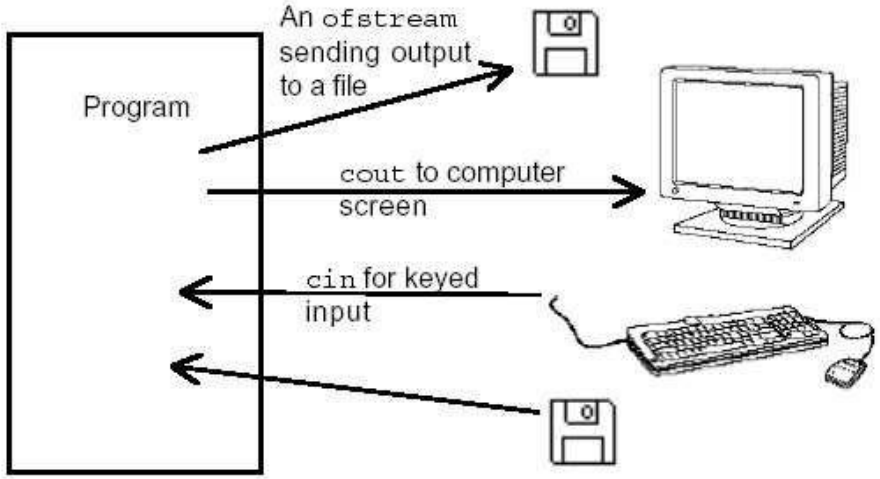
Ripasso(?) I file in C++

Emanuele Ing. Benatti - ebenatti@fermimn.edu.it

Abbiamo già usato i file di fatto..



Le funzioni cin e cout di fatto utilizzano file: la funzione cin intercetta un file che intercetta l'input della tastiera, cout un file che va a scrivere nello standard output (lo schermo del nostro terminale)

La libreria fstream in C++

- Se un programma ha bisogno di elaborare dati da/per file, deve dichiarare oggetti di tipo "filestream".
- La libreria <fstream> contiene le definizioni per tali oggetti.
- Gli oggetti filestream sono di tre tipi:
  - ifstream --- per leggere dati da file (input file stream)
  - ofstream --- per scrivere dati su file (output file stream)
  - fstream --- per leggere e scrivere dati (in differenti momenti) su file

Nota bene: si può usare fstream anche se si fa o solo input o solo output di dati da file.

Esempio di apertura in input di file di testo

```
#include <fstream> // file per i filestream
int main()
{
  ifstream input("test.txt", ios::in); // costruttore
  // accettato anche nella versione 2
  ifstream input2;
  input2.open("file1.txt", ios::in); // uso metodo open ...
  ...
}
```

Lettura di un file di testo

```
#include <fstream>
void main()
{
  ifstream input("test.txt", ios::in);
  long l1, l2; double d3; char ch;
  ...
  input >> d3;  // leggo un double
  ...
  input >> ch;  // leggo un char
  ...
  input >> l1 >> l2;  // leggo due interi
  ...
  input.close();
}
```

Attenzione: di fatto input è una variabile usabile come cin con l'operatore >> . In questo caso i tipi di dati sono importanti, bisogna sapere com'è formattato il file di testo.

Apertura e scrittura di un file di testo

```
#include <fstream>
#include <ctime>
#include <cstdlib>
void main()
{
   ofstream out1("risultati.txt", ios::out); // creo il file
   srand(time(NULL));
   int i; int d;
   out1 << "I risultati sono :" << endl;
   for(i=0;i<100; i++) {
      d=rand()%100;
      out1 << "i : " << i << ", di " << d << endl; // invio un numero casuale
      }
   out1.close();  // chiusura del file.
}
```

Stato di un fstream

Supponendo di aver aperto in lettura un file chiamato input come variabile
ifstream input("indata.txt", ios::in);

si può testare Io stato di input con i metodi:

input.good() ritorna “true" se il file è accessibile e non ci sono stati errori

input.bad() ritorna “true” se c’è stato un errore in input o output.

input.eof() ritorna “true" se siamo arrivati in fondo al file

Scrittura e lettura per carattere

Come con getchar e putchar in C si può scrivere su file carattere per carattere con la funzione get e put.

```
#include <iostream>      // std::cin, std::cout
#include <fstream>       // std::ofstream
int main () {
  std::ofstream outfile ("test.txt");
  char ch;
  std::cout << "Digita una stringa, digita il punto per terminare”\n";
  do {
    ch = std::cin.get();
    outfile.put(ch);
  } while (ch!='.');

  return 0;
```

Modalità di apertura

ios::in aperto per la lettura
ios::out aperto per la scrittura

ios::ate posiziona alla fine del file

ios::app apre il file in modalità “append”

ios::trunc tronca il file in apertura

ios::nocreates e il file non esiste non cerca di crearlo

ios::noreplace l’apertura fallisce se il file esiste

ios::binary l’apertura del file è in modalità binaria

Modalità di apertura combinate

ios::in | ios::nocreate apertura se il file esiste, fallimento altrimenti , **opzione di default**

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, non elimina il file.

ios::out | ios::noreplace | ios::binary apre nuovo file, fallisce se il file già esiste, apre il file in modalità binaria.

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

File binari , strutture e classi

Anche in C++ posso salvare strutture dati complesse in file attraverso i file binari.

Esempio di apertura in lettura `infile.open("dati.dat", ios::binary | ios::in);`
Esempio di apertura in scrittura `infile.open("dati.dat", ios::binary | ios::out);`

Esempio

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
class Persona{
        public:
    string name;
    string professione;
}
```

```
int main()
{
    fstream file;
    file.open("binario.dat", ios::out|ios::binary);
    if(!file.is_open())
    {
        cout<<"Impossibile aprire il file n";
        return 0;
    }
    Persona obj;
    obj.name = "Emanuele Benatti";
    obj.professione = "Insegnante";
    file.write((char*)&obj, sizeof(obj));
    file.close();
    return 0;
}
```