

Laboratório

Algoritmos e

Estruturas de dados I

Arquivo

Ramon – 6º Período – Engenharia da Computação

Roteiro

- ***Definição***
- ***Comandos para manipular arquivos***
- ***Manipulando dados em arquivo***
- ***Exemplos***

Definição

Um arquivo é um recurso que as linguagens de programação tem acesso, onde se permite manipular dados (leitura e escrita) utilizados por dispositivos de entrada e saída.

Comandos para manipular arquivos

Comando	Função												
fopen(nome_arquivo,modo)	Abre um arquivo. Onde <i>nome_arquivo</i> é o caminho e o nome válido do arquivo, <i>modo</i> é a forma que o arquivo será aberto (conforme tabela 2).												
fclose(*p)	Fecha um arquivo. Onde <i>*p</i> é ponteiro de arquivo.												
fread(buffer,numbytes,cont,*p)	Lê um bloco de dados do arquivo. Onde <i>buffer</i> é a região onde será lida, <i>numbyte</i> o número de bytes a ser lido, <i>cont</i> o número de vezes que será lido e <i>*p</i> ponteiro de arquivo.												
fwrite(buffer,numbytes,cont,*p)	Escreve um bloco de dados no arquivo. Onde <i>buffer</i> é a região onde será escrita, <i>numbyte</i> o número de bytes a ser escrito, <i>cont</i> o número de vezes que será escrito e <i>*p</i> ponteiro de arquivo.												
fseek(*p,numbytes,origem)	Movimenta a posição corrente de leitura/escrita no arquivo a partir de um ponto especificado. Onde <i>*p</i> é o ponteiro de arquivo e <i>origem</i> determina a partir de onde os <i>numbytes</i> de movimentação serão contados. O parâmetro <i>origem</i> pode ser:												
	<table><tr><th>Nome</th><th>Valor</th><th>Significado</th></tr><tr><td>SEEK_SET</td><td>0</td><td>Início do arquivo.</td></tr><tr><td>SEEK_CUR</td><td>1</td><td>Ponto corrente do arquivo.</td></tr><tr><td>SEEK_END</td><td>2</td><td>Fim do arquivo.</td></tr></table>	Nome	Valor	Significado	SEEK_SET	0	Início do arquivo.	SEEK_CUR	1	Ponto corrente do arquivo.	SEEK_END	2	Fim do arquivo.
Nome	Valor	Significado											
SEEK_SET	0	Início do arquivo.											
SEEK_CUR	1	Ponto corrente do arquivo.											
SEEK_END	2	Fim do arquivo.											
sizeof(v)	Retorna o tamanho em bytes de uma variável ou tipo de dado. Onde <i>v</i> é uma variável ou tipo de dados.												

Tabela 1.

Modo Significado

r	Abre um arquivo-texto para leitura
w	Cria um arquivo-texto para escrita
a	Anexa a um arquivo-texto
rb	Abre um arquivo binário para leitura
wb	Cria um arquivo binário para escrita
ab	Anexa a um arquivo binário
r+	Abre um arquivo-texto para leitura/escrita
w+	Cria um arquivo-texto para leitura/escrita
a+	Anexa ou cria um arquivo-texto para leitura/escrita
r+b	Abre um arquivo binário para leitura/escrita
w+b	Cria um arquivo binário para leitura/escrita
a+b	Anexa a um arquivo binário para leitura/escrita

Tabela 2.

Manipulando dados em arquivo

❖ Abrindo um arquivo

```
#include<iostream>
using namespace std;

int main()
{
    //declarando ponteiro de arquivo
    FILE *p;
    //criando o arquivo teste.txt para escrita
    p = fopen("teste.txt", "wb+");

    cin.ignore();
    cin.get();
}
```

Embora tecnicamente correto, você geralmente verá o código a cima escrito desta forma:

Para abrir um arquivo chamado Teste, permitindo escrita, pode-se digitar:

```
#include<iostream>
using namespace std;

int main()
{
    //declarando ponteiro de arquivo
    FILE *p;
    //criando o arquivo teste.txt para escrita
    p = fopen("teste.txt", "wb+");
    //verificando a existência do arquivo
    if (p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
        cout << "Arquivo aberto com sucesso!";

    cin.ignore();
    cin.get();
}
```

Manipulando dados em arquivo

❖ Inserindo um dado no arquivo

```
#include<iostream>
using namespace std;

int main()
{
    char nome[20];
    //declarando ponteiro de arquivo
    FILE *p;
    //criando o arquivo teste.txt para escrita
    p = fopen("teste.txt", "wb+");
    //verificando a existência do arquivo
    if (p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
    {
        cout << "Entre com seu nome: ";
        cin.getline(nome, 20);
        //gravando o nome no arquivo
        fwrite(&nome, sizeof(nome), 1, p);
    }
    fclose(p);
    cin.ignore();
    cin.get();
}
```

Para gravar um dado no arquivo utiliza-se o comando *fwrite()* descrito na **tabela 1**. Uma vez executado este programa será criado o arquivo **teste.txt**, e nele, será gravado o nome inserido pelo usuário.

Manipulando dados em arquivo

❖ Lendo um dado no arquivo

```
#include<iostream>
using namespace std;

int main()
{
    char nome[20];
    //declarando ponteiro de arquivo
    FILE *p;
    //abre o arquivo teste.txt para leitura
    p = fopen("teste.txt","rb+");
    //verificando a existência do arquivo
    if(p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
    {
        cout << "Nome lido no arquivo: ";
        //lendo o nome no arquivo
        fread(&nome,sizeof(nome),1,p);
        cout << "\n" << nome;
    }
    fclose(p);
    cin.ignore();
    cin.get();
}
```

Para ler um dado no arquivo utiliza-se o comando **fread()** descrito na **tabela 1**. Uma vez executado este programa será aberto o arquivo **teste.txt** e passado a variável **nome** o valor do nome encontrado gravado no arquivo.

Manipulando dados em arquivo

❖ Inserindo vários dados no arquivo

Segue o mesmo princípio, porém, utiliza-se de uma variável contadora para realizar a contagem de dados gravados e percorrer o arquivo.

```
#include<iostream>
using namespace std;

char nome[20];
//variável utilizada para verificar qtde
//de dados gravada no arquivo
int cont = 0;

int main()
{
    //declarando ponteiro de arquivo
    FILE *p;
    //criando o arquivo teste.txt para escrita
    p = fopen("teste2.txt","wb+");
    //verificando a existência do arquivo
    if(p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
    {
        int x = 0;
        //variável utilizada para saltar dados
        //já existentes no arquivo
        int salto = 0;

        while(x==0)
        {
            //posicionando o ponteiro no início
            fseek(p,0,SEEK_SET);
            //verificando nº de dados no arquivo
            fread(&cont,sizeof(cont),1,p);
            cin.ignore();
            cout << "Entre com seu nome: ";
            cin.getline(nome,20);
            //calculando salto
            salto = sizeof(nome)*cont+sizeof(cont);
            //posicionando o ponteiro p/ gravação
            fseek(p,salto,SEEK_SET);
            //gravando o nome no arquivo
            fwrite(&nome,sizeof(nome),1,p);
            //posicionando o ponteiro no início
            fseek(p,0,SEEK_SET);
            //atualizando e gravando qtde dados
            cont++;
            fwrite(&cont,sizeof(cont),1,p);
            cout << "Deseja continuar 0=sim: ";
            cin >> x;
        }
        fclose(p);
        cin.ignore();
        cin.get();
    }
}
```

Manipulando dados em arquivo

❖ Lendo vários dados no arquivo

```
#include<iostream>
using namespace std;

char nome[20];
//variável utilizada para verificar qtde
//de dados gravada no arquivo
int cont = 0;

int main()
{
    //declarando ponteiro de arquivo
    FILE *p;
    //abrindo o arquivo teste.txt para leitura
    p = fopen("teste2.txt", "rb+");
    //verificando a existência do arquivo
    if(p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
    {
        //variável utilizada para saltar dados
        //já existentes no arquivo
        int salto = 0;
        //posicionando o ponteiro no início
        fseek(p,0,SEEK_SET);
        //verificando n° de dados no arquivo
        fread(&cont,sizeof(cont),1,p);
```

```
for(int i=0;i<cont;i++)
{
    //calculando salto
    salto = sizeof(nome)*i+sizeof(cont);
    //posicionando o ponteiro p/ leitura
    fseek(p,salto,SEEK_SET);
    //Lendo o nome do arquivo
    fread(&nome,sizeof(nome),1,p);
    cout << " " << nome;
}
fclose(p);
cin.ignore();
cin.get();
}
```

Também segue o mesmo princípio, e também utiliza-se da variável contadora para realizar a contagem de dados gravados e percorrer o arquivo.

Manipulando dados em arquivo

❖ Excluindo dados do arquivo

A exclusão de um dado do arquivo é feita da seguinte forma:

1º-> Busca no arquivo o dado a ser excluído.

2º-> Uma vez encontrado, copia o conteúdo da última posição do arquivo sobre o dado a ser excluído.

3º-> Atualiza o contador.

Manipulando dados em arquivo

❖ Excluindo dados do arquivo

```
#include<iostream>
#include<cstring>
using namespace std;

char nome[20];
//variável utilizada para verificar qtde
//de dados gravada no arquivo
int cont = 0;

int main()
{
    char auxnome[20];
    //declarando ponteiro de arquivo
    FILE *p;
    //abrindo o arquivo teste.txt para leitura
    p = fopen("teste2.txt", "rb+");
    //verificando a existência do arquivo
    if(p==NULL)
    {
        cout << "Erro ao abrir arquivo!";
        exit(1);
    }
    else
    {
        //variável utilizada para saltar dados
        //já existentes no arquivo
        int salto = 0;
        //posicionando o ponteiro no início
        fseek(p, 0, SEEK_SET);
        //verificando n° de dados no arquivo
        fread(&cont, sizeof(cont), 1, p);

        cout << "Entre com o nome a ser excluído: ";
        cin.getline(auxnome, 20);

        for(int i=0; i<cont; i++)
        {
            //calculando salto
            salto = sizeof(nome)*i+sizeof(cont);
            //posicionando o ponteiro p/ leitura
            fseek(p, salto, SEEK_SET);
            //Lendo o nome do arquivo
            fread(&nome, sizeof(nome), 1, p);
            //busca o nome a ser excluído
            if(strcmpi(nome, auxnome)==0)
            {
                //calculando salto p/ última posição do arquivo
                salto = sizeof(nome)*(cont-1)+sizeof(cont);
                //salta p/ última posição do arquivo
                fseek(p, salto, SEEK_SET);
                //ler o último nome do arquivo
                fread(&nome, sizeof(nome), 1, p);
                //calculando salto para a posição a ser excluída
                salto = sizeof(nome)*i+sizeof(cont);
                //salta para o nome a ser excluído
                fseek(p, salto, SEEK_SET);
                //grava o último dado do arquivo sobre o dado
                //a ser excluído
                fwrite(&nome, sizeof(nome), 1, p);
                //atualiza o contador
                cont--;
                //posicionando o ponteiro no início do arquivo
                fseek(p, 0, SEEK_SET);
                //grava o contador atualizado
                fwrite(&cont, sizeof(cont), 1, p);
            }
        }
        fclose(p);
    }
}
```

Exemplos

Seja uma estrutura para descrever os carros de uma determinada revendedora, contendo os seguintes campos:

marca: string de tamanho 15
ano: inteiro
cor: string de tamanho 10
preço: real

Elabore um programa que possua o seguinte menu:

- 1 – Cadastrar
- 2 – Listar carros
- 3 – Editar dados de um carro
- 4 – Sair

Obs: Uma vez escolhida a opção 1 ou 3, o programa deve solicitar ao usuário a entrada da marca do carro a ser pesquisado. O programa só deve ser fechado se a opção sair for escolhida. **Todos os dados devem ser gravados em um arquivo.**