TINAZZI Giovanni

RAPPORT TP4

Objectifs

Dans ce TP, le but était de créer un verrou d'enregistrement interactif, c'est-à- dire que l'on peut poser un verrou et l'enlever sur une partie d'un fichier. Un verrou permet de s'assurer qu'une seule personne, ou un seul processus accède à un instant donné. En quelque sorte, cela vas nous permettre de faire un sorte qu'un seul utilisateur/processus puisse utiliser la partie ou le verrou a été placé.

Deux verrous peuvent etre utilises:

> - flock: un verrou facultatif et pas obligatoire
> - fcntl: un verrou qui peut etre soit facultative soit obligatoire

Input et parsing

Etant donné que on doit donner le choix à l'utilisateur de quel type de verrou il veux utiliser et sur quelles lignes de son fichier il veux que le verrou agisse, on va devoir recuperer l'input de l'utilisateur pour ensuite parser l'input (diviser l'input en quelque sorte).

Tout d'abord, on ouvre le fichier de la facon suivante:

```
 int fd = open(argv[1], O_RDWR); // cette variable va nous permettre de
ouvrir le fichier voulu
```

Apres avoir ouvert le fichier voulu et avoir tester qu'il s'est ouvert sans probleme, on va rentrer dans une boucle `for(;;)`. En effet, il est important de faire cette boucle car sinon on va pas pouvoir modifier l'action qu'on a fait. Dans cette boucle, la premiere chose à faire est de recuperer l'input:

```
fgets(input, sizeof(input), stdin);
```

Apres avoir recuperer l'input, on vas le comparer (on utilise la fonction strcmp) pour voir lequel de nos 3 possibilites correspond cet input:

> - "?\n" : l'utilisateur se demande comment marche le programme
> - "exit" : l'utilisateur souhaite quitter le programme
> - si aucun des deux est tape, l'utilisateur sait deja utiliser le programme et souhaite justement l'utiliser

C'est dans la derniere des options qu'on vas pouvoir separer l'input de la facon suivante:

```
sscanf(input, "%c %c %d %d %c", &cmd_input, &ltype_input, &start_input,
&lenght_input, &whence_input);
```

Apres avoir separer l'input, on vas pouvoir utiliser nos parametres d'input dans la structure de fl.

Partie locking

Dans cette partie, on va s'occuper de placer ou enlever un verrou. On a 3 types de verrous en C:

- F_RDLCK: un verrou partages, plusieurs verrous peuvent etre poses
- F_WRLC: un seul verrou peut etre pose a la fois. Celui la va etre utile lorsque un seul processus veux pouvoir modifier cette partie du fichier
- F_UNLCK: on debloque un verrou existant

pour poser/enlever un verrou, on utilise l'appelle systeme suivant:

```
fcntl(fd, cmd, &fl);
```

Maintenant, on a la possibilte d'utiliser 3 types de commandes selon ce qu'on veux faire:

- F_SETLK: permet de poser/enlever un verrou et retourne immmediatement
- F_SETLKW: permet de poser/enlever un verrou et attend en cas de conflit
- F_GETLK: permet d'obtenir des informations sur un verrou

Enfin, on va devoir aussi gerer les erreurs. J'ai choisi d'utiliser errno avec ces 3 types d'erreurs:

- EINVAL: l'argument est invalide
- EACCES ou EAGAIN: on peux pas acceder au lock (cas ou un autre processus a deja place un verrou)

Questions

Que se passera-t-il si nous déverrouillons un fichier (ou une partie du fichier) qui n’est pas verrouillé?

Si nous déverrouiller un fichier qui n’est pas verrouillé, il ne vas rien se passer étant donné que notre fichier est deja considere comme déverrouillé vu qu' aucun verrou agit sur lui.

Que se passera-t-il si nous mettons un nouveau verrou sur une section déjà verrouillée? Le type de verrou changera-t-il le résultat? Expliquer dans la situation avec le même processus et avec 2 processus différents.

- Avec le meme processus, si nous mettons un nouveau verrou sur une section déjà verrouillée, alors le verrou sur cette section va se mettre à jour

- Avec un processus different, cela depend du type de verrou. Si c'est un verrrou partage (F_RDLCK), ce nouveau processus va pouvoir placer son verrou etant donnee que un verrou partage permet à plusieur processus de placer un verrou sur une meme section. Si c'est un verrou exclusif (F_WRLCK), ce nouveau processus va avoir un message d'erreur qui ne vas pas lui permettre de placer un verrou. Si c'est un verrou du type F_SETLKW, le programme va

> essayer de placer un verrou, et attend en cas de conflit. Dans le cas ou un signal est capture (i.e un conflit a eu lieu), alors l'appel se termine avec un retour de -1 et produit l'erreur EINTR (l'appel systeme et interrompu)