Departamento de Ciência da Computação – IME-USP

MAC 115 – Introdução à Computação para Ciências Exatas e Tecnologia

Escola Politécnica – Primeiro Semestre de 1998

Terceiro Exercício-Programa Data de entrega: até 22 de maio de 1998

Turmas: Civil A, Química, Produção, Mecatrônica e Naval

VIDEO LOCADEIRA

Uma nova rede de vídeo locadeiras, a famosa PoliFitas, que consiste por enquanto de apenas uma loja, resolveu contratar os alunos da Escola Politécnica para escrever um programa que facilite o controle do estoque de fitas da loja.

O seu programa deve manter as seguintes informações sobre filmes:

Título do filme;
Ator(es) principais;
Diretor;
Gênero;
Ano;
Quantidade total de fitas deste filme que a loja tem;
Quantidade de fitas deste filme emprestadas;
Diária da fita.

O usuário pode requerer qualquer uma das seguintes operações:

- ler um arquivo com as informações de filmes já cadastrados;
- listar os títulos de todos os filmes cadastrados;
- buscar um filme a partir do título, ator, diretor ou gênero do filme (ou parte desta informação);
- inserir informações de um novo filme;
- remover um filme do cadastro;
- alterar as informações sobre um filme cadastrado.

Para entender melhor como o seu programa deve funcionar, instalamos na rede do CCE um arquivo executável de exemplo. O arquivo se chama `video.exe` e está instalado no diretório `i:\mac115` da rede do CCE. Também neste diretório, você vai encontrar um arquivo com dados, `video.txt`, que você pode usar como cadastro inicial.

Execute `video` e opte por ler um arquivo de dados. Então escolha como arquivo o `video.txt`. A partir daí, execute o que você quiser e veja o resultado.

O seu programa deve funcionar de forma semelhante. Você deve implementar as opções que você vê quando executa o nosso exemplo.

As funções que fazem leitura ou gravação das informações de um arquivo ou do teclado devem ser escritas de forma consistente. Para ajudar, instalamos, no mesmo diretório mencionado acima, o arquivo `alunos.c`, com um programa em C que manipula dados de alunos (nome do aluno, número de créditos e média). Baseie-se neste programa para escrever as suas funções de leitura, gravação, edição e inserção.

O seu programa deve, obrigatoriamente, conter pelo menos as seguintes funções. Caso você julgue conveniente, altere o tipo retornado pelas funções que, conforme descrito abaixo, estão retornando `void`.

1. ```
   int leitura (char *titulo[], char *ator[], char *diretor[],
                char *genero[], int ano[], int quant[],
                int quantemp[], float preco[]);
   /* Lê o nome de um arquivo do teclado e, caso o arquivo exista em disco, lê
      os dados nele gravados e os guarda nos vetores titulo, ator, diretor,
      genero, ano, quant, quantemp, preço. Retorna o número de registros (ou seja,
      cadastros de filmes) lidos. Caso o arquivo não exista, o programa imprime
      uma mensagem e retorna 0 /*
   ```
2. ```
   void gravacao (int n, char *titulo[], char *ator[], char *diretor[],
                  char *genero[], int ano[], int quant[], int quantemp[],
                  float preco[]);
   /* Lê o nome de um arquivo do teclado e grava em disco um arquivo com as
      informações que estão na memória, nas n primeiras posições dos vetores
      titulo, ator, diretor, genero, ano, quant, quantemp e preço. */
   ```
3. ```
   void lista(int i, char *titulo[], char *ator[],
              char *diretor[], char *genero[],
              int ano[], int quant[], int quantemp[], float preco[]);
   /* Lista as informações guardadas na posição i dos vetores titulo,
      ator, diretor, genero, ano, quant, quantemp, preço. */
   ```
4. ```
   void insercao (int n, char *titulo[], char *ator[], char *diretor[],
                  char *genero[], int ano[], int quant[], int quantemp[],
                  float preco[]);
   /* Insere as informações de um novo filme na posição n dos vetores. */
   ```
5. ```
   void editar(int i, char *titulo[], char *ator[],
               char *diretor[], char *genero[],
               int ano[], int quant[], int quantemp[], float preco[]);
   /* Mostra as informações guardadas na posição i dos vetores titulo, ator,
      diretor, genero, ano, quant, quantemp, preço, dando ao usuário a
      possibilidade de alterar qualquer uma destas informações. */
   ```
6. ```
   void remover(int i, int n, char *titulo[], char *ator[],
                char *diretor[], char *genero[],
                int ano[], int quant[], int quantemp[], float preco[]);
   /* Remove as informações guardadas na posição i dos vetores titulo, ator,
      diretor, genero, ano, quant, quantemp, preço. */
   ```
7. ```
   void busca(int *n, char *titulo[], char *ator[],
              char *diretor[], char *genero[],
              int ano[], int quant[], int quantemp[], float preco[]);
   /* Lê uma palavra do teclado e busca esta palavra nas n primeiras
      posições dos vetores titulo, ator, diretor, genero. Mostra um a um
      os registros contendo a palavra, e, para cada um, permite a
      opção de editar o tal registro. Note que, durante a edição, um
      registro pode ser removido e, portanto, *n pode ser alterado.*/
   ```
8. ```
   void listatitulo(int *n, char *titulo[], char *ator[],
                    char *diretor[], char *genero[],
                    int ano[], int quant[], int quantemp[], float preco[]);
   /* Lista os títulos de todos os filmes no cadastro, indexados, de forma que o
      usuário possa pedir a informação completa de alguns dos filmes. */
   ```

Observações Gerais

- Não use variáveis globais.
- Sinta-se à vontade para corrigir os erros ou implementar outras opções que não constem do programa fornecido `video.exe`.
- Faça declaração de variáveis e alocação de memória similar às feitas no programa fornecido `alunos.c`. De outra forma, não haverá memória disponível no QuickC para compilar seu programa.
- Seu programa deve funcionar obrigatoriamente com o arquivo de dados fornecido, o arquivo `video.txt`.
- Você deve entregar uma listagem impressa de seu programa e um disquete contendo, além de seu programa em C, uma versão executável do mesmo.
- Todas as observações gerais dadas anteriormente valem para este exercício-programa.
- O quarto exercício-programa vai ser uma extensão deste, portanto FAÇA UMA CÓPIA DO SEU EXERCÍCIO-PROGRAMA ANTES DE ENTREGÁ-LO. Você vai precisar.

Observações sobre o Exemplo

Nas páginas seguintes você está recebendo uma cópia do programa `alunos.c`. Descrevemos abaixo alguns conceitos usados nesse programa.

- A função `malloc` é usada para alocação dinâmica de memória. Por exemplo, a declaração

  ```
  int *creditos;
  ```

  seguida do comando

  ```
  creditos= (int *)malloc(N*sizeof(int));
  ```

  é o mesmo, para efeito de algoritmo, que a declaração

  ```
  int creditos[N];
  ```

  Isso permite que o programa funcione com vetores maiores.

  O comando `free` libera a memória alocada pelo comando `malloc`.
- A leitura de strings com `scanf` é inconveniente pois considera que o string termina no primeiro *branco*. Para resolver esse problema, a leitura de strings é feita usando o comando `fgets(string, tamanho, arquivo)`. O parâmetro `tamanho` especifica o número máximo de caracteres a serem lidos.
- A palavra `stdin` se refere ao dispositivo padrão de entrada (standard input), que é o teclado. Usamos `stdin` nas funções `fgets` e `fflush`.
- Durante os comandos de leitura do teclado, os dados digitados são armazenados numa memória auxiliar (buffer) e fornecidos ao programa conforme as leituras são feitas. Se por engano o usuário digitar mais dados que o pedido em uma certa leitura, os dados não usados serão fornecidos numa leitura posterior, causando comportamentos imprevisíveis do programa, tais como: pular a próxima leitura, entrar em loop, etc. Para evitar problemas desse tipo, usamos no exemplo o comando `fflush`, que descarrega do buffer de leitura eventuais dados que ainda não foram lidos.