

# Arquivos

## Linguagem de Programação I

Waldson Patrício



# Streams de arquivos de entrada e saída

- C++ também fornece tipos *stream* para leitura e escrita de arquivos que são gravados no disco rígido
- Eles funcionam, em sua maioria, igual aos *streams* padrões (cin e cout)
- Para utilizá-los, devemos incluir o cabeçalho `fstream`:
  - `#include <fstream>`
  - Não há variáveis padrões para escrita e leitura de arquivos, você que deve criar as variáveis:
    - `std::ifstream arquivo_entrada;`
    - `std::ofstream arquivo_saida;`

# Conectando streams aos arquivos

- Por padrão os *streams* não estão conectados a arquivos. Para poder usar um *stream* de arquivo, você deve estabelecer uma conexão entre o arquivo e o *stream*. Isto pode ser feito de duas formas:
- Usando uma função membro do objeto chamada `open()`:
  - `arquivo_entrada.open("arquivo.txt")`
  - `arquivo_saida.open("outro_arquivo.txt")`
- Passando o nome do arquivo como parâmetro na inicialização da variável:
  - `std::ifstream arquivo_entrada("arquivo.txt")`
  - `std::ofstream arquivo_saida("outro_arquivo.txt")`

# Conectando *streams* aos arquivos

- Para *streams* de saída, se o arquivo **não** existir, ele será criado pelo sistema operacional
- Para *streams* de entrada, se o arquivo **não** existir, ele não será criado pelo sistema operacional e o *stream* conterà uma flag de erro

# Conectando streams aos arquivos

- Se o arquivo não existir, a função membro 'fail()' vai retornar true:

---

```
1      #include <iostream>
2
3      //outras partes do código...
4
5      if (arquivo_entrada.fail()) {
6          //não posso usar o stream porque o arquivo não existe
7      }
```

---

# Sobrescrevendo Arquivos

- Por padrão o `std::ofstream` vai sobrescrever o conteúdo anterior do arquivo se ele já existir
- Para adicionar conteúdo no final do arquivo, é necessário informar isto ou na inicialização ou no `open` através da flag `std::ios::app`:

```
1      std::ofstream arquivo_saida("arquivo", std::ios::app)
```

Dica: Evite o uso de abreviações

`app` é a abreviação de `append`.

# Desconectado *streams* dos arquivos

- Quando o programa não precisa mais utilizar o arquivo, devemos fechar a conexão entre o *stream* e o arquivo chamando a função membro `close()`:
  - `arquivo_entrada.close()`
  - `arquivo_saida.close()`
- Isto notifica o sistema operacional que não utilizaremos mais o arquivo e ele deve escrever dados pendentes e atualizar informações do arquivo.

Dúvidas?