

TP

Commandes Linux et les scripts shell

Exercice 1 : commande « touch »

Ce paragraphe est donné pour le cas où la commande « touch » n'aurait pas été vue en cours.

La commande « touch exemple.txt » crée le fichier « exemple.txt » avec une taille de zéro octet.

Exercice 2 : commandes diverses

Placez vous dans le répertoire « /tmp ».

On va construire une arborescence qui simule l'état du monde en 1900 :

- Créez un dossier « monde ».
- Créez dedans un dossier « 1900 ».
- Créez dedans les dossiers :
 - « allemagne »
 - « belgique »
 - « france »
- Dans le dossier « allemagne », créez :
 - un fichier « berlin.txt »
 - un dossier « alsace »,
 - un dossier « burundi »,
 - un dossier « tanzanie ».
- Dans le dossier « belgique », créez :
 - un fichier « bruxelles.txt »
 - un dossier « congo-belge »
 - créer dedans un fichier « leopoldville.txt »
- Dans le dossier « france », créez :
 - un fichier « paris.txt »
 - un dossier « algerie » ;
 - créer dedans un fichier « alger.txt »

Mettez à jour l'arborescence pour simuler la situation actuelle :

- le Congo Belge est autonome et s'appelle maintenant le Congo et sa capitale s'appelle maintenant « Kinshasa »
- l'Algérie est un pays indépendant
- l'Alsace est redevenue française

- la Tanzanie est un pays indépendant
- le Burundi est un pays indépendant

Exercice 3 : commandes diverses

Placez vous dans le répertoire « /tmp ».

Créez le dossier « vacances ». Dedans créez le dossier « paris ». Créez dedans un dossier par arrondissement.

Téléchargez ou copiez des fichiers ou des photos sous forme d'un fichier compressé en zip par exemple. Il représenterait des photos, par exemple, des photos de vos vacances à Paris.

Désarchivez l'archive ZIP et rangez les photos dans les bons arrondissements (utilisez la commande « unzip »). Vous avez photographié :

- Notre Dame dans le 1er arrondissement
- l'université de Jussieu dans le 5è arrondissement
- le Panthéon dans le 5è arrondissement
- les Invalides dans le 7è arrondissement
- la Tour Eiffel dans le 7è arrondissement
- Crazy Horse dans le 8è arrondissement
- Montmartre dans le 18è arrondissement
- l'Arc de Triomphe dans le 16è arrondissement

Exercice 4 : Création d'une arborescence

Pour l'occasion de cet exercice, on change temporairement l'emplacement de votre répertoire d'accueil. Pour cela lancez la commande :

```
bash
```

puis lancez les commandes suivantes :

```
mkdir -p 02-B4
HOME=$HOME/02-B4
export HOME
cd $HOME
```

Sans lancer les commandes suivantes, dessinez sur un papier l'arborescence construite au fur et à mesure et trouvez l'arborescence finale résultat.

La liste des commandes est :

```
cp /etc/hosts a
mkdir b c
cd b
cp ../a d
mkdir ../e f
cd
cp a b/f/g
cd b/f
```

```
cp g ../../e
cd ..
rm ../a
rmdir ../c
mv ../e/g ../e/x
```

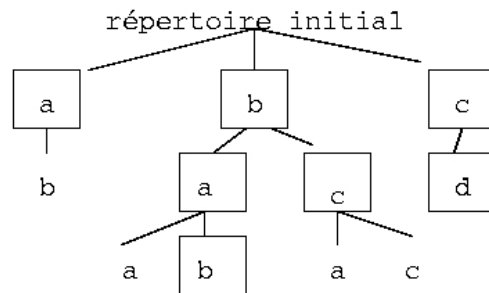
Ne lancez pas ces commandes, dessinez au fur et à mesure les résultats...
 Quel est le répertoire courant à la fin de l'opération ?
 Vérifier que votre résultat est bon en tapant ces commandes.
 Quand vous aurez terminé cet exercice faites alors :

```
exit
```

Cette commande rétablira le répertoire d'accueil normal.

Exercice 5 : Création d'une arborescence

Quelle séquence de commandes vous permet de créer l'arborescence de la figure ci-dessous :



Essayez de minimaliser le nombre de commandes.

Exercice 6 : pratique des raccourcis

Lancez les commandes UNIX pour réaliser les questions suivantes :

1. Entrez quelques commandes de façon à pouvoir faire les questions qui suivent. Pour cela tapez des commandes comme les suivantes (peu importe les commandes, il faut juste en avoir au moins 4 ou 5 en stock) :

```
date
ls -l
ls -al
cal
cat /etc/issue
cat /etc/motd
ls -l /etc/passwd
date
```

2. Au moyen des flèches de curseur, remontez dans les commandes déjà entrées et voyez les effets des flèches dans diverses directions.
3. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-A
4. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-E
5. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-U
6. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-Y
7. Tapez « `more /etc/termc` » puis appuyez sur la touche TAB et constatez ce qui se passe.
8. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-P
9. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-N
10. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-K
11. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-Y
12. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-B et de la combinaison de touches Ctrl-F
13. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Esc-B et de la combinaison de touches Esc-F
14. Voyez l'effet sur votre ligne de commande shell de la combinaison de touches Ctrl-L
15. Voyez l'effet sur votre console shell de la commande « `clear` »

Exercice 7 : Programmation en shell

Le shell n'est pas qu'un simple interpréteur de commande, c'est aussi un langage de programmation. Il est possible d'écrire une série d'instruction shell dans un fichier (un script) et d'exécuter ce script. Chaque shell possède sa propre syntaxe en ce qui concerne les affectations, instructions de boucles, etc.

```
#!/bin/bash
# mon premier script
clear
while :
do
echo 'Menu'
echo '[1] Affichage repertoire courant'
echo '[2] Liste des fichiers du repertoire'
echo '[3] Information sur un fichier'
echo '[4] Changement de repertoire'
echo '[5] n premieres lignes d'un fichier'
echo '[0] Fin'
echo 'Choix:'
read ch
case $ch in
0) exit 0
;;
1) pwd
;;
2) ls
;;
3) echo -n 'Nom du fichier:' ; read file ; ls -l $file
;;
4) echo -n 'Nouveau repertoire:' ; read rep ; cd $rep
;;
5) echo -n 'Nom du fichier:' ; read file
echo -n 'Nb de lignes a afficher:' ; read n
head -n $n $file
;;
*) echo 'choix non propose'
esac
done
```

A l'aide de l'éditeur de texte de votre choix, saisir ce programme en le sauvant sous le nom script1.

Pour l'exécuter (en mode "débogage") taper : `bash -x script1`

Profitez-en pour comprendre ce que fait chaque instruction et sa syntaxe (chaque caractère a son importance).