

T.D. 4 - Le Shell (deuxième partie)

1 Processus

La commande `sleep` sert à attendre pendant un nombre de secondes spécifié. Par exemple, `sleep 5` attend 5 secondes. Cette commande va servir de base pour ces manipulations car c'est une commande qui permet de "simuler" l'exécution d'une longue tâche telle qu'une grosse compilation par exemple.

Lancez la commande `sleep 5`. Que se passe-t-il ?

Lancez la commande `sleep 500` en arrière-plan. Vérifiez avec `jobs` que votre commande est toujours là.

Lancez la commande `sleep 5` en arrière-plan. Que se passe-t-il lorsqu'elle se termine ?

Votre commande `sleep 500` est toujours active. Mettez-la en avant-plan. Suspendez-la. Faites `jobs`. Quel est son état ? Relancez-la en arrière-plan.

Lancez une deuxième commande `sleep 100` en arrière-plan. Passez la première en avant-plan. Suspendez-la. Suspendez la deuxième. Reprenez l'exécution de la première en avant-plan. Repassez la première en arrière-plan et reprenez la deuxième en arrière-plan.

Faites `ps` pour contrôler les processus actifs. Quelles sont les différences avec `jobs` ? Comment faire pour obtenir la liste de *tous* vos processus ?

2 La commande sed

La commande `sed` (*stream editor*) permet de faire des manipulations sur des fichiers. C'est un filtre qui prend un fichier en entrée, lui applique des commandes comparables à celle du mode "global" de `vi`, et renvoie le fichier modifié en sortie.

La syntaxe de `sed` est : `sed -e commande -e commande ...`

Par exemple, `sed -e 's/:.*//' < /etc/passwd` affiche les noms de login des utilisateurs référencés sur le système.

À l'aide de la commande `sed`, faites une commande pour afficher les *noms en clair* des utilisateurs référencés sur le système.

3 Scripts

Écrivez une commande `finger` dont le paramètre est un nom de login, et qui affiche la ligne correspondante dans `/etc/passwd`.

Modifiez votre commande `finger` pour qu'elle n'affiche plus que le nom en clair de l'utilisateur.

4 La commande who

La commande `who` affiche la liste des utilisateurs connectés à cet instant. Plus précisément, dans le cadre de terminaux X-Window, `who` affiche la liste des utilisateurs de fenêtres *shell*.

À l'aide de la commande `sort`, affichez la liste des utilisateurs triée par ordre alphabétique.

Vous devez constater que certains utilisateurs ont plusieurs fenêtres *shell* et apparaissent donc plusieurs fois dans la liste. On désire maintenant avoir le nombre d'utilisateurs *réellement* présents sur le système (c'est à dire sans compter les répétitions). À l'aide des commandes `sed`, `sort` et `uniq`, obtenez une liste d'utilisateurs sans doublons.

Faites maintenant un script pour compter le nombre d'utilisateurs réellement présents sur le système.

5 Une jolie arborescence

À l'aide des commandes `find` et `sed`, obtenez une liste *indentée* des répertoires (et seulement des répertoires) de votre arborescence. Par exemple, si votre arborescence contient les répertoires `a`, `b`, `c`, `a/d`, `a/e`, `a/d/f` et `b/g`, votre commande devra afficher :

```
a
  d
    f
  e
b
  g
c
```