

# Guide de survie à Linux



[www.southgreen.fr](http://www.southgreen.fr)

<https://southgreenplatform.github.io/trainings>



## The objectif!

Lancez vos analyses bioinformatiques sous Linux !



## Applications

Connaître les principales commandes sous Linux

- Se déplacer dans l'arborescence de fichier : *pwd*, *ls*, *cd*, *mkdir*, ...
- Se connecter à un serveur, transférer les données : *ssh*, *wget*
- Manipuler des fichiers : *head*, *tail*, *sort*, *cut*, *wc*, *grep*
- Lancer des logiciels bioinformatiques en ligne de commande



# Introduction

- **Système d'exploitation** réputé pour :
  - sa sécurité
  - ses mises à jour fréquentes
  - son prix et ses programmes gratuits

- Créé en 1991 par ***Linus Torvalds***

- Basé sur l'OS propriétaire Unix (1969)

- Linux ***gratuit*** et ***libre***

on peut avoir le code source, la “recette de fabrication”  
on peut copier, modifier, redistribuer



- **Système robuste et multi-plateforme**

- **Système multi-utilisateurs**

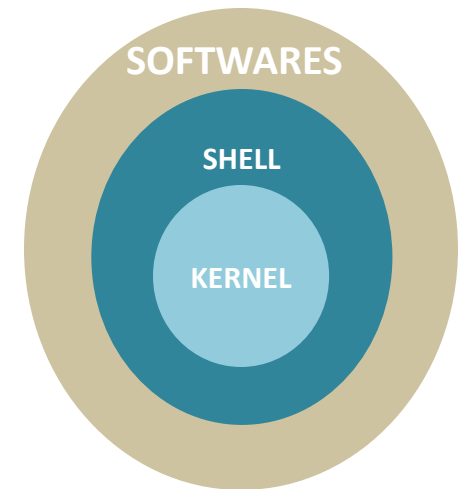
Plusieurs utilisateurs peuvent travailler en même temps

- **Système multi-tâches (processus/programmes)**

Chaque utilisateur peut lancer plusieurs programmes en même temps



**Distribution** : Noyau/Coeur + logiciels/programmes



- 2 façons d'utiliser linux :

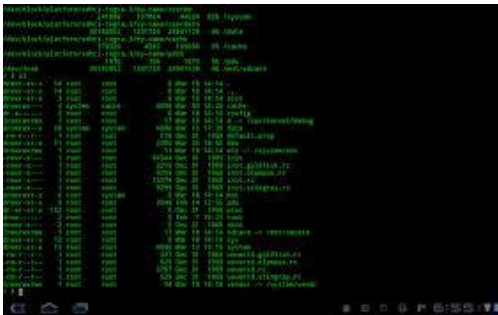
en *mode graphique*



- 2 façons d'utiliser linux :

en *mode graphique*

en *mode console*





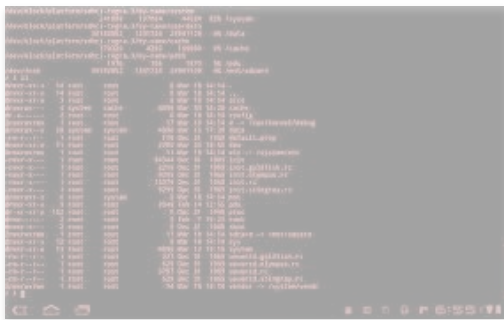


- Nombreux programmes rapides & puissants
- Facile de lier des commandes/programmes entre eux (workflow)
- Nombreux outils bioinformatique disponibles
- 90% des serveurs fonctionnent sous Linux

# Pourquoi utiliser Linux ?



- Nombreux programmes rapides & puissants
- Facile de lier des commandes/programmes entre eux (workflow)
- Nombreux outils bioinformatique disponibles
- 90% des serveurs fonctionnent sous Linux



**Pas d'interfaces  
graphiques**

**Convivialité de la ligne  
de commande ?**





Nécessité de la pratique et de l'expérience

⇔ Investissement non négligeable pour de bons résultats rapidement



# Premiers Pas sur Linux

Commandes relatives à l'arborescence de  
fichiers

**Toujours présent sur le terminal, juste avant de taper la commande**

**Prompt**

```
[tranchant@node6 data]$
```

Toujours présent sur le terminal, juste avant de taper la commande

**Prompt**

```
[tranchant@node6 data]$
```

Nom  
utilisateur

Nom  
serveur

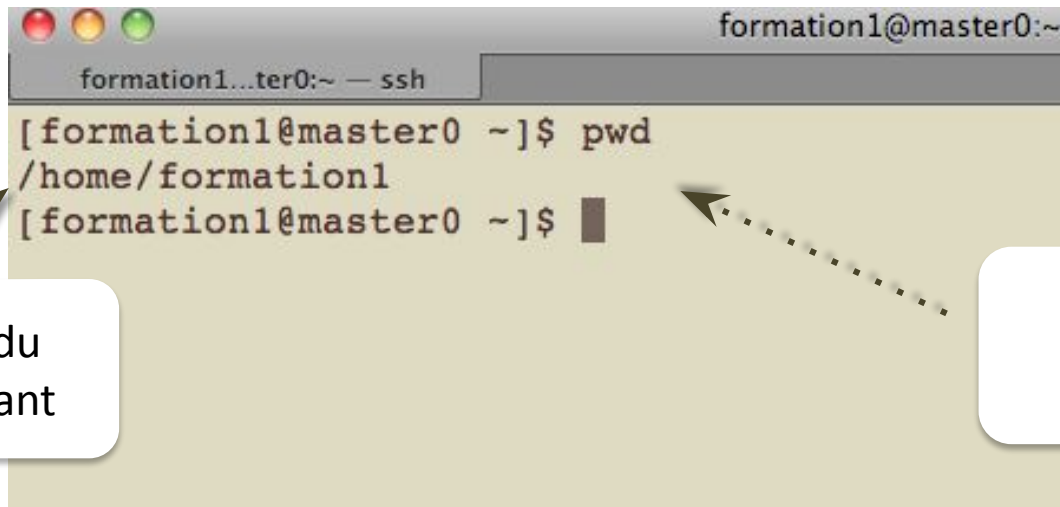
Répertoire  
courant

*commande [ -options ] [ arguments ]*

**pwd**

*Present Work Directory*

*affiche nom du répertoire courant (et son chemin complet)*



```
formation1@master0:~  
[formation1@master0 ~]$ pwd  
/home/formation1  
[formation1@master0 ~]$
```

The screenshot shows a terminal window titled 'formation1@master0:~'. The prompt is '[formation1@master0 ~]\$'. The user enters 'pwd' and the output is '/home/formation1'. The prompt then returns to '[formation1@master0 ~]\$'.

Affiche le nom du  
répertoire courant

Commande  
sans options et  
sans argument



**ls**  
list

*liste fichiers & répertoires présents dans le répertoire courant*



A terminal window titled 'formation1@master0:~ — ssh — 97x37'. The prompt is '[formation1@master0 ~]\$'. The command 'ls' has been entered, and the output shows 'data' in blue and 'scripts' in purple. A dashed arrow points from the 'ls' command in the text box to the 'ls' command in the terminal. Another dashed arrow points from the 'data' and 'scripts' output in the terminal to the text box 'Liste des fichiers du répertoire courant (par défaut)'.

```
formation1@master0:~ — ssh — 97x37
[formation1@master0 ~]$ ls
data  scripts
```

Liste des fichiers du répertoire courant (par défaut)

Commande sans options et sans argument

**ls -l**  
*list long*

*liste les fichiers avec des informations pour  
chaque fichier/répertoire*

Commande avec l'option l et le nom  
d'un répertoire en argument

```
[formation1@master0 ~]$ ls -l /home/
total 312
drwx-----  6 abate      sat      4096 12 mars   2012 abate
drwx-----  5 adam       ggr      4096 23 mars   2012 adam
drwx----- 31 admin     admin    4096  3 août   11:35 admin
drwx-----  9 alizon    ete      4096 21 août   14:23 alizon
drwx----- 12 alvaro-wis effecteurs 4096 17 juin   16:19 alvaro-wis
drwx-----  4 auguy     rhizogenesis 4096  2 mars   2012 auguy
drwx-----  5 ayoub    team1    4096 13 avril  2012 ayoub
drwx-----  5 beule     bdp      4096  8 oct.   17:49 beule
drwx-----  9 bouniol   ggr      4096  2 oct.   15:00 bouniol
drwx----- 10 castillo  bdp      4096 10 oct.   15:55 castillo
```

liste détaillée des  
fichiers

## Comment obtenir de l'aide sur une commande?

- avec l'option `--help` ou `-h` `ls --help`    `blastn -h`
- avec la commande `man` `man ls`
- avec la commande `whatis` `whatis ls`

```

tranchan — tranchant@master0:~ — ssh bioinfo-master.ird.fr -ltranchant — 93x22
...nt@master0:~ — ssh bioinfo-master.ird.fr -ltranchant  ~ — tranchant@master0:~ — -bash  ...  ~ — CLUSTER — -bash
[tranchant@master0 ~]$ whatis pwd
pwd (1)          - Afficher le nom du répertoire de travail en cours
pwd (n)          - Return the absolute path of the current working directory
pwd (lp)         - return working directory name
[tranchant@master0 ~]$

```

speaker notes

## Arborescence linux

**pwd**

Affiche le chemin absolu

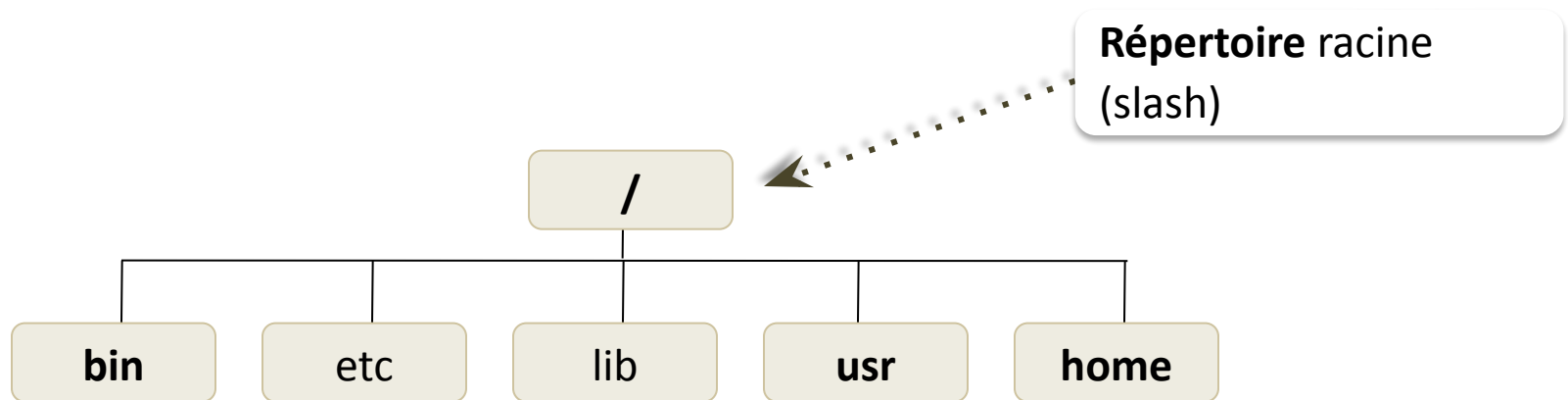
**ls**

Liste tous les fichiers/répertoires

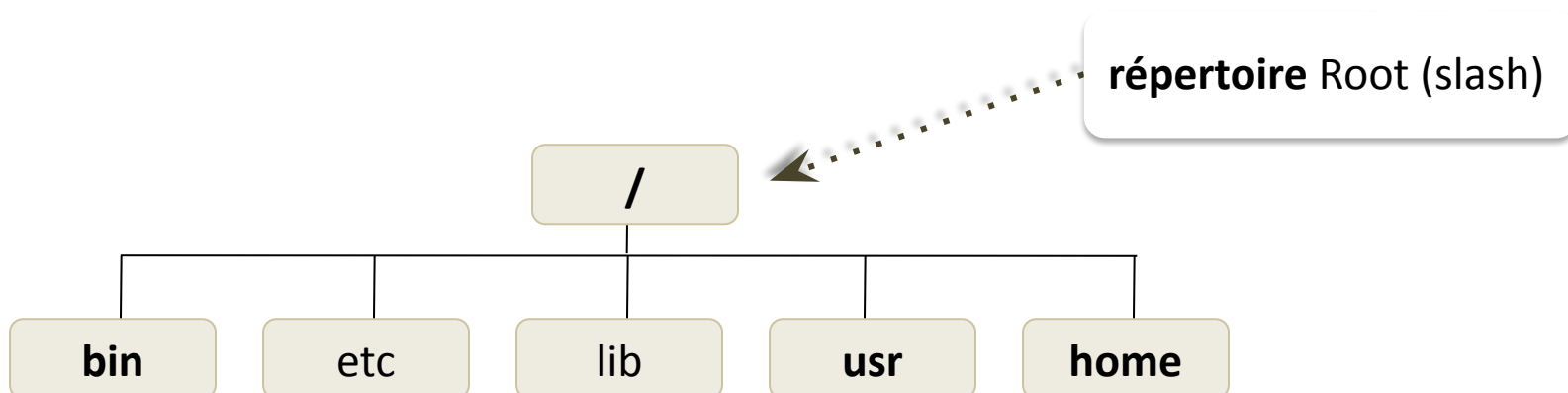
**ls -l**

Affiche toutes les informations sur les fichiers

- Arborescence = système de fichiers
- Répertoire racine appelé “/”



## Principaux répertoires



<b>/bin</b>	<b>principales commandes système</b>
<b>/usr, /opt</b>	<b>Applications et librairies ex : /usr/local</b>
<b>/usr/bin</b>	<b>Autres commandes</b>
<b>/home</b>	<b>répertoire des utilisateurs (1 par utilisateur, name=login)</b>

**Chemin (path)** : chemin d'accès à un fichier/répertoire

**Chemin (path)** : chemin d'accès à un fichier/répertoire

absolu

- chemin complet du fichier en partant du répertoire racine /



**Chemin** (path) : chemin d'accès à un fichier/répertoire

## absolu

- chemin complet du fichier en partant du répertoire racine /
- *commence toujours par /*
- **Toujours correct peu importe où on l'on travaille**

**Chemin** (path) : chemin d'accès à un fichier/répertoire

## absolu

- chemin complet du fichier en partant du répertoire racine /
- commence toujours par /
- Toujours correct peut importe où on l'on travaille

## relatif

- chemin défini par rapport à où l'on est dans l'arborescence

**Chemin** (path) : chemin d'accès à un fichier/répertoire

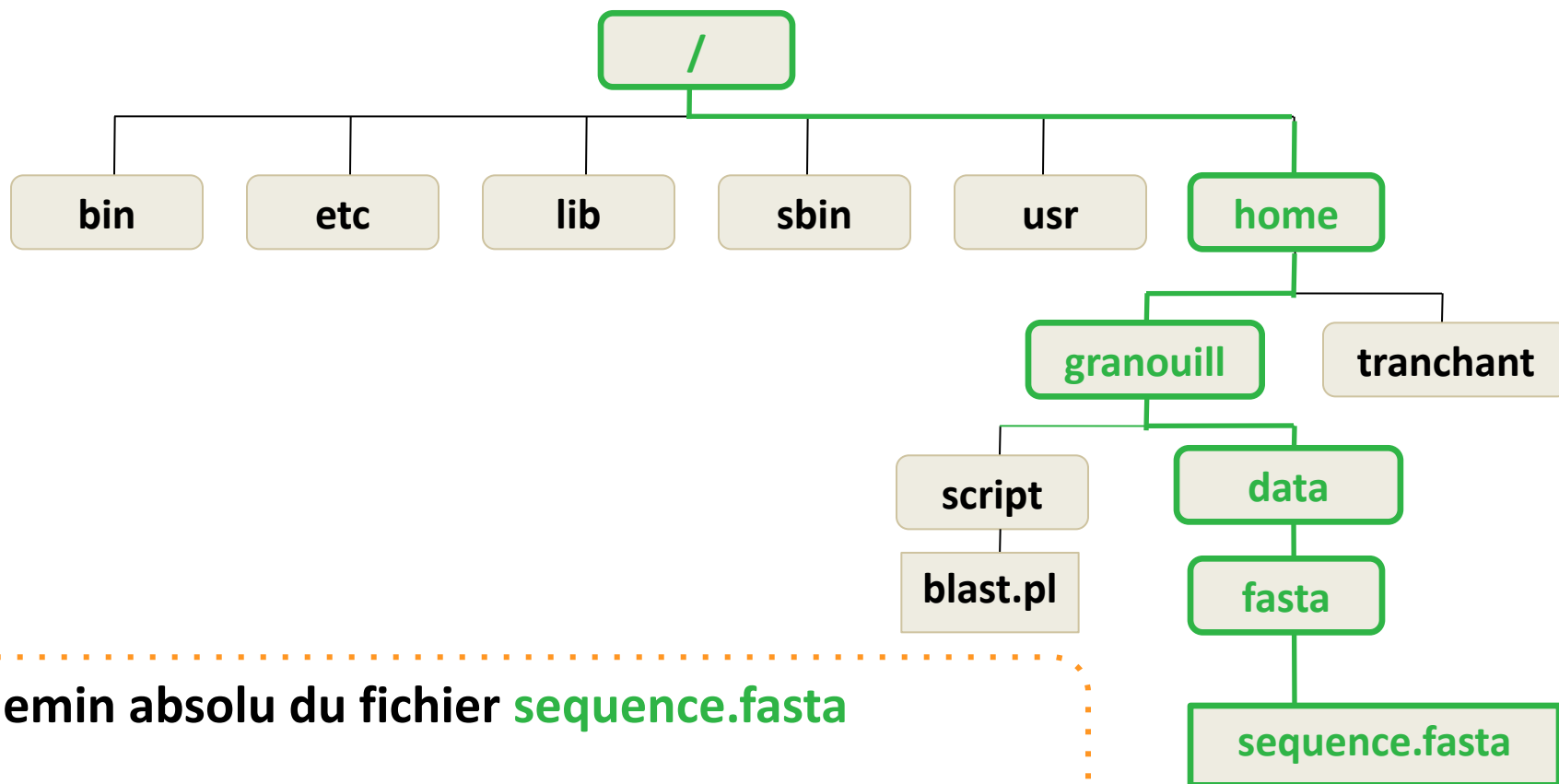
## absolu

- chemin complet du fichier en partant du répertoire racine /
- commence toujours par /
- Toujours correct peut importe où on l'on travaille

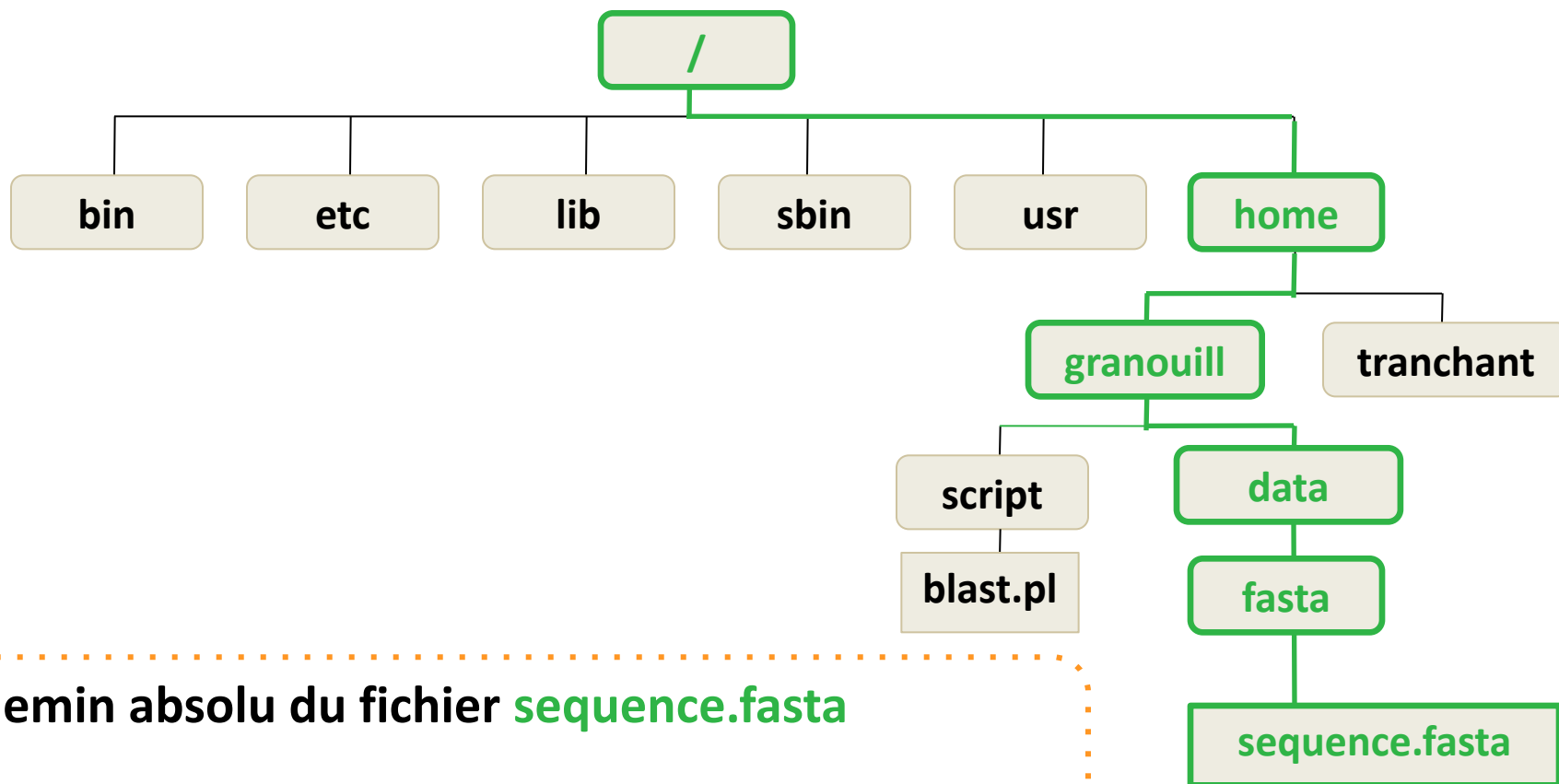
## relatif

- chemin défini par rapport à où l'on est dans l'arborescence
- *Ne commence jamais par /*
- **Change selon où l'on travaille**

- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où l'on travaille



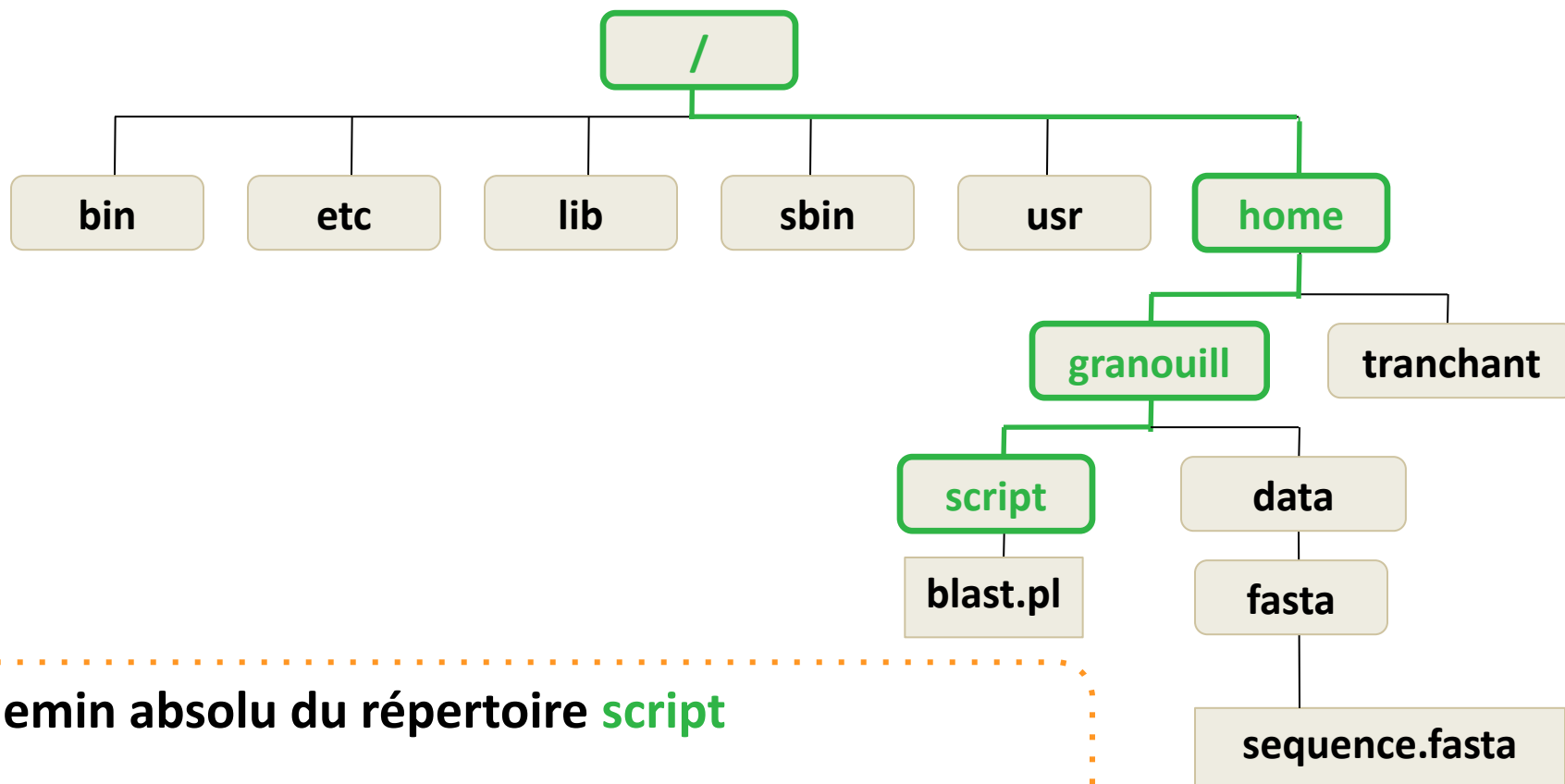
- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où l'on travaille



Chemin absolu du fichier **sequence.fasta**

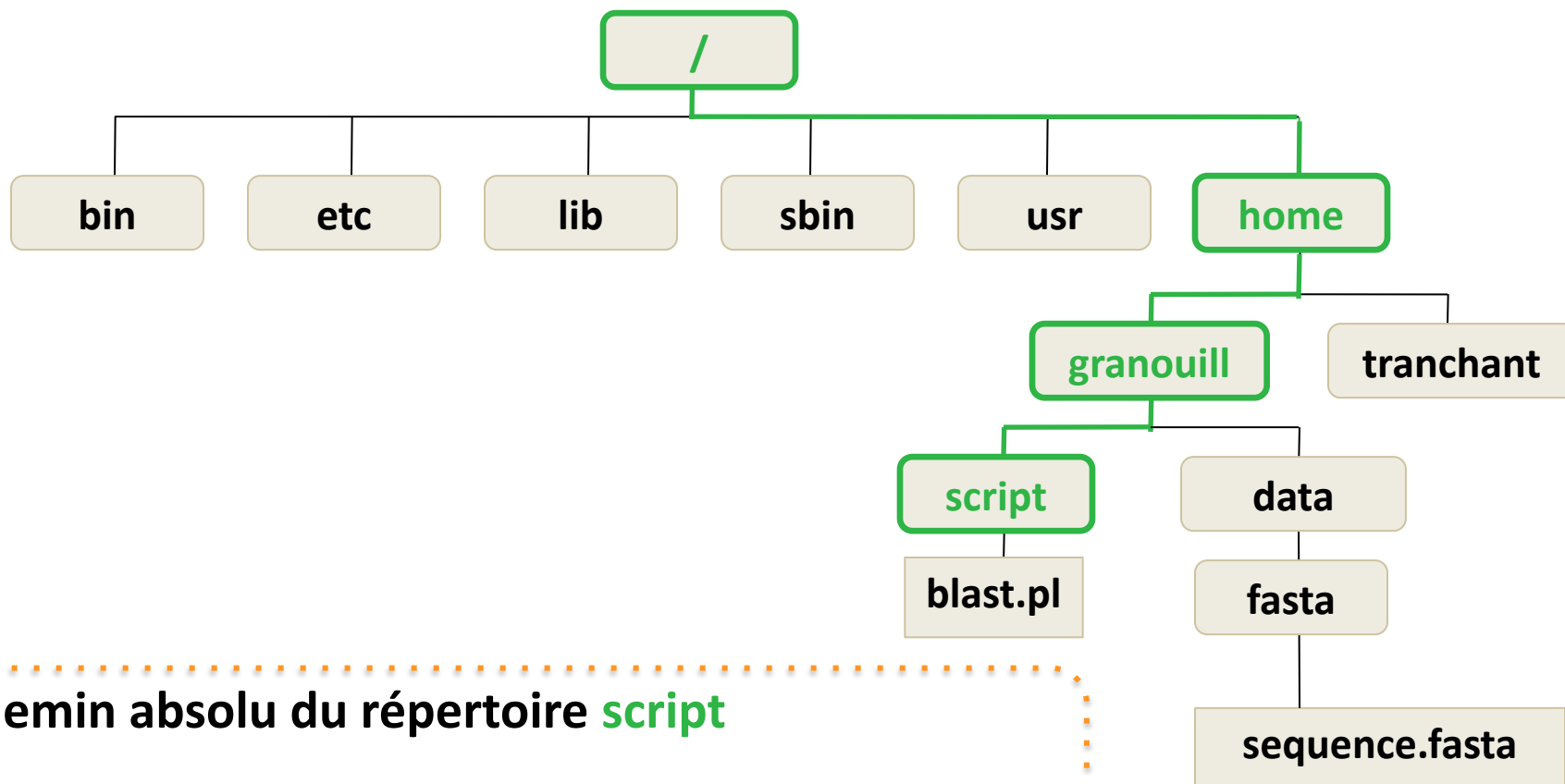
`/home/granouill/data/fasta/sequence.fasta`

- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où l'on travaille



Chemin absolu du répertoire **script**

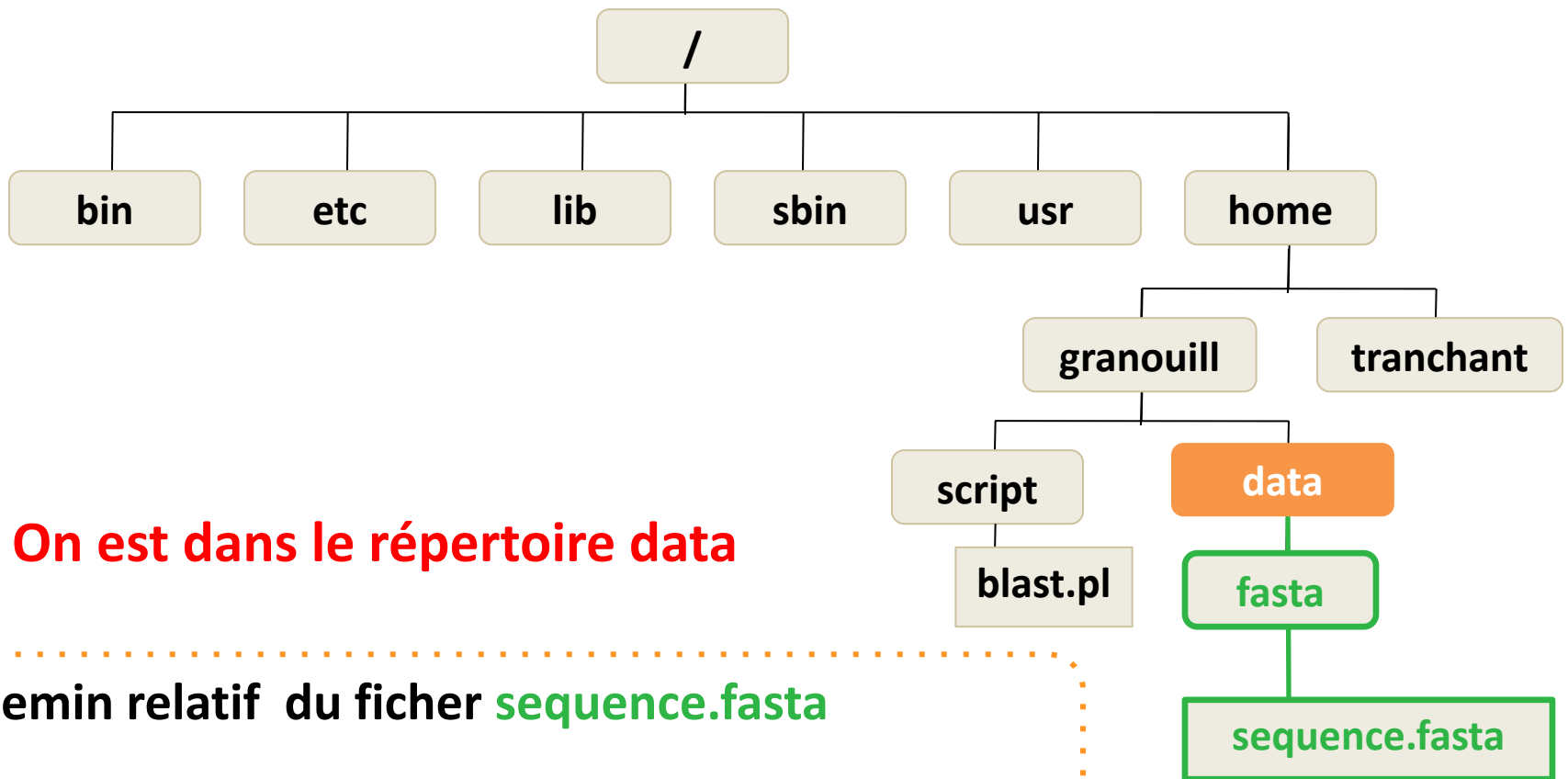
- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où l'on travaille



Chemin absolu du répertoire **script**

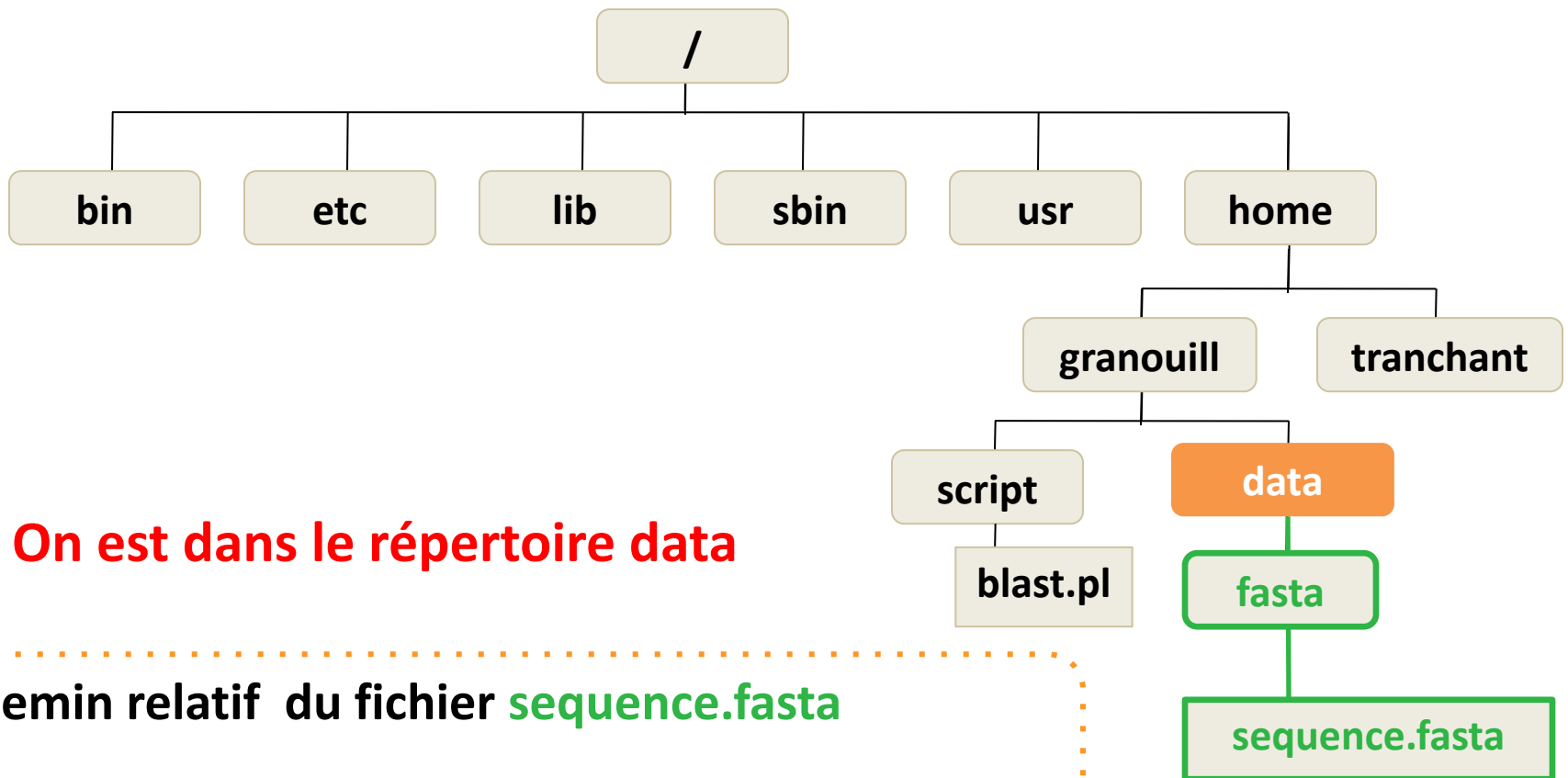
`/home/granouill/script`

- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**





- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**

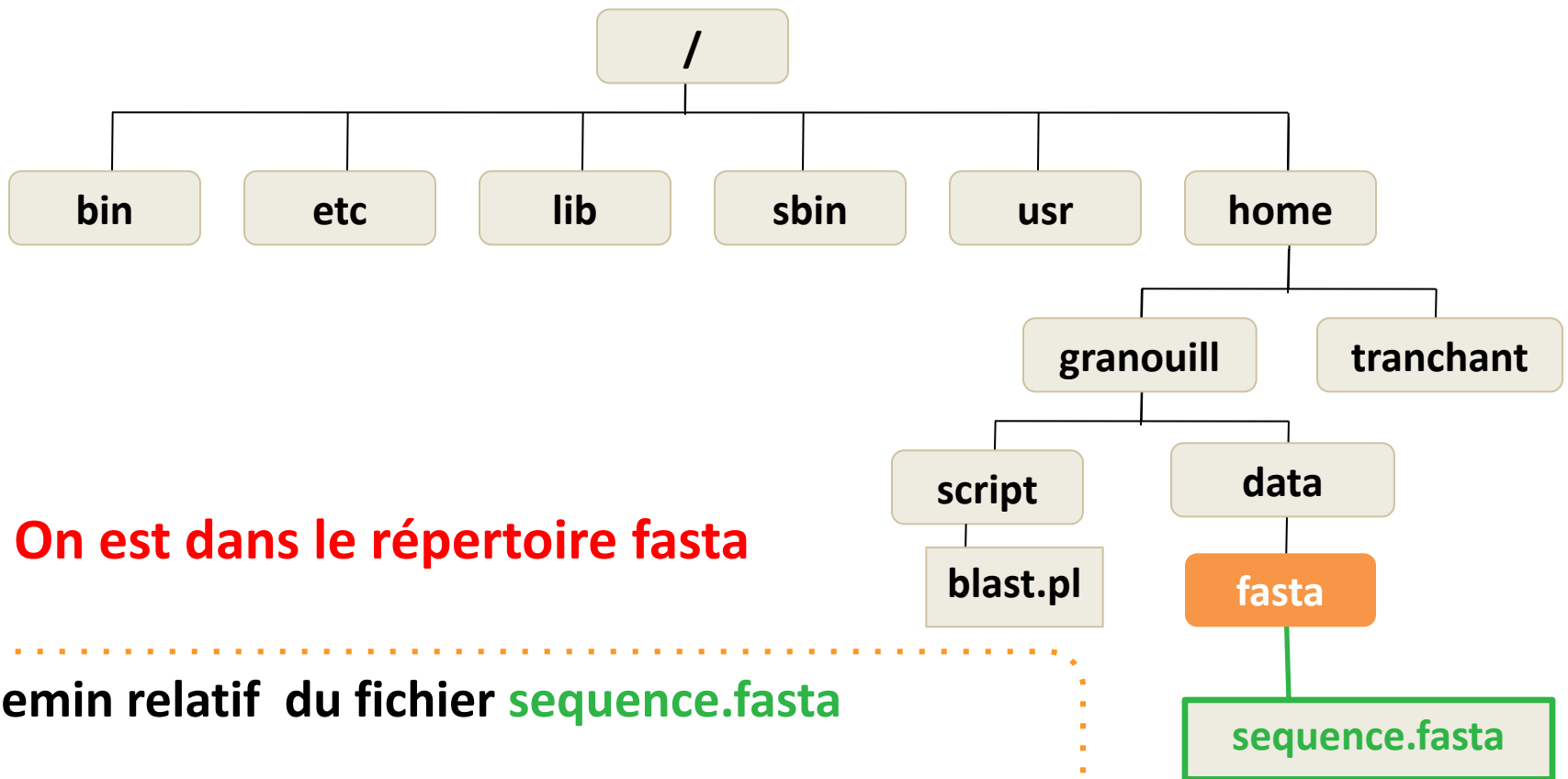


**On est dans le répertoire data**

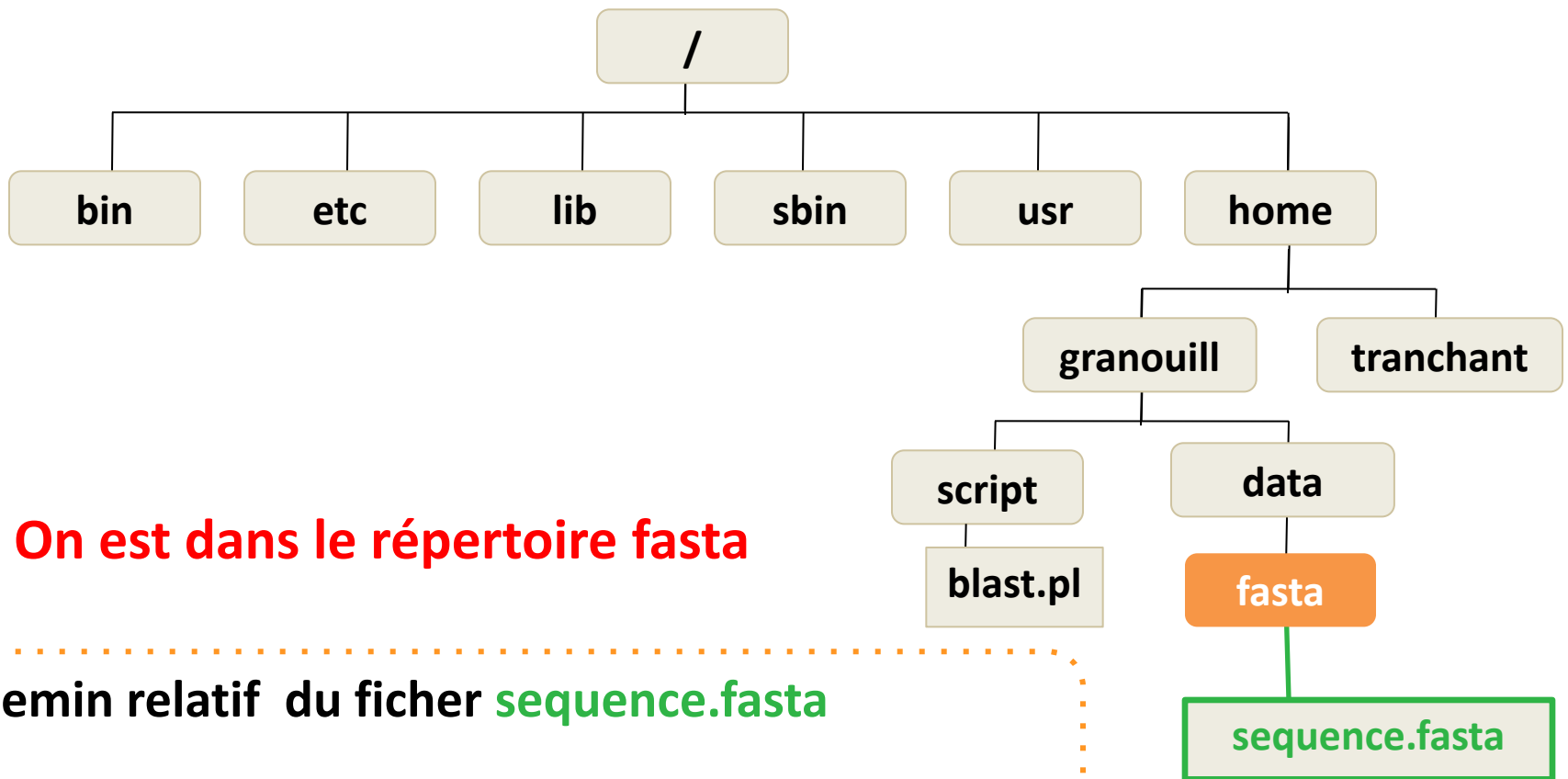
Chemin relatif du fichier **sequence.fasta**

**fasta/sequence.fasta**

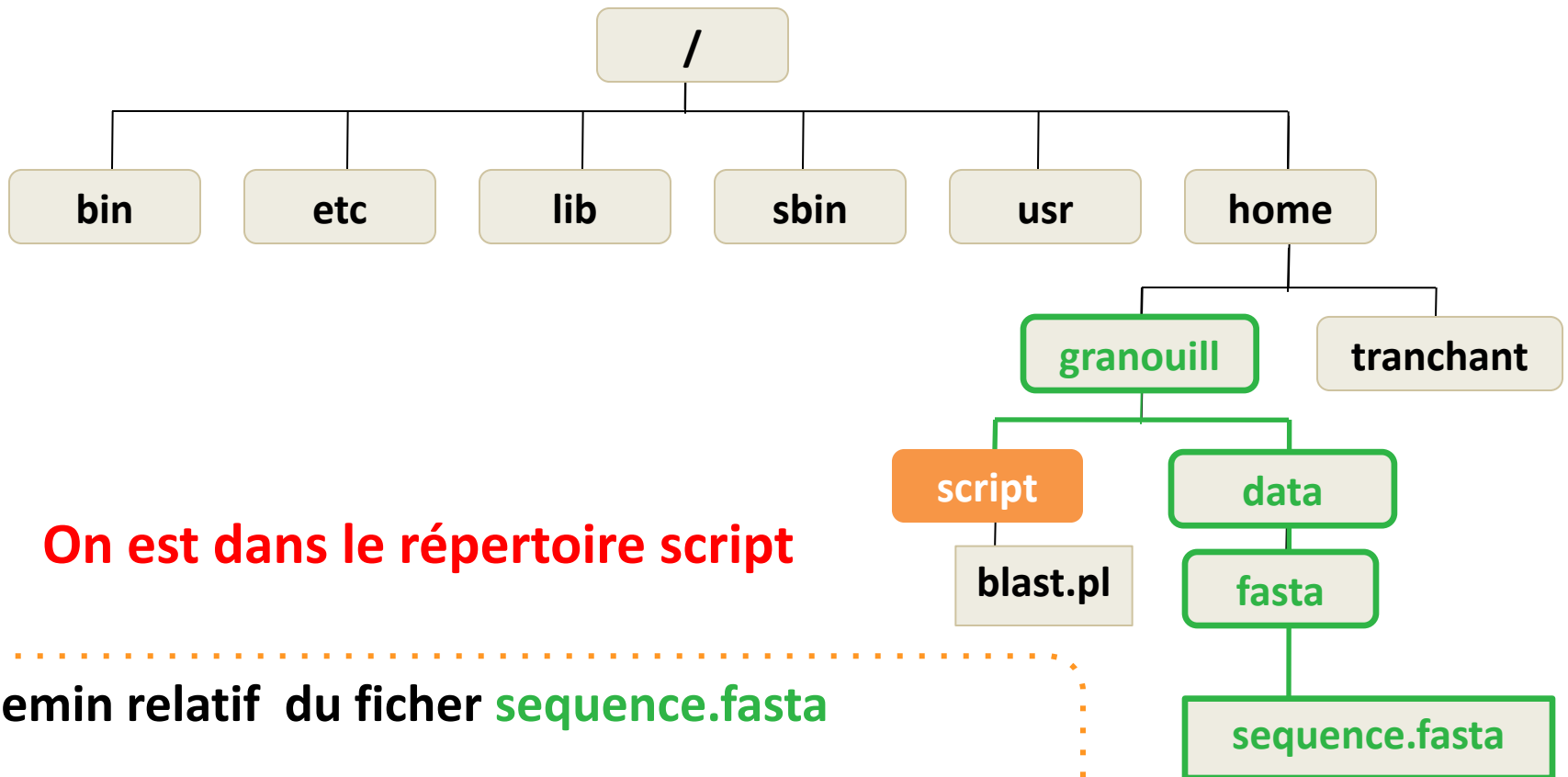
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



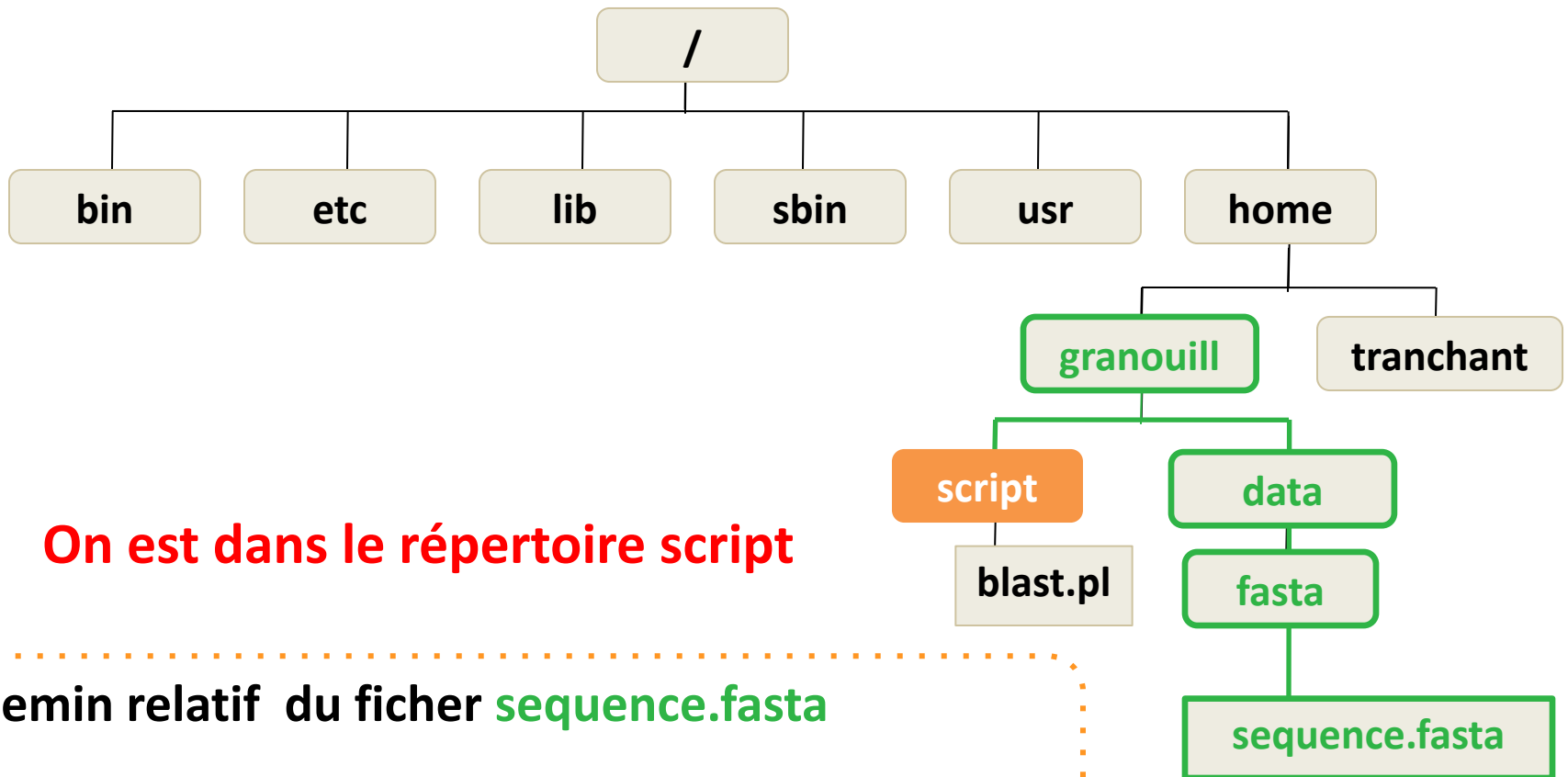
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



Chemin relatif du fichier **sequence.fasta**

**../data/fasta/sequence.fasta**

## Interagir avec les processus

**<Ctrl> + C** Arrêter le processus en cours sous le terminal

**<Ctrl> + Z**

## Tab completion

**<Tab>** Complète automatiquement le nom d'un fichier/  
répertoire qui est en cours de saisie (choix unique)

**<Tab><Tab>** Affiche la liste des différentes possibilités si le  
choix n'est pas unique

- Linux est sensible à la casse

Sequence.fasta  $\neq$  SEQUENCE.fasta  $\neq$  sequence.fasta

- Utiliser uniquement des lettres, chiffres et caractères \_ . –
- PAS d'espaces, accents et caractères spéciaux

& ~ # " ' { ( [ | ` \ ^ @ ) ] } \$ \* % ! / ; , ?

- Le suffixe des noms de fichiers (.txt, . fasta, .fa, .fq etc.) est optionnel



# Commandes relative à l'arborescence de fichiers

commande `cd`



**cd**

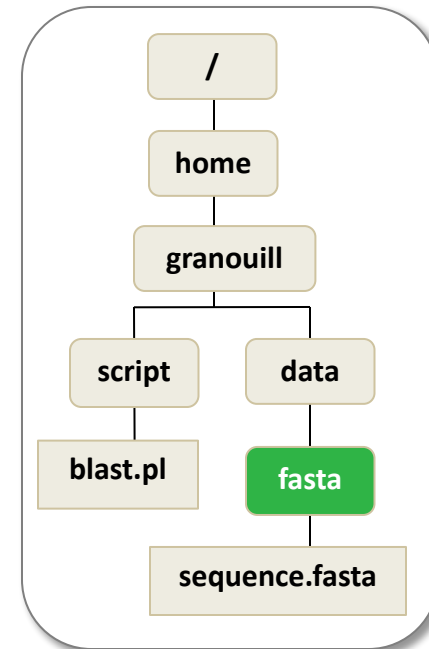
*pour se déplacer dans l'arborescence*

*Change Directory*

`cd nom_repertoire (chemin absolu ou relatif)`

`cd nom_repertoire(chemin absolu ou relatif)`

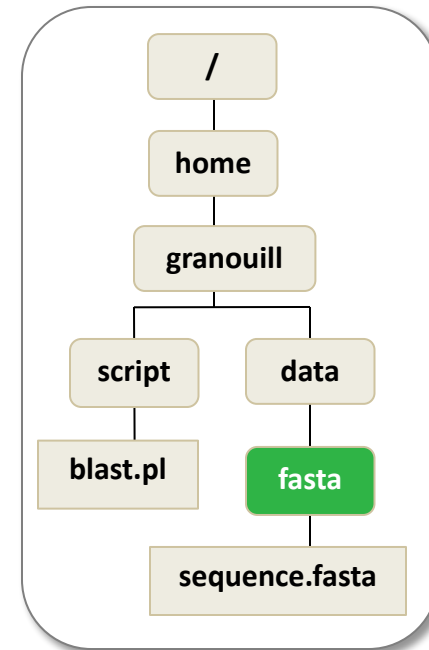
**Chemin absolu :**



*cd nom\_repertoire(chemin absolu ou relatif)*

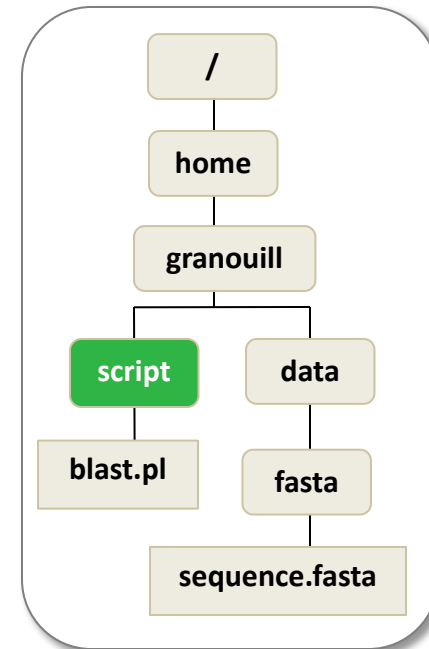
**Chemin absolu :**

**cd /home/granouill/data/fasta**



*cd nom\_repertoire(chemin absolu ou relatif)*

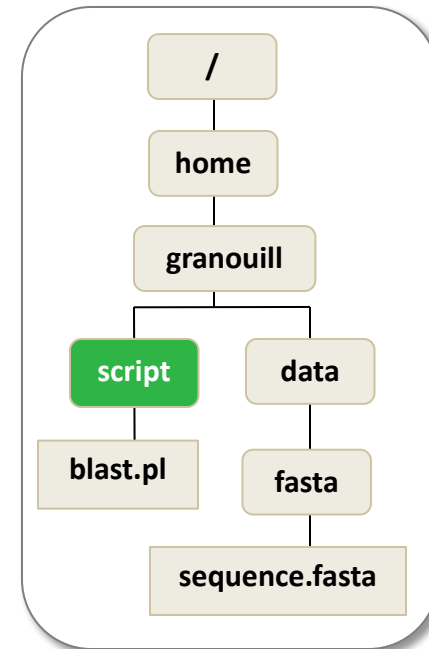
**Chemin absolu :**



*cd nom\_repertoire(chemin absolu ou relatif)*

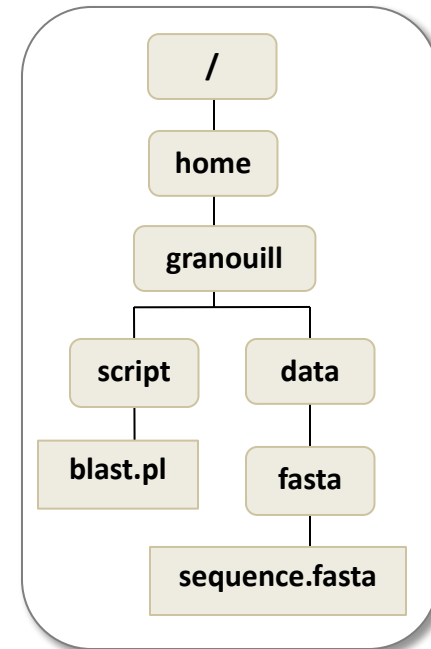
**Chemin absolu :**

**cd /home/granouill/script/**



`cd nom_repertoire(chemin absolu ou relatif)`

## Chemin relatif :



Commande	Se déplace dans
<code>cd directory_name</code>	<code>directory_name</code>
<code>cd</code>	<i>home directory</i>
<code>cd ..</code>	le répertoire parent
<code>cd ../..</code>	le répertoire parent du répertoire parent

Retour au répertoire de l'utilisateur courant home

Remonte d'1 répertoire

Remonte de 2 répertoires

## Arborescence linux

**pwd**

Affiche le chemin absolu

**ls**

Liste tous les fichiers/répertoires

**cd rep\_name**

Se déplace dans rep\_name

**mkdir rep\_name**

Crée un répertoire

**rm nom\_fichier**

Supprime un fichier

**cp fichier\_source repertoire\_cible**

Copie un fichier dans un répertoire

**cp fichier\_source nouveau\_fichier**

Copie un fichier sous un nouveau nom

**cp fichier\_source repertoire\_cible/nouveau\_fichier**

Mix des 2

**head** Affiche les premières lignes d'un fichier  
(n=10, 10 lignes par défaut) `head -n 20  
script.pl`

**tail** affiche les dernières lignes d'un fichier  
(n=10 par défaut) `tail -n 5 script.pl`

**wc** Compter le nombre de lignes, mots ou  
caractères d'un fichier `wc script.pl  
wc -l script.pl`



**grep**

*pour rechercher un motif dans un fichier*

```
grep [options] motif [file1, ... ]
```

**Pour rechercher un motif, le mettre entre simple/double quote  
=> ex : 'gene' ou "gene"**

**grep**

*pour rechercher un motif dans un fichier*

```
grep [options] motif [file1, ... ]
```

Option	Description
-i	Recherche le motif sans tenir compte de la casse
-c	Compte le nombre de lignes dans lesquelles le motif a été trouvées
-v	Affiche seulement les lignes sans le motif
-l	Affiche uniquement les noms de fichiers dans lesquels le motif a été trouvé



# Les entrées / sorties

**pour sauvegarder la sortie d'une  
commande dans un fichier**

```
cut -d: -f1 /etc/passwd > userName.txt
```

## Redirection

## Action

Command **>** file

- si le fichier n'existe pas : il sera créé
- si le fichier existe : efface le contenu

Command **>>** file

- si le fichier n'existe pas : il sera créé
- si le fichier existe : écrit à la fin du fichier

- La sortie d'une 1ère commande peut être envoyée comme input d'une 2ème commande
- Pour connecter/combinaire plusieurs commandes ensemble dans la même ligne de commande (sans fichier intermédiaire)
- Pipelines= *workflow*

*cmd1 | cmd2 | cmd3*

```
cut -d: -f1 /etc/passwd
```

```
Root
```

```
troot
```

```
iroot
```

```
ctroot
```

```
//
```

```
cut -d: -f1 /etc/passwd
```

```
Root  
troot  
iroot  
ctroot  
//
```

```
cut -d: -f1 /etc/passwd | sort
```

```
abate  
adm  
adroot  
ais  
#albar  
alvaro-wis  
anthony  
apache
```

```
cut -d: -f1 /etc/passwd
```

```
Root  
troot  
iroot  
ctroot  
//
```

```
cut -d: -f1 /etc/passwd | sort
```

```
abate  
adm  
adroot  
ais  
#albar  
alvaro-wis  
anthony  
apache
```

```
cut -d: -f1 /etc/passwd | sort | head
```



```
cut -d: -f1 /etc/passwd
```

```
Root  
troot  
iroot  
ctroot  
//
```

```
cut -d: -f1 /etc/passwd | sort
```

```
abate  
adm  
adroot  
ais  
#albar  
alvaro-wis  
anthony  
apache
```

```
cut -d: -f1 /etc/passwd | sort > /etc/passwd.sort
```



# Autres commandes utiles

## Compresser des fichiers **tar, gzip**

```
tar -zcvf tarfile.tar.gz dirToCompress  
gzip fileToCompress
```

## Décompresser une archive **gunzip, tar**

```
gunzip file.gzip  
tar -xvf file.tar  
tar -zxvf file.tar.gz  
gzip -d file.gz
```

## Afficher le contenu d'une archive **zcat**

```
zcat data.txt.gz
```

## Recherche d'expression/motif dans une fichier compressé: **zgrep**

```
zgrep 'NM_000020' data.gz
```

# Merci pour votre attention !



Le matériel pédagogique utilisé pour ces enseignements est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions (BY-NC-SA) 4.0 International:

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

## Commande ls -l

```
$ ls -l filename  
drwxrwxrwx 3 user user 4096 2012-02-11 20:21 file_name
```

Permissions

Proprio

Groupe

Taille

Heure et date de la dernière modification

### Interprétation/Légendes des permissions

Type

- : fichier normal

**d** : répertoire

**l** : lien symbolique

## ls -l command

Permissions

`drwxrwxrwx 3 user user 4096 2012-02-11 20:21 file_name`

other  
group  
user

3 classes

## 3 types of permissions :

Permission	File	Directory
Read r	Ouverture et Lecture	Lister et copier les fichiers
Write w	Modifier et supprimer	Manipuler le contenu : copier, créer, modifier, écraser
Execution x	Executer le fichier	Accès seulement au fichier pour l'exécuter

commande pour la gestion des permissions : **chmod**

```
chmod <perm> file_name
```

Chaque permission = 1 valeur

<b>R</b>	<b>4</b>
<b>W</b>	<b>2</b>
<b>X</b>	<b>1</b>
none	0

## Exemple

```
chmod 740 script.sh
```

```
# Owner=rwx Group=r-- Other=---
```

```
chmod 755 script.sh
```

```
# Owner=rwx Group=r-x Other=r-x
```

chmod, ls

Donner le nom du propriétaire, du groupe et les droits des fichiers contenus dans le répertoire “~/Data/454-projet1/raw”

Changer les droits du fichier Scripts/blast.pl pour qu’il soit :  
en lecture/écriture pour le groupe,  
lecture/écriture/exécution  
pour le propriétaire  
et lecture au public

