

Programmation Système

Juliusz Chroboczek

9 Janvier 2009

La durée de l'examen est de 3 heures. Les documents sont autorisés, le matériel électronique est interdit.

Tous les programmes demandés devront être écrits en C, et les solutions dans d'autres langages de programmation ne seront pas acceptées. La notation prendra en compte la qualité de la gestion des situations exceptionnelles (« erreurs ») ou l'absence de celle-ci.

Les parties sont indépendantes. Il est fortement conseillé de lire l'énoncé, qui consiste de 3 pages.

1 Questions de cours

Quel est le contenu du fichier toto après l'exécution de chacun des fragments de code suivants ? Vous n'avez pas à justifier votre réponse. (On supposera les variables fd et fd2 déclarées et tout le reste de la bureaucratie habituelle effectué.)

1. `fd = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
write(fd, "Yo!", 3);
close(fd);`

Yo!

2. `fd = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
write(fd, "Yo!", 3);
lseek(fd, -3, SEEK_CUR);
write(fd, "X", 1);
close(fd);`

Xo!

3. `fd = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
write(fd, "Yo!", 3);
lseek(fd, 1, SEEK_END);
write(fd, "X", 1);
close(fd);`

Yo! X

4. `fd = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
fd2 = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
write(fd, "Yo!", 3);
write(fd2, "X", 1);
close(fd);`

↓
3

~~Yo!~~

Xo!

```

5.  fd = open("toto", O_CREAT | O_WRONLY | O_TRUNC, 0666);
    fd2 = dup(fd);
    write(fd, "Yo!", 3);
    write(fd2, "X", 1);
    close(fd);

```

~~Yo!~~ ~~XYo!~~
~~Xo!~~ Yo!X

2 Entrées/sorties fichiers

Dans cet exercice, l'utilisation de la bibliothèque `stdio` n'est pas autorisée, sauf pour présenter des messages d'erreur à l'utilisateur.

Écrivez un programme qui prend en paramètre de ligne de commande deux noms de fichiers, *f1* et *f2*, et qui copie *f1* vers *f2* en inversant chaque paire d'octets adjacents. Par exemple, si le fichier *f1* contient la suite d'octets

(1, 2, 3, 4),

alors le fichier *f2* devra, après exécution de votre programme, contenir la suite d'octets

(2, 1, 4, 3).

Dans le cas où *f1* contient un nombre impair d'octets, votre programme devra inverser tous les octets sauf le dernier, et copier le dernier octet inchangé. Par exemple, si le fichier *f1* contient la suite d'octets

(1, 2, 3, 4, 5),

alors le fichier *f2* devra, après exécution de votre programme, contenir la suite d'octets

(2, 1, 4, 3, 5).

Si le fichier *f2* existe déjà, votre programme ne devra pas écraser celui-ci, mais arrêter son exécution avec un message d'erreur compréhensible par un être humain normal. (*Indication : un message fourni par la fonction `perror` n'est pas compréhensible par un être humain normal.*)

3 Processus

Sur ma machine, j'ai un programme « `/usr/local/bin/loadavg` » qui affiche sur sa sortie standard la charge moyenne de la machine¹, présentée comme un nombre en virgule flottante. Par exemple, en ce moment,

```

$ loadavg
0.1

```

¹ La charge moyenne (*load average*) est définie comme le nombre de processus dans un des états *en cours d'exécution* ou *prêt à s'exécuter*, divisé par le nombre de processeurs de la machine, moyenné sur la dernière minute. Mais vous n'avez pas besoin de savoir ça pour cet exercice.

1. Écrivez un programme qui affiche « C'est le moment de faire un gros calcul » si la charge moyenne est inférieure à 0.2, et « C'est pas le moment » sinon. Votre programme ne devra pas afficher la charge moyenne. De plus, il ne devra exécuter aucun programme autre que « loadavg » ; en particulier, il ne devra pas se servir de la fonction `system` (qui exécute un shell).

Vous pourrez vous servir de la fonction

```
double atof(char *s)
```

qui interprète une chaîne de caractères et retourne le nombre en virgule flottante qu'elle représente.

2. Malheureusement, le programme `loadavg` est boggué : si la charge du système est très élevée, il lui arrive de se bloquer indéfiniment sans jamais rien afficher.

Modifiez votre programme pour que, lorsque le programme `loadavg` n'a pas produit de sortie après 10 secondes, il affiche « C'est pas le moment » et termine.

2