

Exercícios

1º Exercício

Elabore um programa, criar.c que crie um arquivo texto cujo nome deve ser passado ao programa a partir da linha de comando. O arquivo deve ser aberto de modo que seja possível acrescentar mais texto a cada nova execução do programa. Cada sessão de inserção deve terminar quando for digitado o caractere "*", caractere este que não deve ser gravado no arquivo.

Utilizar as funções getchar() e putc(...., ...) para ler e gravar os caracteres no arquivo.

2º Exercício

Elabore um programa, listar.c que liste na tela o arquivo criado no exercício anterior. O nome do arquivo também deve ser passado ao programa a partir da linha de comando. Utilizar as funções getc(...) para ler do disco os caracteres do arquivo e a função putc(..., stdout) para escrever os caracteres na tela.

3º Exercício

Escreva um programa para criar um arquivo em disco, "BD.DAT", utilizando a função fprintf(...., ..., ...). O programa deve declarar 3 variáveis inteiras: nvoo, nlug, nmarc (número do vôo, total de lugares por vôo e número de lugares marcado por vôo). Crie um laço infinito que peça ao usuário que digite valores para estas três variáveis e as grave em disco após serem lidas. O ciclo deve terminar quando o usuário digitar 0, para o número do vôo.

4º Exercício

Escreva um programa para ler o arquivo "BD.DAT", criado no exercício anterior, listando os dados na tela. Utilizar a função fscanf(..., ..., ...) para ler o arquivo. O programa deve solicitar ao usuário que digite um número de vôo e, se esse número de vôo for localizado, devem ser exibidos seus dados na tela.

5º Exercício

Crie um programa denominado linhas.c, que declara uma variável global como vetor de ponteiros, char *lin[DIM] (onde DIM é uma constante previamente definida), lê do disco um arquivo, dados.txt, com várias linhas de texto e aloca cada linha lida em memória dinamicamente, referenciada por um ponteiro do vetor. O programa deve ainda conter uma função para listar as linhas de texto depois de alocadas na memória.

6º Exercício

Crie um programa em linguagem C, INFO.C, que possua os seguintes requisitos:

- O programa deve possuir um Menu de Opções para controlar a execução das várias operações sobre um arquivo binário, designado por ALUNOS.DAT, que deverá guardar informação básica relativa aos alunos de uma escola.
- A informação referente a cada aluno está agrupada numa estrutura definida como:

```
typedef struct{
```

```
    unsigned int num;  
    unsigned int ano;  
    char nome[15];  
    char sobrenome[15];  
    unsigned long telefone;  
}FICHA;
```

- As opções do menu serão as seguintes: Inserir registros, Listar arquivo na tela, Pesquisar seqüencialmente por um nome, Pesquisar um registro por posição, Alterar um registro, Eliminar um registro. Para tal devem ser criadas as funções adequadas a cada uma dessas opções.

7º Exercício

Crie um programa ESTRUT.C, que lê do disco os registros do arquivo ALUNOS.DAT, criado no exercício anterior e aloca esses registros em memória dinâmica, referenciado cada um deles por um ponteiro do vetor definido como FICHA *reg[DIM].

O programa deve incluir uma função para listar os registros depois de lidos para memória. O tipo de estrutura da informação correspondente a cada aluno deve ser definido tal como foi indicado no exercício anterior.

8º Exercício

Utilização do algoritmo da Bolha (BubbleSort)

Crie um programa, ORDFICH.C que faça a leitura do arquivo ALUNOS.DAT, criado no exercício 9), para um vetor de estruturas reg[DIM] do tipo FICHA. O programa deve incluir funções de ordenação simples pelos campos num e nome. O vetor deve ser listado após leitura dos registros para memória e após a ordenação.

9º Exercício

Altere o exercício anterior, fazendo a leitura do arquivo para memória dinâmica, em que as estruturas do tipo FICHA são referenciadas por ponteiros do vetor *reg[DIM]. Reimplemente as ordenações do exercício 8) para esta situação.