

Shell

Ghost in the Shell

Fabien Ricca

Job 01

- Comment ajouter des options à une commande ?

Du point de vue d'Unix, une commande n'est rien d'autre qu'une liste d'arguments dont le premier représente l'exécutable. Une option définit le comportement de la commande, elle s'ajoute à la suite de la commande selon une ou plusieurs syntaxes.

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

Les options peuvent être saisies par leur nom entier (*list*), ou par la première lettre de leur nom (*l*) et elles peuvent également se combiner (*la* = *list* et *all*).

La position est la plus simple à mettre en œuvre mais limite les possibilités, il suffit d'un espace entre la commande et l'option. Elle est utilisée par des commandes comme *tar*. Par exemple, dans la commande *tar xzvf fichier.tar.gz* le premier paramètre correspond aux options de tar *x*=extract, *z*=gunzip, *v*=verbose, *f*=file.

Le tiret simple est la plus répandue surtout sur les anciens systèmes, elle consiste à précéder d'un tiret la lettre représentant l'option. Il est également possible avec cette convention de regrouper plusieurs options dans un seul argument. Cependant cette convention représente des inconvénients de lisibilité du code produit. Pour résoudre ce problème il existe le double tiret. Il permet de corriger le manque de lisibilité du simple tiret, mais il ne permet pas de combiner plusieurs options en une seule. Par contre, les options sont beaucoup plus parlantes : *mkdir --parents --verbose* est plus lisible que *mkdir -pv*.

1. Afficher le manuel de la commande *ls*

```
padawan@debian:~$ man ls
```

La commande **man** permet d'accéder aux pages de manuel de n'importe

```
LS(1) Commandes de l'utilisateur LS(1)
NOM
  ls - Afficher le contenu de répertoires
SYNOPSIS
  ls [OPTION]... [FICHIER]...
DESCRIPTION
  Afficher les informations des FICHIERS (du répertoire courant par défaut). Les entrées sont triées alphabétiquement si aucune des options -cftuvSUX ou --sort n'est indiquée.

  Les paramètres obligatoires pour les options de forme longue le sont aussi pour les options de forme courte.

  -a, --all
    inclure les entrées débutant par « . »
  -A, --almost-all
    omettre les fichiers « . » et « .. »
  --author
    avec -l, afficher l'auteur de chaque fichier
  -b, --escape
    afficher les caractères non graphiques sous la forme de caractères d'échappement de style C
  --block-size=TAILLE
    avec -l, ajuster les tailles avec TAILLE quand elles sont affichées ; par exemple « --block-size=M » ; voir le format de TAILLE ci-dessous
```

quelle commande en saisissant **man** avant la commande souhaitée, elle peut être utilisée sur elle même (**man man**). Elle est installée par défaut sur linux mais il est possible de rajouter des pages en installant les paquets correspondants.

2. Afficher les fichiers cachés du home de votre utilisateur

```
padawan@debian:~$ ls -d .*
. .bash_history .bash_logout .bashrc .cache .config .gnupg .local .mozilla .profile .ssh
```

La commande **Ls** est une des commandes de base, lorsqu'elle est utilisée sans options et arguments elle affiche une liste de tous les fichiers du répertoire de travail actuel. Ici nous avons utilisé la commande **ls** avec l'option **d.*** permettant de n'afficher que les dossiers, pas leur contenu.

Il est possible de n'afficher que les dossiers cachés en ajoutant un **/** à la fin de la ligne.

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

```
padawan@debian:~$ ls -la
total 80
drwxrwxrwx 16 padawan padawan 4096 15 sept. 11:09 
drwxr-xr-x  3 root    root    4096 14 sept. 16:56 ..
-rw-----  1 padawan padawan  504 15 sept. 13:48 .bash_history
-rw-r--r--  1 padawan padawan  220 14 sept. 16:56 .bash_logout
-rw-r--r--  1 padawan padawan 3526 14 sept. 16:56 .bashrc
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Bureau
drwx----- 12 padawan padawan 4096 15 sept. 11:09 .cache
drwx----- 11 padawan padawan 4096 15 sept. 09:44 .config
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Documents
drwx-----  2 padawan padawan 4096 15 sept. 11:20 .gnupg
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Images
drwxr-xr-x  3 padawan padawan 4096 14 sept. 16:57 .local
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Modèles
drwx-----  5 padawan padawan 4096 15 sept. 11:09 .mozilla
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Musique
-rw-r--r--  1 padawan padawan  807 14 sept. 16:56 .profile
drwxr-xr-x  2 padawan padawan 4096 14 sept. 16:57 Public
drwx-----  2 padawan padawan 4096 14 sept. 16:57 .ssh
```

La commande **ls -la** (**l** = affiche sous forme de liste détaillée; **a** = affiche tous les éléments) est la méthode la plus courante, elle permet d'afficher tous les dossiers et fichiers y compris les cachés,

```
padawan@debian:~$ ls -ld .*
drwxrwxrwx 16 padawan padawan 4096 15 sept. 11:09 .
drwxr-xr-x  3 root     root     4096 14 sept. 16:56 ..
-rw-----  1 padawan padawan  504 15 sept. 13:48 .bash_history
-rw-r--r--  1 padawan padawan  220 14 sept. 16:56 .bash_logout
-rw-r--r--  1 padawan padawan 3526 14 sept. 16:56 .bashrc
drwx----- 12 padawan padawan 4096 15 sept. 11:09 .cache
drwx----- 11 padawan padawan 4096 15 sept. 09:44 .config
drwx-----  2 padawan padawan 4096 15 sept. 11:20 .gnupg
drwxr-xr-x  3 padawan padawan 4096 14 sept. 16:57 .local
drwx-----  5 padawan padawan 4096 15 sept. 11:09 .mozilla
-rw-r--r--  1 padawan padawan  807 14 sept. 16:56 .profile
drwx-----  2 padawan padawan 4096 14 sept. 16:57 .ssh
```

Cependant, il est plus utile d'utiliser la commande **ls -ld .*** afin d'afficher uniquement les dossiers et les fichiers cachés.

Job 02

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

```
padawan@debian:~$ cat .bashrc
```

La commande **cat** (concatenate) est une des commandes les plus répandues, elle permet entre autres de créer, de fusionner, d'afficher des fichiers dans le terminal... Elle prend pour argument un nom de fichier. C'est également la commande la plus utilisée pour afficher un document en lecture seule.

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

```
padawan@debian:~$ head .bashrc
```

La commande **head** sert à afficher les N premières lignes d'un document à la sortie standard, par défaut elle en affichera les 10 premières, mais ce chiffre peut être modifié avec une option (2, 20, 30...). Il est aussi possible d'afficher toutes les lignes en excluant les N dernières.

3. Afficher les 10 dernières lignes du fichier ".bashrc"

```
padawan@debian:~$ tail .bashrc
```

La commande **tail** permet de lire la fin d'un fichier, par défaut les 10 dernières lignes, mais il est possible tout comme **head** de modifier ce nombre comme on le souhaite. Mais elle peut également servir à surveiller en temps réel les modifications apportées à un fichier spécifique, ou encore de permettre à un seul utilisateur de consulter plusieurs fichiers avec une seule commande.

4. Afficher les 20 premières lignes du fichier ".bashrc"

```
padawan@debian:~$ head -20 .bashrc
```

Comme dit précédemment, en ajoutant une option on peut décider du nombre de lignes à afficher en partant du début du document.

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

```
padawan@debian:~$ tail -20 .bashrc
```

Comme dit précédemment, en ajoutant une option on peut décider du nombre de lignes à afficher en partant de la fin du document.

Job 03

1. Installer le paquet “cmatrix”

Cmatrix est un petit utilitaire qui n’a aucune utilité si ce n’est d’apporter un peu d’esthétique tout en verrouillant le terminal. Pour pouvoir l’utiliser il faut tout d’abord installer son paquet, pour cela nous allons utiliser la commande **sudo apt-get install cmatrix**. La commande **apt-get** est un outil-logiciel qui permet d’effectuer l’installation et la désinstallation de paquets en provenance d’un depot APT. Si **apt-get** fournit un retour facilement analysable par un script et garantit une certaine rétrocompatibilité, il est toutefois plus convivial d’utiliser simplement **apt**, principalement si le but est d’accéder directement à ces fonctionnalités via la ligne de commande.

```
padawan@Padawan-computer:~$ sudo apt-get install cmatrix
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  cmatrix-xfont
Les NOUVEAUX paquets suivants seront installés :
  cmatrix
0 mis à jour, 1 nouvellement installés, 0 à enlever et 4 non mis à jour.
Il est nécessaire de prendre 17,1 ko dans les archives.
Après cette opération, 57,3 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://fr.archive.ubuntu.com/ubuntu jammy/universe amd64 cmatrix amd64 2.0-3 [17,
1 kB]
17,1 ko réceptionnés en 0s (39,3 ko/s)
Sélection du paquet cmatrix précédemment désélectionné.
(Lecture de la base de données... 203312 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../cmatrix_2.0-3_amd64.deb ...
Dépaquetage de cmatrix (2.0-3) ...
Paramétrage de cmatrix (2.0-3) ...
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmu1ubuntu1) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1ubuntu3) ...
Traitement des actions différées (« triggers ») pour gnome-menus (3.36.0-1ubuntu3) ...
Traitement des actions différées (« triggers ») pour man-db (2.10.2-1) ...
```

2. lancer le paquet que vous venez d’installer

Une fois télécharger, il suffit simplement de lancer cmatrix en tapant son nom directement dans le terminal.

```
padawan@Padawan-computer:~$ cmatrix
```

3. Mettre à jour son gestionnaire de paquets

Afin de mettre à jour la liste des fichiers disponibles dans les dépôts APT présents dans le fichier de configuration `/etc/apt/sources.list`, il suffit de rajouter l'option `update` derrière la commande `apt-get`. Il est important de l'exécuter régulièrement est une bonne pratique afin de maintenir à jour la liste de paquets disponibles.

```
padawan@Padawan-computer:~$ sudo apt-get update
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Atteint :2 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Atteint :3 https://dl.google.com/linux/chrome/deb stable InRelease
Atteint :4 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease
Atteint :5 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
110 ko réceptionnés en 1s (126 ko/s)
Lecture des listes de paquets... Fait
```

4. Mettre à jour ses différents logiciels

Il existe plusieurs façons de procéder afin de mettre à jour les différents logiciels. La plus commune consiste simplement à rajouter l'option `update` derrière la commande `apt-get`, elle mettra à jour tous les paquets installés sur le système vers les dernières versions.

Il est aussi possible d'utiliser l'option `dist -upgrade` afin de mettre à jour tous les paquets installés vers les dernières versions en installant de nouveaux paquets si nécessaire, par opposition à l'upgrade simple qui n'installe pas de nouveaux paquets.

A noter qu'il est également possible de simuler la mise à jour des paquets sans réellement les installer grâce à l'option `- -simulate upgrade`.

```
padawan@Padawan-computer:~$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants seront mis à jour :
 gcc-12-base libgcc-s1 libgomp1 libstdc++6
4 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
4 standard security updates
Il est nécessaire de prendre 896 ko dans les archives.
Après cette opération, 4 096 o d'espace disque seront libérés.
Souhaitez-vous continuer ? [O/n] o
```


5. Télécharger les internets : Google

Pour télécharger la page internet www.google.com, nous allons utiliser la commande **wget**. Cette commande permet le téléchargement de fichiers depuis le web, elle supporte les protocoles HTTP, HTTPS et FTP ainsi que les téléchargements au travers des proxies HTTP. Elle peut travailler en arrière-plan, permettant de lancer un téléchargement et de se déconnecter du système, ce qui peut être utile car elle ne requiert pas d'action de l'utilisateur et permet d'effectuer ses tâches en arrière-plan. Cette commande permet le téléchargement simple, mais aussi la copie en local de tout en partie d'un site qui sera par la suite consultable hors ligne.

Pour télécharger Google, nous allons donc saisir la commande suivante : **wget www.google.com**. Le résultat est un fichier `index.html` unique contenant le contenu extrait du site stocké dans un dossier au nom du site.

```
padawan@Padawan-computer:~$ sudo wget -r www.google.com
--2022-09-19 12:12:28-- http://www.google.com/
Résolution de www.google.com (www.google.com)... 142.250.201.164, 2a00:1450:4006:80e::2004
Connexion à www.google.com (www.google.com)[142.250.201.164]:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Enregistre : 'www.google.com/index.html'
```

Pour télécharger des sites complets et toutes les pages il faudrait rajouter l'option **-r**, cela va télécharger de manière récursive jusqu'à 5 niveaux de profondeur. Mais cela pourrait ne pas suffire pour obtenir tout le site, on pourrait donc utiliser l'option **-l** pour définir le nombre de niveaux auxquels on souhaite accéder (**wget -r -l10 www.google.com**), il est possible de mettre une récursion infinie en précisant **-l inf** ou **-lo**.

Il reste cependant encore un problème, nous avons obtenu les pages localement, mais les liens dans les pages pointent vers l'emplacement d'origine, il n'est pas possible de cliquer localement entre les liens sur les pages. Pour contourner ce problème, on peut utiliser le **-k** pour convertir les liens sur les page pour pointer vers l'équivalent télécharger localement.

Pour obtenir un miroir complet d'un site web, on peut utiliser le commutateur **-m**, ce qui élimine la nécessité d'utiliser les **-r -l -k**. Cette simple commande permet d'effectuer une sauvegarde complète d'un site web.

6. Redémarrer votre machin et éteindre votre machine

On entend par shutdown l'arrêt ou la désactivation d'un ordinateur ou d'un serveur. Dans ce cadre, l'alimentation des principales composantes du système est coupée de façon contrôlée : les programmes utilisateurs sont arrêtés, les processus et les protocoles actifs sont enregistrés sur le disque dur, les pilotes des appareils sont supprimés et les paramètres utilisateur sont enregistrés. Avec ses différentes options, la commande **shutdown** permet d'arrêter, de désactiver ou de redémarrer simplement les systèmes d'exploitation Linux.

On peut lui ajouter une option suivie par une heure et un message : **shutdown [option] [heure] [message]**.

```
padawan@Padawan-computer:~$ sudo shutdown -h now
```

Linux s'arrête après une minute, l'option **-h** renvoie explicitement au fait d'éteindre l'ordinateur ou de désactiver le système. On peut obtenir le même résultat en saisissant la commande **shutdown**. l'option **-h o** arrête immédiatement l'ordinateur.

```
padawan@Padawan-computer:~$ sudo shutdown -r now
```

Linux redémarre après une minute, l'option **-r** renvoie ici à reboot, et now provoque un redémarrage immédiat.

Il existe également les commandes **halt**, **poweroff** et **reboot** mais elles sont moins recommandées car elles arrêtent tous les processus en cours sans aucun avertissement.

Job 04

o. Créer un fichier users.txt

Il existe plusieurs commandes permettant de créer des fichiers : *cat*, *echo*, *heredoc*, *dd* ou *fallocate* (pour créer des fichiers volumineux)...

Nous allons ici utiliser la commande *nano users.txt* qui permet de créer un fichier et d'en saisir le contenu ("Je suis votre fichier texte") en passant par l'éditeur de texte nano.

1. Créer un groupe appelé "Plateformeurs"

La commande *groupadd Plateformeurs* permet de créer un nouveau compte de groupe en utilisant les valeurs spécifiées sur la ligne de commande et les valeurs par défaut du système. Le nouveau groupe sera inséré dans les fichiers du système selon les besoins.

2. Créer des utilisateurs appelés "User1" et "User2"

Afin de créer des utilisateurs il a fallu utiliser la commande *useradd* deux fois avec à chaque fois le nom du nouvel utilisateur en option : *useradd User1* et *useradd User2*.

3. Ajouter "User2" au groupe Plateformeurs

Une fois le groupe et les utilisateurs créés, nous allons pouvoir déplacer le User2 dans le groupe Plateformeurs en utilisant la commande suivante : *adduser User2 Plateformeurs*

4. Copier votre "users.txt" dans un fichier "droits.txt"; Puis de "users.txt" dans un fichier "groupes.txt"

La commande *cat users.txt >> droits.txt* va permettre d'afficher (déplacer) le contenu du *users.txt* dans le fichier *droits.txt*. Mais étant donné que le fichier *users.txt* n'existait pas au moment où la commande a été saisie, il va donc le créer automatiquement. Les chevrons sont des commandes de redirection, pour préciser le sens dans la commande. Il suffit de reproduire la commande pour le second fichier : *users.txt >> groupes.txt*.

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

Il existe plusieurs commande pour modifier les droits et permissions sur linux :

- chown (change owner) pour changer le propriétaire et/ou le groupe;
- chgrp (change group) pour changer le groupe;
- chmod (change mod) pour modifier les droits et permissions.

Pour changer nous allons donc utiliser la commande `sudo chown User1 droits.txt`.

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

Le contrôle d'accès sous les systèmes Unix, et par héritage sous Linux, est dit discrétionnaire, c'est-à-dire que tous les objets (répertoires, fichiers, processus, etc.) sont la propriété d'un compte utilisateur ou système, ainsi que d'un groupe de comptes utilisateurs et/ou système. Les droits associés à ces objets sont donc définis :

- Pour le propriétaire noté u (pour "user") ;
- Pour le groupe propriétaire noté g (pour "group") ;
- Pour tous les autres, notés eux o (pour "others").

On peut attribuer des droits sur un objet à l'aide de 9 bits, répartis en 3 groupes de 3 dans l'ordre suivant :

1. rwx pour u
2. rwx pour g
3. rwx pour o

Ce qui nous donne le résultat suivant:

`rwX-rwX-rwX`

1ère groupe de permissions --> propriétaire

2ème groupe de permissions --> groupe

3ème groupe de permissions --> les autres (le reste)

Pour raccourcir les lignes de commandes d'attribution des permissions, il existe un moyen très pratique, en passant notamment par la valeur numérique de chaque bit de permission. En effet, en considérant que :

- le bit x est en position 0, sa valeur lorsqu'il est positionné à 1, en base 2 est donc $1 \times 2^{\text{exp}0}$, soit 1 ;
- Le bit w est en position 1, sa valeur lorsqu'il est positionné à 1, en base 2 est donc $1 \times 2^{\text{exp}1}$, soit 2 ;

- Le bit r est en position 2, sa valeur lorsqu'il est positionné à 1, en base 2 est donc $1 \times 2^{\text{exp}2}$, soit 4 ;

Nous avons donc une valeur en base décimale équivalente à chaque combinaison possible de ces 3 bits telle que :

- - - - > 000 > 0 > Aucun droit
- - - x > 001 > 1 > Exécution
- - w - > 010 > 2 > Écriture
- - w x > 011 > 3 > Écriture et exécution
- r - - > 100 > 4 > Lecture
- r - x > 101 > 5 > Lecture et exécution
- r w - > 110 > 6 > Lecture et écriture
- r w x > 111 > 7 > Lecture, écriture et exécution

Ces valeurs allant de 0 à 7 exprimant les 3 bits des permissions sont dites octales.

On utilise la commande *chmod* en indiquant l'utilisateur (*user*, *group*, *other* ou *all*), les actions (+ ajouter; - enlever; = affecter) et le droit concerné (*r*, *w*, *x*). Afin de modifier les droits des différents utilisateurs en même temps on peut utiliser le système octale : *chmod 750 nom_du_fichier*, dans cet exemple nous avons donné tous les droits au propriétaire du fichier, la lecture et l'exécution seulement au groupe, et aucun aux autres utilisateurs.

Afin que User2 n'ait accès à ce fichier qu'en lecture, nous allons changer le groupe de droits.txt pour Plateformeurs, et modifier les droits de celui-ci. De cette façon User2, qui fait partie de ce groupe, n'aura que la lecture seule :

```
sudo chgrp Plateformeurs droits.txt
sudo chmod g-w droits.txt
```

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

En suivant la même technique qu'au-dessus, nous allons modifier les droits de 'others' à l'aide de la commande *chmod* cette fois-ci : *sudo chmod o=r groupes.txt*.

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

En suivant la même logique nous allons d'abord changer le groupe du fichier avec *sudo chgrp Plateformeurs groupes.txt*, puis affecter les droits de lecture et d'écriture au groupe en utilisant *sudo chmod g=r groupes.txt* et *sudo chmod g=w groupes.txt*.

Job 05

1. Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la"

Les alias sont des substitutions abrégées de commandes répétitives et/ou longues à taper à taper dans la console. On peut définir des alias dans deux fichiers cachés :

- dans `.bashrc` juste après la ligne "some more ls aliases".
- dans `.bash_aliases`, s'il n'existe pas il faut le créer.

Pour créer des alias nous allons donc commencer par créer le fichier `.bash_aliases` dans `/home/padawan` : `nano .bash_aliases`, puis nous allons éditer notre premier alias directement à l'intérieur en tapant *alias la="ls -la"*. une fois créer, taper seulement `la` dans le terminal ne fonctionnera pas, il faut d'abord recharger le `.bashrc` pour que les nouvelles alias soient prises en compte : *source .bashrc*. Maintenant à chaque fois que nous saisissons *la*, cela lancera la commande *ls -a*, même après redémarrage de l'ordinateur.

Il est aussi possible de créer des alias de manière temporaire (uniquement pour la session du terminal en cours, il faut pour cela saisir la commande alias dans le terminal sous la même forme, l'alias sera effective jusqu'à la fermeture du terminal.

2. Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update"

Nous allons donc procéder comme expliqué plus tôt, ajouter le nouvel alias directement dans le fichier `.bash_aliases`, avant de recharger le bash :

```
alias update="sudo apt-get update"  
source .bashrc
```

3. Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade"

Nous allons donc procéder comme expliqué plus tôt, ajouter le nouvel alias directement dans le fichier `.bash_aliases`, avant de recharger le bash :

```
alias upgrade="sudo apt-get upgrade"  
source .bashrc
```

4. Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur

Une variable d'environnement est un moyen d'influencer le comportement des logiciels/applications sur le système (la variable LANG détermine la langue que les logiciels utilisent pour communiquer avec l'utilisateur). La syntaxe utilisée doit être `NOM_VARIABLE=valeur`, nous allons donc créer notre propre variable d'environnement `USER=padawan`. Une fois créé, il faut comme pour les alias recharger le bash avec `source .bashrc` afin de l'enregistrer définitivement.

5. Mettre à jour les modifications de votre bashrc dans votre shell actuel

`.bashrc` est un script shell bash que bash s'exécute chaque fois qu'il démarre de manière interactive. Il initialise une session shell interactive. Il contient des configurations pour la session de terminal. Ces configurations incluent la coloration, l'historique du shell, l'achèvement, les alias de commande, les variables d'environnement et bien d'autres. On recharge le `.bashrc` pour refléter les nouvelles modifications, la commande `source .bashrc` lit et exécute le contenu du fichier `.bashrc`.

Une autre façon de recharger les modifications consiste à exécuter `exec .bashrc`, ce qui va remplacer le shell bash actuel par une nouvelle instance.

6. Afficher les variables d'environnement

Il y a quelques commandes qui peuvent être utiles pour afficher des variables ou leur valeur, ou la modifier :

- `printenv` affiche toutes les variables.
- `echo $NOM_VARIABLE` donne la valeur de la variable
- `echo NOM_VARIABLE` donne le nom de la variable
- `export NOM_VARIABLE="valeur"` change la valeur

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

Le PATH est utilisé pour contenir des chemins d'accès vers des programmes, ces chemins permettent d'exécuter directement une commande dans le terminal.

On peut ajouter un répertoire au PATH avec la commande `export PATH=$PATH:/home/padawan/Bureau`, mais si cette commande est utilisée directement dans le terminal le répertoire sera ajouté mais uniquement pour la durée de la session. Si on souhaite que cette modification reste effective, il faut ajouter la ligne de PATH dans le fichier `.bashrc`, puis le recharger afin d'enregistrer la modification.

Job 06

1. Télécharger l'archive suivante et la désarchiver seulement avec le terminal.

<https://drive.google.com/file/d/11dSelXQuH4tih6zesbv-6OMEpr-sT77X/view?usp=sharing>

la commande pour télécharger un fichier est *wget suivi_du_lien_du_fichier*. Le problème dans cet exercice étant que le lien n'amène pas vers un fichier mais vers un partage visuel, le téléchargement du fichier n'est donc pas possible par ligne de commande. Il a donc fallu se le procurer d'une autre façon (un téléchargement classique depuis le drive). Il existe d'autres commandes pour télécharger des fichiers comme *curl* ou *gdown* (pour les fichiers volumineux).

Une fois l'archive téléchargée il va falloir la décompresser, pour ce faire nous allons utiliser la commande *tar -xvf nom_archive.tar.gz* (*x* = indique à *tar* d'extraire l'archive; *v* = mode verbeux, affiche la progression; *f* = permet de spécifier le nom du fichier d'archive).

Job 07

1. *Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte”*

Afin de n'écrire qu'une seule ligne de commande, nous allons ici utiliser la commande *echo* “Je suis votre fichier texte” > “une_commande.txt”. Elle va afficher la chaîne de caractères “je suis votre fichier texte”. Cependant la sortie standard d'affichage est le terminal, nous allons donc utiliser un chevron (>) afin de rediriger l'affichage vers le fichier une_commande.txt. Étant donné que celui-ci n'existe pas, il va être créé automatiquement. Le nouveau fichier contiendra donc la chaîne de caractère voulue.

2. *Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt”*

Nous avons ici utilisé la commande *wc -l /etc/apt/sources.list* > “nb_lignes.txt”. La commande *wc* permet de compter et d'afficher le nombre de lignes, de mots, de caractères et d'octets si aucune option n'est précisée. Ici, l'option *-l* indique à la commande qu'il ne faut compte que les lignes du document spécifié par le chemin d'accès. De nouveau le chevron va préciser la nouvelle sortie pour le résultat de la commande, ici le fichier nb_lignes.txt, celui-ci va donc être créé automatiquement également.

3. *Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé “save_sources”*

Dans cet exercice nous avons préféré la commande *cat* qui permet d'afficher le contenu d'un fichier entier, contrairement à la commande *echo* qui va être utilisée pour afficher une chaîne de caractère ou une variable. Avec toujours l'utilisation du chevron afin d'indiquer la nouvelle sortie et de créer un nouveau fichier dans le même temps *cat /etc/apt/sources.list* > “save_sources”.

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

Grep est une commande qui permet de rechercher des chaînes de caractères dans un flux de texte (fichier ou sortie d'une autre commande), il en existe plusieurs variantes comme **agrep** (recherche d'une chaîne approximative), **zgrep** (recherche d'un fichier compressé) ou **rgrep** (recherche dans tous les fichiers d'un répertoire), mais il en existe d'autres avec chacune leurs spécificités (**egrep**, **fgrep**, **tcgrep**, **pgrep**...). Il est aussi possible d'affiner la recherche avec différentes options (**-color** = fait ressortir le terme recherché en couleur; **-r** recherche récursive; **-c** retourne le nombre d'occurrences...).

Commande utilisée pour l'exercice : **ls -a | grep .* | grep alias**

Pour aller plus loin...

1. Installer la commande tree

La commande tree permet de montrer de manière graphique et structurée la hiérarchie des répertoires de notre système d'exploitation. Elle permet également de lister les répertoires des périphériques externes. Pour l'installer il faut donc utiliser l'utilitaire habituel d'installation : *sudo apt-get install tree*.

2. 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"

Sur linux un processus désigne un programme en cours d'exécution, le lancement de chaque commande démarre un processus. Un processus peut être lancé de deux manières : foreground process (premier plan) et background process (arrière-plan). Lancer un processus en arrière-plan permet de continuer d'utiliser le terminal.

Pour illustrer, nous allons utiliser la commande *top* (un gestionnaire de tâche) qui est un programme qui utilise tout le terminal en continu jusqu'à ce que l'on décide de le terminer.

Il y a deux façons pour lancer un programme en arrière plan, la première et la plus simple consiste à rajouter le symbole *&* (esperluette) à la fin de la commande. Le terminal nous renverra le PID du process en terminal :

```
top &
```

```
[1] 6997
```

la Seconde méthode consiste à lancer la commande normalement, puis de la mettre en pause en utilisant le raccourci clavier *Ctrl+Z*, pour enfin saisir la commande *bg* (Backgroud) pour l'envoyer en arriere plan.

```
top
```

```
Ctrl+Z
```

```
bg
```

Pour relancer le programme en arrière plan il suffira de saisir la commande *fg* (Foreground).

Pour en revenir à notre exercice il faut donc lancer la commande de cette façon : *tree > tree.save &*

3. *lister les éléments présents dans le dossier courant et utilisé directement le résultat de votre première commande pour compter le nombre d'éléments trouvés*

Pour créer des commandes un peu plus complexes on peut utiliser des opérateurs logiques come *&*, *|*, *&&* ou *||*. L'*&* (esperluette) est utilisé pour lancer un programme en arrière plan. La *|* (pipe) sert à séparer deux commandes. Le double *&&* est un Et logique (n'exécute la seconde commande que si la première est ok). le double *||* est un Ou logique (n'exécute la seconde commande que si la première n'est pas ok).

Ici par exemple nous allons lister tous les fichiers et dossiers du répertoire actuel, puis compter tous les éléments de cette liste : *ls -la | wc -l*.

4. *Lancer une commande pour update 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*

Ici nous avons utilisé les alias créés plus haut, combiné à la double esperluette : *update && upgrade*.