

L1

TDo5

Systeme

Amine NAJAH [\[amine.najahi@univ-perp.fr\]](mailto:amine.najahi@univ-perp.fr)

Les exercices de ce TDo reprennent les notions vues au cours des TDo précédents. Comme toujours, vous pouvez vous aider des TDo précédents, des pages de manuels, ou de toute autre documentation pour résoudre les questions qui vont suivre...

EXERCICE 1 Embrouillez moi !

1. Créez un dossier `Temp`. Descendez dans `Temp`. Créez un fichier `dossiers` et un dossier `fichiers`.
2. Écrivez « Liste des dossiers de / » dans le fichier `dossiers`. Puis, ajoutez-y la liste des noms des dossiers (et uniquement les dossiers du premier niveau) que contient / en utilisant `[find]` et son option de formatage vue au TDo4.

La sortie ne pourra alors pas comporter les entrées suivantes :

```
/
/bin/usr
/root
initrd.img.old
/home/moi/Documents/fichier3
/tmp
```

mais devra ressembler à :

```
bin
root
home
tmp
```

etc...

3. Trouvez l'utilité de la commande `[tail]`. Affichez ensuite les 3 dernières lignes du fichier `dossiers`, puis les deux dernières, puis la dernière.
4. Trouvez, à l'aide des commandes `[tail]` et `[head]` comment afficher la ligne i d'un fichier.
 - En utilisant un fichier temporaire (redirection).
 - En utilisant des tubes.
5. À l'aide des commandes précédentes, créez dans le dossier `fichiers`, un fichier pour chaque entrée que contient le fichier `dossiers`. Le nom du fichier correspondra à son entrée respective dans le fichier.
6. Pour chaque fichier ainsi créé, effectuez le traitement de la question 2. pour les dossiers correspondants.
7. À l'aide de ces fichiers, déterminez quel dossier de / contient le plus de dossier.

8. Auriez vous pu répondre à la question précédente à l'aide d'une simple commande ?

EXERCICE 2 En 30 secondes

1. Copiez `Temp` dans `Temp2`.
2. Supprimez `Temp`.
3. Renommez `Temp2` en `Temp`.

EXERCICE 3 Processus

1. Éditez un fichier `fic` avec le contenu suivant :

```
gedit &
echo bonjour
cat > tmp
cat tmp
sleep 1000000 &
echo 10*2 | bc
sleep 100
echo 10^10^8 | bc &
nohup sleep 10000000 &
emacs ; emacs & ; emacs
find . | cat -n | cat -n | cat -n
```

puis rendez le exécutable.

Dans un premier temps, à l'aide des pages de manuels et sans lancer les commandes, imaginez l'effet qu'aura la commande suivante `[./fic]`.

2. Quels processus sont créés ?
3. Dans quel ordre ?
4. À quels moments l'intervention humaine est requise pour faire avancer l'interprétation des commandes contenues dans `fic`.
5. Que devra t-on alors faire ?
6. Combien de processus seront créé ?
7. Combien seront encore en activité à la fin de l'interprétation du fichier `fic` ?
8. Combien seront encore en activité même après la fermeture du terminal ?

Ouvrez maintenant deux terminaux côte à côte. Exécutez dans le premier le fichier `fic`. Le second terminal vous servira à consulter les pages de `man` et à contrôler l'activité des processus avec `[ps]`.

9. Sans intervenir dans le premier terminal, faite en sorte qu'il interprète toutes les commandes du fichier `fic`. Vous devrez pour ce faire tuer les processus « bloquant ».
10. Lorsque vous pourrez récupérer la main dans le premier terminal. Comparez la liste des processus toujours en activité et vérifiez si cela concorde avec votre intuition.
11. Fermez le premier terminal, puis vérifiez la liste des processus toujours en activité. Cela concorde t-il avec votre intuition ?
12. Enfin, tuez les processus lancés par `fic` toujours en activité.