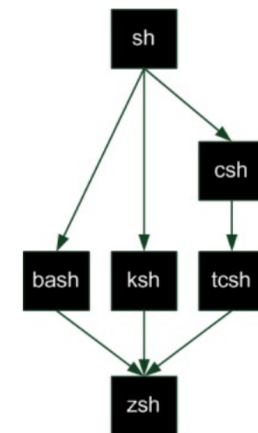


“ Prise en main du shell Bash sous Ubuntu ”

Une interface système ou coque logicielle (*shell* en anglais) est une couche logicielle qui fournit l'interface utilisateur d'un système d'exploitation. Il correspond à la couche la plus externe de ce dernier. Cette appellation est une métaphore pour désigner la couche la plus haute des interfaces des systèmes Unix, par opposition à la couche de bas niveau, appelée noyau (*kernel*).

Voici les noms de quelques-uns des principaux shells qui existent :

- sh : Bourne Shell. L'ancêtre de tous les shells.
- bash : Bourne Again Shell. Une amélioration du Bourne Shell, disponible par défaut sous Linux et Mac OS X.
- ksh : Korn Shell. Un shell puissant assez présent sur les Unix propriétaires, mais aussi disponible en version libre, compatible avec bash.
- csh : C Shell. Un shell utilisant une syntaxe proche du langage C.
- tcsh : Tenex C Shell. Amélioration du C Shell.
- zsh : Z Shell. Shell assez récent reprenant les meilleures idées de bash, ksh et tcsh



Sous Linux, le shell par défaut est le Bash.

JOB 1.

- Afficher le manuel de la commande `ls`

```
anais@ubuntanais:~$ man ls
```

La commande `man` correspond à “manuel”. Suivie du nom d’une commande, elle permet d’en afficher toutes les possibilités. Une fois la page ouverte, tapez `/occurrence_voulue` pour trouver le mot qui vous intéresse. Passez à l’occurrence suivante en tapant `n`

- Afficher les fichiers cachés du home de votre utilisateur

```
anais@ubuntanais:~$ ls -a
```

ou

```
anais@ubuntanais:~$ ls - -all
```

La commande `ls` permet d’afficher les fichiers contenus dans le répertoire actuel. En ajoutant `-a` ou `-all`, on affiche y compris les fichiers cachés.

- Afficher les fichiers cachés plus les informations sur les droits sous forme de liste

```
anais@ubuntanais:~$ ls -l
```

Une autre option de `ls` est `-l`. Elle affiche une liste détaillant chaque fichier du dossier, dont les droits que l’on a sur lui. Elle indique aussi :

- les droits sur le fichier,
- le nombre de liens physiques,
- le nom de la personne propriétaire du fichier,
- le groupe auquel appartient le fichier,

- la taille du fichier, en octets,
- la date de dernière modification,
- et le nom du fichier (ou dossier).

- Comment ajouter des options à une commande ? Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?

Pour ajouter une option à une commande, il faudra utiliser l'une des deux syntaxe admise par Linux. À la suite d'une commande, ajoutez un tiret (-) et la lettre correspondante à l'option souhaitée ou deux tirets (- -) suivis par l'option en toute lettre. Bien souvent, les deux fonctionnent pour la même option mais il existe toutefois des exceptions où il faudra impérativement saisir l'option en entier. On parle de version "courte" et de version "longue".

JOB 2.

- Lire un fichier en utilisant une commande qui permet seulement de lire

```
anais@ubuntanais:~$ cat .bashrc
```

`cat` permet d'afficher le contenu d'un fichier dans le terminal lui-même, mais pas de le modifier. Elle est adaptée si on travaille sur des petits fichiers.

```
anais@ubuntanais:~$ less .bashrc
```

`less` affiche également le contenu du fichier mais au lieu de renvoyer le contenu d'un seul coup, il affiche progressivement le contenu du fichier, page par page, ce qui laisse davantage de temps pour lire. `more` est l'ancêtre de `less`, il fait la même chose mais moins rapidement et avec moins de fonctionnalités.

- Afficher les 10 premières lignes du fichier ".bashrc"

```
anais@ubuntanais:~$ head .bashrc
```

La commande **head** affiche par défaut uniquement les 10 premières lignes du fichier.

- Afficher les 10 dernières lignes du fichier “.bashrc”

```
anais@ubuntanais:~$ tail .bashrc
```

La commande **tail** affiche par défaut uniquement les 10 dernières lignes du fichier.

- Afficher les 20 premières lignes du fichier “.bashrc”

```
anais@ubuntanais:~$ head -n 20 .bashrc
```

En ajoutant l’option **n** suivie d’un nombre, on indique le nombre de lignes que l’on souhaite voir apparaître. En l’occurrence 20 ici.

- Afficher les 20 dernières lignes du fichier “.bashrc”

```
anais@ubuntanais:~$ tail -n 20 .bashrc
```

La logique est la même que pour la commande **head**.

JOB 3.

- Installer le paquet “cmatrix”

```
anais@ubuntanais:~$ sudo apt-get install cmatrix
```

ou

```
anais@ubuntanais:~$ sudo apt install cmatrix
```

Il existe deux façons d'installer un logiciel via un gestionnaire de paquets. `apt-get` est sûrement la plus connue ; `aptitude` (ou `apt` tout court) est présentée comme étant plus efficace lors de la désinstallation, en effet, elle supprime aussi les dépendances inutiles. Cependant `apt-get` sous Ubuntu a évolué aussi et peut supprimer les dépendances non utilisées. Il s'agit donc essentiellement d'une question d'habitude.

- Lancer le paquet que vous venez d'installer

```
anais@ubuntanais:~$ cmatrix
```

Il suffit de taper le nom du logiciel pour le lancer. Pour en sortir, tapez `ctrl + c`

- Mettre à jour son gestionnaire de paquets (`apt-get` ou `apt`)

```
anais@ubuntanais:~$ sudo apt-get upgrade apt-get
ou anais@ubuntanais:~$ sudo apt upgrade apt-get
ou anais@ubuntanais:~$ sudo apt upgrade apt
ou anais@ubuntanais:~$ sudo apt-get upgrade apt
```

Contrairement à `upgrade`, `update` ne prend pas d'argument, donc elle ne peut pas être spécifiquement suivie du nom du paquet à mettre à jour. Pour mettre à jour un gestionnaire de paquets, il faudra utiliser la commande `upgrade`.

- Mettre à jour ses différents logiciels (en utilisant `apt-get` uniquement)

```
anais@ubuntanais:~$ sudo apt -get update
anais@ubuntanais:~$ sudo apt-get upgrade
```

Pour mettre à jour tous les logiciels, il faudra d'abord télécharger les informations sur l'état des paquets sur internet en utilisant `update`. Ce faisant, il met à jour la base de données des paquets. Ensuite seulement, on pourra effectivement les mettre à jour avec `upgrade`.

- Télécharger les logiciels : Google

Pour utiliser la ligne de commande pour installer et désinstaller un logiciel, il faut d'abord connaître le nom du package. Il faudra le chercher sur internet. Puis le saisir à la suite de la commande suivante :

```
anais@ubuntu:~$ sudo apt-get install google-chrome
```

La plupart des programmes sont référencés dans des dépôts et accessibles via `apt-get`. Toutefois, au même titre que google chrome, certains programmes ne sont pas disponibles via `apt-get`. Il est parfois possible de trouver sur le site web du logiciel un paquetage `.deb`. Si la version correspond à la machine et que les dépendances nécessaires sont bien installées, il faudra le décompresser (`dpkg`), pour procéder à l'installation (`-i`) :

```
anais@ubuntu:~$ cd Downloads
```

```
anais@ubuntu:~$ sudo dpkg -i nom_du_paquet
```

Enfin, si le programme n'est pas référencé et qu'il n'est pas non plus disponible en `.deb`, il ne reste qu'une seule solution : récupérer le code source du programme et le compiler soi-même, c'est-à-dire le transformer en binaire de telle sorte à ce qu'il soit compréhensible par la machine. À noter, certains distributeurs, comme Mozilla, publient leurs logiciels sous forme de code binaire, donc déjà compilé, ce qui raccourcit le processus. La marche à suivre devrait être la suivante mais n'en comprenant pas encore tous les enjeux, j'ai préféré reporter l'expérience à plus tard.

<https://doc.ubuntu-fr.org/tutoriel/compilation>

1. installer les outils de compilation :

```
sudo apt-get install build-essential
```

2. télécharger les sources du programme sur le web (souvent archivées au format `.tar.gz`)
3. décompresser l'archive (`tar zxvf archive.tar.gz`) :

`tar zxvf nom_programme.tar.gz`

4. exécuter `./configure` et résoudre les problèmes de dépendances :

`./configure`

5. lancer la compilation :

`make`

6. installer le programme :

`sudo make install`

- Redémarrer votre machine

`reboot` permet de redémarrer l'ordinateur. Il faut être root pour que cela fonctionne :

`anais@ubuntanais:~$ sudo reboot`

- Éteindre votre machine

`halt` commande l'arrêt immédiat de l'ordinateur en tuant tous les processus en cours et sans avertir personne. Il faut être root pour arrêter la machine :

`anais@ubuntanais:~$ sudo halt`

`shutdown` a la même fonction mais permet d'arrêter le système "proprement". Il laissera une minute à l'utilisateur pour fermer tous les processus en cours :

`anais@ubuntanais:~$ sudo shutdown`

Pour l'arrêter instantanément : `anais@ubuntanais:~$ sudo shutdown now`

JOB 4.

- Créer un fichier “users.txt” qui contiendra User1 et User2 séparé par un retour à la ligne

La commande **touch** est à la base faite pour modifier la date de dernière modification d'un fichier. D'où son nom : on « touche » le fichier pour faire croire à l'ordinateur qu'on vient de le modifier alors que l'on n'a rien changé. **touch** peut donc aussi être utilisé pour créer des fichiers, même s'il n'a pas vraiment été créé pour cela à l'origine.

```
anais@ubuntanais:~$ mkdir job4
anais@ubuntanais:~$ cd job4
anais@ubuntanais:~$ touch users.txt
anais@ubuntanais:~$ nano users.txt
    faire les modifications dans le document et sortir (ctrl + x)
```

- Créer un groupe appelé “Plateformeurs”

Seul root pour gérer les utilisateurs :

```
anais@ubuntanais:~/job4$ sudo addgroup Plateformeurs
```

Compte tenu du fait que le nom du groupe contient une majuscule et qu'il n'est pas d'usage d'en utiliser (cf. NAME_REGEX[SYSTEM]), le système nous invite à forcer la création avec **--force-badname**

```
anais@ubuntanais:~/job4$ sudo addgroup --force-badname Plateformeurs
```

- Créer un utilisateur appelé “User1”

```
anais@ubuntanais:~/job4$ sudo adduser --force-badname User1
```


On a créé un utilisateur User1, en forçant la prise en charge de la majuscule. On lui a assigné un mot de passe. Le système lui a automatiquement créé un groupe à son nom dans lequel il l'a ajouté.

À noter, avec `useradd`, on aurait créé un utilisateur mais qui n'aurait pas de session de travail automatiquement. Il aurait fallu le lui spécifier.

- Créer un utilisateur appelé "User2"

```
anais@ubuntanais:~/job4$ sudo adduser --force-badname User2
```

- Ajouter "User2" au groupe Plateformeurs

```
anais@ubuntanais:~/job4$ sudo usermod -g Plateformeurs User2
```

La commande `usermod` permet d'éditer un utilisateur. Elle possède plusieurs paramètres, et `-g` permet de le changer de groupe.

- Copier votre "users.txt" dans un fichier "droits.txt"

```
anais@ubuntanais:~$ cp users.txt droits.txt
```

Cette commande copie le fichier source ("users.txt") et le colle en le renommant ("droits.txt"). En effet, en faisant `man cp`, on peut lire :

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY

- Copier votre "users.txt" dans un fichier "groupes.txt"

```
anais@ubuntanais:~$ cp users.txt groupes.txt
```

- Changer le propriétaire du fichier "droits.txt" pour mettre "User1"

La commande `chown`, qui doit être utilisée en tant que root, attend deux paramètres au moins : le nom du nouveau propriétaire et le nom du fichier à modifier.

```
anais@ubuntanais:~$ sudo chown User1 droits.txt
```

Mais à noter, il appartient toujours au groupe anais. Il aurait fallu ajouter `:nom_groupe` pour l'affecter à un autre groupe d'utilisateurs :

```
anais@ubuntanais:~$ sudo chown User1:Plateformeurs droits.txt
```

- Changer les droits du fichier "droits.txt" pour que "User2" y ai accès seulement en lecture

Comme User2 appartient au groupe Plateformeurs, il faudra faire en sorte qu'il soit propriétaire du fichier en tant que groupe. Pour cela, `chgrp` s'utilise exactement de la même manière que `chown` à la différence près qu'il affecte cette fois le groupe propriétaire d'un fichier.

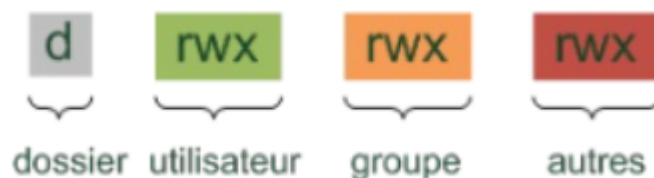
```
anais@ubuntanais:~$ sudo chgrp Plateformeurs droits.txt
```

Pour faire en sorte que User2, qui fait partie du groupe Plateformeurs, il faudra changer ses droits :

```
anais@ubuntanais:~/job4$ sudo chmod 644 droits.txt
```

644 correspond à l'autorisation read and write pour l'utilisateur, read only pour le groupe. Étant donné que User2 appartient au groupe Plateformeurs, il faut modifier les autorisations du groupe entier.

Les droits sont découpés en fonction des utilisateurs :



Voici la liste des droits possibles et la valeur correspondante :

Avec ça, on peut calculer la valeur d'un triplet de droits. Il faut faire le même calcul pour les droits que l'on veut attribuer au propriétaire, au groupe et aux autres. En l'occurrence ici, 644. À noter que tous les utilisateurs ajoutés au groupe Plateformeurs disposeront des mêmes autorisations.

Droits	Chiffre	Calcul
---	0	0 + 0 + 0
r--	4	4 + 0 + 0
-w-	2	0 + 2 + 0
--x	1	0 + 0 + 1
rw-	6	4 + 2 + 0
-wx	3	0 + 2 + 1
r-x	5	4 + 0 + 1
rwX	7	4 + 2 + 1

- Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement

En suivant la logique explicitée plus haut, saisissez :

```
anais@ubuntanais:~/job4$ sudo chmod 444 groupes.txt
```

- Changer les droits du fichier pour que le groupe "Plateformeurs" puisse y accéder en lecture/écriture.

```
anais@ubuntanais:~/job4$ sudo chgrp Plateformeurs groupes.txt
anais@ubuntanais:~/job4$ sudo chmod 464 groupes.txt
```

JOB 5.

<https://doc.ubuntu-fr.org/alias>

https://doc.ubuntu-fr.org/variables_d_environnement

- Ajouter un alias qui permettra de lancer la commande `ls -la` en tapant “la”

```
anais@ubuntanais:~$ alias ls -a='la'
```

Ici l'`alias` est créé uniquement pour anais. Si vous voulez définir des `alias` pour tous les utilisateurs, vous pouvez le faire en une seule fois en éditant en root le fichier `bashrc` global situé dans : `/etc/bash.bashrc`. Et si vous souhaitez rendre ces alias permanents, il faudra agir sur “`.bashrc`” dans le répertoire personnel de l'utilisateur.

- Ajouter un alias qui permettra de lancer la commande “`apt-get update`” en tapant “update”

```
anais@ubuntanais:~$ alias update='sudo apt-get update'
```

- Ajouter un alias qui permettra de lancer la commande “`apt-get upgrade`” en tapant “upgrade”

```
anais@ubuntanais:~$ alias upgrade='sudo apt-get upgrade'
```

- Ajouter une variable d'environnement qui se nommera “USER” et qui sera égale à votre nom d'utilisateur

Les variables d'environnement constituent un moyen d'influencer le comportement des logiciels sur votre système. Si nous utilisons une expression d'affectation pour une variable qui n'existe pas, l'interpréteur de commande créera une « variable de shell », qu'il faudra exporter pour qu'elle devienne une variable d'environnement. L'interpréteur de commandes `bash` propose un raccourci pour créer des variables d'environnement :

```
anais@ubuntanais:~$ export USER=ellenanou
```

Pour vérifier que la variable d'environnement a bien été créée, tapez :

```
anais@ubuntanais:~$ printenv USER
```

ou

```
anais@ubuntanais:~$ echo $USER
```

- Mettre à jour les modifications de votre bashrc dans votre shell actuel

```
anais@ubuntanais:~$ source ~/.bashrc
```

Compte tenu du fait que les variables d'environnement et les alias ont été créés uniquement pour cette session (et pas enregistrées dans “.bashrc”), elles sont toutes réinitialisées et effacées à la fermeture du terminal.

- Afficher les variables d'environnement

```
anais@ubuntanais:~$ printenv
```

- Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"

Le Path est utilisé sur Linux pour contenir des chemins vers des programmes. Ces chemins permettent d'exécuter directement une commande dans le terminal. Si vous exécutez cette commande directement dans le terminal, le répertoire sera ajouté, mais uniquement pour la durée de la session :

```
anais@ubuntanais:~$ export PATH=$PATH:/home/anais/Bureau
```

Pour vérifier :

```
anais@ubuntanais:~$ echo $PATH
```

Attention, ce qui est affiché ci-dessus n'est encore que temporaire et sera perdu au prochain redémarrage du système. Pour modifier la variable PATH de manière permanente, ajoutez la variable `export PATH=$PATH:/home/anais/Bureau` au fichier “.bashrc” situé dans le répertoire utilisateur. Les modifications ainsi apportées seront définitivement prises en charge au redémarrage.

JOB 6.

<https://lecrabeinfo.net/linux-compresser-decompresser-fichiers-dossiers-avec-tar-gzip-bzip2-xz.html>

- Télécharger une archive depuis internet et la désarchiver avec le terminal :

```
anais@ubuntanais:~$ wget https://drive.google.com/file/d/11dSeIXQuH4tih6zesbv-6OMEpr-sT77X/view?usp=sharing -O job6
```

`wget` est largement utilisé pour télécharger des fichiers depuis internet. Le `-O` permet de renommer le fichier à télécharger afin de simplifier sa manipulation.

Ici, il se trouve que le lien ne redirige pas vers une archive directement mais vers un drive donc la procédure ne peut pas se faire uniquement par le terminal. On téléchargera donc une copie de l'archive via le navigateur puis, on le décompressera. Au passage, on peut renommer le fichier :

```
anais@ubuntanais:~/Downloads$ mv 'Copie de Ghost in the Shell.tar.gz' ghost
anais@ubuntanais:~/Downloads$ tar -xzf ghost
```

JOB 7.

- Créer un fichier "une_commande.txt" avec le texte suivant "Je suis votre fichier texte"

```
anais@ubuntanais:~$ touch une_commande.txt
anais@ubuntanais:~$ nano une_commande.txt
    faire les modifications dans le document et sortir (ctrl + x)
```

Pour effectuer tout cela en une seule commande, on peut saisir :

```
anais@ubuntanais:~$ echo Je suis votre fichier texte > une_commande.txt
```

Pour vérifier : `anais@ubuntanais:~$ cat une_commande.txt`

- Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé "nb_lignes.txt"

Le "sources.list" est un fichier texte généralement situé dans /etc/apt/

L'ensemble des adresses web inscrites dans ce ou ces fichiers détermine les dépôts utilisés par votre système pour installer ou mettre à jour vos applications. Ce fichier ne peut être modifié que par root, l'administrateur de la machine.

La commande `wc` signifie *word count*. C'est donc a priori un compteur de mots mais en fait, on lui trouve plusieurs autres utilités : lignes, mots et octets (caractères). Pour avoir uniquement le nombre de lignes, utilisez `-l` :

`anais@ubuntanais:~$ wc -l /etc/apt/sources.list`

- Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save_sources"

Au lieu d'afficher le résultat d'une commande dans la console, grâce au chevron `>` il est possible de le rediriger dans un fichier ou en entrée d'une autre commande pour effectuer des chaînes de commandes.

Le double chevron `>>` sert lui aussi à rediriger le résultat dans un fichier, mais cette fois à la fin de ce fichier.

`anais@ubuntanais:~$ cat /etc/apt/sources.list > save_sources`

Pour vérifier : `anais@ubuntanais:~$ cat save_sources`

- Faire une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier

Pour savoir où se trouve un fichier avec `locate`, il faut d'abord installer le paquet :

`anais@ubuntanais:~$ sudo apt install plocate`

`anais@ubuntanais:~$ locate source save_sources`

Le problème avec `locate`, c'est que les fichiers tout juste créés ne seront pas encore répertoriés et donc introuvables. Vous pouvez forcer la commande `locate` à reconstruire la base de données des fichiers du disque dur. Cela se fait avec la commande `updatedb`, à exécuter en root (avec `sudo`).

Pour connaître les fichiers qui contiennent le mot "source" avec `grep` :

```
anais@ubuntanais:~$ grep source save_sources
```

Pour trouver les fichiers commençant par `.` qui contiennent la chaîne de caractère "alias" :

~~- avec `find` : `anais@ubuntanais:~$ find -name ".*" -type f -exec grep alias {} \;`~~

~~Contrairement à `locate`, `find` ne va pas lire dans une base de données mais au contraire parcourir tout le disque dur. Cela peut être très long mais pas de risque de rater un fichier récent ! Elle s'utilise : `find` « où » « quoi » « que faire avec » (par défaut, elle affiche les résultats trouvés et ne fait rien d'autre avec).~~

- avec `grep` : `anais@ubuntanais:~$ grep -r alias .*`

<https://www.tuteurs.ens.fr/unix/chercher.html>

Cette commande est beaucoup plus simple et permet de chercher "alias" dans tous les fichiers comportant un `.` du répertoire courant et de ses sous-répertoires, grâce au `-r`. Pour rester dans le home, il suffit d'ôter le `-r`.

POUR ALLER PLUS LOIN...

- Installer la commande `tree`

```
anais@ubuntanais:~$ sudo apt-get install tree
```

- Lancer la commande `tree` en arrière-plan qui aura pour but d'afficher toute l'arborescence en de votre / en enregistrant le résultat dans un fichier "tree.save"


```
anais@ubuntanais:~$ tree > tree.save
```

Sinon, en plus d'enregistrer la sortie dans un fichier, la commande `tee` permet d'afficher le résultat dans le terminal :

```
anais@ubuntanais:~$ tree | tee tree.save
```

- Lister les éléments présents dans le dossier courant et utiliser directement le résultat de votre première commande pour compter le nombre d'éléments trouvés

```
anais@ubuntanais:~$ ls -a | wc -l
```

`ls -a` permet de lister les fichiers et les fichiers cachés du dossier, et `wc` permet de compter le nombre de documents présents.

- Lancer une commande pour mettre à jour vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas

```
anais@ubuntanais:~$ sudo apt-get update
```

```
anais@ubuntanais:~$ sudo apt-get upgrade
```