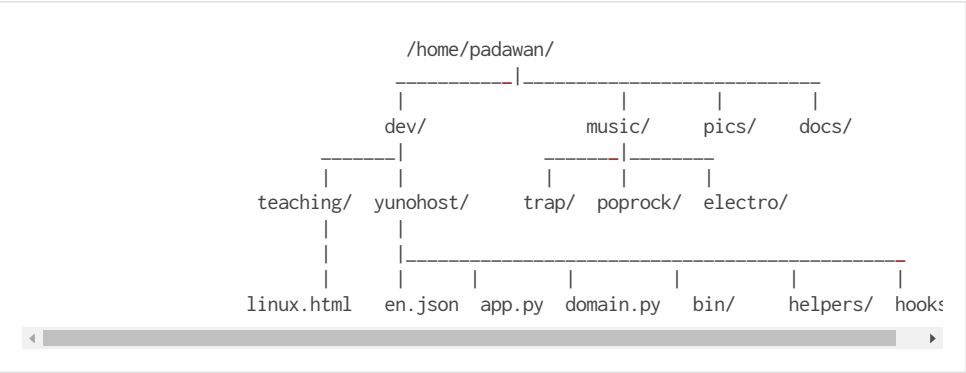




0 - Mémento des commandes shell

On suppose qu'il existe l'arborescence suivante dans votre répertoire personnel (`home`) :



Vous vous trouvez dans le répertoire `music/electro/` (on nomme ce répertoire le *répertoire courant*). Depuis ce répertoire, tous les chemins suivants sont équivalents :

- (absolu) `/home/padawan/dev/yunohost/app.py`
- (absolu) `$HOME/dev/yunohost/app.py`
- (absolu) `~/dev/yunohost/app.py`
- (relatif) `../../dev/yunohost/app.py`
- (relatif) `../../pics/../../dev/teaching/../../yunohost/app.py`
- et aussi: `/var/cache/../../home/./padawan/dev/yunohost/app.py`

Commandes indispensables

commande	description
<code>man <commande></code>	(man ual) Obtenir des informations sur une commande

commande	description
<code>passwd</code>	(pass word) Changer son mot de passe
<code>cd <destination></code>	(change d irectory) Changer de répertoire (si pas de destination, va dans <code>\$HOME</code>)
<code>ls <repertoire></code>	(l ist) Lister le contenu du répertoire
avec option <code>-l</code>	Ajoute des détails (permissions, date de modification, taille)
avec option <code>-a</code>	Liste aussi les fichiers cachés (commençant par <code>.</code>)
avec option <code>-t</code>	Liste en triant suivant la date de modification
avec option <code>-h</code>	Affiche les tailles de fichiers en un format lisible facilement
<code>mkdir <repertoire></code>	(make d irectory) Créer un répertoire
<code>mv <source(s)> <destination></code>	(m ove) Renommer un fichier (ou déplacer plusieurs fichiers dans un répertoire)
<code>cp <fichier> <destination></code>	(copy) Copier un fichier
avec option <code>-r</code>	Copier récursivement un dossier
<code>rm <fichier(s)></code>	(remove) Supprimer définitivement des fichiers
avec l'option <code>-r</code>	Option pour suppression récursive (et définitive) de répertoires
avec l'option <code>-i</code>	Demande confirmation avant les suppressions

Commandes de base

commande	description
<code>echo <chaîne></code>	Afficher une chaîne de caractère dans la sortie
<code>touch <fichier></code>	Créer un fichier (ou changer sa date de modification sans rien changer)

commande	description
<code>pwd</code>	(p rint w orking d irectory) Obtenir le nom du répertoire courant
<code>whoami</code>	(who am I) Obtenir votre nom d'utilisateur
<code>date</code>	(d ate) Afficher la date
<code>chmod <permissions> <fichier></code>	(c hange m ode) Changer les permissions d'un fichier
<code>chown <utilisateur> <fichier></code>	(c hange o wner) Changer le propriétaire d'un fichier
<code>chgrp <groupe> <fichier></code>	(c hange g roupe) Changer le groupe d'un fichier
<code>du <fichier></code>	(d isk u sage) Connaître l'espace disque utilisé par un fichier
avec l'option <code>-h</code>	Donne les tailles dans un format plus facilement lisible
avec l'option <code>-s</code>	Donne uniquement le total pour les dossiers
avec l'option <code>-c</code>	Donne le total de la liste
<code>ln -s <destination> <lien></code>	(l ink) Créer un lien dur
sans l'option <code>-s</code>	Créer un lien dur plutôt qu'un lien symbolique (rare)

Raccourcis de la ligne de commande

commande	description
Flèches du haut/bas	Parcourir les commandes de l'historique
TAB	Auto-compléter une commande ou un nom de fichier
Ctrl + R	Rechercher une commande dans l'historique
Ctrl + A/E	Aller au début/à la fin de la ligne de commande

commande	description
Ctrl + U/K	Supprimer tous les caractères à gauche/droite du curseur
Ctrl + W	Supprimer le mot à gauche du curseur

Caractères speciaux

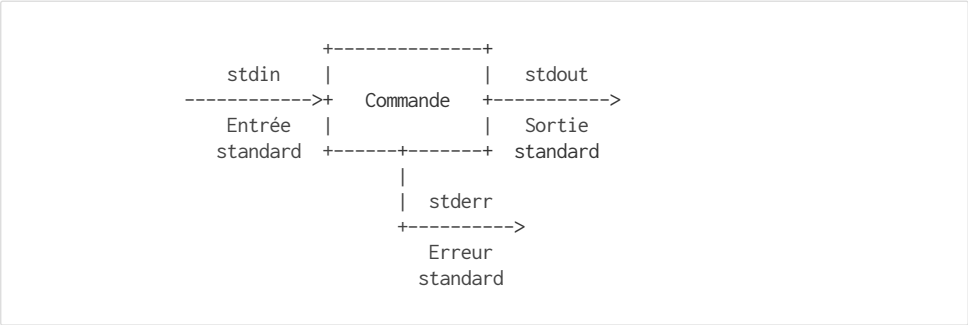
caractère	description
<code>;</code>	Sépare des commandes à exécuter à la suite
<code> </code>	Sépare des commandes à exécuter à la suite (s'arrête si il y a eu des erreurs)
<code>~</code>	Désigne le chemin vers votre répertoire personnel (<code>\$HOME</code>)
<code>.</code>	Désigne le répertoire courant
<code>..</code>	Désigne le répertoire parent\
<code>?</code>	Symbole "joker", interprété comme n'importe quel caractère
<code>*</code>	Symbole "joker", interprété comme n'importe quelle chaîne de caractère
<code>\</code>	Inhibe l'interprétation du caractère spécial suivant
<code>'</code> et <code>"</code>	Délimite des chaînes de caractère
<code>,</code>	Délimite une commande à interpréter dans les arguments d'une autre commande
<code>\$VAR</code>	Désigne le contenu de la variable shell <code>VAR</code>

Exemples

<code>echo "Je suis dans"; pwd</code>	Affiche du texte suivi du chemin absolu du répertoire courant
<code>compile prog.c ./prog.exe</code>	Compile <code>prog.c</code> et exécute <code>prog.exe</code> , sauf si la compilation a échoué
<code>touch ~/toto</code>	Crée un fichier <code>toto</code> dans votre répertoire personnel

Exemples	
<code>touch ./toto</code>	Cr��e un fichier <code>toto</code> dans le r��pertoire courant (��quivalent �� <code>touch toto</code>)
<code>cd ..</code>	Va dans le r��pertoire parent au r��pertoire courant
<code>ls ../../rep2/file?</code>	Liste les fichiers <code>fileb</code> et <code>filec</code> dans <code>rep2</code>
<code>ls prog.*</code>	Liste les fichiers <code>prog.c</code> et <code>prog.exe</code> dans <code>rep1/sousrep1/</code>
<code>echo Une ��toile : *</code>	Affiche le texte <code>Une ��toile : *</code>
<code>echo "Une ��toile : *</code>	Affiche le texte <code>Une ��toile : *</code>
<code>echo "Je suis dans `pwd`"</code>	Affiche du texte suivi du chemin absolu du r��pertoire courant
<code>echo "J'habite dans \$HOME"</code>	Affiche du texte suivi du chemin vers votre espace personnel

Gestion des entr  es-sorties



Syntaxe	description
<code><</code>	Utilise un fichier comme entr��e standard
<code><<<</code>	Utiliser une cha��ne de caract��re comme entr��e standard

Syntaxe	description
<code>></code>	Redirige la sortie standard vers un fichier (��crase l'ancien fichier)
<code>>></code>	Redirige la sortie standard vers un fichier (�� la suite du fichier)
<code>2></code>	Redirige l'erreur standard vers un fichier
<code> ></code>	Redirige la sortie et l'erreur standard vers un fichier
<code> </code>	Redirige la sortie standard d'une commande vers l'entr��e standard d'une autre commande
Exemples	
<code>./prog.exe < inputFile > outputFile</code>	Ex��cute <code>prog.exe</code> avec <code>inputFile</code> en entr��e et met la sortie dans <code>outputFile</code> apr��s l'avoir ��cras��
<code>ls ./prog.exe >> outputFile</code>	Ex��cute <code>prog.exe</code> avec la sortie de <code>ls</code> en entr��e et met la sortie �� la suite de <code>outputFile</code>

Gestion des processus

Syntaxe	description
<code><commande> &</code>	Arr��ter l'ex��cution de la commande en cours
<code><commande> </code>	Ex��cute <code>commande</code> en t��che de fond
<code>Ctrl + Z puis bg</code>	Passe la commande en cours d'ex��cution en t��che de fond
<code>jobs</code>	Lister les processus de votre shell qui sont en t��che de fond
<code>ps -ef</code>	Lister tous les process en cours d'ex��cution sur la machine
<code>kill <PID></code>	Tuer un processus en cours (via son identifiant <code>PID</code>)

Syntaxe	description
<code>pkill</code> <code><processName></code>	Tuer un processus en cours (via son nom <code>processName</code>)
Avec l'option <code>-g</code>	Tuer brutalement le processus
<code>top</code>	Visualiser l'utilisation de la mémoire et du CPU par les processus
Avec l'option <code>-i</code>	Seulement les processus actifs
Avec l'option <code>-p<PID></code>	Seulement le(s) processus correspondant(s) au(x) PID(s) donné(s)
Avec l'option <code>-u<USER></code>	Seulement les processus d'un utilisateur donné
Avec l'option <code>screen</code>	Ouvre une session screen qui permet de lancer des commandes dans un terminal et qui continuera d'exister même si l'on quitte le terminal initial.

Lecture et édition de fichiers

Commandes	Description
<code>cat</code>	Affiche le contenu des fichiers dans la sortie standard
<code>less</code>	Lire et naviguer dans un fichier
<code>nano</code>	Edite un fichier avec <code>nano</code> (éditeur minimaliste)
<code>vim</code>	Edite un fichier avec <code>vim</code> (pour les ninjas)
<code>xemacs</code>	Edite un fichier avec <code>xemacs</code> (pour les pirates)
<code>nedit</code>	Edite un fichier avec <code>nedit</code> (en interface graphique)
<code>evince</code>	Lire un fichier pdf (en interface graphique)

Fichiers de configuration, variables d'environnement

Commandes	Description
<code>~/.bashrc</code>	Fichier de configuration qui est exécuté à chaque connexion
<code>env</code>	Commande qui liste toutes les variables d'environnement définies et leurs valeurs
<code>VAR=3.14</code>	Changer la valeur de <code>VAR</code> (pas d'espaces autour de <code>=</code> !)
<code>export VAR</code>	Rend la valeur de <code>VAR</code> disponible pour tous les process fils de ce shell
<code>echo "VAR vaut \$VAR"</code>	Affiche la valeur de <code>VAR</code>
<code>\$USER</code>	Nom d'utilisateur
<code>\$HOSTNAME</code>	Nom de la machine
<code>\$HOME</code>	Répertoire personnel
<code>\$PATH</code>	Liste des répertoires où les commandes sont recherchées
<code>\$LD_LIBRARY_PATH</code>	Liste des répertoires où les librairies sont recherchées
<code>\$PS1</code>	Décrit la forme de l'invite de commande en bash

Filtres et commandes avancées

Note : les filtres peuvent généralement être utilisés aussi bien sur un fichier que sur l'entrée standard (via `|`)

Commandes	Description
<code>grep</code>	Permet de filtrer ligne par ligne suivant un mot ou un motif

Commandes	Description
<code>sed</code>	Permet de remplacer une expression par une autre
<code>diff</code>	Afficher les différences entre deux fichiers
<code>cut</code>	Manipuler les colonnes d'une entrée
<code>tr</code>	Remplacer ou enlever des caractères
<code>find</code>	Rechercher des fichiers suivant des critères
<code>wc</code>	Permet de compter des mots, des lignes, ..
<code>bc</code>	Effectuer des opérations arithmétiques basiques
<code>ssh</code>	Se connecter de façon sécurisée à une autre machine
<code>scp</code>	Copier des fichiers entre deux machines
<code>tar</code>	Compresser ou décompresser des fichiers (format <code>tar</code>)
<code>gzip</code> , <code>gunzip</code>	Compresser ou décompresser des fichiers (format <code>gz</code>)
<code>alias</code>	Créer des alias
Exemples	
<code>grep -nr "warning" ./</code>	Cherche récursivement les occurences de <code>warning</code> dans les fichiers du répertoire courant
<code>./prog.exe grep -nr "warning" ./</code>	Cherche récursivement les occurences de <code>warning</code> dans la sortie du programme <code>prog.exe</code>
<code>sed "s/search/replace/g" inputFile</code>	Remplace toutes les occurences de <code>search</code> par <code>replace</code>
<code>diff file1 file2</code>	Compare les fichiers <code>file1</code> et <code>file2</code>
<code>cut -d " " -f 2,3 inputFile</code>	Affiche les colonnes 2 et 3 (<code>-f 2,3</code>) du fichier <code>inputFile</code> , par rapport aux espaces (<code>-d " "</code>).

Exemples	
<code>cat inputFile tr " " " : "</code>	Remplace les espaces par des <code>:</code> dans <code>inputFile</code>
<code>find -name "*.cpp" ../</code>	Trouver tous les fichiers se finissant par <code>.cpp</code> dans le répertoire parent
<code>wc -l *</code>	Compte le nombre de ligne pour chaque fichier du répertoire
<code>bc <<< "1+2"</code>	Effectue l'opération 1+2 et affiche le résultat
<code>scp user@host:/home/user/file ./</code>	Copie le fichier <code>/home/user/file</code> depuis la machine <code>host</code> dans le répertoire courant
<code>ssh user@host</code>	Se connecter à la machine <code>host</code> en tant que <code>user</code>
<code>lpr -Pr2d2 outputFile</code>	Imprime le fichier <code>outputFile</code> avec l'imprimante nommée <code>r2d2</code>
<code>tar -xvf archive.tar</code>	Décompresse le fichier <code>archive.tar</code>
<code>gzip inputFile</code>	Comprime <code>inputFile</code> en un fichier <code>inputFile.gz</code>
<code>alias ll='ls -l'</code>	Créer un alias <code>ll</code> qui correspond à la commande <code>ls -l</code>