

Apprivoiser la ligne de commande

Attribution - Partage dans les Mêmes Conditions : <http://creativecommons.org/licenses/by-sa/4.0/fr/>

Table des matières

1. Linux Journey	5
2. L'environnement de commandes (Le Shell)	6
2.1. L'environnement de commandes (Le Shell)	6
2.2. Exercice	6
2.3. Quiz.....	6
3. pwd (Print Working Directory)	7
3.1. pwd (Print Working Directory).....	7
3.2. Exercice	7
3.3. Quiz.....	7
4. cd (Change Directory)	8
4.1. cd (Change Directory).....	8
4.2. Exercice	8
4.3. Quiz.....	8
5. ls (List Directories)	9
5.1. ls (List Directories).....	9
5.2. Exercice	9
5.3. Quiz.....	9
6. touch	10
6.1. touch.....	10
6.2. Exercice	10
6.3. Quiz.....	10
7. file	11
7.1. file	11
7.2. Exercice	11
7.3. Quiz.....	11
8. cat	12
8.1. cat.....	12
8.2. Exercice	12
8.3. Quiz.....	12
9. less	13
9.1. less	13

9.2. Exercice.....	13
9.3. Quiz.....	13
10. history	14
10.1. history.....	14
10.2. Exercice.....	14
10.3. Quiz.....	14
11. cp (Copy)	15
11.1. cp (Copy).....	15
11.2. Exercice.....	15
11.3. Quiz.....	15
12. mv (Move)	16
12.1. mv (Move).....	16
12.2. Exercice.....	16
12.3. Quiz.....	16
13. mkdir (Make Directory)	17
13.1. mkdir (Make Directory)	17
13.2. Exercice.....	17
13.3. Quiz.....	17
14. rm (Remove)	18
14.1. rm (Remove)	18
14.2. Exercice.....	18
14.3. Quiz.....	18
15. find	19
15.1. find.....	19
15.2. Exercice.....	19
15.3. Quiz.....	19
16. help	20
16.1. help.....	20
16.2. Exercice.....	20
16.3. Quiz.....	20
17. man	21
17.1. man.....	21
17.2. Exercice.....	21
17.3. Quiz.....	21

18. whatis	22
18.1. whatis	22
18.2. Exercice	22
18.3. Quiz.....	22
19. alias	23
19.1. alias	23
19.2. Exercice	23
19.3. Quiz.....	23
20. exit	24
20.1. exit	24
20.2. Exercice	24
20.3. Quiz.....	24

1. Linux Journey

Ce module a été réalisé à partir des contenus disponibles sur le site Linux Journey.

De nombreux autres modules existent en anglais sur ce site.



<https://linuxjourney.com>

2. L'environnement de commandes (Le Shell)

2.1. L'environnement de commandes (Le Shell)

Le monde est à votre portée, ou plus précisément l'environnement (ou interpréteur) de commandes (le shell) est à votre portée. C'est quoi l'environnement de commandes? L'environnement de commandes est simplement un programme qui récupère vos commandes du clavier et les envoie au système d'exploitation pour exécution. Si vous avez déjà utilisé une interface graphique, vous avez certainement vu des programmes comme "Terminal" ou "Console"; ce sont juste des programmes qui exécutent l'environnement de commandes pour vous. Tout au long de ce cours, nous allons apprendre les merveilles de l'environnement de commandes.

Dans ce cours, nous utiliserons l'environnement de commandes bash (Bourne Again shell), présent dans presque toutes les distributions Linux. Il existe d'autres environnements de commandes tels que ksh, zsh, tsch, mais nous n'en parlerons pas.

Commençons! En fonction de la distribution, votre invité de commandes peut différer, mais la plupart suivent le format suivant:

```
1 nom_utilisateur@nom_de_la_machine:repertoire_courant
2 pete@icebox:/home/pete $
```

Avez-vous remarqué le \$ à la fin de la commande? La commande peut varier en fonction de l'environnement de commandes, dans notre cas, le \$ est pour un utilisateur normal de l'environnement de commandes Bash, Bourne ou Kon. Vous ne précisez pas ce symbole lorsque vous tapez la commande, sachez juste qu'il est là.

Commençons avec une simple commande, echo. La commande echo affiche juste le texte en paramètre.

```
1 $ echo Hello World
```

2.2. Exercice

Essayez d'autres commandes Linux et voyez ce qu'elles affichent:

1. \$ date
2. \$ whoami

2.3. Quiz

Que devrait s'afficher quand vous saisissez la commande echo Hello World?

3. pwd (Print Working Directory)

3.1. pwd (Print Working Directory)

Tout dans Linux est un fichier, vous comprendrez cela dans votre voyage dans Linux, mais retenez ça pour le moment. Chaque fichier est organisé selon une arborescence hiérarchique. Le premier répertoire dans le système de fichiers est appelé le répertoire racine. Le répertoire racine contient beaucoup de dossiers et fichiers dont vous pouvez stocker plus de répertoires et fichiers, etc. Ceci est un exemple d'arborescence:

```
1 /
2 |-- bin
3 |   |-- fichier1
4 |   |-- fichier2
5 |-- etc
6 |   |-- fichier3
7 |   |-- répertoire1
8 |       |-- fichier4
9 |       |-- fichier5
10 |-- home
11 |-- var
12
```

L'emplacement de ces fichiers et répertoires est indiqué par des chemins. Si vous avez un répertoire nommé "home" avec un dossier à l'intérieur appelé "pete" et lui même possédant un dossier nommé "Movies", le chemin serait comme ceci: /home/pete/Movies, très simple hein?

La navigation dans le système de fichiers, un peu comme dans la vraie vie, est utile si vous savez où vous êtes et où vous allez. Pour voir où vous êtes, vous pouvez utiliser la commande pwd, elle signifie "print working directory" (affiche le répertoire de travail) et il vous montre le répertoire dans lequel vous êtes. Notez que le chemin commence au répertoire racine.

```
1 $ pwd
```

Où êtes-vous? Où suis-je? Essayez.

3.2. Exercice

Pas d'exercices dans cette leçon.

3.3. Quiz

Comment trouve t-on le répertoire dans lequel on se trouve actuellement?

4. cd (Change Directory)

4.1. cd (Change Directory)

Maintenant que vous savez où vous vous trouvez, voyons si nous pouvons un peu parcourir le système de fichiers. Souvenez-vous, nous nous déplacerons en utilisant des chemins. Il existe deux façons différentes de donner un chemin: chemins absolu et relatif.

- Chemin absolu: C'est le chemin ayant pour origine le répertoire racine. La racine est le grand patron. Le répertoire racine est généralement représenté à l'aide d'un slash (/). Chaque fois que votre chemin commence avec un /, cela signifie que vous venez du répertoire racine. Par exemple, /home/pete/Desktop.
- Chemin relatif: C'est le chemin ayant pour origine le répertoire dans lequel vous vous trouvez. Si je me trouvais dans l'emplacement /home/pete/Documents et voudrais me rendre dans le répertoire "taxes" se trouvant dans Documents, j'ai pas besoin de préciser le chemin entier à partir de la racine comme ceci /home/pete/Documents/taxes, je peux juste faire ça à la place: taxes/

Maintenant que vous savez comment les chemins fonctionnent, nous avons besoin de quelque chose pour nous aider à nous rendre dans le répertoire que nous voulons. Heureusement, nous avons cd ou "change directory" (changer de répertoire) pour le faire.

```
1 $ cd /home/pete/Pictures
```

```
1 $ cd Hawaii
```

- . (répertoire courant). C'est le répertoire dans lequel vous êtes actuellement.
- .. (répertoire parent). Vous amène dans le répertoire au dessus de votre répertoire courant.
- ~ (répertoire home). Ce répertoire conduit à votre "répertoire maison" par défaut (on parle de "home"). Comme /home/pete.
- - (répertoire précédent). Il vous conduira au répertoire précédent dont vous venez.

```
1 $ cd .  
2 $ cd ..  
3 $ cd ~  
4 $ cd -  
5
```

Essayez!

4.2. Exercice

1. Exécutez la commande cd sans paramètres, où vous conduit-il?

4.3. Quiz

Si vous êtes dans /home/pete/pictures et voulez aller à /home/pete, quel est le bon raccourci à utiliser?

5. Is (List Directories)

5.1. Is (List Directories)

Maintenant que nous savons comment se déplacer dans le système, comment savoir ce qui nous est accessible? Actuellement, c'est comme si on progressait dans le noir. Bien, nous pouvons utiliser la formidable commande `ls` pour lister le contenu d'un répertoire. La commande `ls` va lister les dossiers et fichiers présents dans le répertoire où vous vous trouvez. Cependant, vous pouvez préciser le chemin du répertoire dont vous voulez lister le contenu.

```
1 $ ls
2 $ ls /home/pete
```

`ls` est un outil très utile, il vous montre aussi les informations sur les fichiers et dossiers que vous regardez.

Notez aussi que ce ne sont pas tous les fichiers dans le répertoire qui seront visibles. Les noms de fichiers commençant avec "." sont cachés, vous pouvez cependant les voir en ajoutant le paramètre "-a" (a pour dire "tout", all en anglais) à la commande `ls`.

```
1 $ ls -a
```

Il existe aussi un autre paramètre utile, "-l" pour "long", il donne une liste détaillée des fichiers dans un grand (long) format. Il vous affiche des informations détaillées, allant de la gauche: droit d'accès, nombre de liens, nom du propriétaire, groupe propriétaire, taille, date de la dernière modification, et le nom du fichier/répertoire.

```
1 $ ls -l
1 pete@icebox:~$ ls -l
2 total 80
3 drwxr-x--- 7 pete penguingroup 4096 Nov 20 16:37 Desktop
4 drwxr-x--- 2 pete penguingroup 4096 Oct 19 10:46 Documents
5 drwxr-x--- 4 pete penguingroup 4096 Nov 20 09:30 Downloads
6 drwxr-x--- 2 pete penguingroup 4096 Oct 7 13:13 Music
7 drwxr-x--- 2 pete penguingroup 4096 Sep 21 14:02 Pictures
8 drwxr-x--- 2 pete penguingroup 4096 Jul 27 12:41 Public
9 drwxr-x--- 2 pete penguingroup 4096 Jul 27 12:41 Templates
10 drwxr-x--- 2 pete penguingroup 4096 Jul 27 12:41 Videos
```

Les commandes ont ce qu'on appelle des paramètres (ou arguments ou options, peu importe comment vous les appelez) pour ajouter plus de fonctionnalités. Rappelez comment nous avons ajoutés -a et -l, bien vous pouvez les utiliser ensemble en faisant "-la". L'ordre des paramètres détermine quel ordre suit le résultat, bien des fois c'est pas problématique alors vous pouvez faire "ls -al" et ça fonctionnera aussi.

```
1 $ ls -la
```

5.2. Exercice

Exécutez `ls` avec différents paramètres et regardez le résultat reçu.

5.3. Quiz

Quelle commande devriez vous utiliser pour voir les fichiers cachés?

6. touch

6.1. touch

Voyons comment créer quelques fichiers. Une façon plus simple est d'utiliser la commande "touch". touch vous permet de créer des fichiers vides.

```
1 $ touch monsuperhyperfichier
```

Et boom! un nouveau fichier!

touch est aussi utilisé pour changer les informations de date des fichiers/dossiers existants. Essayez, faites un ls -l sur un fichier et notez la date, puis "touch" ce fichier et il va mettre à jour l'information de date.

Il existe plusieurs autres méthodes pour créer des fichiers qui impliquent d'autres choses comme la redirection et les éditeurs de texte, mais nous en parlerons dans le cours de Traitement de Texte.

6.2. Exercice

1. Créez un nouveau fichier
2. Relevez la date
3. Exécutez la commande touch sur ce même fichier et vérifiez de nouveau la date

6.3. Quiz

Comment créer un fichier avec le nom "monfichier"?

7. file

7.1. file

Dans la leçon précédente, nous avons étudié la commande `touch`, revenons un peu dessus. Avez-vous remarqué que le nom du fichier que nous avons créé ne répond pas à la nomenclature standard que vous avez déjà probablement vu chez les autres systèmes tel que Windows? Normalement, vous créez un fichier appelé `banane.jpeg` et attendez en retour un fichier image JPEG.

Sur Linux, les noms de fichier ne sont pas obligés de représenter le contenu de ces fichiers. Vous pouvez créer un fichier `drole.gif` qui ne contient en fait pas un GIF.

Pour connaître exactement quel genre de fichier est un fichier, vous pouvez utiliser la commande "file". Elle vous présentera la description du contenu du fichier.

```
1 $ file banane.jpg
```

7.2. Exercice

Exécutez la commande `file` sur quelques répertoires et fichiers et regardez les résultats.

7.3. Quiz

Quelle commande pouvez-vous utiliser pour trouver le type de fichier d'un fichier?

8. cat

8.1. cat

On a presque fini avec le parcours des fichiers, mais avant apprenons comment lire le contenu d'un fichier. Une simple commande à utiliser est la commande "cat", abréviation de "concatenate". Il n'affiche pas juste le contenu d'un seul fichier mais il peut aussi afficher celui de plusieurs fichiers à la fois.

```
1 $ cat fichierchien fichieroiseau
```

Il n'est pas adapté pour la lecture de gros fichiers et est seulement destiné pour des petits contenus. Pour la lecture des gros fichiers, il existe beaucoup d'outils que nous aborderons dans la leçon suivante.

8.2. Exercice

Exécutez la commande cat sur différents fichiers et répertoires. Puis essayez sur plusieurs fichiers.

8.3. Quiz

Quel est un bon moyen de lire le contenu d'un fichier?

9. less

9.1. less

Si vous trouvez que les fichiers text sont plus lourds qu'un simple fichier, "less" est beaucoup plus. (Il existe en fait une commande appelée "more" qui fait la même chose). Le texte est affiché par page, ainsi vous pouvez parcourir le texte page par page.

Allez-y et regardez le contenu d'un fichier avec la commande less. Une fois dans la commande less, vous pouvez en fait utiliser des raccourcis clavier pour parcourir le fichier.

```
1 $ less /home/pete/Documents/texte1
```

Utilisez la commande suivante pour en apprendre plus sur less:

- q - Utilisé pour quitter less et revenir dans l'invité de commandes.
- Page up, Page down, Haut et Bas - Parcourir le fichier en utilisant les flèches du clavier.
- g - Se positionne au début du fichier.
- G - Se positionne à la fin du fichier.
- /search - Lancer une recherche dans le document. Précédez les mots que vous voulez rechercher avec "/"
- h - Si vous voulez un peu d'aide sur comment utiliser less pendant que vous êtes dans la commande less, utilisez "h".

9.2. Exercice

Exécutez la commande less sur un fichier, puis parcourez le. Lancez la recherche d'un mot. Allez rapidement au début du fichier ou à la fin.

9.3. Quiz

Comment quitte t-on la commande less?

10. history

10.1. history

Dans votre invité de commande, il y a l'historique des commandes que vous avez précédemment saisies, vous pouvez en fait consulter ces commandes. C'est bien utile quand vous voulez chercher et exécuter une commande précédemment utilisée sans la saisir à nouveau.

```
1 $ history
```

Envie de réexécuter une même commande, appuyez juste sur la flèche haut du clavier.

Envie de réexécuter une commande sans la saisir à nouveau? Taper juste "!!". Si vous avez exécuté la commande "cat fichier1" et voulez la réexécuter, vous pouvez juste faire !! et il exécutera votre dernière commande.

Un autre raccourci de l'historique est ctrl+R, c'est une commande de recherche inversée, si faites ctrl+R et commencez à saisir des termes de la commande que vous désirez, il vous affichera toutes les commandes correspondantes et vous pouvez les voir en faisant les touches ctrl+R. Une fois que vous trouvez votre commande, tapez juste la touche Entrée.

Notre invité de commande devient un peu touffu. Non? Faisons un peu de nettoyage, utilisez la commande "clear" pour effacer le tableau.

```
1 $ clear
```

Là c'est mieux. N'est-ce pas?

Parlant des choses utiles, une des fonctionnalités les plus importantes dans n'importe quel invité de commandes est la complétion à l'aide de la touche de tabulation. Si vous commencez à saisir le début d'une commande, d'un fichier, un répertoire, etc et appuyez la touche de tabulation, il va compléter automatiquement en se basant sur ce qu'il a trouvé dans le répertoire dans lequel vous êtes et si vous n'avez pas d'autres fichiers débutants par les mêmes caractères. Par exemple, si vous essayez d'exécuter la commande chrome, vous pouvez saisir chr et tapez la touche Tab et il complétera chrome.

10.2. Exercice

Consultez l'historique de vos commandes avec les touches Haut et Bas. Amusez-vous avec le recherche inversée ctrl+R.

10.3. Quiz

Quelle est la commande pour nettoyer la console?

11. cp (Copy)

11.1. cp (Copy)

Commençons par faire des copies de ces fichiers. Tout comme le copier/coller dans d'autres systèmes, l'invite de commandes nous offre un moyen plus simple de le faire.

```
1 $ cp monfichiercool /home/pete/Documents/docscool
```

monfichiercool est le fichier à copier et /home/pete/Documents/docscool est la destination.

Vous pouvez copier plusieurs fichiers et dossiers avec l'utilisation des métacaractères. Un métacaractère est un caractère qui peut être substitué par une sélection de motifs, vous donnant plus de flexibilité dans la recherche. Vous pouvez utiliser des métacaractères dans chaque commande pour plus de flexibilité.

- * le métacaractère des métacaractères, il est utilisé pour représenter n'importe quel caractère ou n'importe quelle chaîne de caractères.
- ? utilisé pour représenter un caractère.
- [] utilisé pour représenter n'importe quel caractère dans les crochets.

```
1 $ cp *.jpg /home/pete/Pictures
```

Ceci va copier tous les fichiers d'extension .jpg du répertoire courant vers le dossier Pictures.

Une commande utile est l'utilisation du paramètre -r, il va copier de façon récursive les fichiers et répertoires.

Essayez d'exécuter cp sur un répertoire contenant des fichiers vers le répertoire Documents. Ça ne marche pas. Si? Bien c'est parce que vous devez copier tous les fichiers et répertoires dans le dossier avec le paramètre -r.

```
1 $ cp -r Pumpkin/ /home/pete/Documents
```

Une chose à savoir, si vous copier un fichier vers un répertoire qui possède un fichier avec le même nom, il va écraser ce dernier. C'est désagréable si vous ne le désiriez pas. Vous pouvez utiliser le paramètre -i (interactive) qui demande votre accord avant d'écraser le fichier.

```
1 $ cp -i monfichiercool /home/pete/Pictures
```

11.2. Exercice

Copiez des fichiers, faites attention à ne pas écraser quelque chose d'important. ;)

11.3. Quiz

Quel paramètre dois-tu utiliser pour copier tout un répertoire?

12. mv (Move)

12.1. mv (Move)

Utilisée pour déplacer des fichiers et les renommer aussi. Identique à la commande cp avec les paramètres et fonctionnalités.

Vous pouvez renommer un fichier comme ceci:

```
1 $ mv vieuxfichier nouveaufichier
```

Ou bien vous pouvez déplacer un fichier vers un répertoire différent:

```
1 $ mv fichier2 /home/pete/Documents
```

Et déplacer plus d'un fichier:

```
1 $ mv fichier_1 fichier_2 /unrepertoire
```

Vous pouvez aussi renommer les dossiers:

```
1 $ mv repertoire1 repertoire2
```

Comme la commande cp, si vous déplacez un fichier ou répertoire, il va écraser le fichier/répertoire ayant le même nom. Donc vous pouvez utiliser le paramètre -i qui demande votre accord avant d'écraser.

```
1 mv -i repertoire1 repertoire2
```

Disons que vous avez voulu déplacer un fichier en écrasant l'existant. Vous pouvez faire une récupération du fichier écrasé et il sera nommé avec un ~.

```
1 $ mv -b repertoire1 repertoire2
```

12.2. Exercice

Renommez un fichier, puis déplacez dans un répertoire différent.

12.3. Quiz

Comment renommer le fichier "cat" en "dog"?

13. mkdir (Make Directory)

13.1. mkdir (Make Directory)

Nous allons avoir besoin des répertoires pour stocker les fichiers sur lesquels nous avons travaillé. La commande "mkdir" (Make Directory - Créer un dossier) va nous aider, il va créer un répertoire s'il n'existe pas encore. Vous pouvez également créer plusieurs dossiers simultanément.

```
1 $ mkdir livres paintings
```

Vous pouvez aussi créer des sous-répertoires directement en ajoutant le paramètre -p (parent).

```
1 $ mkdir -p livres/hemingway/favorites
```

13.2. Exercice

Créez des répertoires et déplacez quelques fichiers à l'intérieur.

13.3. Quiz

Quelle commande utilise t-on pour créer un répertoire?

14. rm (Remove)

14.1. rm (Remove)

Maintenant, je crois que nous avons beaucoup de fichiers, supprimons en quelques-uns. La commande 'rm' (remove - supprimer) est utilisé pour supprimer les fichiers et répertoires.

```
1 $ rm fichier1
```

Faites attention en utilisant la commande rm, il y a pas de corbeille magique où vous pouvez récupérer les fichiers supprimés. Une fois supprimés, ils sont supprimés pour de bon, donc attention.

Heureusement il existe des mesures de sécurité mises en place, donc un utilisateur normal ne peut pas supprimer un ensemble de fichiers importants. Une confirmation vous sera demandée avant la suppression. Si le répertoire est protégé en écriture, il sera aussi difficile à supprimer.

Maintenant si vous n'avez que faire de ça, vous pouvez bien sûr supprimer un ensemble de fichiers.

```
1 $ rm -f fichier1
```

L'option -f ou force dit à rm de supprimer tous les fichiers, qu'ils soient protégés en écriture ou pas, sans demande de confirmation (tant que vous avez les droits d'accès nécessaires).

```
1 $ rm -i fichier
```

L'ajout du paramètre -i vous demandera confirmation avant la suppression des fichiers ou répertoires.

```
1 $ rm -r repertoire
```

Normalement, vous ne pouvez pas supprimer un répertoire avec juste rm, vous devez ajouter le paramètre -r (récursif) pour aussi supprimer les fichiers et sous-répertoires qu'il pourrait contenir.

Vous pouvez supprimer un répertoire avec la commande rmdir

```
1 $ rmdir repertoire
```

14.2. Exercice

1. Créez un fichier appelé "-fichier" (n'oubliez pas le tiret)
2. Supprimez ce fichier.

14.3. Quiz

Comment supprimer un fichier appelé monfichier?

15. find

15.1. find

Avec tous ces fichiers que nous avons dans notre machine, il peut devenir fiévreux de trouver un fichier spécifique. Bien, il existe une commande pour nous aider, 'find' !

```
1 $ find /home -name puppies.jpg
```

Avec find, vous devez préciser le dossier dans lequel vous allez faire la recherche et ce que vous voulez rechercher. Dans notre cas, on veut trouver un fichier à travers son nom puppies.jpg.

Vous pouvez préciser le type de fichier vous chercher.

```
1 $ find /home -type d -name MonDossier
```

Vous pouvez voir que j'ai mis le type de fichier à chercher comme un dossier avec (d) et je cherche aussi par le nom MonDossier.

Une chose géniale à savoir est que find ne s'arrête pas dans le répertoire où vous vous trouvez, il cherche aussi dans les sous-répertoires.

15.2. Exercice

1. Cherchez un fichier dans le répertoire racine (root) qui contient le terme "net".

15.3. Quiz

Quelle option devrais-je spécifier si je veux effectuer une recherche par nom?

16. help

16.1. help

Linux a quelques supers outils intégrés pour vous aider à utiliser une commande ou vérifier quels paramètres sont disponibles pour une commande. Un outil, 'help', est un composant de la commande bash qui fournit l'aide des autres commandes bash (echo, logout, pwd, etc).

```
1 $ help echo
```

Ceci vous donnera une description et les options/paramètres utilisables quand vous voulez exécuter echo. Pour d'autres programmes exécutables, la convention est d'avoir une option appelée --help ou quelque de similaire.

```
1 $ echo --help
```

Tous les développeurs qui distribuent des programmes ne se conforment à ce standard, mais c'est probablement votre meilleur moyen pour trouver de l'aide sur un programme.

16.2. Exercice

Exécutez help sur les commandes echo, logout et pwd.

16.3. Quiz

Comment obtenez-vous une aide rapide sur une ligne de commande bash?

17. man

17.1. man

Eh bien, j'espère que certains programmes ont un manuel tel que nous pouvons avoir plus d'informations à propos d'eux. Heureusement, ils en ont! Appelées pages de manuel, vous pouvez voir le manuel d'une commande avec la commande man.

```
1 $ man ls
```

Les pages de manuel sont des manuels par défaut présents dans la plupart des systèmes d'exploitation Linux. Elles fournissent la documentation des commandes et d'autres aspects du système.

Essayez la sur quelques commandes pour avoir plus d'infos sur elles.

17.2. Exercice

Exécutez la commande man sur la commande ls.

17.3. Quiz

Comment voit t-on le manuel d'une commande?

18. whatis

18.1. whatis

Ouf, nous avons appris bien des commandes jusque là, si vous avez des doutes sur le fonctionnement d'une commande, vous pouvez utiliser la commande `whatis`. La commande `whatis` fournit une description brève de la ligne de commande.

```
1 $ whatis cat
```

La description provient du manuel de chaque commande. Si vous exécutez "`whatis cat`", vous verriez qu'il y a un petit texte de présentation avec une brève description.

18.2. Exercice

Exécutez la commande `whatis` sur la commande `less`.

18.3. Quiz

Quelle commande peut-on utiliser pour voir une brève description d'une commande?

19. alias

19.1. alias

Des fois taper des commandes peut devenir répétitif, or si vous devez taper une longue commande à plusieurs reprises, il est mieux d'avoir un alias que vous pouvez utiliser pour. Pour créer un alias pour une commande, vous donnez simplement le nom de l'alias et y affectez la commande.

```
1 $ alias foobar='ls -la'
```

Ainsi, au lieu de taper ls -la, vous pouvez taper foobar et il exécutera la commande, un outil super. Sachez que votre alias n'existera plus après un redémarrage. Vous aurez besoin d'ajouter un alias permanent dans le fichier:

```
1 ~/.bashrc
```

ou fichiers similaires si vous voulez qu'il existe même après un redémarrage.

Vous pouvez supprimer les alias avec la commande unalias:

```
1 $ unalias foobar
```

19.2. Exercice

Créez quelques alias puis supprimez-les.

19.3. Quiz

Quelle commande utilise-t-on peut créer un alias?

20. exit

20.1. exit

Excellent, vous avez sûrement bien fait de commencer par les bases. Nous avons juste survolé les choses, maintenant que vous avez appris à ramper, dans les prochains cours, je vais vous apprendre à marcher.

Pour le moment, vous pouvez vous féliciter et prendre une pause. Pour sortir de la console, vous pouvez utiliser la commande exit.

```
1 $ exit
```

Ou la commande logout:

```
1 $ logout
```

Ou si vous travaillez à partir de l'interface graphique de la console, vous pouvez juste la fermer. Rendez-vous dans le prochain cours!

20.2. Exercice

Exécutez la commande exit dans l'invité de commande et regardez ce qu'il se passe. Rassurez-vous que vous n'avez plus rien à faire dans l'invité de commande.

20.3. Quiz

Comment peut-on sortir de l'invité de commande?