

Manipulação de Arquivos em C++ (`<fstream>` e `<filesystem>`)

Guia que contém todos os métodos para manipulações de arquivos em c++.

1. Biblioteca `<fstream>`

Usada para **abrir, ler e escrever arquivos**.

Tipos principais

Tipo	Uso	Descrição
<code>std::ifstream</code>	Leitura	Abre arquivo para entrada
<code>std::ofstream</code>	Escrita	Abre arquivo para saída
<code>std::fstream</code>	Leitura + Escrita	Permite ambas as operações

Modos de abertura (flags)

Flag	Significado
<code>std::ios::in</code>	Leitura
<code>std::ios::out</code>	Escrita
<code>std::ios::app</code>	Adiciona ao final
<code>std::ios::trunc</code>	Apaga o conteúdo existente
<code>std::ios::binary</code>	Abertura em modo binário

Exemplo: Ler arquivo linha a linha

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

int main() {
    std::ifstream file("dados.txt");
    std::string linha;

    if (file.is_open()) {
        while (std::getline(file, linha)) {
            std::cout << linha << "\n";
        }
    }
}
```

Exemplo: Escrever em arquivo

```
#include <fstream>

int main() {
    std::ofstream file("saída.txt", std::ios::app);
    file << "Nova linha no arquivo.\n";
}
```

2. Biblioteca `<filesystem>` (C++17+)

Usada para **verificar, criar, mover e remover arquivos e diretórios**.

```
#include <filesystem>
namespace fs = std::filesystem;
```

Funções mais comuns

Função	Propósito	Exemplo
<code>fs::exists(path)</code>	Verifica se existe	<code>fs::exists("arquivo.txt")</code>
<code>fs::is_directory(path)</code>	Checa se é diretório	<code>fs::is_directory("pasta/")</code>
<code>fs::create_directory(p)</code>	Cria diretório	<code>fs::create_directory("nova_pasta")</code>
<code>fs::remove(path)</code>	Remove arquivo	<code>fs::remove("a.txt")</code>
<code>fs::remove_all(path)</code>	Remove diretório recursivamente	<code>fs::remove_all("pasta/")</code>
<code>fs::copy(a, b)</code>	Copia arquivos/pastas	<code>fs::copy("a.txt", "b.txt")</code>
<code>fs::rename(a, b)</code>	Renomeia ou move	<code>fs::rename("velho.txt", "novo.txt")</code>
<code>fs::file_size(path)</code>	Obtém tamanho em bytes	<code>auto s = fs::file_size("a.txt");</code>
<code>fs::current_path()</code>	Obtém caminho atual	<code>std::cout << fs::current_path();</code>

Exemplo: Criar diretório se não existir

```
#include <iostream>
#include <filesystem>

namespace fs = std::filesystem;

int main() {
    if (!fs::exists("logs")) {
        fs::create_directory("logs");
        std::cout << "Pasta criada.\n";
    }
}
```

Exemplo: Listar arquivos de um diretório

```

#include <filesystem>
#include <iostream>

namespace fs = std::filesystem;

int main() {
    for (auto& entry : fs::directory_iterator(".")) {
        std::cout << entry.path() << "\n";
    }
}

```

3. Resumo Geral

Biblioteca	Função Principal	Quando usar
<fstream>	Abrir, ler e escrever conteúdo de arquivos	Manipular dados internos do arquivo
<filesystem>	Gerenciar arquivos e diretórios no sistema	Estrutura, existência, cópia, remoção

Referências

- https://en.cppreference.com/w/cpp/io/basic_fstream
- <https://en.cppreference.com/w/cpp/filesystem>