

Stage Unix IMAC 2008-2009



Fabrice Boutarel
fabrice.boutarel@gmail.com

6 octobre 2008

1 Tomber du nid - Premiers contacts

Vous évoluez maintenant dans le système GNU/Linux. Cet environnement sera votre poste de travail durant les trois prochaines années. Il est impératif que tous les étudiants s'identifient afin de vérifier les mots de passe, et d'informer le CRI en cas de problème. Une fois les mots de passe vérifiés, vous avez quartier libre pour vous familiariser avec l'environnement de travail.

1.1 A faire

- Ouvrez un navigateur internet, trouvez comment changer le fond d'écran.
- Définissez les termes de système d'exploitation, de gestionnaire de fenêtre, d'environnement du bureau.

2 Souricide - La ligne de commande

Fini de jouer on ouvre un terminal et on pose la souris. Loin. Les systèmes Unix possèdent une interface en ligne de commande très puissante. La ligne de commande est une manière rapide, efficace et facilement automatisée d'utiliser le système. Votre environnement de travail s'appuie très largement sur cette interface, il est donc nécessaire de la maîtriser. De plus elle tolère rarement les erreurs et exige de la rigueur, c'est une très bonne initiation à la philosophie de la programmation. Une ligne de commande se présentera toujours sous la forme

```
% commande <options> <arguments>
```

Attention Unix est traditionnellement sensible à la casse.

2.1 A faire

- Ouvrez un terminal.
- Taper votre première commande *xeyes*. Utilisez le raccourci Ctrl+C pour l'arrêter.
- Expérimitez les autres raccourcis utiles. Ctrl+W Ctrl+A Ctrl+E Ctrl+U Ctrl+K Ctrl+L Flèches.

3 RTFM!! - Le manuel

La commande *man* permet d'accéder à la documentation du système, aux programmes, aux fonctions, tout y est.

```
% man <commande>
```

3.1 A faire

- Utilisez le manuel et la commande *date* pour produire la sortie suivante : 24/09/07 23h40

4 Promenons nous dans les bois - Le système de fichier

Attention la plupart de ces commandes modifient vos données et ne disposent pas de fonction "annuler".

4.1 Explorer la forêt

Unix dispose les fichiers en arborescence de répertoires. Découvrez grâce au manuel l'utilité des commandes *ls*, *cd*, *pwd*, *pushd* et *popd*,

4.1.1 A faire

- Utilisez les nouvelles commandes pour dessiner une arborescence du système Unix.
- Donner la fonction des répertoires : / /bin /sbin /dev /boot /usr /media /proc /sys /tmp /root /home . et ..

4.2 Planter des arbres

Vous allez maintenant créer votre propre arborescence vous aurez besoin des commandes *mv*, *cp*, *ln*, *mkdir*, *touch*, *rm*, *rmdir*

4.2.1 A faire

- Creez un répertoire *stageunix* à la racine de votre *home*.
- Creez deux sous-répertoires *dir1* et *dir2*.
- Creez deux fichiers *file1* et *file2* dans *dir1*.
- Copiez *dir1/file1* dans *dir2*.
- Copiez *dir2* en *dir3*.
- Renommez *dir3* en *dirX*.
- Effacez *dirX*.
- Effacez *dir2* en une seule commande.
- Creer un lien */hardlinkfile1* à partir de *dir1/file1*.
- Creer un lien symbolique */symlinkdir1* à partir de *dir1*.
- Effacez les liens.

4.3 Installer les clôtures - le système des droits

Unix est un système fondamentalement multi-utilisateurs. Pour chaque fichier il existe un propriétaire et un groupe. Seul le propriétaire du fichier peut modifier sa parténité. On distingue trois types d'accès à un fichier, l'accès en écriture, en lecture et en exécution. Ces accès peuvent être autorisés ou interdits au possesseur, au groupe et aux autres utilisateurs distinctivement. Les accès peuvent s'écrire de deux façons, en octal ou en littéral :

```
777 rwxrwxrwx
755 rwxr-xr-x
744 rwxr--r--
```

Les commandes nécessaires à la gestion des droits sont *umask*, *ls -l*, *chmod*, *chgrp* et *chown*.

4.3.1 A faire

- Créer un répertoire *rep3* et modifier ses droits pour empêcher tout le monde sauf sois-même d'y écrire.
- Afficher les droits du fichier */etc/services*. Pourquoi ne peut-on pas le supprimer ?
- Donner les commandes (sous forme octale et explicite) permettant de changer les droits selon les schémas suivant :
=> r-xr-rwx
=> -wx-x-r-
=> r---rwx
 - Que signifie le droit d'exécution pour un répertoire ?
 - Lister les droits du fichier */bin/mount*. Quel est la différence ?
 - Entrez la commande *umask*, de manière à ce que les fichiers lors de leur création aient par défaut les droits 446.

4.4 La cueillette - chercher des fichiers

On utilise la commande *find* pour rechercher dans une arborescence. Sa syntaxe est :

```
find <arborescence> -name <masque>
```

4.4.1 A faire

- Se placer dans le répertoire `/usr` et y rechercher le fichier `gcc`.
- Comment fait-on pour indiquer que le fichier recherché a été modifié il y a plus de 30 jours ? Il y a 30 jours ? Il y a moins de 30 jours ?
- Comment faire pour dire que le fichier a été modifié plus récemment qu'un autre fichier donné ?
- Comment spécifier que le fichier recherché a une taille supérieure à une taille donné ?

5 Survivre dans la jungle - Un peu de système

5.1 Le Quota

Votre compte vous autorise à 500 Mo , c'est très peu et vous allez forcement le dépasser. En cas de dépassement important et prolongé vous ne pourrez plus vous identifier sur votre session. Voici la marche à suivre pour récupérer votre session. Appuyez sur `Ctrl+Alt+F1`. Vous venez de passer en mode console, identifiez vous. Vous allez maintenant pouvoir réduire la taille de votre compte grâce aux commandes `du`. Déloggez vous (`exit`) et revenez en mode graphique en appuyant sur `Ctrl+Alt+F7`.

5.1.1 A faire

- Testez la procédure et afficher la taille occupée par votre compte.

5.2 Les Archives

Les commandes `tar`, `gzip`, `zip` et `unzip` permettent de créer et décompresser des archives.

5.2.1 A faire

- Creer une archive du répertoire `stageunix` au format `tgz`.

5.3 Les Processus

Les commandes `ps` et `kill` permettent respectivement d'afficher et d'envoyer des signaux aux processus en cours.

5.3.1 A faire

- Relancer `xeyes`, visualiser le processus et lui envoyer le signal de fin.

6 Apprendre à nager - Traiter les flux

Les commandes `cat less echo grep cut wc sort sed unic tail head` sont des commandes de traitement de flux de texte. Elles prennent en entrée un flux, ou un fichier et effectuent un traitement (mise en forme, modification, recherche...) avant de rediriger le flux.

6.1 A faire

- Trouver les commandes pour :
- Afficher les 5 premières lignes de `/etc/group`
- Afficher les 10 dernières lignes de `/etc/passwd`
- Afficher 'Hello, World !'
- Afficher le contenu de `debian_version`
- Afficher uniquement la ligne de l'utilisateur root dans `/etc/group`

7 Un fantôme dans le coquillage - Le Shell

Le Shell est l'interpréteur de commande que vous utilisez actuellement, cet interpréteur est capable de faire bien plus que d'exécuter une commande après une autre. Dans cette section nous verrons les outils qui vont permettre de répondre aux exercices de la section suivante.

7.1 Les variables

Le shell est capable de mémoriser des valeurs et de les affecter à une variable.

```
% export foo='bar'  
% echo $foo  
bar
```

Il utilise lui-même couramment de nombreuses variables d'environnement qui décrivent la configuration du système. Certaines de ces variables peuvent être consultées grâce à la commande *printenv*.

7.1.1 A faire

- Donnez la signification des variables \$SHELL \$HOME et \$PATH

7.2 Les alias

Les commandes peuvent parfois être longues à taper notamment en cas de longues options. Il est possible de définir des alias pour les commandes les plus courantes grâce à la commande *alias*.

7.2.1 A faire

- Faites en sorte que la commande ll affiche tous les fichiers d'un répertoire avec un fichier par ligne.

7.3 Les trois flux standards

Tous les programmes Unix disposent de trois flux de communication standards. L'entrée standard *stdin* est par défaut reliée au clavier. La sortie *stdout* est par défaut reliée à l'écran ainsi que *stderr* la sortie d'erreur. Les programmes Unix se branchent sur les flux standards pour afficher et recevoir des données.

7.4 Les redirections

Le shell permet de rediriger les flux standards vers des fichiers ou d'autres programmes. Par exemple, la commande *echo* affiche traditionnellement sur la sortie *stdout* à l'écran. Grâce à la commande

```
% echo ''bar''  
bar  
% echo ''bar'' > bar.txt
```

Nous venons de rediriger la sortie *stdout* en la connectant à un fichier. Nous venons donc d'écrire dans un fichier. Il existe différentes redirections vers les fichiers : > redirige vers un fichier en écrasant son contenu, >> rediriger vers un fichier en ajoutant à son contenu, 2 > et 2 >> fonctionne de la même façon pour la sortie d'erreur. L'entrée standard se redirige ainsi :

```
% cat < bar.txt  
bar
```

Encore plus intéressant, il est possible de rediriger la sortie standard d'un programme vers l'entrée d'un autre programme, grâce au caractère pipe.

7.4.1 A faire

- Testez et expliquez la commande :

```
% cat /etc/group | grep root
```

7.5 Les métacaractères

Le shell interprète et substitue un certain nombre de caractères avant d'exécuter une commande. La commande reçoit les caractères comme si vous les aviez tapé.

```
% echo *
```

Cette commande affiche tous les fichiers du répertoire car le caractère * sert à masquer n'importe quelle chaîne de caractère (sauf le . en première position).

```
% echo ???
```

Cette commande affiche tous les fichiers du répertoire ayant un nom de 3 lettres car le caractère ? sert à masquer n'importe quelle chaîne de caractère (sauf le . en première position). Les énumérations '[...]' sont des suites explicites de caractères autorisés.

Les double quotes “ “ et le \ permettent d'empêcher l'interprétation des métacaractères.

7.6 Multi-tâches

Il est possible de lancer plusieurs commandes à la suite séparées par des ; Les commandes s'exécutent alors les unes après les autres. Il est aussi possible de lancer les commandes en tâche de fond avec le caractère &. Les commandes *jobs*, *bg* et *fg* permettent de reprendre la main sur les processus en tâche de fond.

7.6.1 A faire

- Lancez deux instances de *xeyes* sans perdre la main sur le terminal.

8 Dans la fosse - Exercices

8.1 A faire en une seule commande

- Ecrire une commande pour compter les fichiers de /etc
- Ecrire deux commandes pour trouver toutes les occurrences de la chaîne “root” dans l’ensemble des fichiers de /etc
- Afficher les lignes 10 à 15 du fichier /etc/group
- Donner deux façons d’afficher uniquement l’user Id de l’utilisateur man

8.2 Expliquer les commandes suivantes

```
% sed -i s/Foo/Bar/g myfile.txt  
% ps aux | sed s/\ /:/g | cut -f1 -d":" | sort | uniq | wc -l  
% cat /etc/passwd | sed s/:/\\ /g | cut -f1,3 -d" "  
% find /usr/bin -user root -name "????" -exec du {} \\; | sort -n
```

Références

- [1] ESME Sudria, *Unix-initiation*, <http://linuxfr.org/redirect/53710.html>