

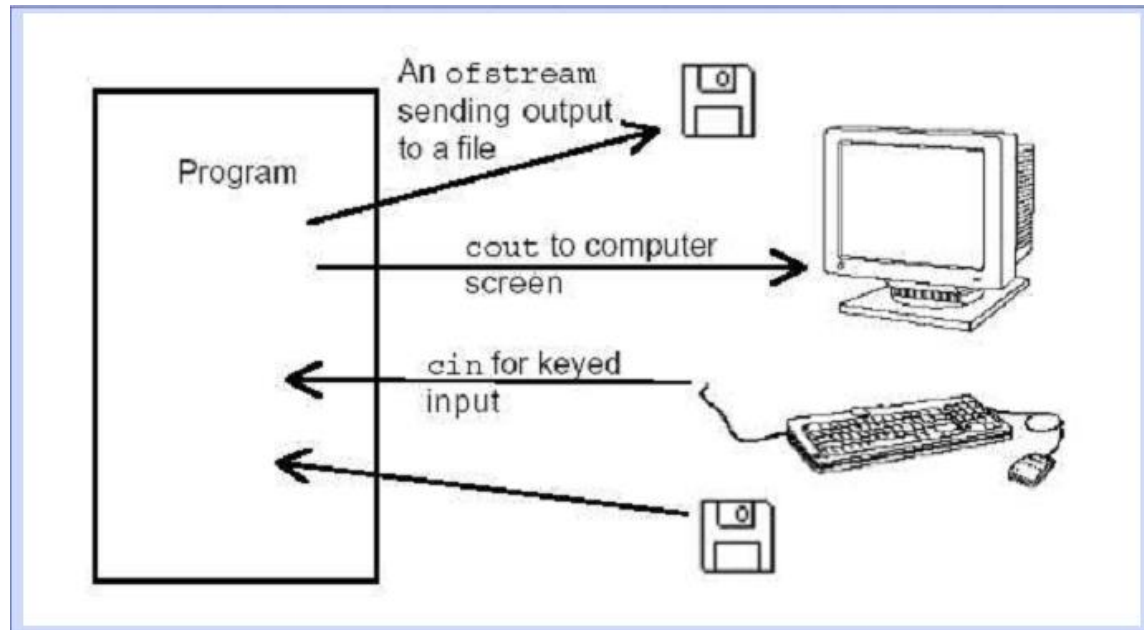
I FILES IN C++



I files in C++

Il C++ gestisce il transito di dati come un flusso:

- da o verso i dispositivi di IO
- da o verso i files



I files in C++

In C++ sono gestiti dalla libreria `<fstream>` che consente di gestire tre tipi di oggetti:

`ifstream` → per gestire l'input dal file (leggere dal file)

`ofstream` → per gestire l'output sul file (scrivere sul file)

`fstream` → per leggere e scrivere sul file (in diversi momenti)

ESEMPIO: `ifstream FILE`

Dichiarazione di
un flusso per la
lettura dal file

Apertura di un file

Per effettuare qualunque operazione di lettura, scrittura o modifica del contenuto di un file è necessario prima di tutto aprirlo (crearlo se non esiste).

Per aprire un file in C++ è necessario usare l'istruzione

```
FILE.open(“nome_file”, “modalità”);
```

Apertura di un file

```
FILE.open(“nome_file”, “modalità”);
```

FILE è l’identificativo che rappresenta il flusso di dati da o per il file.

nome_file è il nome del file che deve essere indicato con l’estensione.

Modalità

<code>ios::in</code>	aperto per la lettura
<code>ios::out</code>	aperto per la scrittura
<code>ios::ate</code>	posiziona alla fine del file
<code>ios::app</code>	apre il file in modalità “append”
<code>ios::trunc</code>	tronca il file in apertura
<code>ios::nocreate</code>	se il file non esiste non cerca di crearlo
<code>ios::noreplace</code>	l’apertura fallisce se il file esiste

Apertura di un file

Modalità

Combinazioni tipiche per l'apertura di file sono:

`ios::in | ios::nocreate` apertura se il file esiste, fallimento altrimenti

`ios::out | ios::noreplace` apre un nuovo file in output, fallimento se il file già esiste

`ios::out | ios::ate` (ri)apre in output un file esistente, aggiunge i dati alla fine, dopo quelli già presenti

`ios::out | ios::noreplace | ios::translate` apre nuovo file, fallisce se il file già esiste, traduce i caratteri di newline

`ios::out | ios::nocreate | ios::trunc` apre un nuovo file in input, fallimento se il file non esiste, elimina dati precedentemente contenuti

Chiusura di un file

FILE.close();

**Il file FILE potrà essere poi riaperto
anche con un'altra modalità.**

Fine del file

La fine dei file è segnalata da EOF (End of File) che, nei diversi SO, corrisponde a diverse sequenze di caratteri.

Il metodo

FILE.eof();

Restituisce un booleano: TRUE se il file è finito, FALSE altrimenti.

Scrittura file

Dopo aver generato il flusso sul file in modalità scrittura o append, è possibile inserire dei dati attraverso l'operatore >>

FILE >> variabile;

Lettura file

La lettura, invece, avviene grazie all'operatore <<

FILE << variabile;

Lettura e scrittura file ad accesso sequenziale

Esempio:

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

```
using namespace std;
```

```
main()
```

```
{ fstream FILE;
```

```
string nome="Mario«
```

```
string nome2;
```

```
int voto = 30, voto2;
```

```
FILE.open("prova.txt",ios::out);
```

```
FILE << nome << "\n";
```

```
FILE << voto;
```

```
FILE.close();
```

```
FILE.open("prova.txt",ios::in);
```

```
FILE >> nome2;
```

```
FILE >> voto2;
```

```
cout << nome2 << endl;
```

```
cout << voto2;
```

```
FILE.close();
```

```
}
```