

C: TP 08 Programmation en C

DAKKAR Borhen-eddine

Lycée le Corbusier

BTS SN

January 20, 2021

1 Objectifs du TP

Dans ce TP nous allons découvrir comment interagir avec fichier externe du programme. Lire et écrire des données d'un fichier sont des actions essentielles pour un programme, nous allons pratiquer différentes techniques du langage C permettant ces opérations.

2 Logiciels à utiliser

Sous Linux les outils nécessaires à la compilation du C sont déjà présents, il s'agit de "gcc" et d'un éditeur de texte "gedit" par exemple.

3 Exercices

3.1 Exercice 1 :

1. Ecrivez un programme C qui stocke les nombres suivants dans un fichier nommé **result.dat**: 16.25, 18.96, 22.34, 18.94, 17.42, 22.63.
2. Ecrivez un programme C pour lire les données du fichier **result.dat** créé précédemment et affichez les. De plus, votre programme doit calculer et afficher la somme et la moyenne de ces données. Vérifiez la somme et la moyenne affichées par votre programme à l'aide d'un calcul manuel.

3.2 Exercice 2 :

1. Écrivez un programme C qui invite l'utilisateur à entrer cinq nombres. Le programme doit écrire le nombre entré dans un fichier nommé **utilisateur.dat**.
2. Écrivez un programme C qui lit les données du fichier **utilisateur.dat** et affiche chaque élément de données individuellement.

3.3 Exercice 3 :

1. Ecrivez un programme qui crée un fichier avec les données suivantes :

Numéro d'élèves	Niveau	Groupe
123	1ère	G1
124	1ère	G1
125	1ère	G1
126	2nde	G2
127	2nde	G2
128	2nde	G2
129	terminale	G3
130	terminale	G3
131	terminale	G3

2. Ecrivez un programme C qui récupère les données du fichier créé dans la question précédente et les affiche. L'affichage doit comprendre le numéro de l'élève, son niveau et son groupe.

3.4 Exercice 4:

Écrire un programme permettant de créer un fichier « répertoire » comportant pour chaque personne :

- son nom (15 caractères maximum) ;
- son prénom (15 caractères maximum) ;
- son âge (entier) ;

Les informations relatives aux différentes personnes seront lues au clavier ainsi que le nombre personnes à enregistrer.

Ecrivez les fonctions suivantes:

- **void Ecrire_str(FILE*nom_fich,char chaine[15]):** une fonction qui permet d'écrire une chaîne de caractères dans un fichier.
- **void Ecrire_int(FILE *nom_fich,int ag):** une fonction qui permet d'écrire un entier dans un fichier.
- **void position(FILE *nom_fich):** une fonction qui vous permet de positionner le curseur sur un emplacement désiré pour écrire.

Indication:

- **strlen(chaine)** est une fonction de <string.h>, elle permet de connaître le nombre de caractères de la variable chaîne de type **char**.

3.5 Exercice 5 :

Supposons qu'un fichier de données constitué d'un groupe de lignes individuelles. Écrivez un programme C qui lit et affiche tout numéro de ligne du fichier saisi par l'utilisateur. Par exemple lire la ligne 2 du fichier.

3.6 Exercice 6 : Mise à jour d'un fichier

Nous souhaitons mettre à jour un fichier d'inventaire à partir d'un autre fichier contenant les informations des ventes.

Le fichier à mettre à jour est appelé fichier maître (master) et le fichier contenant les données de mise à jour est appelé fichier de transactions.

Supposons que le fichier maître (master) actuel nommé **actu_livres.txt**, contient les numéros d'identification des livres et leurs quantités en stock (voir Tableau 1).

N° ID du livre	Quantité stock
125	98
289	222
341	675
467	152
589	34
622	125
722	15

Table 1: Données du fichier "actu_livres.txt"

Le fichier des transactions nommé **trans_livres.txt** contient toutes les informations de vente N°ID du livre, date, quantités, livres retournés et livres achetés (voir Tabeleau 2).

ID N.	Date	stock	Retourné	Achetés
289	1/10/92	125	34	50
341	1/10/92	300	52	0
467	1/15/92	50	20	200
467	1/20/92	225	0	160
589	1/31/92	75	10	55

Table 2: Données du fichier "trans_livres.txt"

Une solution à ce problème consiste à conserver toutes les transactions dans le même ordre des numéros d'identification que les enregistrements du fichier principal. Puisque les enregistrements des numéros d'identification du fichier maître (master) sont en augmentation croissante, les transactions doivent également être conservées par ordre croissant.

La procédure consiste à lire un enregistrement à partir du fichier maître (actu_livres.txt) et un autre à partir du fichier de transaction ("trans_livres.txt). Si les deux fichiers ont le même numéro d'identification, nous appliquons un enregistrement des informations de la transaction aux fichier principal

(maître) et une autre transaction est lue. Tant que la transaction (le numéro d'identification de l'enregistrement) correspond au numéro d'identification de l'enregistrement principal, la mise à jour du l'enregistrement du fichier maître se poursuit.

"Lorsque le numéro d'identification de l'enregistrement de la transaction ne correspond pas au numéro d'identification du fichier maître, ce qui indique qu'il n'y a plus de mise à jour à appliquer au fichier maître, un enregistrement mis à jour est écrit dans le nouveau fichier maître. Voyons comment cette procédure fonctionne avec les deux fichiers affichés dans les tableaux 8-4 et 8-5. "

Le nouveau fichier maître peut être un fichier complètement nouveau, dans notre procédure de mise à jour, nous allons créer un nouveau fichier maître appelé "mise_jour_livres.txt" afin qu'on puisse garder les données d'origine dans le l'ancien fichier maître.

Le pseudocode suivant résume les opérations nécessaires pour effectuer cette

Algorithm 1: Mise à jour d'un fichier de données

Début

Ouvrir le fichier "actu_livres.txt"

Ouvrir le fichier "mise_jour_livres.txt" (initialement vide)

Ouvrir le fichier des transactions "trans_livres.txt"

Lire le premier enregistrement du fichier "actu_livres.txt"

Tant que (pas fin du fichier) **faire**

 Lire l'enregistrement de la transaction

Tant que (N°ID transaction diff N°ID maître) **faire**

Mettre à jour l'enregistrement dans "mise_jour_livres.txt"

Lire l'enregistrement suivant du fichier "actu_livres.txt"

Fin Tq

Calculer le nouveau stock

Fin Tq

//– Le dernier enregistrement de transaction qui vient d'être lu doit être mis dans "mise_jour_livres.txt"–/

Ecrire la dernière mise à jour dans "mise_jour_livres.txt"

Tant qu'il reste des enregistrements dans l'ancien fichier maître **faire**

Ecrire un nouveau enregistrement dans "mise_jour_livres.txt"

Lire un ancien enregistrement du "actu_livres.txt"

Fermez tous les fichiers

FIN

Le calcul du nouveau stock est donné par :

$$stock = Quantité_stock + N_livre_achetés - stock + N_livre_retournés$$

Traduisez le pseudocode en programme C.

mise à jour.