquivo auxiliar podem ser copiadas para o texto, a partir da linha em que se encontra o cursor, inclusive. A função é truncada caso não possa exibir todas as linhas a serem copiadas ou seja, atingindo o fim do texto auxiliar. À medida que as linhas são copiadas do arquivo auxiliar, a linha seguinte do arquivo auxiliar é exibida na tela. A função aceita argumento numérico indicando o número de linhas a copiar. As linhas a serem copiadas devem poder ser exibidas na tela, ou a execução da função é truncada.

Trocar duas linhas

A linha que contém o cursor é trocada com a linha que está imediatamente abaixo e o cursor desce uma linha. A função não é executada caso o cursor esteja na linha de base da tela.

Retirar a linha sob o cursor

A linha que contém o cursor é retirada do texto. As linhas que estão abaixo dela são roladas para cima e uma nova linha é incluída na linha de base da tela. Esta função aceita um argumento numérico que indica o número de linhas a serem retiradas. Só podem ser retiradas do texto linhas que estejam sendo exibidas na tela. A última linha do texto, indicativa do fim do arquivo, não pode ser retirada. A posição do cursor não é modificada.

Inserir uma linha sob o cursor

Uma linha em branco é inserida sob o cursor. As linhas a partir do cursor são roladas para baixo e a última linha sai da tela. O cursor permanece sobre a linha inserida. Esta função aceita um argumento numérico que indica o número de linhas a serem inseridas. Só podem ser inseridas linhas no limite da tela. Caso um número maior de linhas seja especificado, são inseridas linhas em branco até o fim da tela e a função é truncada.

IV . IMPLEMENTAÇÃO

Definição do Programa

O editor de textos é um programa orientado para a edição interativa de programas escritos em linguagem de alto nível.

Tem como entradas o teclado, um arquivo em disco com o índice dos arquivos catalogados no disco, um arquivo contendo o texto a ser editado e pode acessar a outros arquivos do mesmo disco para copiar linhas. Como saída utiliza o próprio arquivo de texto e a tela da unidade de vídeo do terminal.

Estrutura do arquivo de texto

A determinação de uma estrutura para o arquivo de texto que permitisse o tratamento da inclusão de novas linhas e a alteração do texto na sequência desejada para o usuário sem degradar o desempenho do editor constitui o ponto crucial do projeto.

As técnicas de compressão de dados e de utilização de listas encadeadas não foram utilizadas por acarretarem os seguintes inconvenientes:

1 - A modificação de uma única linha do texto, muito frequente na depuração de programas, implicaria na necessidade de comprimir o texto e expandi-lo em seguida, porque o compilador existente não utiliza textos compactados..

2 - O acesso aleatório a qualquer ponto do texto seria dificultado por não se poder calcular diretamente a posição do texto no arquivo.

3 - Um maior número de acessos a disco seria necessário de modo a fazer a leitura sequencial de todas as linhas entre a posição corrente da tela e a posição para a qual se deseja rolar.

Considerando estes inconvenientes, optamos por utilizar um arquivo em seu formato original de registros de 80 caracteres, mais adequado para a edição de programas.

O problema seguinte consistiu em determinar onde colocar as novas linhas inseridas. Foram consideradas duas alternativas:

1 - Colocar em um arquivo de trabalho;

2 - Colocar no próprio arquivo de texto.

Optamos pela segunda alternativa, pois embora implique na utilização de um espaço em disco maior do que o texto sendo utilizado, provoca menos deslocamentos da cabeça de leitura/escrita do disco, resultando num menor tempo de resposta.

Na impossibilidade de deslocar todas as linhas do arquivo a cada inserção de nova linha, optamos por colocar as linhas inseridas depois do texto e reorganizar o arquivo ao final da sessão de edição de modo que o texto fique na forma sequencial. Os espaços deixados pelas linhas retiradas são cobertos e as linhas inseridas são colocadas em suas posições finais.

Níveis de implementação

Para facilitar o desenvolvimento e depuração do editor e torná-lo mais flexível para posteriores modificações, o editor foi desenvolvido em dois níveis. O nível mais interno implementa um editor de linhas capaz de ler uma linha, escrever uma linha, retirar uma linha e inserir uma linha. Este nível realiza também a leitura inicial do arquivo para determinar seu tamanho e a reorganização do arquivo ao final da sessão de edição.

O nível mais externo, constitui um editor de tela, correspondente ao modelo definido pelo editor.

Módulos

Para facilidade de desenvolvimento, depuração e manutenção, o editor foi desenvolvido segundo as técnicas da programação estruturada, sendo constituído dos seguintes módulos:

- 1 - Módulo principal
- 2 - Módulo de entrada de argumento
- 3 - Módulo de exibição de mensagens
- 4 - Módulo de interpretação da função
- 5 - Módulo de controle da edição
- 6 - Módulo de movimento do cursor
- 7 - Módulo de movimento do texto
- 8 - Módulo de pesquisa da cadeia
- 9 - Módulo de modificação do texto
- 10 - Módulo de inserção retirada
- 11 - Módulo de entrada e saída para disco

Descrição dos módulos

1 - Módulo principal

O módulo principal contém o corpo do editor e serve de interface entre o usuário e o editor. Por ser paramétrico, facilita modificações que visem adaptar o editor a outros equipamentos ou aplicações.

No início da execução, inibe as funções de depuração do terminal. Estas funções utilizam a tela do terminal de vídeo para listar o conteúdo da memória ou as operações executadas pelo terminal. Se não fossem inibidas, o usuário poderia estragar o texto digitando inadvertidamente uma das teclas que acionam estas interrupções.

Em seguida, apaga a tela e verifica a existência de um parâmetro de execução. Se o parâmetro não foi fornecido, assume que o arquivo a ser editado é o de nome FONTE.DAD. Se foi especificado um parâmetro, verifica se ele corresponde a um nome válido de arquivo. Se não corresponde, emite uma mensagem indicativa de erro e termina o processamento.

Consulta o diretório e verifica se existe um arquivo com o nome fornecido. Caso o arquivo não exista, emite uma mensagem e termina o processamento. Caso exista, a entrada correspondente do diretório é utilizada para atualizar as tabelas do arquivo e o arquivo é aberto para processamento.

Calcula o número de linhas que o arquivo pode conter, a partir do número de setores alocados. Lê o arquivo para determinar a última linha do texto e calcula o espaço disponível para inserções. Atualiza a tabela de linhas deslocadas e a tabela do arquivo.

A linha de referência é feita igual a primeira linha do texto e a primeira página do texto é lida para memória, sendo exibida na tela do terminal.

Posiciona o cursor na primeira coluna da primeira linha, desliga o indicador de erro e liga o indicador de edição ativa.

Enquanto o indicador de edição ativa estiver ligado, é executada a parte interativa do programa que consiste em:

- 1 - acender o cursor;
- 2 - ler um caracter do teclado;
- 3 - apagar o cursor;
- 4 - apagar a mensagem de erro
- 5 - verificar se a função é entrada de argumento, se for, ler o argumento;
- 6 - executar a função solicitada;
- 7 - se o cursor sair dos limites do texto, reposicionar o cursor;
- 8 - se ocorreu erro, exibir mensagem de erro.

Quando a função terminar edição é fornecida, o indicador de edição ativa desliga e termina a parte interativa.

O editor então reorganiza o arquivo de texto, caso tenha sido incluída ou retirada alguma linha.

Finalmente, as interrupções de depuração são desinibidas e termina a edição, voltando o controle ao sistema operacional.

2 - Módulo de exibição de mensagens

Coloca o texto da mensagem correspondente ao código do indicador de erro na linha de mensagens da tela do terminal de vídeo. O código de erro zero causa o apagamento da linha de mensagens.

3 - Módulo de entrada de argumento

Executa as funções associadas à entrada de uma cadeia de caracteres como argumento para execução da função subsequente.

Apaga a linha de argumento da tela do terminal de vídeo e coloca um cursor na primeira posição da linha. Os caracteres fornecidos através do teclado substituem o cursor e este avança para direita em direção ao final da linha. Ao chegar ao

final da linha, o cursor para de avançar, mas o caracter continua a ser substituído. O cursor pode ser deslocado sobre a linha de argumento por meio das teclas de avanço e retrocesso utilizadas na edição. Se a tecla de entrada de argumento for digitada durante a entrada do argumento, a linha é apagada e o cursor volta para o início da linha. Quando uma tecla correspondente a uma função de edição for digitada, a entrada de argumento termina. O argumento é copiado na tabela de argumento e seu valor numérico é calculado. As tabelas delta 1 e delta 2, utilizadas nas pesquisas de ocorrência de cadeias de caracteres no texto são também calculadas. O indicador de presença de argumento é ligado.

4 - Módulo de interpretação de função

Para cada caracter fornecido através do teclado, consulta a tabela de funções, retornando com o código de operação interno correspondente ao caracter.

O primeiro algarismo do código de operação interno corresponde ao grupo de operações a que pertence a função e seleciona a rotina a ser chamada.

O segundo algarismo do código de operação interno corresponde a operação específica a ser executada e é utilizado como parâmetro de chamada da rotina.

5 - Módulo de controle da edição

Executa as operações de controle da edição.

a - Indicar arquivo auxiliar

Verifica se a cadeia de argumento é um nome de arquivo válido. Verifica se existe no diretório um arquivo com o nome fornecido. Se ocorrer erro, atribui código de erro ao indicador de erro. Caso contrário, mover endereço para tabela interna de leitura, abrir o arquivo, ler o arquivo até o último registro para determinar seu tamanho, ler primeira linha do arquivo para linha do arquivo auxiliar na tela do terminal de vídeo e ligar indicador de presença de arquivo auxiliar.

b - Posicionar arquivo auxiliar

Calcular o número da linha a ser exibida, utilizando o valor fornecido como argumento. Se for maior do que a última linha ou menor do que a primeira, ligar indicador de erro e posicionar na última ou primeira linha, conforme o caso. Ler linha do arquivo auxiliar para linha do arquivo auxiliar na tela do terminal de vídeo.

c - Guardar linhas do texto

Verificar se o valor do argumento não é maior do que o número de linhas entre o cursor e o fim da página. Se for, atribuir o código correspondente ao indicador de erro e considerar como argumento o número de linhas. Mover as linhas para área de memória interna e armazenar o número de linhas guardadas.

d - Exibir estado do editor

Converter para código ASCII os valores das variáveis a serem exibidas e mover os rótulos e as variáveis para linha de mensagem do terminal de vídeo.

e - Terminar sessão de edição

Desligar o indicador de edição ativa.

6 - Módulo de movimento do cursor

De acordo com o código da operação,

- a - soma 1 à coluna;
- b - subtrai 1 da coluna;
- c - soma 1 à linha
- d - subtrai 1 da linha;
- e - atribui 0 à coluna;
- f - atribui 0 à coluna e soma 1 à linha;
- g - atribui 0 à coluna e à linha;
- h - soma 4 a coluna e subtrai o resto da divisão da coluna por 4;

- i - subtrai 4 da coluna e soma o resto da divisão da coluna por 4.

Verifica se a coluna é maior do que a margem direita ou menor do que a margem da esquerda, se for atribuir valor da margem correspondente à coluna e o código de erro ao indicador ' de erro.

Verificar se a linha é maior do que a margem inferior ou menor do que a margem superior, se for, atribuir valor da margem correspondente à linha e o código de erro ao indicador de erro.

7 - Módulo de movimento do texto

Executa as funções associadas ao rolamento do texto sobre a tela do terminal de vídeo.

Este módulo calcula o valor de uma nova linha de referência (número da linha exibida na primeira linha da tela) de acordo com a função de movimento especificada:

- a - soma o valor do argumento à referência anterior;
- b - subtrai o valor do argumento da referência anterior;
- c - soma o valor do argumento multiplicado pelo tamanho da página em linhas à referência anterior.
- d - subtrai o valor do argumento multiplicado pelo tamanho da página em linhas da referência anterior;
- e - atribui o número da última linha menos o tamanho da página;
- f - atribui o número da primeira linha do texto;
- g - pesquisa cadeia de argumento no texto para frente e, se encontra, atribui número da linha.

h - pesquisa cadeia de argumento no texto para trás e, se encontra, atribui o número da linha.

Se a nova referência é menor do que a primeira linha do texto ou maior do que a última linha do texto, o valor correspondente é atribuído a nova referência e o indicador de erro é atribuído o código de erro correspondente.

Se a nova referência é igual à referência anterior, a função se encerra.

Se a nova referência é a mais do que uma página distante da referência atual, as linhas da tela são gravadas em disco e a nova referência é atribuída a referência atual e novas linhas são lidas para a tela.

Se as referencias estão distantes menos de uma página e d é a diferença entre a referência anterior e a referência atual, então:

Se a diferença é positiva, as d linhas do topo da página são transcritas para o disco (caso tenham sido modificadas), as linhas restantes são roladas para cima e novas linhas são lidas na base da tela. Se a diferença for negativa, a operação inversa é executada.

Este módulo utiliza uma rotina encarregada de:

- a - gravar linhas da tela no disco;
- b - rolar linhas sobre a tela;
- c - ler linhas do disco para a tela.

8 - Módulo de pesquisa à cadeia

Este módulo executa as funções associadas a pesquisa da primeira ocorrência de uma cadeia de caracteres no corpo do texto. A pesquisa pode ser para frente, em direção ao fim do texto, ou para trás, em direção ao início do texto. A função retorna com o número da linha em que ocorre a cadeia e a posição (número da coluna) em que esta se inicia.

Para permitir a execução mais rápida da pesquisa, o módulo utiliza um algoritmo de procura rápida, em que as comparações são feitas a partir do último caracter da cadeia permitindo uma significativa redução no número de comparações necessárias.

9 - Módulo de modificação do Texto

Este módulo executa as funções relacionadas com a modificação do conteúdo das linhas do texto.

a - Substituir o caracter sob o cursor

Compara o caracter fornecido com o que se encontra sob o cursor. Se for diferente, substitui o caracter e marca a linha como modificada. O cursor é deslocado para direita.

b - Copiar o caracter acima do cursor

Copia o caracter acima do cursor para posição sob o cursor. Se o cursor estiver na primeira linha da tela, assume um caracter branco (espaço) a ser copiado. O cursor é deslocado para direita e a linha marcada como modificada.

c - Retirar caracter sob o cursor

Desloca os caracteres à direita do cursor para a esquerda e insere um branco na última posição da linha. A linha é marcada como modificada.

d - Apagar resto da linha a partir do cursor

Os caracteres do cursor até o fim da linha são substituídos por brancos e a linha é marcada como modificada.

e - Copiar linha da memória

Verifica se o número de linhas a copiar, valor do argumento, é maior do que o número de linhas armazenado na memória, se for, trunca o argumento. Verifica se o número de linhas a copiar é maior do que o número de linhas do cursor à

base da tela e ao final do arquivo. Se for, trunca o argumento .
Copia as linhas.

f - Copiar linhas do arquivo auxiliar

Verifica se foi definido um arquivo auxiliar e se existem linhas em número suficiente da posição exibida ao fim do arquivo auxiliar. Se não existir, trunca o argumento. Verifica se existe espaço na tela até o final do texto. Se não existir trunca o argumento. Copia linhas do arquivo auxiliar. Exibe próxima linha do arquivo auxiliar.

g - Trocar duas linhas

Verificar se o cursor está sobre a linha de base da tela ou sobre a última linha do texto. Caso esteja, a função não é executada. Caso contrário, as linhas são trocadas. Ambas as linhas são marcadas como modificadas.

10 - Módulo de retirar e inserir linhas

Este módulo executa as funções de retirar e inserir linhas do texto.

Inicialmente, o módulo verifica se há espaço na tabela de linhas deslocadas para realizar a operação, caso não exista, a função não é executada e o código de erro correspondente é atribuído ao indicador de erro.

Se há espaço, as funções são executadas:

a - Retirar uma linha

Verifica se as linhas a serem retiradas estão na tela, se não estão, o argumento é truncado e o código de erro correspondente é atribuído ao indicador de erro.

Verifica se está sendo retirada a última linha, se estiver, o argumento é truncado e o código de erro é atribuído ao indicador. Solicita a retirada das linhas à rotina de edição

de linha. Copia as linhas retiradas para a tabela de linhas guardadas. Desloca as linhas abaixo das linhas retiradas para cima. Lê linhas para base da tela de modo a completar a página.

b - Inserir linhas

Verifica se as linhas a serem inseridas cabem na tela, se não cabem, trunca o argumento e atribui o código de erro ao indicador de erro. Verifica se há espaço no disco para colocar as linhas inseridas, se não houver, trunca o argumento e atribui o código de erro ao indicador de erro.

Copia para o disco as linhas que vão sair da tela e foram modificadas.

Desloca as linhas abaixo do cursor para abrir espaço para as linhas inseridas. Solicita a inclusão de linhas à rotina de E/S em disco. Limpa a área inserida na tela do terminal de vídeo.

11 - Entrada e Saída em Disco

Este módulo implementa um editor de linhas, que é utilizado pelo editor de tela. Executa as seguintes funções:

- a - Inicializa a leitura do arquivo, determina o seu tamanho efetivo, lendo os registros até a ocorrência de fim de arquivo, e inicializa a tabela de linhas deslocadas.
- b - Lê uma linha do texto a partir do número da linha em relação ao início do texto.
- c - Grava uma linha do texto a partir do endereço da linha na memória e do número da linha em relação ao início do texto.
- d - Retira uma linha do texto utilizando o número da linha em relação ao início do texto como parâmetro.

- e - Insere uma linha utilizando o número de linha a partir do início do texto como argumento.

- f - Reorganiza o arquivo de disco, fazendo com que os endereços físicos das linhas no disco estejam na mesma ordem que os números das linhas em relação ao início do texto.

Tabelas

As principais áreas de dados do editor estão organizadas em tabela, de acordo com sua finalidade.

- 1 - Tabela de constantes externas
- 2 - Tabela de argumento
- 3 - Tabela de mensagens
- 4 - Tabela de funções
- 5 - Tabela de variáveis de estado
- 6 - Tabela do cursor
- 7 - Tabela da página
- 8 - Tabela de linhas modificadas
- 9 - Tabela de linhas deslocadas
- 10 - Tabela de linhas guardadas
- 11 - Tabela do arquivo de texto

Descrição das Tabelas

1 - Tabela de constantes externas

A tabela de constantes externas contém os valores de várias externas ao editor de textos.

constante

endereço do parâmetro de execução

endereço da inibição de depuração

endereço do mapeamento do vídeo

2 - Tabela de argumento

A tabela de argumento contém as variáveis associadas à utilização de uma cadeia de caracteres como argumento para execução de uma função subsequente. Ocupa 292 bytes de memória.

<u>variável</u>	<u>significado</u>	<u>valor</u>
valor	valor numérico atribuído ao argumento	0001-9999
tamanho	tamanho da cadeia de caracteres fornecida como argumento	00-80
cadeia	vetor de 80 caracteres ' contendo os caracteres ' fornecidos como argumento	000-127
delta 2	vetor de 80 caracteres contendo valores compreendidos entre 1 e o tamanho da cadeia, utilizado para procura de ocorrências da cadeia no texto	01-80

2 - Tabela de argumento (continuação)

<u>variável</u>	<u>significado</u>	<u>valor</u>
delta 1	vetor de 127 caracteres contendo a posição de cadeia em que ocorre cada caracter. Valores entre 1 e o tamanho da cadeia	01-80

3 - Tabela de mensagens

A tabela de mensagens contém os textos das mensagens emitidas pelo editor. A tabela ocupa 200 bytes da memória e consta de 10 mensagens de 20 caracteres.

<u>código</u>	<u>mensagem</u>
0	(em branco)
1	.NOME INVALIDO
2	.ARQUIVO INEXISTENTE
3	.ERRO NO DISCO
4	.ESPAÇO ESGOTADO
5	.CURSOR RETIDO
6	.LIMITE DO TEXTO
7	.FUNÇÃO INEXISTENTE
8	.EXECUÇÃO INCOMPLETA
9	.CADEIA INEXISTENTE

4 - Tabela de funções

A tabela de funções associa a cada caracter digitado no teclado um código interno dorrespondente à função a ser executada pelo editor. A utilização desta tabela facilita a implementação do módulo de interpretação de função, diminui o tempo necessário a analisar a função a ser executada e fornece grande flexibilidade para modificar as funções associadas à teclas ou adaptar o editor a outro tipo de teclado. A tabela ocupa 256 bytes e fornece os seguintes códigos:

<u>código</u>	<u>função</u>
00	Inexistente
11	Entrar com argumento
12	Indicar arquivo auxiliar
13	Posicionar arquivo auxiliar
14	Guardar linhas do texto
15	Exibir estado do editor
16	Terminar sessão de edição
21	Mover cursor para direita
22	Mover cursor para esquerda
23	Mover cursor para baixo
24	Mover cursor para cima
25	Mover cursor para início da linha corrente
26	Mover cursor para início da linha seguinte
27	Mover cursor para início da página
28	Mover cursor para tabulação à direita
29	Mover cursor para tabulação à esquerda
31	Rolar linha para frente
32	Rolar linha para trás
33	Rolar uma página para frente
34	Rolar uma página para trás
35	Rolar para frente até encontrar cadeia
36	Rolar para trás até encontrar cadeia
37	Rolar para última página do texto
38	Rolar para primeira página do texto

4 - Tabela de funções (continuação)

<u>código</u>	<u>função</u>
41	Substituir o caracter sob o cursor
42	Copiar o caracter acima do cursor
43	Retirar o caracter sob o cursor
44	Inserir um espaço sob o cursor
45	Apagar o resto da linha
46	Copiar linha da memória
47	Copiar linha do arquivo auxiliar
48	Trocar duas linhas
51	Retirar a linha sob o cursor
52	Inserir uma linha sob o cursor

5 - Tabela de variáveis de estado

Contém as variáveis indicativas do estado do editor.

<u>variável</u>	<u>significado</u>	<u>valor</u>
editor\$ativo	indica atividade do editor	
	inicial	1
	fim de sessão	0
indicador\$erro	não ocorreu erro	0
	ocorreu erro	≠ 0
tem\$parâmetro	tem	1
	não tem	0
tem\$argumento	tem	1
	não tem	0
tem\$arquivo\$auxiliar	tem	1
	não tem	0

6 - Tabela do cursor

A tabela do cursor contém as variáveis associadas ao posicionamento do cursor sobre a tela do editor. Ocupa 16 bytes de memória e é formada por 8 variáveis de 2 bytes cada.

<u>variável</u>	<u>significado</u>	<u>valor</u>
cursor	indica a posição da tela onde está o cursor	0000-1599
linha	indica em que linha da tela está o cursor	00-19
coluna	indica em que coluna da tela está o cursor	00-79
margem superior	constante indicativa do menor valor que pode assumir a variável linha	00
margem inferior	constante indicativa do maior valor que pode assumir a variável linha	19
margem esquerda	constante indicativa do menor valor que pode assumir a variável coluna	00
margem direita	constante indicativa do maior valor que pode assumir a variável coluna	79
reservado	não utilizado	00

7 - Tabela de página

A tabela de página consta de uma área de 1290 caracteres e contém as 24 linhas de 80 caracteres que são apresentadas na tela do terminal de vídeo.

8 - Tabela de linhas modificadas

A tabela de linhas modificadas consta de uma tabela de 20 entradas, com 1 byte cada uma, correspondendo a cada linha do texto exibida na tela. Tem por finalidade possibilitar a minimização do número de acessos realizados ao arquivo em disco. Quando uma linha do texto é lida para a tela, a entrada correspondente da tabela é colocada em '0'. Se qualquer caracter da linha for substituído por um caracter diferente, a entrada da tabela é colocada em '1'. No caso de rolamento das linhas na tela, as entradas da tabela de linhas modificadas são deslocadas de modo a manter a correspondência entre as linhas e os indicadores de modificação. Quando uma linha é rolada para fora da tela, se estiver modificada, é copiada de volta para o disco.

9 - Tabela de linhas deslocadas

A tabela de linhas deslocadas consta de dois vetores de 64 entradas, com dois bytes para cada entrada, fazendo um total de 256 bytes. Tem por finalidade indicar em que registro do disco se encontra determinada linha do texto. A cada seção contínua do arquivo, correspondem duas entradas na tabela, uma em cada vetor. No primeiro vetor temos a indicação da última linha de cada seção. No segundo vetor, a entrada correspondente contém uma variável de deslocamento a ser adicionada ao número da linha do texto, de forma a localizar sua posição no arquivo em disco.

10 - Tabela de linhas guardadas

Consta de uma área de 1600 bytes, onde podem ser guardadas até 20 linhas da tela, para serem copiadas para outro ponto do texto. As linhas retiradas são guardadas também nesta tabela.

11 - Tabela do arquivo de texto

Esta tabela contém as variáveis associadas ao arquivo de texto.

<u>variável</u>	<u>significado</u>
tamanho\$arquivo	tamanho do arquivo, em setores
tamanho\$texto	tamanho de texto, em linhas
primeira\$insertão	endereço, no disco, da primeira linha inserida
próxima\$insertão	endereço da linha, no disco, reservada para próxima inserção
número\$seções	número de seções contínuas em que o arquivo está dividido

V . CONCLUSÃO

Avaliação

A primeira versão do editor foi colocada em operação em janeiro de 1980 e foi utilizada durante um ano por profissionais do NCE e por alunos do curso de informática. Críticas e sugestões destes primeiros usuários permitiram a eliminação de alguns inconvenientes menores da versão original.

De um modo geral, o desempenho do editor para o desenvolvimento de programas foi considerado bastante satisfatório. Ocasionalmente, o editor foi também utilizado para a criação de textos livres e o rendimento foi considerado aceitável.

Para analisar o tempo de resposta consideramos os casos:

1 - Resposta a digitação de teclas

O editor responde à digitação de uma tecla apagando o cursor da tela. Este apagamento é instantâneo e informa ao operador que o comando foi recebido.

2 - Movimento do cursor e substituição de caracteres

Do ponto de vista do usuário estas funções são executadas instantaneamente, podendo o editor ser utilizado mesmo pelos datilógrafos mais rápidos.

3 - Rolamento do texto

Estas funções são executadas em um tempo plenamente satisfatório, iniciando-se a troca da primeira linha em menos de um segundo para a maioria dos casos. Quando muitas linhas modificadas são roladas para fora da tela um pequeno retardo ocasionado pela necessidade de gravar as linhas no disco pode ser observado.

4 - Retirada e inserção de novas linhas

Esta função é também bastante rápida, notando-se uma pequena demora para inserção de muitas linhas em um espaço onde foram feitas muitas modificações.

5 - Pesquisa de uma cadeia no texto

A execução desta função, apesar da utilização de um algoritmo de comparação muito eficiente, pode resultar em demoras superiores a 15 segundos.

6 - Inicialização/reorganização do texto

A leitura inicial do arquivo, dependendo de seu tamanho, pode tomar alguns minutos. Caso linhas tenham sido retiradas ou inseridas, a reorganização também pode ser demorada. Estas demoras, embora inconvenientes, não interferem com a operação interativa, uma vez que ocorrem antes e depois da mesma.

Extensões

Como o programa utiliza pouco mais de 8K bytes da memória do terminal, acreditamos que possa ser estendido de modo a tornar-se mais poderoso, mais rápido e mais genérico.

1 - Função de substituição de cadeia

Criar uma função que permita a substituição de uma cadeia encontrada por outra fornecida anteriormente.

2 - Criação da função inversa

Criar uma função que reverta a última operação executada, restabelecendo o texto na forma anterior.

3 - Introdução de parâmetros

Introduzir parâmetros para estabelecer novas margens e posições tabuladas definidas pelo operador no início da sessão de edição.

4 - Limpar o texto

Criar uma opção para limpar um número de linhas a no início da sessão de edição.

5 - Numeração de linhas

Criar uma função para identificar e numerar sequencialmente as linhas no fim da sessão de edição.

6 - Instruções para utilização

Criar um arquivo com instruções para utilização do editor. Este arquivo seria lido para tela no início da edição, por opção do operador.

7 - Utilização de áreas de leitura

Utilizar mais áreas de leitura para acelerar a leitura inicial do arquivo e sua reorganização ao final da edição.

VI. BIBLIOGRAFIA

1. BRATMAN, M. - Program Composition and Editing with an on-line display, Fall Joint Computer Conference, 1968.
2. DEUTSCH, L. - An on-line editor. CACM 10(12):793-799, 1967.
3. FRANÇA, Paulo Mario Bianchi e PAIVA, Paulo Fernando Melo - Sistema Preparador de Programas. NCE/UFRJ - 1975.
4. FRASER, C. - A Compact, Portable CRT-based Text Editor. Software_Practice&Experience 9:121-125, 1979.
5. FAJMAN, R. - WILBUR: An interactive Text Editing and Remote Job Entry System. CACM 16(5):314-321, 1973.
6. HANSEN, W. - User engineering principles for interactive systems. Fall Joint Computer Conference, 1968.
7. IRONS, E. - A CRT editing system. CACM 15(1):16-20.
8. MELO, Paulo Cesar Moraes - Uma linguagem de alto nível para microcomputadores - PLTI. NCE/UFRJ.
9. MILLER, R. - Response time in man-computer conversational transactions. Fall Joint Computer Conference, 1968.
10. PEREIRA, Oswaldo Serra Alves - Editor de Textos para Terminal Inteligente. COPPE/UFRJ 1978.
11. RODRIGUES, Guilherme Chagas - Sistema Operacional para Terminal Inteligente. NCE/UFRJ.
12. SCHNEIDER, B. - SITAR: An Interactive Text Processing System for Small Computer. CACM 20(6):495-499, 1977.
13. VAN DAN, A. - On-line text editing: A Survey. Computing Surveys, 3(3):93-114, 1971.

14. BOYER, R. - A Fast String Searching Algorithm.
CACM 20(10): 762-772, 1977.
15. AHO, A. - Fast pattern matching: An aid to bibliographic
search. CACM 18(6):333-340, 1975.
16. MACLEOD, I. - Design and Implementation of a Display
Oriented Text Editor.
Software_Practice&Experience, 7:771-778, 1977.