Utiliser le terminal avec MacOS 10.11

Kevin Cazelles kevin(dot)cazelles(at)gmail(dot)com

19 octobre 2015

Résumé

Le terminal est une interface qui propose une communication avec l'ordinateur à l'aide du langage bash (par défaut). Pour communiquer avec l'ordinateur, il est nécessaire de connaître les instructions à lui donner, c'est-à-dire les lignes de commande à utiliser. Je recense dans le présent document 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 utilsés 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

Chapitre 1

Avant-propos

1.1 Mes ressources

L'ensemble de ce qui suit des heures de recherche sur Internet à essayer de résliser tellle ou telle tâche en ligne de commande. J'ai aussi profité de différents échanges avec des amis qui utilisent aussi le terminal. Voici quelques ressources qui vous seront, je l'espère, utiles :

- Base du Terminal
  - graffitix
- Liste des commandes
  - ss64
- Astuces à la volée :
  - maclife
  - A vos mac (il faut être abonné).
- Sites généralistes :
  - Regarder 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, dans le cas de Linux, la documentation est très abondante et la plupart des commandes sont identiques.

1.2 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

1.3 Quelques généralités sur l'application *Terminal*

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 de piloter son ordinateur à l'aide de lignes de commande. L'utilisation du bash est le premier aspect du Terminal développé dans la suite du document. 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.

1.4 Raccourcis clavier (par défaut) du terminal

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).

1.4.1 Gérer les fenêtres et les onglets du terminal

— 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**)
— Sauver le conteniu d'un onglet sous forme de text : **cmd+S**
— Imprimer le contenu de la console : **cmd+P**

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**

1.4.2 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**)
— Aller de mot en mot (avant ou arrière) : **alt+flèche**
— Aller au début d'une ligne : **ctrl + A**

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

1.4.3 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)**
— Effacer le caractère à droite du curseur : **ctrl + D**
— 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.

1.4.4 Executer/arrêter une ligne de commande

— Exécuter une commande : écrire la commande puis **entr**
— Annuler l'exécution d'une commande : **ctrl + C**
— Stopper une tâche : **ctrl + Z**

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

1.5 Que sont les commandes ?

De manière générale, une commande dans un langage donné, est un ensemble de charactères interprétés par l'ordinateur comme une instruction induisant la réalisation d'une tâche par celui-ci (si la commande est correctement entrée). Quelques spécifications pour le langage bash :

— Utilisation d'une commande. Dans « Terminal », il est nécessaire d'entrer le nom de la commande puis éventuellement l'option qui est en général 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 **entr**). Les options peuvent être mises les unes à la suite des autres -option1option2.

— Ex 1 :

ls

```
Il n'y a que la commande _ls_ qui liste les fichiers du repertoire  courant,
- Ex 2 :

> ls -a

la même commande avec une option -a (afficher tous les fichiers,  même les fichiers cach

- Ex 3 :

> ls -a ~/Documents

je rajoute le répertoire pour lequel je veux la listes de tous les  fichiers

- Ex 4 :

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

Crée un fichier avec deux options *-n* et *-v* (raccourcis en   *-nv*), le paramètre 100
```

— 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.

1.5.1 Les commandes basiques du *bash*

Dans la suite je présente différentes commandes du bash. Je ne détaillerai que très peu d'options, pour les connaître, utiliser le manuel. Dans le *bash*, le charactère # est utilisé pour introduire des commentaires.

Naviguer à travers les différents répertoires

— Changer de répertoire :

```
cd adr
# Exemple :
cd /dossiera/dossierb/dossierC/file1.txt
```

— quelques cas particuliers :

```
# Aller à la racine :
cd /
# Aller au dossier utilisateur :
cd ~
# Aller au dossier supérieur :
cd ..
```

— Connaître le répertoire actuel :

```
pwd
```

— 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).

— 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

— 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
```

— 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
```

— Déplacer un fichier ou un dossier :

mv adr1/fichier1 adr2/fichier2

— 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/*

— Ouvrir un fichier :

open + adr/fichier

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

— 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.

1.6 Éteindre/redémarrer et relancer le Finder

— 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

1.7 Gérer l'ordinateur

1.7.1 La date

— Obtenir la date :

date

— Changer la date :

date 0613162785

— Changer la date :

date 1758

1.7.2 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.

1.7.3 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'à n*100% de CPU disponible ou n est le nombre de coeur.). 4. **%TIME** : la durée d'activité du processus 4. **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.

— 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.

1.7.4 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 ~*)

1.7.5 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

1.7.6 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]

1.7.7 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

1.7.8 Connaître son Hardware :

— system_profiler SPHardwareDataType
— sysctl hw
— sysctl machdep.cpu

1.7.9 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)

1.7.10 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 !

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 ^sep.r.te /usr/share/dict/words

- Recherche dans l'historique : history | grep gcc
- Liste des expressions régulières : ———————— ^ 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 [^] 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 ————————

1.7.11 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

1.7.12 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 2

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

2.1 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

2.1.1 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

2.1.2 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

2.2 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.

2.2.1 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)"

2.2.2 Commande principales

> brew update brew upgrade brew remove brew list brew doctor brew prune brew tape brew info

2.2.3 Mon installation

> brew tap homebrew/science
>
> brew install r imagemagick ffmpeg postgresql lynx

NB : cask est une extension de brew il suffit d'utiliser brew cask ou mettre le chemin complet:

> brew cask install java julia mactex dropbox copy rstudio brew install Caskroom/cask/libreoffice Caskroom/cask/appcleaner Caskroom/cask/atom Caskroom/cask/xquartz Caskroom/cask/filezilla

Chapitre 3

Des logiciels libres dans le terminal

3.1 ImageMagick

3.2 FFmpeg