

Mise à niveau L3: Linux

Stefan Schwoon

DER Informatique, ENS Paris-Saclay

5 septembre 2023

Linux et tout ça

Linux: système d'exploitation

- ▶ Unix: système d'exploitation développé à partir de 1969, très utilisé dans le milieu académique
- ▶ Linux: développé par *Linus Torvalds* à partir de 1991, inspiré d'Unix mais comme logiciel libre

POSIX: standard pour uniformiser des interfaces d'Unix et autres

- ▶ Linux est largement conforme à Posix
- ▶ *macOS* l'est entièrement
- ▶ *Windows* n'implèmente que quelques chapîtres

GNU: collection de logiciels

- ▶ projet de *Richard Stallman* à partir de 1983
- ▶ développement de bibliothèques et logiciels libres, habituellement utilisés avec Linux

Objectifs de la mise à niveau

Vous rendre opérationnel pour les tâches à venir (projets de programmation, rédaction de documents, tâches de routine, ...)

Aujourd'hui :

- ▶ Système de fichiers : organiser les données persistentes
- ▶ Processus : organisation interne des tâches et de la mémoire
- ▶ Shell / terminal : utilitaire pour interagir avec les deux

Système de fichiers

Fichier

Collection de données associé à un nom
(texte, exécutable, fichier Excel, image, base de données, ...)

Système de fichiers

Organisation hiérarchique de fichiers, consiste d'un arborescence de *dossiers* qui eux contiennent des fichiers ou d'autres dossiers

- ▶ stocké sous forme pérenne sur disque dur/clé USB/etc
- ▶ *ouvrir* un fichier : en charger une copie dans la mémoire (non pérenne) de l'ordinateur
- ▶ *sauvegarder* un fichier : en écrire une copie dans le système de fichiers (pérenne)

Organisation d'un système de fichiers

Contenu

- ▶ fichiers essentiels pour le système d'exploitation
- ▶ applications
- ▶ données personnelles de toutes les utilisateurs

gérés par des *droits d'accès*

- ▶ système d'exploitation / applications : modifiables par un utilisateur admin (*root*)
- ▶ tout utilisateur possède un *dossier/répertoire personnel* (home directory en anglais), il en règle l'accessibilité aux autres utilisateurs
- ▶ dans la salle de machine : vos dossiers personnels sont stockés sur un serveur, automatiquement disponibles où que vous soyez connectés

Chemins

Chemin absolu

emplacement d'un fichier dans le système, construit par les noms des dossiers hiérarchiques depuis la *racine*, séparés par des slash (/)

- ▶ /home/schwoon (dossier personnel)
- ▶ /home/schwoon/text1.txt (fichier dans dossier personnel)
- ▶ /bin/bash (fichier exécutable, un shell)

Chemin relatif

emplacement d'un fichier depuis un dossier autre que la racine, où .. note le dossier hiérarchiquement supérieur

- ▶ ../ailleurs (dessine /home/ailleurs si interprété depuis /home/schwoon)

Note : un seul point (.) note le dossier actuel

Shell (première excursion)

Shell vs terminal

- ▶ Shell: programme pour gérer de façon plus fine vos fichiers et processus, accepte des commandes sous forme de texte (nom : bash / sh ou similaire)
- ▶ Terminal : programme (graphique ou autre) qui gère les entrées/sorties du shell et les processus lancés depuis celui-ci

Naviguer la hiérarchie des dossiers

- ▶ *pwd*: afficher le dossier actuel
- ▶ *cd*: changer de dossier actuel
- ▶ *mkdir*: créer un nouveau dossier
- ▶ *rmdir*: supprimer un dossier vide
- ▶ *ls*: afficher le contenu d'un dossier (options utiles: -a, -l)

Gestion simple de fichiers

Quelques commandes utiles

- ▶ *cat*: afficher contenu d'un fichier
- ▶ *less*: paginer un fichier long
- ▶ *rm*: supprimer un fichier
- ▶ *cp*: copier un fichier dans un autre
- ▶ *mv*: renommer ou déplacer un dossier ou fichier
- ▶ *touch*: création d'un fichier vide

Créer un fichier

- ▶ avec un éditeur texte (p.ex. *nano*) ou autre application
- ▶ simple création : *echo bonjour > fichier*

Astuce

Afficher la syntaxe et toutes les options d'une commande avec *man*

- ▶ *man ls*: afficher le manuel de *ls*

Droits d'accès

`ls -l` affiche tout fichier et dossier avec ses droits d'accès, son propriétaire et son groupe

- ▶ read (r), write (w), execute (x)
- ▶ trois groupes: utilisateur (u), groupe (g), autres (o)

Note: pour avoir accès à un dossier, il faut r+x !

Quelques commandes utiles

- ▶ `whoami, id`: afficher son identité
- ▶ `chmod`: modifier les droits d'accès

Exemples:

- ▶ `chmod +x fichier`: ajouter x pour tout le monde
- ▶ `chmod u-r fichier`: retirer droit de lecture à l'utilisateur

Excursion : les shell script

Shell script simple

```
#!/bin/bash
while true
do
    echo bla
    sleep 1
done
```

- ▶ Sauvegarder ce texte dans un fichier nommé, disons, *bla*.
- ▶ Le rendre executable : `chmod +x bla`
- ▶ En exécuter une copie : `./bla`
- ▶ pour annuler, taper Ctrl+C

Gérer les processus

Quelques commandes utiles

- ▶ *ps* : afficher liste de processus (option utile: *a*)
- ▶ *top* : afficher liste dynamique de processus
- ▶ *kill* : envoyer signal (typiquement pour annuler)
- ▶ *killall* : annuler tous les processus avec un certain nom

Signaux

- ▶ *kill -l* : liste de signaux
- ▶ *kill -TERM* : demander gentiment à un processus de se terminer
- ▶ *kill -KILL* : annuler un processus toute de suite

Redirection et enchaînement dans le shell

Redirection :

- ▶ `cmd > fichier` : copier résultat de `cmd` dans un fichier
- ▶ `cmd >> fichier` : ajouter résultat de `cmd` à un fichier
- ▶ `cmd < fichier` : fournir les entrées de `cmd` depuis un fichier

Plusieurs façons :

- ▶ `cmd1 ; cmd2` : enchaîner
- ▶ `cmd1 || cmd2` : faire `cmd2` seulement si `cmd1` a échoué
- ▶ `cmd1 && cmd2` : faire `cmd2` seulement si `cmd1` a réussi
- ▶ `cmd1 | cmd2` : passer le résultat de `cmd1` à `cmd2`

Les outils de texte

Liste non-exhaustive :

- ▶ *wc* : compter le nombre de lignes
- ▶ *grep* : chercher un motif
- ▶ *sort* : trier alphabétiquement (ou numériquement: -n)
- ▶ *uniq* : supprimer les doublons (option utile: -c)
- ▶ *cut* : sélectionner une partie de chaque ligne
- ▶ *tr* : échange de caractères
- ▶ *sed* : remplacements de texte (expressions régulières)
- ▶ *tac* : inverser un fichier
- ▶ *seq* : séquence numérique
- ▶ *nl* : numéroter des lignes
- ▶ *diff* : comparer deux fichiers