

UNIX Utilisation

Utilisation

Olivier D.

Table des matières

1	Unix – Utilisation	2
1.1	Les bases.....	2
1.2	Les commandes	3
1.3	Arborescence Unix.....	4
1.4	Les types de fichiers.....	5
1.5	Les caractères spéciaux	7
1.6	Gestion des liens.....	8
1.7	Commandes de recherche.....	9
1.8	Édition de texte avec vi – kit de survie	10
1.9	Processus et mécanismes	13
1.10	Les redirections	14
1.11	Permissions.....	15
1.12	Archivage et compression	16
1.13	Le Shell.....	17

www.informatique1.fr

1 Unix – Utilisation

Avant-propos : cette documentation est une remise à jour de 2021 d'une doc datant de 2010.

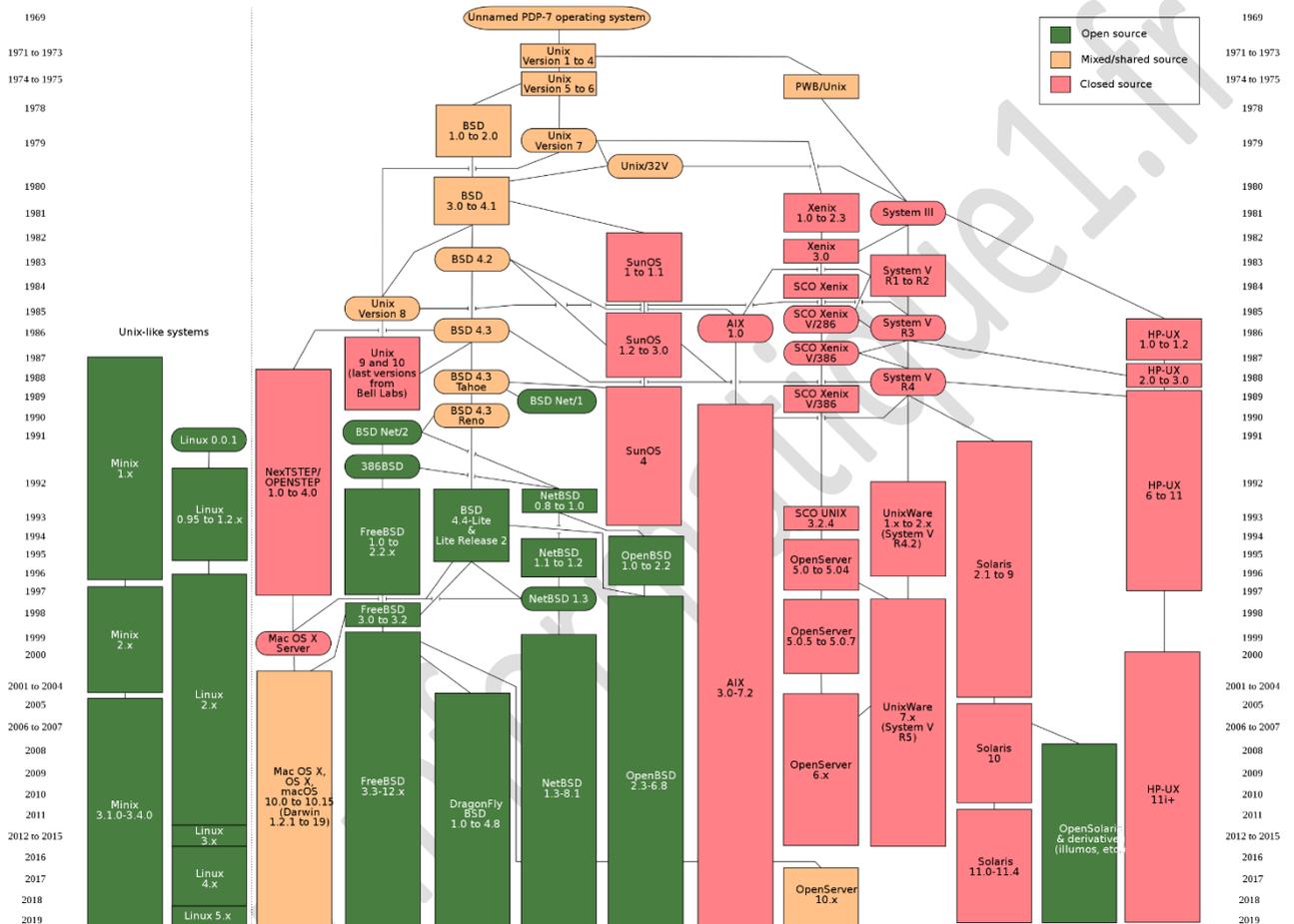
J'ai notamment rajouté beaucoup de liens et de références.

1.1 Les bases

Les [versions LTS de Ubuntu](#) sortent tous les 2 ans : fait pour l'entreprise

Historique

Source : <https://fr.wikipedia.org/wiki/Unix>



Historique des versions majeures de systèmes Unix

Shell les plus communs

sh [Bourne shell](#)

csch [C shell \(BSD\)](#)

ksh [Korn shell \(System V\)](#)

bash [Bourne again shell \(Linux\)](#)

Syntaxe des commandes

Commande [options] [arguments] (en général les options sont préfixées d'un -)

- On peut ajouter le chemin de la commande
- Attention à la casse : fic.txt ≠ fic.TXT ≠ Fic.txt

Le prompt

...\$ utilisateur

...# super utilisateur : root

Se déconnecter de Putty (putty est un outils permettant de se connecter par le biais d'une connexion ssh)

exit

[Ctrl] + D = ^D

Quelques commandes

tty : sur quel terminal suis-je

- /dev/pts/10 (pts : pseudo terminal)
- /dev/tty/10 (tty : terminal)

id : identifiant utilisé actuellement

who -m : identifiant à la connexion

who : tous les comptes connectés à l'instant t

- who -q liste abrégée
- who -a liste de tous les utilisateurs

passwd : changer son mot de passe (vérifier son id actuel avec id avant de le faire)

date : date actuelle

- date "+%n%tcoucou%n" affiche entrée tabulation coucou entrée
- date "+Fichier-%Y_%m_%d" pratique pour nommer un fichier par script

cal 12 2010 : affiche le calendrier de décembre 2010

- cal 10 affiche le calendrier de l'an 10

L'aide

--help : affiche une aide succincte (ex : date --help)

man : affiche les pages de manuel

- man -s 5 passwd affiche le manuel de la section 5 de passwd

grep SECTION /etc/manpath.config : ordre d'affichage des sections

man man : manuel de man (contient les numéros et clair des sections)

man -k copy|copie : trouver une commande qui sert à copy|copie

<http://www.delafond.org/> : traduction des pages de man

<https://www.ubuntu-fr.org/> : page de Ubuntu, beaucoup de documentation

- man man /VOIR (recherche le mot VOIR)
- n (occurrence suivante)

1.3

Arborescence Unix

Les lettres de lecteur (type Windows) sont des répertoires contenus quelque part sous la racine /

Les dossiers peuvent être de différents types de fichiers (FAT32, NTFS, ISO9660, CIFS ...)

Chemin absolu

`cd /` : positionne à la racine

`cd /etc` : positionne dans /etc

Chemin relatif

`cd ../stag10` : remonte dans le répertoire parent puis va dans le sous répertoire stag10

`cd cours` (= `cd ./cours`) positionne dans le sous dossier cours à partir du répertoire actuel

Un chemin absolu commence toujours par / (position à partir de la racine)

Un chemin relatif ne commence jamais par / (chemin à partir de ma position)

Le . représente le dossier actuel

Le .. représente le répertoire parent

Le FHS (FileSystem Hierarchy Standard)

Objectif : standardiser les arborescences Unix et Linux

<code>/bin</code>	commandes de base pour tous les utilisateurs
<code>/usr/bin</code>	
<code>/sbin</code>	commandes indispensables au démarrage du système
<code>/usr/sbin</code>	commandes d'administration système
<code>/dev</code>	(devices) fichiers spéciaux de périphériques
<code>/etc</code>	fichiers de configuration du système
<code>/home</code>	répertoire par défaut des utilisateurs
<code>/lib</code>	bibliothèques principales + modules du noyau
<code>/usr/lib</code>	autres bibliothèques
<code>/lost+found</code>	fichiers récupérés par la commande fsck
<code>/media</code>	répertoire de base de montage des périphériques de stockage amovibles
<code>/mnt</code>	vide par défaut. Utilisé pour servir de point de montage ponctuel
<code>/opt</code>	répertoire pour les applications tierces
<code>/root</code>	répertoire d'accueil du super utilisateur root
<code>/srv</code>	données pour les services hébergés par le système. Peu utilisé
<code>/tmp</code>	répertoire temporaire
<code>/usr</code>	de tout sur tout
<code>/var</code>	fichiers de tailles variables (log, spool d'impression, mails, cache ...)
<code>/proc</code>	tous les processus en cours

Les types de fichiers

- Les fichiers ordinaires : tous les fichiers standards (fichiers txt, commandes, etc.) **f**
- Les fichiers répertoires (exemple : /Home2) **d**
- Les fichiers de périphériques : tous dans /dev
- Certains en mode blocs : stockage
- Certains en mode caractère (raw) : claviers, souris, etc.
- Les liens symboliques (\approx raccourcis Windows) **l**

Les inodes

L'inode 5000 (par exemple) contient :

- Propriétaire, groupe propriétaire, permissions, type de fichier, dates, taille du fichier, nombre de liens physiques
- Pointeurs : liens directs, liens indirects

(!) L'inode ne contient pas le nom du fichier !!

Fichier dossier → Nom de fichier + Inode. L'inode pointe vers le contenu du fichier

Manipuler les fichiers

pwd : affiche le chemin du répertoire courant

cd : changer de répertoire

- **cd -** revenir au répertoire précédent
- **cd ~** aller dans le répertoire d'accueil → **cd ~/Travail = cd /Home2/stag16/Travail**

mkdir : créer un|des répertoires (en chemin absolu|relatif)

- **mkdir ~/Dossier1/Sous_dossier** Dossier1 doit exister
- **mkdir -p ~/Dossier1/2010/Nov** créé les répertoires parents si nécessaire

mkdir "Dossier 1/Sous_dossier"

- **rmdir** supprime un répertoire vide
- **rmdir -p "Dossier 1/Sous_dossier"** → Supprime les répertoires parents (si **vide**)

ls : liste le contenu du répertoire courant

- **ls -a** affiche les fichiers et dossiers cachés
- **ls -A** affiche les fichiers et dossiers cachés sauf . et ..
- **ls -R /etc** affiche le contenu de /etc et de ses sous-dossiers
- **ls -li Edition** affiche le n° d'inode de Edition

```
ls -li Edition -rw-r--r- stag16 stag16 586 2010-11-15 Edition
```

Types de fichiers :

- **-** : fichier standard
- **d** : fichier répertoire
- **l** : lien symbolique
- **b** : fichier spécial mode bloc
- **c** : fichier spécial mode caractère)

rw- / **r--** / **r--** : autorisation propriétaire / groupe propriétaire / autres utilisateurs (**r** : lecture / **w** : écriture / **x** : exécution)

1 nombre de liens physiques

stag16 propriétaire

stag16 groupe propriétaire

586 taille du fichier (en octets)

2010-11-15 date de dernière modification

Edition nom du fichier

`ls -ld /etc` affiche les informations détaillées du répertoire /etc

`touch` modifie la date de modification | crée le fichier s'il n'existe pas

- `touch fic.txt` crée le fichier fic.txt
- `touch Edition` modifie la date de modification de Edition

`stat` affiche toutes les dates d'un fichier

`cp` copie des fichiers

- `cp Edition edition.save` copie Edition dans edition.save
- `cp -i /etc/passwd edition.save` prévient avant d'écraser edition.save
- `cp -f /etc/passwd edition.save` force la copie sans prévenir
- `cp -p ...` preserve les permissions, le propriétaire, le groupe propriétaire, les dates
- `cp -a ... = cp -dpR` crée une archive avec -p -R (recrée l'arborescence source)

`rm` suppression de fichier

- `rm -R Dossier1` supprime le dossier Dossier1 **et son contenu**
- `rm -i Dossier1/*` supprime le contenu en posant les questions
- `rm -f Dossier1/*` supprime le contenu sans poser de question

Si root, penser à créer un répertoire d'accueil /root en cas de cd puis `rm -Rf*`

`mv` déplacement de fichier ; renommage

- `mv fic.txt fic.doc` renomme fic.txt en fic.doc
- `mv fic.doc Dossier /` déplace fic.doc dans Dossier/ (touche aux « fichiers dossiers » source et destination) =
`mv fic.txt Dossier/fic.doc`
- `mv Dossier1 Dossier2 :`
 - 1) si Dossier2 n'existe pas → renomme Dossier1 en Dossier2
 - 2) si Dossier2 existe → Dossier2/Dossier1/
- `mv Dossier1/* Dossier2` copie le contenu de Dossier1 dans Dossier2

`file` indique le type de fichier

`cmp|diff` : compare les fichiers

- `cmp fic1 fic2` : si les 2 fichiers sont identiques : n'affiche rien ; sinon affiche la 1ere différence

`diff fic1 fic2` : indique les différences et ce qu'il faut faire pour les rendre identiques

1.6

Gestion des liens

Tuto : https://doc.ubuntu-fr.org/lien_physique_et_symbolique

`ln` : lien physique

`ln -s` : lien symbolique

Lien Physique : tout pareil, que le nom différent

`ln Edition edition.txt` : crée un fichier edition.txt qui pointe sur le même inode que Edition.

- Si on supprime Edition, on peut toujours accéder au fichier avec edition.txt (ça diminue juste le nombre de liens physiques)

Avantages :

- Peu gourmand, crée juste une ligne de texte
- Très performant
- Déplacement de la source possible (**dans la même partition**)

Inconvénients :

- on ne peut pas créer de lien physique entre FS différents
- on ne peut pas faire de lien physique de répertoire

Info : il ne faut pas faire de lien physique d'un fichier qui risque de changer de FS (archives, etc.)

Lien Symbolique (idem qu'un raccourci Windows)

`ln -s ~/Edition EDITION.TXT` : crée un lien symbolique EDITION.TXT pointant vers ~/Edition

Il est conseillé de donner le chemin absolu de la source

Avantages :

- On peut faire des liens symboliques sur des partitions différents

Inconvénients :

- Il ne faut pas changer la source du lien (nom, emplacement ...)
- Beaucoup de manipulations pour l'ordinateur
- Consomme des inodes

Info : `mkdir Dossier10 -> drwxr-xr-x 2 ...`

`2` représente le nombre de sous répertoires dans le répertoire, avec `.` et `..` inclus)

Commandes de recherche

`grep` : affiche un ligne à partir d'une recherche de chaînes dans les fichiers

`find` : recherche dans l'arborescence

grep utilise des expressions régulières (tout comme sed, awk, vi ...)

Les expressions régulières

`.` : 1 caractère quelconque

`ls|grep -> "f.c" : fic, fac, f0c ...`

`*` : répétition du caractère qui précède * de 0 à n fois

`ls|grep "a*" -> <rien>, a, aa, aaa ...`

`.*` : une séquence de caractère quelconque

`[0-9]` : 1 caractère, de 0 à 9

`[aei]` : 1 caractère soit a soit e soit i

`[^0-9]` : 1 caractère QUI N'EST PAS de 0 à 9

`^fic` : ce qui commence par fic

`txt$` : ce qui finit par txt (`grep "^$" -->` renvoie les lignes vides (ne prend pas en compte les espaces))

`\<fic` : 1 mot qui commence par fic

`txt\>` : 1 mot qui finit par txt

`0{x,y}` : 0 répété au moins x fois, au maximum y fois

`0{x}` : 0 répété x fois

`0{x,}` : 0 répété au moins x fois, pas de maximum

La commande `grep` sert à chercher des chaînes dans un fichier

`grep "Jean" Edition` : affiche les lignes contenant la chaîne Jean

`-i` : ignore la casse dans le motif de recherche

`grep "Jean" *` : affiche les lignes contenant Jean dans tous les fichiers

`-l` : affiche les noms de fichiers contenant le motif de recherche

`-v` : affiche les lignes NE CONTENANT PAS le motif de recherche

Exemple : `grep -v [0-9]$ Edition`

`egrep -v "[0-9]$|^$" Edition` : affiche les lignes non-vides ET qui ne finissent pas par [0-9]

La commande `find` sert à chercher dans l'arborescence

`find = find .`

`find /home2 -name Edition` : indique l'emplacement de chaque Edition dans /home2

`find /home2 -iname edition` : `-i` : sans tenir compte de la casse

`find /home2 -inum 90119` : `-inum` : recherche les n° inode

`find /home2 -user stag16` : `-user` : utilisateur ; `-group` : groupe

`find /home2 -type f -user stag16` : recherche les fichiers standards appartenant à stag16

Info : `find /` : recherche partout à partir de / (y compris les autres FS)

`-xdev` : ne pas chercher dans les autres FS

`-maxdepth 4` : recherche maximum dans le 4e sous-répertoire

`find -user stag16 -o -user stag18` faire une recherche avec ou logique (`-o`)

`find ~-type f -name "*.txt" -exec echo coucou \;`

`-exec cp {} {}.save \;{} -->` le résultat trouvé par `find`

Trouver dans **/etc** les fichiers standard, puis chercher le fichier qui contient **andromède** dans ces fichiers :

`find /etc -type f -exec grep -l "andromede" \;`

man vi : aide de vi

vi Edition : ouvrir Edition avec vi

Sortir de vi

:q! : sans enregistrer

:wq! : enregistrer avant

Mode commande --> [Echap] --> sous-mode ex [:]

Mode insertion --> i (insère au caractère courant) a (insère au caractère après)

Supprimer

1 caractère : x

1 ligne : dd

Annuler : u

Supprimer : d – tout supprimer : ggdG – supprimer de la position actuelle au début du mot suivant : dw

d<déplacement> : jusqu'au <déplacement>

Se déplacer

h : gauche

j : haut

k : bas

l : droite

Sauts

w : (*word*) début de mot suivant

W : début du mot suivant le prochain espace

b : (*before*) début de mot précédent

B : début du mot précédant l'espace avant

e : (*end*) fin du mot

E : fin du mot situé après l'espace après

G : fin du document (12G : 12e ligne)

gg : début du document

^ : début de ligne

\$: fin de ligne

Recherche

fx : place au prochain caractère x sur la même ligne

/Jean : recherche Jean de haut en bas

n : suivant

N : précédent

Insertion

A = \$a : insertion en fin de ligne

I = ^i : insertion en début de ligne

o : crée une nouvelle ligne après et place en mode insertion

O : crée une nouvelle ligne avant et place en mode insertion

rZ : changer le caractère actuel par Z (reste en ligne de commande)

c<déplacement> : changer tout jusqu'au <déplacement> par autre chose

>> : insère une tabulation

Recopie

p : (paste) recopie le contenu du tampon (de la dernière suppression)

dd 10G 2p : coupe la ligne courante, se déplace à la ligne 10, la recopie 2 fois

yy : copie la ligne

y7G : copie jusqu'à la ligne 7

y/Dupont : copie jusqu'à Dupont (sans intégrer Dupont)

J : replace la ligne courante à la fin de la ligne précédente

Recherche

:g/Dupont/s//Dupond/g --> :1/2/3/4/5/6 (= :%s/Dupont/Dupont/g)

1 : g : recherche globale sur toutes les lignes du document

2 : Dupont : rechercher les lignes qui contiennent Dupont (*expression régulière*)

3 : s : substituer

4 : <vide>-->2-->Dupont : motif Dupont à substituer

5 : Dupond : remplacer par Dupont (*expression régulière*)

6 : g : rechercher tous les motifs sur la ligne

exemples :

```
:10,20 g/Dupont/s/^/ - /g=e_
```

```
:g/universe/s/^/#/g
```

```
:g/universe/s/^#/g
```

```
:g@/usr/bin@s@@/opt@g (on peut remplacer / par @)
```

```
:g/[Jj]ean/s/&-Marc/g (& récupère (5))
```

Les tampons nommés (pour garder en mémoire plusieurs copies/suppression)

"ayw : la copie du mot s'appelle a

"byy : la copie de la ligne s'appelle b

"Byy : la copie de la ligne est rajoutée dans b

"ap : colle a

Execution de commandes depuis vi

:!cmd : execute cmd sans quitter vi

:r!date : insère la date après la ligne courante

:w : sauvegarder

:w!~/toto.txt : sauvegarder sous ~/toto.txt

:e!fich2 : ouvrir fich2 par dessus (:w avant)

:e# : passer d'un doc à l'autre

(info : pour copier une ligne d'un document à l'autre, passer par les tampons nommés)

Options utiles dans vi

:set number : affiche les n° de lignes

:set all : afficher toutes les options disponibles

:set nonumber : enleve les n° de lignes

:set ignorecase : ignorer la casse

:set tabstop=4 : définir le nombre d'espaces pour >>

:set syntax=on : afficher les couleurs de syntaxes

:set autoindent : repositionner à l'indentation précédente

:set nocompatible : passe en mode vim

En dehors de vi, pour régler les options par défaut

vi ~/.exrc : on crée le fichier des options de base de vi

```
set ignorecase
set number
set nocompatible
```

Pas de ligne vide dans .exrc !!

Transformer un texte Unix en texte Dos

dans vim :

:set fileformat=dos équivalent à :se ff=dos

en Shell :

od -c [nom du fichier]

ou alors, utiliser l'outil unix2dos

`ps` : lister les processus en cours d'exécution

Un daemon (=démon) est un processus en attente de requêtes (exemple : processus de syslog → `syslogd`). Un daemon se rapproche de l'idée qu'on se fait d'un service de type Windows

`ps -ef` : tous les processus qui tournent

`ps aux` : idem que `ps -ef`

- **PID** : process Id
- **PPID** : PID parent (si on tu un PPID, les PID enfants y passent aussi)
- **UID** : User Id propriétaire du processus
- **PRI** : priorité du processus.

`ps -ef | grep sshd` : trouve dans les processus le daemon du ssh (donc un ssh en cours d'exec.)

`ps -ef | grep log` : trouve un fichier d'événements (log) en cours d'exécution

`kill -l` : renvoie tous les signaux de kill existants

`kill 1515 = kill -15 1515` « kill standard, gentil »

`kill -2 1515` : « halte au feu, interruption »

`kill -9 1515` : kille instantanément et de façon très sale un processus

`time` : référence les temps d'exécution

`time sleep 3` : durée exacte d'un `sleep 3` (sleep ne consomme pas de ressources)

`^Z` : envoie un processus en arrière plan

`jobs` : affiche les processus en arrière plan

`bg` : (background) relance le dernier processus et le laisse en arrière plan

`fg` : (foreground) ramène le processus d'arrière plan en avant plan

`<commande> &` : lance une commande en arrière plan et qui n'est pas un process enfant

`commande > /dev/pts/13` : renvoie la commande vers un autre écran

Exemple : `find ~ > /dev/pts/9 &` : renvoie la commande vers `/dev/pts/9` et la place en arrière plan

Les priorités



Gestion des priorités

Il n'y a que root qui peut mettre des priorités inférieures à 0

Moyen mnémotechnique : ce sont les temps de latence des commandes.

`nice <commande>` : exécute une commande avec une priorité +10 par défaut

`nice -5 indexba.sh` : exécute la commande avec une priorité -5

`nice -n +5 indexba.sh` : exécute une commande en ajoutant +5 à la priorité de base (+10), donc +15 en somme

`renice <commande>` : fait un nice sur un processus en cours

1.10 Les redirections



Redirections et canaux

`ls fic* > resultat.txt` : redirige le canal 1 (par défaut) vers le fichier qui est créé et écrasé

exemple : `ls fic* 1> trouvé.txt 2> erreurs.txt`

`date >> resultat.txt` : redirige le canal 1 vers le fichier qu'il incrémente (ou crée si besoin)

`wc -l < Edition` : redirige Edition (canal 0) vers `wc -l`

`ls fic* xyz > result.txt 2>&1` : redirige les canaux 1 et 2 vers `result.txt`

`2>/dev/null` : ne pas afficher les erreurs

`find /home2 -type f -exec grep -l "xyz" {} \; 2>/dev/null` : n'afficher que le résultat de la recherche

Les pipelines

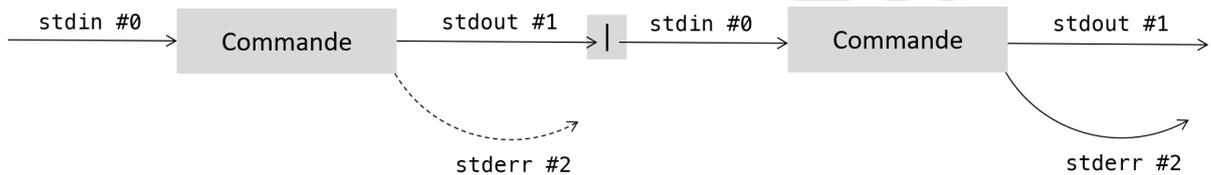


Illustration des pipelines

`ps -ef | grep log` : trouver les processus actifs contenant la chaîne `log`

`ls -l /etc | grep "^-" | grep "\.conf$" | wc -l` : lister les éléments de `/etc` et les mettre en entrées pour y chercher la chaîne dont la ligne commence par `-` et la ligne finit par `.conf` et les mettre en entrées pour y compter le nombre de lignes

`ls fic*|tee resultat.txt` : afficher à l'écran ET dans `resultat.txt` qu'il écrase

`ls fic*|tee -a resultat.txt` : afficher à l'écran ET dans `resultat.txt` qu'il incrémente

1.11 Permissions

Lien utile : <https://doc.ubuntu-fr.org/permissions>

- Aucun droit : 0
- Lecture : +4
- Ecriture : +2
- Exécution : +1

Exemple : un fichier avec les permissions 644 est défini de la façon suivante :

- Utilisateur propriétaire : lecture + écriture = 4+2 = 6
- Groupe propriétaire : lecture = 4
- Autres (non-membres du groupe propriétaire) : lecture = 4

Permission	Fichiers (hors type d)	Dossiers
R	Lecture	Voir le contenu
W	Ecriture	
X	Exécution	Explorer

`umask` : masque des permissions des fichiers et dossiers créés par l'utilisateur

`umask 000` : change le umask dans le shell jusqu'à la prochaine ouverture de session

Fonctionnement de l'`umask` : lorsqu'on crée un fichier, par défaut ses droits sont au maximum (666 pour un fichier, 777 pour un dossier). L'`umask` applique une soustraction à ses droits par défaut. L'`umask` par défaut est 022

- Donc, quand on crée un fichier : $666 - 022 =$ les droits du fichier créé sont 644
- Quand on crée un dossier : $777 - 022 =$ les droits du dossier créé sont 755

Pour modifier le umask par défaut, on peut modifier le fichier des paramètres de l'utilisateur (fichier de profil) :

```
umask 055 # par exemple
```

Bash : `~/profile` (profil de l'utilisateur) ou `/etc/profile` (profil commun à tous les utilisateurs) ou `~/bashrc` (profil de l'utilisateur en dehors de bash)

Changer les permissions d'un fichier ou d'un dossier

`chmod 666 perm.txt` : change les permissions absolues de perm.txt en 666

`chmod g-w` : enlève la permission w à g

u : propriétaire	+ : ajouter	w : écrire
g : groupe propriétaire	- : enlever	r : lire
o : autres	= : absolu	x : exécuter

`chown toto perm.txt` : changer le U (besoin d'être root)

`chown toto:TSRITE-AR perm.txt` : changer le U:G

`chgrp` : changer le G

1.12 Archivage et compression

Archivage : regrouper plusieurs fichiers et dossiers en un seul fichier (extension .tar ou .cpio)

Compression : compresser un seul fichier (une seule archive) .gzip .bzip2

Archivage (fonctionnement de tar)

-x : extract

-c : créer (archiver)

-u : update (mise à jour)

-t : lister

Exemple : tar -x

-f : fichier

-v : verboze

-z : invoque gzip (pour compresser et décompresser)

Exemple : décompresse un fichier.tar.gz et place son contenu dans /tmp

```
tar -xvzf fichier.tar.gz -C /tmp
```

Compression

gzip : compresser

gunzip : décompresser (= gzip -d)

gzip Edition : **remplace** Edition par Edition.gz

gzip -c Edition > Edition.gz : crée un fichier Edition.gz à partir de Edition

Les variables

Une variable est temporaire !! Pour la modifier tout le temps, il faut modifier le profil de l'utilisateur (~/.bashrc)

PS1 : prompt1

PS2 : prompt2

PS2 = ">>>" : remplace le PS2 par >>>

set : afficher toutes les variables

env : afficher les variables d'environnement

PATH : chemin qui indique où aller chercher les commandes

PATH=\$PATH:. : pour ajouter . à la fin de la variable PATH

Les Alias

alias ll = "ls -l" : crée un alias ll qui exécute ls -l

unalias ll : supprime l'alias ll

Rappel de commandes

En bash :

[flèche haut] : rappelle et exécute la commande précédente (=!!)

!chm : rappelle et exécute la dernière commande commençant par chm

!?eth0 : rappelle et exécute la dernière commande contenant eth0

history|grep ps : affiche l'historique et y cherche les lignes contenant ps

!479 : exécute la commande 479 (voir dans l'historique)

apti [tab] : remplit la ligne (utilise l'autocomplétion)

\$HIST[tab][tab] : affiche les possibilités

/etc/a[tab][tab] : affiche tout ce qui commence par a dans /etc