Université Ibn Tofail
جامعة ابن طفيل
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏ ⴱⵓⵃⵟⴼⴰⵢⵍ
Ibn Tofail University

POO sous C++
Flux Entrées Sorties - Fichiers -

Med. AMNAI

Filière SMI - S5

Département d'Informatique

21 janvier 2021

Plan

Plan

Plan

Plan

Plan

Principe

- La **lecture** et l'**ecriture** depuis et dans un **fichier** peuvent se faire via les opérateurs « et ».
- Il faut par contre les utiliser avec des **objects** qui représentent des **flux** vers les fichiers et non pas avec **cout** ou **cin**.
- Un **flux** (canal de données) est un moyen de communication entre un **programme** et l'**extérieur**.
- La bibliothèques **fstream** définit les classes nécessaires (**ofstream, ifstream**) pour les flux des fichiers. Il faut donc commencer par l'inclure.

#include<fstream>

Principe

- En réalité les **flux** sont des **objets**.
- Un **flux sortant**, est un flux permettant d'**écrire dans un fichier**.
- On déclare un flux exactement de la même manière qu'une variable, une variable (**monFlux**) dont le type serait **ofstream** et dont la valeur ("D :/temp/test.txt") serait le chemin d'accès du fichier à lire :
  ofstream monFlux("D :/temp/test.txt") ;
- Les règles à suivre pour le choix du **nom du flux** :
  - Uniquement de *lettres*, de *chiffres* et du **tiret-bas** "_" ;
  - Premier caractère une *lettre* (majuscule ou minuscule) ;
  - on ne peut **pas** utiliser d'**accents** ;
  - on ne peut **pas** utiliser d'**espaces**.

Ecriture dans les fichiers

La classe **ofstream** permet d'écrire dans un fichier.

```
#include<iostream>
#include<fstream>
using namespace std;
                    // wfile.cpp
int main(){
    ofstream monFlux("D:/temp/test.txt");

    if(monFlux){

        monFlux << "Je suis un test de phrase." << endl;
        monFlux << 45.23 << endl;
        int age=40;
        monFlux << "J'ai " << age << " ans. " << endl;
    }
    else {
        cout << " Erreur." << endl;
    }
    monFlux.close();
    return 0;
}
```

Ecriture à la fin d'un fichier (Mode io : :app)

```cpp
#include<iostream>
#include<fstream>
using namespace std;
                    // woefile.cpp
int main(){

    string const nomFichier("D:/temp/test.txt");

    ofstream monFlux(nomFichier.c_str(), ios::app);


    if(monFlux){

        monFlux << "Je suis un test de phrase." << endl;
        monFlux << 45.23 << endl;
        int age=40;
        monFlux << "J'ai " << age << " ans. " << endl;
    }
    else {
        cout << " Erreur." << endl;
    }
    monFlux.close();
    return 0;
}
```

Lecture d'un fichier (1)

La classe **ifstream** permet de lire un fichier.

```
#include<iostream>
#include<fstream>
using namespace std;
                    // rfile.cpp
int main(){
    ifstream monFlux("D:/temp/test.txt");

        string mot;

        monFlux >> mot ; // On lit un mot depuis le fichier

        cout << mot <<endl;

        monFlux.ignore(); // On change vers le mode ligne par ligne

        string ligne;

        getline(monFlux, ligne); // On lit une ligne complète

        cout << ligne;

        return 0;
}
```

Lecture d'un fichier (2)

- Ligne par ligne, en utilisant **getline() ;**
- Mot par mot en utilisant »
- Caractère par caractère, **get() ;**

```
char a ; monFlux.get(a) ;
```

Lecture d'un fichier (3)

```
#include<iostream>
#include<fstream>
using namespace std;
                    // rAllfile.cpp
int main(){

    ifstream monFlux("D:/temp/test.txt");
    if(monFlux)
    {
      //L'ouverture s'est bien passée, on peut donc lire

      string ligne;

      while(getline(monFlux, ligne)) //Tant qu'on n'est pas à la fin, on lit
      {
        cout << ligne << endl;
      }
   }
   else
   {
      cout << "ERREUR !! d'ouverture du fichier en lecture." << endl;
   }
    return 0;
}
```

Modes d'ouverture

- **ios : :in** Fichier ouvert en lecture.
- **ios : :out** Fichier ouvert en écriture.
- **ios : :app** Ajoute les données, en écrivant toujours à la fin.
- **ios : :ate** Aller à la fin du fichier à l'ouverture.
- **ios : :trunc** Supprime le contenu du fichier, s'il existe déjà ; cette suppression est automatique pour les fichiers ouverts en écriture, sauf si **ios : :ate** ou **ios : :app** a été précisé dans le mode.
- **ios : :binary** Fichier binaire, ne faire aucun formatage.

Ex : fstream fl("exemple.cpp", ios : :in|ios : :out|ios : :app) ;
ouvre le fichier *exemple.cpp* en **lecture** et **écriture**, avec **ajout** des nouvelles données **à la fin**.

Positions dans les fichiers

```
int current_pos = ifstream.tellg(); // Position actuelle en lecture
int current_pos = ofstream.tellp(); // Position actuelle en écriture

ifstream.seekg ( position_souhaitee );
ofstream.seekp ( position_souhaitee );

ifstream.seekg ( position_souhaitee, ios::end );
ofstream.seekp ( position_souhaitee, ios::cur );
```

ios::beg	offset counted from the beginning of the stream
ios::cur	offset counted from the current position
ios::end	offset counted from the end of the stream

- Se placer 10 caractères **après le début** du fichier :**flux.seekp(10, ios : :beg) ;**.
- Aller 20 caractères **plus loin** que l'endroit où se situe le curseur : **flux.seekp(20, ios : :cur) ;**.

Exemple

```cpp
#include<iostream>
#include<fstream>
using namespace std;
                    // posiFile.cpp
int main(){
    streampos begin, end;

    ifstream myFile("D:/temp/linksBiblio.txt", ios::app);

    begin = myFile.tellg();

    myFile.seekg(0, ios::end);

    end = myFile.tellg();

    myFile.close();

    cout << (end-begin);

    return 0;
}
```