Computação para Informática - Prof. Adriano Joaquim de Oliveira Cruz
Aula prática sobre arquivos binários

1 Escrita de Arquivos Binário

Esta aula será sobre escrita e leitura de arquivos binários.

Nesta parte você irá usar a função `fwrite` descrita abaixo.

```
escritos = fwrite (ponteiroParaDados, sizeof(tipo), quantidade, ponteiroArquivo );
```

No protótipo da função a variável `escritos` corresponde ao número de itens gravados com sucesso. A função `fwrite` escreve um bloco de dados para o arquivo apontado por `ponteiroArquivo`. O conjunto tem tamanho `quantidade`, sendo cada um de tamanho de `sizeof(tipo)` bytes e ficam na área de memória apontada por `ponteiroParaDados`.

Exercício 1: A listagem 1 mostra o uso da função `fwrite`. Escreva o programa mostrado na listagem 1 e verifique como ele funciona.

Listagem 1: Programa exemplo.

```c
/* fwrite example : write buffer */
#include <stdio.h>

int main (void) {
  FILE * pFile;
  int buffer[ ] = { 10, 20, 30 };

  pFile = fopen ( "myfile.bin" , "wb" );
  if (!pFile) return 1;
  fwrite (buffer , sizeof(int), 3, pFile );
  fclose (pFile);
  return 0;
}
```

2 Leitura de Arquivos Binários

Nesta etapa você irá usar a função `fread` descrita abaixo.

```
lidos = fread (ponteiroParaDados, sizeof(tipo), quantidade, ponteiroArquivo );
```

No protótipo da função a variável `lidos` corresponde ao número de itens gravados com sucesso. A função `fread` lê um bloco de dados do arquivo apontado por `ponteiroArquivo`. Espero ler `quantidade` de dados, cada um de tamanho de `sizeof(tipo)` bytes e os dados devem ir para área de memória apontada por `ponteiroParaDados`.

Exercício 2: A listagem 2 mostra um exemplo de uso da função `fread`. Escreva o programa mostrado na listagem 2 e verifique se ele leu os dados do primeiro exercício.

Listagem 2: Programa exemplo.

```
/* Le arquivo com três números */
#include<stdio.h>

int main () {
  FILE * pFile;
  int respos[3];
  int i;

  pFile = fopen ( "myfile.bin" , "r" );
  if (!pFile) return 1;
  fread (respos , sizeof(int) , 3, pFile );
  for (i = 0; i < 3; i++) {
     printf("%d ", respos[i]);
  }
  printf("\n");
  fclose(pFile);
  return 0;
}
```

Exercício 3: Escreva um programa que grave em um arquivo binário chamado `reais.bin` três números reais lidos do teclado.

Exercício 4: Escreva um programa que leia do arquivo `reais.bin` os três números escritos.

---

3 Posicionando o ponteiro do arquivo

Neste exercício você irá usar a função descrita abaixo.

```
int fseek (pontArq, deslocamentoEmBytes, deOnde );
```

Esta função serve para podermos andar pelo arquivo binário e lermos e escrevermos em qualquer posição que quisermos. **Atenção: Não funciona bem com arquivos texto.**

Por exemplo, considere que em um ARQUIVO BINÁRIO estão gravados os seguintes números 10, 20 e 30. Se quisermos alterar o número 20 podemos posicionar o ponteiro do arquivo em cima dele e escrevermos um novo valor.

A função posiciona o indicador de posição do arquivo em uma posição definida pela soma de um deslocamento à uma posição de referência padrão.

Parâmetros

`pontArq`: ponteiro para o arquivo.
`deslocamentoEmBytes`: número de bytes que o indicador de posição do arquivo deve ser deslocado a partir da origem indicada.
`deOnde`: posição a partir da qual o ponteiro deve ser deslocado. Esta posição é uma das seguintes:

`SEEK_SET` começo do arquivo
`SEEK_CUR` posição atual
`SEEK_END` final do arquivo

Retorno Em caso de sucesso retorna zero. Retorna um valor diferente de zero no caso de acerto.

Exercício 5:

A listagem 3 mostra um exemplo de uso da função `fseek`. Este programa grava no arquivo o número 5 no lugar do número 20 previamente gravado. Verifique o funcionamento do programa. Depois modifique-o de forma que o usuário informe qual é a posição do número a ser substituído.

Listagem 3: Programa exemplo.

```
#include<stdio.h>
int main () {
  FILE * pFile;
  int respos[3], i, novoValor = 5;

  /* abre arquivo para leitura e escrita */
  pFile = fopen ( "myfile.bin" , "r+" );
  if (!pFile) return 1;
  /* le o que esta escrito */
  fread (respos , sizeof(int) , 3, pFile );
  for (i = 0; i < 3; i++)  {
     printf("%d ", respos[i]);
  }
  printf("\n");
  /* pular 1 inteiro a partir do inicio
     ir para o segundo inteiro */
  fseek(pFile, 1*sizeof(int), SEEK_SET);
  /* escreve um novo valor na posicao do segundo inteiro */
  fwrite (&novoValor, sizeof(int), 1, pFile );
  printf("novo %d\n", novoValor);
  /* voltar para o inicio do arquivo */
  fseek(pFile, 0, SEEK_SET);  /* ou rewind tambem serve */
  /* le os novos valores */
  fread (respos, sizeof(int), 3, pFile );
  for (i = 0; i < 3; i++)  {
     printf("%d ", respos[i]);
  }
  printf("\n");
  fclose(pFile);
  return 0;
}
```

4 Descobrindo o tamanho de um arquivo binário

Neste exercício você irá usar a função descrita abaixo.

long int `ftell ( FILE * stream );`

Retorna a posição atual no fluxo de dados. Para arquivos binários, o valor retornado corresponde ao número de bytes desde o início do arquivo.

Parâmetro

Ponteiro para o arquivo identificado por `FILE`.

Retorno

Em caso de sucesso retorna o valor corrente do indicador de posição do arquivo. Em caso de erro retorna `-1L`.

Exercício 6:

A listagem 4 mostra um exemplo de uso da função `ftell`. Verifique o funcionamento do programa.

Listagem 4: Programa exemplo.

```c
/* ftell example : getting size of a file */
#include <stdio.h>

int main () {
  FILE * pFile;
  long size;

  pFile = fopen ("myfile.txt","rb");
  if (pFile==NULL) {
     perror ("Error opening file.");
  }
  else {
    fseek (pFile, 0, SEEK_END); // go to the end of file
    size=ftell (pFile);
    fclose (pFile);
    printf ("Size of myfile.txt: %ld bytes.\n",size);
  }
  return 0;
}
```

Exercício 7: Escreva um programa que calcule em número de bytes o tamanho do arquivo `ex1.bin`

---

5 Estruturas e arquivos binários

Exercício 8: A listagem 5 mostra um exemplo de uso da função `fwrite` com estruturas.

Listagem 5: Programa exemplo.

```
#include<stdio.h>

typedef struct _JOGADOR {
     char nome[40];
     int recorde;
} JOGADOR;

int main () {
  FILE * pFile;
  JOGADOR recordistas[3];
  int i;
  char linha[80];

  pFile = fopen ( "estruturas.bin" , "w" );
  if (!pFile) return 1;

  for (i = 0; i < 3; i++) {
     printf("Nome Jogador (so um nome!!!)%d: ", i);
     scanf("%s", recordistas[i].nome);
     printf("Recorde Jogador %d: ", i);
     gets(linha);
     fscanf(linha, "%d", &recordistas[i].recorde);
  }
  fwrite(recordistas, sizeof(JOGADOR), 3, pFile);
  fclose(pFile);
  return 0;
}
```

Exercício 9: Escreva um programa que leia do arquivo `estruturas.bin` os dados gravados no exercício anterior para verificação.

Exercício 10: Na página da aula prática há um arquivo chamado `myfile.bin` contendo um conjunto de números inteiros entre 0 e 26. Este conjunto contém uma mensagem codificada. Para decodificar a mensagem cada número inteiro deve ser convertido para uma letra segundo a seguinte regra: 0 = 'A', 1 = 'B', 2 = 'C' e assim por diante. Escreva um programa que imprima esta mensagem.

Exemplo de mensagem codificada: 2030
Mensagem decodificada: CADA

O primeiro aluno que decodificar esta mensagem durante a aula prática tira 10 na nota do teste 22. A nota será dada a somente um aluno(a).