Linux Cheat Sheet

by solsolsol via cheatography.com/32070/cs/10383/

Manipulation de fichiers

cat *file1...* : Concatène les files passés en paramètre et les affiche sur stdout (écran)

tac *file1...* : Idem cat mais à l'envers

cd *path* : se déplace dans le rep. indiqué

cd : reviens dans le rep. de connexion (home)

cd - : reviens au rep. précédent

cp *source1... dest* : Copie source vers dest

cp -r *dir1... dir2* :d1 to d2, create d2 if no d2

-i: demande confirmation si écrasement

-p :préserve dates d'accès et de modification

-r : copie récursive

mv : *file1 file2* : rename f1 to f2 & delete f1

ls [-aldf] *path1...* : peut etre file ou dir

-a : liste TOUS les fichiers (hiden...)

-l : format long (taille,date,droits,...)

-d : décrit le dir et non son contenu

-F : format court + type de file (*=exe /=dir)

-i : affiche numéro *d'inode* des files

-R : list subdir recursively

mkdir *path/dir* : crée un répertoire

touch *filename* : crée un fichier

rm *file/dir-name*: delete file/dir(-r ou **rmdir**) **-r** : recursive **-f** : force **-rf** : both

Echo

echo *txt* : affiche le txt. Méta-char et var d'env ds txt évalué et traduit avt affichage

-n : supprime saut de ligne

Guillemets ou apostrophes modifient comp.

$ ls

```
Modèles Musique a.txt
```

$ echo M*

```
Modèles Musique
```

$ echo M* $fichier

```
Modèles Musique a.txt
```

$ echo "M* $fichier"

```
M* a.txt
```

$ echo 'M* $fichier'

```
M* $fichier
```

Echo (cont)

echo [txt] > *file* : [txt] into f. If no f, make f

echo [txt] >> *file* : append [txt] to file

find

find *folder1...* [options]

-empty : fichiers vides

-name *nom_fichier* : f portant *nom_fichier*

-type *type_fichier* : f défini sous *type_fichier*

find . **–type d** –print (type répertoire)
find . **–type l** –print (type lien symbolique)
find . **-type f** -print (type normal file)

-newer *nom_fichier* :f + récent que *nom_f*

find . **–newer file1** –print (+ récent que file1)

-size [+-] *n* : Recherche par taille. **-n** = "inférieur à" **+n** = "supérieur à" (bloc 512 o)

find . **–size 4M** –print (4Mo exactement)
find . **–size –10G** –print (+ petit que 10 Go)
find . **–size +100k** –print (+ grd que 100 ko)

find . /etc –size +4c –type f –print : Normal file, find in curent and /etc

find . –perm –7 –type f -print: Normal file with **rwx** permission, find from curent folder

L'option **-exec** execute la commande entre **{}**

find . -type f -exec file '{}' \;

{} remplacés par le nom du fichier trouvé

cmd exécutée une fois par fichier trouvé

grep

grep [option] `motif` *file1...*

Affiche chaque ligne des *file* contenant le `motif`. Le motif est une expression régulière

-v : lignes qui ne contiennent **pas** le motif

-c : seulement le nombre de lignes

-n : numéros des lignes trouvées

-i : **pas** sensible à la casse

grep <motif> <fichiers> : parcourt les f et affiche les lignes qui corresp au motif

grep erreur *.log: affiche les lignes contenant **erreur** dans les fichiers *.log

grep -i erreur *.log : idem + osef casse

grep -ri erreur .: idem + récursif

grep -v info *.log: affiche toutes les lignes des fichiers sauf celles qui contiennent **info**

Affichage

head [-n] *file*: Affiche les n premières lignes

tail [+n | -n] *file* : La forme tail +n permet d'afficher un fichier à partir de la ligne n. La forme tail -n affiche les n dernières lignes

more *file* : affiche page par page

less idem more + revenir en arrière

man [n] *cmd* : affiche page de manuel (n spécifie la section)

Info sur les commandes et fichiers

file *file* : indication sur le type de données contenues dans un fichier. **(file /usr/bin/ls)**

type *cmd* : indique si une cmd possède une implémentation interne. **(type cd) (type ls)**

type indique qu'une cmd est interne, mais ne précise pas qu'il existe une implem. ext si il en existe une.

Cheatography

Linux Cheat Sheet

by solsolsol via cheatography.com/32070/cs/10383/

Info sur les commandes et fichiers (cont)

which *cmd* Affiche le chemin du fichier exec

ls *file* : indique le nom du fichier

Méta-caractères (remplace ...)

*	Suite de caractères. (0 -> max)
?	Un seul caractère quelconque.
[]	Par un des caractère entre crochet. Intervalle : [a-zA-Z]
{1,2,..,n}	Remplace par chaque élément de la liste.
ls *.c	files(not folder) that finish with .c
ls *g*	contenant un g
ls g*	qui commence par un g
ls *.?	. en avant dernier char
ls * g??	g en avant avant dernier
ls [hg]	qui commence par h ou g
ls [hg]*	contenant soit un h soit un g
.[a-z].	contenant [a-z] entre deux .
[1-9a-z]	

```
$ ls
file_bsd file_linux file_unix
$ ls file_{unix, linux}
file_linux file_unix
```

Pour empêcher interprétation des méta-char par le shell, placer arg entre apostrophes '

Divers

echo **$RANDOM** : return a random number

```
for i in {1..5}; do echo $RANDOM; done
```

/dev/null : poubelle. Utile pour y rediriger les sorties indésirables.

/dev/zero : Les lectures à partir de ce file renverront des char \0

/dev/full : renvoie une err disque full

Divers (cont)

<< est un délimiteur. Le flux d'entrée est directement connecté au flux qui donne la commande. Le délimiteur indique le début et la fin du txt.

```
$ sort -n'<< VERGE
>5
>1
>VERGE
1 5
```

date

cal : affiche un calendrier

diff *file1 file2* : compare ligne a ligne 2 files

wc [-cwl] *file1...* : affiche le nombre de **-c** char, **-w**: words, **-l** : lignes

who ([am i]) : liste les users co au système.

zcat *file1..* :cat + décomppression

touch [-acmt] *fichiers* : modifie la date d'accès et la dete de modif des fichiers.

-a :mod l'heure du dernier accès

-c : ne pas créer les fichiers n'existant pas

-m : change l'heure de la der mod du fichier

-t : Use la date indiquée à la place de actual date. arg = nb déci [[SS]AA]MMJJhhmm[.ss]

sort [-rnu] *file* : trie les lignes du fichier et écrit le resultat sur l'**STDOUT**. **-r** : renverse l'rodre du tri, **-n** : tri numérique

tr [options] *str1 str2* : recopie **STDI** sur **STDOU** en remplaçant tout char de str1 par char de position corresp dans str2

uniq [-cud] *file* : check ligne / ligne et détermine les l dupliquées consécutives.
-d : retient que les lignes dup**-u** : retient que l'non-dup **-c** : compte l'indice de répét

Contrôle de tâches

Unix prend en charge le multi-task préemptif:

-capacité d'un système d'exploitation multitâche à exécuter ou arrêter une tâche planifiée en cours.

"Tout dans Unix est fichier. Tout dans Unix qui n'est pas un fichier est un processus"

Contrôle de tâches (cont)

Processus: Instance d'un soft en cours d'exéc. Plusieurs inst d'un même soft peuvent s'exécuter en même temps.

Données associées aux processus: fichiers ouverts, mémoire allouée, pile, id processus, parent, priorité, état...

& à la fin de la ligne de cmd start une commande en bg.
$ calcul &
$ ls -Ral / > ls-Rl.txt &
$

2 cmd s'exéc en parallèle, tandis que le shell attend notre prochaine instruction.

jobs : liste des bg task lancées de ce shell
$ jobs
[1] Running calcul
[2] Running ls - Ral / > ls-Ral.txt

Pour kill une tache on utilise le n entre []
$ kill [-signal] %n

fg remet une tache de fond au premier plan

<ctrl>-z met la tache courante en fond

sleep n : suspend l'exec du process **n** sec

kill *-sig* PID : Envoie le signal*sig* au precess de num PID. *sig* peut etre soit le num du signal soit son non. Exemple:

kill -STOP 1023 = kill -19 1023

ps [-e][-l][aux] : affiche la liste des process **-l** plus d'info, **-e** process de tous les users **aux** : affiche tous les process sur le système

UID: ID du propriétaire du process

PID: numéro du process

PPID: PID du père du process

NI: priorité (nice[prio]) (**nice** [-prio] cmd)

S: état du process (**R** actif **S** bloqué **Z** terminé)

Contrôle de tâches (cont)

top : affiche en continu les process les plus actifs triés par utilisationn CPU.

Tar

tar [options] *fichier ou répertoire*

Permet d'archiver files ou une arborescence de fichiers CAD les regrouper dans 1 file.

tar cvf <archive> <fichiers ou répertoires>
c : créer **v**: suit progression **f**: fichier

tar tvf <archive> : affiche contenu d;une archive ou vérifie son intégrité (**t** : test)

tar xvf <archive>: ext tous les f d'1 archive

tar xvf <archive> <fichiers ou rép.> : ext seulement quelques fichiers d'une archive

Les fichiers ou répertoires sont donnés avec un chemin relatif au répertoire racine de l'archive.

rsync (remote sync)

Conçu pour synchroniser des répertoires sur 2 machines reliées par un lien à faible débit.

Ne copie que les fichiers qui ont changé. Les fichiers de taille identique sont comparés au moyen de sommes de contrôle.

Ne transfère que les blocs qui diffèrent au sein d'un fichier

Peut compresser les blocs transférés

Conserve les liens symboliques et les permissions sur les fichiers

Peut fonctionner à travers ssh (shell sécurisé)

Variables d'environnement

Le système Unix défini pour chaque process une liste de var d'env pout définir certains paramètres:

-rép d'installation des utilitaires

-type de terminal...

Chaque soft peut accéder à ces var pour obtenir des infos sur la config du système

Peut être use dans une cmd. Shell remplace chaque var par sa val avt d'exec la cmd.

Dans Bash, var d'env sont manip par des cmd

env : liste de toutes les variables

Variable=valeur : donne une val à une variable locale (processus actif)

export Variable=valeur: donne val à une var globale (proces actif + sous-process)

echo $Viande : affiche val var Viande

Pour ajouter une variable de façon permanente, il faut ajouter la ligne **export nomVar=var** au fichier **.bashrc** dans rep connexion.

Redirection des IO

Chaque programme sous UNIX dispose au moins de trois flux de données:

-**STDIN**, utilisée en lecture, qui est normalement associée au clavier du terminal.

-**STDOUT**, utilisée en écriture, normalement associée à l'écran du terminal

-**STDERR**, utilisée en écriture, pour afficher les messages d'erreur, normalement associée à l'écran du terminal

Redirection vers/depuis des fichiers

> permet de rediriger la sortie standard d'un programme vers un fichier.

$ **ls** > *file*

La commande **ls** va créer *file* et y écrire le résultat de la commande.

Pour append à un fichier on utilise >>

Pour rediriger l'entrée standard, on utilise <

$ cat < UnFichier more < fichier.txt

Il est possible de rediriger l'entrée et la sortie en même temps:

$ cat < fichier.txt > sortie.txt

$ ls / 2> error.log
Redirige les msg d'erreur

$ ls / > fichier.txt 2> error.log
stderr dans un file et stdout vers un autre

$ ls / &> fichier.txt
stdout et stderr sur le meme fichier

$ cat < input.txt > output.txt 2> error.log
Redirection de tous les flux

Redirection vers des tubes (pipe)

Redirige la sortie standard d'une commande vers l'entrée standard d'une autre. Les commandes s'exécutent alors en parallèle.

$ ls | sort -r : Affiche le contenu du répertoire trié à l'envers

Séquences de commandes

<commande 1> ; … < commande n>

Chaque cmd sera exécutée séquentiellement.

$ echo "Vous êtes le plus beau"; sleep 10; echo " des menteurs"

<commande 1> || <commande 2> : OR

N'exécute la seconde commande que si la première commande échoue.

Séquences de commandes (cont)

<commande 1> && <commande 2> : AND

Commande tee

Envoie en même temps stdout vers un fichier et vers l'écran.

tee [-a] *file* -a : append [CTRL]+c to exit

make | **tee** build.log
Lance make et stocke sa sortie dans build.log

make install | **tee** -a build.log
Lance la commande make install et rajoute sa sortie à la fin du fichier build.log

Commande yes

yes <string> | <command>
Remplit l'entrée standard de <command> avec <string> (y par défaut)

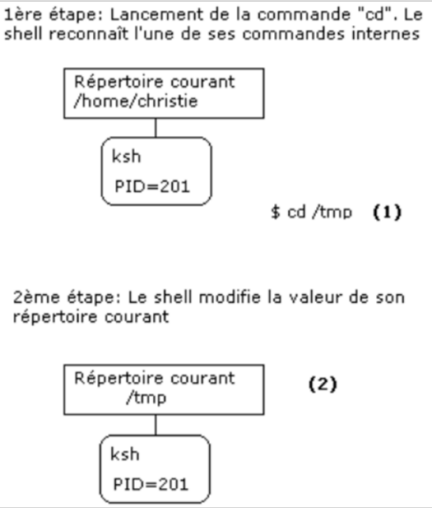
yes | rm -r dir/

yes *no* | credit_applicant

yes "" | make oldconfig

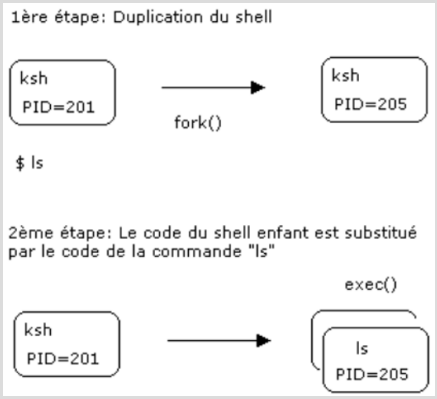
Equivaut à appuyer sur Entrer pour accepter les paramètres par défaut.

Commande Internes



-intégrée au processus shell
-ne correspond pas à un fichier sur le disque
-exécutée par le shell courant

Commande Externes



fichier localisé dans l'arborescence avec l'un des format suivants:
-binaire exécutable
-txt représentant un script (Shell, python...)

Commandes Internes et Externes

Certaines commandes ont une implémentation interne et externe. La **cmd interne** est lancée en prio car son exec est plus rapide.

pwd = cmd interne et externe

Pour forcer exec cmd externe, indiquer l'emplacement de la cmd ext.

Substitution de commande

Syntaxe: **$(cmd)**

Une commande **cmd** entourée par des **parenthèses** précédées d'un caractère **$** est :

1. exécutée par le shell
2. la chaine $(cmd) est remplacée par les résultats de la commande

Scripts

Langage interprété (shell, python...)

First ligne starts with **#!** + path interpretor

chmod transforme un fichier txt en exec

$ chmod a+x *script*

Contexte

GNU is **N**ot **U**nix: 1984 Richard Stallman projet système **à la Unix** entièrement libre.

Les logiciels libres offre ces 4 libertés:

-d'exéc soft, qu'elle que soit le but

-étudier fonctionn, et de l'adapt à ses needs

-redistribuer des copies pour aider autrui

-améliorer soft, et share ses améliorations

Licence **G**eneral **P**ublic **L**icence (soft libre):

toute modif soft libre reste soft libre

Linux: 1991 Linus Torvald. Noyau libre semblable au noyau Unix utilisant outils GNU =>Système complet = **GNU/Linux** (soft libre)

Noyaux = Kernel: cœur du système, fournit aux softs une interface de programmation pour utiliser le matériel.

Distrib Linux(ab de lang)=distrib GNU/Linux ensemble cohérent de logiciels assemblés autour du noyau Linux.
Distribution(ang)=collection de logiciels(fr) = système GNU/Linux.

Distrib comm.: assistance technique. Le code source libre, mais pas les binaires. (Red hat, Suse, Mandriva...)

Distrib commun.: sources + bin sont libres mais pas forcément d'assistance technique. (Fedora Core, Ubuntu Linux, Debian...)

Linux Cheat Sheet

by solsolsol via cheatography.com/32070/cs/10383/

Outils

Tout système UNIX inclut:

-compilateur C

-interpréteurs de commandes (shells)

-commandes de manipulation de fichiers

-Editeurs de texte

-Outils de dév (compilateur, débogueurs, analyseurs lexicaux et syntax...)

Shell

Interpréteur de commande en mode texte. Il peut s'utiliser en mode interactif ou pour exec des soft écrit en **shell scripts**.

Bash: Bourn Again SHell (standard) probablement le shell le plus utilisé avec **Ksh**

Bourn Shell (sh) : 1970 by Steve Bourne. Shell le plus ancien (maybe). Bash est issu d'innovations sur ce dernier.

Systèmes de gestion de fichiers

SGF=système de fichiers=FS =File System:

-Façon de stocker data et de les organiser dans des fichiers sur des mémoires second

-Permet de traiter, conserver data +share entre programmes informatiques

-Offre à l'user vue abstraite sur ses data + permet de les localiser à partir d'un path.

Concepts:

Systèmes de gestion de fichiers (cont)

-fichier: **niveau logique** (organisation visuelle dans le browser de fichiers)

-Enregistrement(sur périf): **niveau physique**: organisation sur le périphérique de stockage

Microsoft:	FAT, NTFS
Apple:	HFS
Linux:	ReiserFS, ext3, ext4

chaque fichier est caractérisé par un nom qui permet de repérer les data associées store sur un perif.

\n interdit dans le nom

Répertoires

Les fichiers sont organisés en répertoires et en sous-rep formant une arborescence.

Au moins 2 fichiers/répertoire: . et ..

(.) référence le répertoire lui meme.

(..) permet d'accéder au répertoire parent (^)

La racine de l'arbre est /

Structure de fichiers GNU/Linux

Peut varier d'un système à l'autre!

/	Répertoire racine
/bin	Commandes de base du système
/boot	Noyau et fichiers de configuration
/dev	Fichiers représentant des périph.
/etc	Fichiers de configuration du sys.
/home	Répertoire utilisteurs

Structure de fichiers GNU/Linux (cont)

/lib	Bibliothèques de base du système
/lost+found	Files détériorés que le sys. a essayé de récup.
/mnt	Sys. de fich montés `/mnt/usbdik/,...`
/proc	Accès info sys. `/proc/cpuinfo, /proc/version,...`
/root	Répertoire utilisateur de l'admin
/sbin	Commandes admin
/sys	Contrôle sys. et périph. (frq cpu...)
/tmp	Temp files
/usr	Users soft, non essentiels au sys.

Path

Un fichier qui commence par / est dit**absolu:** cd /home/user/sol/viande.txt

Un fichier qui ne commence pas par / est **relatif** et est interprété en partant du répertoire courant: cd ../../Bureau/boule.txt

Un nom de de fichier qui commence par ~ spécifie un chemin de type **user**. Interprété from curent user folder.

Si ~ suivi d'un username: Interprété en partant du répertoire perso de l'utilisateur spécifié.

~ correspond à /home/sol

cd ~ = cd /home/sol

Chaque user connu du sys. a un **home** directory = courant @ start bash.

cd sans argument nous envoie au **home** dir

By **solsolsol**
cheatography.com/solsolsol/

Not published yet.
Last updated 5th January, 2017.

Types de fichiers

Fichiers:

-contiennent des données

-UNIX ne fait aucune différence entre les fichiers de texte et les fichiers binaires

-dans un txt, les lignes sont séparés par **\n**

Répertoires: contiennent une liste de références à d'autres fichiers UNIX.

Fichiers spéciaux: associés à des pilotes de périphériques.

Tubes et sockets: utilisés pour la com. entre process.

Liens symboliques: fichiers "pointant" sur un autre fichier.

Droits d'accès

File users sont divisés en 3 ensembles:

–le propriétaire du fichier

-users du même groupe de travail que prop.

-les autres users ayant accès au système

User from one of those groups à accès ou non au fichier en: read(**r**), write(**w**), exec(**x**).

Ces droits (ou permissions) d'accès ne peuvent être changés que par le propriétaire du fichier, grâce à la commande **chmod**

$ ls -l pol.tex

```
-rwxr----- 1 sol users 67504 Mar 25 23:29 pol.tex
```

Le fichier **pol.tex**:

-contient 67504 char

-appartient à l'user sol

-fait partie du groupe users

Droits d'accès (cont)

-date et heure dernière modification

-rwxr----- :

-	type de fichier (fichier ordinaire)
rwx	droits du propriétaire du file
r--	droit users meme grp. que owner
---	droits des autres users

Liens symboliques

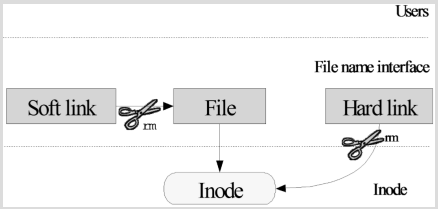
Fichier est une référence au nom d'un autre (fichier ou répertoire)

ln -s *file symbole* : crée un lien symbolique

ln -s *file1, file2 ...* Rep_dest_shortcut: crée un "raccourcis" des files dans le Rep_dest

rm *symbole* : ne del que le symbole

Liens physiques



-Par défaut, ln crée des liens physiques
-Les liens physiques sont indiscernables des fichiers d'origine
-Si fichier d'origine supprimé, cela n'affecte pas le contenu du lien physique
-Contenu est supprimé si il n'y a plus aucun fichier (lien physique) qui y fait référence

Alias

Raccourcis pour les commandes

alias ls='ls -la'

alias rm='rm -i'