

Arquivos

Prof. Thiago Felski Pereira, MSc.

Biblioteca fstream

```
#include <fstream>
```

- A biblioteca `fstream` fornece classes que nos permite realizar a leitura e a escrita em arquivos de maneira similar ao que acontece com o `cin` e `cout` que já utilizamos para leitura e escrita no console.
- A única diferença é que precisamos associar nossos `streams` a arquivos físicos.

Biblioteca fstream

- Uma vez adicionada a biblioteca, teremos três classes para leitura e escrita em arquivos
 - `ofstream`
 - Classe stream para **escrita** em arquivos
 - `ifstream`
 - Classe stream para **leitura** de arquivos
 - `fstream`
 - Classe stream para **leitura e escrita** em arquivos

Escrevendo em um arquivo

- A escrita em um arquivo acontece em quatro passos

- PASSO 1: crie um stream do tipo ofstream

```
ofstream meu_arquivo;
```

- PASSO 2: associe um arquivo ao seu stream

```
meu_arquivo.open ("exemplo.txt");
```

- PASSO 3: envie dados para esse stream

- Similar ao que você já faz com o cout, inclusive os comandos \...

```
meu_arquivo << "Escrevendo nesse arquivo.\n";
```

- PASSO 4: encerre a utilização do arquivo

```
meu_arquivo.close();
```

Escrevendo em um arquivo

- Exemplo

```
1  #include <fstream>
2  using namespace std;
3
4  int main () {
5      ofstream meu_arquivo; //PASSO 1
6      meu_arquivo.open ("exemplo.txt"); //PASSO 2
7      meu_arquivo << "Escrevendo nesse arquivo.\n"; //PASSO 3
8      meu_arquivo.close(); //PASSO 4
9      return 0;
10 }
```

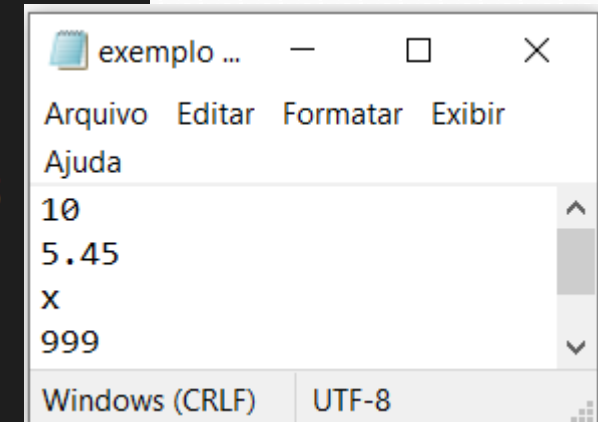
Escrevendo em um arquivo

- Podemos escrever outras coisas além de `strings` prontas.

`"string pronta"`

- Observe o exemplo

```
1  #include <fstream>
2  using namespace std;
3  int main () {
4      float valor=999;
5      ofstream meu_arquivo;
6      meu_arquivo.open ("exemplo.txt");
7      meu_arquivo << 10 <<"\n";
8      meu_arquivo << 5.45 <<"\n";
9      meu_arquivo << 'x' <<"\n";
10     meu_arquivo << valor <<"\n";
11     meu_arquivo.close();
12     return 0;
13 }
```



Praticando escrita em arquivos

- Atividade 01
 - Faça um programa que armazene os dados de 5 alunos de uma turma em arquivo.
- Atividade 02
 - Faça uma função em que o usuário forneça como parâmetros os valores de A, B e C para uma equação do segundo grau e salva em arquivo a equação e o seu resultado.
 - Por exemplo, se o usuário forneceu os respectivos valores: A=1, B=-5, C=6
 - O arquivo deverá conter:
 - $1x^2 - 5x + 6 = 0$
 - $x' = 3$ e $x'' = 2$



Escrevendo em um arquivo

- Verificar se o arquivo foi aberto corretamente
- Em muitos programas é necessário ter certeza que o arquivo foi corretamente aberto antes de tentar executar qualquer operação.
 - No fstream esse comando chama-se

```
.is_open();
```


Escrevendo em um arquivo

- Exemplo: Verificação de arquivo corretamente aberto

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  int main () {
5      ofstream meu_arquivo;
6      meu_arquivo.open ("exemplo.txt");
7      if ( meu_arquivo.is_open() ) {
8          meu_arquivo<<"Salvando no arquivo aberto";
9          meu_arquivo.close();
10     } else {
11         cout<<"O arquivo não pode ser aberto";
12     }
13     return 0;
14 }
```

Lendo de um arquivo

- A leitura de um arquivo acontece em três passos

- PASSO 1: crie um stream do tipo ifstream

```
ifstream meu_arquivo;
```

- PASSO 2: associe um arquivo ao seu stream

```
meu_arquivo.open ("exemplo.txt");
```

- PASSO 3: leia dados para desse stream

- Observe que esse comando lê um caractere por vez

```
char lido;  
lido = meu_arquivo.get();
```

- PASSO 4: encerre a utilização do arquivo

```
meu_arquivo.close();
```

Lendo de um arquivo

- Exemplo

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  int main () {
5      char lido;
6      ifstream meu_arquivo; //PASSO 1
7      meu_arquivo.open ("exemplo.txt"); //PASSO 2
8      lido = meu_arquivo.get(); //PASSO 3
9      meu_arquivo.close(); //PASSO 4
10     return 0;
11 }
```

Lendo de um arquivo

- Se seu objetivo for ler uma palavra inteira do arquivo o comando fica bem parecido com o cin

```
string palavra;  
meu_arquivo >> palavra;
```

- Se seu objetivo for ler uma linha inteira

```
string frase;  
getline(meu_arquivo, frase);  
cout<<frase;
```

- Se seu objetivo for ler o arquivo inteiro

```
string frase;  
while (meu_arquivo.eof()==false){  
    getline(meu_arquivo, frase);  
    cout<<frase<<"\n";  
}
```

Praticando leitura em arquivos

- Atividade 03
 - Faça um programa que, caso o arquivo tenha sido corretamente lido, leia ele inteiro e imprima no console
- Atividade 04
 - Faça um programa que
 - Que carregue vários nomes e idades em um arquivo
 - Permita procurar por um nome e, caso encontre, imprima a idade daquela pessoa
- Atividade 05
 - Leia uma matriz que está em um arquivo e carregue ela em uma matriz

Arquivos CSV

- CSV (Comma-Separated Values) é um arquivo que separa os dados por virgulas.
- Esse arquivo comumente é usado no **Microsoft Excel** e o **LibreOffice Calc**

```
1997,Ford,E350,"ac, abs, moon",30100.00
```

```
1999,Chevy,"Venture ""Extended Edition""",,49000.00
```

```
1996,Jeep,Grand Cherokee,"MUST SELL!
```

```
air, moon roof, loaded",479699.00
```

	A	B	C	D	E
1	1997	Ford	E350	ac, abs, moon	30100.00
2	1999	Chevy	Venture "Extended Edition"		49000.00
3	1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	479699.00

Escrevendo em um arquivo CSV

- Siga os mesmos 4 passos de uma escrita normal

- PASSO 1: crie um stream do tipo ofstream

```
ofstream meu_arquivo;
```

- PASSO 2: associe um arquivo ao seu stream

```
meu_arquivo.open("tabela.csv");
```

- PASSO 3: envie dados para esse stream

```
meu_arquivo << "Escrevendo nesse arquivo.\n";
```

- PASSO 4: encerre a utilização do arquivo

```
meu_arquivo.close();
```

Escrevendo em um arquivo CSV

- Escrevendo em uma única linha

```
meu_arquivo <<"sep=,\n";  
for (int i=0; i<10; i++){  
    meu_arquivo << i <<','';  
}
```

- Para que o EXCEL abra corretamente seu arquivo é necessário indicar na primeira linha o caractere utilizado na separação de colunas `meu_arquivo <<"sep=,\n";`

- Escrevendo em uma única coluna

```
for (int i=0; i<10; i++){  
    meu_arquivo << i <<"\n";  
}
```


Escrevendo em um arquivo CSV

- Você também pode escrever comandos EXCEL dessa forma

```
meu_arquivo<<"=SOMA(A1:A10)";
```

- Exemplo

```
int main () {  
    ofstream meu_arquivo;  
    meu_arquivo.open("tabela.csv");  
  
    for (int i=0; i<10; i++){  
        meu_arquivo << i <<"\n";  
    }  
    meu_arquivo<<"=SOMA(A1:A10)";  
    meu_arquivo.close();  
    return 0;  
}
```

Lendo de um arquivo CSV

- A leitura de um arquivo CSV acontece da mesma forma

- PASSO 1: crie um stream do tipo ifstream

```
ifstream meu_arquivo;
```

- PASSO 2: associe um arquivo ao seu stream

```
meu_arquivo.open("tabela.csv");
```

- PASSO 3: leia dados desse stream

- **Atenção:** Esse comando vai ler todos os chars, incluindo `sep=,` as vírgulas e o caractere de fim de linha

```
char lido;  
lido = meu_arquivo.get();
```

- PASSO 4: encerre a utilização do arquivo

```
meu_arquivo.close();
```

Exercícios

- A seguinte função é utilizada para ler uma matriz de um arquivo .txt e escrever em uma matriz de caracteres

```
void carrega_arquivo (char m[L][C]){  
    ifstream arquivo ("file1.txt");  
    char letra;  
    for (int i=0;i<L;i++) {  
        for (int j=0;j<C+1;j++) { //após cada linha do arquivo existe um \n  
            arquivo.get(letra);  
            if (j!=C){           //esse if ajuda a ignorar o \n  
                m[i][j]=letra;  
            }  
        }  
    }  
}
```

- Agora tente você: leia uma matriz de um arquivo .csv e salve em uma matriz de inteiros

Modos de arquivo disponíveis no c++

- `ios_base::in` Abre arquivo para leitura.
- `ios_base::out` Abre arquivo para escrita.
- `ios_base::ate` Procura o final do arquivo ao abrir ele.
- `ios_base::app` Anexa os dados à serem escritos ao final do arquivo.
- `ios_base::trunc` Trunca os dados existentes no arquivo.
- `ios_base::binary` Abre e trabalha com arquivos em modo binário.

Obrigado pela atenção

contato: Felski@univali.br

