

Septembre 2023

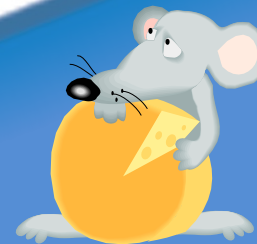
<https://perso.isima.fr/loic>



SB

Loïc YON

Patrice LAURENÇOT



**ISIMA** 

# Semaines bloquées

- 2 semaines // piscine
- Système d'exploitation UNIX / Linux
- Langage C
- Bases fondamentales pour les trois années
  - Langage C++, C#, JAVA, Swift, Kotlin
  - Cours de système
  - Utilisation des machines
- Evaluation le dernier jour

Septembre 2023

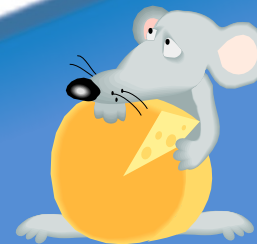
<https://perso.isima.fr/loic>



Unix

Loïc YON

Patrice LAURENÇOT



**ISIMA** 

# Plan



- Ordinateur ?
- Système d'exploitation ?
- Fichiers / Chemins / répertoires
  
- Droits / Liens / Redirections / Commande find
  
- Processus / Pipes
  
- Réseau / Web

# Cours 1

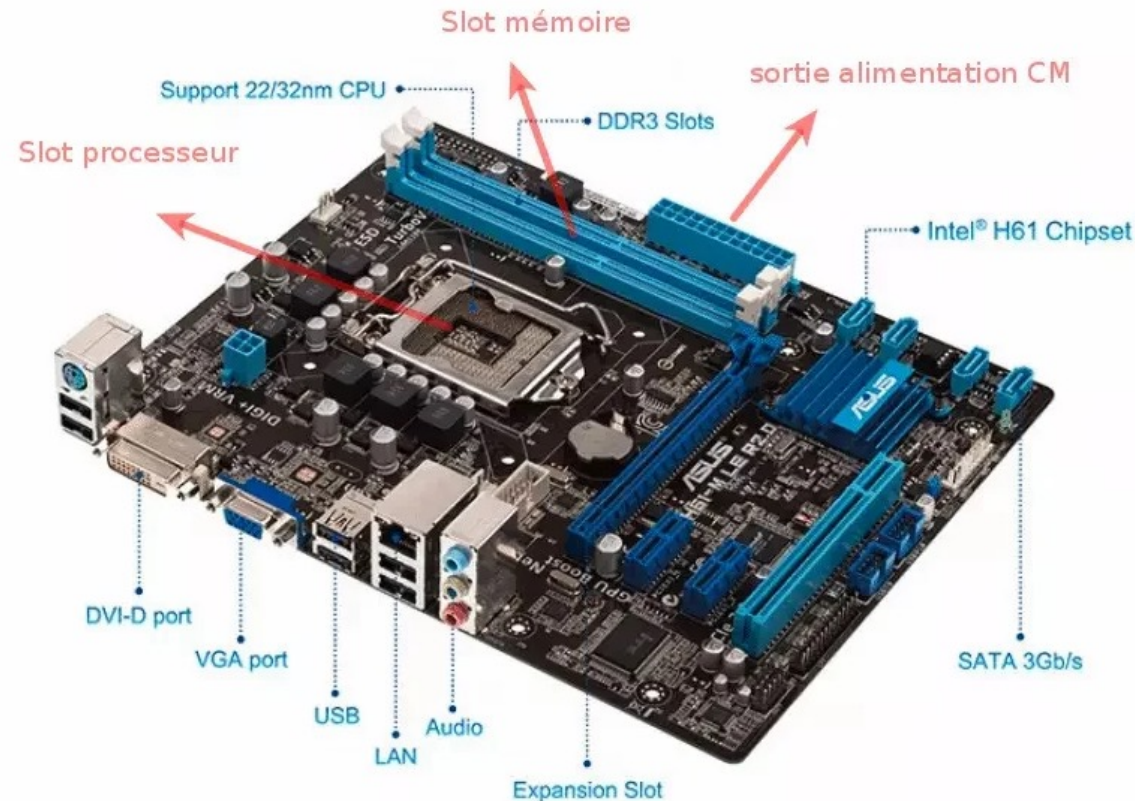
Ordinateur ?

Fichiers

Chemins / Répertoire

# Ordinateur ?

- Composant principal : Carte mère



- Assure la transmission des informations entre le processeur, chipset, mémoire, carte d'extensions...
- Différents formats ATX, mini-ITX, flex-ATX,...

# Carte mère ?

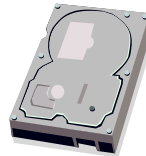
- Processeur



- Mémoire vive / RAM



- Mémoires auxiliaires



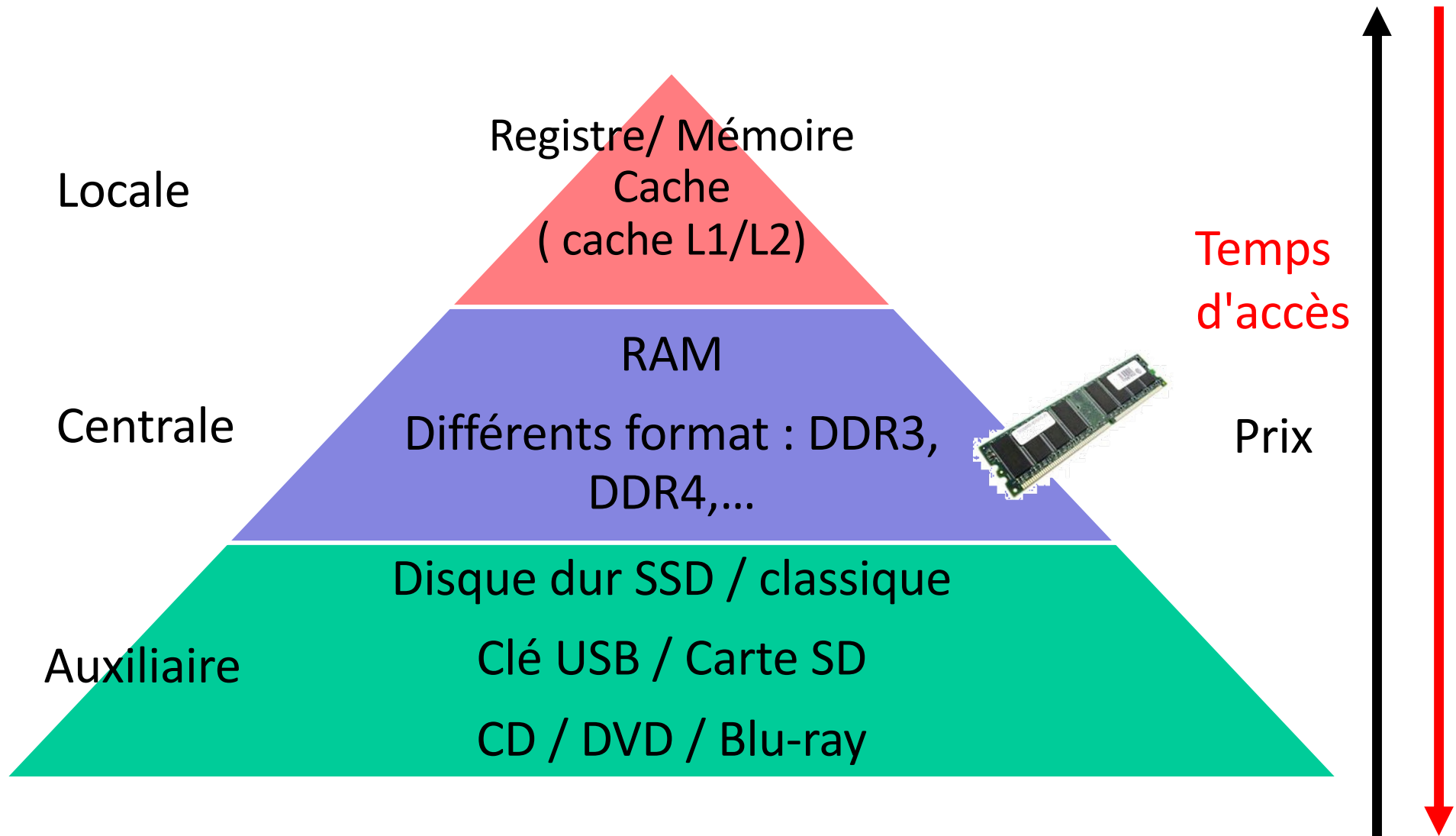
- Carte graphique (GPU)



- Périphériques - Entrées / Sorties



# Mémoire

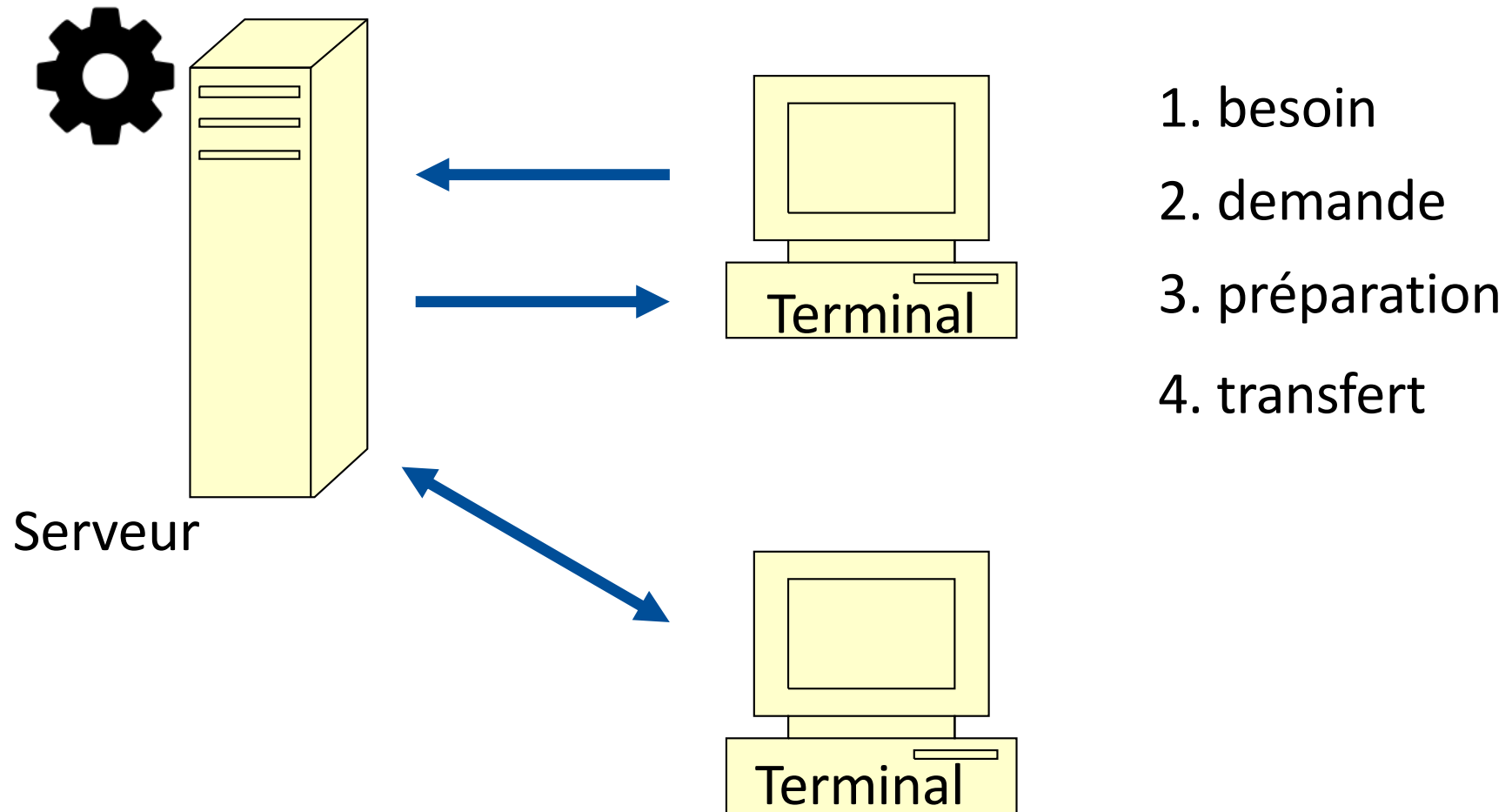


Une mémoire particulière : ROM ou EPROM  
-> contient **le BIOS ou l'UEFI**



# Relation client/serveur

## Relation client/serveur



Protocole de communication (http, smtp, rtp, ...)

# Systeme d'exploitation (SE) *Operating System (OS)*



- Interface utilisateur / machine

- Texte / Graphique

- Gestion des ...

- Ressources (fichiers, mémoires, périphériques)
- Données / programmes des utilisateurs
- Conflits entre les utilisateurs
  - Architecture multiutilisateur



- Plusieurs couches

- Couche basse : gestion matériel, périphérique (dépendant du matériel)
- Couche haute : fonction plus évoluée (gestion de fichiers, ...), indépendante du matériel

# Systeme d'exploitation (SE) *Operating System (OS)*








- Un OS est composé :
  - d'un noyau (gestion des processus) -> cœur du système
  - Composants logiciels assurant les fonctions de l'OS
    - Gestion de fichiers
    - Gestion de la mémoire
    - **Langage de commandes** ( shell / DOS, powershell),...
      - Interpréteur de commandes
    - ...




- Différents OS existent :
  - Windows, Linux, UNIX, Mac OS, AS400, ...

# ISIMA


- Terminaux X
- Serveurs
  - nightmare 
  - ada 
  - TSE1/ TSE2 
  - Guacamole (via navigateur)
- PCs pour TP spécifiques avec SE virtuels (  ,  )
- Machines virtuelles
- Mac



# Et pour vous ?

- Distribution LINUX d'usage libre 
- Windows avec WSL2
- Guacamole
  - Machines virtuelles personnelles
  - Serveurs pédagogiques
- Libre Office



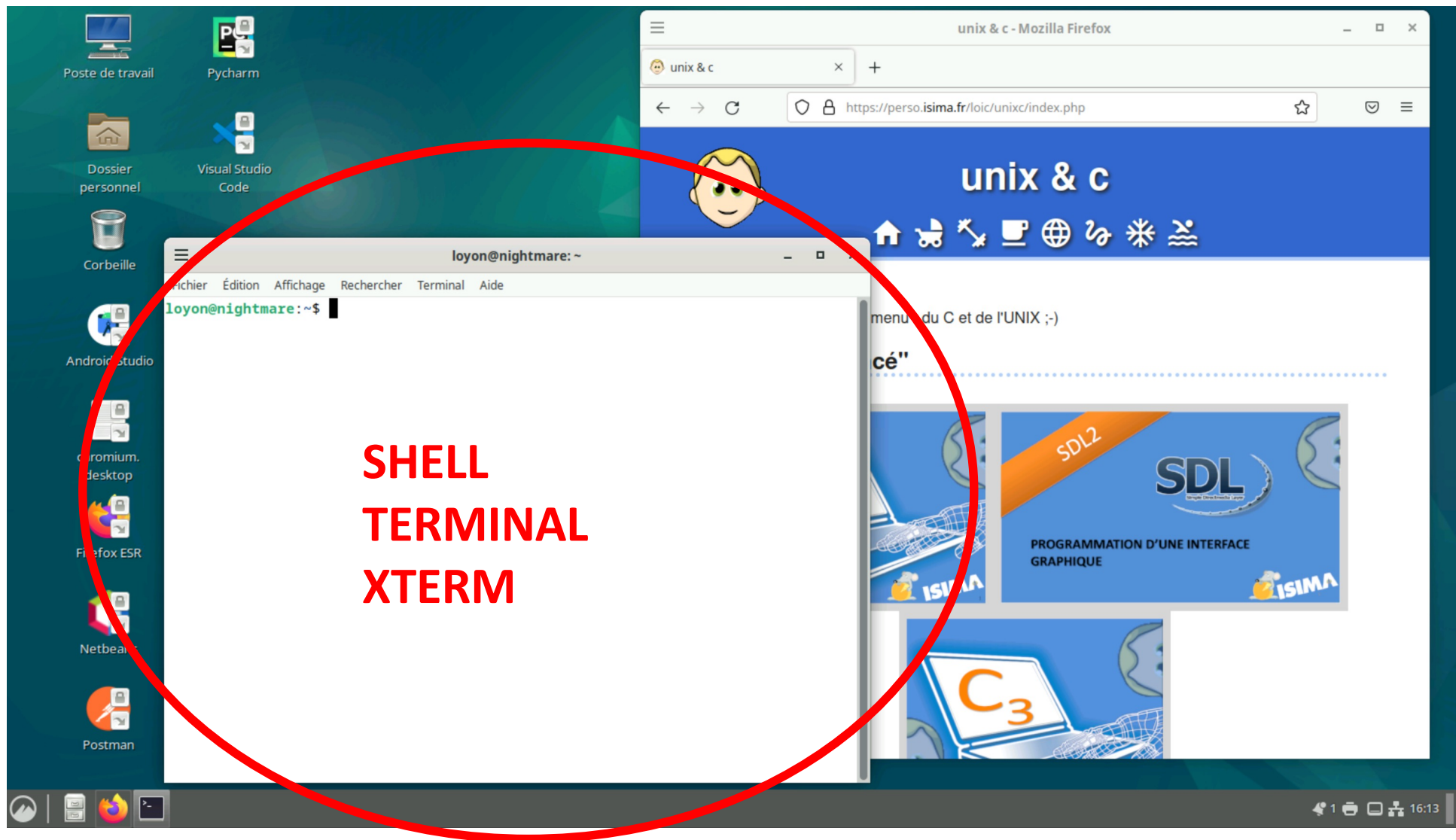
- Microsoft Azure 
  - Windows
  - Produits de développement (Visual, Visio, ...)
  - Office 365 (ENT UCA)



# UNIX ?

- Multitâches
- Multiutilisateurs
- Bonne répartition des ressources
- Outils de développement
- Intimement lié au langage C
- Portable → multi architecture

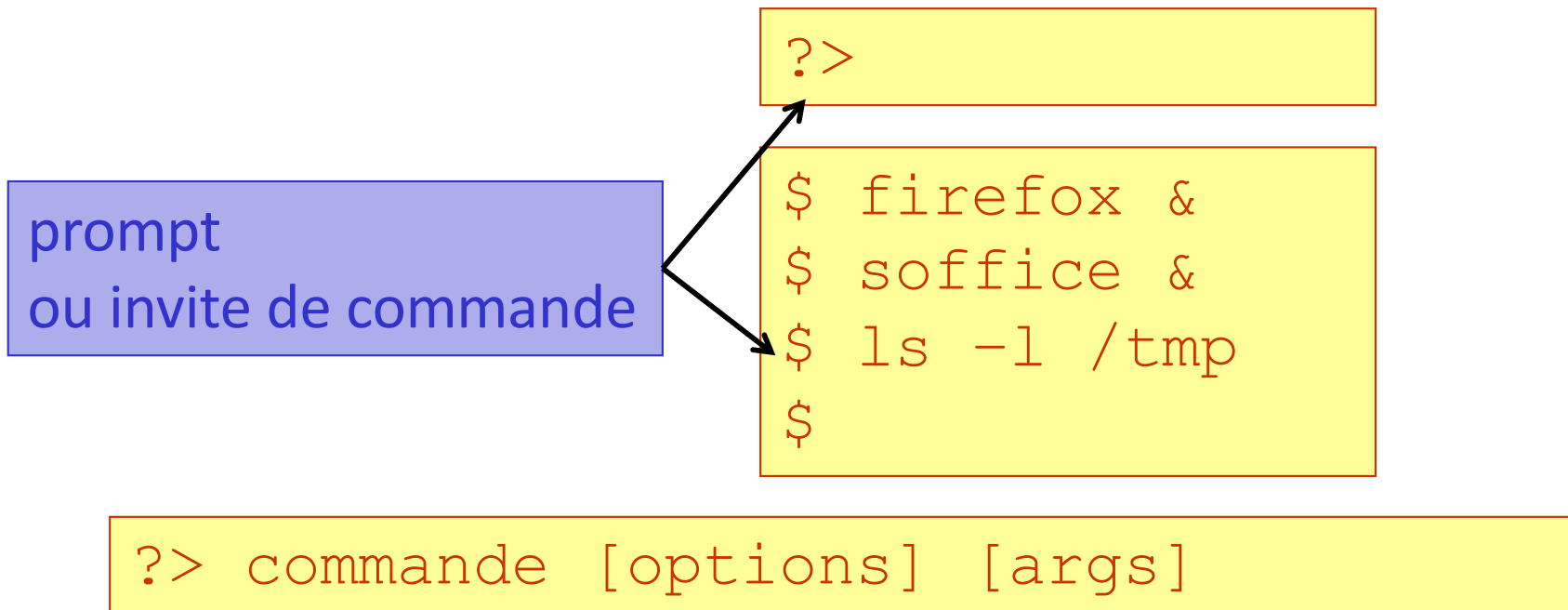
# Exemple d'interface graphique



Debian 12 > Cinnamon

# Shell (1)

- Donner des ordres au système d'exploitation
- Langage de commandes sous UNIX
  - « Interpréteur de commandes »
  - « ligne de commandes » / terminal / xterm



- Les espaces sont importants (au moins un)



# Shell (2)

- Choix du type de shell à la connexion
  - 2 grandes familles

Bourne Shell ( <b>sh</b> )	Cshell ( <b>cs</b> h)
Korn Shell ( <b>ksh</b> )	<b>tcsh</b>
Bourne Again Shell ( <b>bash</b> )	
Z shell (zsh)	

- Toute commande effectuée via une interface graphique peut être faite textuellement (pas réciproque)
- Les serveurs en production n'ont souvent pas d'interface graphique car inutile voire dangereuse

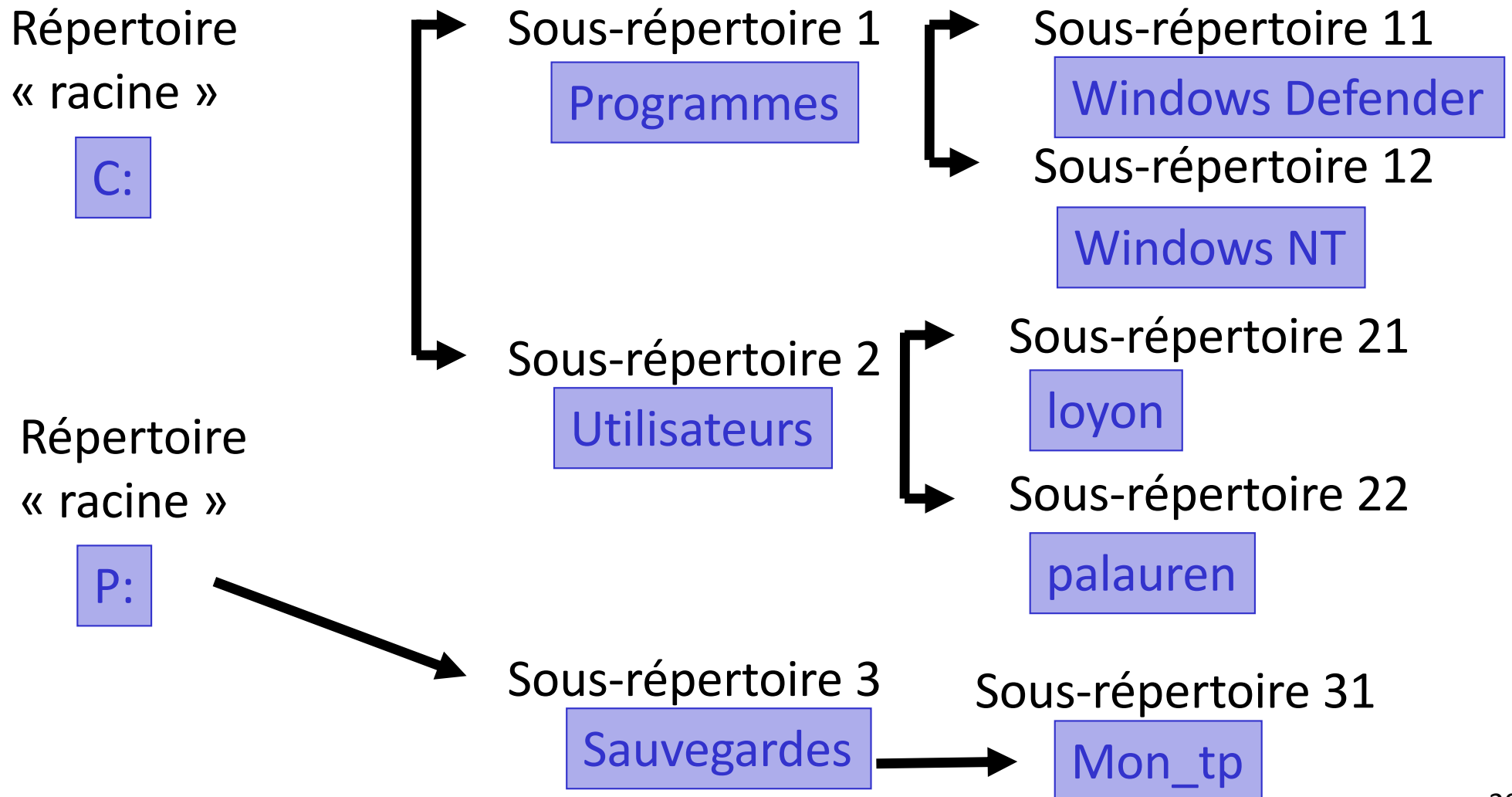
# Tout est fichier ...

- Fichiers "classiques"
  - Fichier texte
  - Fichier binaire
- Répertoires
- Liens symboliques
- Autres
  - Tubes nommés, sockets, ...
- Périphériques (*devices*)
  - Écran (s), clavier, souris (terminal / display)
  - imprimante, disque

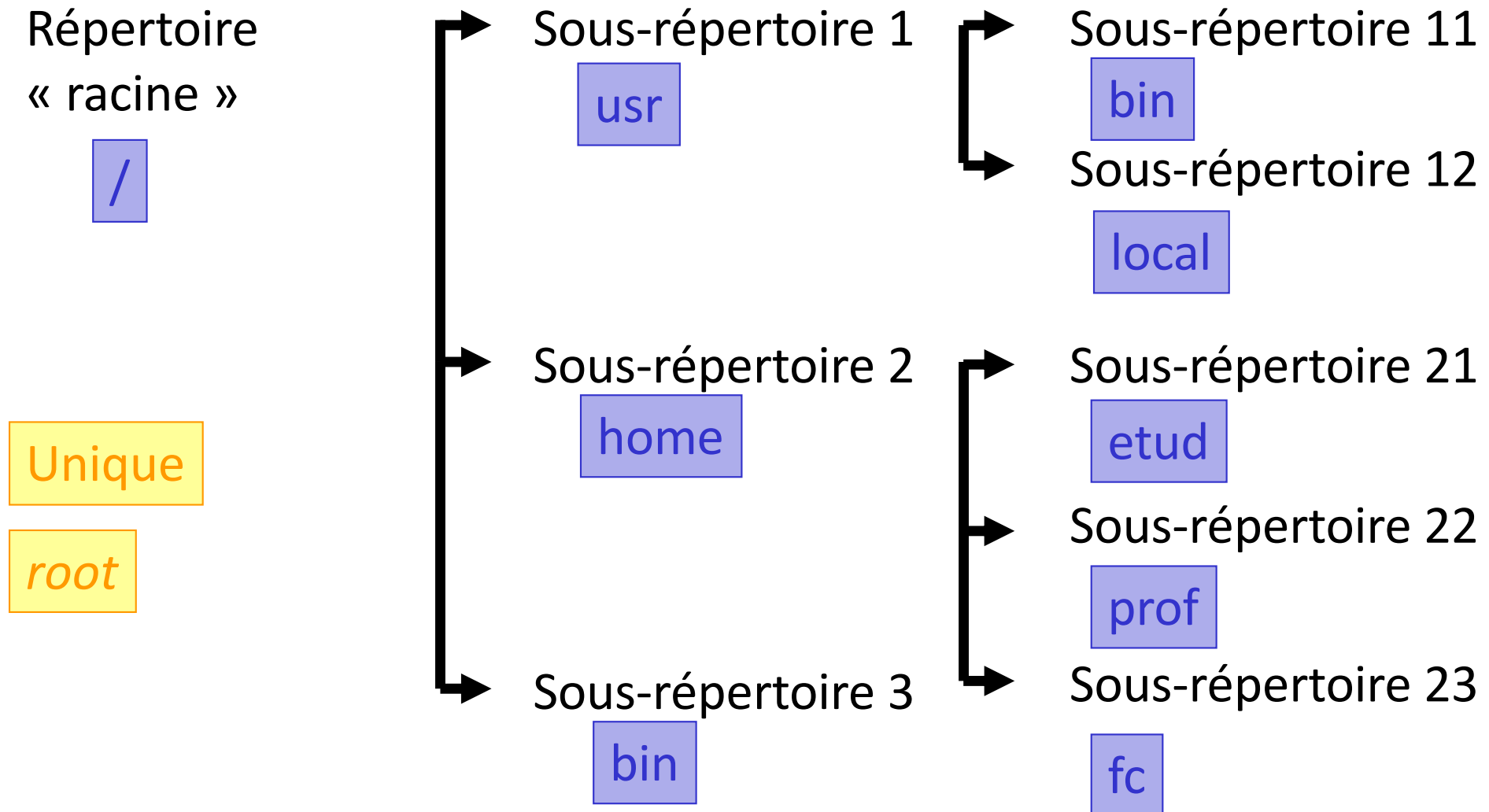
# Fichiers "classiques"

- Texte
  - Affichable et compréhensible à l'écran
  - Gestion des accents, fin de ligne ?
  - Ex : code source, script shell
- Binaire
  - Exécutable
  - Données d'une application
  - Pas lisible à l'écran
  - Ex : fichier d'un éditeur de texte (OO, Word, mp3/4)

# Structure arborescente Windows



# Structure arborescente UNIX



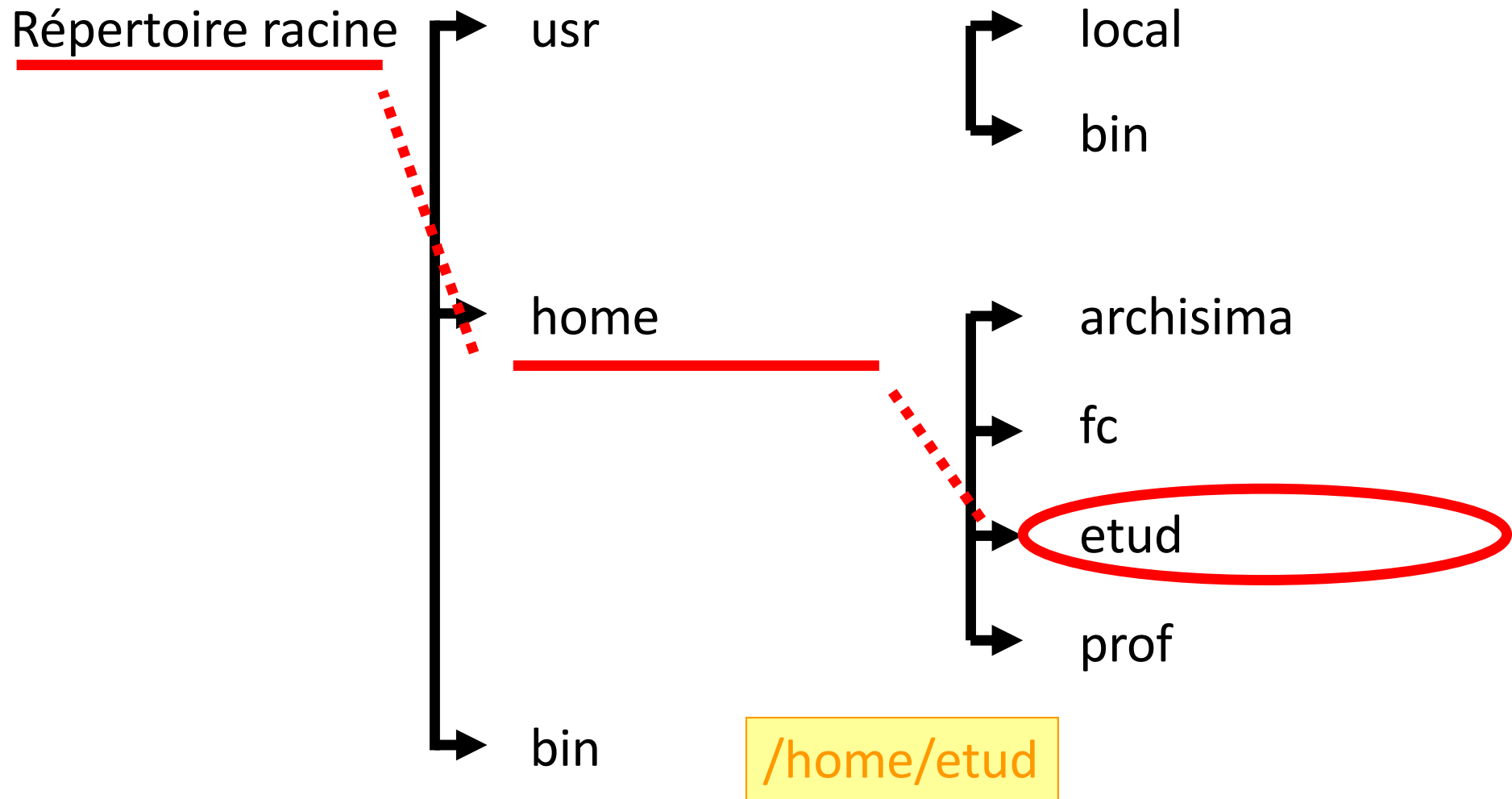
# Chemin = *path*

- Emplacement où trouver un répertoire ou un fichier
- Énumération des différents répertoires nécessaires
- Séparateur de répertoires : *slash*

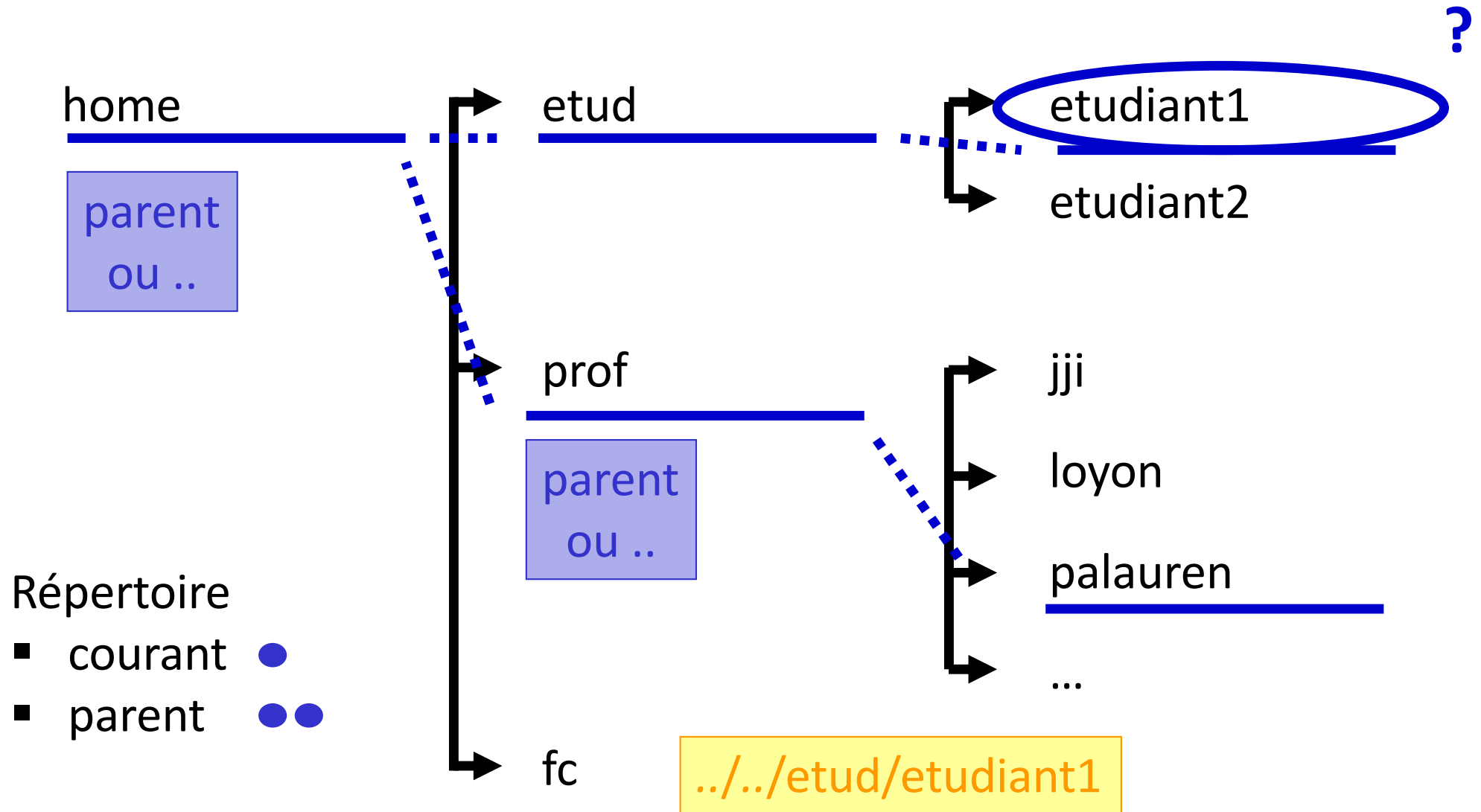
**home/etud**

- Faire attention à la casse

# Chemin absolu

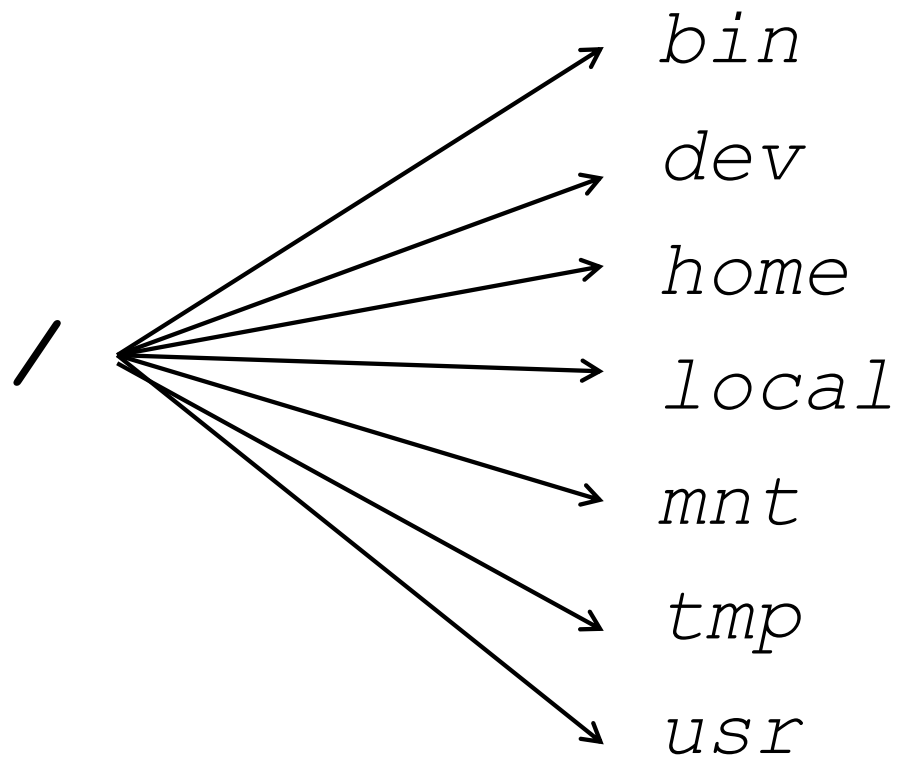


# Chemin relatif





# Répertoires particuliers



Répertoire principal  
d'un utilisateur :

`$HOME`

`~`

`~loyon`

# Commandes shell (1)

\$ ls *list segments*

```
$ ls  
$ ls -al
```

C:\> dir  
D:\>dir /p

\$ cd *change directory*

```
$ cd  
/home/local.isima.fr/loyon  
$ cd ..  
$ cd $HOME  
$ cd ~palauren  
$ cd
```

C:\> cd Windows  
C:\> d:  
D:\> cd toto

\$ pwd *print working directory*

# Commandes shell (2)

\$ mkdir *make directory*

```
$ mkdir tps
```

\$ rmdir *remove directory*

```
$ rmdir tps
```

tps doit être VIDE

\$ cp *copy*

```
$ cp source destination  
$ cp original copie  
$ cp fic.txt ../sauv  
$ cp -r * ../sauv
```

C:\>Copy source destination

\$ tree

# Commandes shell (3)

\$ mv                    *move*                    → C:\>ren source destination

```
$ mv original nouveau_nom  
$ mv fic.txt /tmp/  
$ mv * ../a_sauver
```

\$ rm                    *remove*

```
$ rm fic.txt  
$ rm -ri repertoire  
$ rm -rf /poubelle/*
```

→ C:\> del fichier



# Obtenir de l'aide ?

- Son voisin ;-)
  - En cours et en accès libre

- Internet 

- Le *manuel*

```
?>man <commande>
```

```
?>man man
```

```
?>apropos <commande>
```

```
?>info <commande>
```

# Connexion

- Environnement multiutilisateur
- Connexion sécurisée
  - Identifiant : `login`
  - Mot de passe : `password`
- Sécurité des mots de passe
  - Dictionnaire
  - Fréquence des changements
  - Caractères "spéciaux" : chiffres majuscules @ # ~ –
- Session

Le mot de passe se change sur l'ENT UCA

```
?> passwd
```

# Déconnexion / *logout*

- **Obligatoire** à la fin de chaque utilisation
  - Commande *logout* en mode texte
  - Icône ou menu en mode graphique
- Sortie propre de la session
  - Courtoisie
  - Sécurité
- Terminal "locké" pendant la mise en veille
  - Blocage temporaire sinon ...
  - Courtoisie

# Administrateur / *root*

- Utilisateur particulier avec droits
  - Privilèges systèmes
- Administration de la machine
  - Vérification de la bonne marche
  - Gestion des utilisateurs
  - Sauvegarde / installation
  - Démarrage / Arrêt
- **Interdiction**
  - de se connecter
  - d'essayer de se connecter



# Manipuler des fichiers textes

\$ touch

\$ cat

\$ more

\$ head

\$ tail

\$ vi

\$ wc

\$ which

```
$ commande fichier.txt
```

```
$ more resultats.txt
```

```
$ tail -20 fic.log
```

```
$ vi roman
```

```
$ vi
```

# VI (alias vim)



- Toujours installé (non graphique)
- Quitter `:q` `:wq` `:q!` `ZZ`
- Se déplacer `:n`
- Effacer `x`
- Copier-coller `yy` `dd` `p` `P`
- Chercher-remplacer `:/s/n1/n2`

# Cours 2

Droits

Liens

Redirections

Find

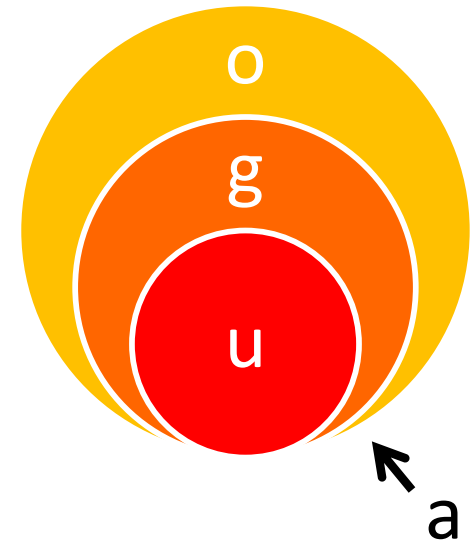
# Protection des données

- Environnement multi utilisateurs
- Objets protégés
  - Répertoires
  - Fichiers
  - Commandes
  - Ressources
- Utilisateurs
  - Administrateur (root)
  - Utilisateur

DROITS /  
PRIVILÈGES



# Droits "usuels"



- 3 niveaux

- Propriétaire *user* **u**
- Groupe du propriétaire *group* **g**
- Autres *others* **o**

Tous les utilisateurs  
*all* **a**  
( **a = u g o** )

- 3 types de droits

- Lecture *read* **r**
- Écriture *write* **w**
- Exécution *execute* **x**

# Commandes

```
$ ls -l
```

```
-rwxr--r--  1 loic users  8699 Sep 08 2023 Xfloodit.c  
drwxr-xr-x  2 loic users  4096 Sep 08 2023 xlib  
-rwxr--r--  1 loic users  7237 Sep 08 2023 xlib2.c
```

```
$ chmod
```

d rwx rwx rwx  
u g o

```
$ chgrp
```

```
$ chown
```

Administrateur !

# chmod

```
$ chmod +x dir
```

```
$ chmod -R a+r index.html
```

```
$ chmod go-rwx fic.log
```

```
$ chmod -R u+rwx,go= *
```

```
$ chmod 755 fic
```

<b>r</b>	<b>w</b>	<b>x</b>
<b>4</b>	<b>2</b>	<b>1</b>

<b>d</b>	<b>u</b>			
<b>d</b>	<b>rw</b>	<b>x</b>	<b>rw-</b>	<b>rw-</b>
	<b>7</b>			

# Quelques droits particuliers

- *sTicky bit*

- Permet à un exécutable de rester en mémoire même s'il est terminé
- Permet à tout le monde d'écrire dans un répertoire, mais rend impossible d'effacer les fichiers des autres.

```
drwxr-xr-t 25 root root 680 Sep 08 23 tmp
drwxr--r-x 13 root root 4096 Sep 08 23 usr
```

- *Suid*

- Sur un fichier exécutable
- Permet de lancer le fichier avec le droit du propriétaire ou du groupe

```
-rwsr-xr-t 1 root root 63738 Sep 08 23 passwd
```



# Lien physique

## \$ ln fichier raccourci

- Deux noms distincts pour un même fichier
- Original effacé ? Lien toujours valide
- Même numéro d'inode
- Limité à une partition
- Limité aux fichiers

INODE = numéro unique attribué au fichier par le SE

```
ls -i
```

# Lien symbolique / Raccourci ?

```
$ ln -s fichier raccourci
```

- Fichier / répertoire
- 2 inodes différents
- Original effacé ? Lien invalide
- Passe la limite de la partition
- Très utilisé pour éviter la duplication de données

```
$ which firefox  
$ ls -l /usr/local/bin/firefox  
[...] firefox -> firefox91esr
```

# Path (1)

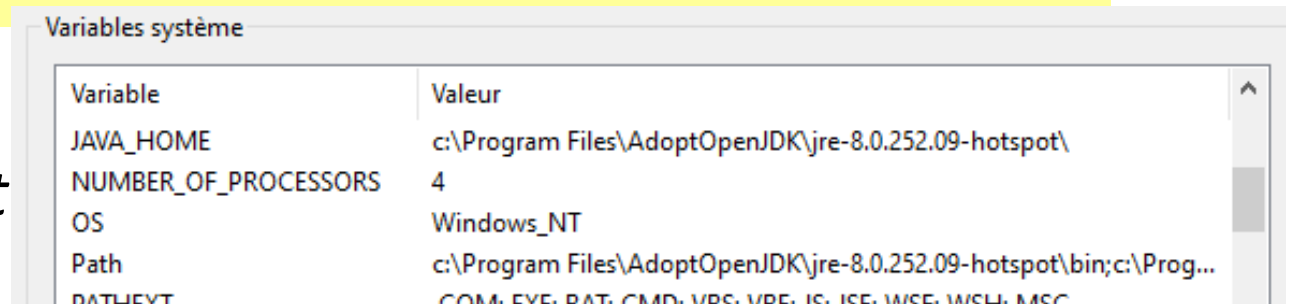
- Liste de répertoires où trouver les programmes

```
$ waterfox  
bash : waterfox: command not found $
```

- Visualisation

```
$ echo $PATH  
/usr/lib64/qt8/bin:/usr/local/sbin:/usr  
/sbin:/bin:/usr/local/bin  
$
```

Sous windows :  
*variable d'environnement*  
Commande : PATH



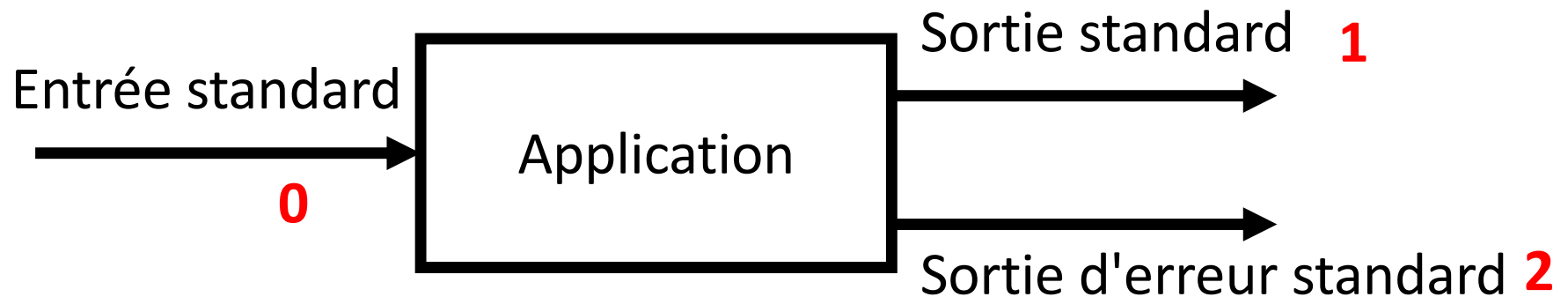
Variable	Valeur
JAVA_HOME	c:\Program Files\AdoptOpenJDK\jre-8.0.252.09-hotspot\
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	c:\Program Files\AdoptOpenJDK\jre-8.0.252.09-hotspot\bin;c:\Prog...
PATHLEV1	COM, EVE, PAT, CMD, VBS, VBE, IC, ICE, WCE, WCL, MCF

# *Path (2)*

- Le répertoire courant ?
  - "." : peut être ajouté au PATH, mais pas obligatoire
- L'ordre des répertoires est important
- Commandes utiles
  - *whereis* : permet de rechercher un fichier dans l'arborescence connu
  - *which* : permet de savoir quel exécutable est utilisé
  - *locate fichier* : permet de localiser fichier dans un index construit au préalable ( commande `root updatedb`)
  - *find* : permet la recherche

# Redirections

- Une application = boîte noire
- Fichiers standards
  - Entrée standard
  - Sortie standard
  - Sortie d'erreur standard



- Ces fichiers peuvent être remplacés par d'autres fichiers plus classiques

```
$ cat riri fifi loulou > famille
```

```
$ cat donald >> famille
```

```
$ ps -aux > res
```

```
$ more res
```

Ne pas appeler un fichier test

```
$ commande 2> fichier
```

```
$ wc < fic ⇔ wc fic
```

```
$ moyenne < donnees
```

```
$ commande < fichier
```

```
$ moyenne
```

```
Saisir un nombre (-1 pour arrêter)
```

```
15
```

```
20
```

```
13
```

```
-1
```

```
La moyenne est : 16 (3 éléments)
```

```
$
```

```
15
```

```
20
```

```
13
```

```
-1
```

```
$ moyenne < liste1.txt
```

```
Saisir un nombre (-1 pour arrêter)
```

```
La moyenne est : 16 (3 éléments)
```

```
$
```

# FIND

**find <répertoire> <critères de sélection> <action>**

- Exploration récursive dans les répertoires
- Traite/Affiche les fichiers qui répondent aux critères
- Les critères de sélection peuvent être multiples

```
$ find rep -name nom -print
```

```
$ find . -name "essai.c" -print
```

```
$ find / -name nom -print 2> /dev/null
```

```
$ find . -type f -size +2M -print
```



# Caractères génériques

- Remplacer un ou plusieurs caractères
  - \* : toute chaîne de caractères, y compris la chaîne vide.

```
$ ls fic*  
fic1, fic2, fichier1, fichier2 $
```

- ? : un seul caractère

```
$ ls fic?  
fic1, fic2 $
```

# Cours 3

Processus

Pipes

# Processus (1)

- Programme en train de s'exécuter
  - Instructions machine
- Ensemble des données manipulées
  - Mémoire
- Informations utiles au système
  - Contexte d'exécution (pile, fichiers, ...)
- Problème : gestion du temps / processeur
  - Ordonnanceur
  - Multiprocesseur
- Numéro unique (*PID*)

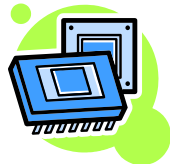
# Processus (2)

- Processus **ystème** ou *daemon*
  - Pas associé à un terminal
  - Propriétaire : administrateur
- Processus **utilisateur**
  - Tâche particulière
  - Durée de vie limitée
  - Généralement associé à un terminal
- Nombre de processus limité par le système



# Etats d'un processus

• Actif



• *Ready*

▪ Prêt à être exécuté



• Terminé

• Endormi / suspendu

▪ En attente d'une saisie par exemple



• Zombie (*defunct*)

▪ Terminé mais toujours référencé (PID)



# Process Control Block (PCB)

- Structure de données décrivant le processus
- IDs : **PID**, processus parent (PPID) , utilisateur (UID)
- Etat : nouveau : prêt, en attente, exécution
- Valeurs des registres
- Compteur ordinal (pointeur d'instruction)
- Pointeur de pile : indique la position du prochain emplacement disponible dans la pile mémoire
- Espace d'adressage du processus ;
- Les descripteurs de fichiers ;
- La liste de gestion des signaux ;
- Le temps d'exécution cumulé
- D'autres informations...

# Liste des processus

```
$ ps
```

PID	TTY	TIME	CMD
1224	pts/1	00:00:00	bash
3836	pts/1	00:00:00	ps

```
$ ps -l
```

- PPID, UID

```
$ ps -aux    ou    ps aux
```

```
$ ps -ef
```

```
$ top
```

# Commande *kill*

- Demande de fin

```
$ kill -15 PID      $ kill PID
```

- Mauvaise fin

```
$ kill -9 PID
```

- Arrêt temporaire

```
$ kill -STOP PID
```

- Reprise après arrêt

```
$ kill -CONT PID
```



# Modes d'exécution des processus

- Avant-plan (*foreground*)
  - Exécution séquentielle
  - Interaction avec l'utilisateur
- Mode arrière-plan (*background*)
  - Utilisation concurrente
- Job
  - Processus en arrière-plan
  - Processus suspendu
  - Processus graphique détaché

# Contrôle de processus (1)

\$ commande ← interactif

```
$ vi options.txt
```

\$ commande & ← batch

```
$ firefox &
```

```
[1] 1720
```

```
$ xeyes &
```

```
[2] 2345
```

[n° Job] PID

\$ jobs

```
[1] En cours d'exécution firefox &
```

```
[2] En cours d'exécution xeyes &
```

# Contrôle de processus (2)

```
$ xeyes &
```

```
[3] 1837
```

```
$ fg
```

```
xeyes
```

```
[CTRL+Z]
```

```
[3] Stoppé xeyes &
```

```
$ bg
```

```
[3] xeyes &
```

```
$ kill %3
```

```
$ fg
```

```
$ fg numéro
```

```
$ fg %numéro
```

```
$ bg
```

```
$ bg numéro
```

```
$ bg %numéro
```

```
$ kill %numéro
```

```
[CTRL + C]
```

```
Arrêt prog en cours
```

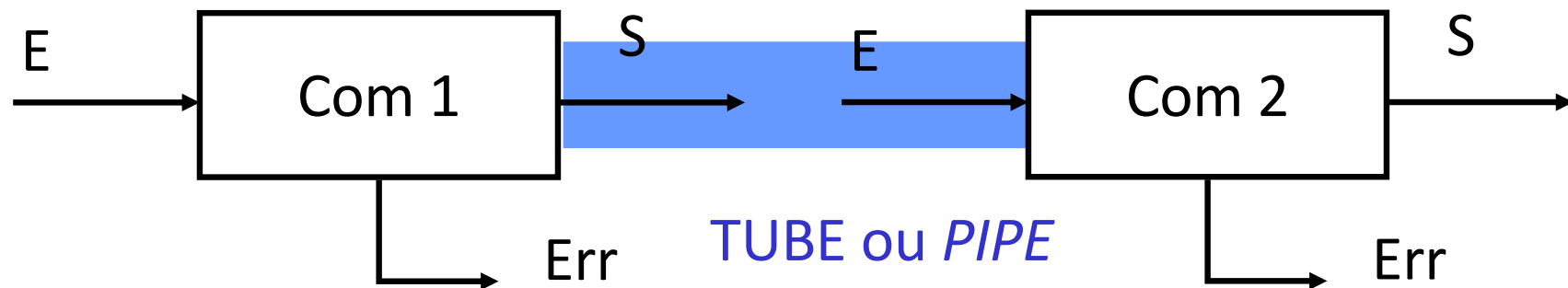
# Exécution de processus

- Séquentielle
  - Processus après processus

```
$ (c1;c2;c3;...;c4) > fichier
```

- Concurrente
  - En même temps

# Exécution concurrente



Idée : Lancer des processus "en parallèle"  
communiquant entre eux

Com 2 peut commencer avant que Com 1 ne termine

Rediriger la sortie de l'un vers l'entrée de l'autre

```
?>Com1 | Com 2  
?>ps -aux | less
```

```
cat fichier | grep loic
```

# NOHUP

- Déconnexion  $\Rightarrow$  Arrêt des programmes
- Calcul long ?
  - *No hang up* (signal d'arrêt)

```
$ nohup commande  
    [< fichier1] [> fichier2] &
```

Désactivé sur les machines étudiantes

# Autres commandes

\$ *at heure*

- Exécuter les commandes de l'entrée *std* à la date spécifiée

\$ *crontab*

- Tâches périodiques en arrière-plan
- Fichier *crontab*, démon *cron*

```
?> crontab -l  
?> crontab -e
```

Commandes soumises à autorisation



# FBI

\$ rm \*

- Sans savoir où l'on est (pwd)

\$ rm \* ; o

- Au lieu de rm \* . o

\$ mv à 14h du mat

- Appeler un fichier du nom d'une commande ou d'un périphérique : null, test



# Variables d'environnement

- DISPLAY
- ENV
- HOME
- PATH
- PS1 et PS2
- PWD
- SHELL
- TERM
- USER

```
$ echo $PATH
$ vi .bashrc
PATH=$PATH:~/bin

$ source .bashrc
    ou
$ . .bashrc
$ echo $PATH
```

# alias

```
$ alias rm="rm -i"
```

```
$ alias ll="ls -l"
```

```
$ ll
```

```
$ alias
```

```
$ unalias
```

# Regrouper des fichiers

- Mettre des fichiers dans un seul fichier pour pouvoir les transporter/archiver plus facilement
- Non compressé
  - Commande tar
  - Options c v x f
- Compressé
  - zip / unzip
  - gzip[2] / gunzip
  - compress / uncompress

**tar cvfz arch.tar repertoire**

.z  
.gz  
.zip

# Fichiers de configuration

- Fichiers ou répertoires cachés
- Initialisation (connexion/session)

`.bash_profile`

`.bashrc`

A MANIPULER  
AVEC PRECAUTION

- Application

`.exrc`

`.mozilla`

# Script shell (1)

- Liste de commandes dans un fichier texte
  - Fichier *batch*
- 2 familles de syntaxe
  - *Bourne shell* et dérivés (bash)
  - *C-shell* et dérivés (tcsh)
- Choisir le shell
  - Spécifié à la première ligne du script
- Scripts autorisés sont dans `/etc/shells`

# Script shell (2)

- Première ligne du script

```
#!/bin/bash
```

- Rendre le fichier exécutable

```
chmod u+x script
```

- Exécuter

```
$ ./script
```

```
#!/bin/bash  
  
# dire bonjour  
echo "Bonjour"  
echo $USER
```

# Cours 4

Réseau  
Web

# Réseau



- Communication entre des ordinateurs
  - Reliés entre eux (avec ou sans fil)
- Internet
  - La toile
  - *World Wide Web WWW*
- Protocoles
  - Manière de communiquer
  - IP, TCP, UDP, SFTP, SMTP, HTTPS, POP/IMAP ...

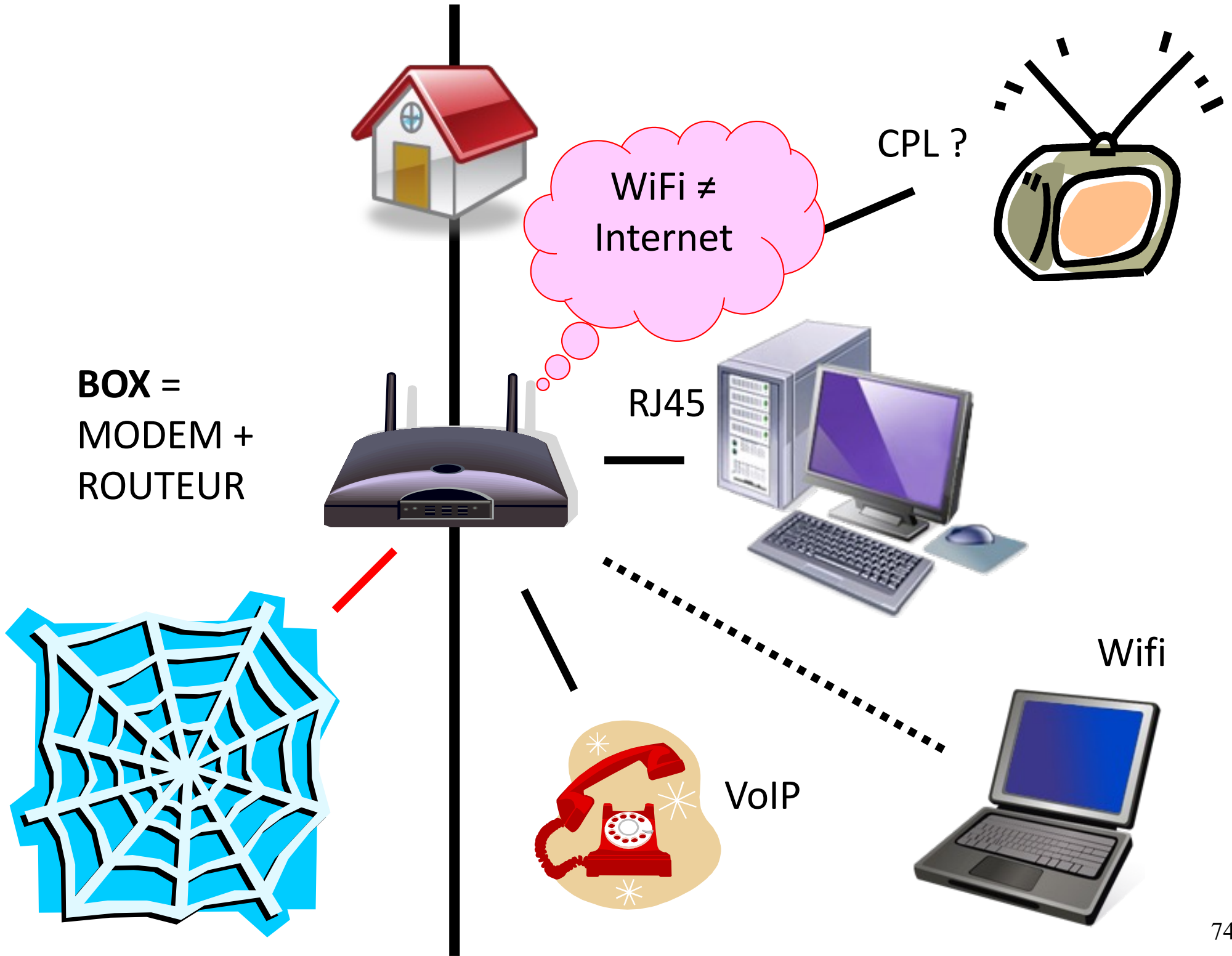


# Adresse ?

- **URL** *Uniform Resource Locator*
  - `https://www.isima.fr:80`
- **DNS** *Domain Name Server*
  - Transcription Nom ↔ Adresse IP
  - `https://193.55.95.26`
  - Sinon fichier `/etc/hosts`

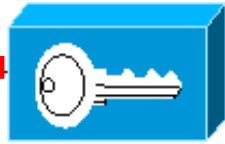
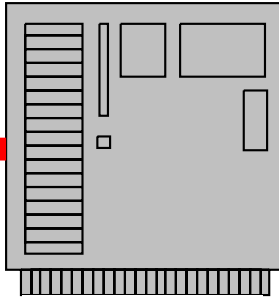
- DHCP

```
127.0.0.1    localhost
127.0.0.1    www
193.55.95.26 www.isima.fr
```



Pare-feu  
Proxy

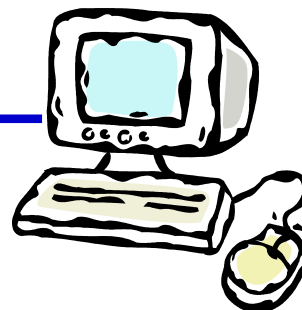
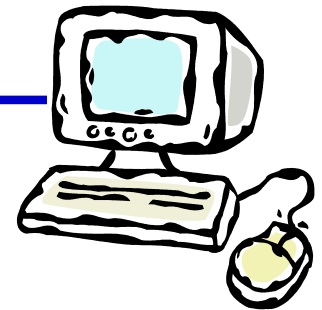
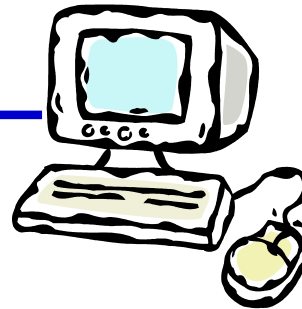
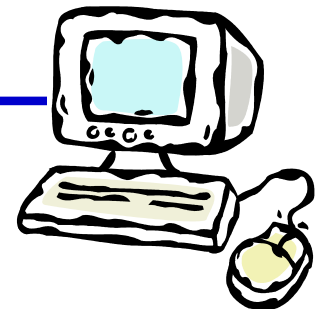
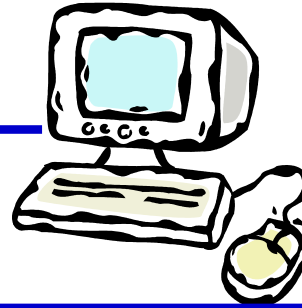
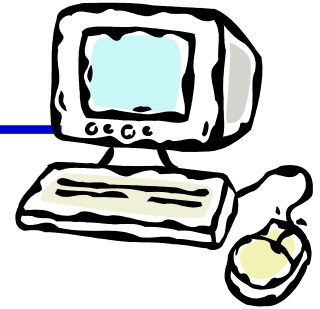
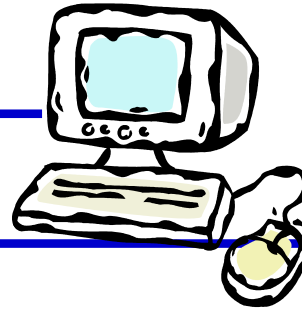
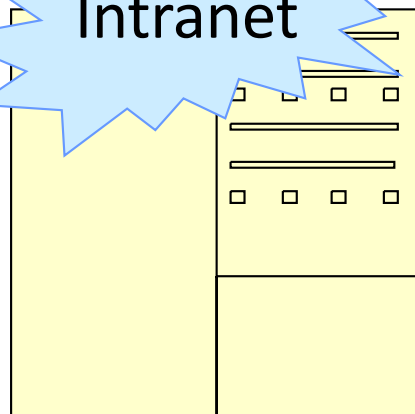
Serveurs



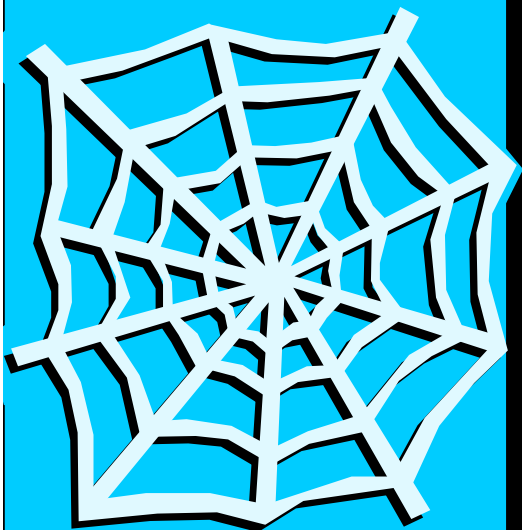
Concentrateur  
VPN

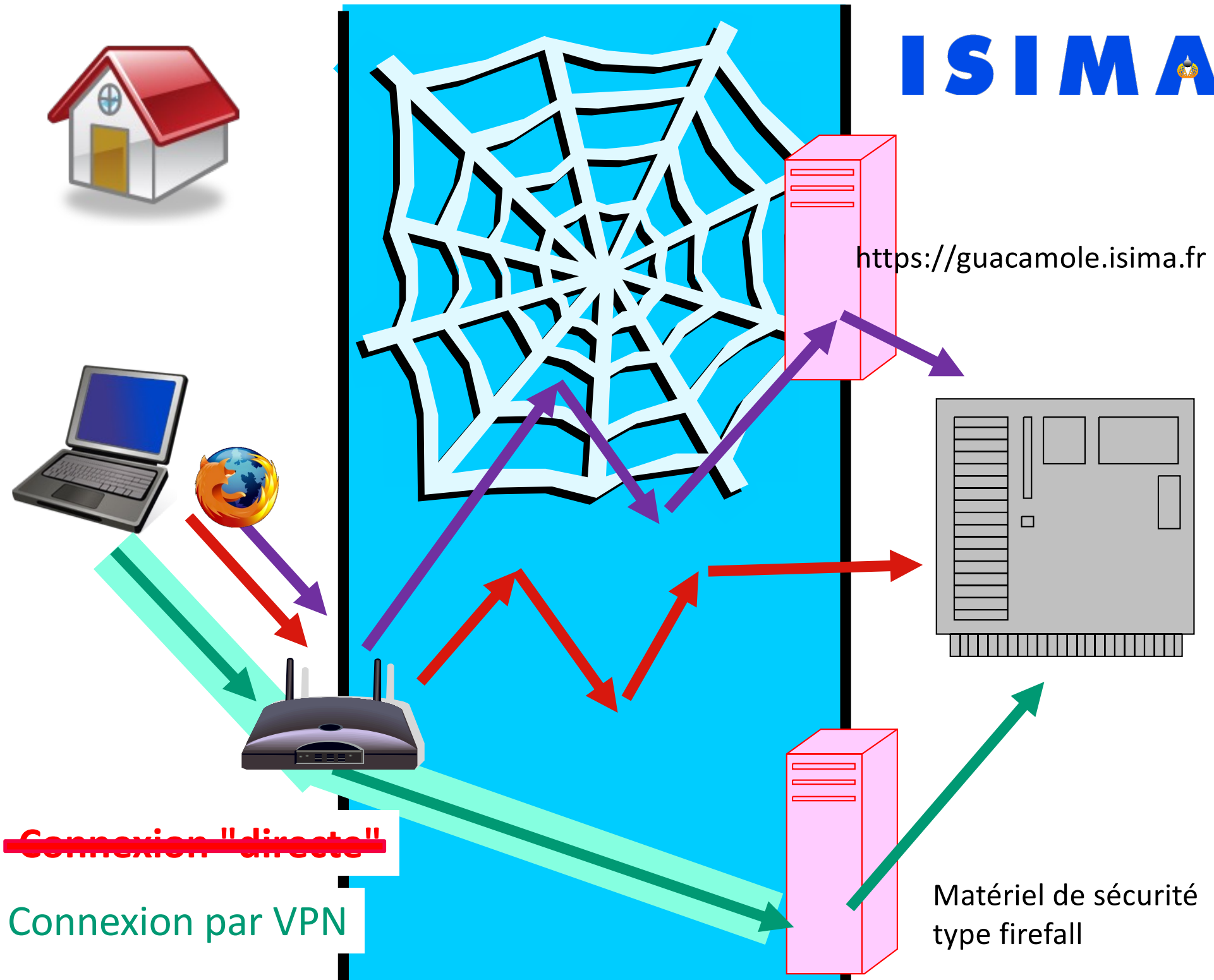
**ISIMA**

Intranet



TX





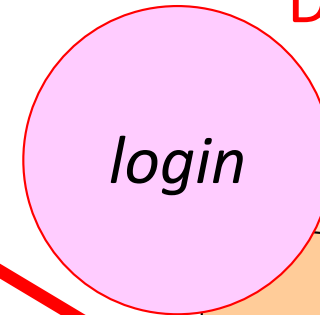
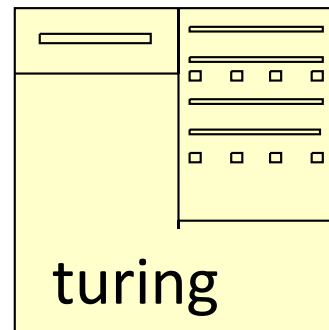
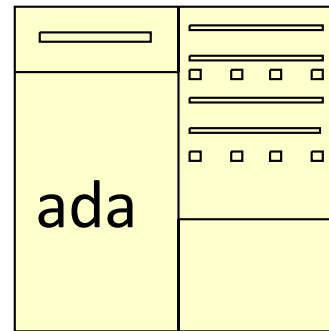
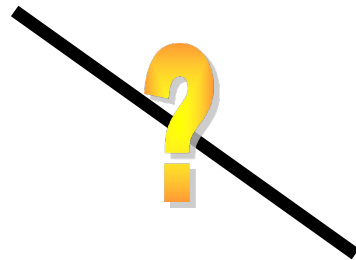
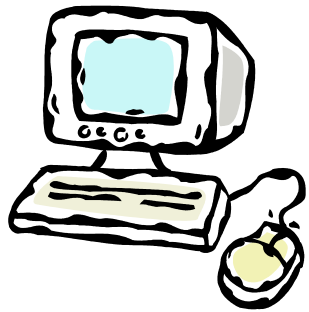
~~Connexion "directe"~~

Connexion par VPN

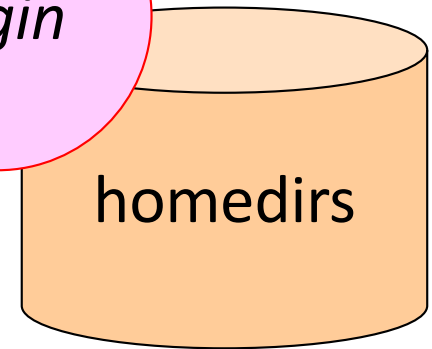
<https://guacamole.isima.fr>

Matériel de sécurité  
type firefall

# Mes données ?



Données  
Dupliquées?



/local.isima.fr/

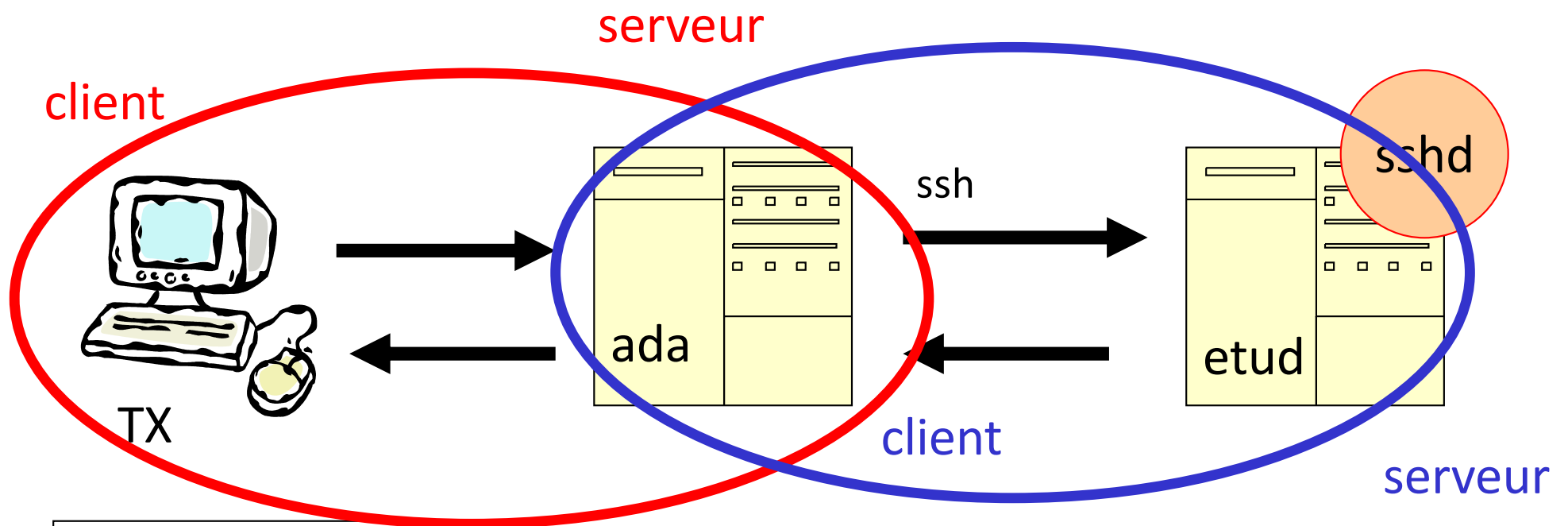
Système de  
fichiers  
Partagés par *shared*



Annuaire

# Connexion à distance classique

« Manipuler une machine comme si on y était »



```
ada> ssh etud
```

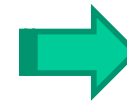
```
etud>
```

Relations client-serveur

# ssh / telnet

- ssh
  - Communication sécurisée
  - Pas dispo sous Windows < 10

- `ssh -X login@machine`



- PuTTY
- Virtualisation
- WSL

- telnet (obsolète)
  - Connexion en clair
- Adresse IP ou nom logique
- Résolution nom via DNS ou /etc/hosts

```
$ ssh loic@etud
loic@etud's password :

Last login: Mon Sep  2 10:53:43 2022 from
172.16.32.21

$ hostname
etud
$ who am i
loic      pts/0          2022-09-02 12:02
(192.168.79.48)
$ last

$ ping www.isima.fr

$ exit (ou quit)
connection to etud closed
```



# Transfert de fichiers

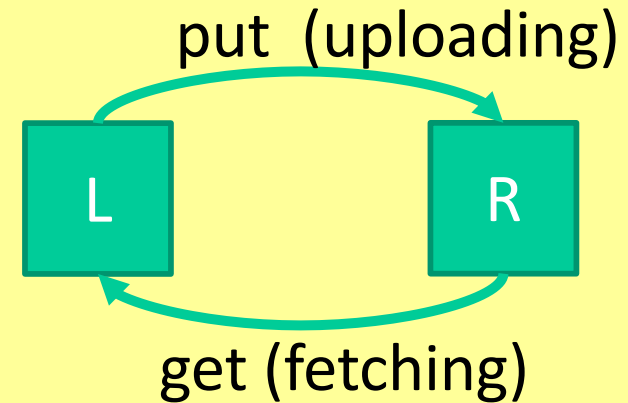
- Commande `scp`

```
scp toto.txt loic@perso.isima.fr:temp
```

- Clients texte :
  - `sftp` : sécurisé
  - `ftp` : obsolète
- Client graphique. Ex : filezilla
- **ATTENTION** : de l'extérieur, utiliser le VPN

# [s]ftp

```
$ sftp loyon@etud
loyon@etud's password :
Connected to loyon@etud.
sftp> pwd
Remote working directory:
/home/local.isima.fr/loyon
sftp> ls
xlib.c
sftp> get xlib.c
Fetching /home/local.isima.fr/loyon/xlib.c
to xlib.c
/home/local.isima.fr/loyon/xlib.c
100% 7850      31.7KB/s   00:00
sftp> bye
$
```



[|]cd

[|]ls

[|]pwd

cp

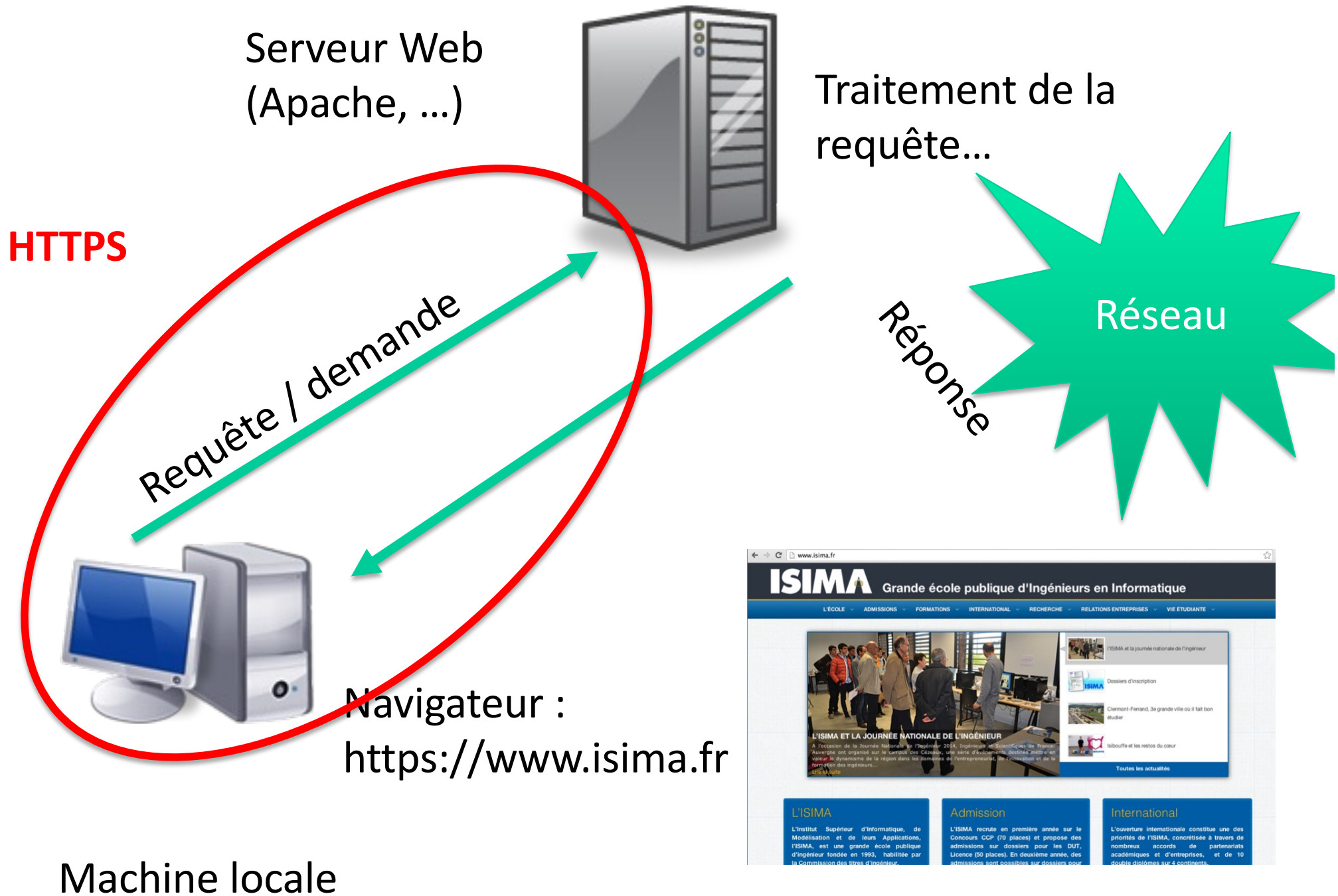


# Site internet perso ?

- Charte sur le contenu

`https://perso.isima.fr/~login`

- Répertoire `$HOME/shared/public_html`
- Attention aux droits (a+rx) des répertoires et des fichiers
- `index.html` doit exister avec droits
- Connexion `sftp` impossible sans VPN





# Hyper Text Markup Language

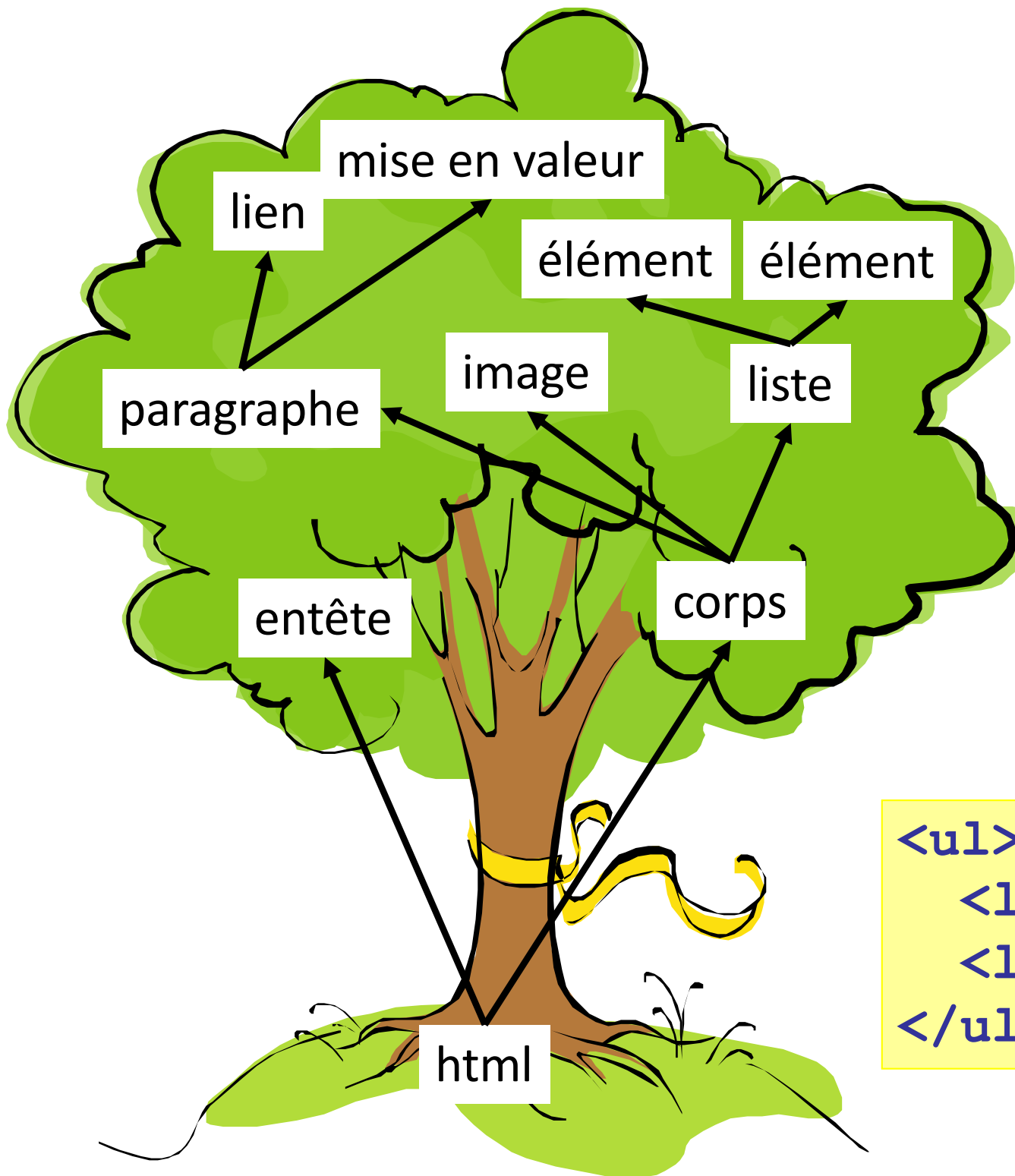
- Interprété par le navigateur
  - Standard ?
- Mélange "subtil"
  - Informations non affichées
  - Contenu (sémantique)
  - Forme (style)
- Langage ?
  - Balise | élément
  - Attribut



```
<p>paragraphe</p>
```

```

```



```

<p> texte
  <a>lien</a>
</p>

```

- Éviter d'oublier les balises fermantes
- Ne pas "croiser" les balises

```

<ul>
  <li>puce 1</li>
  <li>puce 2</li>
</ul>

```

```
<!doctype html>
<html>
  <head>
    <title> Page du loïc </title>
  </head>
  <body>
    <p>Bienvenue sur ma page</p>
  </body>
</html>
```

Fichier texte : index.html

Texte du premier paragraphe  
sur plusieurs lignes

Texte du 2eme

```
<p> ... </p>
```

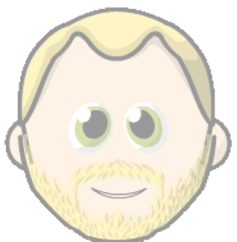
```
<p> ... </p>
```

Cliquer lien

```
<p><a href="lien.html"> lien</a></p>
```

C'est **important** ou *pas* !

```
<p>C'est <b>important</b> ou  
<i>pas</i> !</p>
```



```

```



# Titre important

```
<h1>Titre important</h1>
```

## Titre moins important

De h1 à h6

```
<h3>Titre moins important</h3>
```

### Liste de courses :

- Croissant
- Café

```
<p>Liste :</p>  
<ul>  
  <li>Croissant</li>  
  <li>Café</li>  
</ul>
```

`<ol>` pour une liste ordonnée

# Style (1)

- Style | apparence par défaut
- Pour changer
  - Attribut style (pour une balise)
  - Balise `<style></style>` dans l'entête (`<head>`)
  - Fichier texte d'extension **css**

```
<p style="color : red">texte</p>
```

```
... style="background-color : yellow" ...
```

```
... style="p1:v1;p2=v2" ...
```

# Que peut-on changer ?

- Couleurs (police, fond, dégradés)
  - Tailles (largeur, hauteur, police)
  - Marges (internes ou externes)
  - Bordures
  - Positions
  - Animations
- 
- S'il y a une erreur, l'information est abandonnée
  - S'il y a des informations contradictoires, la dernière donnée gagne

# Qui ?

- Par type d'élément

```
Tous les liens
```

- Un élément **unique** sur la page

```
<p id="moi"> ... </p>
```

- Un élément ayant une certaine classe

```
<p class="bleu"> ... </p>  
<h1 class="bleu rouge"> ... </h1>
```

- Une sélection !

```
Jocker ....
```

```
<p>
```

```
p {  
    text-align : justify;  
}
```

Tous les paragraphes !!

```
<p class="just">
```

```
p.just {  
    text-align : justify;  
}
```

Tous les paragraphes de **classe** "just"

```
<p id="just">
```

```
#just {  
    text-align : justify;  
}
```

L'élément **UNIQUE** "just"

# Lien style/HTML

```
<html>
  <head>

    <link rel = "stylesheet"
          href="fichier.css" >

    <style>
      img {
        border : dashed red;
      }
    </style>
  </head>
  <body>
    <p style="color:green" >vert</p>
  </body>
</html>
```