

08/06/2022

Version 2



UNIX

JÉRÉMY PERROUAULT

A decorative wavy line in light blue and white, flowing vertically along the left edge of the slide.

HISTOIRE

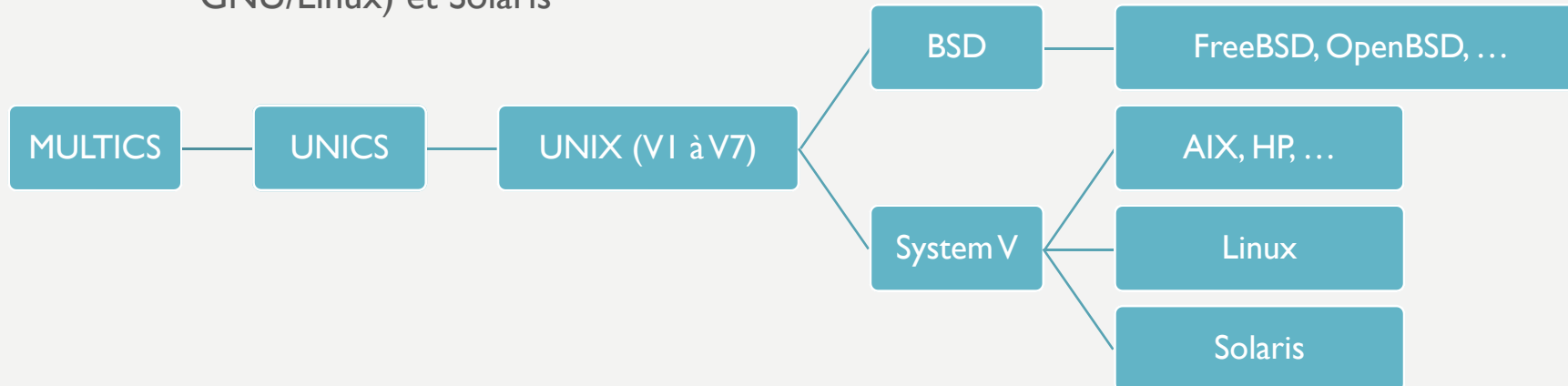
HISTORIQUE ET CONTEXTE

UN PEU D'HISTOIRE

1964	Projet <i>MULTICS</i> (MULTiplexed Information and Computing Service) Développé par un consortium : MIT, General Electric, Laboratoires Bell et AT-T
1969	Projet <i>UNICS</i> (UNiplexed Information and Computing Service) Développé en Assembleur par Ken Thomson et les Laboratoires Bell
1971	Publication du manuel « The UNIX Programmer »
1973	Réécriture de UNIX en C
1979	Reprise du projet par le monde académique (Université de Californie à Berkeley)

UN PEU D'HISTOIRE

- Après 7 versions de UNIX (elles-mêmes après MULTICS et UNICS)
 - UNIX BSD en 1977
 - Qui pose les bases des systèmes tels que FreeBSD, NetBSD, OpenBSD et MacOSX
 - UNIX System V en 1983
 - Qui fournit le socle aux générations futures telles que AIX, HP-UX, IRIX, UnixWare
 - Mais est également précurseur de deux autres branches, très connues par le grand public : Linux (ou GNU/Linux) et Solaris



UN PEU D'HISTOIRE

- Les systèmes UNIX étaient payants et devenaient de plus en plus cher
 - Il fallait créer une alternative : le projet **GNU** (GNU's Not UNIX) est né
 - Cette solution est gratuite et libre
 - C'est une « copie » de UNIX
- Linus Torvald, étudiant Finlandais, crée dans le même temps son système d'exploitation libre, qui prend le nom de Linux
- Les deux projets sont complémentaires, et ont fusionné pour donner GNU/Linux
 - GNU (programmes libres : copie des fichiers, suppression des fichiers, éditeur de texte, ...)
 - Linux (noyau d'OS)

UN PEU D'HISTOIRE

- Quant à GNU/Linux, ce système donnera naissance à de multiples distributions
 - Slackware
 - Mandriva
 - Red Hat
 - SuSE
 - Debian
 - Ubuntu
 - Knoppix
 - ...

A decorative wavy line in light blue and white, flowing from the top left towards the bottom left of the slide.

FONDAMENTAUX

LA STRUCTURE UNIX

LES COMMANDES UNIX

FONDAMENTAUX

- UNIX se distingue par les caractéristiques suivantes
 - Multi-utilisateurs (plusieurs utilisateurs peuvent utiliser, simultanément, le système)
 - Multi-tâches (un utilisateur peut exécuter plusieurs programmes en même temps)
 - Repose sur un noyau (kernel) qui utilise 4 concepts
 - Fichiers
 - Droits d'accès
 - Processus
 - Communication interprocessus (IPC)

FONDAMENTAUX

- Arborescence à parcourir à partir du chemin racine, appelé **root**
- La racine est accessible avec le chemin « / »

Chemin absolu	Utilité
/bin	Exécutables essentiels au système, utilisable par un utilisateur
/boot	Fichiers permettant au système de démarrer
/dev	Points d'entrée des périphériques (dev comme device)
/etc	Fichiers et exécutables liés à l'administration du système (utilisateurs, groupes, réseau, ...)
/home	Répertoire personnel des utilisateurs
/lib	Bibliothèques partagées essentiels au système au démarrage
/mnt (ou /media)	Points de montage des partitions temporaires (CD, Clé USB, lecteur réseau, ...)

FONDAMENTAUX

Chemin absolu	Utilité
/opt	Packages d'application supplémentaires
/proc	Fichiers mémoire, E/S, périphériques, compatibilité avec le noyau
/root	Répertoire de l'administrateur principal root
/usr	Exécutables ou autres fichiers liés aux utilisateurs
/var	Données variables, on y trouve notamment les fichiers journaux (logs)
/tmp	Données temporaires

FONDAMENTAUX

- Dans un système UNIX, tout est fichier
 - Un périphérique est considéré comme tel
 - Son point de montage (ou point d'accès) est mappé sur un répertoire
 - Un répertoire est un fichier signé « d » (comme directory)
 - Les fichiers classiques ne sont pas signés (signés « - »)

FONDAMENTAUX

- Lorsque qu'un périphérique est inséré (clé USB par exemple)
 - Son point d'accès se trouvera dans « /dev »
 - Puisqu'il s'agit d'un disque SCSI, son nom commencera par « sd » pour « **SCSI Disk** »
 - Chaque disque a sa lettre : a, b, c, d, ...
 - Chaque partition de chaque disque a un numéro : 1, 2, 3, ...
 - Donc
 - La première partition du premier disque SCSI inséré pointera sur /dev/sda1
 - La troisième partition du deuxième disque pointera sur /dev/sdb3
 - Il faudra ensuite « monter » la partition souhaitée dans un répertoire
 - Souvent dans « /mnt » ou dans « /media »
 - /dev/sda1 → /media/projets

FONDAMENTAUX

- Pour communiquer avec le système
 - Utilisation d'une interface graphique (type bureau)
 - Utilisation d'un interpréteur de commandes Shell
 - Interface entre l'utilisateur et le système d'exploitation
 - Application chargée d'interpréter les commandes des utilisateurs et de les transmettre au système
 - Plusieurs types de shell existent
 - bsh ou sh Bourne shell, développé par Stephen Bourne
 - bash Bourne again shell (version améliorée de sh)
 - csh C shell (évolution de sh, syntaxe proche du langage C)
 - ...
 - Le nom du shell correspond, en général, au nom de son fichier exécutable
 - /bin/sh
 - /bin/bash

FONDAMENTAUX

- En bash, l'invite de commande se présente comme suit
 - login@machine\$ ou login@machine#
 - \$ si utilisateur normal
 - # si super-utilisateur (administrateur)
- Quelques raccourcis clavier
 - Ctrl+C Terminer le programme en cours
 - Ctrl+Z Mettre en attente le programme en cours
 - Ctrl+D Terminer l'entrée standard
 - Tabulation Compléter le nom d'un répertoire, d'un fichier ou d'une commande existante

FONDAMENTAUX

- Lorsqu'une commande est exécutée, un processus est créé, qui ouvre trois flux
 - stdin Entrée standard (clavier par défaut), identifié par 0
 - stdout Sortie standard (écran par défaut), identifié par 1
 - stderr Sortir d'erreur standard (écran par défaut), identifié par 2
- Il est possible de rediriger les flux
 - > Redirection de la sortie standard vers un fichier
 - < Redirection de l'entrée standard
 - >> Redirection de la sortie standard vers un fichier, avec concaténation
 - >& Redirection de la sortie standard et de la sortie d'erreur standard
 - >! Redirection avec écrasement de fichier
 - | Redirection de la sortie standard vers l'entrée standard (*pipe* ou *tube*)

FONDAMENTAUX

- Affiche "Bonjour"

```
echo "Bonjour"
```

- Ecrit "Bonjour" dans le fichier *coucou.txt* (et remplace son contenu)

```
echo "Bonjour" > coucou.txt
```

- Ajoute "Bonjour" dans le fichier *coucou.txt*

```
echo "Bonjour" >> coucou.txt
```


EXERCICE

- Se connecter à la machine
- Créer un répertoire « prenom »
- Ecrire « Hello World » dans le fichier hello.txt

FONDAMENTAUX

- Il faut savoir que sous UNIX, on se déplace dans des répertoires
 - Chemins absolus
 - Retour à la racine avec le « / » en tout premier lieu
 - Chemins relatifs
 - On peut retourner au répertoire de l'utilisateur en cours en utilisant le tilde « ~ »
 - On peut poursuivre à partir du répertoire sur lequel on est positionné, en utilisant le point « ./ » ou sans rien utiliser
 - On peut remonter au répertoire parent en utilisant deux fois le point « ../ »

FONDAMENTAUX

- Aide

```
man commande
```

- Déterminer le répertoire dans lequel vous êtes
 - **p**rint **w**orking **d**irectory

```
pwd
```

- Se déplacer dans l'arborescence
 - **c**hange **d**irectory

```
cd
```

Exemples

```
man pwd
```

Aide sur la commande **pwd**

```
pwd
```

/home/ajc

```
cd projets
```

/home/ajc/projets

```
cd ../
```

/home/ajc

```
cd /usr/local
```

/usr/local

EXERCICE

- Vérifier le répertoire en cours
- Revenir à la racine
- Revenir au répertoire précédent
 - Soit avec un chemin relatif (relatif à l'utilisateur)
 - Soit avec un chemin absolu
- Vérifier le répertoire en cours

FONDAMENTAUX

- Certaines commandes ou programmes peuvent avoir des options
 - Les options se précèdent généralement
 - D'un tiret « - » suivi d'une lettre (nom raccourcis de l'option)
 - De deux tirets « -- » suivi d'un mot (nom complet de l'option)
 - Il est possible de faire suivre plusieurs options avec un seul tiret
 - -ab
 - -Rf

FONDAMENTAUX

- Lister le contenu d'un répertoire
 - `ls`

```
ls [options] [chemin]
```

- Liste tous les fichiers du répertoire courant avec des détails

```
ls -la
```

- Visualiser le contenu d'un ou de plusieurs fichiers
 - `cat`

```
cat [options] <fichier> [fichiers]
```

EXERCICE

- Lister tous les fichiers en liste du répertoire /dev
- Ecrire cette liste de fichiers dans un fichier « fichiers.txt »
- Imprimer le contenu de ce fichier

FONDAMENTAUX

- Visualiser le contenu d'un fichier avec une pagination (si celui-ci est trop long)

```
more [options] <fichier> [fichiers]
```

- Visualiser un fichier avec pagination, navigation et recherche (dans un flux)

```
less [options] <fichier> [fichiers]
```

- Editer un fichier

```
vi [options] <fichier>
```

```
nano [options] <fichier>
```

- Statistiques sur un fichier

```
wc <option> <fichier>
```

```
wc -l fichier
```

Nombre de lignes d'un fichier

```
wc -c fichier
```

Nombre de caractères d'un fichier

EXERCICE

- Créer un fichier « super.txt » avec nano (y mettre du contenu)
- Enregistrer le fichier
- Afficher le contenu des fichiers « super.txt » et « fichiers.txt »

FONDAMENTAUX

- Il est possible de donner à un autre programme le résultat de la commande
 - Utilisation du *pipe* (ou *tube*) par le symbole trait vertical « | »

```
programme1 | programme2
```

- Ici, **programme2** va utiliser le résultat de **programme1** comme s'il s'agissait du contenu d'un fichier

EXERCICE

- Avec les commandes vues précédemment
 - Combien de fichiers et répertoires sont contenus dans le répertoire racine ?
- Placer ce résultat dans un nouveau fichier « count.txt »
 - Vérifier son contenu

FONDAMENTAUX

- Copier un fichier
 - `copy`

```
cp [options] <chemin_source> <chemin_destination>
```

- Copie *mon_fichier.txt* et nomme ce nouveau fichier *mon_fichier.bck.txt*

```
cp mon_fichier.txt mon_fichier.bck.txt
```

- Copie le répertoire *projets* dans le répertoire */tmp*

```
cp -R projets /tmp
```

FONDAMENTAUX

- Déplacer un fichier
 - `move`

```
mv [options] <chemin_source> <chemin_destination>
```

- Renomme *mon_fichier.txt* en *mon_fichier.bck.txt*

```
mv mon_fichier.txt mon_fichier.bck.txt
```

- Déplace le répertoire *projets* dans le répertoire */tmp*

```
mv projets /tmp
```

EXERCICE

- Renommer « count.txt » en « count-files.txt »

FONDAMENTAUX

- Supprimer un fichier

- `remove`

```
rm [options] <chemin>
```

- Supprime *mon_fichier.txt*

```
rm mon_fichier.txt
```

- Supprimer récursivement le répertoire *projets*

```
rm -r projets
```

- Supprime tous les fichiers se terminant par *.txt* sans demander confirmation

```
rm -f *.txt
```

FONDAMENTAUX

- Créer un répertoire
 - **make directory**

```
mkdir <chemin>
```

- Crée le répertoire *projets*

```
mkdir projets
```


EXERCICE

- Créer un répertoire « projets »
- Déplacer tous les fichiers .txt dans ce nouveau répertoire
- Vérifier le contenu du répertoire projets et de son répertoire parent

FONDAMENTAUX

- Rechercher un motif dans un fichier
 - **g**eneral **r**egular **e**xpression **p**rocessor

```
grep [options] <expression régulière> <chemin>
```

- Cherche dans tous les fichiers ceux qui contiennent "bonjour"

```
grep "bonjour" /home/ajc/*
```

EXERCICE

- Lister tous les fichiers avec leur détails du répertoire /dev
 - Ne garder que ceux qui font référence à « sda »
 - Enregistrer le résultat de cette recherche dans un fichier « disks.txt »

EXERCICE

- Un fichier carnet contient plusieurs lignes sous la forme nom:premier:telephone
 - Compter le nombre de personnes
 - Afficher toutes les lignes correspondant à « DUPONT »
 - Afficher toutes les lignes ne correspondant pas à « DUPONT »
 - En utilisant les commandes **head** et **tail**, afficher la 5^{ème} ligne du fichier

FONDAMENTAUX

- Créer un lien
 - `ln`

```
ln [options] <chemin_source_absolu> <lien>
```

- Il existe deux types de liens
 - Lien physique (ou matériel)
 - Permet de donner plusieurs noms à un même fichier
 - Tant qu'un emplacement n'est pas supprimé, le fichier existe toujours
 - Lien symbolique
 - Permet de donner un autre point d'accès à un fichier
 - Lorsque la source est supprimée, tous les liens symboliques sont supprimés

FONDAMENTAUX

- Crée un lien

```
ln /home/ajc/toto.txt /home/ajc/projets/toto.txt
```

- Crée un lien symbolique

```
ln -s /home/ajc/toto.txt /home/ajc/projets/toto.txt
```

FONDAMENTAUX

- Connaître des informations sur les processus (programmes en cours d'exécution)
 - **process status**

```
ps [options]
```

- Connaître l'activité du système

```
top [options]
```

- Tuer un processus
 - **kill**

```
kill [options] <PID>
```

- Tue le processus identifié par le numéro PID 1337

```
kill -9 1337
```

EXERCICE

- Exécuter la commande suivante avec une pagination

```
ps -aux
```

- Chercher un processus qui s'appelle « bash »
 - Le tuer en utilisant l'option « -9 »
 - Que s'est-il passé ?

FONDAMENTAUX

- Exécuter une commande en tant que super-administrateur
 - super user **do**

```
sudo <commande>
```

- Nettoyer la console
 - **clear**

```
clear
```



ADMINISTRATION

LES COMMANDES

LES COMMANDES D'ADMINISTRATION

- Chaque fichier a des permissions
 - Pour un utilisateur propriétaire (u)
 - Pour un groupe (g)
 - Pour les autres (o)
 - Pour tout le monde (a)

- Chaque permission a trois actions
 - Lecture
 - Ecriture
 - Exécution

- Un fichier aura par conséquent 3 numéros
 - 777
 - 644
 - 660

Accès à tout, par tout le monde

Seul le propriétaire peut lire et écrire, les autres peuvent lire, personne ne peut exécuter

Le propriétaire et le groupe ont des droits en lecture et écriture

#	Définition	rwX
0	Aucune permission	---
1	Exécution	--X
2	Ecriture	-W-
3	Ecriture et exécution (1+2)	-WX
4	Lecture	r--
5	Lecture et exécution (1+4)	r-X
6	Lecture et écriture (2+4)	rw-
7	Lecture, écriture et exécution (1+2+4)	rwX

LES COMMANDES D'ADMINISTRATION

- Modifier les droits d'un fichier

- **change mode**

```
chmod [options] <droits> <fichier> [fichiers]
```

- Autorise le fichier en exécution pour tout le monde

```
chmod +x script.jar
```

- Le fichier n'est autorisé à être exécuté par personne

```
chmod 644 script.jar
```

- Accès complet au répertoire *projets*, de façon récursive

```
chmod -R 777 projets
```

- Retire le fichier en exécution, seulement pour les autres

```
chmod o-x script.jar
```

LES COMMANDES D'ADMINISTRATION

- Modifier le propriétaire d'un fichier
 - **change owner**

```
chown [options] <utilisateur> <fichier> [fichiers]
```

- Le nouveau propriétaire du fichier *script.jar* est maintenant jeremy

```
chown jeremy script.jar
```

LES COMMANDES D'ADMINISTRATION

- Modifier le groupe d'un fichier

- **change group**

```
chgrp [options] <groupe> <fichier> [fichiers]
```

- Le nouveau groupe du répertoire *cours* est maintenant formateurs

```
chgrp -R formateurs cours
```

LES COMMANDES D'ADMINISTRATION

- Ajouter un utilisateur

- **user add**

```
useradd [options] <login>
```

- Ajouter un groupe

- **group add**

```
groupadd [options] <group>
```

LES COMMANDES D'ADMINISTRATION

- Modifier un utilisateur
 - **user mode**

```
usermod [options] <login>
```

- Ajoute le groupe "formateurs" à l'utilisateur "julie"

```
usermod -a -G formateurs julie
```

- Modifier le groupe de julie

```
usermod -G formateurs julie
```

```
passwd [options] <login>
```

- Modifier le mot de passe d'un utilisateur
 - **password**

```
usermod -a -G formateurs julie
```

```
usermod -G formateurs julie
```


LES COMMANDES D'ADMINISTRATION

- Modifier un utilisateur

- **user mode**

```
usermod [options] <login>
```

- Ajoute le groupe "formateurs" à l'utilisateur "julie"

```
usermod -a -G formateurs julie
```

- Le groupe de "julie" est "formateurs"

```
usermod -G formateurs julie
```

LES COMMANDES D'ADMINISTRATION

- Supprimer un groupe

- **group delete**

```
groupdel [options] <group>
```

- Supprimer un utilisateur

- **user delete**

```
userdel [options] <login>
```

LES COMMANDES D'ADMINISTRATION

- Les utilisateurs sont contenus dans le fichier
 - */etc/passwd*
- Les mots de passe sont contenus dans le fichier
 - */etc/shadow*
- Les groupes sont contenus dans le fichier
 - */etc/group*