Utiliser le terminal sous MacOS 10.11

Kevin Cazelles

3 Mai 2016

Résumé

Le terminal est une fenêtre d'invite de commande, c'est-à-dire une interface graphique doté d'une interprétateur de commande, qui propose une communication avec l'ordinateur. Pour y communiquer avec l'ordinateur il est alors nécessaire de connaître les instructions à lui donner, c'est-à-dire les lignes de commande à utiliser. Je recense ici un certain nombre de ces lignes commandes, les plus usitées. Elles sont très utiles pour naviguer dans les différents répertoires de son ordinateur, créer des fichiers, les supprimer... Je présente également des installateurs de logiciels libres en ligne de commande et certains de ces logiciels libres qui peuvent être utilisées directement dans le terminal. La première version de ce document date du 20 avril 2013, j'utilisais alors MacOS 10.8 (Mountain Lion). La dernière version du document date du 18 octobre 2015 et mon système d'exploitation est MacOS 10.11 (El Capitan).

Table des matières

Avant-propos **3**
0.1 Avertissement . . . 3
0.2 Mes sources . . . 3
0.3 Mes Abréviations . . . 3

1 Le terminal et l'application *Terminal* **5**
1.1 Sémantique . . . 5
1.2 Pourquoi utiliser le terminal . . . 6
1.3 Raccourcis clavier (par défaut) de l'application terminal . . . 6
1.4 Les racourcis de l'application terminal . . . 6
1.4.1 Gestion des fenêtres . . . 6
1.4.2 Naviguer dans la console . . . 6
1.4.3 Naviguer au sein d'une ligne de commande . . . 7
1.4.4 Ecrire efficacement une ligne de commande . . . 7
1.4.5 Executer/arrêter une ligne de commande . . . 7

2 Le langage Bash, pour dialoguer avec son ordinateur **8**
2.1 Quelques considérations historiques . . . 8
2.2 Les commandes du langage Bash . . . 8
2.3 S'informer d'une commande . . . 9

3 Les commandes basiques du *bash* **10**
3.1 Naviguer à travers les différents répertoires . . . 10
3.1.1 Chemins relatifs/ Chemins absolu . . . 10
3.1.2 Connaitre le répertoire courant et en changer . . . 10
3.1.3 Afficher le comtenu d'un répertoire . . . 10
3.2 Créer des fichiers et des dossiers et les manipuler . . . 11
3.2.1 Création . . . 11
3.2.2 Copie . . . 11
3.2.3 Déplcer des fichier . . . 11
3.2.4 Supprimer un fichier . . . 11
3.2.5 Ourvir un fichier . . . 12
3.2.6 Gérer les droits . . . 12
3.3 Gérer l'ordinateur . . . 12
3.3.1 La date . . . 13
3.3.2 Veille de l'ordinateur et suspensions des executions . . . 13
3.3.3 Visualiser l'acitivité de son ordinateur . . . 13
3.3.4 Disques et clef USB . . . 14
3.3.5 Réseaux . . . 14
3.3.6 Astuce Apple . . . 14
3.3.7 Créer une archive : . . . 15
3.3.8 Connaître son Hardware : . . . 15

3.3.9 Fichiers cachés . . . 15
3.4 Recherche de fichiers . . . 15
3.4.1 Les expressions régulières . . . 15
3.4.2 Chercher un fichier . . . 16

4 Les éditeurs de texte **17**
4.1 Vim / emacs / pico . . . 17

5 Faire des scripts Bash **18**

6 Améliorer l'utilisation du Bash **19**
6.1 Les fichiers inputrc et bash_profile . . . 19
6.2 Créer un alias . . . 19

7 Connection à distance avec le terminal **20**
7.1 Les protcoles FTP et SSH . . . 20
7.2 Commandes SSH . . . 20
7.3 Commandes FTP . . . 20

8 Les gestionnaires de paquets sous MacOS **21**
8.1 MacPorts . . . 21
8.1.1 Installation de MacPorts . . . 21
8.1.2 Utiliser MacPorts . . . 21
8.2 Homebrew . . . 22
8.2.1 Installation et désinstallation . . . 22
8.2.2 Commande principales . . . 23

9 Compiler en ligne de commandes **24**
9.1 Compilation C/C++/Fortran . . . 24
9.1.1 Valgrind . . . 24
9.2 Compilation Latex . . . 24

10 Quelques logiciels libres et des commandes bien utiles **25**
10.1 ffmpeg . . . 25
10.2 ImageMagick . . . 25
10.3 Pandoc . . . 25
10.4 R, Python et Julia . . . 25
10.5 Versionning avec Git . . . 25
10.6 Postgresql . . . 25
10.7 Jekyll . . . 25

Avant-propos

0.1 Avertissement

L'ensemble de ce qui suit est le fruit de nombreuses heures d'apprentissage sur Internet. En cherchant des solutions à des problèmes concrets pour réaliser différentes opérations, j'ai accumulé de nombreuse astuces qui facilitent aujourd'hui mon travail. Ne trouvant pas une telle compilation déjà toute écrite (ce qui n'est pas étonné car une combinaison d'utilisation est spécifique), j'ai décidé de les rassembler dans un même document. Pour tout ce qui est informatique, je suis autodidacte et je n'ai certainement pas des bases théoriques suffisantes. Cela dit, je communique quotidiennement avec mon ordinateur sous forme de lignes de commande. Ainsi, le document présent est le résultat d'une approche très pragmatique et égo-centrée du terminal. Néanmoins, je suis persuadé que beaucoup de matériel ici présenté peut-être utile à beaucoup d'utilisateur même occasionnel du terminal.

0.2 Mes sources

Il y a beaucoup de source d'information sur Internet mais je n'ai pas trouvé une complilation d'usage multiple du terminal, alors je l'ai fait, au moins pour moi. Voici quelques ressources qui vous seront, je l'espère, utiles :

- Généralités
- Terminal informatique
- Shell Unix
- Base du Terminal
  - graffitix
- Liste des commandes
  - ss64
- Astuces à la volée
  - maclife
  - A vos mac (il faut être abonné)
  - OSX daily
- Sites généralistes
  - Regardez sur openclassroom (ancien site du zéro)
  - Beaucoup d'astuces sur Comment ça marche (pour Linux mais beaucoup sont directement utilisables)
- Avancé
  - Faire des scripts bash
  - Le Bash est un langage commun aux systémes UNIX qui a une documentation officielle.

0.3 Mes Abréviations

Tout au long du document, j'emploie plusieurs abbréviations:

- Rq : Remarque
- NB : Nota Bene
- Ex : Exemple
- ctrl : touche « ctrl », touche « contrôle »
- alt : touche « alt » aussi appelée touche option.
- cmd : touche « cmd », « commande »
- fn : touche « fn », touche « fonction »
- tab : touche tabulation
- espace : la barre d'espace
- entr : la touche entrée

— « adr » désigne un chemin et « adr/fichier1 », le chemin jusqu'au fichier *fichier1* ; de même « adr/dossier » désigne le chemin pour accéder à un dossier précis.
— CPU : *Central Processing Unit*, unité centrale de traitement.
— GPU : *Grpahics Processing Unit*, processeur graphique.

Chapitre 1

Le terminal et l'application *Terminal*

1.1 Sémantique

Le mot terminal fait référence au 'terminal informatique' qui désigne la partie d'un réseau informatique avec lequel un humain peut communiquer. Cette communication consiste à lui faire exécuter une ou plusieurs opérations. Aujourd'hui, pour la plupart des utilisateurs, le travail avec l'ordinateur peut se faire de manière intuitive avec l'utilisation d'une interface graphique, sans utiliser de ligne de commande. En raison de l'utilisation de ligne de commande, les fenêtres d'invite de commandes sont appelées 'terminal'.

De manière générale, pour un langage donné, une commande est un ensemble de charactères interprété par l'ordinateur et qui induit l'exécution d'une tâche corespondante par celui-ci (si la commande est correctement entrée). Ces commandes peuvent être agencées par l'utilsateur pour mener à bien un ensemble d'opération générant des suite de cacractères plus ou oins complexe. Pour l'utilsateur novice qui utilise n'a pas l'habitude de l'emploie de ligne de commande, les suites de clics (ou de raccourcis), sont remplacées par des commandes qu'il faut connaître (ou savoir retrouvées). Dialoguer avec l'ordinateur demande d'avoir un langage commun. Par défautlt la fenêtre s'ouvre et c'est le langage Bash qui est utilisé.

L'application terminal sous MacOSX L'application *Terminal* est une console pour de nombreux langages de programmation. Par défaut, l'application lance le langage bash (Bourne-Again shell). On peut remarquer ces 4 lettres dans la barre de titre. Ce langage permet d'interagir avec son ordinateur à l'aide de lignes de commande. L'utilisation du bash est le premier aspect du Terminal développé plus bas. L'aspect minimaliste et un peu ésotérique d'un terminal dissimule des possibilités immenses d'utilisation de son ordinateur. Concrètement, au lieu d'utiliser diverses interfaces graphiques pour différentes applications il est possible, au prix d'un effort d'apprentissage, d'utiliser juste le *Terminal*. Utiliser le *Terminal* pour regarder des photos, écouter de la musique ou naviguer sur Internet n'est pas le plus évident. C'est tout de même possible et dans certains cas, cela peut être d'une efficacité redoutable.

L'application *Terminal* est une interface graphique possible et native sous MacOSX, il en existe une seconde très poplaire : iTerm2.

Xterm

Sous Linix Le terminal GNU/Linux

https://doc.ubuntu-fr.org/terminal

Utilisez l'une ou l'autre est, éà mon sens, une question de préférence en terme de faciliter dMutilisatuoib de design. Quand on utilise une langage dans une fenêtre de terminal ce n'est pas le choix du terminal qui change grand chose.

1.2 Pourquoi utiliser le terminal

— on peut faire la plupart (même toutes) des actions sans, alors pourquoi ?
— tout faire en dans une même fenêtre,
— améliorer son workflow /
— les emballages bien que utiles consomme des ressources.
— parfois des actions qu'on crois possible qu'avec certains logiciels payants sont possibke pas avec des freware en ligne de commandes
— ça ouvre un monde de software puissant qui ne sont pas toujours dans un emballage GUI améliorer notre literacy
— invite à mieux conprendre son ordinateur

1.3 Raccourcis clavier (par défaut) de l'application terminal

La configuration des raccourcis peut se faire via l'édition d'un ficher '.inputrc' dans le dossier utilisateur (à créer).

Par déafault on a Ces raccourcis sont valides pour le bash (le langage pas défaut), mais aussi pour différentes applications. Cependant, ils peuvent aussi être complètement occultés dans certains modes du terminal. Ainsi, si vous utilisez l'éditeur de text *nano* les raccourcis ne sont plus valides (il y en a d'autres).

1.4 Les racourcis de l'application terminal

Il s'agit des raccourcis clavier par défault de MacOS 10.11.4 pour cette application. Vous les retrouverez dans le menu de la console, je liste ceux que je pense les plus utiles. Les raccourcis avec les cmd reste valides quel que soit l'utilisation de la console. Ce n'est pas le cas pour ceux qui utilisent la touche **ctrl** qui dépende de l'utilisation ils sont valides au quand la console est utiliser pour entrer des commandes Bash (mais parfois dans d'autres situations aussi).

1.4.1 Gestion des fenêtres

— Créer une nouvelle fenêtre : **cmd + N**

— Créer un nouvel onglet : **cmd + T**
— Diviser la fenêtre du terminal : **cmd + D**
— Annuler la division : **cmd + shift + D**
— Efface le contenu d'un onglet : **cmd+K** (ou **ctrl+L**, ou utiliser la commande **clear**)
— Effacer la dernière ligne de commande et le(s) résultat(s) associé(s) : **cmd+L**
— Faire une recherche dans le contenu de la console : **cmd+F**
— Sauver le conteniu d'un onglet sous forme de text : **cmd+S**
— Imprimer le contenu de la console : **cmd+P**

1.4.2 Naviguer dans la console

En utilisant le terminal, il est fréquent d'accumuler bien plus de lignes dépassant largement la hauteur de la fenêtre du terminal, afin de naviguer dans le contenu d'un onglet on peut utiliser la barre de défilement, mais aussi :

— Remonter à la première ligne : **fn + flèche de gauche**
— Descendre à la dernière ligne : **fn + flèche de gauche**
— Remonter d'une page : **fn + flèche de gauche**
— Descendre d'une page : **fn + flèche de gauche**

1.4.3 Naviguer au sein d'une ligne de commande

— Avancer d'un symbole : **flèche de droite** (ou **ctrl+p**)
— Reculer d'un symbole : **flèche de gauche** (ou **ctrl+b**)
— Avancer d'un symbole : **alt + flèche de droite**
— Avancer d'un symbole : **alt + flèche de droite**
— Aller au début d'une ligne : **ctrl + A**

— Aller à la fin d'une ligne : **ctrl + E**

— Aller d'un bout de la ligne à l'autre : **ctrl + X** (2 fois)

1.4.4 Ecrire efficacement une ligne de commande

— Auto-complétion: **tab** ; l'auto-complétion est l'action de complèter un mot / une commande dont on a entré les premiers charactères.
— Si l'auto-complétion ne fonctionne pas deux situations sont à envisager :

1. il n'y a pas de correspondance avec un nom de fichier / de commande / . . . existant,
2. plusieurs choix sont disponibles, dans ce cas, en utilisant **tab** de nouveau, on affiche la liste des possibilités.

— Effacer le caractère à gauche du curseur : **backspace** (le classique) ou **ctrl + H**
— Effacer le caractère à droite du curseur : **ctrl + D** attention si utilisé sur une ligne vite, entraîne la déconnection)
— Effacer tout ce qui se trouve à gauche du curseur : **ctrl + U**
— Effacer tout ce qui se trouve à droite du curseur : **ctrl + K**
— Effacer le mot juste à gauche du curseur : **ctrl + W**
— Inverser les deux dernières lettres : **ctrl + T**
— Coller la saisie précédente : **ctrl + Y**
— Rappeler la dernière commande : **flèche du haut** (ou **ctrl+P**)
— Revenir vers une commande plus récente : **flèche du bas** (ou **ctrl+N**)
— Recherche dans l'historique : **ctrl + R**, il suffit alors d'entrer les premières charactère de la ligne ayant été utilisée précedemment.

1.4.5 Executer/arrêter une ligne de commande

— Exécuter une commande : écrire la commande puis **entr** (ou **ctrl+J** ou)
— Annuler l'exécution d'une commande : **ctrl + C**
— Stopper une tâche : **ctrl + Z**
— Déconnection de la session terminal: **ctrl + D** (sur une ligne vide)

NB : Pour ne pas entrer manuellement un chemin, on peut faire glisser le dossier ou le fichier concerné, son chemin apparaîtra alors.

Chapitre 2

Le langage Bash, pour dialoguer avec son ordinateur

2.1 Quelques considérations historiques

Quand on ouvre l'application *Terminal*, nous avons une fenêtre de dans laquelle l'utilisateur peut communiquer avec lordinateur en utilsant ls langage Bash.

2.2 Les commandes du langage Bash

Pour le Bash, les commandes sont une suite de charactères qui forme le nom de la commande suivi éventuellement d'une ou plusieurs options reconnaissable par un tiret ("-") et une ou ou deux lettres puis éventuellement des paramètres suivis éventuellement de l'objet qu'elle concerne (souvent un fichier représenter par son chemin) : nomcommande -option1 -option2 paramètre objet (il suffit alors de taper **enter** ou **cmd + M**). Les options peuvent être mises les unes à la suite des autres -option1option2.

Commencons par une commande seule :

```
pwd
```

Une fois entré, nous avons un résultat associé qui est le chemin absolu du répertoire courant. Nous entrons maintenant une nouvelle commande avec un option :

```
ls -a
```

Cette commande liste le contenu du répertoire courant. L'option -a permet de voir tous les fichiers, même les fichiers cachés. Nous allons maitenant ajouter un chemin pour lequel nous aimerions avoir la liste des fichiers:

```
ls -a ~/Documents
```

Enfin, une nouvelle commande poue laquelle je rajoute une commande qui nécessite un paramètre

```
mkfile -nv 100k ~/Desktop/exemple1.txt
```

Crée un fichier avec deux options *-n* et *-v* (raccourcis en *-nv*), le paramètre 100k (taille du fichier) et le nom du fichier *~/Desktop/exemple1.txt*, L'adresse d'un fichier qui va être créé.

— Une commande est en fait un exécutable dont on peut facilement trouver l'adresse : **which + nom_commande**, une recherche large est possible avec : **locate + nom_commande**

— Les commandes ne sont pas sensible à la casse. ex : ls = LS = lS = Ls. *Attention!!* ce n'est pas le cas pour les options.
— Pour tout renseignement relatif è une commande (notamment pour en connaître les options) : **man + nom_commande** (**q** pour quitter le manuel), celui-ci décrit la fonction de la commande. Pour savoir comment faire défiler l'écran, une fois dans dans le mode manuel, il faut entrer h (en bref on peut utiliser les flèches, les flêches avec **fn** (ou encore **b** et **z** ou **espace** également).
— Parfois un droit d'administrateur est requis pour certaines actions, typiquement lorsqu'un message "permission denied" apparaît suite à l'entrée de la commande. Pour pouvoir exécuter la commande il faut avoir les droits d'administrateurs et faire précéder la commande de : **sudo** (« super user do »).Le mot de passe associé au compte administrateur est alors demandé.

2.3 S'informer d'une commande

```
man nomcommande
```

Les commentaires sont précédents '#'

Chapitre 3

Les commandes basiques du *bash*

Dans cette section, je présente les commandes fréquemnemt utilisées pour faire les opérations de base. Pour illuster he cocidérauaire des dossiers 'dossier'

3.1 Naviguer à travers les différents répertoires

3.1.1 Chemins relatifs/ Chemins absolu

répertoire courant

3.1.2 Connaitre le répertoire courant et en changer

— Connaître le répertoire actuel :

```
pwd
```

— Changer de répertoire :

```
cd /dossiera/dossierb/dossierC/file1.txt
```

— quelques cas particuliers :

```
cd /  # aller à la raison
cd ~  # aller au dossier utilisateur
cd .. #aller au dossier supérieur
```

3.1.3 Afficher le comtenu d'un répertoire

— Lister les fichiers d'un dossier :

```
ls adr
```

si aucun nom de dossier n'est donné, ce sont les fichiers du répertoire courant qui sont listés. Option *-a* pour voir les fichiers cachés et option *-l* pour obtenir des informations sur les fichiers (date de modification et qui est propriétaire).

```
ls adr -alth
```

— Connaître la taille des fichiers

```
du adr
```

option *-s* pour faire la somme de la taille des fichiers d'un dossier et option *-h* pour avoir des nombres en unité facile à lire. ex :

```
du -sh ~/Documents
```

```
wc
```

3.2 Créer des fichiers et des dossiers et les manipuler

3.2.1 Création

— Créer un nouveau fichier :

```
mkfile adr/nom_fichier
```

— On peut préciser la taille du fichier ex: > mkfile 140k truc.txt
— Créer un nouveau dossier :

mkdir + adr/nom_dossier

3.2.2 Copie

— Copier un fichier :

cp adr1/fichier1 adr2/fichier2

Si fichier2 n'est pas préciser, fichier1 est utilisé comme nom de fichier.

— Copier un dossier :

cp -R adr1/dossier1 adr2/dossier2

3.2.3 Déplcer des fichier

— Déplacer un fichier ou un dossier :

mv adr1/fichier1 adr2/fichier2

3.2.4 Supprimer un fichier

— Suppimer un fichier :

rm +adr/fichier

— Suppimer un dossier :

rm -R adr/dossier (option f pour forcer) rmdir adr/dossier

— Vider la corbeille :

rm -r ~ /.Trash/*

3.2.5 Ourvir un fichier

— Ouvrir un fichier :

open + adr/fichier

Pour visualiser un fichier en lecture seule on utilise la commande less :

option *-a* permet de spécifier l'application avec laquelle pouvrir le fichier, ex: open -a Finder /)

3.2.6 Gérer les droits

— Changer le propriétaire d'un fichier :

chown owner[:group] adr/fichier

Défini à qui appartient le fichier.

— Changer les droits d'accès d'un fichier :

chmod mode adr/fichier

Les modes permettent de définir les actions possibles sur un fichiers pour tous types d'utilisateurs, propriétaire ou non du fichier. Les modes s'écrivent sous forme d'un nombre précisant les droits des utilisateurs sur un fichier.

— Imprimer un fichier :

lpr adr/fichier

Attention, tous les types de fichiers ne sont pas supportés.

3.3 Gérer l'ordinateur

— Eteindre :

sudo shutdown -h now

— Eteindre dans 30 minutes :

sudo shutdown -h +30

— Eteindre le 06/12/2015 à 15h30 :

sudo shutdown -h 1512061530

— Redémarrer :

```
sudo shutdown -r now
# plus simplement :
sudo reboot
```

— Relancer le Finder :

sudo killall Finder

3.3.1 La date

— Obtenir la date :

date

— Changer la date :

date 0613162785

— Changer la date :

date 1758

3.3.2 Veille de l'ordinateur et suspensions des executions

— Laisser l'ordinateur en activité (pas de veille) :

```
caffeinate
```

option *-d* pour que l'écran ne se mette pas en veille, option *-t* pour choisir le temps (en seconde). Ex : caffeinate -dt 600 => permet de maintenir l'ordinateur mais aussi l'écran en activité pendant 10 minutes.

— Suspendre les exécutions pendant une durée « tps » exprimé en secondes :

```
sleep + tps
```

Ex : si j'utilise "sleep 10" et que j'essaye ensuite d'entrer la commande ls, je devrais attendre la fin des des 10 secondes pour que la commande soit exécutée.

3.3.3 Visualiser l'acitivité de son ordinateur

— Visualiser l'activité :

```
top
```

Cette commande affiche un grand tableau, la table des processus (Table Of Processes). Un processus est une tâche entreprise par l'ordinateur qui est répertorié tant qu'elle a court. Parmi les colonnes, on a : 1. **PID** : le numéro d'identité du processus 2. **COMMAND** : le nom du processus 3. **%CPU** : le pourcentage CPU utilisé, il s'agit d'une mesure unstantannée de la charge de travail du processeur (on a jusqu'àn*100% de CPU disponible ou n est le nombre de coeur.). 4. **%TIME** : la durée d'activité du processus 5. **MEM** : la mémoire utilisée par le processus

Une fois que l'on affiche le tableau, le terminal fonctionne un peu différemment. Pour connaître ce qu'on peut faire, il suffit d'entrer **?**. Une action pratique est de trier le tableau, il fa=ut alors taper **o** puis le nom de la colonne, par exemple **cpu** pour afficher par utilisation de cpu et **cpu** pour trier par quantité de mémoire utilisée. Pour quitter le tableau, entrer **q**.

— Arrêter une tâche :

kill + Pid

Le Pid est le numéro d'indentité donné par le taleau top.

kill -STOP Pid kill -COUNT Pid

kill

— Connaître son «load average » :

uptime

Le load average désigne, sous les systèmes UNIX, une moyenne de la charge système, une mesure de la quantité de travail que fait le système durant la période considérée. (cf. Wikipedia)

- Exécuter des tâches en arrière plan :
- Chaque commande peut être exécutée en arrière plan, pour cela on ajoute & à la fin de celle-ci. L'identité « pid » du processus est alors donné. Il s'agit d'un numéro de tâche qui s'inscrivent à chauqe nouvelle tâche et cela, depuis que vous avez redémarrez votre ordinateur. Ce numéro est accessible grâce à la commande top, pour le supprimer : kill pid
- Ex : sleep 10& (cette fois vous pourrez exécuter ls !)
- Pour faire revenir le processus au premier plan : fg %njob (premier numéro sur la gauche donné par la commande jobs)
- Un processus stoppé peut être passé à l'arrière plan : bg %njob
- Avec top, on peut avoir accès aux identités des processus : kill pid, supprime le processus ; kill -stop pid le met en pause, kill -cont pid le réactive.

Ajouter ctrl Z / bg / fg pour un stopper et reprendre un processus

3.3.4 Disques et clef USB

- Forcer l'éjection d'un cd : > sudo drutil eject
- Liste des volumes montés et leurs caractéristiques : df (-h «human »pour avoir des préfixes connus, *pour l'ensemble des dossiers d'un dossier (avec un espace) ! ex : du -sh ~*)

3.3.5 Réseaux

- Afficher l'ensemble des disques utilisés :

  diskutil list

- Visualiser/configurer les réseaux :

  ifconfig

- Obtenir son adresse IP :

  ipconfig getifaddr en0

ou en1 si « en2 » dans ifconfig existe

- Regarder l'état des réseaux que l'on utilise :

  netstat netstat -A

3.3.6 Astuce Apple

- Changer le type du fichier de capture d'écran (.png par défaut):

  defaults write com.apple.screencapture type PDF (ou jpg ou png)

- Afficher les fichiers cachés defaults :

  write com.apple.finder AppleShowAllFiles 1

Remplacer le 1 par 0 pour les cacher, pour que la manipulation soit effective, il faut relancer le Finder.

— Reconstruire reconstruire l'index du spotlight :

sudo mdutil -E /

— Reconstruire reconstruire l'index du spotlight d'un volume :

sudo mdutil -E /Volumes/[DriveName]

3.3.7 Créer une archive :

— Créer une archive tar : tar adr/fichier (beaucoup de possibilité, ajouter dans une archive. . . , à développer)
— Créer une archive zip : gzip adr/fichier (option -k pour garder la copie non zippée)
— Dezipper une archive zip : gunzip adr/fichier

3.3.8 Connaître son Hardware :

— system_profiler SPHardwareDataType
— sysctl hw
— sysctl machdep.cpu

3.3.9 Fichiers cachés

— Rq : Pour créer rapidement un fichier caché, il suffit de faire commencer son nom par un « . ».
— Faire d'un fichier caché, un fichier apparent : sudo chflags nohidden nomdufichier (option -R pour un dossier)

3.4 Recherche de fichiers

Nous sommes souvent en train de rechercher des fichiers et bien que « Spotlight » soit souvent d'une grande aide il est parfois difficile d'avoir exactement ce que l'on veut. Pour des recherches très efficace, il existe des commandes bash d'une très grande efficacité mais qui demandent un peu d'entrainement !

3.4.1 Les expressions régulières

grep grep -rwl niche ~/Desktop/interactionLS/ options : -r dans le subfolder -w le mot entier -l liste de fichier -v l'inverse de ce qui est chercher grep .tex grep ˆsep.r.te /usr/share/dict/words

— Recherche dans l'historique : history | grep gcc
— Liste des expressions régulières : ———————— ˆ debut $ fin ————————
— plusieurs fois le caractère précédent (au moins une fois)
— zéro ou plusieurs fois le caractère précédent ? zéro ou une seule fois le caractère précédent {x} x fois le caractère précédent {x,} x fois ou plus le caractère précédent {i,j} minimum i fois, maximum j fois le caractère précédent ———————— ( ) ces règles peuvent s'appliquer sur un ensemble de caractères/expression entre parenthèse ———————— | ou . n'importe quel caractère une seule fois ———————— [ ] liste de caractères autorisés sur lesquels peuvent s'appliquer les règles précédentes [ˆ] liste de caractères interdits
— permet de définir des intervalles [a-z][:alpha:] ou [a-zA-Z][:digit:] ou [0-9][:album:] ou [a-zA-Z0-9][:blank:] [:punct:] caractère de ponctuation ————————

3.4.2 Chercher un fichier

find ~/ -iname ‘evol’ find ~/Desktop -iname ‘*.txt’

— Trouver les différences entre des fichiers :

vimdiff file1 file2 file3

— Réunir des textes en un seul :

cat sample1.txt sample2.txt sample3.txt > sample-all.txt

— metadata search : > mdfind word1 word2 word3

ex : mdfind wavelet ecology markov test fourier royal society chavez using or (royal or society)

mdfind wavelet ecology markov test fourier (royal | society) mdfind -onlyin ~/Movies couleur

Chapitre 4

Les éditeurs de texte

4.1 Vim / emacs / pico

Chapitre 5

Faire des scripts Bash

Chapitre 6

Améliorer l'utilisation du Bash

6.1 Les fichiers inputrc et bash_profile

6.2 Créer un alias

— Lorsqu'on utilise à répétition une commande, il peut être intéressant de créer un alias. Par exemple, j'ai souvent besoin d'aller dans mon dossier de thèse : cd ~/Documents/PHD, j'aimerais simplement taper «phd», pour cela : alias php 'cd ~/Documents/PHD'

— De manière générale le principe est le suivant : alias +nom_de_la_commade_désirée + 'la commande entre guillemets'

— La procédure précédente permet seulement d'avoir une commande pendant la session, pour la garder en mémoire il est nécessaire de l'enregistrer dans le fichier «./bash_profile » dans le dossier utilisateur (~). Une procédure possible est alors : 1- cd ~ 2- nano ~/bash_profile 3- on ajoute la ligne : alias php 'cd ~/Documents/PHD' 4- on enregistre et voila ! Il faudra alors soit rouvrir un onglet, relancer le terminal ou entrer : source ~/bash_profile pour que ce soit ok

— Si par hasard, on souhaite créer un alias qui a un nom qui existe déjà (typiquement pour ne pas rentrer une option à chaque fois), l'antislash « » rétabli la commande par défaut. Ex : si j'utilise très fréquemment ls -la tout le temps => je peux alors créer l'alias : alias ls 'ls -la' et alors pour avoir le ls d'origine => la commande : \ls est alors le ls par défaut.

Chapitre 7

Connection à distance avec le terminal

7.1 Les protcoles FTP et SSH

```
ftp
```

=> ncftp

7.2 Commandes SSH

```
ssh
```

7.3 Commandes FTP

```
scp
```

```
screen
```

```
screen -d
```

voir

Chapitre 8

Les gestionnaires de paquets sous MacOS

Il s'agit d'application qui facilitent l'installation de logiciels libres. Ces différents logiciels sont installés facilement en quelques lignes de commandes. C'est une pratique qui existe par défaut dans les différentes distributions de Linux (apt-get, yum,...) mais qui doit être installé pour MacOS. Il existe plusieurs gestionnaires de paquets : Fink, MacPort et Homebrew. Je présente MacPort et Homebrew dans les sections qui suivent. Je n'utilise plus MacPort depuis mi-2014, je suis passé à Homebrew, il se peut donc que certaines indications ne soient pas correctes. Une installtion propore + GEM + apm + Python

8.1 MacPorts

Avec MacPorts, tout ce que vous demandez sera installé dans le répertoire : /opt/local. Pour voir la liste des exécutables (les logiciels libres) qui seront appelés par une commande le plus souvent de ce même nom : ls /opt/local/bin

8.1.1 Installation de MacPorts

- Il faut commencer par l'installer : http://www.macports.org/
- Toutes (très très complet!) les indications sont à l'adresse suivant : http://guide.macports.org/
- Pour apprendre : http://fr.openclassrooms.com/informatique/cours/utilisez-macports
- NB : Il faut veiller à avoir la dernière version de « Xcode » (ça change entre les OS ! télécharger toujours la dernière version de Xcode adaptée à l'OS) : puis installer le guide d'outil du développeur avec la commande : xcode-select –install

8.1.2 Utiliser MacPorts

- Pour entrer en mode actif : port
- Pour entrer avec les droits admin : sudo port
- Sinon on met port (ou sudo port) en début de commande (on entre pas dans le mode actif) (option installed, uninstalled)
- Voir la liste des packages : list
- Chercher avec un mot clef : search + mot
- Des infos sur un port : info + nom du port
- Les dépendances d'un port : info + nom du port
- Installer des packages/ mises à jour :
- Attention pour pouvoir installer un package, il faut avoir les droits d'administrateur et donc il est nécessaire de se connecter en « sudo port »

- Installer un package : install +nom package
- Désinstaller un package : uninstall +nom package (option —follow-dependents, pour désinstaller les dépendances)
- Mise à jour de la liste de package : sync
- Port pas à jour : port outdated (on peut ajouter list pour un défilement : port list outdated) /etc/macports/sources.conf,
- Mise à jour de la liste et des packages installés : selfupdate
- NB : si le message d'erreur suivant apparaît : Error: Error synchronizing MacPorts sources: command execution failed cela est vraisemblablement du à votre connexion (cela m'arrive quand je suis au labratoire!).
- Mise à jour d'un seul package : upgrade + nom package
- Mise à jour des ports pas à jour : sudo port upgrade outdated
- Chercher port qui dépendent d'un nom donné : port echo depends:nompackage
- Cleaning package (je suis pas bien capable de vous dire ce que ça fait) : port clean — all installed ; port clean — dist
- Par défaut les versions anciennes, inactives (donc plus requises) ne sont pas enlever, pour le faire : sudo port -v uninstall inactive
- J'ai eu pas mal de problème pour synchroniser il semble que parfois on rencontre des problèmes avec Xcode (encore une fois vérifier que vous avez bien la dernière version, la solution est à nouveau sur le site de Macports)...
- Pour régler le problème de synchronisation que j'ai rencontré, il suffit de ne pas prendre la voie courante, on peut alors aller dans le fichier « mdfind source.conf » (où est-il ? => mdfind -name sources.conf), il faut commenter la dernière ligne «rsync ... » et ajouter en-dessous: https://distfiles.macports.org/ports.tar.gz //To use the rsyncd server you must copy /opt/local/etc/rsyncd.conf.example to rsyncd.conf and add your modules there. //See 'man rsyncd.conf' for more information
- Désinstallation : enlever tous les packages: sudo port -fp uninstall installed
- Enlever toute trace de MacPort: sudo rm -rf
  /opt/local
  /Applications/DarwinPorts
  /Applications/MacPorts
  /Library/LaunchDaemons/org.macports.*
  /Library/Receipts/DarwinPorts.*pkg*
  /Library/Receipts/MacPorts.pkg
  /Library/StartupItems/DarwinPortsStartup
  /Library/Tcl/darwinports1.0
  /Library/Tcl/macports1.0
  ~/.macports
- Encore une fois le guide officiel est complet: guide officiel

8.2 Homebrew

Homebrew est, à mon sens, encore plus facile d'utilisation et très prometteur pour les années à venir (il y a de grandes facilités pour les dévreloppeurs). Homebrew est aussi un gestionnaire de package qui est équivalent MacPort, le vocaublaire employé (les commandes) font écho au homebrew, les noms des paquets deviennent des formules/recette que l'on met à jour et que l'on brasse/remue (brew) pour les installer/mettre à jour sur notre ordinateurs. Visitez OpenClassroom pour des informations complémentaires. les paquets seront stockés dans /usr/local/Cellar.

8.2.1 Installation et désinstallation

Avant d'installer MacPort vérifier que Xcode soit installé, que vous ayez accepté les closes de la license et d'avoir entrer la ligne de commande suivante : > xcode-select –install

Installation, entrez dans le terminal :

ruby -e “$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)”

Désinstallation :

/usr/bin/ruby -e “$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/uninstall)”

8.2.2 Commande principales

brew update brew upgrade brew remove brew list brew doctor brew prune brew tape brew info brew update; brew upgrade; brew cleanup; brew cask cleanup;

Chapitre 9

Compiler en ligne de commandes

9.1 Compilation C/C++/Fortran

9.1.1 Valgrind

9.2 Compilation Latex

Chapitre 10

Quelques logiciels libres et des commandes bien utiles

pas macports / installer avec brew

10.1 ffmpeg

10.2 ImageMagick

10.3 Pandoc

10.4 R, Python et Julia

10.5 Versionning avec Git

10.6 Postgresql

10.7 Jekyll