

Abims⁴

11/05/2014

Linux Initiation

Formation 2014

Le Corguillé – 1.07

UPMC
SORBONNE UNIVERSITÉS



INTRODUCTION

- Open-source et gratuit (le plus souvent)
 - Accès au code
 - Modifier / adapter / redistribuer le code
- Un nombre important de logiciels disponibles dès l'installation:
 - office suites (Open Office)
 - networking: web browser (Firefox), email client (Thunderbird), messenger (gaim), firewall
 - multimedia: audio (amarok), video (VLC), image (Gimp), 3D modeling (Blender)
 - programming & development: C, C++, java, perl, python, php, mysql, ...
 - servers: web, email, ftp, ssh ...
- et de logiciels scientifiques (pour beaucoup exclusifs)
 - Emboss, Blast, R, Phylip, Arb,

- Usage

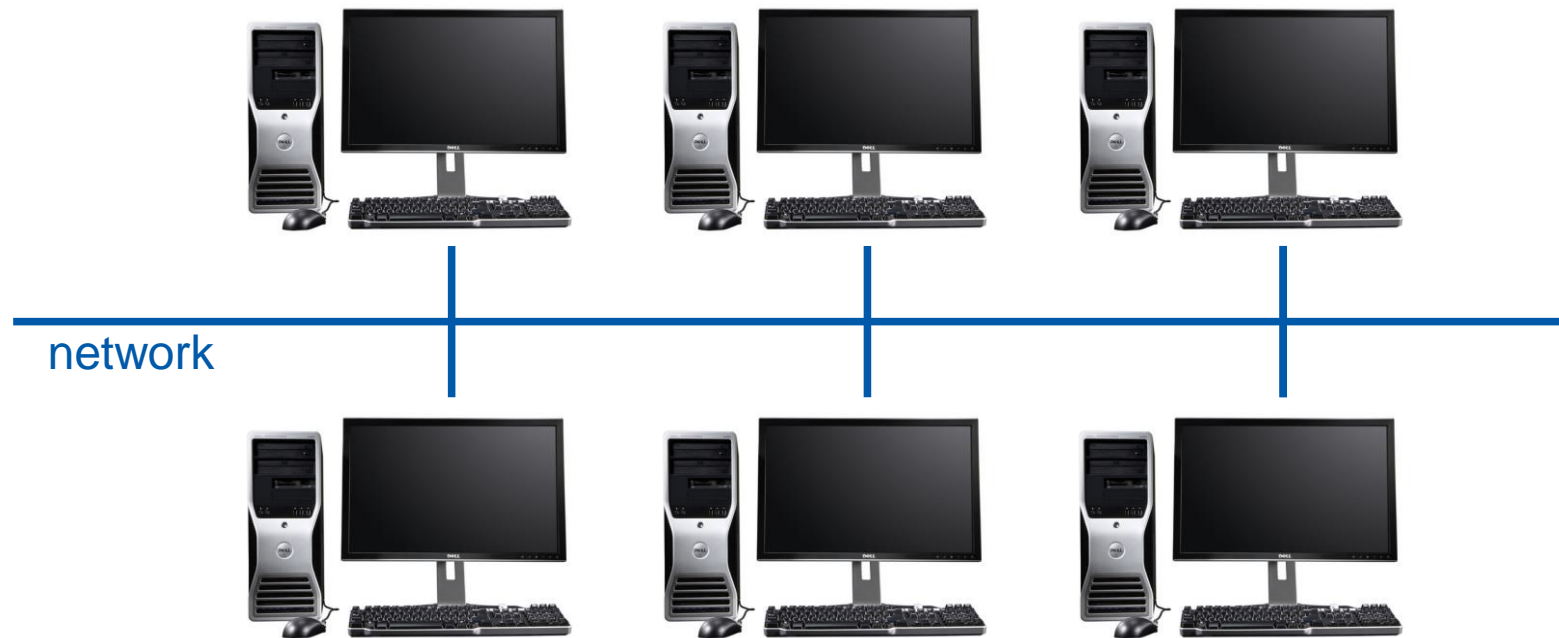
- mono-utilisateur



- multi-utilisateur : client/server architecture

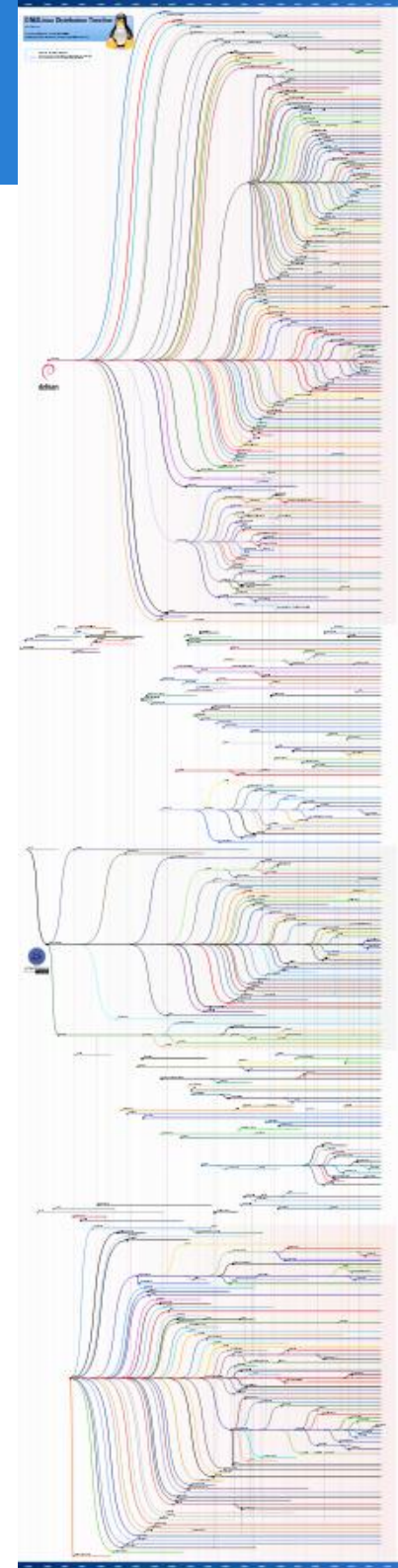


server



Introduction | distribution

- Nos distributions :
 - postes de travail :
 - Ubuntu 14.04 LTS
 - fondée sur Debian
 - serveurs :
 - CentOS 6.x
 - fondée sur RedHat

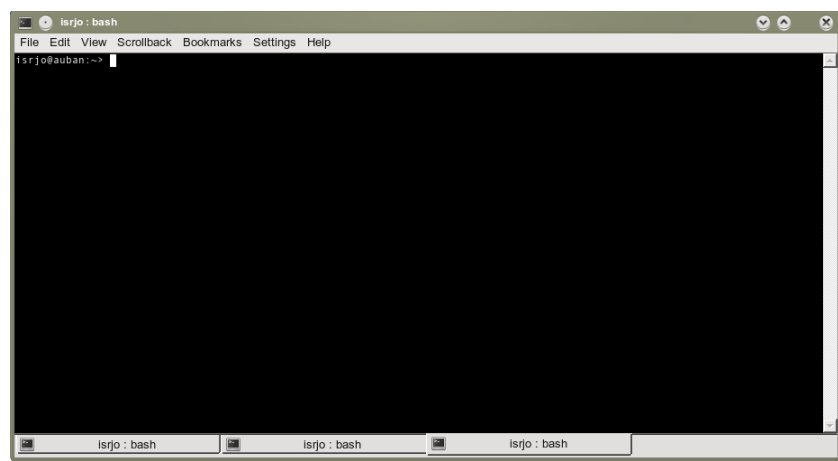


Linux distribution timeline

- Interfaces
 - Graphic User Interface (GUI)



- Command Line Interface (CLI)



CONNEXION

- Le protocole SSH
 - Secure Shell : SSH
 - = protocole de communication sécurisé :
 - échange de clés de chiffrement au moment de la connexion
 - échange chiffré des "données"
 - Ce protocole/programme permet de se connecter à un serveur distant

- A partir d'un Linux ou d'un Mac
 - dans un terminal

```
$ ssh -Y stageXX@bioinfo
```

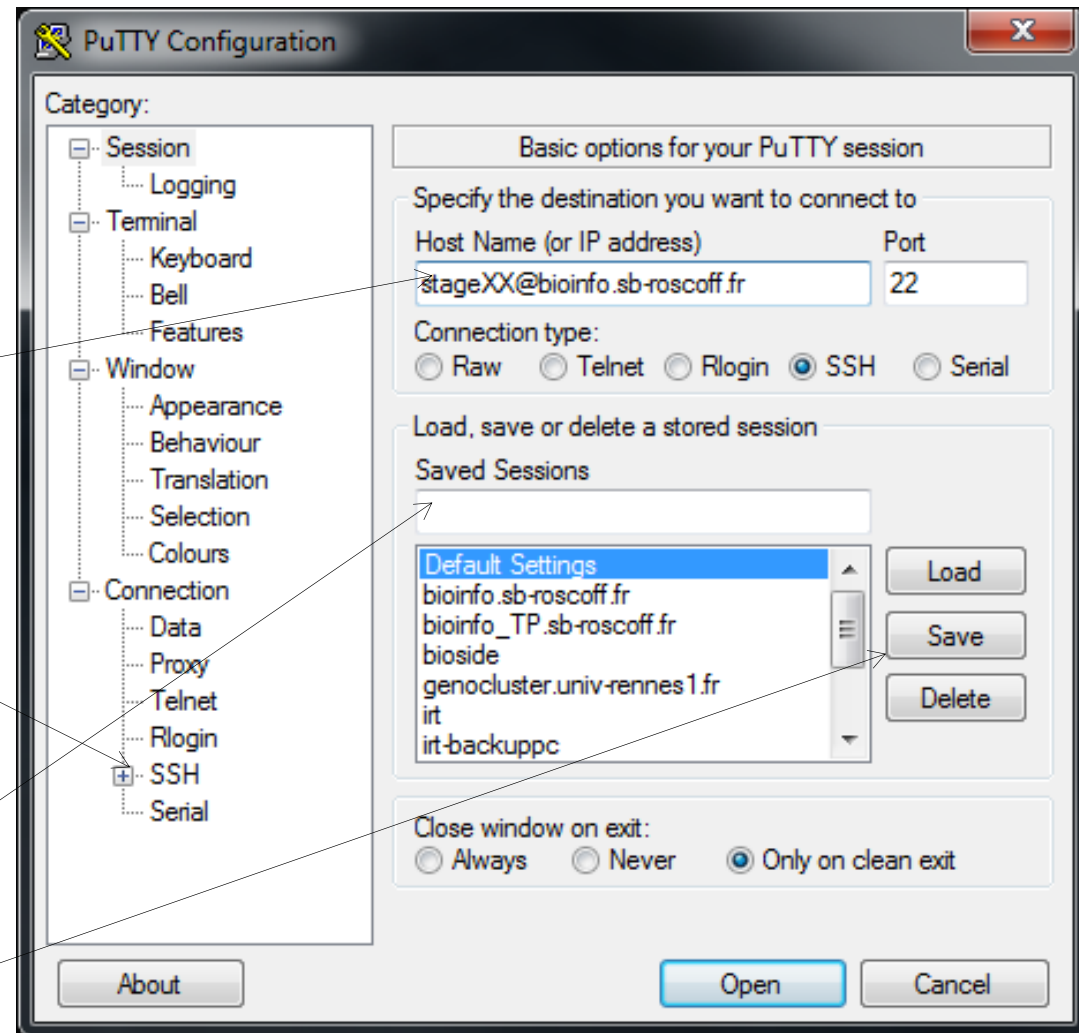
- ssh : le nom du programme
- -Y : l'option pour les programmes X11 : graphique
- stageXX : le nom de l'utilisateur
- bioinfo : le nom du serveur distant

Connexion SSH | A partir de Windows

- A partir de Windows :
 - Le logiciel Putty : www.putty.org



- Session
 - Host Name
 - **stageXX@bioinfo**
- Connection
 - SSH
 - X11
 - Enable X11 forwarding
- Session
 - Saved Sessions
 - **stageXX@bioinfo**
 - Save
- Open



Connexion SSH | A partir de Windows

- A partir de Windows :
 - Le logiciel MobaXterm



The screenshot displays the MobaXterm application window. The 'Sessions' panel on the left has a 'New session' button circled in red (1). The main terminal window shows a prompt and some text. A 'Session settings' dialog box is open, with the 'SSH' icon circled in red (2). In the 'SSH settings' tab, the 'Remote hostname' field contains 'bioinfo' and is circled in red (3). The 'Port' is set to 22. At the bottom of the dialog, the 'OK' button is circled in red (4). The terminal window contains the following text:

```
• MobaXterm Personal Edition v5.0 •  
(Outils Unix et serveur X pour Windows)  
  
> Vos disques durs sont  
> Le DISPLAY est positif  
> Lors d'une connexion  
> Le statut de chaque  
  
• Important:  
Vous utilisez la version  
En achetant la version  
personnaliser le logiciel  
vos options, votre pro  
que plusieurs autres p  
de générer une install  
En plus de vous assure  
modifier le programme  
selon vos besoins. Pou  
sur le lien suivant :
```

[31/08/2012 15:22.21] ~
[lecorguille.sbr4-l021] >

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

LIGNE DE COMMANDE

- Récupération des données

```
$ cd
$ cp -r /tmp/cours/* .
$ ls
    acteur.csv   cours           insulin.fas     insulin_vs_nt.blast
    tmp
```

- Nomenclature pour ce cours

```
$ head insulin.fas # display the 10 first lines  
>gi|163659904|ref|NM_000618.3| Homo sapiens insulin-like
```

- \$ is the prompt symbol
- Words in `courrier` type are commands and should be typed as they are printed
- Words in `bold` type should be substituted with the appropriate filename, directory, login ...
- Linux is case-sensitive – UPPER and lowercase letters have different meanings
- [TAB] [Ctrl] [ESC] → keys on the keyboard
- Words in grey are console returns
- # are comments

- Exemple de lignes de commande et de programmes :
 - head : affiche les 10 premières lignes d'un fichier

```
$ head insulin.fas
```

- Le raccourci et chemin absolu

```
$ which head # permet d'obtenir le chemin du programme  
/usr/bin/head
```

```
$ /usr/bin/head insulin.fas # avec chemin complet
```

- Obtenir de l'aide :
 - Help : à la charge du développeur donc parfois très épuré voir absent

```
command --help  
command -help  
command
```

```
command -h  
command help  
...
```

```
$ head --help
```

- Manuel : plus complet mais plus rare (commandes unix)
 - man command

```
$ man head  
[q] # pour quitter
```


- Arguments : désignent plutôt les fichiers d'entrée (input)

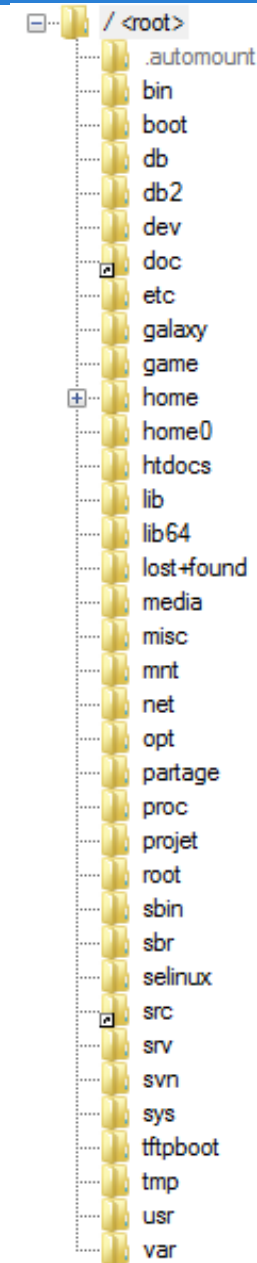
```
$ head insulin.fas # 10 premiers lignes
```

- Options : désignent le paramétrage
 - -n : version court / --lines : version longue
 - [--lines] : option facultative
 - --lines <Integer> : type de l'option (ici un entier)
 - Default = '10' : valeur par défaut en cas d'omission

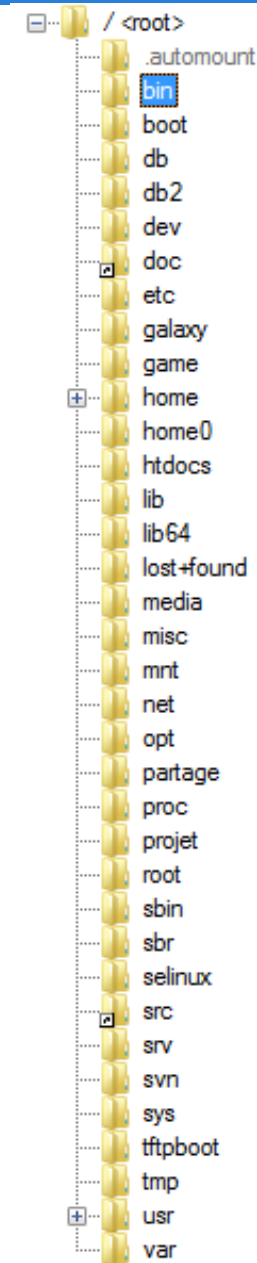
```
$ head -n 4 insulin.fas # 4 premiers lignes  
$ head --lines 4 insulin.fas # 4 premiers lignes  
$ head -n 1 insulin.fas # premier ligne
```

NAVIGATION

- /
 - La racine de l'arborescence s'appelle root et est noté /
 - A la différence de windows où les différents disques sont disponibles sous les lecteurs : c:, d:
 - les dossiers peuvent être "montés" sur plusieurs disques durs sans que cela se voit
- Lecture seule

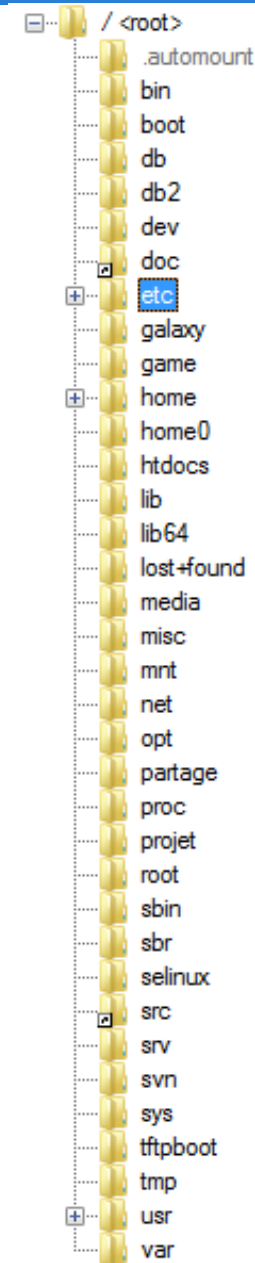


- /bin
 - Ce répertoire contient les programmes essentiels au système et sont utilisés par les utilisateurs.
 - Ces programmes sont installés au moment de l'installation et correspondent aux commandes unix : ls, cd, head ...
- /sbin
 - Ce répertoire contient des programmes dédiés à l'administration du système.
- Lecture seule



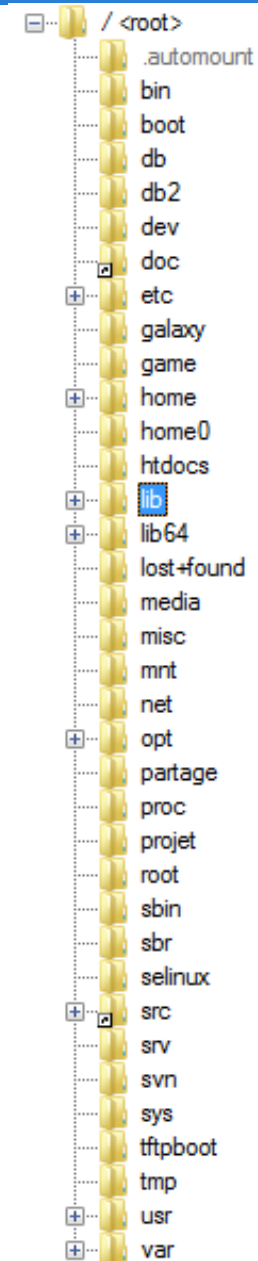
- /etc
 - Contient les fichiers de configuration nécessaires à l'administration du système
 - Ex : fichiers passwd, group, inittab, ld.so.conf, lilo.conf, ...

- Lecture seule et encore ...



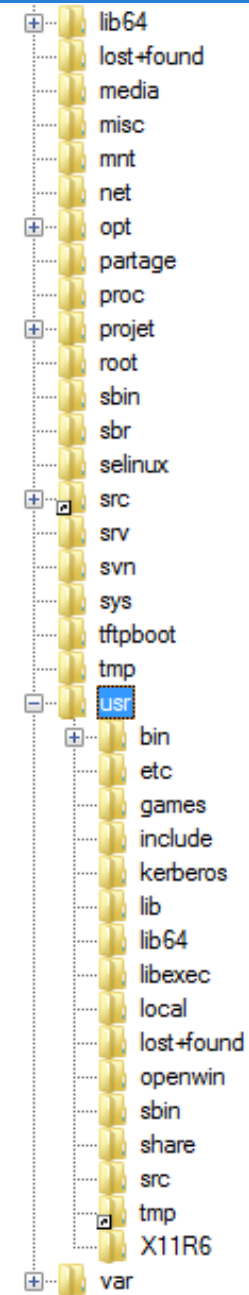
- /lib
 - Contient les bibliothèques standards partagées entre les différentes application du système.
 - Ces fichiers sont les équivalents des dll de Windows

- Lecture seule

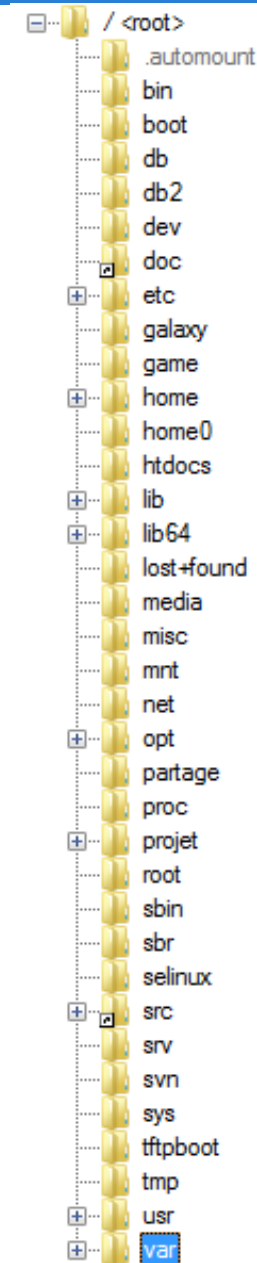


- /usr
 - Hiérarchie secondaire
 - Il contient les équivalents de bin, etc, lib ...
 - C'est l'emplacement dans lequel sont installés les programmes non essentiels au système : Firefox, Gimp, OpenOffice ...

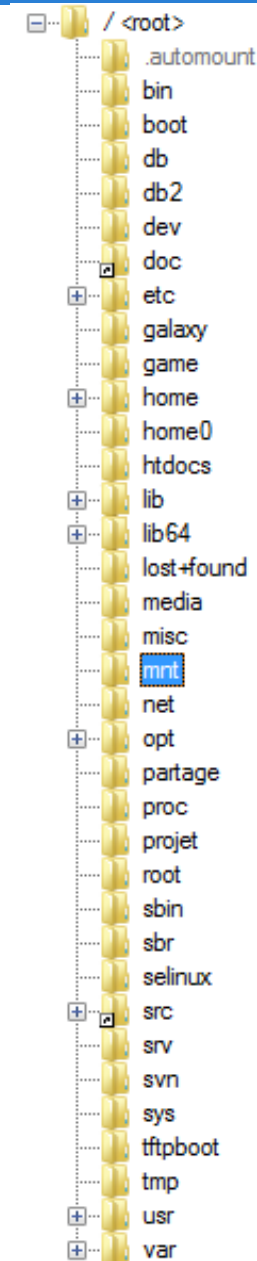
- Lecture seule



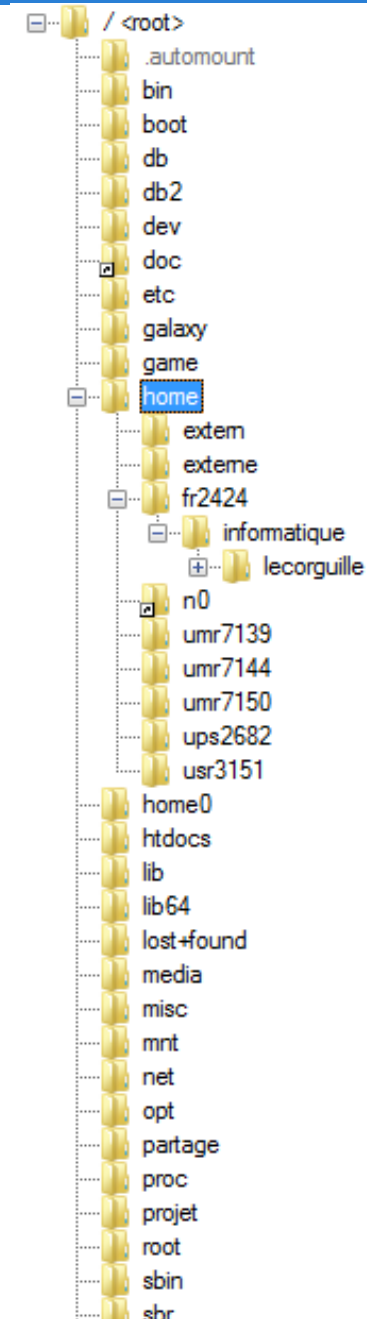
- /var
 - Répertoire des données :
 - site web (pour un serveur wouaib)
 - emails (pour un serveur email)
 - logs
 - ...



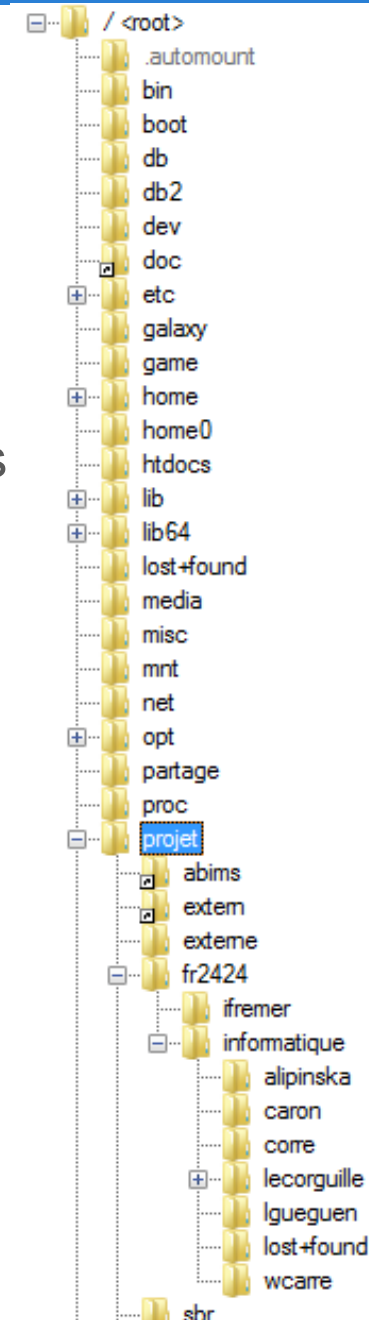
- /mnt
 - Permet d'accueillir les points de montage des partitions temporaires (cd-rom, disquette, ...).
- /media
 - Equivalent de mnt suivant les linux
- Lecture seule mais suivant les medias l'écriture peut être accordée



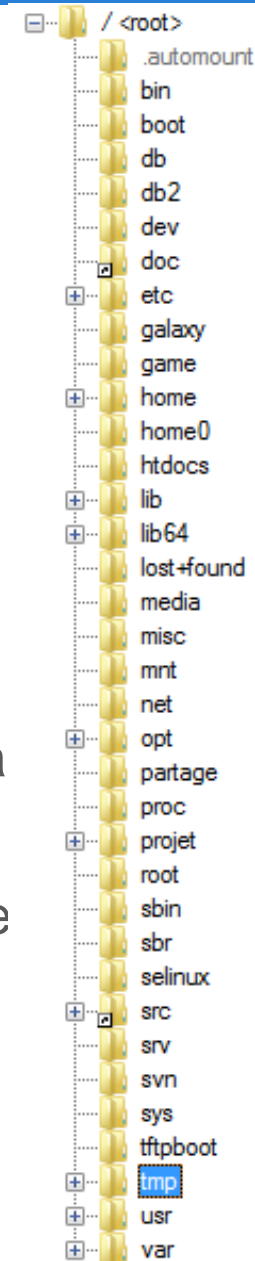
- /home
 - Contient les répertoires personnels des utilisateurs.
 - Un chacun et on ne regarde pas chez le voisin.
 - Dans votre répertoire, vous avez les droits en lecture et écriture.
- A la Station, les home sont hébergés sur un serveur de disques dédié. Le répertoire home est donc un montage NFS. Ainsi vous retrouvez vos répertoires quel que soit le serveur.

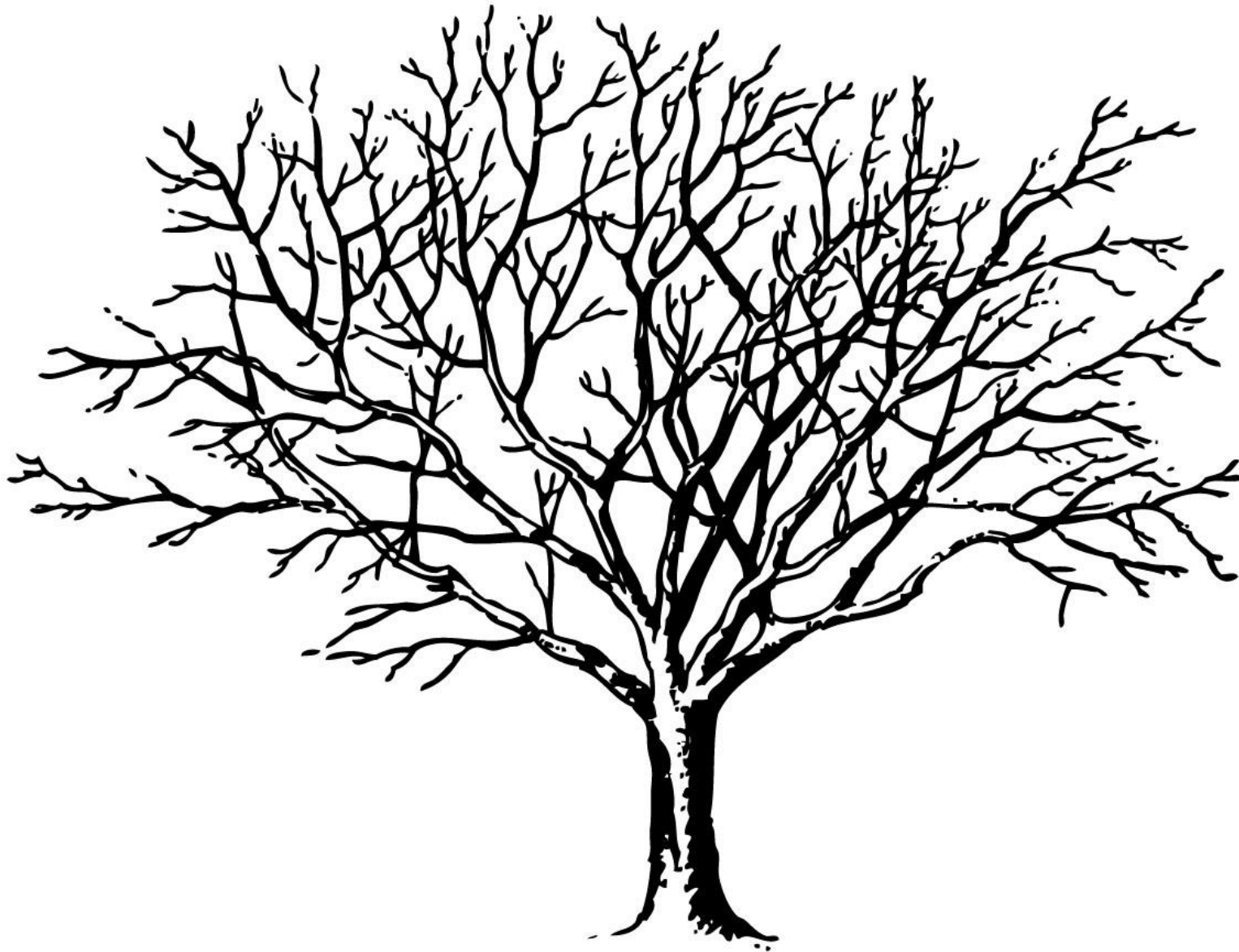


- /projet
 - Ce répertoire n'est pas dans l'arborescence par défaut de Linux.
 - Ce répertoire a été créé sur les serveurs de calcul de la Station pour accueillir les données d'entrée et de sortie des logiciels de calcul
 - /projet est un montage depuis un serveur de disques dédié au cluster de calcul.
 - Cela permet de soulager le serveur qui héberge les home.
 - L'arborescence est la même que dans /home et sera créée sur demande (cf : module cluster)



- /tmp
 - Contient des fichiers temporaires
 - Sera vidé de ces vieux fichiers régulièrement
 - Ce répertoire est en lecture et écriture. Il peut servir de dossier d'échange entre utilisateurs.
- /home/[...]/tmp
 - Dans chaque home, il y a un dossier tmp qui a le même rôle que le dossier /tmp.
 - Il contient des fichiers temporaires créés par le programmes que vous lancez.
 - A nettoyer par l'utilisateur





- Lister le contenu du répertoire courant : ls

```
$ ls
acteur.csv      exemples.desktop  Modèles          Vidéos
Bureau          Images            Musique
cours           insulin.fas       Public
Documents       insulin_vs_nt.blast  Téléchargements
```

- Lister les fichiers cachés (.fichier)

```
$ la # raccourci pour ls -a
acteur.csv      .gconf           Musique
.bash_history   .gconfd          .nautilus
.bash_logout    .gksu.lock       .profile
.bashrc         .gnome2          Public
```

- Lister les fichiers qui commencent par ... :

```
$ ls insulin*  
    insulin.fas  insulin_vs_nt.blast
```

- Lister les fichiers qui contiennent ... :

```
$ ls *nt*  
    insulin_vs_nt.blast
```

- ...

- L'autocomplétion : "vous allez l'aimer !"
 - Pour les programmes

```
$ he [TAB][TAB]
    head helixturnhelix help hesinfo hetparse hexdump
$ hea [TAB]
$ head
```

- Pour les fichiers (fonction "ls")

```
$ head [TAB][TAB]
    .bashhistory      .bash_logout      .bash_profile    ...
$ head insulin[TAB][TAB]
    insulin.fas      insulin_vs_nt.blast
$ head insulin.[TAB]
    insulin.fas
```


- Lister le contenu d'un répertoire en aval

```
$ ls cours  
Linux-Initiation.pdf
```

- Lister le contenu d'un répertoire en amont

```
$ ls ..  
caron          jkervellec    legrand       mhoebeke     ppericard  
corre         jmaroumougom lgueguen     duvignac     lecorguille  
oquenez       wcarre  
  
$ ls ../..  
accueil        crbm          externe       infrastruct  
restaurant    administration direction      ifremer  
lbm           sre          communication documentati  
sib           mer          valorisation
```

- Tree

- Affichage de l'arborescence un peu plus "graphique" que ls

```
$ tree
```

```
.
|-- acteur.csv
|-- cours
|   |-- Linux-Initiation.pdf
|   `-- Thumbs.db
|-- insulin.fas
|-- insulin_vs_nt.blast
`-- tmp
```

```
$ ls -R
```

```
.:
acteur.csv  cours  insulin.fas  insulin_vs_nt.blast  tmp

./cours:
Linux-Initiation.pdf  Thumbs.db

./tmp:
```

- Où suis-je ? : pwd

```
$ pwd  
/home/fr2424/informatique/lecorguille
```

- Changer de répertoire : cd

```
$ cd cours  
$ pwd  
/home/fr2424/informatique/lecorguille/cours  
$ ls  
Linux-Initiation.pdf
```

- Remonter dans les répertoires

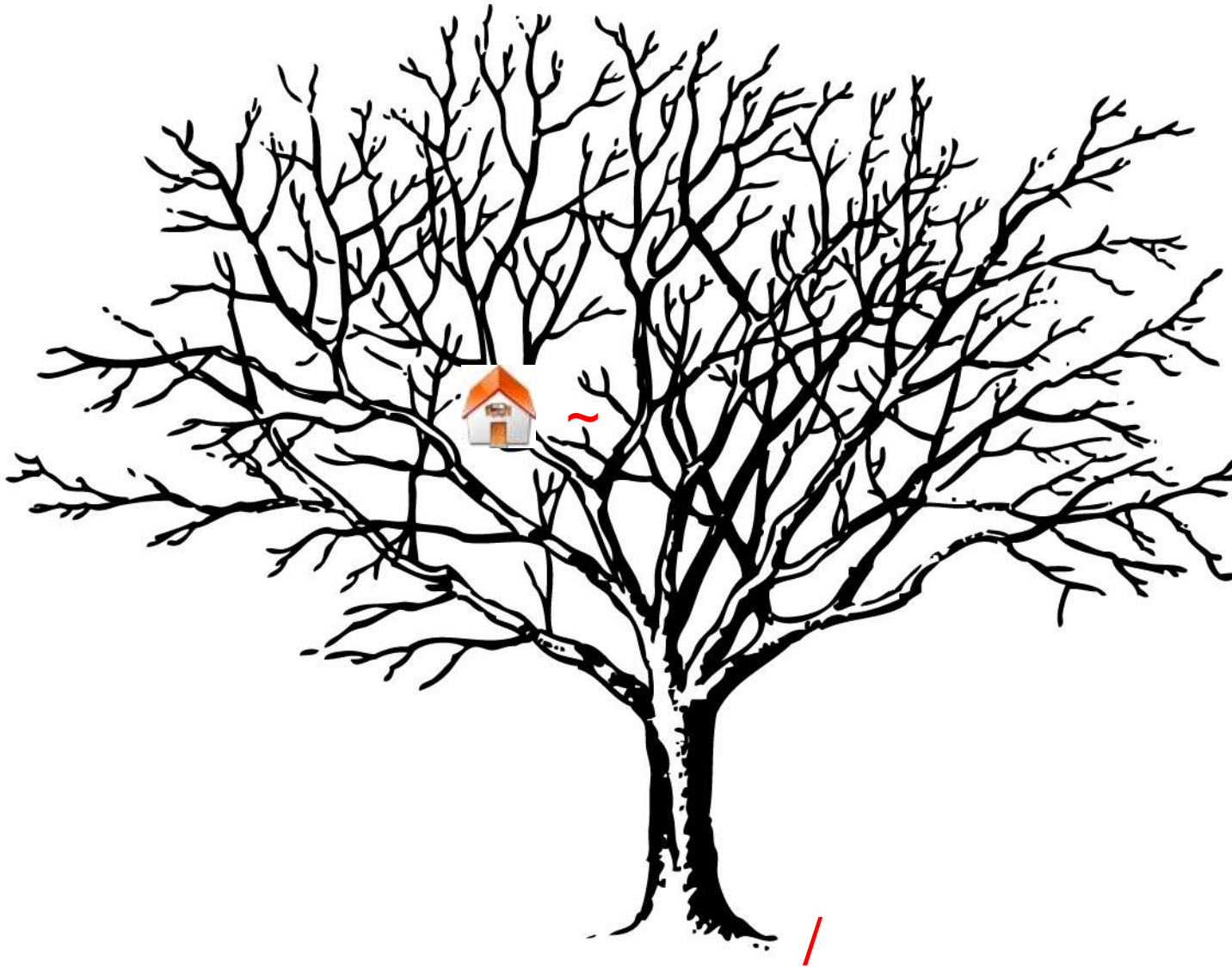
```
$ cd ..  
$ pwd  
/home/fr2424/informatique/lecorguille
```

- La racine du système

```
$ ls /
```

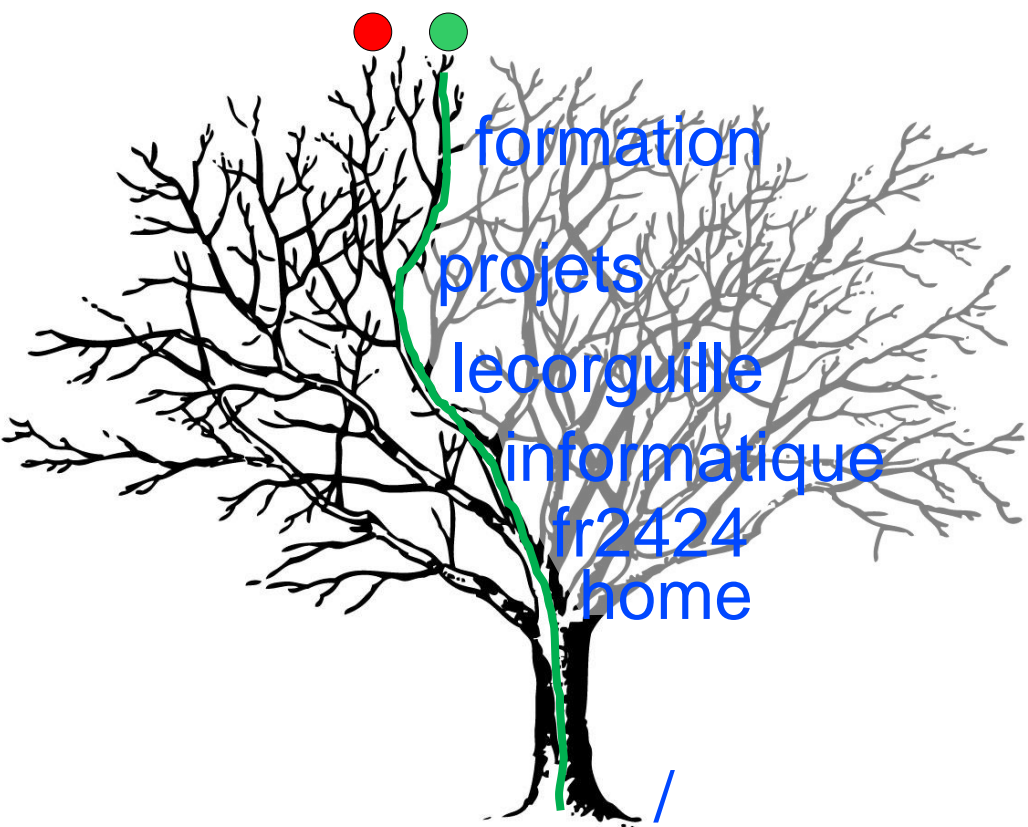
- Retour à la maison (home directory)

```
$ cd  
$ cd ~ # ~ désigne aussi votre répertoire home  
$ pwd  
/home/fr2424/informatique/lecorguille
```

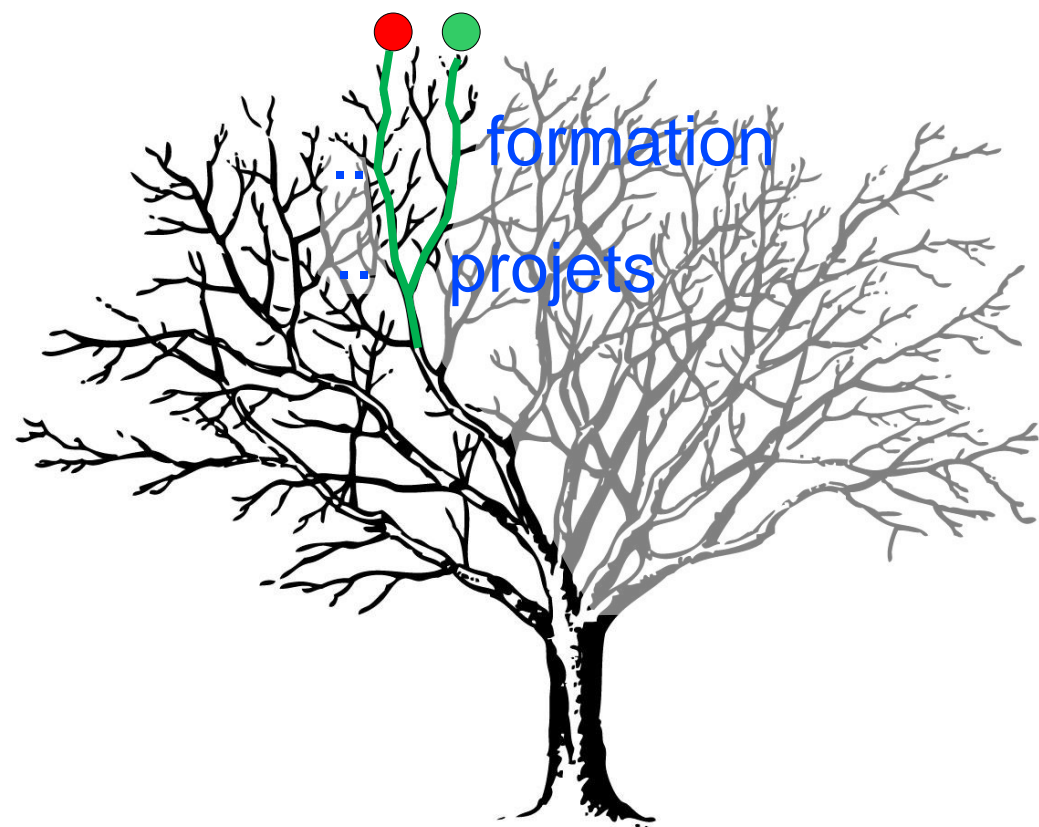


Navigation | Chemin absolu VS relatif

- Chemin absolu



- Chemin relatif

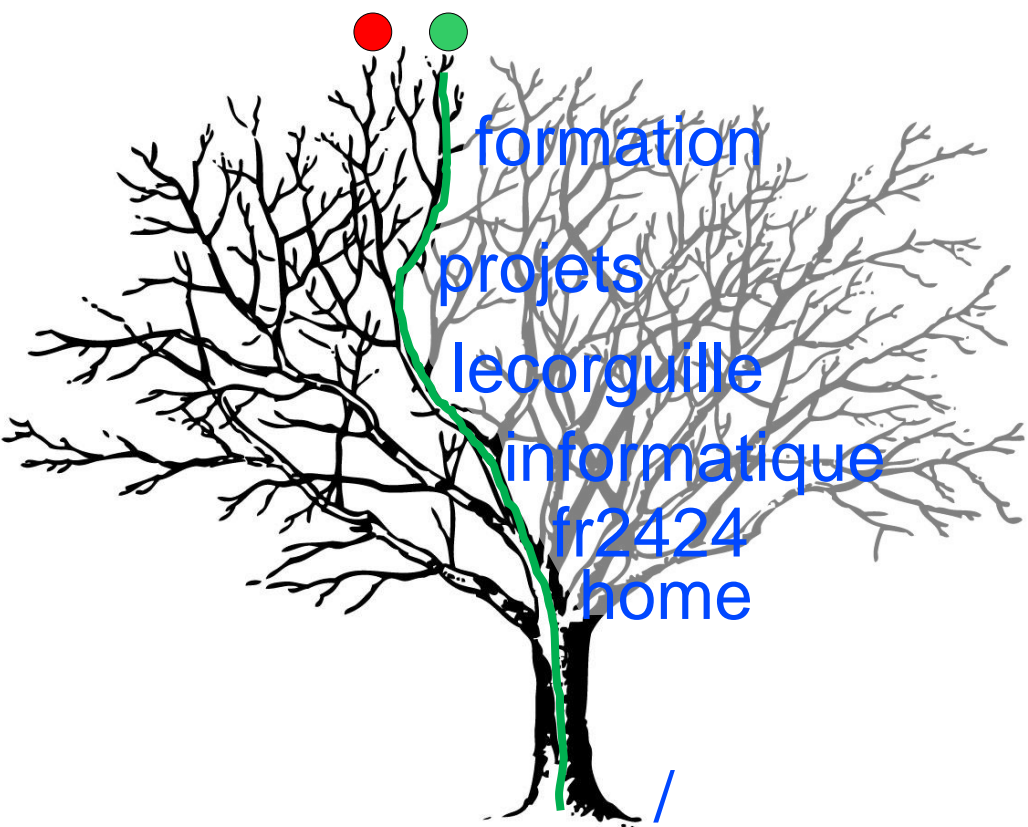


```
$ cd /home/fr2424/informatique/lecorguille/projets/formation
```

```
$ cd ../../projets/formation
```

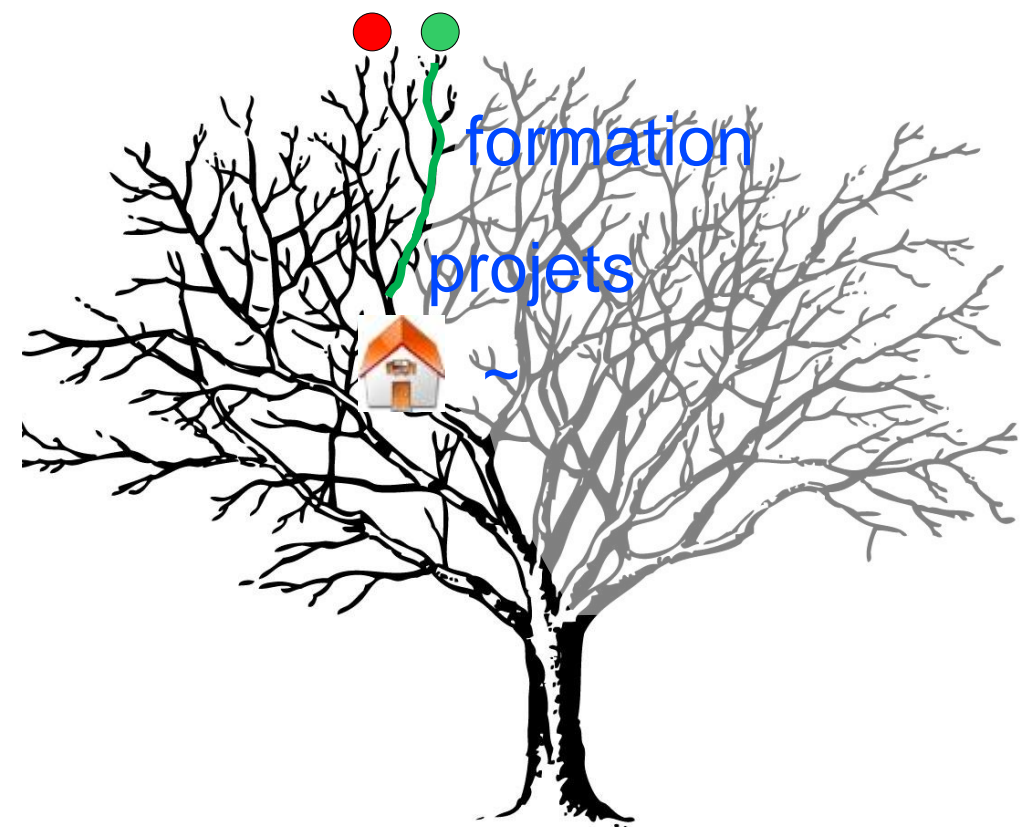
Navigation | Chemin absolu VS relatif

- Chemin absolu



```
$ cd /home/fr2424/informatique/lecorguille/projets/formation
```

- Chemin absolu alternative



```
$ cd ~/projets/formation
```

- Le chemin absolu part toujours de la racine /

```
$ cd /home/fr2424/informatique/lecorguille/documents
$ pwd
/home/fr2424/informatique/lecorguille/documents
$ cd /home/fr2424/informatique/lecorguille
$ pwd
/home/fr2424/informatique/lecorguille
```

- Le chemin relatif part toujours de l'emplacement actuel

```
$ pwd
/home/fr2424/informatique/lecorguille
$ cd documents
$ pwd
/home/fr2424/informatique/lecorguille/documents
$ cd ..
$ pwd
/home/fr2424/informatique/lecorguille
```


- Le chemin absolu

```
$ cd  
$ pwd  
    /home/fr2424/informatique/lecorguille  
$ cd /tmp  
$ pwd  
    /tmp
```

- Le chemin relatif

```
$ cd  
$ pwd  
    /home/fr2424/informatique/lecorguille  
$ cd tmp  
$ pwd  
    /home/fr2424/informatique/lecorguille/tmp
```

ASTUCES / CONSEILS

- Le copier/coller sauce Linux : "vous allez aussi l'aimer !"

```
$ ls
  acteur.csv  cours      insulin.fas  insulin_vs_nt.blast
$
```

1 – commencé votre commande

```
$ ls
  acteur.csv  cours      insulin.fas  insulin_vs_nt.blast
$ head
```

2 – surligné le texte d'interêt / double cliquer

```
$ ls
  acteur.csv  cours      insulin.fas  insulin_vs_nt.blast
$ head
```

3 – clic milieu (ou pour putty : clic droit)

```
$ ls
  acteur.csv  cours      insulin.fas  insulin_vs_nt.blast
$ head insulin_vs_nt.blast
```

- Lors de la création de fichiers ou répertoires, il est recommandé d'utiliser les `_` aux espaces

ex : mon fichier texte.txt → mon_fichier_texte.txt

- Les espaces sont souvent interprétés comme plusieurs arguments

```
$ mkdir mon dossier                # 2 dossiers seront créés
$ ls
dossier      insulin.fas mon
$ mkdir mon_dossier                # 1 dossier sera créé
```

- Pour indiquer un espace, il faut l'échapper avec `\`

```
$ mkdir mon\ dossier                # 1 dossier sera créé
$ ls
insulin.fas mon dossier
$ mkdir "mon dossier"                # 1 dossier sera créé
```

MANIPULATION

- Affichage

```
cat insulin.fas # affiche tout le fichier  
# (pas très lisible quand le fichier est long)  
less insulin.fas # affiche par page - navigable  
[q]  
head insulin.fas # affiche les 10 premières lignes  
tail insulin.fas # affiche les 10 dernières lignes
```

- Editer

```
vi insulin.fas # éditeur console pour les durs : on passe  
[Esc][:]q![Enter] # inscription pour sortir :P  
  
gedit # éditeur graphique simple  
gedit insulin.fas
```

- File : obtenir le type d'un fichier

```
file insulin.fas  
insulin.fas: ASCII text
```

```
file cours  
cours: directory
```

```
file /usr/bin/file  
/usr/bin/file: ELF 32-bit LSB executable, Intel 80386, ...
```

- Fichier pdf

```
$ file cours/Linux-Initiation.pdf  
cours/Linux-Initiation.pdf: PDF document, version 1.4
```

- evince

```
$ evince cours/Linux-Initiation.pdf &
```


- Rappel sur les extensions (Linux et Windows)
 - Les extensions ne sont souvent qu'indicatives
 - Elles permettent à l'OS de choisir le programme qu'il va utiliser
 - Ex du même fichier sous Windows:

```
First Name;Last Name;Age
Chuck;Norris;72
Sylvester;Stallone;66
Steven;Seagal;61
```

 - acteur.txt → sera ouvert par un éditeur de texte (Notepad)
 - acteur.csv → sera ouvert par un tableur (Excel)csv = comma-separated values
 - En mode console, c'est à vous de trouver les bons outils
 - Par contre, vous pouvez créer vos extensions (.blast, .sorted, ...)

- Création d'un dossier (make directory)

```
$ mkdir insulin-files
```

- Suppression d'un dossier vide (remove directory)

```
$ rmdir insulin-files
```

- Suppression d'un dossier rempli (remove recursive)

```
$ rm -r insulin-files
```

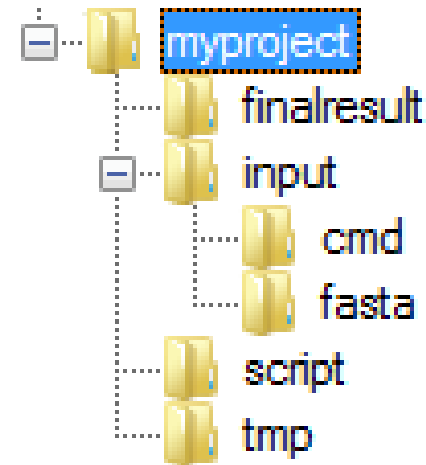
- Suppression d'un dossier rempli si on est sûr de son coup

```
$
```

- Créer cette arborescence

Corrections

```
$ cd
$ mkdir myproject
$ cd myproject
$ mkdir finalresult input script tmp
$ cd input
$ mkdir cmd fasta
```

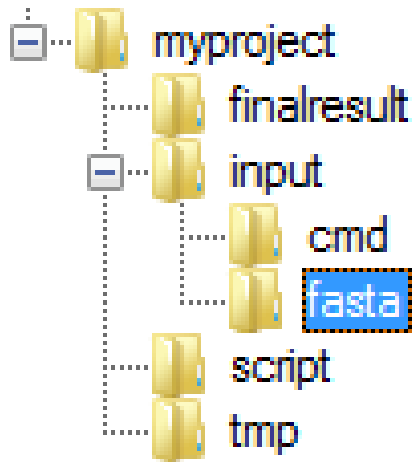


- Vérifier l'arborescence en lançant de votre home :

Corrections

```
$ tree
$ tree -L 1
$ tree -L 2
```

- Déplacer vous dans le dossier *fasta*



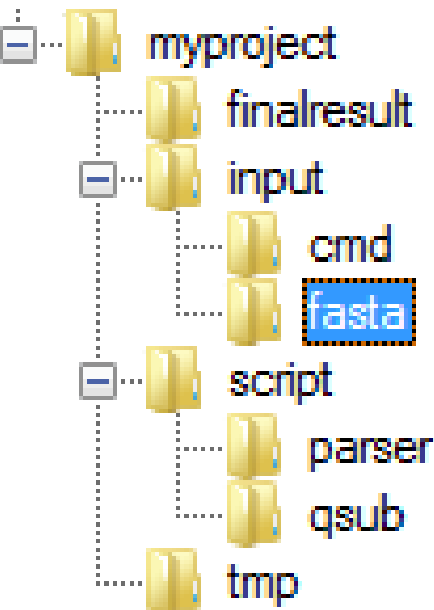
Corrections

```

$ cd
$ cd myproject/input/fasta
  
```

- En une ligne de commande créer un dossier *script*

dans



Corrections

```

$ mkdir ../../script/parser
  
```

- Copier (copy)

```
$ cp insulin.fas insulin_vs_nt.blast insulin-files
$ ls insulin-files/
  insulin.fas  insulin_vs_nt.blast
```

- Déplacer / renommer (move)

```
$ mv insulin.fas insulin_protein.fas # renommage
$ mv insulin* insulin-files # déplacement
$ ls insulin-files/
  insulin_protein.fas  insulin_vs_nt.blast
```

- Effacer (remove)

```
$ rm insulin-files/insulin3.fas
```

- Recherche de lignes contenant un mot dans un fichier

```
$ grep ">" insulin.fas  
>gi|163659904|ref|NM_000618.3| Homo sapiens insulin-like  
>gi|163659900|ref|NM_001111284.1| Homo sapiens insulin-li
```

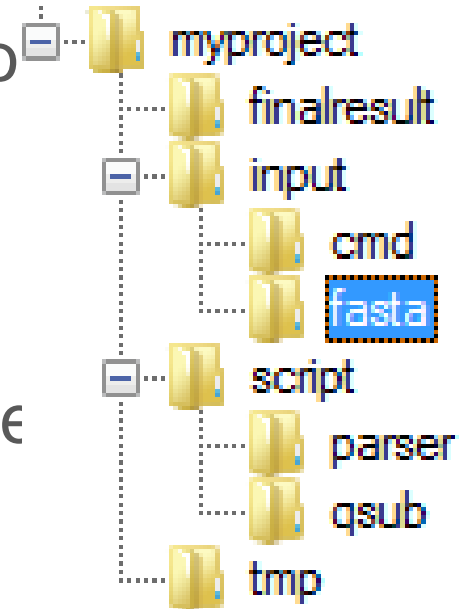
- Comptage de lignes contenant un mot dans un fichier

```
$ grep ">" -c insulin.fas  
5
```

- Comptage des lignes dans un fichier

```
$ wc -l insulin.fas  
516 insulin.fas
```

- Déplacer le fichier *insulin.fas* dans le réperto



Corrections

```
$ mv insulin-files/insulin.fas myproject/input/fasta
```

- Déplacer le fichier *insulin_vs_nt.blast* dans le finalresult

Corrections

```
$ mv insulin-files/insulin_vs_nt.blast myproject/input/fasta
```

- Effacer le répertoire *insulin-files*

Corrections

```
$ rmdir insulin-files  
$ rm -r insulin-files
```

ARCHIVAGE

- Connaître la taille d'un dossier

```
$ du --help
Usage: du [OPTION]... [FILE]...
-h, --human-readable  print sizes in human readable format
-s, --summarize       display only a total for each
                       argument

$ du -sh *
28M   Archive
32K   Desktop
0     html
4.5M  Images
15G   projets
27M   tmp
```

- Archivage

```
$ tar -cvf insulin.tar insulin.fas insulin_vs_nt.blast  
# Create insulin.tar from files insulin*
```

- Compression

```
$ gzip insulin.tar  
# Create insulin.tar.gz from insulin.tar
```

- Archivage et Compression

```
$ tar -czvf insulin.tar.gz insulin.fas insulin_vs_nt.blast  
# Create insulin.tar.gz from insulin*
```

l'extention .tgz signifie la même chose que .tar.gz

- Décompression

```
$ gunzip insulin.tar.gz  
# Decompress insulin.tar.gz to insulin.tar
```

- Désarchivage

```
$ tar -xvf insulin.tar  
# De-tar insulin.tar to files insulin.fas ...
```

- Décompression et Désarchivage

```
$ tar -xzvf insulin.tar.gz  
# Decompress insulin.tar.gz to files insulin.
```

TRANSFERT DE FICHER

- Les protocoles du transfert de fichier

- FTP : File Transfert Protocol

- On retrouve ftp dans les navigateurs :

- `ftp://ftp.ncbi.nih.gov/genomes/`



Index de ftp://ftp.ncbi.nih.gov/genomes/

[Vers un rép. de plus haut niveau](#)

Nom	Taille	Dernière modification
Acyrtosiphon_pisum		15/03/2012 20:29:00
Aedes_aegypti		21/09/2010 00:00:00
Ailuropoda_melanoleuca		01/09/2011 00:00:00
Amphimedon_queenslandica		15/03/2012 20:50:00
Anolis_carolinensis		15/03/2012 20:50:00
Anopheles_gambiae		14/01/2008 00:00:00
Apis_florea		13/04/2012 19:43:00
Apis_mellifera		14/06/2011 00:00:00
Arabidopsis_lyrata		11/05/2011 00:00:00

- ftp est aussi un outil linux pour naviguer et manipuler (récupérer/déposer) des fichiers sur un serveur ftp distant

- SFTP (SSH FTP)

- est la version chiffrée (ssh) de FTP

- Client en ligne de commande : scp
 - un mix entre les commandes cp et ssh

cp

```
$ cp fichier destination
```

+ ssh

```
$ scp fichier user@server:destination # déposer un fichier  
user@server's passwd:
```

```
$ scp user@server:fichier destination # récupérer un fichier  
user@server's passwd:
```

- user est facultatif si vous utilisez votre login actuel

```
$ scp fasta/*.fas brazil:/tmp  
stageXX@brazil's passwd:
```

- Le téléchargement en ligne de commande : wget
 - wget (World Wide Web + get)
 - permet de récupérer du contenu d'un serveur Web ou FTP.

```
$ wget http://www.sb-roscoff.fr/images/stories/intranet/Annuaire_
Organigrammes/Annuaire.pdf

--2012-04-20 15:36:36-- http://www.sb-roscoff.fr/images/stories/intranet/Annuaire_Organigrammes/Annuaire.pdf
Resolving www.sb-roscoff.fr... 192.168.1.202
Connecting to www.sb-roscoff.fr|192.168.1.202|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47937 (47K) [application/pdf]
Saving to: `Annuaire.pdf'

100%[=====>] 47,937      --.-K/s   in 0s

2012-04-20 15:36:36 (381 MB/s) - `Annuaire.pdf' saved [47937/47937]
```

- Client graphique

- FileZilla



- Cyberduck



- Client graphique

- FileZilla



- Cyberduck



- Montage NFS
 - Le protocole NFS permet à un ordinateur d'accéder à des fichiers via un réseau.
 - C'est le cas des montage /home, /projet, /db ...

```
$ df | grep "home"  
brazil:/home/umr7139/defenses 453G 206G 225G 48% /home/umr7139/defenses  
brazil:/home/umr7139/genetique 335G 270G 49G 85% /home/umr7139/genetique  
brazil:/home/umr7139/genomique 315G 242G 58G 81% /home/umr7139/genomique
```

- Client graphique

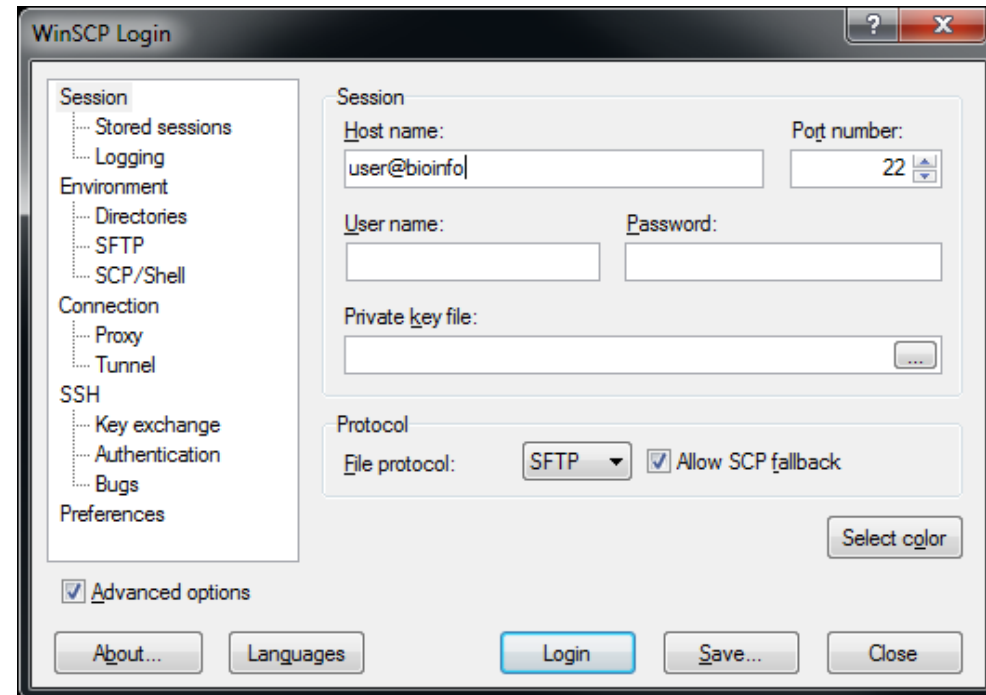
- FileZilla



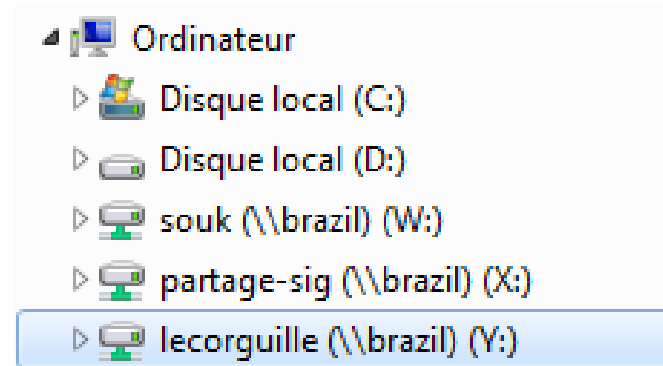
- Cyberduck



- WinSCP



- Montage Samba
 - Le protocole CIFS permet à un ordinateur Windows d'accéder à des fichiers hébergés sur un serveur Linux via un réseau.
 - C'est le cas à la Station des lecteurs réseaux : souk et "espace personnel" et de mybackup

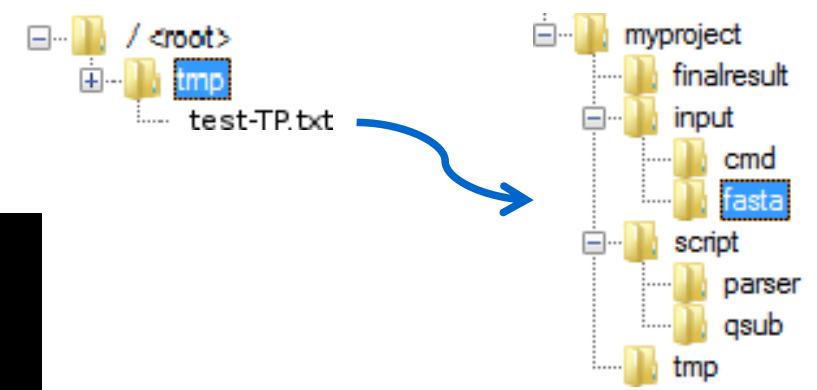


Transfert de fichier | Exercice

- En une ligne de commande :
 - récupérer le fichier nommé test-TP.txt sur le serveur sbr2 dans le répertoire /tmp



| ssh



Rappel :

```
$ scp fichier user@server:destination
user@server's passwd:
$ scp user@server:fichier destination
user@server's passwd:
```

Corrections

```
$ scp sbr2:/tmp/test-TP.txt myproject/finalresult
```

LES RÔLES

- Obtenir les droits sur les fichiers

```
$ ls -l
```

```
$ ll
```

```
$ ll insulin.fas  
-rw-rw-r-- 1 lecorguille sib 312 Aug 10 09:19 insulin.fas
```

- Les Utilisateurs / user / uid
 - utilisateur lambda

```
-rw-rw-r-- 1      lecorguille sib 312 Aug 10 09:19 insulin.fas
```

- Les Groupes / group / gid
 - contient des utilisateurs

```
-rw-rw-r-- 1    lecorguille sib 312 Aug 10 09:19 insulin.fas
```


- Les actions d'administration :
 - modifier les fichiers systèmes et fichiers de configurations
 - créer des comptes user et group
 - gérer les services et processus
 - peut se faire passer pour n'importe quel compte classique sans mot de passe
 - ...
- ne peuvent être effectuées par les utilisateurs classiques.
- Suivant les distributions :
 - CentOS : il faut passer par l'utilisateur root
 - Ubuntu : il faut passer par la commande sudo

```
$ sudo command  
passwd :
```

LES DROITS

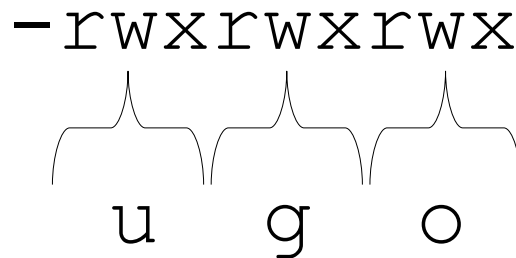
• Obtenir les droits sur un fichier

```

$ ll insulin.fas # raccourci de ls -l
-rw-rw-r-- 1 lecorguille sib 312 Aug 10 09:19 insulin
-rwxrwxrwx user group ...
$ ll ~/..
drwx----- 4 caron sib 4096 Aug 31 09:55 caron
drwx----- 4 corre sib 4096 Aug 31 10:00 corre
    
```

– Les droits :

- r : read
- w : write
- x : execute



– Les acteurs :

- u : user
- g : group
- o : other

- Obtenir les droits sur un fichier

- Exemples :

`-rw-rw-r--` :

le propriétaire peut lire et modifier
le group du propriétaire peut lire et modifier
les autres peuvent lire

`-rwxr-xr-x` :

le propriétaire peut lire, modifier et executer
le group du propriétaire peut lire et executer
les autres peuvent lire et executer

- Notes :

- Les répertoires doivent être exécutable pour être ouvert

- Modifier les propriétaires d'un fichier
 - Seul root peut modifier les propriétaires

```
$ chown dupond insulin.fas # change le propriétaire
```

```
$ chgrp dupondetdupont insulin.fas # change le groupe
```

```
$ chown dupond.dupondetdupont insulin.fas  
# change le user et le group
```

- Modifier les accès
 - Seul le propriétaire peut modifier les accès à son fichier

```
$ chmod g+w insulin.fas
           # donne le droit d'écriture sur le fichier

$ chmod o-r insulin.fas
           # retire le droit de lecture sur le fichier

$ chmod -R g+w ../cours
           # donne les droits de lecteur sur
           # tous les éléments du répertoire
```

PROCESSUS

- Un processus est un programme en cours d'exécution. Par exemple, chaque fois que l'on lance la commande `ls`, un processus est créé durant l'exécution de la commande.

```
$ sleep 1000 # fonction d'attente
```

- On observe que l'on a “perdu la main” sur la console
- Pour la récupérer, il faut fermer gedit

- Pour garder la main, il faut lancer le processus en tache de fond
- Lancer un processus en tâche de fond (background)

```
$ gedit & # Lancement du processus en tache de fond  
[1] 10799 # PID du processus
```

- Mettre un processus en tâche de fond si on n'a pas utilisé de &

```
$ gedit # Lancement du processus  
[Ctrl][Z] # Suspension du processus  
bg # Placement du processus en background
```

Au commencement, il y eut l'allumage du système

*Naquit le **pseudo-processus** (PID : 0)*

*Puis, il y eut le premier processus appelé **init** (PID : 1)*

*Init **enfant** ensuite tous les **processus système** (PID : 2, 3, 4 ...)*

Enfin, le système fût lancé

*Et d'autres processus comme la console purent voir le jour et eux même
enfanter*

...

Un processus est identifié par un numéro unique que l'on appelle le **PID** (Process IDentifiant).

Un processus dispose d'un processus père que l'on appelle le **PPID** (Parent PID).

Un processus possède les droits de celui qui l'a lancé.

- “Qui Tue le père, tue les fils”
- Manipulation :
 - Ouvrir un terminal
 - Lancer gedit
 - Fermer la console

On observe que gedit (fils) est tué quand on tue le terminal (père)

- Certains programmes de bioinformatique peuvent tourner pendant des semaines. Donc comment faire pour garder ces processus ouverts sur le serveur même après avoir couper la connection avec celui-ci ?

- Solution : nohup

```
$ nohup blastn -query insulin.fas -db /db/blast/all/nt  
# nohup va éviter la mort du processus fils  
# blastn quand on tuera la console (le père)
```

- Si il est trop tard : si le processus court déjà et que vous auriez du le lancer avec nohup : disown

```
$ blastn -query insulin.fas -db /db/blast/all/nt &  
[1] 10799 # Lancement du processus en tache de fond  
# PID du processus
```

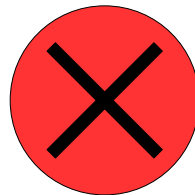
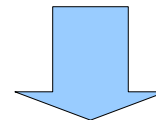
```
$ disown 10799
```

Vous pouvez ensuite fermer votre terminal et observer que
blastn reste ouvert

- Solution : nohup

```
lecorguille@sbr4-l021: ~  
lecorguille@sbr4-l021:~$ nohup sleep 3000 &  
[1] 19805  
lecorguille@sbr4-l021:~$ nohup: les entrées sont ignorées et la sortie est ajoutée à «nohup.out»  
lecorguille@sbr4-l021:~$
```

```
lecorguille@sbr4-l021: ~  
lecorguille@sbr4-l021:~$ ps -edf | grep sleep  
589      19805 19746  0 14:22 pts/4    00:00:00 sleep 3000  
589      19822 19289  0 14:23 pts/7    00:00:00 grep --color=auto sl  
leep  
lecorguille@sbr4-l021:~$
```



```
lecorguille@sbr4-l021: ~  
lecorguille@sbr4-l021:~$ ps -edf | grep sleep  
589      19805      1  0 14:22 ?          00:00:00 sleep 3000  
589      19837 19289  0 14:23 pts/7    00:00:00 grep --color=auto sl  
leep  
lecorguille@sbr4-l021:~$
```

- Pour obtenir la liste des processus les plus gourmands

```
$ top
top - 16:45:33 up 63 days, 1:38, 25 users, load average: 3.25, 3.13,
Tasks: 496 total, 4 running, 311 sleeping, 177 stopped, 4 zombie
Cpu(s): 25.9%us, 10.6%sy, 0.0%ni, 62.4%id, 1.1%wa, 0.0%hi, 0.0%si,
Mem: 8174164k total, 7052324k used, 1121840k free, 131820k buffe
Swap: 4192924k total, 1668292k used, 2524632k free, 3370312k cache

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 31131 lecorgui  25   0 3425m 3.1g 1768  R  90.9  39.5  464:24.84 blastp
  8357 corre    18   0  109m  952  872  R   0.1   0.0 14780:27 man ln
...
```

```
$ man top # pour de plus amples informations
2. FIELDS / Columns
2a. DESCRIPTIONS of Fields
  a: PID -- Process Id
     The tasks unique process ID, which periodically wraps, though
     restarting at zero.

  b: PPID -- Parent Process Pid
     The process ID of a tasks parent.
```

- Obtenir tous vos processus

```
$ ps
```

```

    PID TTY          TIME CMD
  29392 pts/25    00:00:00 bash
  29694 pts/25    00:00:00 gedit
  29781 pts/25    00:00:00 ps
```

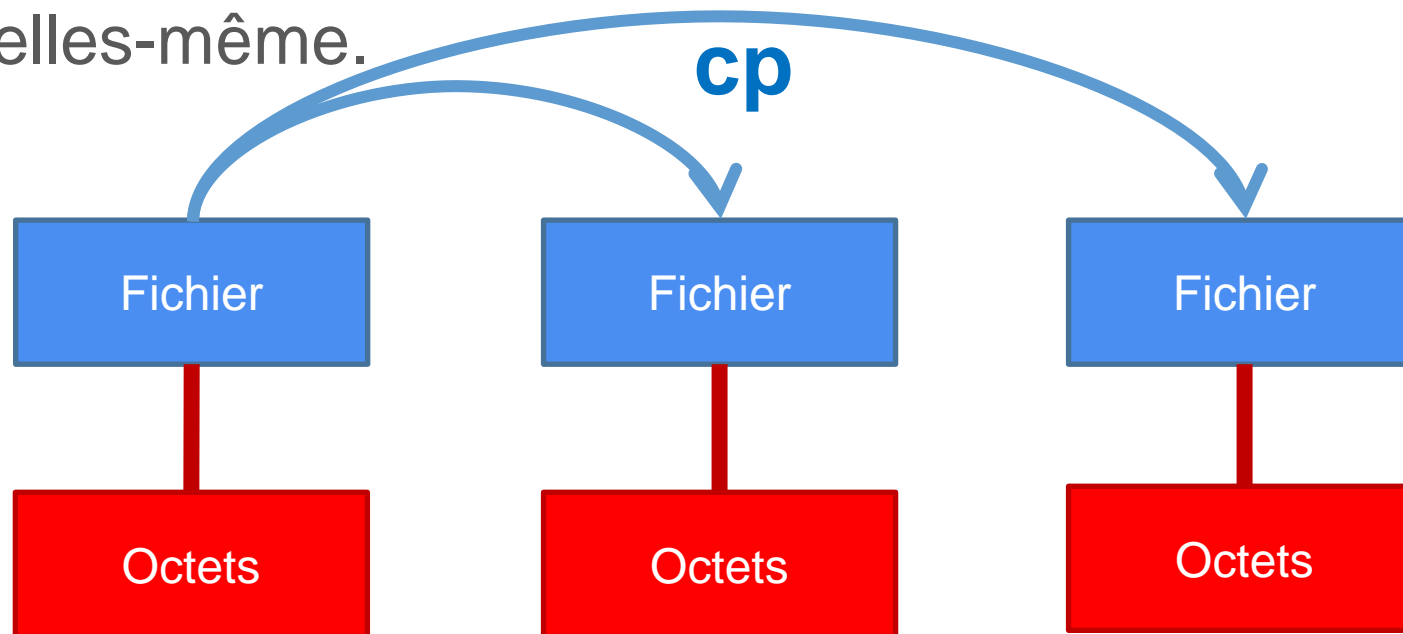
```
$ ps -edf
```

```

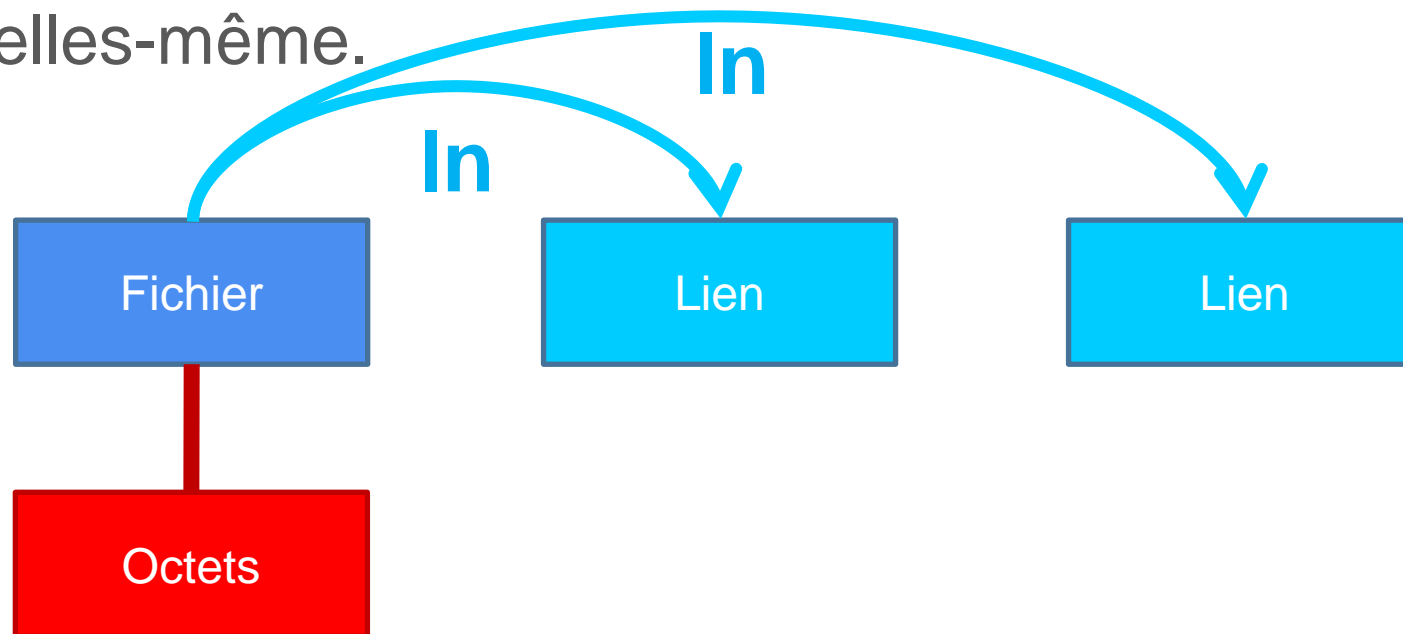
UID          PID    PPID  C  STIME TTY          TIME CMD
root           1        0  0   Jul27 ?           00:00:05 init [5]
root           2        1  0   Jul27 ?           00:00:02 [migration]
root           3        1  0   Jul27 ?           00:00:00 [ksoftirqd]
root           4        1  0   Jul27 ?           00:00:00 [watchdog]
...
lecorguille 29392        1  0  14:05 pts/25    00:00:00 bash
lecorguille 29694  29392  0  16:35 pts/25    00:00:00 gedit
lecorguille 29781  29392  0  16:36 pts/25    00:00:00 ps
```


LIEN SYMBOLIQUE

- Les liens symboliques sont des raccourcis mais en mieux ...
- Ils permettent de multiplier les points d'entrée vers un fichier ou un répertoire sans dupliquer les données en elles-même.

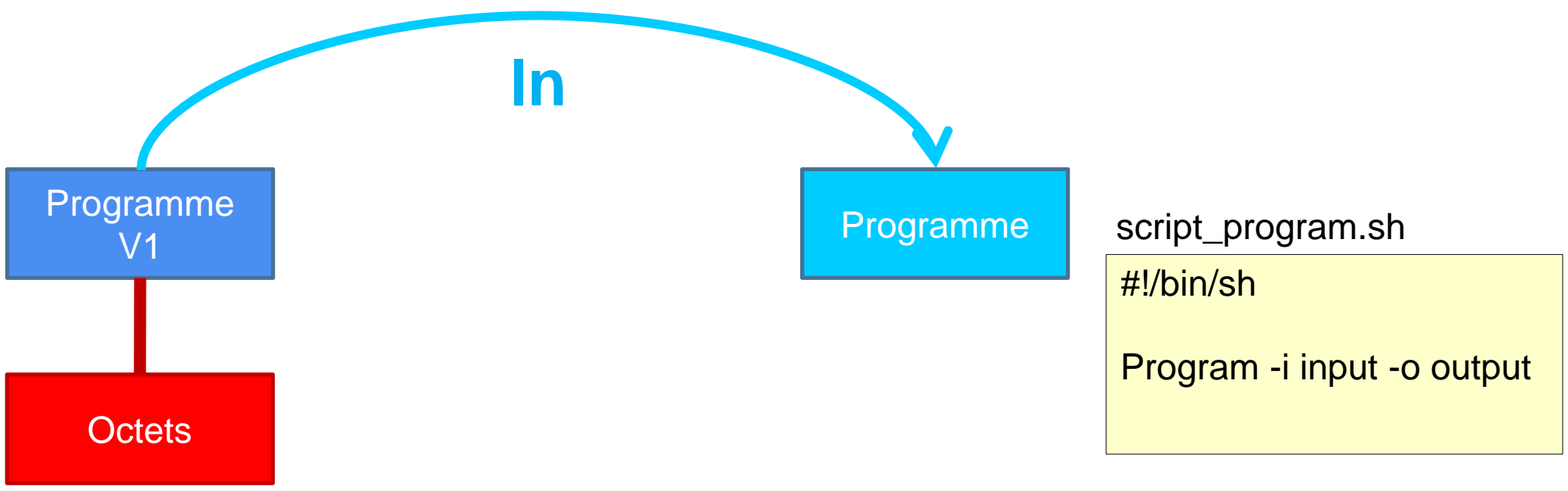


- Les liens symboliques sont des raccourcis mais en mieux ...
- Ils permettent de multiplier les points d'entrée vers un fichier ou un répertoire sans dupliquer les données en elles-même.



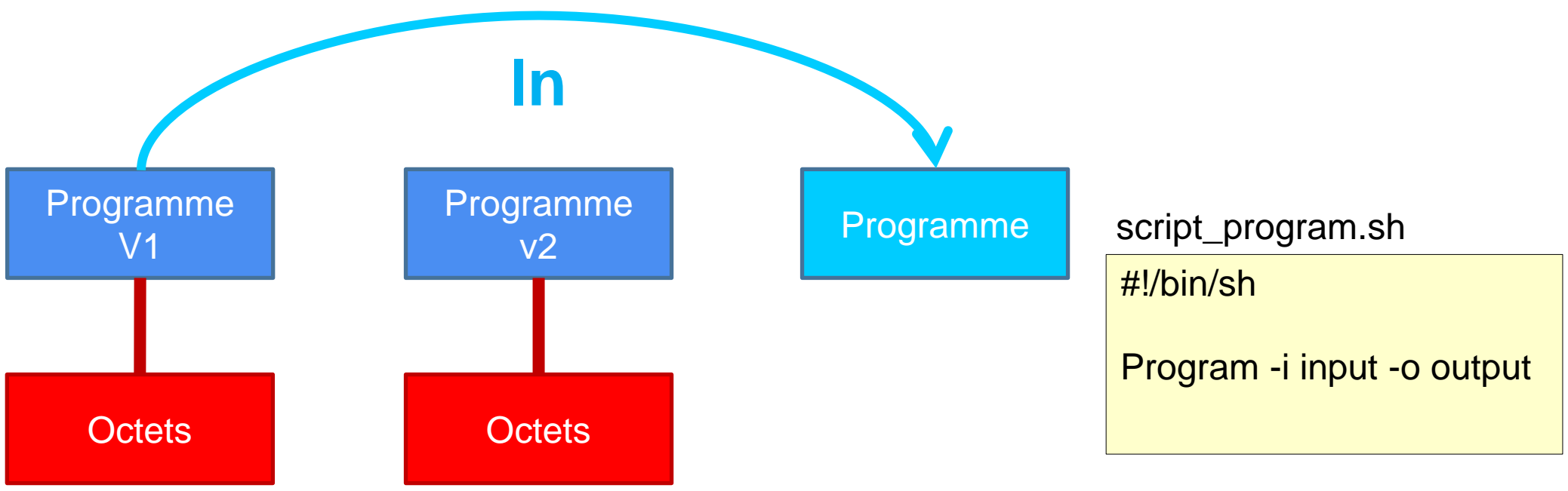
Lien symbolique | Exemple

- Une de nos applications des liens symboliques :
 - L'installation des programmes



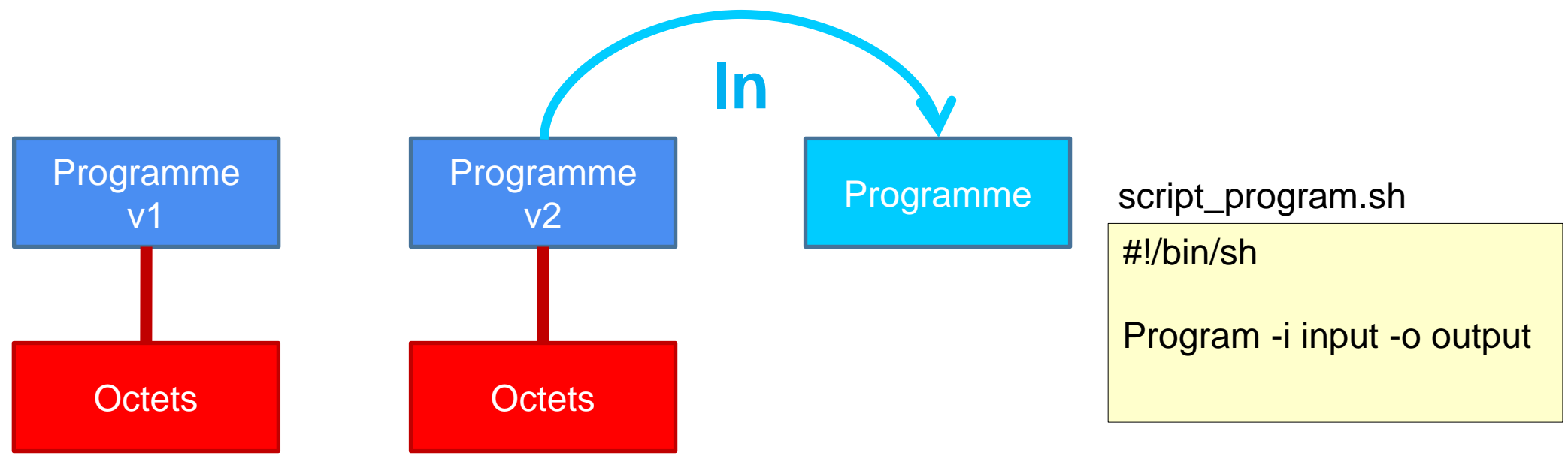
Lien symbolique | Exemple

- Une de nos applications des liens symboliques :
 - L'installation des programmes



Lien symbolique | Exemple

- Une de nos applications des liens symboliques :
 - L'installation des programmes



- La commande ln fonctionne sur le même schéma que la commande cp

```
$ cp fichier destination
```

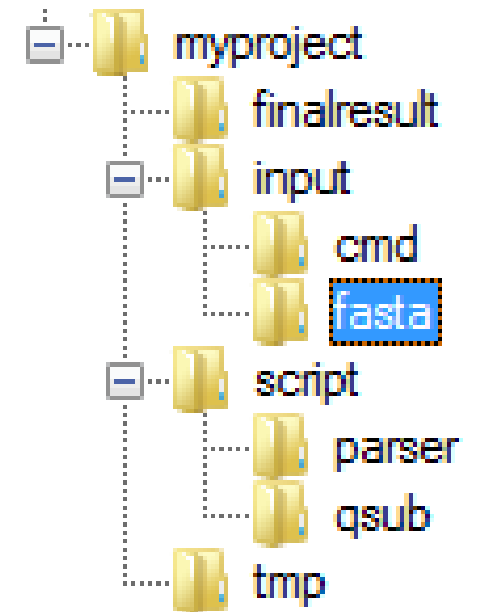
```
$ ln -s fichier destination
```

```
$ ln -s répertoire destination
```

```
$ ln -s fichier1 fichier2 destination
```

```
$ ll
drwxr-xr-x  2 plop sib 4096 Jan 11 17:03 finalresult
drwxr-xr-x  2 plop sib 4096 Jan 11 17:03 input
lrwxrwxrwx  1 plop sib   12 Apr 23 16:20 documents -> finalresult/documen
drwxr-xr-x  2 plop sib 4096 Jan 11 17:03 script
drwxr-xr-x  2 plop sib 4096 Jan 11 17:03 tmp
$ ll /usr/local/genome2/
lrwxrwxrwx  1 plop sib   14 Oct  8 2010 phylobayes -> phylobayes3.2f
drwxrwxr-x+ 6 plop sib  4096 Sep 30 2009 phylobayes2.3c
drwxrwxr-x+ 8 plop sib  4096 Sep 24 2009 phylobayes3.2c
drwxrwxr-x+ 8 plop sib  4096 Oct  8 2010 phylobayes3.2f
$ ll /usr/local/genome2/bin
```

- Créer un lien symbolique du répertoire script dans votre home
- Créer un lien du fichier test-TP.txt situé dans finalresult dans votre home
- Afficher le fichier test-TP.txt présent dans votre home
- Supprimer le fichier finalresult/test-TP.txt
- Observer



Corrections

```
$ cd  
$ ln -s myproject/script .
```


.bashrc

- Les Alias permettent de créer des raccourcis vers des commandes Linux

```
$ alias
alias la='ls -lart'
alias ll='ls -l '
alias ls='ls --color=tty'
```

- Un alias n'est actif que dans la console où on l'a lancé.
- Pour rendre durable un alias, il faut le renseigner dans le fichier .bashrc situé dans chaque home

```
$ la # raccourci pour ls -a
acteur.csv .gconf Musique
.bash_history .gconfd .nautilus
.bash_logout .gksu.lock .profile
.bashrc .gnome2 Public
```

- Cas pratique

```
$ grep ">" insulin.fas
>gi|163659904|ref|NM_000618.3| Homo sapiens insulin-like g
>gi|163659900|ref|NM_001111284.1| Homo sapiens insulin-lik
$ grep ">" --color insulin.fas
>gi|163659904|ref|NM_000618.3| Homo sapiens insulin-like g
>gi|163659900|ref|NM_001111284.1| Homo sapiens insulin-lik
```

- Editer le fichier .bashrc avec le programme gedit
 - Ajouter sous la mention : # User specific aliases and functions :
 - alias grep='grep --color'

```
$ source ~/.bashrc # pour que la console recharge .bashrc
```

```
$ grep ">" insulin.fas
>gi|163659904|ref|NM_000618.3| Homo sapiens insulin-like g
>gi|163659900|ref|NM_001111284.1| Homo sapiens insulin-lik
```

