

STI 1^{ère} année – Programmation Système

TD 5: Verrou

J. Briffaut - J.-F. Lalande - C. Toinard

1 Verrou

1.1 Verrouillage et deadlock

Exercice 1 Créez les macros *read_lock*, *readw_lock*, *write_lock* et *writew_lock* permettant de poser des verrous de façon non bloquante ou bloquante en lecture et en écriture. Ces macros utilisent la même fonction C que vous écrivez et dont le prototype est :

```
int lock_reg (int fd, int lockCmd, int lockType, off_t regRffset, int
    regWhence, off_t regLen);
```

Vous devrez utiliser une structure *flock* et la fonction *fcntl* dans la fonction *lock_reg*.

Exercice 2 Ecrivez un programme *lockfile* qui crée un fichier appelé *templock1* vide accessible en lecture et écriture pour le propriétaire, lecture pour le groupe et les autres. Puis il écrit la chaîne "ab" dans le fichier.

Exercice 3 Ajoutez une fonction **static void** *lockabyte*(**const char** *name, **int** fd, **off_t** offset) qui pose un verrou bloquant en écriture sur le fichier identifié par un descripteur de fichier *fd* à l'offset donné. *name* correspond au nom du programme appelant est sera utilisé pour afficher des informations sur la prise du verrou.

Exercice 4 Ajoutez à votre programme *lockfile* un appel à *lockabyte* puis utilisez la fonction *sleep()* pour faire attendre votre programme 30 seconde avant de libérez le verrou.

Exercice 5 Ecrire un second programme *trytolockfile* qui ouvre le fichier *templock1* en lecture et qui essay de poser un verrou en lecture. Testez vos de programme en lançant dans une première console *lockfile*, puis lancer *trytolockfile* dans une seconde console.

Exercice 6 Modifier les programmes *lockfile* et *trytolockfile* afin de tester les différentes combinaisons de verrous bloquant/non bloquant en lecture/ecriture.