

Mini-Projet “Systèmes d’exploitation”

P. Laroque



mars 2016

Sujet : interpréteur de commandes

Présentation du sujet

Il s’agit d’écrire un interpréteur de commandes **minish**, sur la base d’un sous-ensemble de ce qui est disponible sur les shells habituels (**bash** notamment).

A la base, l’algorithme est donc relativement simple :

```
while not EOF(stdin)
do
    read cmd
    fork
    child: exec(cmd)
    parent: wait(child)
```

Fonctionnalités attendues

Les fonctionnalités à implémenter obligatoirement sont réduites, compte tenu du temps disponible : votre interpréteur doit être capable d’effectuer la lecture, l’analyse et l’exécution d’une commande quelconque, avec nombre de paramètres quelconque.

En fonction de l’avancement de votre travail, vous pouvez ajouter à ce socle une ou plusieurs de fonctionnalités suivantes :

- Exécution au premier plan ou en tâche de fond (symbole “&”).
- Redirection de l’entrée (symbole “<”), de la sortie (symboles “>” et “>>”) d’une commande dans un fichier.
- Gestion des tubes pour enchaîner des commandes.
- Gestion de variables shell (notamment l’accès à la valeur de retour d’une commande, utile pour les scripts).
- Gestion de caractères spéciaux (symboles “*”, “?”, ...).
- Gestion de l’arithmétique entière.
- Gestion de structures de contrôle pour écrire des scripts
- Gestion de l’historique des commandes.
- ...

Travail à fournir

Le travail à rendre est à faire seul ou par groupes d’au plus deux étudiants.

Le rendu final sera présenté sous la forme d’une archive au format **tar** (compressé ou non) ou **zip**, à l’exclusion de toute autre format. Le nom de l’archive reprendra celui du ou des deux membres du groupe.

Cette archive contiendra :

- les fichiers sources de votre programme ;
- un **makefile** pour automatiser la production de l’exécutable et l’installation de votre logiciel sur une plate-forme UNIX.