

# Programmation Système

## Verrous

Juliusz Chroboczek  
Khouloud Zine Elabidine

3 décembre 2012

### Exercice 1 (Introduction aux verrous).

1. Écrivez un programme qui ouvre le fichier `/tmp/compteur`, lit un entier  $n$  qui y est contenu, puis le remplace par l'entier  $n + 1$ , et recommence cent mille fois<sup>1</sup>. Créez manuellement un fichier `/tmp/compteur` qui contient l'entier 0, et exécutez votre programme. Vérifiez que le fichier `/tmp/compteur` contient bien 100000.
2. Recommencez l'expérience en exécutant votre programme deux fois simultanément. Que constatez-vous ?
3. Modifiez votre programme pour qu'il corrige le problème à l'aide de l'appel système `flock`<sup>2</sup>. (Pensez au type de *lock* qu'il faudra utiliser, et quelle sera sa portée.)

**Exercice 2 (Agenda).** On se propose d'écrire un programme qui manipule un agenda. L'agenda sera stocké dans un fichier `~/agenda.dat` qui aura la forme suivante :

```
1354483503 Écrire le TP 10
1354527000 Aller assurer le TP 10
1354534200 Aller déjeuner
```

Le fichier est constitué d'une suite triée d'événements, un par ligne. Chaque événement est représenté par sa date en secondes depuis l'Époque (comme la valeur retournée par `time`), suivi d'une description de l'événement.

1. Écrivez un programme `affiche-agenda` qui affiche le contenu de l'agenda sous une forme lisible par un être humain (vous pourrez vous servir des fonctions `strftime` et `localtime`) sans omettre de verrouiller le fichier à l'aide de la fonction `flock`. Deux instances de `affiche-agenda` peuvent-elles accéder au fichier en même temps ? Vérifiez votre hypothèse en ajoutant un appel à `sleep` au bon moment.
2. Modifiez votre programme `affiche-agenda` pour que lorsqu'il est invoqué avec l'option `-1` il n'affiche que le prochain événement.
3. Écrivez un programme `ajoute-evenement` qui ajoute un événement passé en paramètre de ligne de commande. (Vous pourrez vous servir de la fonction `strftime`.) Une instance de `ajoute-evenement` peut-elle accéder au fichier en même temps qu'une instance de `affiche-agenda` ? Comme ci-dessus, vérifiez en ajoutant des appels à `sleep` aux endroits judicieux.

---

1. Si vous utilisez `stdio` plutôt que des appels systèmes, il faudra penser à utiliser la fonction `fflush` au bon moment.

2. L'auteur de ces lignes refuse d'utiliser les *locks* POSIX.