



## Devoir surveillé du 13 juin 2007

CSH : Initiation au C et au shell  
Première année



Tous documents interdits. Les exercices sont indépendants. La correction tiendra compte de la qualité de la rédaction et de la présentation. Barème approximatif.

- ★ **Exercice 1.** On souhaite écrire une fonction qui échange le contenu de deux zones mémoires de taille  $x$  et d'adresses respectives  $a$  et  $b$ . Si les deux blocs se recouvrent, le comportement de la fonction n'est pas défini. Le prototype de la fonction est le suivant : `void memex(void *a, void *b, size_t x);`

▷ **Question 1.** (3 pts) Écrivez cette fonction.

- ★ **Exercice 2.** Lecture d'une chaîne de caractères (7 pts)

Dans cet exercice, nous nous intéressons aux fonctions `gets` et `fgets`. La fonction `gets` permet de lire une ligne (tous les caractères jusqu'au premier `'\n'` inclus ou bien jusqu'à la fin de fichier) sur l'entrée standard (le clavier en général) et place le résultat (en y ajoutant un `'\0'` final) en mémoire à l'adresse donnée `str` en paramètre. Cette fonction renvoie NULL si elle n'a lu aucun caractère (fin de fichier) et dans ce cas la mémoire pointée par `str` reste inchangée. Dans tous les autres cas, elle renvoie le pointeur qui lui est passé en paramètre. Son prototype est le suivant : `char *gets(char *str);`

La fonction `fgets` est quasiment identique à `gets` si ce n'est qu'elle lit les caractères depuis un flux passé en paramètre et qu'elle lit au plus `size` moins un caractères. Elle renvoie NULL si elle n'a lu aucun caractère et dans ce cas la mémoire pointée par `str` reste inchangée. Son prototype est le suivant :

`char *fgets(char *str, int size, FILE *stream);`

▷ **Question 1.** (2 pts) Peut-on écrire `fgets` en utilisant `gets`? `gets` en utilisant `fgets`? Justifiez vos réponses soit en expliquant pourquoi c'est impossible, soit en écrivant la fonction répondant à la question.

▷ **Question 2.** (1 pt) Pourquoi est-il fortement déconseillé d'utiliser la fonction `gets` bien qu'elle fasse partie de la bibliothèque standard du C?

▷ **Question 3.** (4 pts) Écrivez la fonction `fgets` en utilisant la fonction `fgetc` qui permet de lire un caractère sur un flux donné : `int fgetc(FILE *stream)`.

- ★ **Exercice 3.** (5pts) Soit un fichier regroupant les notes d'étudiants sous la forme suivante :

```
1 bob123 12 16
2 henri4 7 14
3 leon43 4 1
```

La première colonne est le login unix de l'étudiant, la seconde est la note en projet et la troisième est la note à l'examen final. Les colonnes sont séparées par une seule espace.

▷ **Question 1.** ( $\frac{1}{2}$  pt) Écrire un script shell listant tous les étudiants ayant eu 20 en projet.

▷ **Question 2.** ( $\frac{1}{2}$  pt) Écrire un script shell donnant la note de projet de l'étudiant toto42.

▷ **Question 3.** (2 pts) Écrire un script shell calculant la moyenne à l'examen final.

**1 point de bonus** si votre solution calcule les moyennes avec un chiffre après la virgule en n'utilisant que `expr` pour faire les calculs.

*Indication :* `expr 10 / 4` donne le résultat de la division entière entre 10 et 4 (c'est-à-dire 2).

`expr 10 % 4` donne le reste de cette division (c'est-à-dire 2 également).

▷ **Question 4.** (2 pts) Écrire un script shell calculant la moyenne de chaque étudiant.

- ★ **Exercice 4.** (5 pts) Écrire un script shell `menage dir` proposant d'effacer tous les fichiers du répertoire `dir` indiqué dont les noms se terminent par `~` ou par `.BAK` ou `.bak`. Pour chaque fichier, le script doit demander à l'utilisateur la confirmation de la suppression avant de l'effacer.

▷ **Question 1.** (2 pts) Écrivez ce script.

*Indication :* `read toto` permet de lire une chaîne de caractères du clavier et de la stocker dans la variable shell `toto`.

▷ **Question 2.** (2 pts) Modifiez votre script pour parcourir récursivement les sous-répertoires de `dir`.

▷ **Question 3.** (1 pt) Modifiez votre script pour qu'il ne pose la question qu'une fois après avoir présenté la liste de tous les fichiers qui seraient écrasés.