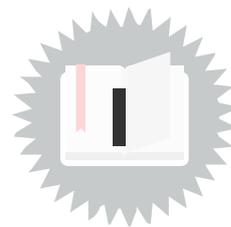


Prendre en main le terminal sous Linux

Table des matières

| | |
|---|-----------|
| I - Contexte | 3 |
| II - Qu'est-ce que la console ? | 4 |
| III - Exercice : Appliquer la notion | 6 |
| IV - La console sous Linux | 7 |
| V - Exercice : Appliquer la notion | 9 |
| VI - Premières commandes en console | 10 |
| VII - Exercice : Appliquer la notion | 12 |
| VIII - Auto-évaluation | 13 |
| 1. Exercice final | 13 |
| 2. Exercice : Défi | 14 |
| Conclusion | 16 |

Contexte



Durée : 1h

Environnement de travail : Environnement Linux en console (ordinateur ou serveur en SSH)

Pré-requis : Aucun

Lorsque l'on commence à utiliser Linux et son interface graphique, on se rend compte qu'il y a beaucoup de similitudes avec ce que l'on retrouve sur d'autres OS. Cependant il y a un élément que l'on ne retrouve pas forcément et qui s'avère particulièrement puissant : la console (ou terminal). Le console est cette interface textuelle que l'on voit souvent dans les films, qui donne l'impression d'être complexe, mais qui est en fait pas si dure à prendre en main et à laquelle on s'habitue très vite. Nous allons, dans ce cours, décrire rapidement pourquoi la console existe, à quoi elle sert, puis nous verrons comment elle fonctionne.



Qu'est-ce que la console ?

Objectifs

- Comprendre ce qu'est la console
- Comprendre pourquoi on a inventé la console
- Comprendre l'intérêt de la console aujourd'hui

La console



La console est une interface humain-machine textuelle qui permet d'écrire des commandes à exécuter sur la machine et de visualiser le résultat de ces commandes.

```
kyane@europa:~$ ls
Bureau  code  Documents  Images  Logiciels  mobile  Musique  Nextcloud  obs-records  Téléchargements  Vidéos
kyane@europa:~$ cat /etc/
Display all 332 possibilities? (y or n)
kyane@europa:~$ cat /etc/ss
ssh/ ssh/
kyane@europa:~$ cat /etc/ssh/
moduli          ssh_config.d/          sshd_config.d/          ssh_host_dsa_key.pub   ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub
ssh_config      sshd_config            ssh_host_dsa_key        ssh_host_ecdsa_key    ssh_host_ed25519_key    ssh_host_rsa_key
kyane@europa:~$ cat /etc/ssh/ssh_config
# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.
#
# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.
#
# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.
#
Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
# Port 22
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
```

Une console

Historique



Aux débuts de l'informatique, les premiers ordinateurs étaient beaucoup moins puissants que ceux que l'on utilise aujourd'hui. Il était nécessaire de pouvoir piloter ces ordinateurs à l'aide d'une interface simple et peu gourmande en ressources, utiliser une interface graphique avec des millions de pixels de couleurs et un pointeur de souris n'était pas imaginable à l'époque.

C'est pourquoi la console a été inventée : une interface entièrement textuelle et en noir et blanc.

Utilité de la console



On pourrait penser que, maintenant que les ordinateurs savent largement faire tourner une interface graphique, la console ne sert plus à rien.

C'est en réalité tout le contraire, la console s'avère utile dans de très nombreuses situations :

- pour administrer un serveur à distance, qui n'a pas d'écran branché à lui
- pour réaliser des tâches d'administration, par exemple qui relèvent du système d'exploitation, que n'ont pas une interface graphique dédiée
- pour réaliser des opérations courantes qui sont parfois plus simple à faire qu'en utilisant l'interface graphique.

Pour ce dernier point, il faut savoir que la console permet d'utiliser plusieurs centaines ou milliers de commandes différentes et de les combiner entre elles. Les possibilités sont donc très importantes, et il est impossible de permettre de réaliser toutes les opérations avec une interface graphique aboutie.

La console, plus rapide que l'interface graphique



Prenons l'exemple d'une opération classique qui est beaucoup plus simple à réaliser en console qu'à l'aide d'une interface graphique : compter le nombre d'image JPG qui se trouve dans un dossier, avec d'autres fichiers.

Avec l'interface graphique il faudra ouvrir son explorateur de fichier, se rendre dans le dossier en question, compter manuellement tout les fichiers en JPG (éventuellement en les triant par types), vérifier qu'on a pas fait une erreur, etc. C'est long et fastidieux.

Dans la console, ça se fait en une ligne de texte :

```
1 $ ls -l | grep -i ".jpg" | wc -l
2 386
```

Bien entendu pour le moment cela peut sembler incompréhensible tant que l'on ne connaît pas les commandes en question, mais ce sont ici en réalité des commandes très basiques (la première qui liste les fichiers, la seconde qui filtre les JPG, la troisième qui compte), que vous connaîtrez bientôt par cœur.

À retenir

La console existe historiquement car les premiers ordinateurs ne pouvaient pas avoir d'interface graphique complexe. Cette interface humain-machine s'est avérée tellement performante que l'on continue de l'utiliser au quotidien aujourd'hui. Ce n'est pas un outil réservé à une élite ou aux personnes qui "connaissent l'informatique", et elle s'avère très pratique pour réaliser certaines des tâches classiques sur un ordinateur.

Exercice : Appliquer la notion



Question

En cherchant un peu sur Wikipédia, de quelle année date le premier ordinateur ayant une interface graphique ?

Indice :

La page sur les interfaces graphiques¹ devrait vous aider.

¹. https://fr.wikipedia.org/wiki/Interface_graphique

La console sous Linux



Objectifs

- Savoir ouvrir une console sur sa machine
- Savoir lire un prompt

Ouvrir un terminal

Sur Ubuntu, on utilise l'application "Terminal" pour ouvrir une console (le terme `terminal` est équivalent à `console`). Une fenêtre s'ouvre, nous sommes dans la console.

```
Terminal - kyane@europa: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
kyane@europa:~$ █
```

Comme prévu, c'est minimaliste : du texte, un rectangle indiquant que l'on peut écrire au clavier, et rien de plus.

Le prompt



La première chose que l'on constate est ce qui s'appelle le prompt. C'est le texte qui se trouve à gauche de la zone pour entrer la commande, et qui indique le contexte dans lequel on se trouve.

Le prompt se compose de 3 parties principales :

- le nom d'utilisateur
- le nom de la machine
- le dossier dans lequel on se trouve

Le nom d'utilisateur et le nom de la machine sont séparés par le caractère @, le caractère : permet de séparer le nom de la machine et le nom du dossier.



Dans notre cas le prompt est :

```
1 kyane@europa:~$
```

Ce qui signifie donc que :

- l'utilisateur est kyane
- la machine se nomme europa
- le dossier dans lequel on se trouve est ~ (c'est un raccourci pour dire que l'on est dans le répertoire de l'utilisateur)

Le symbole \$ à la fin signifie simplement que l'utilisateur est "normal", par opposition à un super-utilisateur (ou administrateur) symbolisé par le caractère #.

Une autre manière d'accéder à la console



En réalité, ouvrir l'application "Terminal" dans notre environnement graphique n'est pas la manière originelle d'utiliser la console. En effet, même si nous avons une interface textuelle, cela reste un outil qui fonctionne en mode graphique.

Linux (et en fait la plupart des OS basés sur Unix) propose aussi d'accéder à une "vraie" console, dans le sens où c'est uniquement une interface texte, en dehors de l'interface graphique. Pour cela il faut utiliser les touches CTRL + Alt + F1 (on peut utiliser aussi F2, F3, etc jusqu'à F6), qui bascule notre écran de l'interface graphique à une console. Pour revenir en arrière, on fait CTRL + Alt + F7.

Dans la pratique on utilise pas cette console sur un ordinateur de bureau, cela s'avère pratique pour des serveurs par exemple (qui n'ont souvent pas de logiciels pour gérer une interface graphique).

À retenir

Sur Ubuntu, on peut ouvrir très simplement une console avec l'application "Terminal". On se retrouve ainsi avec une interface textuelle dans la laquelle on pourra entrer des commandes. Mais avant cela, il est important de savoir déchiffrer le prompt, ce petit bout de texte dans le terminal qui nous permet de connaître le contexte dans lequel on se trouve.

Exercice : Appliquer la notion



Après avoir ouvert un terminal, on se retrouve avec le prompt suivant.

```
1 sarah@neptune.picasoft.net:/etc/ssh$
```

Question 1

Quel est le nom de la machine sur laquelle on se trouve ?

Question 2

Dans quel dossier se trouve notre console ?

Question 3

Quel est le compte utilisant l'ordinateur actuellement ?

Premières commandes en console



Objectifs

- Savoir utiliser une première commande sous Linux
- Savoir obtenir des informations sur une commande

Une première commande sous Linux

Maintenant que l'on sait ouvrir une console et décoder un prompt, on peut enfin utiliser la console. Pour cela entrons notre toute première commande :

```
1 kyane@europa:~$ ls
2 Documents Musique Images firefox.config
```

Cette commande `ls` a une fonction toute simple : lister les fichiers et dossiers du répertoire. On voit ici 4 résultats : Documents, Musique, Images et `firefox.config`

Les paramètres

À ce stade on ne sait pas si les résultats retournés sont les noms des fichiers ou ceux des dossiers. En effet la commande `ls` se contente de lister ce qui se trouve dans le dossier courant. Si l'on veut clairement identifier les dossiers des fichiers simples, on peut faire :

```
1 kyane@europa:~$ ls -p
2 Documents/ Musique/ Images/ firefox.config
```

L'ajout de `-p` ici permet d'ajouter un caractère `/` à la fin des noms de dossiers (ici donc seul `firefox.config` est un fichier). C'est ce que l'on appelle **un paramètre**. En effet la plupart des commandes acceptent des paramètres (on par aussi d'options) qui permettent de modifier légèrement le comportement de la commande. Ces paramètres sont généralement des lettres préfixées par un tiret (`-p`) ou parfois des mots entiers préfixés par deux tirets (`--ignore`).

Lire le manuel

Nous avons dit qu'il y avait plusieurs centaines, voire milliers, de commandes, et maintenant on apprend que les commandes peuvent avoir différents paramètres. À ce stade on pourrait se dire « *ce n'est pas pour moi, je ne pourrais jamais retenir tout ça par cœur* », mais ce sera une erreur. Personne ne connaît toutes les commandes qui existe, la plupart du temps on se sert de 20 ou 30 commandes différentes quotidiennement (cela devient donc une habitude) et de la même manière on apprend vite les paramètres les plus importants.

De plus, une pratique qui est très courante lorsque l'on se met à utiliser Linux est... de lire le manuel ! En effet il existe une commande qui permet d'obtenir le manuel d'utilisation des autres commandes : `man`.

```
1 kyane@europa:~$ man ls
```

```

ls(1)                                     User Commands                               ls(1)
NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphical characters

  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

  -B, --ignore-backups
      do not list implied entries ending with ~

  -c
      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

  -C
      list entries by columns

  --color[=WHEN]
      colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

  -d, --directory
      list directories themselves, not their contents

  -D, --dired
      generate output designed for Emacs' dired mode

  -f
      do not sort, enable -al, disable -ls --color

  -F, --classify
      append indicator (one of */->@) to entries

  --file-type
      likewise, except do not append '*'

  --format=FORMAT
      across -x, commas -m, horizontal -x, long -l, single-column -l, verbose -l, vertical -C
  Manual page ls(1) line 1 (press h for help or q to quit)

```

On peut ainsi naviguer dans le manuel de la commande `ls` à l'aide des touches du clavier, et sortir de celui-ci avec la touche `Q`

? Exemple

En descendant un peu dans le manuel, on peut voir les informations sur le paramètre que l'on a utilisé.

- 1 `-p, --indicator-style=slash`
- 2 `append / indicator to directories`

À retenir

On sait désormais utiliser une première commande dans la console, même si c'est encore rudimentaire. On a aussi compris comment consulter les informations sur les paramètres d'une commande avec `man`.

Exercice : Appliquer la notion



On va entrer dans le terminal la commande suivante.

```
1 cat /etc/hostname
```

Question 1

D'après le manuel, à quoi sert la commande `cat` ?

Question 2

Que retourne comme résultat la commande complète ci-dessus ?

Auto-évaluation



1. Exercice final

Exercice 1 : Quiz - intro console

Exercice

La console est aujourd'hui complètement dépassée par rapport aux interfaces graphiques.

- Vrai
- Faux

Exercice

Quelle sont les intérêts de la console par rapport à l'interface graphique ?

- Elle nécessite moins de ressources
- Elle propose une meilleure accessibilité
- Elle permet de réaliser des actions, sur le système d'exploitation, qui ne sont pas disponibles dans l'interface graphique
- Elle permet de faire certaines actions plus rapidement

Exercice 4 : Quiz - Console Linux

Exercice

Dans le terminal, qu'est-ce que le prompt ?

- L'ensemble des commandes qu'il est possible d'utiliser
- Le texte en début de ligne indiquant le contexte dans lequel on se trouve
- L'instruction qui permet de sortir du terminal

Exercice

Sous Ubuntu, pour accéder à la console on peut

- Utiliser l'application "Terminal" installée par défaut.
- Choisir le mode console au démarrage de l'ordinateur (écran BIOS)
- Utiliser les raccourcis CTRL + ALT + F1 à F6

Exercice

Quelles sont les informations que l'on trouve par défaut dans le prompt ?

- Le nom d'utilisateur
- L'heure actuelle

- Le nom de la machine
- Le dossier dans lequel on se trouve
- Le temps d'exécution de la précédente commande

Exercice

Quel est l'utilisateur dans le prompt suivant : `samy@fred : / jamy$`

- samy
- fred
- jamy

Exercice 9 : Quiz - Commandes

Exercice

Pour obtenir de l'aide sur une commande, j'utilise la commande

- help
- manual
- man
- info

Exercice

En général, comment est-ce que l'on ajoute un paramètre à une commande ?

- À l'aide d'une lettre précédée d'un tiret (ex. `COMMANDE -p`)
- À l'aide de un ou plusieurs mots précédés d'un double tiret (ex. `COMMANDE --preserve`)
- À l'aide de mots écrits avant la commande dans le terminal (ex. `preserve COMMANDE`)

Exercice

D'après le manuel que fait l'option `-R` de la commande `ls` ?

- Elle ignore les fichiers qui commencent par la lettre R
- Elle indique la taille utilisée par les fichiers sur le disque.
- Elle liste aussi les fichiers présents dans les dossiers, ainsi que les sous-dossiers, etc. de manière récursive.

2. Exercice : Défi

On souhaite afficher dans notre terminal des informations sur la version de Ubuntu que l'on utilise.

Question 1

En cherchant sur le Web, quelle est la commande qui permet d'afficher les informations spécifiques à notre distribution Linux ?

Question 2

Lorsque l'on fait uniquement `lsb_release` on ne voit en réalité aucune informations. Comment faire ?

```
1 $ lsb_release
2 No LSB modules are available.
```

Indice :

Il faut chercher dans les paramètres de la commande.

Question 3

On ne veut maintenant afficher que le numéro de la version, c'est à dire afficher uniquement `20.04`, comment est-ce possible ?

Indice :

La commande `lsb_release` a un paramètre pour cela.

Conclusion



Dans ce cours nous avons découvert ce qu'est la console sous Linux, et pourquoi elle est très utilisée. Pas à pas nous avons ouvert notre premier terminal et entré notre première commande. C'est un outil particulièrement puissant, qui semble parfois compliqué, mais qui s'avère plus pratique que l'interface graphique pour de nombreuses opérations. De plus il est indispensable de savoir utiliser cette interface lorsque l'on veut comprendre un peu mieux comment fonctionne son ordinateur.