

SSC0501 - Introdução à Ciência da Computação I (Teórica)

Professor responsável: *Fernando Santos Osório*

Semestre: 2010/1

Bolsista PAE: Gustavo Pessin

Monitor: Matheus Lin (Seg. e Sexta 18h -19h)

Horário: Seg. 21h-22h40 e Terça 19h-20h40

E-mail: fosorio .at. icmc .dot. USP .dot. br

fosorio .at. gmail .dot. com

pessin .at. gmail .dot. com

matheus.lin .at. gmail .dot. com

Web: <http://www.icmc.usp.br/~fosorio/>

LISTA DE EXERCÍCIOS - Nro. 07

[Programas com Arquivos]

1. Faça um programa que crie um arquivo TEXTO em disco, com o nome “dados.txt”, e escreva neste arquivo em disco uma contagem que vá de 1 até 100, com um número em cada linha. Abra este arquivo em um editor de textos, como por exemplo o Notepad ou o Wordpad do Windows.
2. Faça um programa que crie um arquivo BINÁRIO em disco, com o nome “dados.bin”, e escreva neste arquivo em disco uma contagem que vá de 1 até 100, com um número em cada linha. Abra este arquivo em um editor de textos e observe como ficou o seu conteúdo (ilegível!).
3. Faça 2 programas, um que leia o arquivo TEXTO criado no exercício 1 e outro que leia o arquivo BINÁRIO criado no exercício 2. Exibir na tela os dados lidos dos respectivos arquivos.
4. Faça um programa que leia um arquivo texto do disco, lendo linha a linha, e exibindo cada uma das linhas numeradas na tela. A idéia é podermos pegar um arquivo um arquivo texto qualquer (pode ser inclusive o arquivo do programa fonte – arquivo “.c” – de um destes exercícios) e mostrar na tela com as linhas numeradas. As primeiras linhas do arquivo iriam ser exibidas na tela da seguinte forma:

```
1: #include <stdio.h>
2: #include <conio.h>
3:
4: FILE *ArqEntrada;
5: ...
```

5. Faça um programa que imite um editor de textos. Inicialmente você irá ler os dados digitados pelo usuário (linhas de texto) e criar um vetor em memória onde serão armazenados os textos fornecidos pelo usuário (texto de 1 até no máximo 50 linhas). O usuário vai escrever o seu texto, terminando por uma linha onde ele irá escrever apenas a palavra ‘FIM’, o que determina que ele não deseja mais digitar linhas de texto. Sendo assim, o texto final pode ter um número variável de linhas, entre 1 e 50. Salvar o conteúdo armazenado na memória neste vetor, em um arquivo texto em disco. Sugestão: use o comando “gets” ou “fgets” para ler as linhas de texto digitadas pelo usuário.
6. 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.

7. Faça um programa que leia os dados gerados pelo programa anterior do exercício 6 e mostre na tela as informações referentes aos alunos, calculando também a média geral da turma. LEMBRETE: É bem mais fácil armazenar informações em arquivos do tipo texto, colocando um dado em cada linha (“imitando” a entrada de dados do teclado: dado <enter>, dado <enter>, dado <enter> ...).
8. 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. Exemplo de código de criptografia:

```
tamanho = strlen(linha);
for (i=0; i < tamanho; i++)
    linha_cripto[i] = linha[i] + 1; /* Soma 1 ao código de cada caracter da string */
linha_cripto[tamanho]='\0';      /* Adiciona a marca de fim de string */
```

Desafios: Você consegue desenvolver uma função de criptografia/descriptografia que seja mais sofisticada e menos óbvia do que esta? Você conseguiria adaptar este programa para criptografar qualquer tipo de arquivo, seja ele um arquivo texto ou binário?

9. Faça um programa que peça para o usuário entrar um texto que deseja procurar (uma palavra) e que em seguida leia um arquivo texto do disco e procure por esta palavra no arquivo. Caso seja encontrada a palavra digitada pelo usuário em alguma das linhas do arquivo texto lido do disco, o programa deverá exibir na tela o número da linha do arquivo onde encontrou esta ocorrência da palavra, bem como o conteúdo da linha onde se encontra a palavra procurada. Dica: use o comando “strstr” para procurar a palavra.
10. Faça um programa para armazenar compromissos em uma agenda (Ver exercício da Lista anterior). A agenda deve possuir uma matriz que armazene a data e hora de um compromisso, assim como um texto que descreve o compromisso agendado. Cada compromisso deve possuir um mês, dia e hora (data) e mais um texto associado a este de até 30 caracteres (string) que o descreve. Inicialmente o usuário deve poder indicar a data e hora de seus compromissos, entrando um a um, e depois indicando se deseja informar mais um compromisso, até que ele responda que não deseja mais fornecer compromissos para sua agenda. O programa deve então gravar em um arquivo todos os compromissos agendados, indicando: dia, mês, hora e informações sobre o compromisso agendado. Dicas: (1) Use uma matriz do tipo Agenda [mes][dia][hora] que armazene textos, (2) Não use espaço em branco em textos se for usar o comando fscanf com “%s”, podendo usar em seu lugar o comando fgets.
11. Faça um programa que leia os dados armazenados em um arquivo que foram gerados no programa anterior, passando os dados do arquivo para uma matriz em memória. Exibir na tela o conteúdo que está armazenado em memória.
12. Faça um programa que leia os dados (números de ponto flutuante) de uma planilha criada no Excel e armazene estes dados em uma Tabela em memória (tamanho máximo da tabela 10x10). Cuidado com os caracteres de “espaço”, “vírgula”, “ponto”.

Exercícios – QUESTÃO DESAFIO:

1. Questão Desafio: 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
```

2. Questão Desafio: Faça um editor de textos, inspirado no NotePad (Bloco de Notas) ou WordPad/Word, que permita 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 (laço), 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, modifica linhas de texto) e salvar em disco o texto novo que foi editado, executando as operações usuais de um editor de textos simples.