

# Examen partiel de système (FLIN502)

Michel Meynard

Durée : 1h30

Tous documents autorisés

22 novembre 2009

## 1 Recherche dichotomique dans un fichier trié et structuré en lignes

On utilise des fichiers structurés en en lignes composées de :

- un champ de 8 char : **nom** de personne;
- un champ de 4 car : **valeur** entière;
- un retour à la ligne sur 1 char : `'\n'`;

Ces fichiers sont triés en ordre croissant sur le premier champ de nom et sont **non bornés** (il est donc impossible de les recopier en mémoire centrale dans un tableau)! Un exemple d'un tel fichier `compte.txt` est donné ci-dessous :

```
albert 1234
bernard 0012
claudia 0450
pierre 7894
pierrett0012
```

On souhaite écrire un programme affichant la valeur d'une ligne dont le nom est passé en paramètre :

```
$valeur compte.txt pierrett
12
$
```

### 1.1 Questions

On supposera dans ce qui suit que le processus courant est le seul à accéder au fichier.

1. Quelles sont les positions de début de ligne dans ce type de fichier structuré?
2. Comment connaître la position de début de la dernière ligne?
3. Ecrire un algorithme de recherche dichotomique d'une ligne correspondant à un nom : on suppose que chaque nom est unique dans le fichier et qu'il est constitué de 8 char exactement.
4. Ecrire le programme C `valeur.c` correspondant à cet algorithme.
5. On veut maintenant afficher la somme des valeurs des lignes associée à un préfixe de nom. Par exemple :

```
$sommevaleur compte.txt pier
7906
$
```

De cette façon, on pourra également gérer les homonymes du fichiers (2 personnes ayant le même nom). Ecrire l'algorithme.

6. Ecrire le programme C `sommevaleur.c` correspondant à cet algorithme.

## 2 Processus

Un processus parent doit :

- ouvrir un fichier `f` passé en paramètre à la ligne de commande;
- créer deux processus enfants, nommés avant et arrière;
- attendre la fin de ses enfants et afficher un message avant de mourir;

Chaque processus enfant doit :

- afficher un message indiquant sa naissance;
- rechercher un char donné en paramètre par le parent dans le fichier `f` :
  - soit en marche avant (du début vers la fin);
  - soit en marche arrière (de la fin vers le début);
- afficher un message indiquant la position du char trouvé;
- afficher un message indiquant sa fin;

Par exemple, une exécution possible sur le fichier `test.txt` contenant `azertyuiopqsdfg` sera :

```
$recherche test.txt e
avant: naissance du pus avant
arriere: naissance du pus arriere
avant: caractère e trouvé en position 2 dans test.txt
avant: fin du pus avant
arriere: caractère e trouvé en position 2 dans test.txt
arriere: fin du pus arriere
parent: fin du pus parent
```

Le fichier est **non borné** (il est donc impossible de le recopier en mémoire centrale dans un tableau)!

## 2.1 Questions

1. Les accès concurrents des processus au même fichier sont-ils possibles? Dans le cas positif, sont-ils efficaces?
2. Comment le processus “avant” parcourt-il le fichier dans sa recherche séquentielle?
3. Comment le processus “arriere” parcourt-il le fichier dans sa recherche séquentielle?
4. Comment le parent transmet-il les informations utiles à ses enfants?
5. Ecrire un algorithme de recherche.
6. Ecrire le programme C `recherche.c` correspondant à cet algorithme.
7. Cela pose-t-il un problème si le caractère recherché existe en plusieurs exemplaires dans le fichier?