

# Guide de survie sur Linux

La ligne de commande est mon amie

Formation du collectif de baillarguet: c2b

# Programme

- Historique
- Concepts
- Manipulation de fichiers (TP1)
- Execution de programme / écriture d'un scripts (TP2 - Faire un blast)
- Faire un mapping ?

# Historique

Années 60: Création d'UNIX et du langage C dans les laboratoires Bell (aujourd'hui AT&T)



Ken Thompson & Dennis Ritchie

# Historique

Le materiel informatique dans les années 70



# Historique

Le materiel informatique dans les années 80: Le terminal !



# Historique

Arrivée des ordinateurs personnels (90)



# Historique

Arrivée des ordinateurs personnels (90)

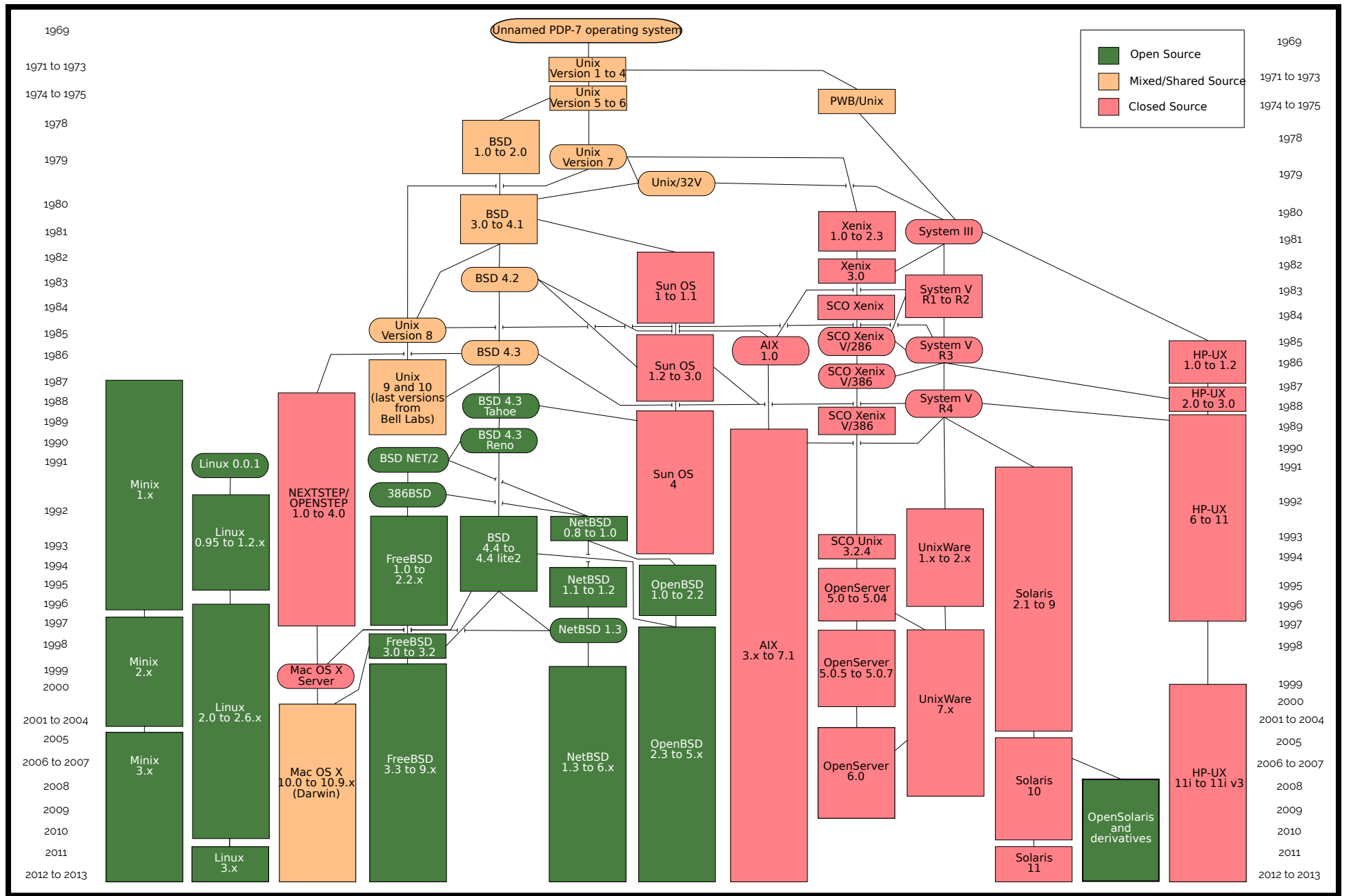


Pendant ce temps ...

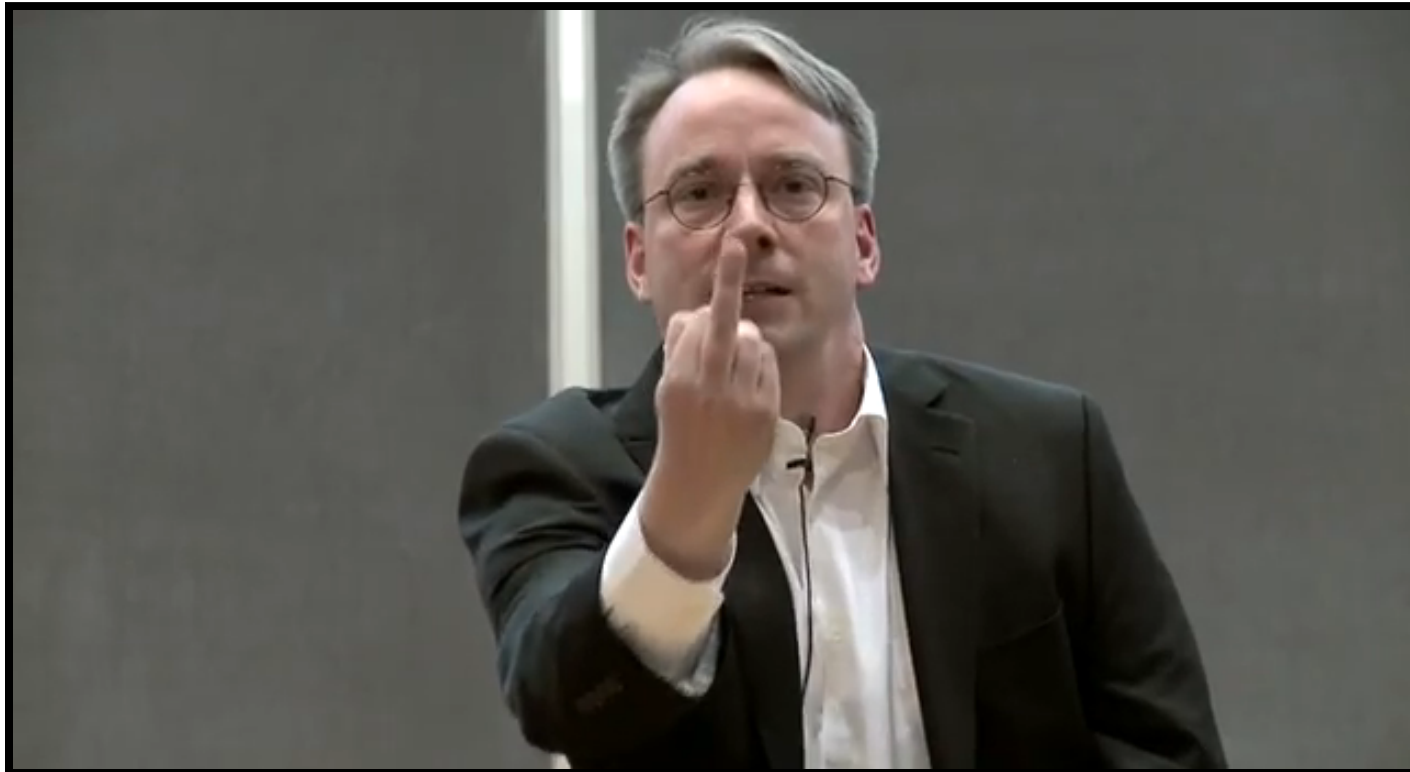


*Brent Romano*

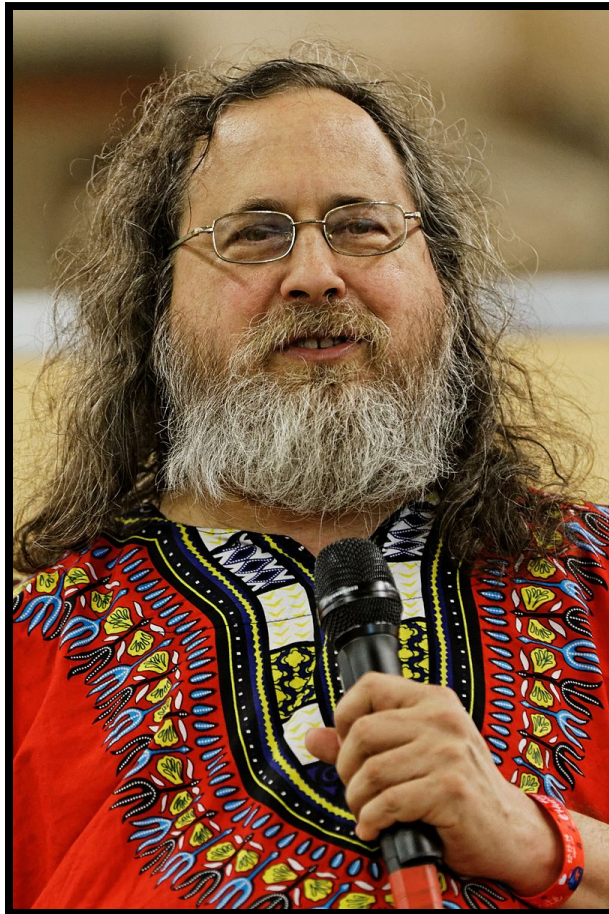




## Création de Linux (Linus Torvald, 1991, Helsinki)



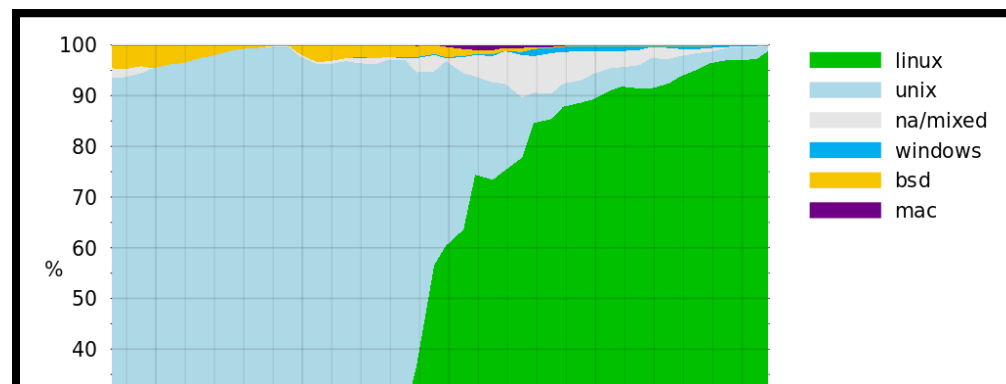
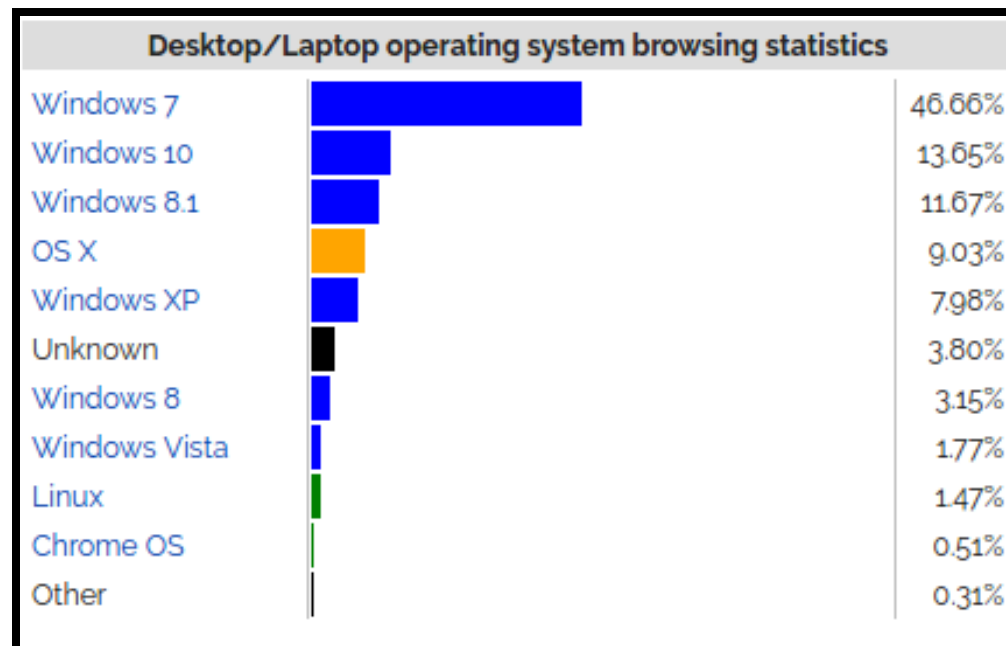
## Création de GNU (Richard Stallman, 1983, MIT)

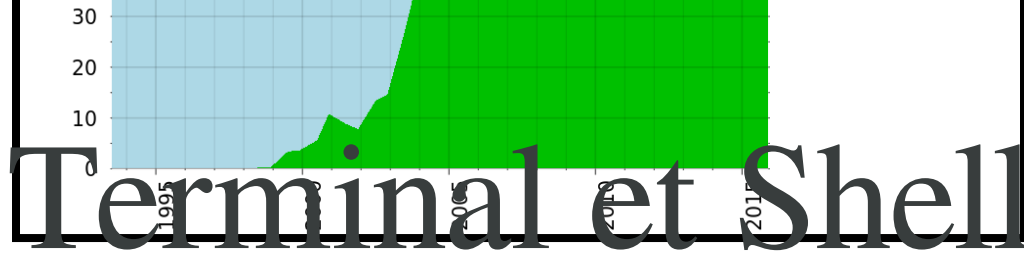


*Free software foundation*, patrimoine mondial de l'humanité  
(Unesco, 2004)

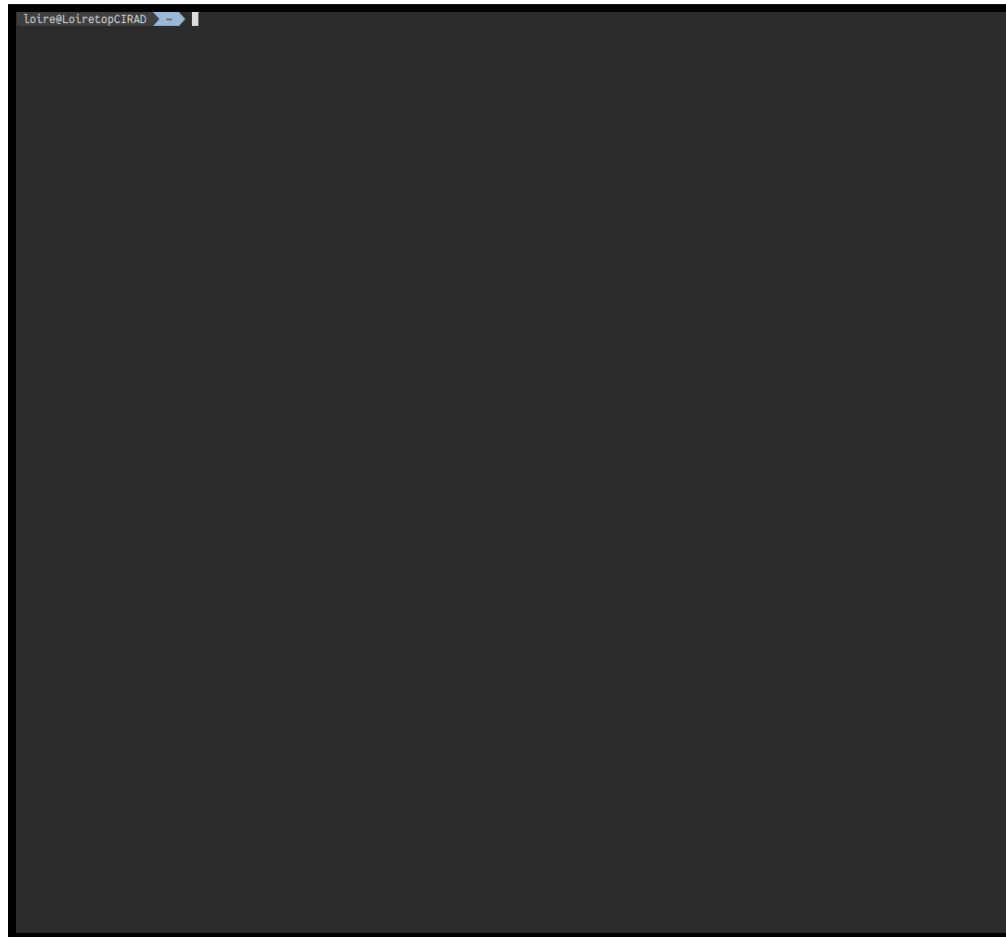
## Pourquoi GNU/Linux ?

- Open source
- Communautaire
  - Stable (vous choisissez ou pas de faire des mises à jour)
  - Multi-tâches et multi-utilisateurs
  - Correction des bugs (pas besoin que MS ou Apple s'intéresse à votre problème)
  - Dépôt de codes (liste de programmes faciles à installer et sans virus)
  - Rapide car vous choisissez exactement ce que vous installez sur la machine
  - Les bioinformaticiens (programmeurs) n'ont pas le temps de développer des interfaces graphiques





Votre mode d'interaction avec le materiel

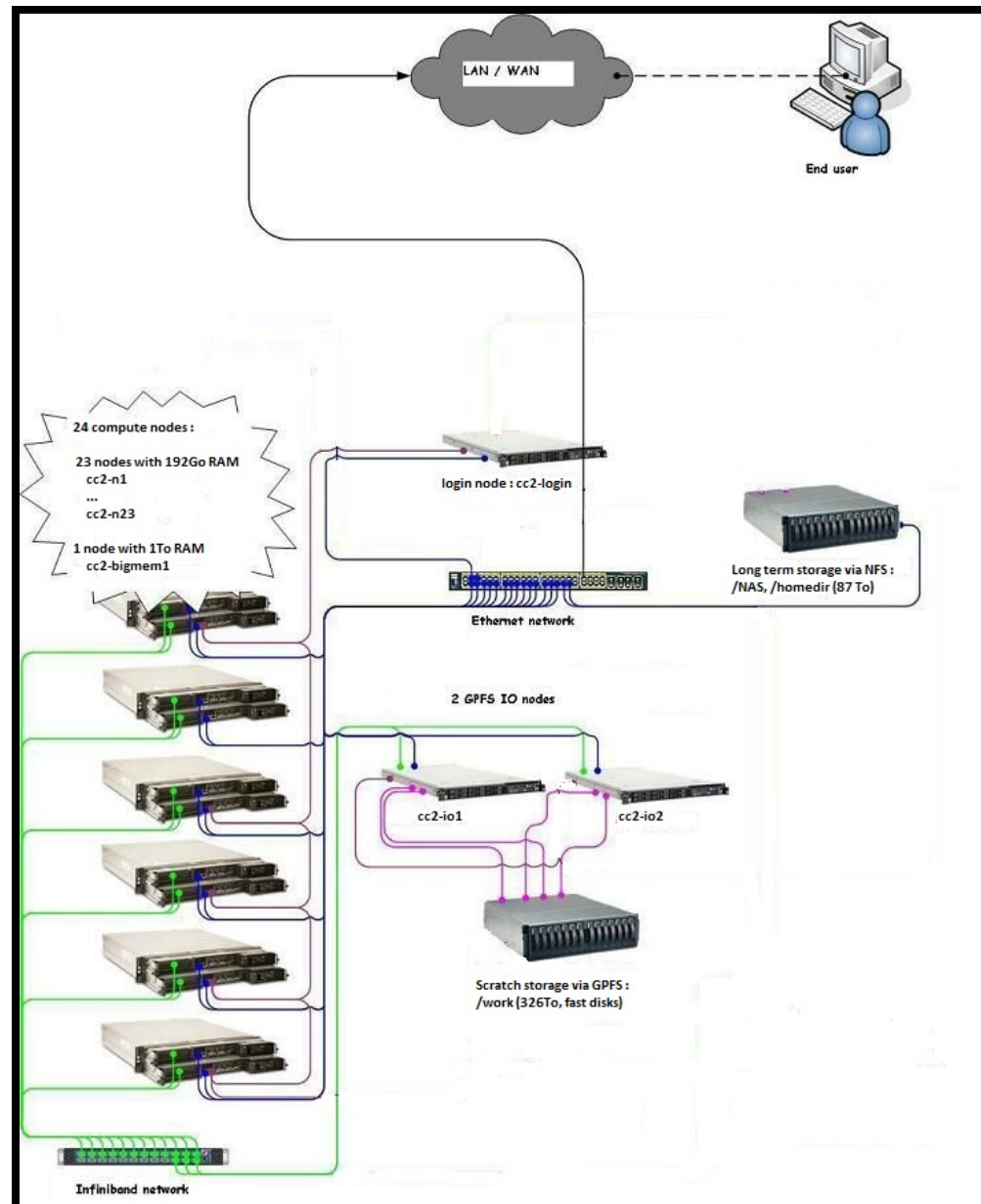


# Shell

Ensemble de commande qui permettent d'interagir depuis le terminal avec le noyau linux

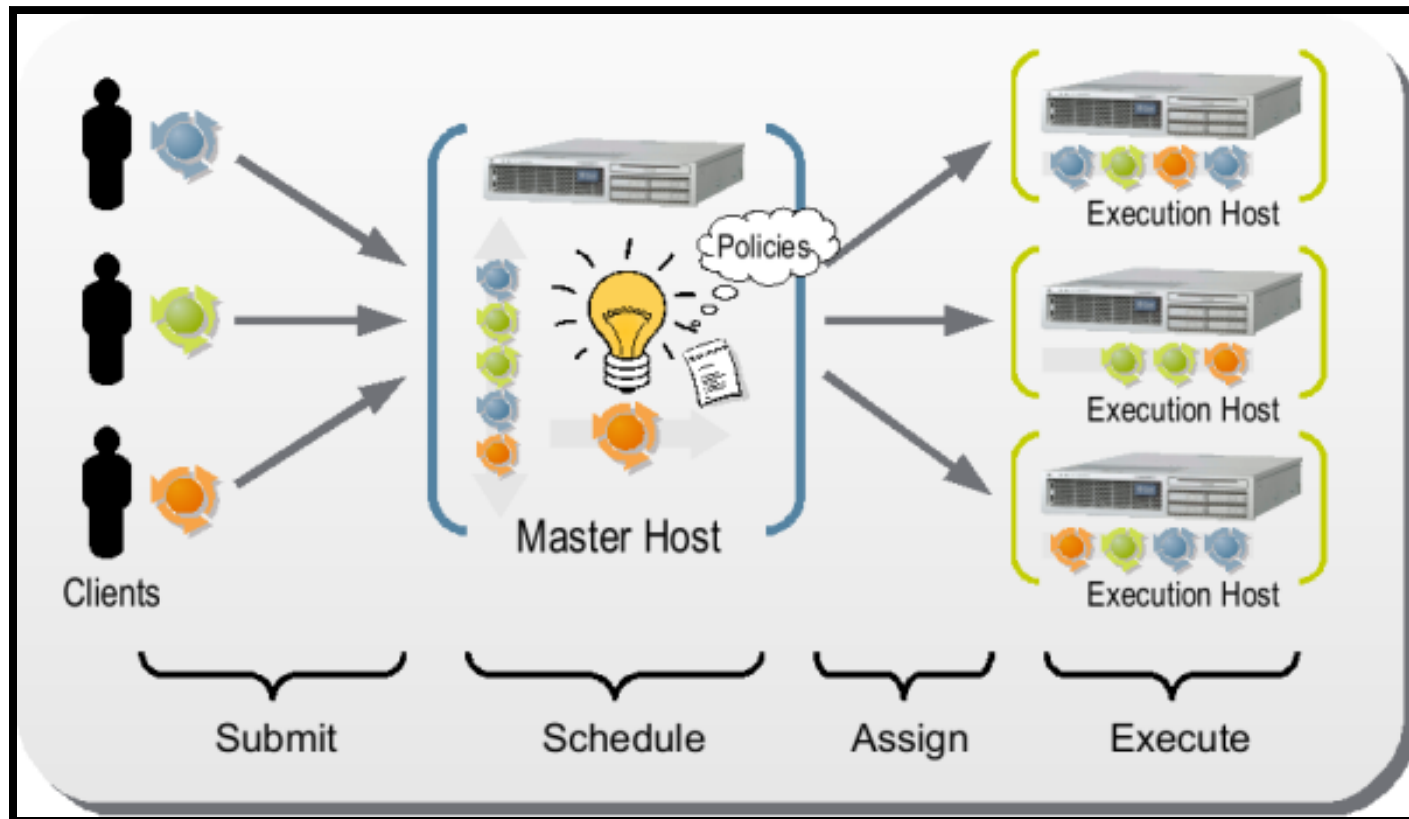


# Le cluster du CIRAD





# SGE: Sun Grid Engine



Pour se connecter au cluster de l'université de la guadeloupe, trois actions:

- `ssh "login"@sasc3i.univ-ag.fr`
- `ssh "login"@wahoo`
- `qrsh -q normal.q`

# Fichiers

## TOUT EST FICHER

- Fichier texte == "human readable" (ASCII)
  - une page html
  - un fichier fasta
  - un script R

cat nomdefichier
- Fichier binaire == "human unreadable"
  - Fichier .docx
  - Programme Word.exe
  - Les commande unix (cd, ls, mkdir etc ...)
  - Un fichier compressé (zip)

# Fichiers

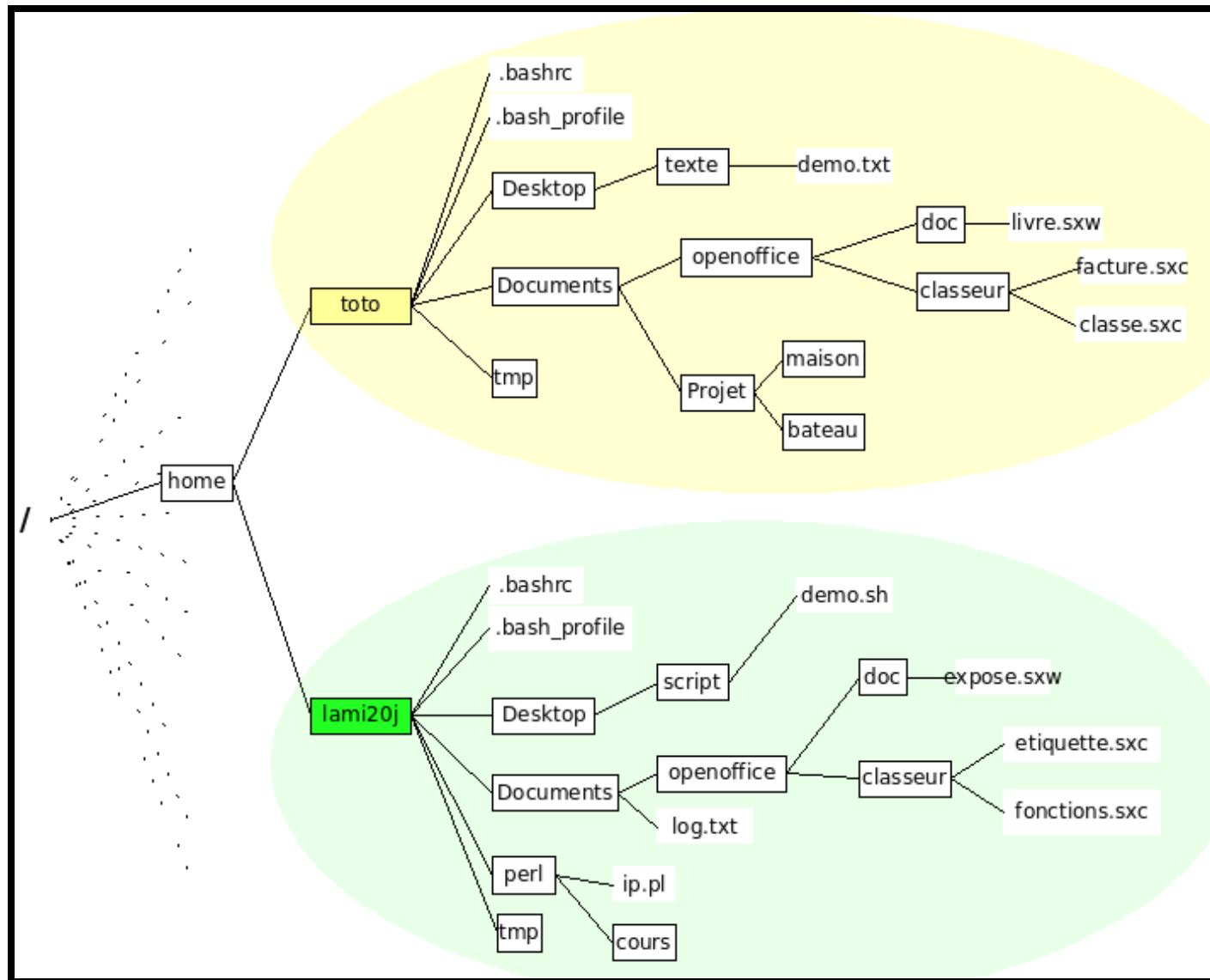
## REMARQUES

- Un fichier a des droits
  - r : "readable"
  - w : "writable"
  - x : "eXecutable"
- un fichier est stocké sur un support (disque dur)
  - Il a une adresse (inode)
  - On peut creer des liens (raccourci) vers cette adresse
  - Il est positionné dans un repertoire (d: directory), au sein d'une arborescence

# Arborescence

- Repertoire (dossier) == Contient des fichiers
- Arborescence de dossiers == hierarchie des repertoires

# Arborescence



# Les chemins absolus et relatifs

Spécifier où se trouve un fichier

- depuis l'endroit où vous êtes == relatif
- depuis un point fixe (la racine) == absolu

# Les chemins absolus et relatifs

Exemple:

- absolu: `/home/cirad/eloire/share/fichier.txt`
- relatif: `share/fichiers.txt`

Remarques

- pour savoir où vous êtes placé dans l'arborescence, tapez `pwd` (Print Working Directory)
- Pour vous déplacer dans le répertoire, utilisez `cd` (Change directory)
- Pour lister les fichiers contenus dans un répertoire, utilisez `ls` (LiSt)



# Les chemins absolus et relatifs

## Remarques

- pour savoir où vous êtes placé dans l'arborescence, taper pwd (Print Working Directory)
- Pour vous déplacer dans les répertoire, utiliser cd (Change directory)

## Raccourci

- .. Le répertoire parent (au dessus dans l'arborescence)
- / La racine (lieu qui contient TOUTE l'arborescence)
- ~ votre répertoire utilisateur (home, où tout les fichiers vous appartiennent)

# Les chemins absolus et relatifs

## Remarques

- pour savoir où vous êtes placé dans l'arborescence, taper `pwd` (Print Working Directory)
- Pour vous déplacer dans le répertoire, utiliser `cd` (Change Directory)

## Exemples

- `cd /home/eloire/share` : Aller dans le répertoire `share` sur mon compte
- `cd ..` : Remonter d'un niveau dans l'arborescence
- `cd TP1` : Aller dans le répertoire `TP1` (doit exister dans votre répertoire courant !)
- `cd ~` : Aller dans votre "home"

- `cd ~` : Allez dans votre home

# Structure d'une commande bash

$$F(x,y, ...) = z$$

- F est la commande
- z, le résultat de la commande, est appelé "sortie standard"
- x, y et autre sont des arguments (optionnels)
- s'ajoute des "flag" == options qui modifie le comportement de la commande
- s'ajoute une sortie d'erreur, utile quand quelque chose ne se passe pas bien (souvent)

syntaxe: `F -options x y > z`

# Exemple de commande

- `cd`
- `ls`
- `pwd`
- `mkdir TP1` : Créer le repertoire TP1 dans votre repertoire courant
- `rm fichier.txt` : Supprime le fichier fichier.txt
- `ln -s fichier.txt share/fichier.txt` : creer un raccourci vers le fichier dans le repertoire share
- `mv fichier.txt share/` : Deplace le fichier dans le repertoire share
- `nedit fichier.txt` : Edition de fichier.txt
- `ls > listefichier.txt` : copie la liste des fichiers dans un fichier

# Exemple de commande GNU avancée

- `wc fichier.txt` : Compte le nombre de caractères, de ligne
- `wc -l fichier.txt` : Compte le nombre de ligne
- `grep toto fichier.txt` : Affiche toutes les lignes du fichiers qui contiennent le mot "toto"
- `grep -c toto fichier.txt` : compte toutes les lignes du fichiers qui contiennent le mot "toto"
- `awk '{print $2}' fichier.csv` : Affiche la deuxième colonne de fichier.csv
- `du -h fichier.txt` : Affiche la taille (en Mo) d'un fichier
- `edit fichier.txt` : Edition de fichier.txt
- `ls > listefichier.txt` : copie la liste des fichiers dans un fichier

# Redirection / pipe

Une commande peut afficher du text à l'écran (appelé sortie standard et erreur standard).

Pour stocker la sortie standard dans un fichier, on utilise le symbole de redirection ">"

Pour stocker l'erreur standard dans un fichier, on utilise le symbole de redirection "2>"

Pour rediriger la sortie standard d'une commande vers une autre commande, on utilise le symbole pipe "|"

exemple

# Tips!

- Appuyer sur TAB pour autocompléter les noms de repertoire et de fichiers
- Utiliser le symbole "\*" (wildcard) pour signifier n'importe quelle chaine de caractères
- La touche ↑ rappelle les commandes précédentes
- la commande history liste toutes les commandes que vous avez exécutées
- Ctrl + C pour interrompre une tâche  
exemple

# TP1: Demonstration de manipulation de fichiers via des commandes unix

But:

- se deplacer dans l'arborescence
- creer un repertoire
- copier un fichier
- lancer des commandes
- chaîner des commandes



# TP2: Faire un blast

But:

- éditer un fichier texte contenant des instructions
- soumettre la tâche au cluster

# TP3: Faire un mapping