

Chapitre 6 Fichier séquentiel en langage C/C++

Chapitre 6 Fichier séquentiel

- I INTRODUCTION
- II LECTURE D'UN FICHIER
- III ECRITURE D'UN FICHIER
- IV BILAN

1

I Introduction

Nous allons étudier comment enregistrer des informations saisies dans un fichier sur une mémoire de masse.

Le fichier de base est un fichier texte (fichiers lisibles par un simple éditeur de texte). Cependant il est souvent nécessaire qu'un programme écrive dans un fichier ou lise les informations contenues dans un fichier texte.

2

I Introduction

En C++, on utilise la bibliothèque `fstream`, on l'inclut donc en début de programme :

```
#include <fstream>
```

3

II Ecriture dans un fichier

Pour créer un fichier :

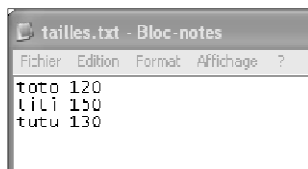
- il faut l'ouvrir en écriture,
- écrire les données dans le fichier.
- fermer le fichier.

```
ofstream monFichier("tailles.txt", ios::out); // ouverture en écriture
if (monFichier) {
    monFichier<<"toto"<<" "<<"120"<<endl;
    monFichier<<"titi"<<" "<<"150"<<endl;
    monFichier<<"tutu"<<" "<<"130"<<endl;
    monFichier.close(); // fermeture fichier
}
else
    cout<<"probleme ouverture fichier";
```

4

II Ecriture dans un fichier

Quand on ouvre le fichier `tailles.txt` avec le bloc note :



5

II Ecriture dans un fichier

Si le fichier existe déjà, il sera supprimé et remplacé. Cependant il est possible d'ajouter des données à la fin d'un fichier existant :

```
ofstream monFichier("tailles.txt",ios::app); //ouverture en écriture/ajout
if (monFichier) {
    monFichier<<"tata"<<" "<<"170"<<endl;
    monFichier<<"tyty"<<" "<<"050"<<endl;
    monFichier.close(); // fermeture fichier
}
else
    cout<<"probleme ouverture fichier";
```

6

Chapitre 6 Fichier séquentiel en langage C/C++

II Ecriture dans un fichier

Quand on réouvre le fichier tailles.txt avec le bloc note après l'exécution du code précédent :



```
tailles.txt - Bloc-notes
Fichier  Edition  Format  Affichage  ?
toto 120
titi 150
lulu 130
tata 170
tyty 050
```

7

III Lecture fichier

Pour lire des données écrites dans un fichier :

- on l'ouvre en lecture.
- on lit les données en provenance du fichier.
- on ferme le fichier.

```
int taille;
string nom;
ifstream leFichier("tailles.txt",ios::in); // ouverture en lecture du fichier
if (leFichier) {
    leFichier>>nom>>taille; // lecture une ligne
    cout<<"Nom : "<<nom<<" Taille : "<<taille;
    leFichier.close(); // fermeture du fichier
}
```

8

III Lecture fichier

Lecture de plusieurs lignes : l'accès à une ligne est séquentiel, il faut avoir lu les lignes précédentes pour accéder à une ligne donnée.

```
string nom;
int taille;
ifstream leFichier("tailles.txt",ios::in); // ouverture en lecture du fichier
if (leFichier) {
    leFichier>>nom>>taille; // lecture 1ère ligne
    while (leFichier) {
        cout<<"Nom : "<<nom<<" Taille : "<<taille<<endl;
        leFichier>>nom>>taille; // lecture ligne suivante
    }
    leFichier.close(); // fermeture du fichier
}
```

9

III Lecture fichier

Ouverture fichier en lecture

Lecture 1^{ère} ligne

Tant que Non fin fichier

 // Traitement des informations de la ligne

 Lecture ligne suivante

Fin Tant que

Fermeture fichier

10

IV BILAN

Pour ouvrir un fichier en écriture :

On utilise le type ofstream

Pour ouvrir un fichier en lecture

On utilise le type ifstream.

Pour lire une ligne d'un fichier

nomFlux>>variable;

Pour écrire une ligne dans un fichier

nomFlux<<expression;

11

IV BILAN

Une base de données est un logiciel (ensemble de programmes) qui manipule des fichiers séquentiels tout en permettant au programmeur de bénéficier :

- d'accès direct par clé (au travers des fichiers d'index),
- de gestion des structures des lignes,
- de vérification de cohérence des données (exemple : intégrité référentielle)...

Cependant l'exploitation de fichiers de base (dits fichiers séquentiels) est utilisée pour les gestions d'ensembles restreints de données, des fichiers temporaires ou pour de très gros ensembles de données.

12