

Commandes Linux

Le terminal est un programme qui ouvre une console dans une interface graphique, il permet de lancer des **commandes**. Il est parfois plus simple de taper une commande que d'effectuer des manipulations demandant beaucoup de clics de souris dans une interface graphique. Cependant, même si le terminal peut être plus efficace qu'une interface graphique aucune des deux méthodes ne remplacera complètement l'autre car elles se complètent plus qu'elles ne rivalisent.

Ouvrir un terminal : Depuis le *Tableau de bord* saisir terminal ou bien <CTRL><Alt><T>

Tapez une commande (ou copiez-collez la) et faites *Entrée* (clavier) pour l'exécuter.

Les raccourcis pour le copier-coller sont : *Ctrl+C* et *Ctrl+V*. Pour voir les raccourcis clavier : Lancer un Terminal, dans *Édition* → *Préférences* → *Raccourcis clavier pour afficher* les raccourcis.

Personnaliser son terminal : *Pour configurer l'interface* : Aller dans le menu *Édition* → *Préférences du profil*. Mettre les palettes prédéfinie sur *Personnalisée*, puis choisir vos couleurs pour le texte et pour l'arrière plan ainsi que la police et la taille des caractères

Complétion automatique : Après avoir tapé 2 ou 3 caractères, il est possible d'appuyer sur la touche *Tabulation* pour compléter la suite de la commande ou le nom du dossier ou du fichier. Cela évite les erreurs de frappe et est très appréciable pour éviter de saisir le nom de fichier parfois long et complexe comme le nom de certains paquets. Peut-être utilisé plusieurs fois si plusieurs noms ont une partie commune.

Une commande est une ligne de texte, par exemples : *ls* ou bien *sudo apt-get update*. Une commande est un programme exécutable comme les autres, par exemple pour ouvrir Firefox, vous pouvez taper : *firefox* dans le terminal. Des commandes peuvent demander le mot de passe pour être exécutées. Dans ce cas tapez le mot de passe demandé. Parfois une commande peut vous demander de répondre à une question. La plupart des commandes (comme *cd* , *mv* , *cp* , *ls* , *rm*, ... *commandes de gestion de fichiers et de dossiers*) s'exécutent en silence et ne répondent qu'en cas d'erreur.

Pour savoir ce que fait une commande, il suffit de taper : *man commande* pour lire le manuel d'utilisation de la commande ou bien *commande --help* pour afficher l'aide succincte de la commande. Par exemple: *man iconv* affiche le manuel d'utilisation de la commande *iconv*

Historique des commandes

Appuyer sur la touche ↑ (flèche haut) pour accéder à la commande précédente pour retrouver les commandes déjà exécutées. Vous pouvez appuyer plusieurs fois sur ↑ pour aller plus loin dans l'historique ou taper la commande *history*. Cette commande liste les 500 dernières commandes exécutées. Si vous souhaitez relancer la commande numéro 492, au lieu de la retaper, il suffit de lancer cette commande : *!492*

Il est également possible de vider cet historique en utilisant la commande suivante: *history -c* Il est possible aussi de ne supprimer qu'une ligne, pour cela utiliser l'option *-d* suivit du n° de ligne : *history -d 494* par exemple. Toutes les commandes exécutées sont enregistrées dans un fichier *.bash_history* dans votre dossier personnel ~ que vous pouvez manipuler à l'aide d'un éditeur de texte (*gedit*, *atom*, ...).

Commandes de base

Se déplacer dans un autre répertoire	<code>cd</code>
Évaluer l'espace disque occupé par des fichiers	<code>du</code>
Informations détaillées sur un fichier	<code>stat</code>
Lister les fichiers et dossiers	<code>ls</code>
Afficher le chemin vers le répertoire courant	<code>pwd</code>
Copier des fichiers	<code>cp</code>
Déplacer des fichiers	<code>mv</code>
Supprimer des fichiers	<code>rm</code>
Créer un dossier	<code>mkdir</code>
Supprimer un dossier vide	<code>rmdir</code>
Afficher un fichier	<code>more</code> ou <code>cat</code>
Rechercher des fichiers	<code>find</code>
Faire des recherches plein texte	<code>grep</code>
Compter les lignes, mots,... d'un fichier	<code>wc</code>
Remplacer des occurrences de texte dans un fichier	<code>sed</code> (voir le manuel de sed)
Ecrire un message a l'écran ou dans un fichier	<code>echo</code>
Trier un fichier text	<code>sort</code>
transformer le texte d'un fichier	<code>tr</code>
Afficher les processus en cours	<code>ps</code>
Tuer un processus	<code>kill PID</code> ou <code><CTRL><C></code>
Modifier les permissions d'accès aux fichiers/dossiers	<code>chmod</code>
Archiver/désarchiver des répertoires et des fichiers	<code>tar</code>
Extraire des fichiers d'une archive ZIP	<code>unzip</code>
Compresser/décompresser des fichiers avec l'extension .gz	<code>gzip</code>
Afficher la mémoire libre / utilisée du système	<code>free</code>
Afficher/modifier les partitions d'un disque dur	<code>fdisk</code>
Indiquer depuis quand le système a été mis en route	<code>uptime</code>
Créer un nouveau compte d'utilisateur	<code>adduser</code>
Supprimer un compte d'utilisateur	<code>deluser</code>
Exécuter des commandes en tant qu'un autre utilisateur	<code>sudo</code>
Modifier le mot de passe d'un utilisateur	<code>passwd</code>
Installer des paquets	<code>apt-get install</code>
Désinstaller des paquets	<code>apt-get remove</code>

- La commande **du** (*disk usage*) permet de connaître l'espace utilisé par des fichiers. Par exemple `du -sh rep` affiche la place mémoire en Mo occupé par le dossier rep.

- La commande *stat nom_fichier* affiche l'état du fichier : taille, permissions, dates de dernières modifications et d'accès.
- La commande *find* permet de retrouver des fichiers à partir de certains critères. La syntaxe est la suivante : *find <répertoire de recherche> <critères de recherche>*. Par exemple, pour chercher un fichier dont le nom contient la chaîne de caractères *toto* à partir du répertoire */usr*, vous devez taper : *find /usr -name toto*. D'autres critères sont disponibles pour cette commande.
- La commande *grep* recherche un mot dans un fichier et affiche les lignes dans lesquelles ce mot a été trouvé. La syntaxe simplifiée est : *grep texte nomfichier*. Par exemple *grep alias .bashrc* chercher le mot « alias » dans le fichier *.bashrc* et affiche toutes les lignes dans lesquelles le mot a été trouvé.
- La commande *sort* permet de trier les lignes d'un fichier texte. Par exemple *sort toto*
- La commande *ifconfig* affiche les paramètres réseau des interfaces (notamment le MAC du PC).

La commande grep cherche un motif dans un fichier et affiche les lignes dans lesquelles ce motif a été trouvé. Sa forme basique est : *grep texte nomfichier*. Ses options les plus simples sont :

- *i* : ne tient pas compte de la casse (majuscules / minuscules)
- *n* : affiche les numéros des lignes qui contiennent le texte cherché
- *v* : inverser la recherche : ignorer un mot, si vous voulez connaître toutes les lignes qui ne contiennent pas le texte
- *r* : recherche dans tous les fichiers et sous-dossiers.

Si vous ne savez pas dans quel fichier se trouve le texte que vous recherchez, utilisez l'option *-r* (*recursive*). Cette fois, il faudra indiquer en dernier paramètre le nom du répertoire dans lequel la recherche doit être faite.

La commande find retrouve des fichiers à partir de certains critères. La syntaxe est la suivante : *find <répertoire de recherche> <critères de recherche>*
Les critères de recherche sont les suivants :

- *name* : recherche sur le nom du fichier,
- *type* : recherche sur le type (d=répertoire, c=caractère, f=fichier normal),
- *size* : recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets),
- *atime* : recherche par date de dernier accès en lecture du fichier,
- *mtime* : recherche par date de dernière modification du fichier,
- *ctime* : recherche par date de création du fichier.

Les alias

Les alias sont des raccourcis de commandes trop longues. En effet, le terminal est très pratique mais les commandes sont parfois lourdes. Un alias permet de créer une commande courte pour une séquence que l'on tape fréquemment.

Comment créer des alias : A partir du terminal, taper : *alias votre_alias='commande_a_executer'* par exemple : *alias lister='ls -lsa'*. Lorsque vous saisissez *nom_de_votre_alias* dans un terminal, ce sera réellement la commande de votre alias qui sera exécutée.

Ou bien, éditer votre fichier *.bashrc* se trouvant dans votre *~* (avec *gedit* par exemple). Tout à la fin de votre fichier vous pouvez inscrire vos alias. Vous devez les écrire de la manière suivante : *alias votre_alias='commande_a_executer'*. Ce qui donne par exemple : *alias agu='sudo apt-get update'*. Une fois l'alias créé et le fichier enregistré, vous devez relancer une nouvelle console pour qu'il soit prit en compte.

Lister les alias : saisir la commande seule : *alias*

Supprimer un alias : saisir *unalias nom_alias*

Taches de fond. On peut lancer une commande en tâche de fond pour pouvoir garder la main pour lancer d'autres commandes depuis le terminal (sans quoi on ne peut plus lancer aucune commande jusqu'à la fin de l'exécution de la commande). Pour cela, faire suivre le nom de la commande par le symbole *&*. Par exemple la commande *atom &* lance l'éditeur atom en tache de fond.

Rediriger l'entrée/sortie standard. Quand on exécute une commande elle affiche le résultat sur la console de sortie (l'écran par défaut). On peut rediriger la sortie (le résultat de la commande) vers un fichier en utilisant le symbole *>*. Par exemple, *ls > fich* écrit la liste des fichiers et dossiers dans le fichier *fich*. Il est possible d'ajouter les sorties d'une commande à la fin d'un fichier existant en utilisant le double symbole *>>* Par exemple *ls -l >> fich*

Rediriger l'entrée standard permet d'entrer des données provenant d'un fichier au lieu du clavier et s'exprime à l'aide de *<*. Par exemple *tr a-z A-Z < fichier*. On peut combiner les deux redirections comme par exemple : *tr a-z A-Z < fichier1 > fichier2*

Tp1. Commandes Linux

- Lister le contenu du répertoire */etc*.
- Lister de manière détaillée le contenu du répertoire */etc*.
- Lister le contenu du répertoire */dev*.
- Afficher le contenu du fichier */etc/passwd*.
- Afficher le contenu du fichier */etc/shadow*.
- Copiez */etc/passwd* dans votre dossier personnel (*~*) sous le nom *toto*. Affichez les propriétés de *toto*. Quelle est la taille de *toto* ? Copiez *toto* dans votre *~* sous le nom *.tata*. Utilisez les commandes *ls* et *ls -a* pour lister le contenu de votre *~*. Créer un nouveau dossier *Brouillon* et y déplacer les fichiers *toto* et *.tata*. Vérifier à l'aide de *ls -a*. Se placer dans *Brouillon* et supprimer *.tata*. Revenir au dossier *~* et supprimer *toto*.
- A l'aide de la commande *echo* afficher à l'écran le texte: *to be or not to be*. A l'aide de *echo* et de l'opérateur *>*, créer un fichier *shakespeare* contenant le texte

précédent. Affichez le fichier `shakespeare` ensuite le supprimez. A l'aide de `echo` et de `>` écrire le texte `to be` dans un fichier `shake`. Afficher le contenu de `shake`. En utilisant l'opérateur `>>`, ajouter à la fin de `shake` la chaîne `or not to be`. Afficher le nouveau contenu de `shake`. Utiliser `echo` pour ajouter en fin de `shake` la ligne `Tp de septembre 2017`. Copiez ce fichier dans `Brouillon` ensuite supprimez le dossier `Brouillon`.

- A quoi servent les commandes `whoami` et `pwd` et `history` ?
- Faire un `alias` qui permet de voir les fichiers cachés et d'afficher les fichiers en couleur.
- Utiliser la commande `man` ou `help` pour pour comprendre la commande `passwd`.
Changer votre mot de passe (si vous avez la permission !)
- Utiliser `man` ou `help` pour pour comprendre la commande `touch` ensuite, modifier la date de l'un de vos anciens fichiers. Utiliser `ls -l` pour vérifier cette date.
- Créez , à l'aide de la commande `touch`, dans `Documents`, un fichier vide nommé `toto`. Utilisez la commande `man` ou `help` pour pour comprendre la commande `find` et l'utiliser pour retrouver le fichier `toto` à partir de votre `~`. Utilisez `find` pour retrouver le fichier `stdio.h` à partir de la racine /
- Utiliser la commande `man` ou `help` pour pour comprendre la commande `wc`. Afficher le nombre de lignes, de mots et de caractères du fichiers `sakespeare` et du fichier `/etc/passwd`
- Utilisez la commande `grep` pour afficher les lignes de l'un de vos fichiers `Processing` qui contiennent le mot `println`. Ensuite afficher les lignes de `/etc/passwd` qui contiennent `login`.
- Transformer `shake` selon la règle `toe 0@:` et ranger le résultat dans un fichier `shake_crypte`. Créer la commande `alias crypter` qui réalise la transformation `toe 0@:` sur le contenu d'un fichier. Ensuite `crypter` l'un de vos fichiers `Processing`
- Quelle est la taille totale des fichiers contenus dans votre `~`
- Interdire à l'utilisateur la permission de lecture pour le fichier `shake` (utiliser `chmod`). Essayer d'afficher `shake` avec `more`.
- Lancer l'éditeur à l'aide de la commande `gedit` & ensuite fermer l'éditeur à l'aide de la commande `kill`.
- Modifier le prompt (`PS1`) pour qu'il affiche : `votre_identifiant:l'heure :.`
Redonner à `PS1` sa valeur initiale.
- Ouvrir votre fichier `.bash_history` dans sous l'éditeur `gedit` et en extraire le corrigé de ce TP dans un fichier `corrige_tp1`, y ajouter vos Noms Prénoms.