

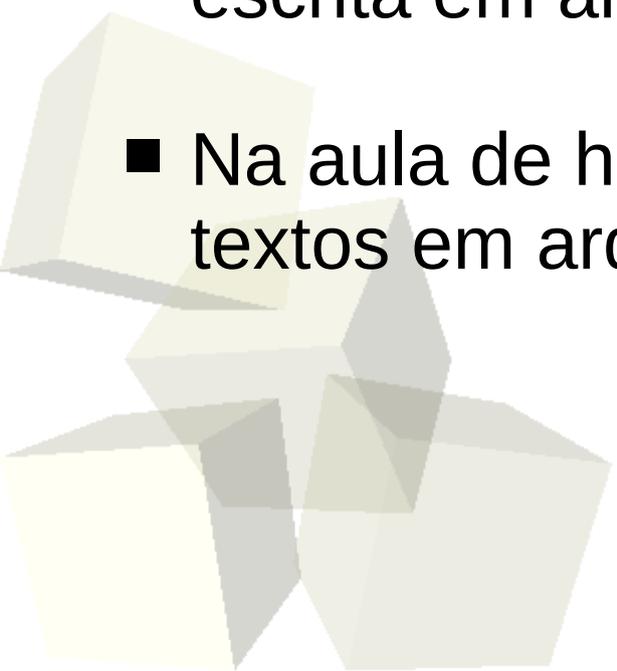
Linguagem C++ *Entrada e saída com arquivos*

Prof.: Bruno E. G. Gomes
IFRN





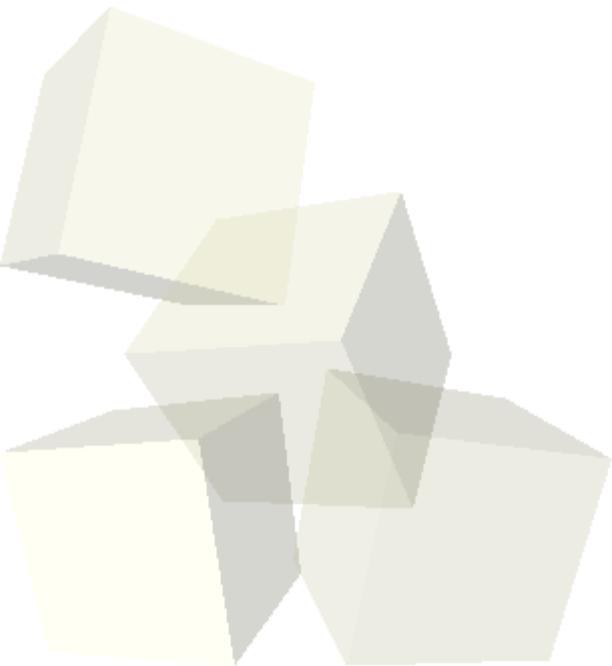
- Entrada e saída de dados pode ser feita:
 - Para dispositivo de entrada/saída (monitor, impressora, etc.)
 - Para um *arquivo* no disco rígido
 - Para um banco de dados (PABD – 3º ano)
- C++ fornece tipos para trabalharmos com leitura e escrita em arquivos
- Na aula de hoje, trataremos da leitura e escrita de textos em arquivos





Classes para entrada/saída

- Tipos para trabalhar com arquivos:
 - **ofstream** – para escrever em um arquivo
 - **ifstream** – para ler dados de um arquivo
 - **fstream** – pode ser utilizada tanto para leitura quanto para escrita em arquivos
- Fazem parte da biblioteca `<fstream>`:
#include `<fstream>`





```
#include <iostream>
#include <fstream>

using namespace std;

int main () {
    // ofstream - arquivo apenas para saída de dados
    ofstream arq1;

    // função open - abre o arquivo. Cria o arquivo caso ele não exista.
    arq1.open ("nomes.txt");

    // Insere nomes no arquivo (operador "<<")
    arq1 << "Bruno Gomes" << endl;
    arq1 << "Maria Dantas" << endl;

    // função close - fecha o arquivo
    arq1.close();

    return 0;
}
```





```
#include <iostream>
#include <fstream>
```

```
using namespace std;
```

```
int main () {
```

```
    /* Outra forma: arquivo a ser aberto já é colocado diretamente
       após nome da variável arquivo */
```

```
    ofstream arq1("nomes.txt");
```

```
    //função is_open – testa se o arquivo está realmente aberto
```

```
    if (arq1.is_open()) {
        arq1 << "Bruno Gomes" << endl;
        arq1 << "Maria Dantas" << endl;
        arq1.close();
    }
```

```
    return 0;
```

```
}
```





Opções da função *open*

- *open* (<nome_arq>, <modo>)
- <modo> um valor ou a combinação dos seguintes valores:
 - **ios::in** – abre o arquivo para leitura (entrada)
 - **ios::out** – abre o arquivo para escrita (saída)
 - **ios::binary** – abre arquivos que não são de texto (binários)
 - **ios::ate** – coloca a posição inicial de leitura ou escrita no final do arquivo
 - **ios::app** – indica que todas as operações de escrita devem ser feitas no final do arquivo (apenas para escrita - *ofstream*)
 - **ios::trunc** – se o arquivo aberto para leitura já existia anteriormente, seu conteúdo é apagado





ofstream arq1;

- `arq1.open ("nomes.txt", ios::out)`
 - Modo padrão para *ofstream* (`ios::out`)
- `arq1.open ("nomes.txt", ios::out | ios::app)`
 - Abre no final do arquivo “nomes.txt”. Ou seja, na posição depois do último caractere inserido.
- `arq1.open ("nomes.txt", ios::app);`
 - Mesmo que o anterior, uma vez que “out” é o padrão para *ofstream*



Exemplo: escrita no final do arquivo

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

int main () {
    ofstream arq1;
    //ios::app - abre o arquivo no final (depois do último caractere)
    arq1.open ("nomes.txt", ios::app);

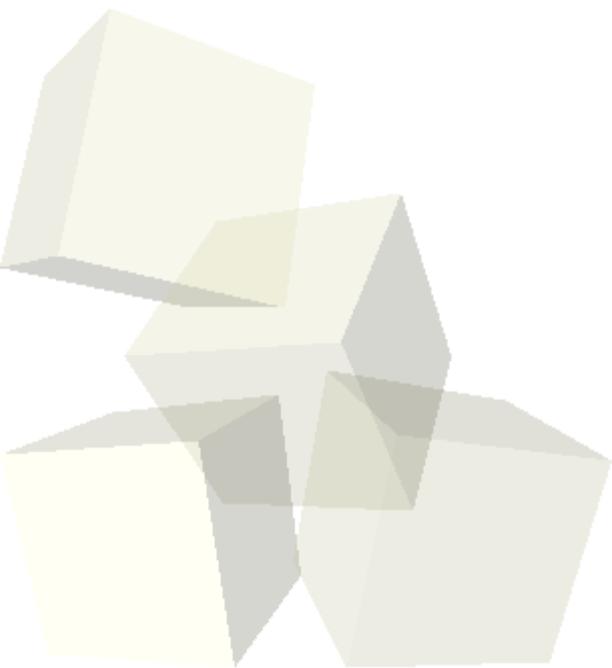
    if (arq1.is_open()) {
        //tellp() - retorna a posição atual do apontador para escrita
        long pos = arq1.tellp();
        cout << "Posição atual no arquivo: " << pos << endl;

        // Insere nomes no final do arquivo (foi aberto com ios::app)
        arq1 << "Álvares de Azevedo" << endl;
        arq1 << "Machado de Assis" << endl;

        arq1.close();
    } else {
        cout << "ERRO: arquivo não foi aberto ou não existe" << endl;
    }
    return 0;
}
```



- Semelhante à saída, mas usa a classe *ifstream* ao invés de *ofstream*
 - ex.: ***ifstream*** arquivo (“*arq.txt*”);
 - ou: ***ifstream*** arquivo;
arquiv.**open**(“*arq.txt*”);





Exemplo de leitura

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main () {
    string linha;
    //ifstream – abre o arquivo apenas para leitura
    ifstream arq_in ("nomes.txt");

    if (arq_in.is_open()) {
        //eof() - retorna true ao atingir o fim do arquivo
        while (! arq_in.eof() ) {
            getline (arq_in, linha);
            cout << linha << endl;
        }
        arq_in.close();
    } else {
        cout << "ERRO: arquivo não foi aberto ou não existe" << endl;
    }

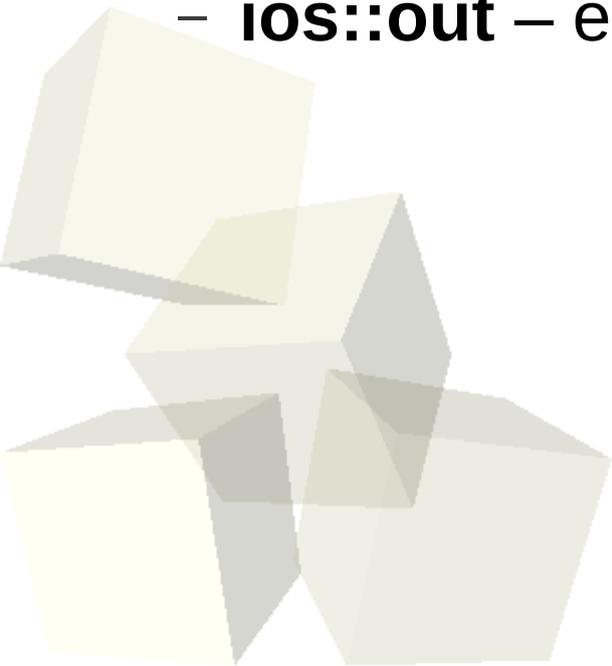
    return 0;
}
```





fstream – leitura/escrita

- Caso se necessite tanto de operações de leitura quanto de escrita em um mesmo arquivo
- Uso igual aos tipos de arquivo anteriores
- Na abertura do arquivo deve-se indicar qual operação deve ser feita
 - **ios::in** – leitura
 - **ios::out** – escrita





fstream - Exemplo

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

int main () {
    string linha;
    fstream arq; //fstream - leitura e escrita

    arq.open ("nomes.txt", ios::out | ios::app); //abre para escrita (ios::out)

    if (arq.is_open()) {
        //realiza uma escrita
        arq << "Roberto Carlos" << endl;
        arq.close();
        arq.open("nomes.txt", ios::in); //abre no modo de leitura (ios::in)

        while (! arq.eof() ) {
            getline (arq, linha);
            cout << linha << endl;
        }
        arq.close();
    } else {
        cout << "ERRO: arquivo não foi aberto ou não existe" << endl;
    }
    return 0;
}
```

