

Analyse de données de séquençage haut débit

Formation bioinformatique

Thiès - Sénégal, 4 au 12 Octobre 2018

LOGO CERAAS

Programme de la formation

1^{er} au 3 octobre

Bertrand, Ndomassi

Installation du cluster de CERAAS & formation à l'installation/
administration de cluster

4 octobre

Bertrand, Ndomassi

Christine

Introduction à linux et aux
lignes de commandes

5 octobre (matin)

Bertrand, Ndomassi

Christine

Initiation à l'utilisation du
cluster bioinformatique du
CERAAS

Programme de la formation

Lundi 8 octobre

*François, Christine,
Ndomassi*

- Présentation des technologies de séquençages
- Quelques cas d'utilisation
- Manipulations avancées de données sous linux
- 1^{ère} analyses de données de séquençage

Mardi 9 octobre

*François, Christine
Manuel, Gautier*

- Introduction aux gestionnaires de workflow TOGGLE & Galaxy
- Mapping (TOGGLE)
- Détection de SNPs (TOGGLE)

Programme de la formation

Mercredi 10 oct.

*François, Christine
Manuel, Gautier*

- Détection de SNP (filtres, annotation)
- Détection de variants structuraux
- Post analyse de données SNPs sous Galaxy

Jeudi 11 octobre

Manuel, Gautier

- Post analyse de données SNPs sous Galaxy
- Visualisation de variants structuraux sous Galaxy

Vendredi 12 oct.

*François, Christine
Manuel, Gautier*

- Questions & discussions



Carte France / SG



South Green

bioinformatics platform



www.southgreen.fr



<https://github.com/SouthGreenPlatform>



The South Green portal: a comprehensive resource for tropical and Mediterranean crop genomics, Current Plant Biology, 2016

Session de formation 2018



- Toutes nos formations :
<https://southgreenplatform.github.io/trainings/>
- Environnement de travail : [Logiciels à installer](#)

Guide de survie à Linux

www.southgreen.fr

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

Thiès - Sénégal, 4 au 12 Octobre 2018



LOGO CERAAS

The objectif!

Lancez vos analyses sous Linux !



Après ce module, vous serez capable de :

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



Introduction

- **Système d'exploitation** réputé pour :
 - sa sécurité
 - ses mises à jour fréquentes
 - son prix et ses programmes gratuits
- Crée en 1991 par *Linus Torvalds*
- Basé sur l'OS propriétaire Unix (1969)
- Code source de linux **gratuit** et **libre** : copier, modifier, redistribuer



- **OS robuste et multi-plateforme**

(ordinateur, serveur, android...)



- **Système multi-utilisateurs**

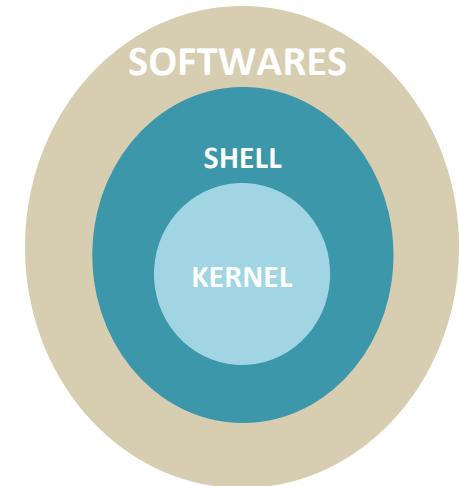
Plusieurs utilisateurs peuvent travailler en même temps

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

Chaque utilisateur peut lancer plusieurs programmes en même temps

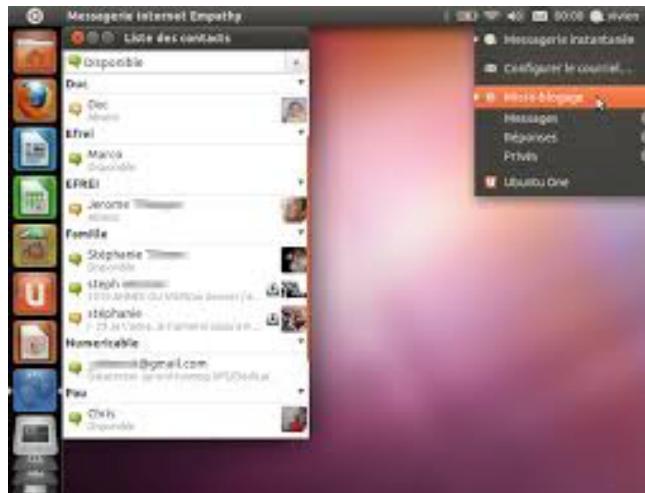
Distribution sous Linux

Distribution : Noyau/Coeur + logiciels/programmes



- 2 façons d'utiliser linux :

en *mode graphique*



- ## ■ 2 façons d'utiliser linux :

en *mode graphique*

en ***mode console*** (terminal)



Pourquoi utiliser Linux ?



- Nombreux programmes rapides & puissants
- Facile de lier des commandes/programmes entre eux (workflow)
- Nombreux outils bioinformatique disponibles
- Pas besoin de ressources matérielles importantes
- 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
 - Pas besoin de ressources matérielles importantes
 - 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**

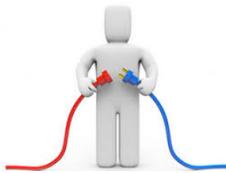


Environnement de travail

*Comment copier un fichier de son PC sur
le serveur ?*



Copier un fichier de son PC sur le serveur ?

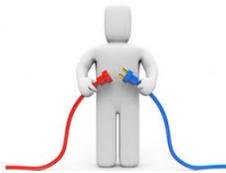


- En se connectant sur un serveur linux distant de son portable windows ou mac via le **protocole sftp**

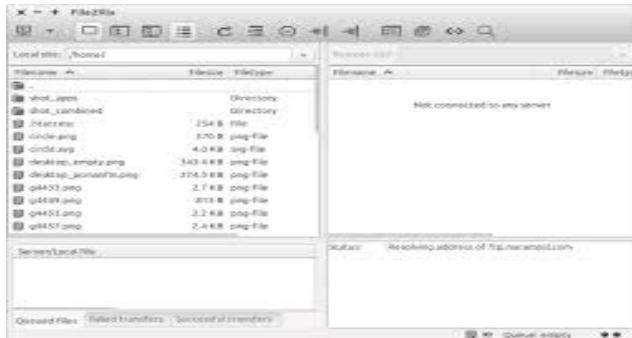
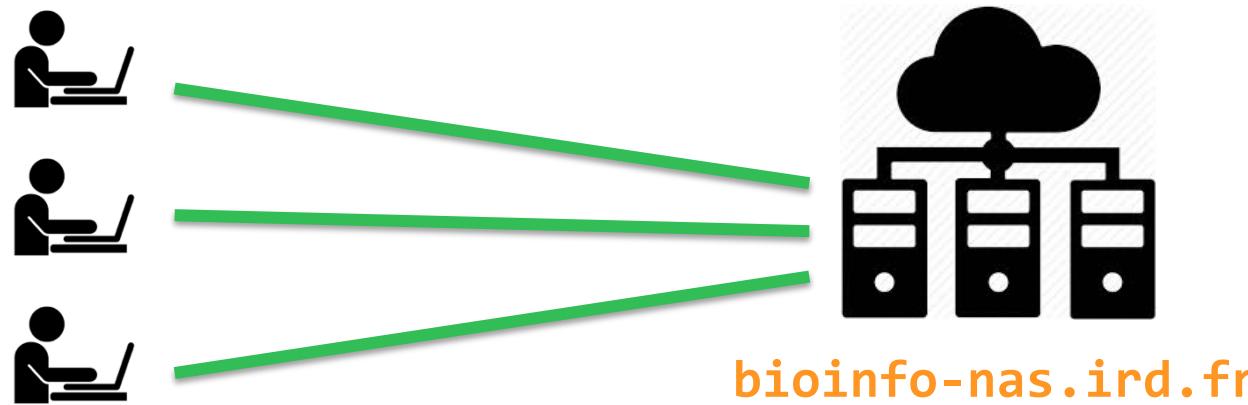




Copier un fichier de son PC sur le serveur ?



- En se connectant sur un serveur linux distant de son portable windows ou mac via le **protocole sftp**



- Capture écran filezilla



Practice

filezilla, sftp

1

Go to [Practice 1](#) on our github

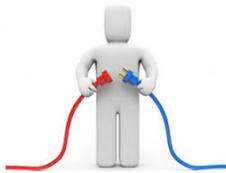


Environnement de travail

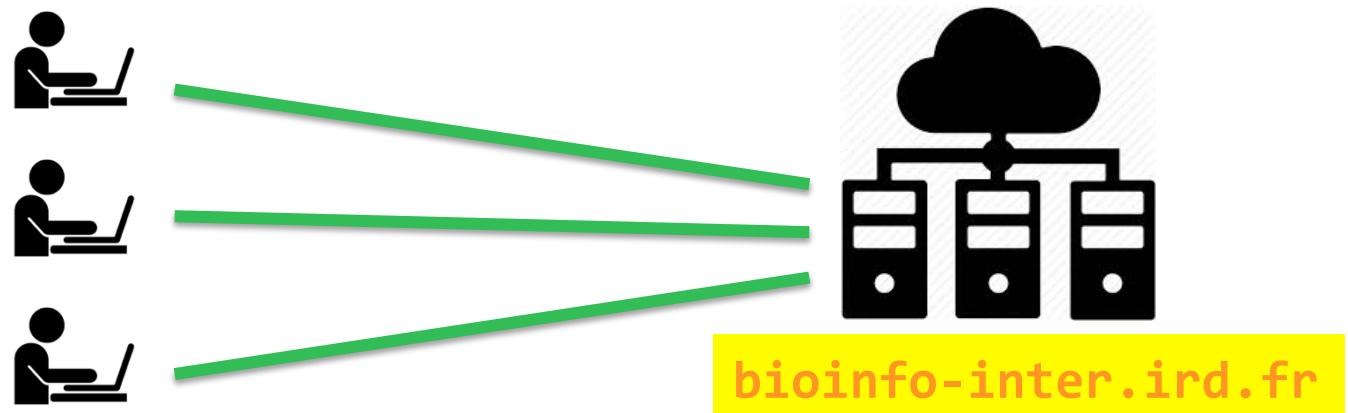
Comment travailler sur le serveur ?



Comment travailler sur le serveur ?

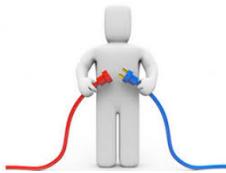


- En se connectant sur un serveur linux distant de son portable windows ou mac via le **protocole ssh**

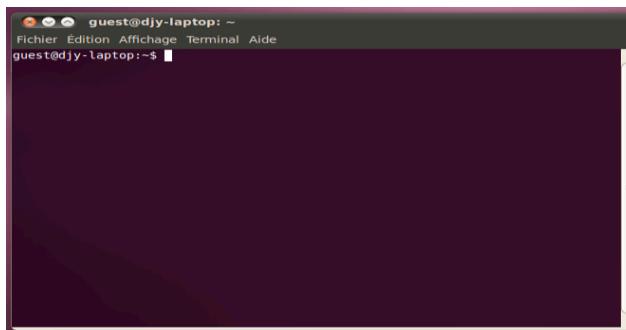




Comment travailler sur le serveur ?



- En se connectant sur un serveur linux distant de son portable windows ou mac via le **protocole ssh**





Practice

putty,
terminal, ssh

2

Go to [Practice 2](#) on our github



Premiers Pas sur Linux

**Se déplacer dans l'arborescence de fichiers et
manipuler des fichiers/répertoires**

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

Syntaxe d'une commande

commande [-options] [arguments]

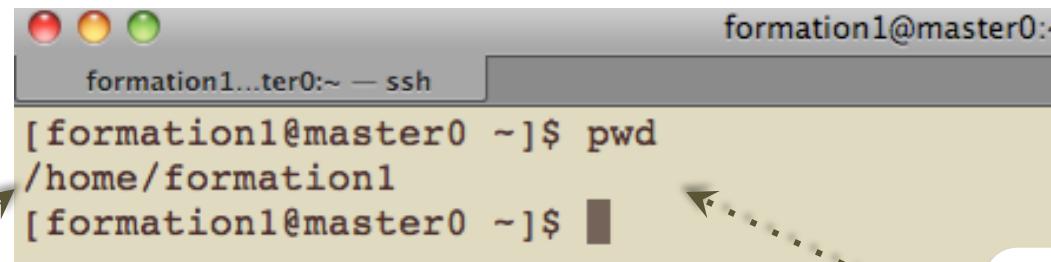
1ère commande

pwd

pwd

Present Work Directory

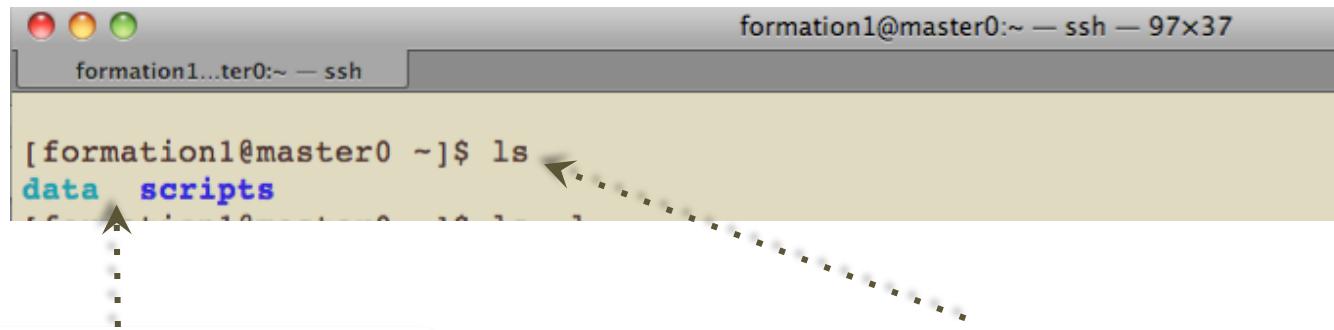
affiche nom du répertoire courant (et son chemin complet)



A screenshot of a terminal window titled "formation1...ter0:~ - ssh". The window shows the command [formation1@master0 ~]\$ pwd followed by the output /home/formation1. The terminal has a dark grey background and light grey text. The title bar is white with colored window control buttons.

Affiche le nom du répertoire courant

Commande sans options et sans argument

ls
list*liste fichiers & répertoires présents dans le répertoire courant*

A screenshot of a terminal window titled "formation1@master0:~ — ssh — 97x37". The window shows the command [formation1@master0 ~]\$ ls being run, followed by the output "data scripts". A dotted arrow points from the text "Liste des fichiers du répertoire courant (par défaut)" to the "data" file in the terminal output.

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 ayouba    team1       4096 13 avril 2012 ayouba
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?

Arborescence linux

pwd	Affiche le chemin absolu
ls	Liste tous les fichiers/répertoires
ls -l	Affiche toutes les informations sur les fichiers



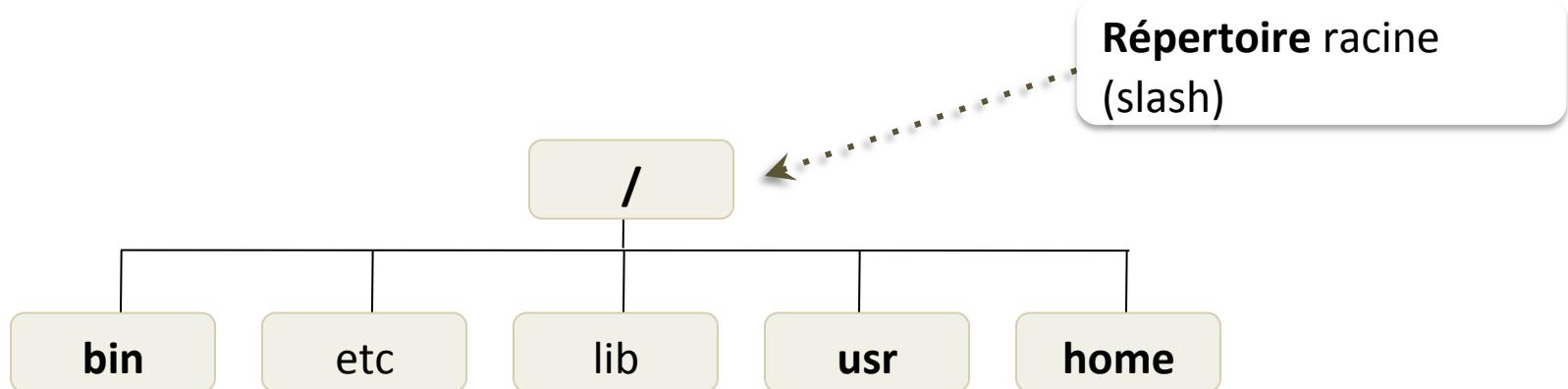
Practice

prompt, pwd

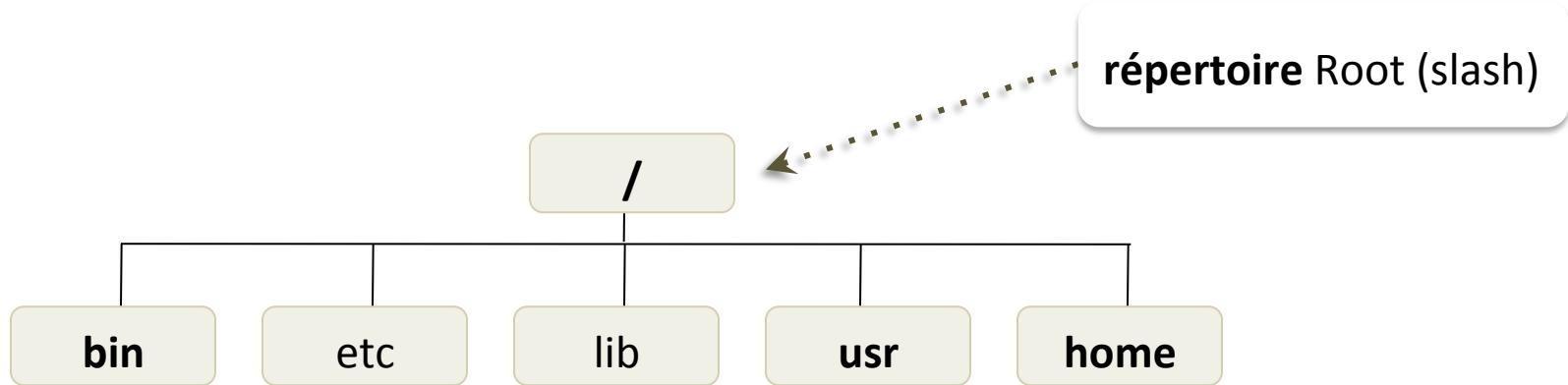
3

Go to [Practice 3](#) on our github

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



Principaux répertoires



/bin	commandes principales
/usr, /opt	Applications et librairies
/usr/bin	Autres commandes
/home	répertoire des utilisateurs (1 par utilisateur/login)

Chemin d'un fichier

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, peut 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 où on est dans l'arborescence

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

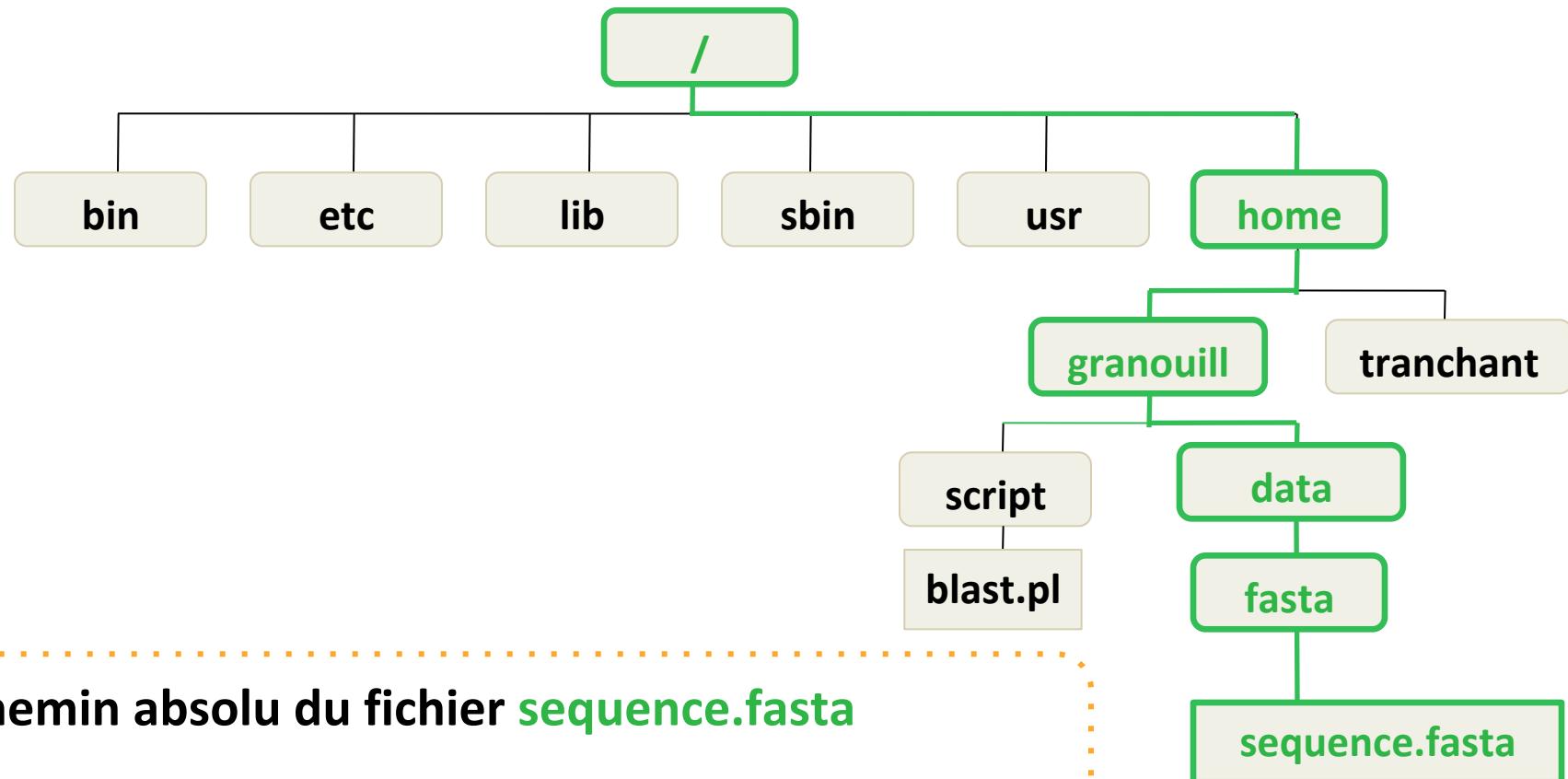
absolu

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

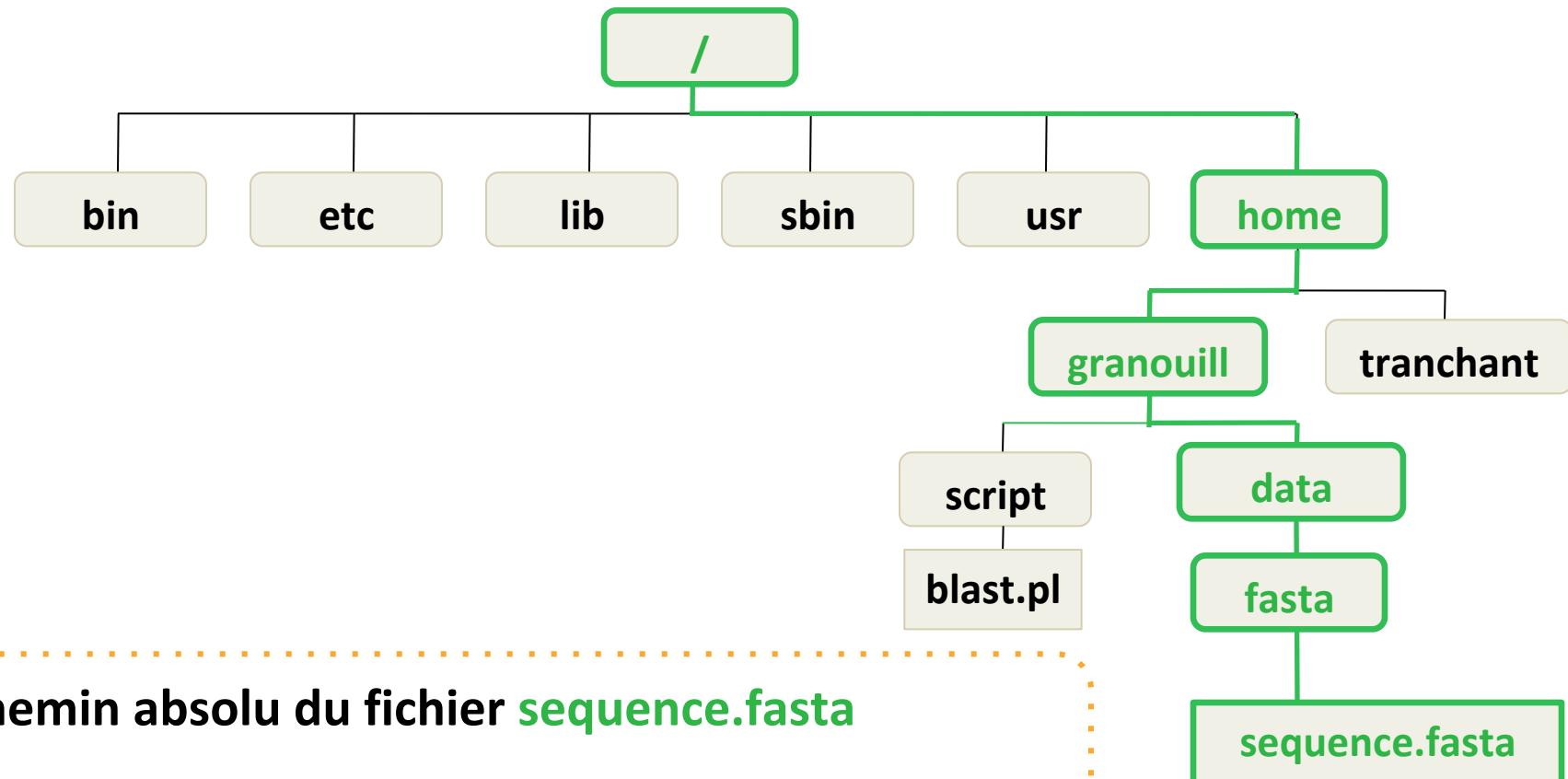
relatif

- chemin défini par rapport où on est dans l'arborescence
- *ne commence jamais par /*
- **change selon on l'on travaille**

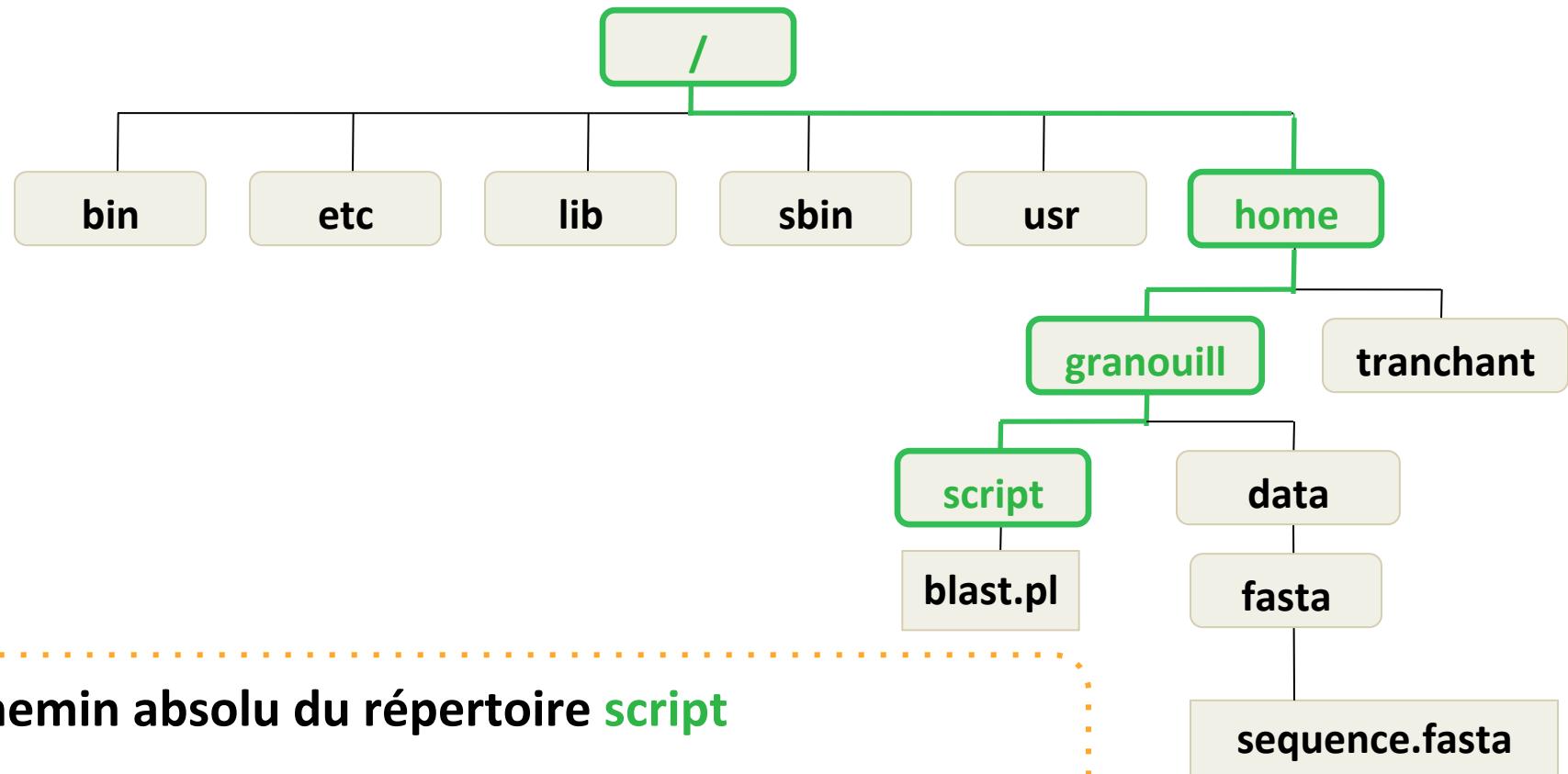
- Commence toujours par / (le répertoire racine)
- **Toujours correct peu importe où on l'on travaille**



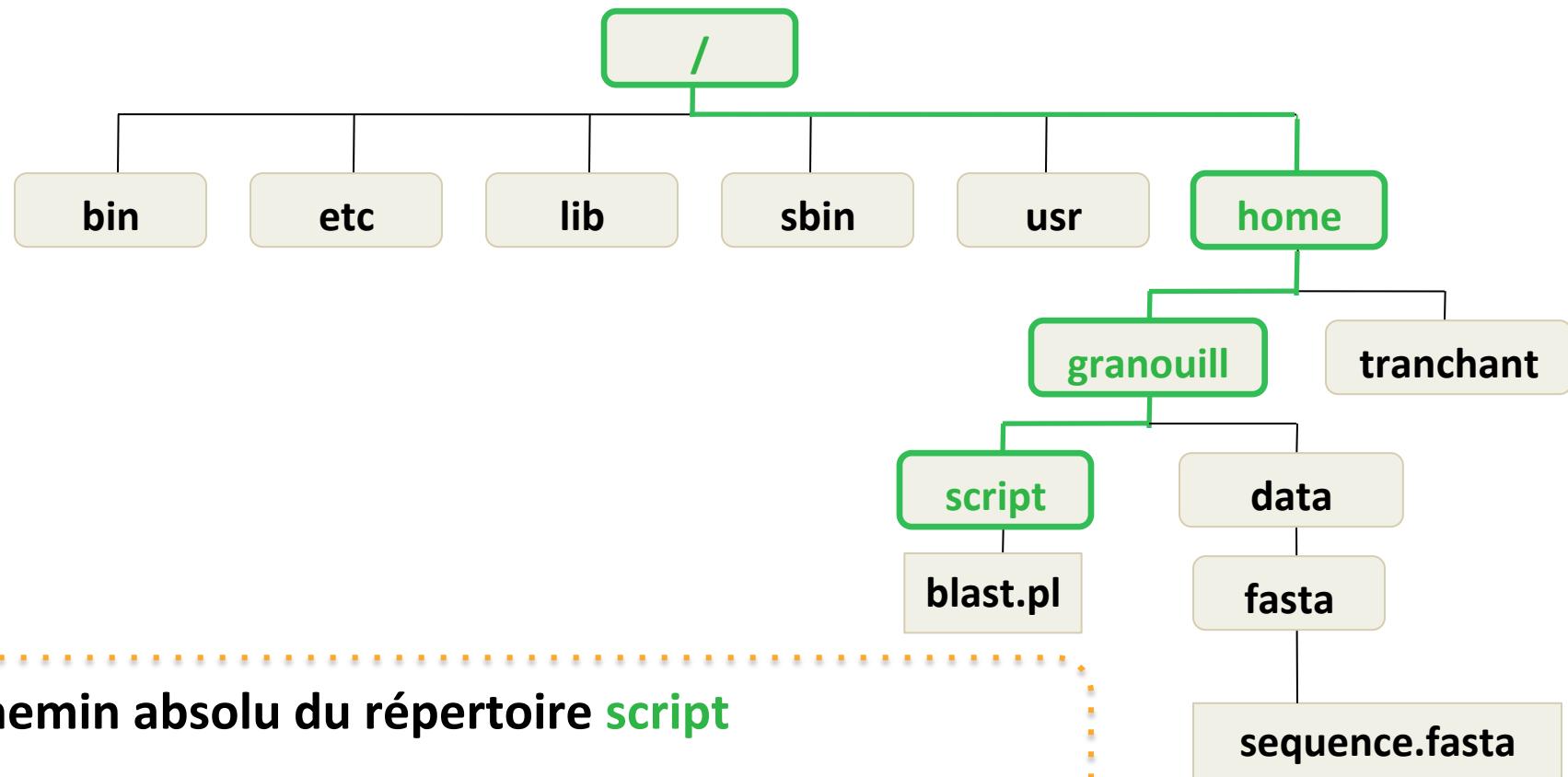
- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où on l'on travaille**



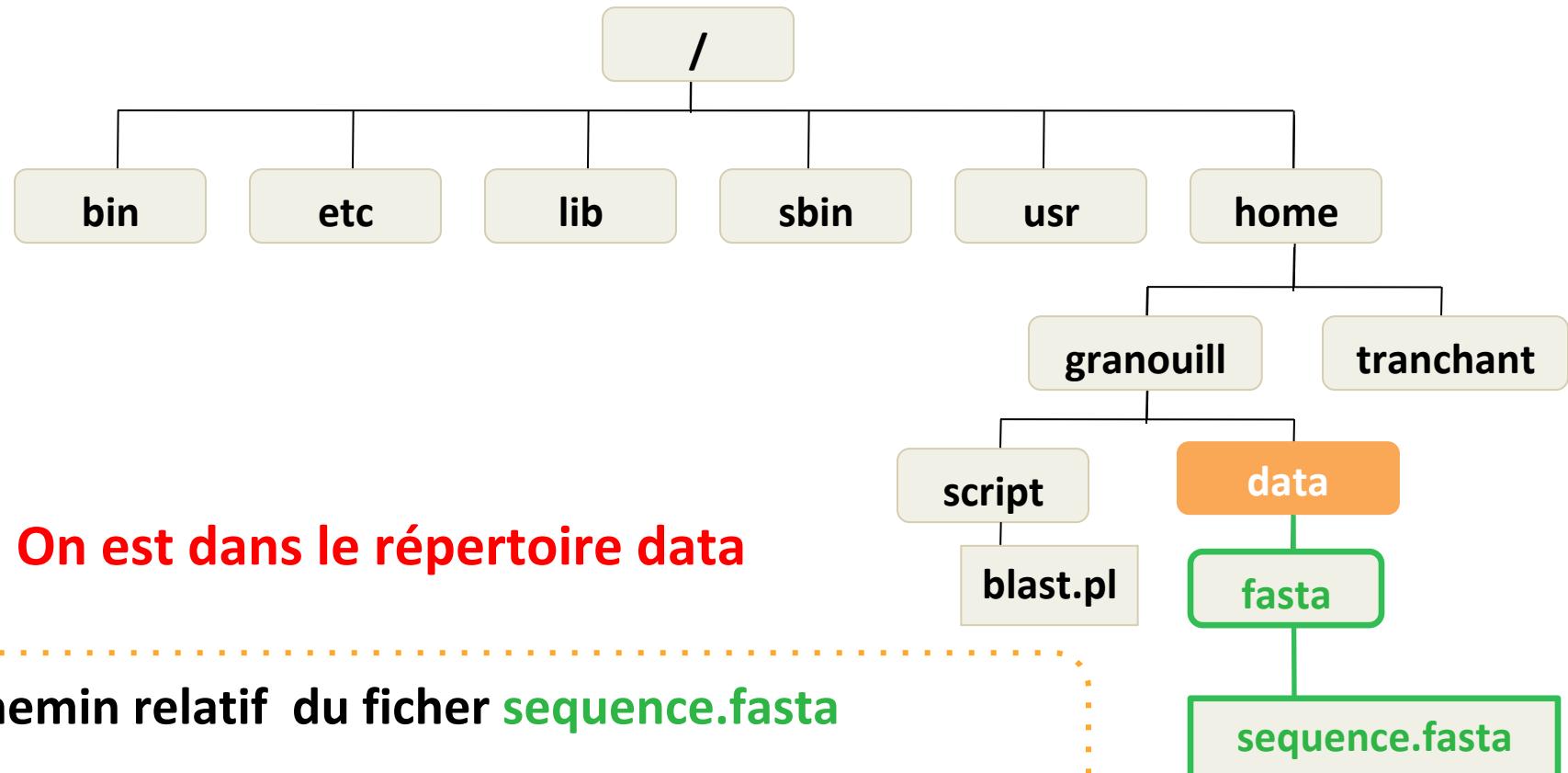
- Commence toujours par / (le répertoire racine)
- **Toujours correct peu importe où on l'on travaille**



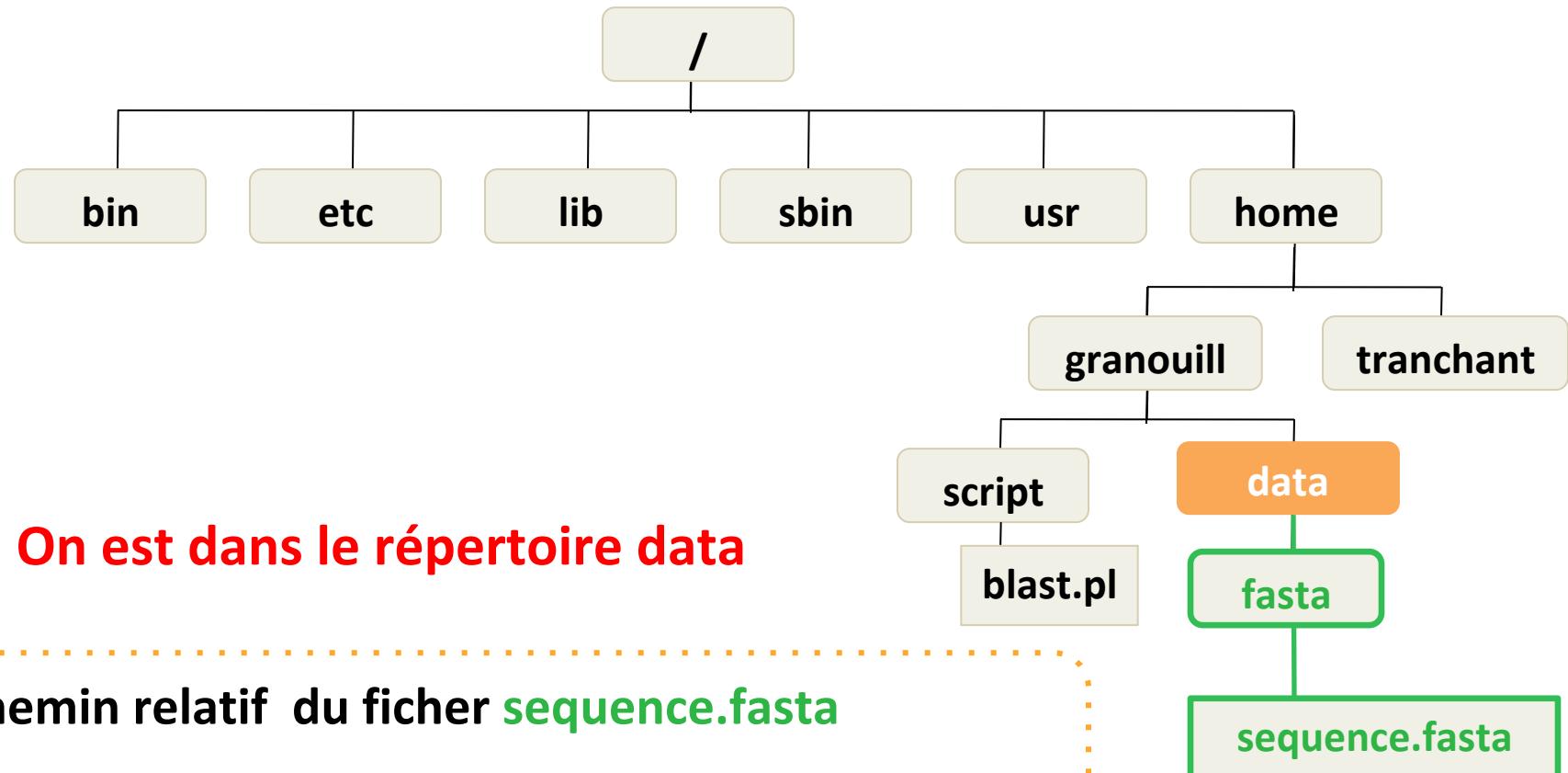
- Commence toujours par / (le répertoire racine)
- **Toujours correct peu importe où on l'on travaille**



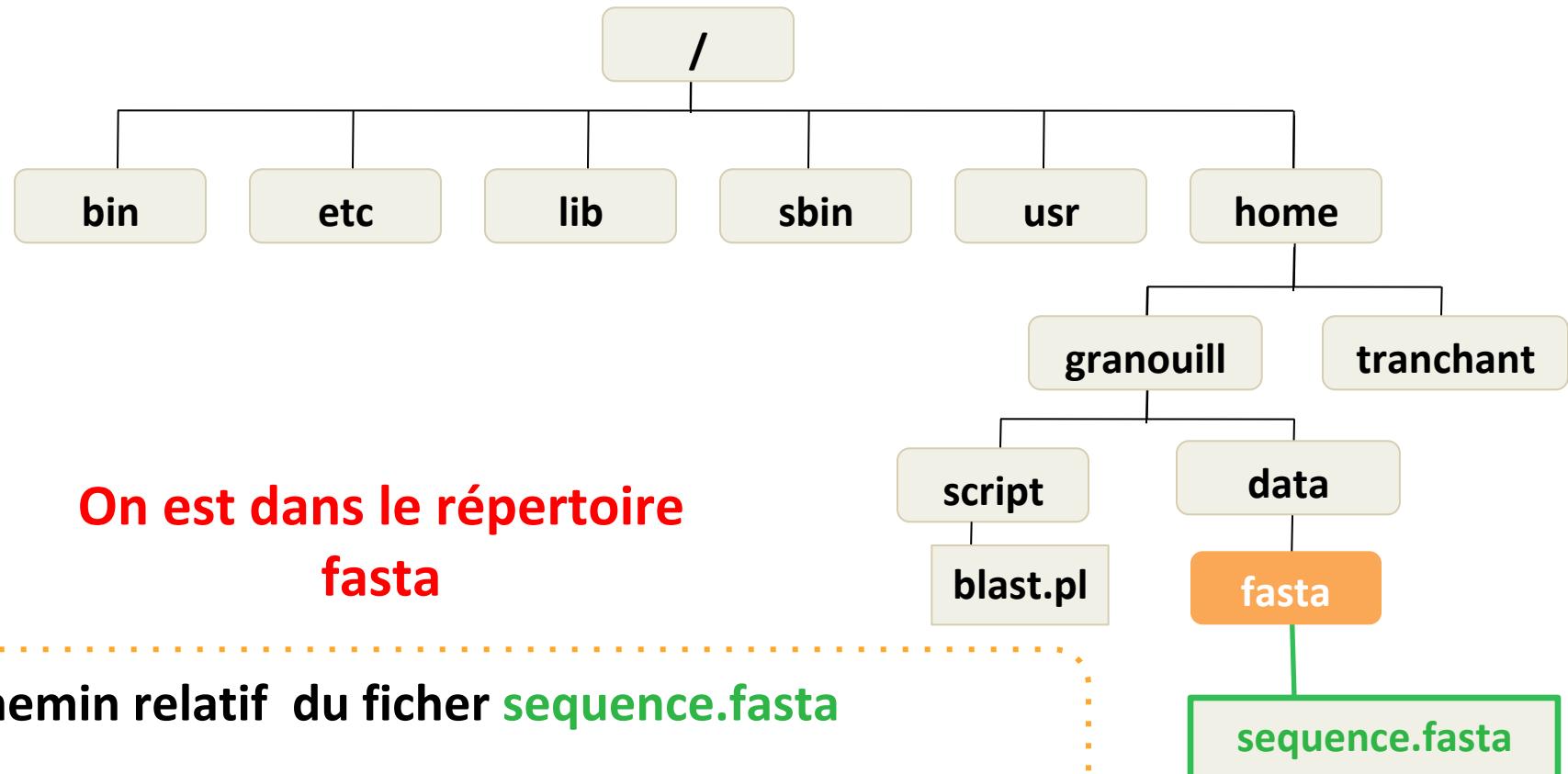
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



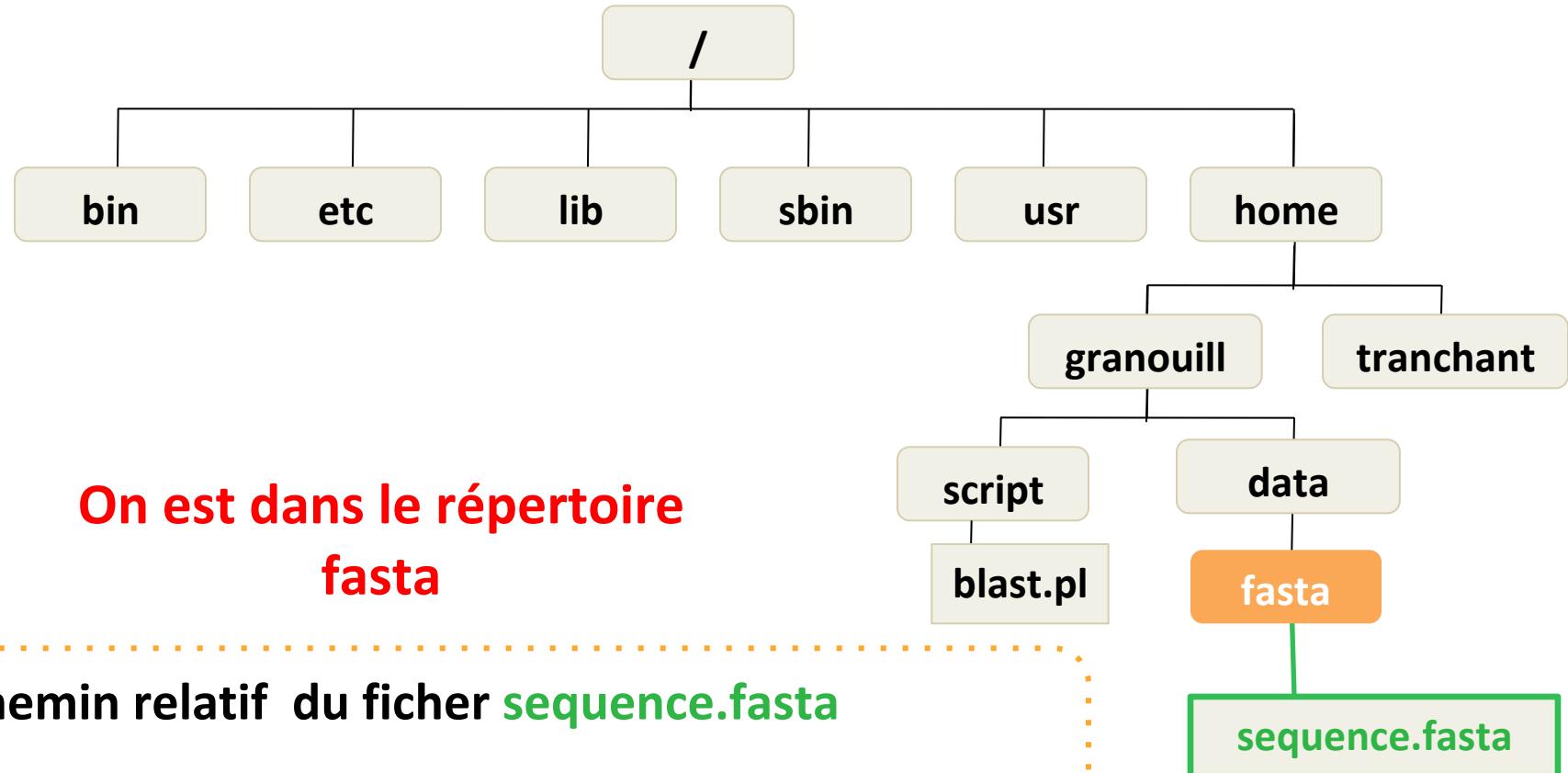
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



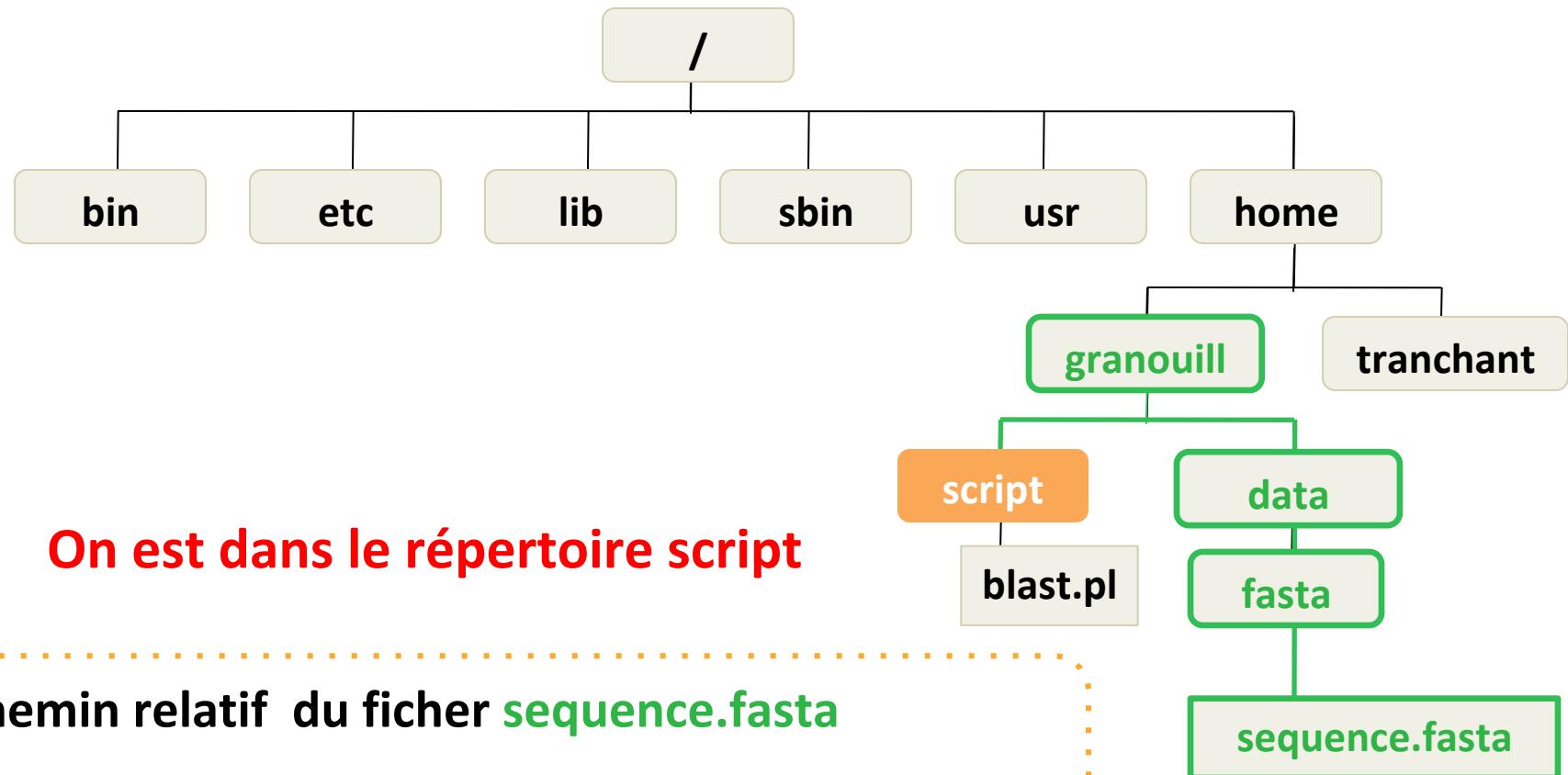
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



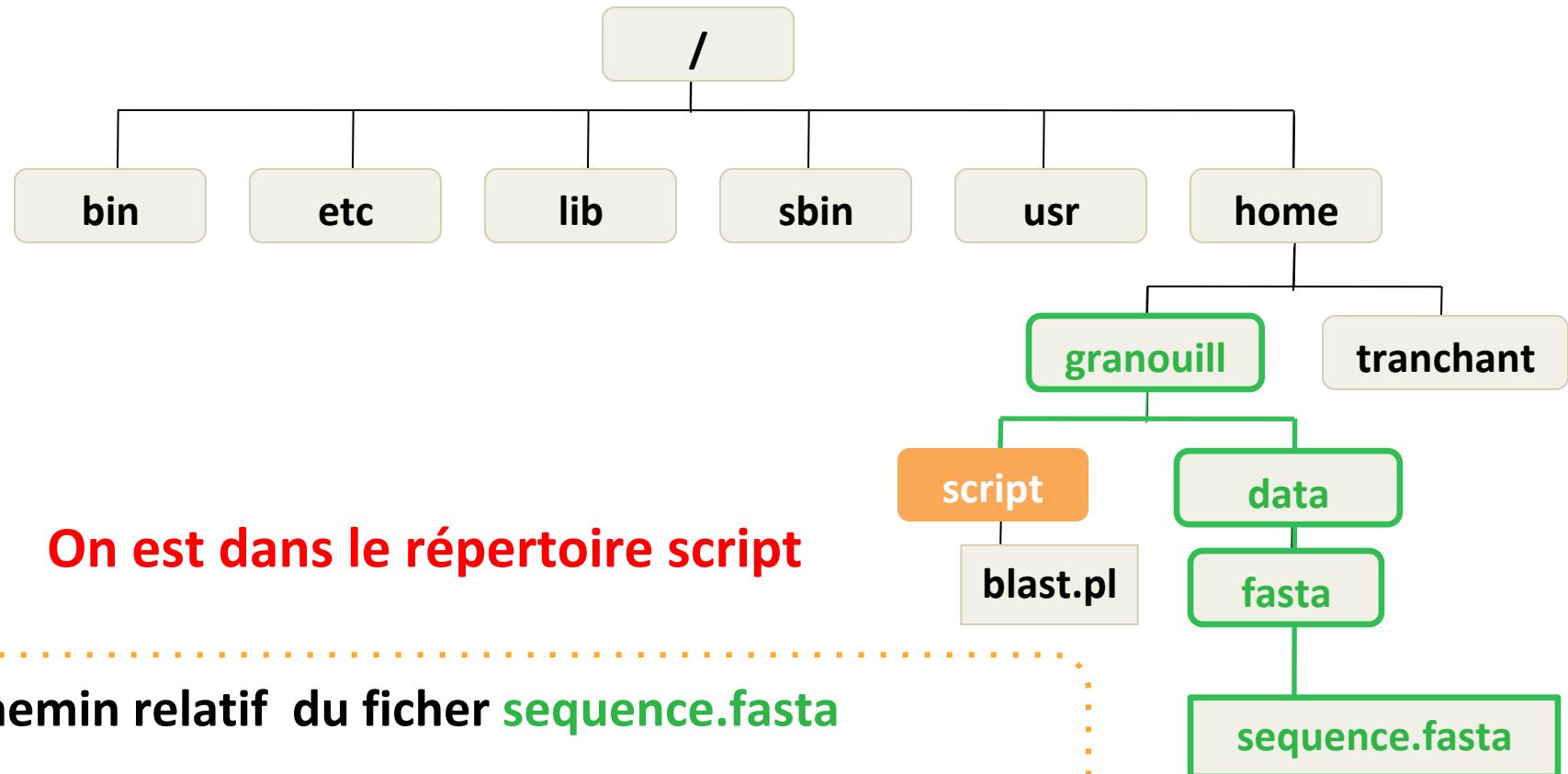
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- 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 /**





Practice

ls

4

Go to [Practice 4](#) on our github

Interagir avec les processus

<Ctrl> + C

Arrêter le processus en cours sous le terminal

<Ctrl> + Z

Tab completion

<Tab>

fichier/

Complète automatiquement le nom d'un répertoire qui est en cours de saisie (fichier unique)

<Tab><Tab>

Affiche la liste des différentes possibilités si le choix n'est pas unique

Interagir avec l'historique de commandes

Flèche bas/haut

- Afficher la commande précédente/suivante
- Presser plusieurs fois pour naviguer dans l'historique

<Ctrl> + R

- Afficher la dernière commande qui contient les caractères saisis.
- Presser les touches et commencer à taper la commande recherchée

- **Linux est sensible à la case**

Sequence.fasta **≠** SEQUENCE.fasta **≠** 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



Travailler avec les “wildcard”

Métacaractères : *, []

Qu'est ce un “wildcard” ?

C'est un caractère qui peut être utilisé comme substitut de plusieurs caractères ou classes de caractère dans une recherche

Peut être utilisé avec de nombreuses commandes linux

Comment utiliser les “wildcards” ?

- * N'importe quel caractère ou chaîne de caractère



KYVF-01.R1.fastq	KYVF-02.R1.fastq	KYVF.sam	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	KYVF.bam	ZO16.bam

Comment utiliser les “wildcards” ?

- * N'importe quel caractère ou chaîne de caractère



KYVF-01.R1.fastq	KYVF-02.R1.fastq	KYVF.sam	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	KYVF.bam	ZO16.bam

```
ls *fastq
```

- * N'importe quel caractère ou chaîne de caractère



KYVF-01.R1.fastq	KYVF-02.R1.fastq	KYVF.sam	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	KYVF.bam	ZO16.bam

```
ls *fastq
```

KYVF-01.R1.fastq	KYVF-02.R1.fastq	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	

Comment utiliser les “wildcards” ?

- * N'importe quel caractère ou chaîne de caractère



KYVF-01.R1.fastq	KYVF-02.R1.fastq	KYVF.sam	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	KYVF.bam	ZO16.bam

```
ls KYVF*fastq
```

- * N'importe quel caractère ou chaîne de caractère



KYVF-01.R1.fastq KYVF-02.R1.fastq KYVF.sam ZO16.fastq
KYVF-01.R2.fastq KYVF-02.R2.fastq KYVF.bam ZO16.bam

```
ls KYVF*fastq
```

```
KYVF-01.R1.fastq    KYVF-02.R1.fastq  
KYVF-01.R2.fastq    KYVF-02.R2.fastq
```

Comment utiliser les “wildcards” ?

- * N'importe quel caractère ou chaîne de caractère
- [] Tous les caractères entre crochets



KYVF-01.R1.fastq	KYVF-02.R1.fastq	KYVF.sam	ZO16.fastq
KYVF-01.R2.fastq	KYVF-02.R2.fastq	KYVF.bam	ZO16.bam

```
ls *.[sb]am
```

Comment utiliser les “wildcards” ?

- * N'importe quel caractère ou chaîne de caractère
- [] Tous les caractères entre crochets



KYVF-01.R1.fastq KYVF-02.R1.fastq KYVF.sam ZO16.fastq
KYVF-01.R2.fastq KYVF-02.R2.fastq KYVF.bam ZO16.bam

```
ls *.[sb]am
```

KYVF.sam ZO16.bam
KYVF.bam

```
= ls *.[!f]*
```



Practice

ls, *

5

Go to [Practice 5](#) on our github



Commandes relative à l'arborescence de fichiers

commande cd

cd

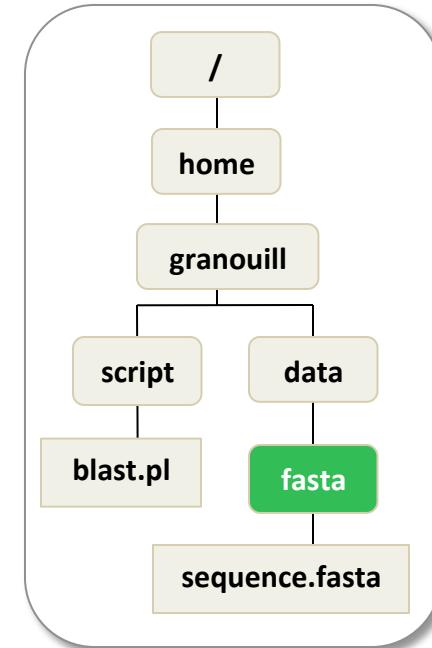
pour se déplacer dans l'arborescence

Change Directory

cd nom_repertoire(chemin absolu ou relatif)

`cd nom_repertoire(chemin absolu ou relatif)`

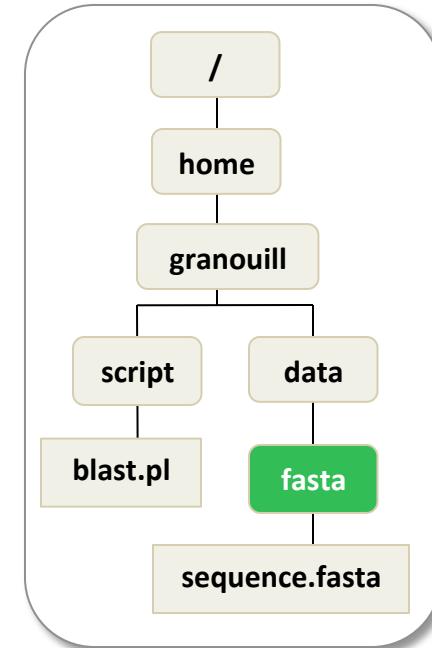
Chemin absolu :



`cd nom_repertoire(chemin absolu ou relatif)`

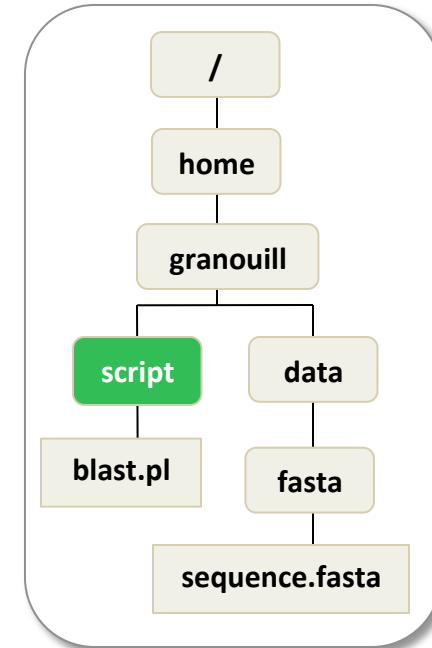
Chemin absolu :

`cd /home/granouill/data/fasta`



cd nom_repertoire(chemin absolu ou relatif)

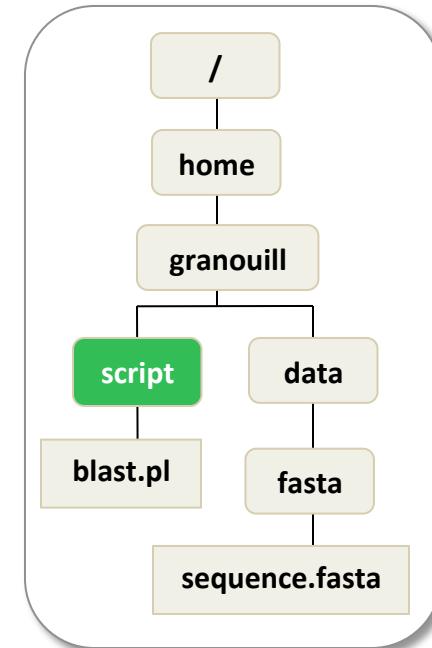
Chemin absolu :



cd nom_repertoire(chemin absolu ou relatif)

Chemin absolu :

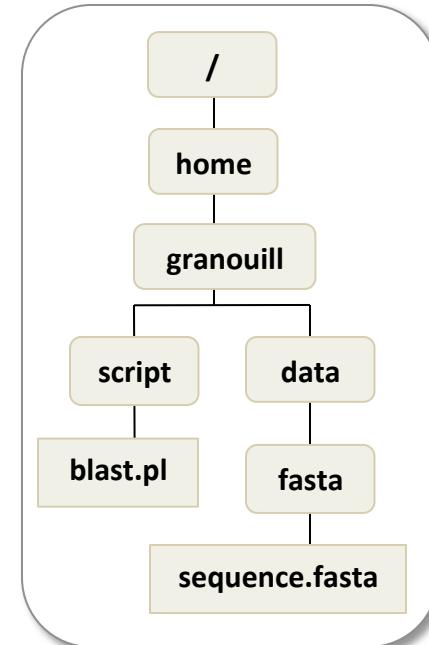
cd /home/granouill/script/



cd nom_repertoire(chemin absolu ou relatif)

Chemin relatif :

Commande	Se déplace dans	
cd directory_name	directory_name	Retour au répertoire de l'utilisateur courant home
cd	home directory	
cd ..	le répertoire parent	Remonte d'1 répertoire
cd ../../	?	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



Practice

pwd, cd, ls

6

Go to [Practice 6](#) on our github



Practice

mkdir, mv, ls, cp, rm

7

Go to [Practice 7](#) on our github



Des commandes pour éditer des fichiers et les manipuler

Commande cat

cat nom_fichier

Affiche le contenu d'un fichier sans pagination

```
MacBook-Pro-de-Christine:Data tranchan$ cat Data/Fasta/sequence.fasta
>Gxbjbsjxbjs
CCACCCCTCTTACAGTCTTACCAAATGTCCTTAAACTCCACCTAAAGTATCCAAAGA
CTCGAGAAATGCTGTGCCACAACCAGCTTGAGTCATCCATGACCGTTGATCTCCTT
GCCCCCAGAGTGGGCCTAGCACCATCTAGCTACTACTTGCCCTTCATAACCCATCATTGG
GATACCCTGAATACCTATCTTATAAGTTCCATATGGCTTATTTCTAAGTAAGAGATGC
ACTTAGTAAGTGCATGTCGTCTGACTGTTACTCTAATGTATGATATTTATATCCC
TATAATATAGTGTACTAATATATGTTGGTATTGTGTAGACTCCATTGTACCATGGTGT
GCTAATTAGAAATAACATGCCAGCTTGCTATTGTGGTTGCAAGTAAAGTAAAAAAA
MacBook-Pro-de-Christine:Data tranchan$
```

Commande less

less nom_fichier

Affiche le contenu d'un fichier

<space> pour défiler
q pour quitter
/ suivi par le mot

less Data/Fasta/EST-68566-Coffeacanephora.fasta

```
>gi|315911382|gb|GT649553.1|GT649553 CC00-XX-SH3-053-F02-EM.F Coffea canephora SH3 Coffea canephora cDNA clone CC00-X  
X-SH3-053-F02-EM, mRNA sequence
```

```
AATATTGTTGGTGTACCGTTCACCGCTAACATCTTCTGCACAAAGTAAGAGAAGGGTAACAAGTAAAAGGG  
AGCGAGACTATGGCTGCCACATCTGCTGCAGTACTAAATGGATTGGCTCTCCCTCTTGACTGGTGGAA  
GGAAAAGTCAGGCCCTGCTGGCTCACCAAGCTGCAGCTAGAGTCGGTGGTCTGCTGCTGCTCCAAA  
GAGATTAGTTGTGGTAGCTGCTCGTCCCTCCAAGAAGTCATGGATTCTGGTTAGAACTGGTCCAGGC  
TTCCTTGACCCCCGAGTATCTCGATGGCTCGTCCCTGGTACTACGGTTTGATCCACTGGTCTTGGCA  
AGGATCCAGCATTCTGAAATGGTATAAGAGAAGCTGAGCTATTGAGCTGGCAATGGCAATGACTGCAGT  
TGTTGGAATCTTGTGGCCAGGCATGGAGTGGTATCCCATGGTTCAAGCTGGCGCTGACCCCTGGTGCC  
ATTGCCATTCTCCTTGGTCTCGTCCCTCGTACTCAACTCCTGCTCA
```

```
>gi|315911381|gb|GT649552.1|GT649552 CC00-XX-SH3-077-D04-EM.F Coffea canephora SH3 Coffea canephora cDNA clone CC00-X  
X-SH3-077-D04-EM, mRNA sequence
```

```
ATTCATCGTATCTGCCCTCTTCATCTCCTCTTGTACTCAGCATCAGAGGCACCGCAGC  
TTTTAAAAGCTTATCACACACACACACATTCTTGACTCAGTCAGTGGCAACAATTACTGAGTACT  
ACTAGTTACTTATCGAACAAATCGATTAGCTTGCTAGGAGGTACCGTTCGGTTGGTTGACTTGTGTTG  
AGTATAAAATATTGATGGCTGGGATGCTGTGCGGAGTTAACATTGGAGGCGAGACTGAAACAGCTAAACCA  
GTCGAGGCTAATTCTCAGTCGGCTAGGGAGGAGGATGAAATTCACTCATTTGTAGCTTCGG
```

Data/Fasta/EST-68566-Coffeacanephora.fasta

Autres commandes utiles

head	Affiche les premières lignes d'un fichier (n=10, 10 lignes par défaut)	<i>head -n 20 script.pl</i>
tail	affiche les dernières lignes d'un fichier (n=10 par défaut)	<i>tail -n 5 script.pl</i>
wc	Compter le nombre de lignes, mots ou caractères d'un fichier	<i>wc script.pl</i> <i>wc -l script.pl</i>



Practice

ls

8

Go to [Practice 8](#) on our github

Commande grep

grep

pour rechercher un motif dans une ligne

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

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

grep*pour rechercher un motif dans une ligne**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



Practice

9

Go to [Practice 9](#) on our github

sort

Trier une ou plusieurs colonnes d'un fichier tabulé

sort -k2 fileName

sur la colonne 2 (tri alphanumérique)

sort -k2r fileName

sur la colonne 2, décroissant

sort -k2g -k1r fileName

sur la colonne 2 et 1

sort -t: -k3g fileName

sur la colonne 3 (nombre), le délimiteur de colonne est :

cut

Sélection de colonnes/champs d'un fichier

```
cut -d separateurColonne -f fieldNumber fileName
```

```
cut -d ":" -f1,5 /etc/passwd
```

Sélection des champs 1 et 5 dans un fichier tabulé
dont les colonnes sont séparées par un :



Practice

10

Go to [Practice 10](#) on our github



Les entrées / sorties

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

La sortie d'une commande peut être écrite
dans un fichier avec les caractères :

>

>>

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

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

Redirection	Action
Command > file	<ul style="list-style-type: none">si le fichier n'existe pas : il sera créési le fichier existe : efface le contenu
Command >> file	<ul style="list-style-type: none">si le fichier n'existe pas : il sera créési le fichier existe : écrit à la fin du fichier

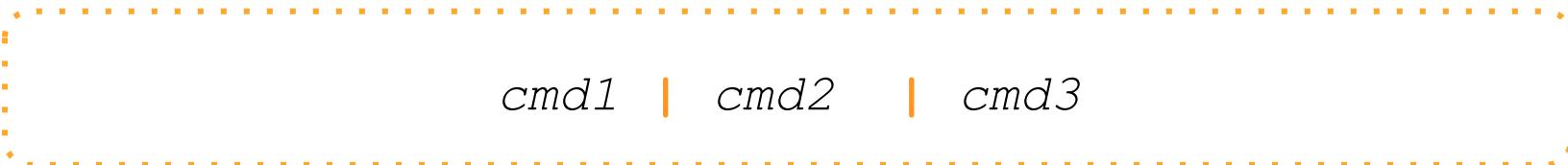


Practice

2

Go to [Practice 2](#) on our github

- La sortie d'une 1ère commande peut être envoyée comme input d'une 2ème commande
- Pour connecter/combiner 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

Connecter des commandes ensemble

```
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 -n 2
```



Practice

3

Go to [Practice 3](#) on our github

Formateurs itrop / South Green

- Christine Tranchant-Dubreuil
- Sébastien Ravel
- Alexis Dereeper
- Ndomassi Tando
- François Sabot
- Gautier Sarah
- Bruno Granouillac

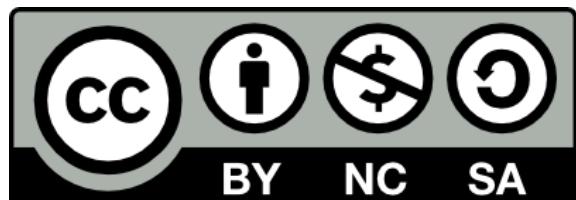




LOGO CERAAS



Merci !



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/>



Autres commandes utiles

Compresser des fichiers **tar,gzip**

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

Décompresser une archive **gunzip, tar**

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

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

```
zmore data.txt.gz
```

Rechercher une expression/motif dans une archive **zgrep**

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

Renommer des fichiers

rename

Exemple	Description
<code>rename 's/.txt/.fasta/' *</code>	rename the extension of all files
<code>rename 'y/a-z/A-Z/' *</code>	rename files in uppercase

Attributs des fichiers / permission

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

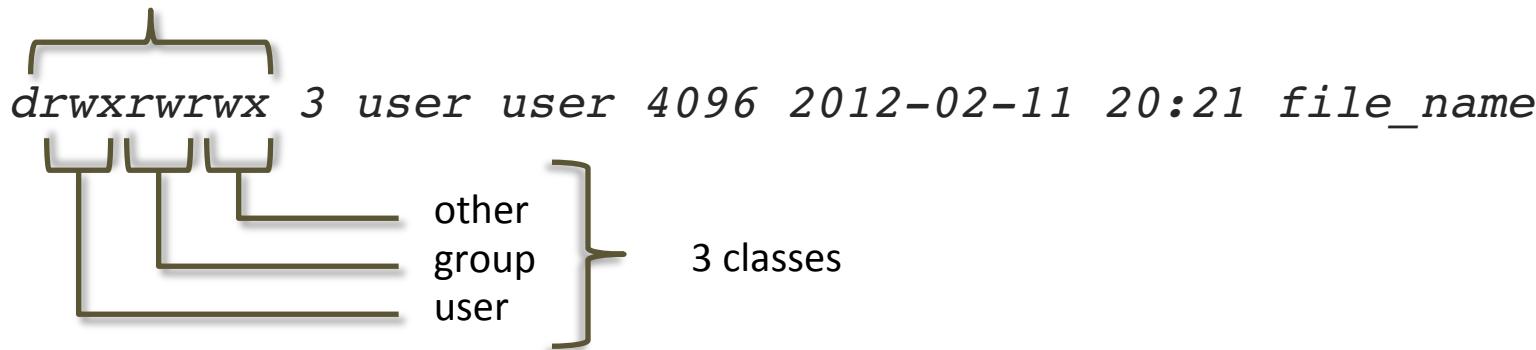
Type

-Interprétation/Légendes des permissions

- : fichier normal
- d** : répertoire
- l : lien symbolique

ls -l command

Permissions



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

Attributs des fichiers / permission

commande pour la gestion des permissions : chmod

chmod <perm> file_name

Chaque permission = 1 valeur

R	4
W	2
X	1
none	0

Exemple

chmod 740 script.sh

Owner=rwx

Group=r-- Other=---

chmod 755 script.sh

Owner=rwx Group=r-

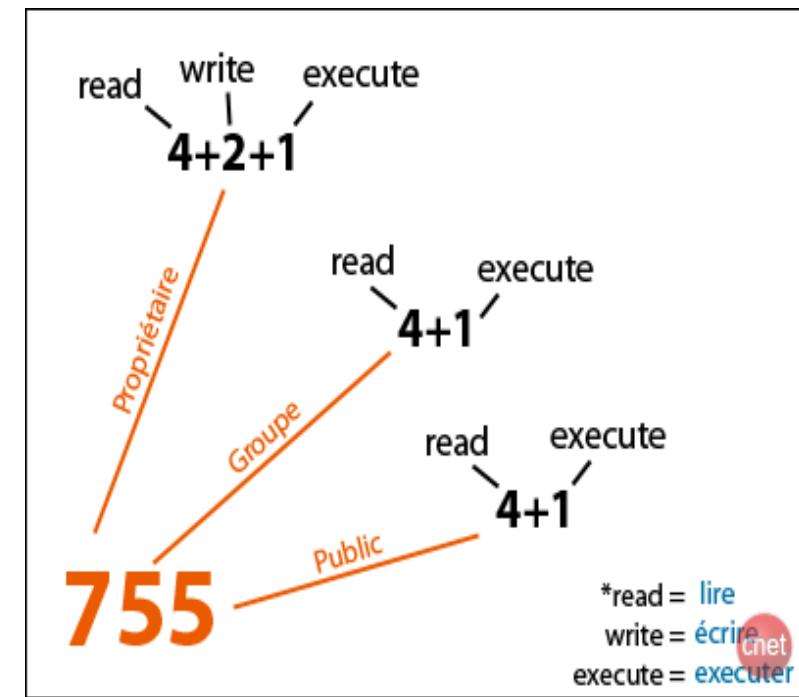
x Other=r-x

Visualiser et changer des droits

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





Performing basic text transformations on a file

Sed command

```
sed "s/Search/Replace/" file
```

Replace the **SEARCH** motif by the **REPLACE** one in **FILE**

Some examples

Example	Description
<code>sed "s/ATTAAT/NN/" file</code>	Replace the FIRST occurrence of <i>ATTAAT</i> by <i>NN</i> in FILE
<code>sed "s/ATTAAT/NN/g" file</code>	Replace ALL occurrences of <i>ATTAAT</i> by <i>NN</i> in FILE
<code>sed "s/attaat/NN/ig" file</code>	Replace ALL occurrences of <i>ATTAAT</i> by <i>NN</i> in FILE, case insensitive



Practice

6

Go to [Practice 6](#) on our github

Processing or analyzing text file

awk



Processing and Modifying a file

awk: manipulating a file line by line...

- AWK stand for "Aho, Weinberger, Kernighan", the 3 authors
- Programming language to easily manipulate tabular files (BLAST, SAM, VCF...)
- Motif search, operation, actions...

awk: manipulating a file line by line...

- For ***awk***, the input file is tabulate
- ***awk*** can perform computing on number and text chains
- ***awk*** can generate reporting following these manipulations

Processing and Modifying a file

Syntax : awk [-F] 'program' file

Option	Description
-F	To provide the type of field delimiter (tab in standard)

Processing and Modifying a file

Syntax : awk [-F] 'program' file

Option	Description
-F	To provide the type of field delimiter (tab in standard)

Predefined variables used by awk

Variable	Description
\$0	The whole line
NR	Number of the read line
NF	Umber of the fields in the line

Processing and Modifying a file

Helene	56	edu	hcyr@sun.com
jean	32	ri	jeanc@inexpress.net
julie	22	adm	juliem@sympatico.ca
michel	24	inf	michel@uqo.ca
richard	25	inf	r�aron@videotron.ca

File: contact.txt

Processing and Modifying a file

Helene	56	edu	hcyr@sun.com
jean	32	ri	jeanc@inexpress.net
julie	22	adm	juliem@sympatico.ca
michel	24	inf	michel@uqo.ca
richard	25	inf	rcaron@videotron.ca

File: contact.txt

```
awk '{print $0}' contact.txt
```

```
Helene 56 edu hcyr@sun.com
jean 32 ri jeanc@inexpress.net
julie 22 adm juliem@sympatico.ca
michel 24 inf michel@uqo.ca
richard 25 inf rcaron@videotron.ca
```

Print every line

Processing and Modifying a file

Helene	56	edu	hcyr@sun.com
jean	32	ri	jeanc@inexpress.net
julie	22	adm	juliem@sympatico.ca
michel	24	inf	michel@uqo.ca
richard	25	inf	r��aron@videotron.ca

File: contact.txt

```
$awk '{print NR,$1,$2}' contact.txt
```

```
1 Helene 56
2 jean 32
3 julie 22
4 michel 24
5 richard 25
```

Print the **line number**, then the **FIRST** and **SECOND** fields

Processing and Modifying a file

Helene	56	edu	hcyr@sun.com
jean	32	ri	jeanc@inexpress.net
julie	22	adm	juliem@sympatico.ca
michel	24	inf	michel@uqo.ca
richard	25	inf	r��aron@videotron.ca

```
$awk '{print $1,$2};  
END { print NR « lines read »; }' contact.txt
```

```
Helene 56  
Jean 32  
Julie 22  
Michel 24  
Richard 25  
5 lines read
```

Perform the previous command then print the number of lines read

Processing and Modifying a file

Helene	56	edu	hcyr@sun.com
jean	32	ri	jeanc@inexpress.net
julie	22	adm	juliem@sympatico.ca
michel	24	inf	michel@uqo.ca
richard	25	inf	r��aron@videotron.ca

```
$awk ' {print $1,$3; sum+=$2}
END { print « Age sum : », sum; }' contact.txt
```

```
Helene edu
jean ri
julie adm
michel inf
richard inf
Age sum : 159
```

print for each line the 1st and third fields, then sum the 2nd and output the sum at the end

Processing and Modifying a file

```
awk '$2>24 && $2<50{ print $1, " :", $2; }'  
contact.txt
```

Helene : 56
jean : 32
richard : 25

Print only if higher than 20 AND lower than
50



Processing and Modifying a file

```
awk '$3 == "inf" {print $0}' contact.txt
```

```
michel 24 inf michel@uqo.ca
richard 25 inf rcaron@videotron.ca
```



Practice

7

Go to [Practice 7](#) on our github