EPC 6 - ARGUMENTOS E MANIPULAÇÃO DE ARQUIVOS

Monitor: Weslley de Aquino Ferreira

1) Escreva um programa que faça uso dos parâmetros argv e argc. O programa deve receber da linha de comando o dia, mês e ano correntes, e imprimir a data em formato apropriado. Veja o exemplo, supondo que o executável se chame data:

   data 19 04 99

   O programa deve imprimir:

   19 de abril de 1999

2) Faça um programa que leia um ficheiro de utentes de um parque de estacionamento, contendo, em cada linha, o código, a categoria e o nome de cada utente, por esta ordem. Estes dados devem ser carregados numa matriz de estruturas. O programa deve escrever num ficheiro a base de dados de utentes ordenada por nome ou por código. A execução do programa é feita com passagem de três argumentos (usando argc e argv de main()):

   programa entrada saída campo

   Em que campo indica o campo pelo qual a ordenação é feita, podendo tomar os valores nome ou código.

   Durante a ordenação a matriz de utentes permanece inalterada, para o que se deve utilizar uma matriz adicional de ponteiros para a estrutura de utente.

   Exemplo 1:

   ordena entrada.txt saida.txt codigo

   Este comando lê o ficheiro entrada.txt, carrega-o para a matriz de utentes, ordena-a pelo código (indirectamente, não esquecer), e escreve o resultado no

ficheiro saida.txt.

Exemplo 2:

ordena entrada.txt saida.txt nome

Este comando lê o ficheiro entrada.txt, carrega-o para a matriz de utentes, ordena-a pelo nome do utente, e escreve o resultado no ficheiro saida.txt.

3) Escreva um programa que leia vários números inteiros e grave-os num arquivo texto. O programa termina a leitura dos números quando o número zero for digitado. Na sequência o programa imprime na tela os números ***gravados no arquivo***.

4) Faça um programa que leia (do teclado) um cadastro de 10 alunos, indicando o nome, nota1, nota2. Calcule a média aritmética simples dos 10 alunos e depois escreva em um arquivo texto os dados de cada aluno: nome, nota1, nota2 e média. Lembre-se de que as notas e média deverão ser apresentadas como valores que possuem até 2 casas após a vírgula.

5) Faça um programa de criptografia de dados, ou seja, um programa capaz de ler um arquivo texto, codificar este arquivo através de alguma técnica de alteração do código ASCII (exemplo: somar 1 ao valor ASCII de cada caracter), e escrever em disco o arquivo codificado. Crie um outro programa que descriptografe um arquivo criado pelo programa de criptografia, realizando a operação inversa: ler o arquivo do disco, decodificar e escrever o novo arquivo em disco descriptografado. Lembre-se que para que seja possível criptografar/descriptografar um arquivo a função de codificação deve possuir uma função inversa.

6) Faça um que abra um arquivo HTML e elimine todas as “tags” do texto, ou seja, o programa deve gerar um novo arquivo em disco que elimine todas as tags HTML que são caracterizadas por comandos entre “<” e “>”. Veja abaixo um exemplo de um texto em HTML e do texto que deverá ser gerado pelo programa após eliminar as tags HTML.

```
<HTML>
<HEAD>
<TITLE>Minha Pagina Web</TITLE>
</HEAD>
<BODY>
Meu Texto<BR>
Minha Imagem<IMG SRC=”img.jpg”>
<A HREF=”pag1.html”>Meu Link</A>
</BODY>
</HTML>
```

```
Minha Pagina Web


Meu Texto
Minha Imagem
Meu Link
```

7) Faça um editor de textos, inspirado no NotePad (Bloco de Notas) ou WordPad/Word, que permite ao usuário inicialmente ler um texto de um arquivo texto armazenado em disco, armazene este texto em um vetor de strings em memória, onde cada linha do texto terá um número indicando a linha correspondente. Uma vez lido o arquivo, o programa deve ler comandos do usuário, que poderá ser um dos seguintes comandos:

   Listar (opção 1: indicar o intervalo de linhas que deseja exibir na tela, linha inicial até linha final);

   Editar (opção 2: indicar qual a linha deseja editar, mostrar o seu conteúdo atual, ler um novo conteúdo e substituir o conteúdo da linha antiga pelo novo conteúdo);

   Inserir (opção 3: indicar depois de qual linha desejo inserir uma nova linha de texto, “abrindo espaço” após esta linha e inserindo um novo texto);

   Apagar (opção 4: indicar qual linha deseja apagar, exibir seu conteúdo, confirmar a remoção e remover esta linha do texto);

   Abandonar (opção 5: sair do programa, sem salvar o texto editado, onde é pedida uma confirmação do usuário sobre a execução desta opção);

Sair (opção 6: sair do programa, onde o texto será salvo em um arquivo em disco sobrescrevendo o arquivo original, sendo pedida uma confirmação do usuário sobre a execução desta opção).

O programa no final deve portanto ter a capacidade de ler um arquivo de texto do disco, editar (listar, incluir, excluir, modificar linhas de texto) e salvar em disco o texto novo que foi editado, executando as operações usuais de um editor de textos simples.