

TP - LIF12 système d'exploitation

Programmation d'un shell et préparation du projet

8 fev 2016

I Mini Shell

Le shell est un programme qui lit une commande sur l'entrée standard et l'exécute. L'une des difficultés est d'analyser la ligne de commande. Le code : `cli.skel.c` effectue une analyse basique, décompose la ligne de commande et appelle la fonction `executepv`. Cette fonction ne fait qu'afficher la tâche à effectuer. Vous devez remplacer cette appel par celui de la commande système `execvp`.

Q.I.1) - Faire un shell qui permet d'exécuter une commande simple en premier plan par exemple :

— `ls -l /tmp/`
— `emacs`

Q.I.2) - Assurez-vous que :

2(a) - Le shell continue d'exister après la fin de la tâche (il faut faire un `fork()` en plus du `execvp`).

2(b) - Si la commande lancée est erronée, le shell le signale et retourne dans un état convenable.

Q.I.3) - Faire un shell qui permet d'exécuter des commandes en tâche de fond ou au premier plan.

— `emacs &`

Q.I.4) - Assurez-vous que :

4(a) - Lorsqu'une tâche quelconque se termine, le shell signale la chose immédiatement. Ce signalement doit se faire même s'il y a déjà une tâche en premier plan.

4(b) - La fin d'une tâche ne crée pas de processus *zombie*.

4(c) - La fin d'une tâche lancée en tâche de fond ne débloquent pas le shell

Q.I.5) - Faire un shell qui permet d'exécuter des commandes avec un tube par exemple :

— `ls -l | less`
— `ls | xargs emacs &`

Q.I.6) - (*question subsidiaire*) ajouter la possibilité d'exécuter des commandes avec plusieurs tubes, ou des redirections vers des fichiers.