

Programmation Système

TP 4 : détails des processus

Juliusz Chroboczek
Matthieu Boutier

15 octobre 2012

Exercice 1 (Rappels sur `lseek`). Écrivez un programme qui prend en paramètre de ligne de commande deux noms de fichier f et g , puis recopie f en g en inversant l'ordre des octets qui le constituent : si f consiste de la suite d'octets $a_1 \cdots a_n$, alors g devra contenir la suite d'octets $a_n \cdots a_1$.

Votre programme devra utiliser une quantité bornée de mémoire, mais il devra faire des entrées sorties de taille raisonnable — un tampon de 4096 octets est un bon choix.

Exercice 2. Pour chacun des fragments de code suivants, indiquez quel est le contenu final du fichier f . Vérifiez votre hypothèse à l'aide de la commande `hexdump -C`¹.

- ```
1. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
 write(fd, "Ouille", 6);
 lseek(fd, 2, SEEK_SET);
 write(fd, "Aie", 3);
 close(fd);
```
- ```
2. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
   write(fd, "Ouille", 6);
   lseek(fd, 2, SEEK_CUR);
   write(fd, "Aie", 3);
   close(fd);
```
- ```
3. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
 fd2 = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
 write(fd2, "Ouille", 6);
 write(fd, "Aie", 3);
 close(fd);
 close(fd2);
```
- ```
4. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
   fork();
   write(fd, "Ouille", 6);
   close(fd);
```
- ```
5. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
 pid = fork();
 if(pid == 0)
 lseek(fd, 2, SEEK_SET);
```

---

1. J'ai omis le code de gestion des erreurs de ces fragments de code. J'ai le droit — vous pas.

```

else
 sleep(2);
write(fd, "Ouille", 6);
close(fd);
6. fd = open("exemple", O_CREAT | O_WRONLY | O_TRUNC, 0666);
fd2 = dup(fd);
write(fd2, "Ouille", 6);
write(fd, "Aie", 3);
close(fd);
close(fd2);

```

### Exercice 3.

1. Écrivez une fonction `premier` qui teste si l'entier passé en paramètre est premier en utilisant l'algorithme naïf. Écrivez un programme qui utilise cette fonction pour afficher la liste des entiers de la forme  $2^n - 1$  qui sont premiers, pour  $n$  allant de 1 à 31.
2. Modifiez votre programme pour que chaque appel à `premier` se fasse dans un processus séparé; vous utiliserez la valeur de retour du processus pour indiquer si l'entier était premier.
3. Modifiez votre programme pour qu'il garantisse que l'affichage des résultats se fait dans l'ordre croissant; vous utiliserez pour cela un appel à `waitpid` avec un premier paramètre non nul.

**Exercice 4** (S'il vous reste du temps). Modifiez le programme du premier exercice ci-dessus pour qu'il inverse l'ordre des lignes du fichier plutôt que l'ordre des octets.