



TP LINUX n°1 : Principales commandes

version du 19 septembre 2012 - Tous droits réservés - Christophe VARDON

Nom : _____ Prénom : _____ Classe : _____ Date : _____	Appréciation :	Note :
Objectif : Être capable d'utiliser diverses commandes sur un terminal UNIX (Linux)		durée : 2h
Matériel : 1 PC avec système Ubuntu		
Travail à réaliser : Répondez aux questions posées en réalisant le TP suivant		

LE MODE CONSOLE OU MODE TERMINAL

→ Expliquez ce qu'est le mode console ? 

→ Pourquoi est-il parfois intéressant de travailler dans ce mode ?

Pourquoi utiliser le mode console (= mode terminal virtuel) ?

Le mode console ressemble visuellement au mode ms-dos de Microsoft (mode texte); Il s'agit toutefois d'un Linux complet en mode multi-taches et multi-utilisateur;

Ce mode est utile dans 2 cas précis :

- on désire disposer d'un maximum de puissance (le système graphique est consommateur en ressources processeur et mémoire)
- Le système graphique n'est pas disponible pour diverses raisons (panne, terminal en mode texte..)
- contrairement à ce que l'on pourrait croire, le mode console est plus simple que le mode graphique pour certaines opérations (ex : montage de lecteur, transfert d'un fichier, ajout d'un utilisateur,...)

→ Que peut-on faire en mode console ?

Que peut-on faire en mode console ?

- A peu près tout (logiciels en ligne de commande) ce qu'on peut faire en mode graphique, à l'exception évidemment des applications graphiques (dessin, photos, vidéos, ...)
- Les commandes de base d' Unix sont aussi diverses que puissantes... (voir ci-dessous)
- Les scripts combinent plusieurs de ces commandes pour réaliser des actions complexes.

Les descriptions des commandes présentées ci-dessous sont tirées des pages man traduites en français

PRINCIPALES COMMANDES UNIX

man [options] [section] [commande]

Affiche la page du manuel correspondant à la commande spécifiée. Suivant les commandes, vous devez également spécifier la section à laquelle elle appartient.

- tapez « man ls » pour obtenir une aide sur la commande ls
- comment se présente cette page d'aide? est-elle en Français ?

ls [options] [noms]

Affiche le contenu d'un répertoire (répertoire courant si non spécifié). Si un nom de fichier est spécifié, affiche la liste des fichiers correspondants.

- tapez dans une console : « ls »
- quels sont les éléments listés et comment sont-ils présentés?
- tapez « man ls » pour obtenir une aide sur la commande ls
- Expliquez la fonction des options " -a " et " -l "; testez-les.

cd [répertoire]

Change le répertoire de travail.

- tapez : cd /usr/share/doc
- Recopiez le **prompt** affiché? Que constatez-vous?
- tapez ls
- citez 2 ou 3 répertoires contenus dans /usr/share/doc
- déplacez-vous dans un de ces répertoires
- quelle commande avez-vous utilisé?
- tapez « cd .. », observez ce qui se passe
- Conclusion : quelle commande faut-il taper pour reculer d'une branche dans l'arborescence?
- Tapez « cd / »
- A quel endroit de l'arborescence vous trouvez-vous maintenant ?
- La commande « cd ~ » permet d'aller dans son répertoire personnel, appelé en anglais "*home*"
- Testez cette commande. Dans quel répertoire êtes-vous maintenant?

pwd

Affiche répertoire courant.

- tapez : « pwd »
- Que constatez-vous?

mkdir [répertoire]

Crée le nouveau répertoire.

- tapez : « mkdir achille »; puis « ls »
- Que constatez-vous?

cat [options] [fichiers]

Cette commande permet d'afficher un ou plusieurs fichiers sur la sortie standard (effectue la lecture à partir de l'entrée standard si aucun fichier n'est spécifié). Vous pouvez utiliser les opérateurs > pour combiner plusieurs fichiers en un seul, et >> pour concaténer des fichiers à un fichier existant.

Options les plus courantes :

- n : Numérote les lignes, en commençant par 1.
- s : Regroupe en une seule ligne plusieurs lignes blanches consécutives.

Placez-vous dans le répertoire /etc

- Tapez la commande « ls -d s* »
- Que réalise cette commande? Quel est le point commun entre tous les fichiers listés ?

- Tapez « cat services », puis faites (ctrl-c) pour arrêter;
- Qu'avez-vous constaté ?

- Tapez « cat services | more »
- Que constatez-vous ? Que fait la commande « more »

- Tapez « ls pas* »
- Quels sont les fichiers de "/etc" dont le nom commence par "pas" ?

- Tapez « cat passwd »
- Que contient ce fichier ? (recopier les 2 premières lignes)

- Tapez « cat -n passwd »
- Que constatez-vous ? A quoi sert l'option -n ?

cp [options] source destination

Copie un fichier source vers un autre fichier (qui sera écrasé si ce fichier existe). Copie également un ensemble de fichiers vers un répertoire existant.

- Placez-vous sous le répertoire /etc/X11 (faire **pwd** pour vérifier)
- faites un "**ls**"; notez les fichiers présents

- Tapez « cp xorg.conf ~ »
- Que constatez-vous ?

- Déplacez-vous dans votre répertoire personnel (cd ~)
- Que constatez-vous ?

- Faire « man cp »
- Quelle commande pour copier un répertoire et son contenu ? Tester cette commande.

mv [options] source destination

Renomme le fichier *source* en *destination* ou déplace un fichier *source* vers le répertoire *destination* . Déplace également un ensemble de fichiers vers un répertoire existant.

- Tapez «**touch** bidon.test», puis « ls »;
- Que constatez-vous ? Quelle est le résultat de cette commande?

- Tapez «mv bidon.test bidon.test.new», puis « ls »;
- Que constatez-vous ? Quelle est le résultat de cette commande?

- Tapez «mv bidon.test.new /tmp», puis « cd /tmp »; puis « ls »;
- Que constatez-vous ? Quelle est le résultat de cette commande?

rm [options] nom

Supprime le fichier ou le répertoire cible.

- Tapez «rm bidon.test.new», puis « ls »;
- Que constatez-vous ? Quelle est le résultat de cette commande?

- Tapez «man rm »
- Quelle option pour supprimer un répertoire et son contenu, éviter le message de confirmation?

df [options] [nom]

Affiche l'espace disque disponible sur tous les systèmes de fichiers montés ou sur celui dont le nom est spécifié.

- Tapez « **df** »
- Que constatez-vous ? Quelles informations cette commande donne-t-elle? (5)

free [options]

Affiche un certain nombre de renseignements sur la mémoire (mémoire utilisée, disponible, ...).

- Tapez « **free** »
- Que constatez-vous ? Quelles informations cette commande donne-t-elle? (les 3 1^{ères} lignes)

ps [options]

Affiche un certain nombre de renseignements sur les processus en cours

- Tapez «ps aux»
- Que constatez-vous ? Quelles informations cette commande donne-t-elle? (les 3 1^{ères} lignes)

top / htop

Affiche les processus en cours en temps réel

- Tapez « top »
- Quelles informations cette commande donne-t-elle?

date [options] [format] [date]

Permet de configurer l'heure et la date du système. Sans argument de date, permet de visualiser l'heure et la date du système.

- Tapez « date »
- Que constatez-vous ? Quelles informations cette commande donne-t-elle?

grep [options] expression [fichiers]

Effectue dans un ou plusieurs fichiers spécifiés la recherche d'une expression. Par défaut, la commande renvoie les lignes contenant l'expression.

- Tapez « cat /etc/passwd »
- Que constatez-vous ? Quelles informations cette commande donne-t-elle?

- Tapez « cat /etc/passwd | grep root »
- Que constatez-vous ? Comparez avec le résultat de la commande précédente.

wget [options] [fichiers]

Télécharge un fichier sur le réseau local ou internet.

- Délacez-vous dans votre "home" , puis tapez « wget http://www.cvardon.fr/test.tar.gz »
- Que constatez-vous ?

- Tapez « ls »
- Que contient maintenant ce répertoire ?

tar [options] [fichiers]

Comprime ou décomprime une archive au format "tar". l'option "xf" indique qu'il faut extraire les fichiers de l'archive, et l'option "cf" indique qu'il faut créer une nouvelle archive. On rajoute l'option "z" pour utiliser la compression "zip".

- Tapez « tar xzf test.tar.gz », puis « ls »
- Que constatez-vous ? Que contient maintenant ce répertoire ?

halt / reboot

La commande **halt** permet d'arrêter l'ordinateur, et la commande **reboot** permet de le redémarrer

- Tapez « reboot »
- Que constatez-vous ?

- Tapez « halt »
- Que constatez-vous ?

```
#!/bin/bash
```

Cette instruction doit être écrite sur la première ligne d'un fichier pour indiquer que ce fichier est un script bash. Un script est un ensemble de lignes de commande qui seront exécutées dans l'ordre quand on le lancera

- Déplacez-vous dans le répertoire : `/root` ; puis faites : `ls`
→ Quels fichiers voyez-vous ?

- Créer un fichier nommé « test.sh » avec l'éditeur de votre choix ; écrivez les lignes :

```
#!/bin/bash  
echo "Bonjour"
```

*Après avoir enregistré le fichier, faire un **ls** pour vérifier que le fichier est bien là*

- Autoriser l'exécution du fichier avec la commande : `chmod +x test.sh`
- Tapez la commande : `./test.sh` (ne pas oublier le point avant le slash !!)
→ Que constatez-vous ?

→ A votre avis, quel est le rôle de la commande `echo` ?

```
$
```

Cette lettre désigne la valeur d'un paramètre (aussi appelé variable en mathématique) ; par si je fixe la valeur `n=10` , `$n` renvoie le nombre 10

- Créer un fichier nommé « test2.sh » avec l'éditeur de votre choix ; écrivez les lignes :

```
#!/bin/bash  
a=Toto  
b=Bonjour  
c=ans  
d=3  
echo $b $a $d $c
```

*Après avoir enregistré le fichier, faire un **ls** pour vérifier que le fichier est bien là ; puis ne pas oublier de faire le :
chmod +x test2.sh*

- Tapez la commande : `./test2.sh` (ne pas oublier le point avant le slash !!)
→ Que constatez-vous ?

- Modifier les valeurs de a, b, c, ou d dans le fichier « test2.sh » est testez le résultat

For ... do ... done

Cette instruction sert à créer une boucle ; une boucle est une instruction qui va s'exécuter en boucle jusqu'à ce que la condition de sortie soit remplie ;

- Créer un fichier nommé « test3.sh » avec l'éditeur de votre choix ; écrivez les lignes :

```
#!/bin/bash
for (( i=1; i<=5; i++ ))
do
    echo "Bonjour $i fois"
done
```

- Tapez la commande : `./test2.sh` (ne pas oublier le point avant le slash !!)
- Que constatez-vous ?

→ Dans la ligne `echo "Bonjour $i fois"` , que désigne le `$i` ?

(Dans les trois questions suivantes, si vous avez un doute, modifiez la valeur et testez le résultat)

→ Dans la ligne `for ((i=1; i<=5; i++))`, que signifie le `i=1` ; ?

→ Dans la ligne `for ((i=1; i<=5; i++))`, que signifie le `i<=5` ; ?

→ Dans la ligne `for ((i=1; i<=5; i++))`, que signifie le `i++` ?