

# Unix par le shell

---



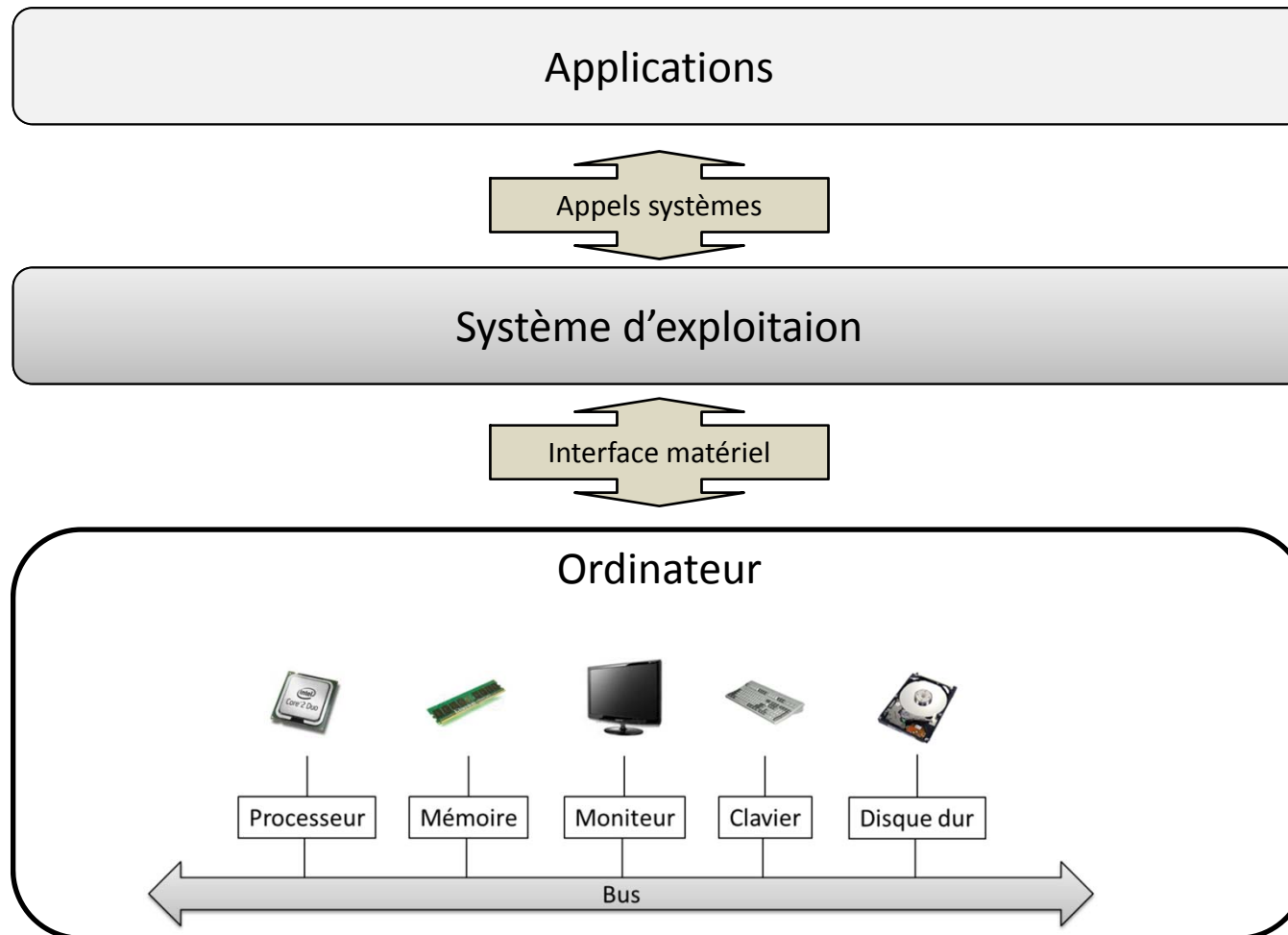
Laurent Réveillère

Enseirb-Matmeca  
Département Télécommunications

`reveillere@enseirb-matmeca.fr`  
`http://uuu.enseirb-matmeca.fr/~reveille/`

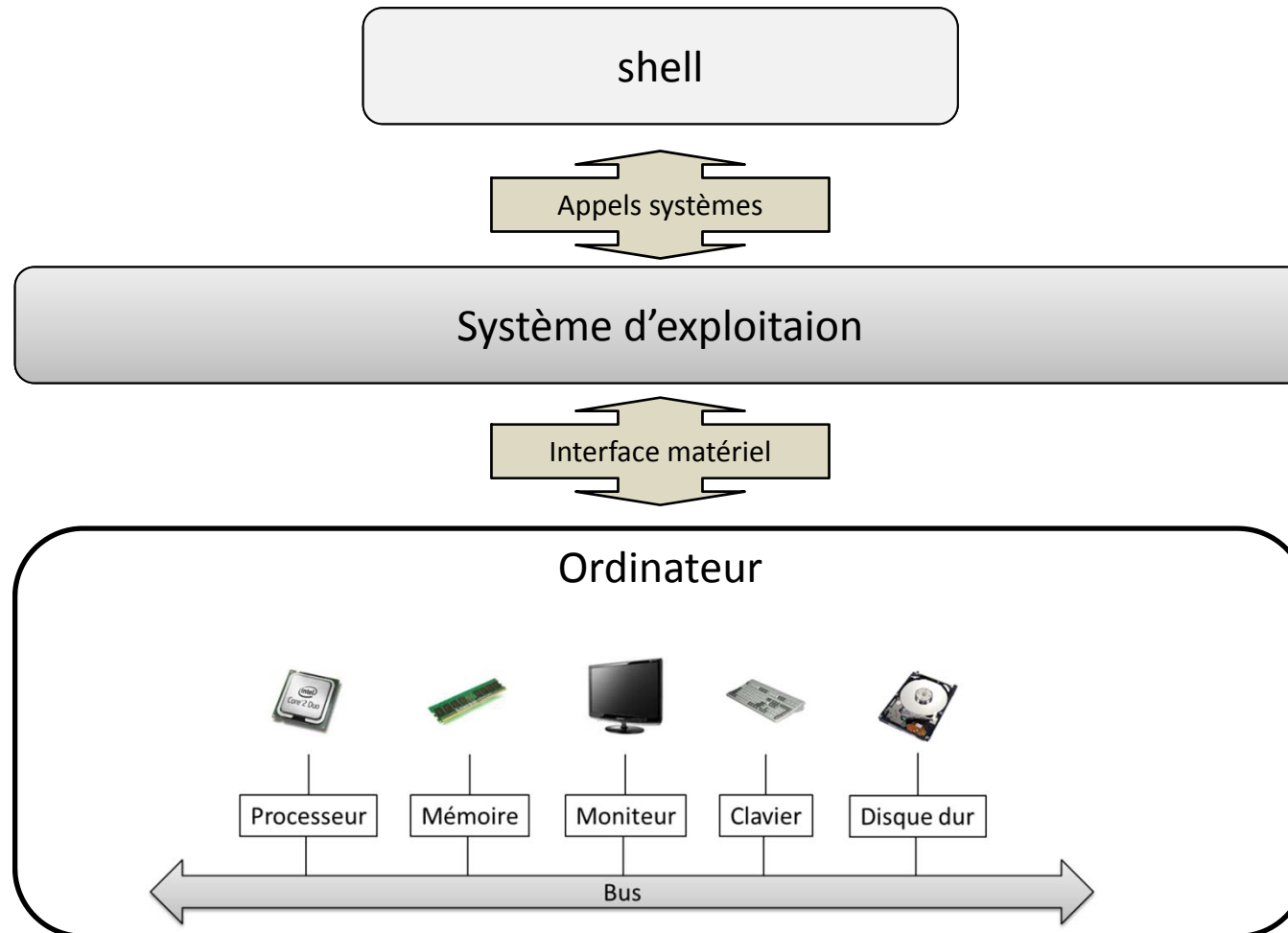


# Vue complète d'un SE





# Unix par le shell





# Manipulation des fichiers

---

## □ Fichier

- Nom de fichier
- Séquence d'octets

## □ Abstraction fichier utilisée pour

- Fichiers simples
  - » structure dépendant de l'application
- Répertoires
- Fichiers spéciaux
  - » périphériques



# Opérations sur les fichiers simples (Shell)

---

- Affichage

  - `$ cat f`

  - `$ more f`

- Création

  - `$ > f`

- Destruction

  - `$ rm f`

- Renommage

  - `$ mv f1 f2`

- Copie

  - `$ cp f1 f2`

- Aide

  - `$ man`



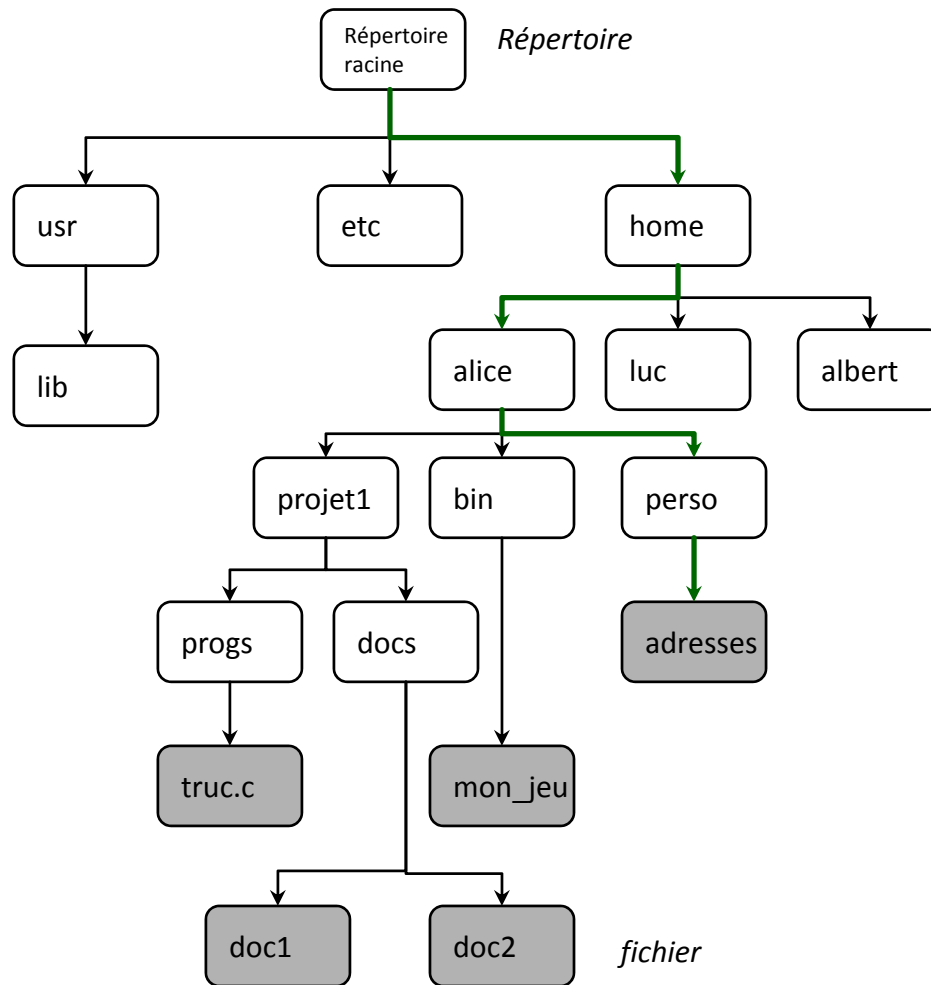
# Conventions de nommage des fichiers

---

- ❑ Nom `'.'` extension
  - Programmes sources : `.c` (C) `.p` (Pascal)
  - Fichiers textes : `.txt`
  - Fichiers de session : `.log`
  - Fichiers du traitement de texte Word : `.doc`
  - Fichiers pour formateur de texte Latex : `.tex`
- ❑ Nommage des répertoires
- ❑ Nom de fichiers de configuration : `'.'` nom



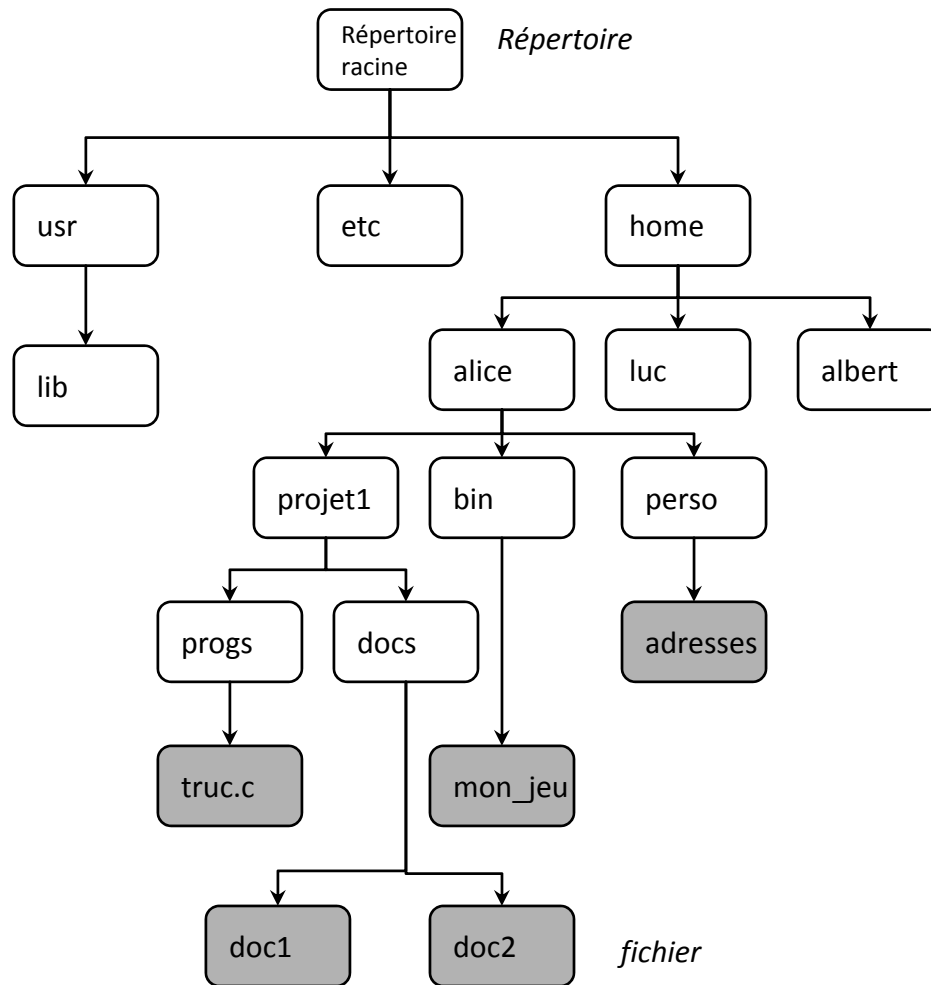
# Manipulation de fichiers: répertoires



- Organisation hiérarchique des fichiers
- Arborescence Unix
- Répertoire racine
- Répertoire personnel
- Nommage des fichiers dans l'arborescence :  
`/home/alice/perso/adresses`



# Représentation des répertoires



Répertoire *racine* **No 1**

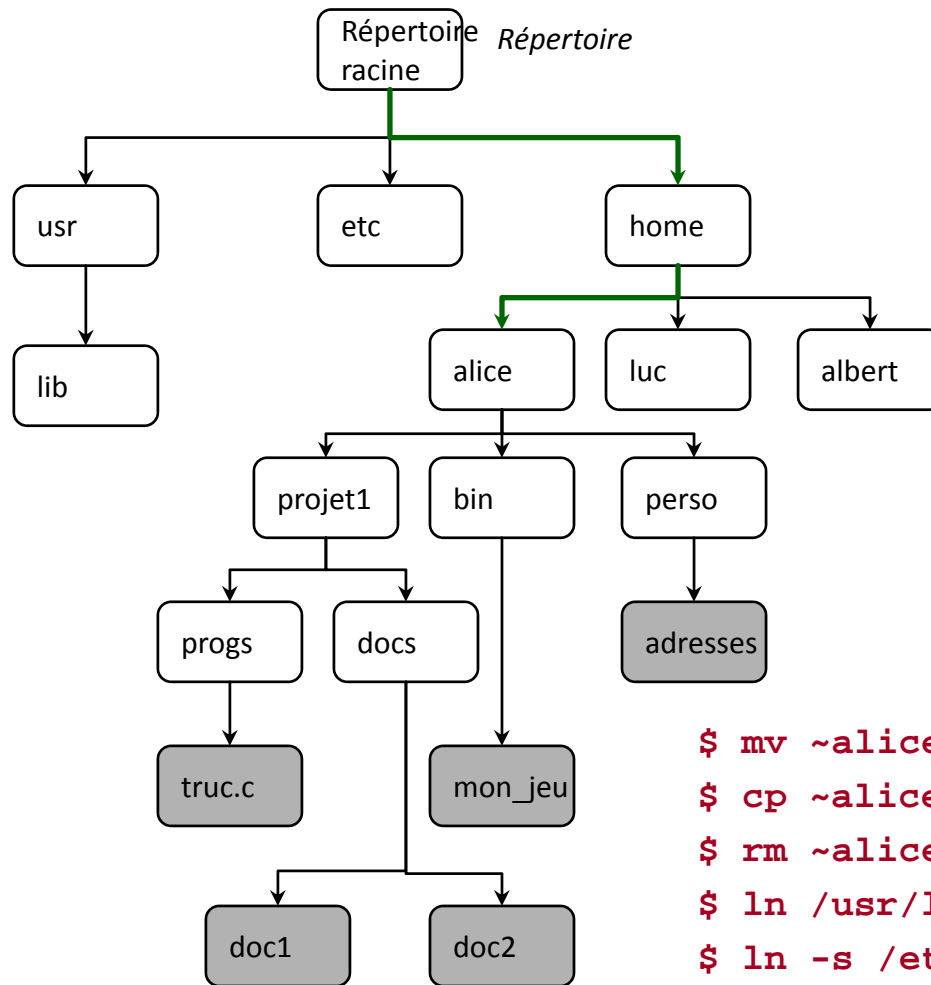
Nom du composant	No du descripteur
.	1
..	1
usr	11
etc	3
home	133

Répertoire *Home* **No 133**

Nom du composant	No du descripteur
.	133
..	1
alice	141
luc	399
albert	1933



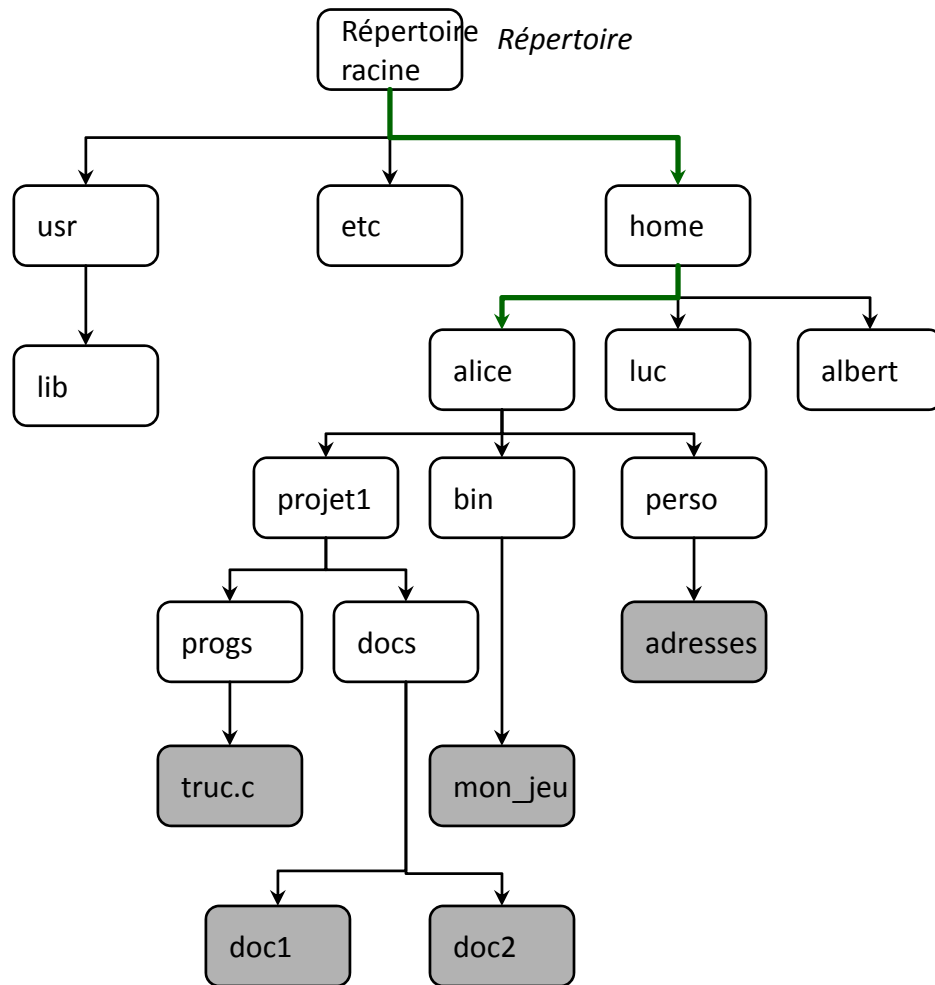
# Fichiers et répertoires



```
$ mv ~alice/perso/adresses ~alice/perso/mes_adrs  
$ cp ~alice/bin/mon_jeu ~alice/bin/my_game  
$ rm ~alice/bin/my_game  
$ ln /usr/lib ~alice/project1/lib  
$ ln -s /etc ~alice/etc
```



# Répertoire de travail (RT - WD)



- ❑ Pour chaque processus
  - RT maintenu par le SE
- ❑ Changement :  
`$ cd /home/alice`
- ❑ Affichage :  
`$ pwd`
- ❑ Manipulation :  
`$ cd projet1/docs`  
`$ more doc2`  
`$ rm ../progs/truc.c`



# Opérations sur les répertoires

---

- ❑ Création :

```
$ mkdir denis
```

- ❑ Destruction :

```
$ rmdir albert
```

- ❑ Copie :

```
$ cp -r ~alice/projet1 ~denis
```

- ❑ Affichage :

```
$ ls docs
```

```
$ ls
```

```
$ ls -a
```



# Répertoires importants

---

<b>/bin</b>	Commandes Unix essentielles
<b>/usr/bin</b>	Autres commandes
<b>/usr/sbin</b>	Commandes superutilisateur pour admin. syst.
<b>/sbin</b>	Commandes essentielles pour admin., config. et démarrage
<b>/boot</b>	Noyau et certains fichiers d'amorçage
<b>/etc</b>	Fichiers pour certains services de config. syst., mots de passe, réseau, ...
<b>/var</b>	Fichiers d'admin. (traces de fonctionnement)
<b>/var/spool</b>	Espace de stockage temporaire (ex. fichiers en cours d'impression)
<b>/usr/lib</b>	Bibliothèques standards
<b>/usr/src</b>	Sources nécessaires à la compilation du système



# Les fichiers spéciaux

---

- ❑ Accès sur les périphériques fournis par le SE pour les applications
- ❑ Lecture et écriture sur un périphérique
- ❑ Répertoire `/dev`  
`/dev/tty0`
- ❑ Pseudo-périphériques  
`/dev/null`



# Les processus

---

- ❑ Unix est multitâches
- ❑ Commandes de manipulation de processus
  - Processus
    - » séquentiels
    - » concurrents
- ❑ Identificateur de processus
  - *process identifier -- PID*



# Processus en avant-plan

---

- Avant-plan (*foreground*)
- Une tâche à la fois
- Contrôle du terminal (ou fenêtre)
- Accepte entrées
- Séquentiel :
  - `travail_A`
  - `travail_B`



# Processus en arrière-plan

---

□ Arrière-plan (*background*)

□ Plusieurs tâches à la fois

```
$ travail_A &
```

```
[1] 207
```

```
$ travail_B &
```

```
[2] 288
```

□ Affichage

```
$ jobs -l
```

```
[1]- 207 Running travail_A &
```

```
[2]+ 288 Running travail_B &
```



# Avant-plan / arrière-plan

---

- Arrière-plan → avant-plan

```
$ %1
```

```
$ fg %1
```

- Avant-plan → arrière-plan

```
$ travail_A
```

```
$ ^Z
```

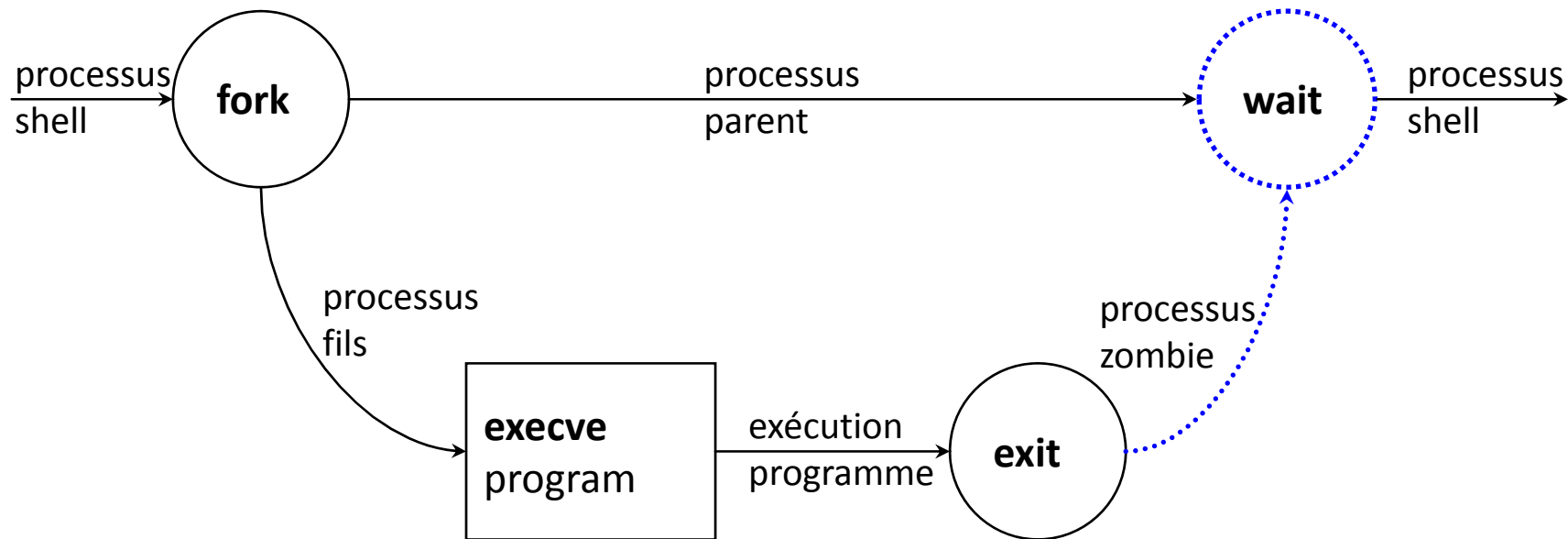
```
$ jobs
```

```
[1]+ 488 Stopped travail_A
```

```
$ bg %1
```



# Exécution de programme par le Shell





# Affichage des processus: `ps`

```
reveille@ubuntu: ~  
File Edit View Terminal Help  
reveille@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 1804 pts/0        00:00:00 bash  
 1823 pts/0        00:00:07 emacs  
 5413 pts/0        00:00:00 ps  
reveille@ubuntu:~$ █
```



# Communications et E/S

---

- Communication inter-processus
  - Les signaux
  - Les tubes
- Les Entrees/Sorties
  - Les redirections



# Les signaux

---

- ❑ Message d'un processus à un autre
- ❑ Traitement des conditions exceptionnelles
- ❑ Numéros et noms de signaux
- ❑ Liste des signaux

```
$ kill -1
```



# Signaux et touches de contrôle

---

- ❑ Touche de contrôle pour l'envoi de signaux

- STOP    **^Z**

- INT     **^C**

- QUIT    **^\**

- ❑ Configuration des touches de contrôle

- \$ stty nom-du-signal caractère**

- \$ stty -a**

- ❑ Certains signaux pas de touches de contrôle



# Envoi de signaux avec `kill`

- ❑ Envoi de signaux à nos processus
- ❑ Signaux pour terminaison du processus (généralement)

```
$ kill %1
```

```
$ kill -QUIT %1
```

```
[1]+ Exit 131 travail
```

```
$ kill -KILL %1
```

```
[1]+ Killedtravail
```

- ❑ Processus peut *catcher* un signal et effectuer certaines actions (par ex. éditeur de texte)



# Entrées/sorties standards

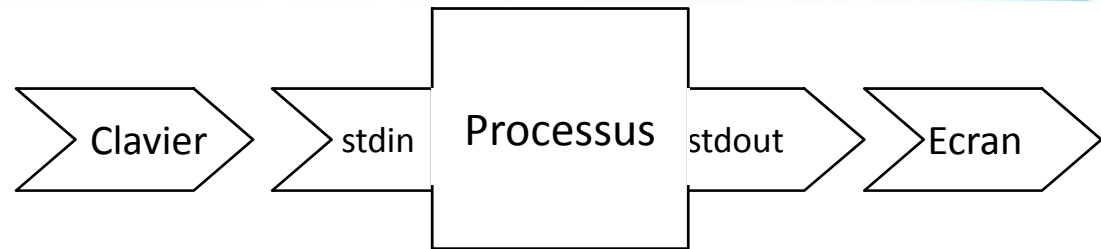
---

- ❑ Processus (généralement)
  - Flot de données en entrée (entrée standard)
  - Traitement
  - Flot de données en sortie (sortie standard)
- ❑ Fichiers initialement ouverts pour un processus
  - Entrée standard
  - Sortie standard

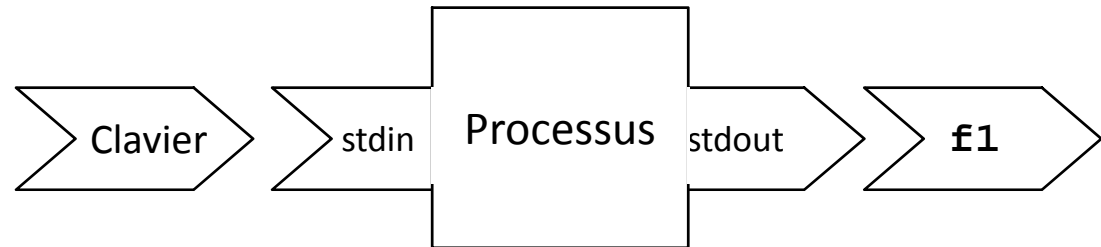


# Redirection des entrées/sorties

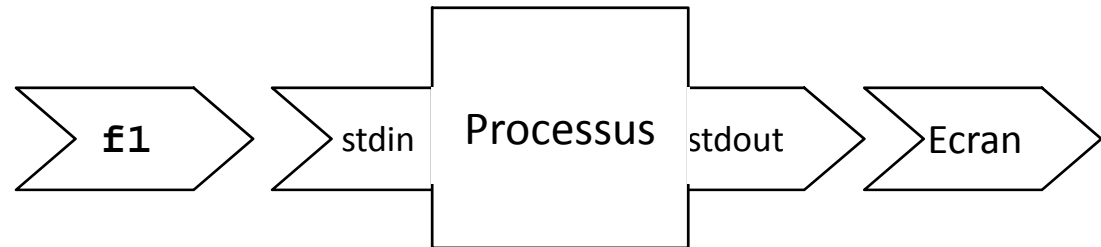
`travail`



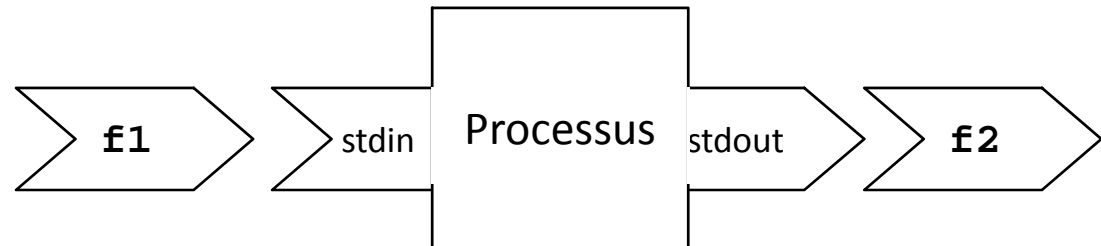
`travail > f1`



`travail < f1`



`travail < f1 > f2`





# Redirection des entrées/sorties

---

## □ Arguments vs E/S

```
trelba:9XF0:301:277:albert trelba:/home/trelba:/bin/bash
```

```
$ wc -l /etc/passwd
```

```
$ wc -l < /etc/passwd
```

```
$ echo truc > ff
```

```
$ grep bash /etc/passwd
```

## □ Résultats intermédiaires

```
$ cut -d: -f1 < /etc/passwd > fa
```

```
$ sort < fa > fb
```

```
$ lpr < fb
```

```
$ rm fa fb
```



# Les tubes

---

- ❑ Tubes (*pipes -- pipeline*)
- ❑ Rediriger la sortie d'un processus sur l'entrée d'un autre
- ❑ Multiple redirections

```
$ cut -d: -f1 < /etc/passwd | sort | lpr
```



# Sécurité

---

## □ Authentification

- Qui est-on ?
- Peut-on le prouver ?

## □ Autorisation

- Alice peut lire le fichier  $f$  ; Albert ne peut pas
- Roger peut exécuter le programme  $p$  ; Alice ne peut pas
- Luc peut lire et écrire le fichier  $doc$  ; Albert et Alice peuvent seulement le lire



# Authentification

---

- Au démarrage

*Qui est-on ?*

**Login:** luc

*Peut-on le prouver ?*

**Password:** \*\*\*\*

- A la fin

**\$ logout**



# Authentification

---

- ❑ Fichier des mots de passe

  - `/etc/passwd`

```
trebla:9XF0:301:277:albert trebla:/home/trebla:/bin/bash
```

- ❑ Changer de Shell

  - `$ echo $SHELL`

  - `$ chsh`

- ❑ Changer de mot de passe

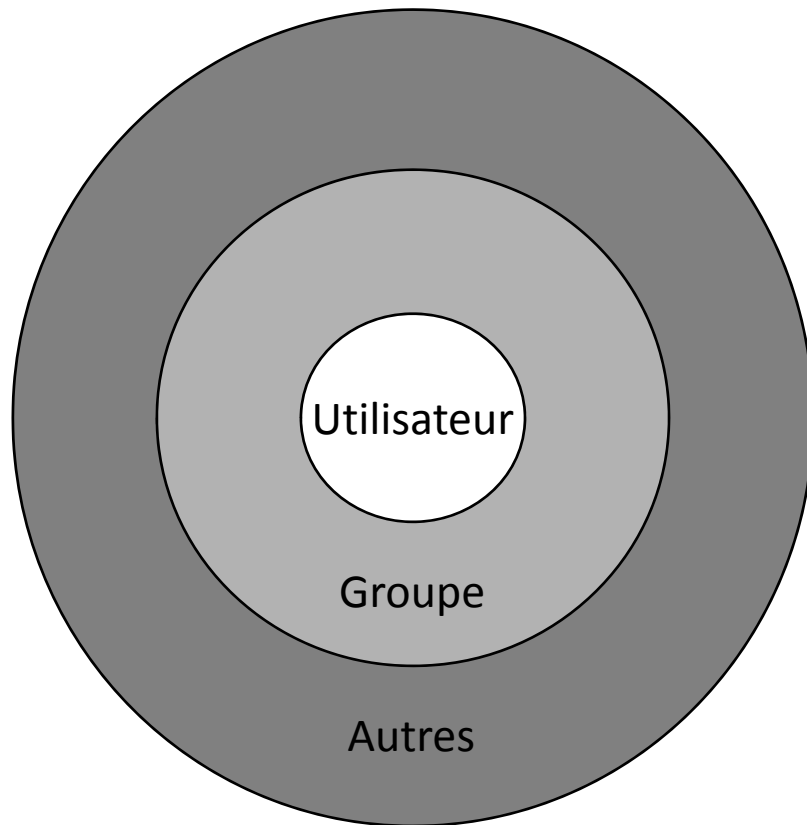
  - `$ passwd`

- ❑ Changer d'identité

  - `$ login`



# Autorisation : abstraction fichier



- Abstraction fichier
  - Simple
  - Répertoire
  - Périphérique
- Niveaux
  - Utilisateur
  - Groupe
  - Autres
- Protections
  - Lecture
  - Ecriture
  - Exécution



# Autorisation : fichier simple

---

## Lecture

- permission de consulter le contenu

## Ecriture

- permission de modifier le contenu

## Exécution

- permission d'exécuter le fichier en tant que programme



# Autorisation: répertoire

---

## Lecture

- permission de consulter le contenu d'un répertoire

## Ecriture

- permission d'ajouter ou de supprimer des fichiers dans le répertoire

## Exécution

- permission de se déplacer dans ce répertoire



# Autorisation: les niveaux

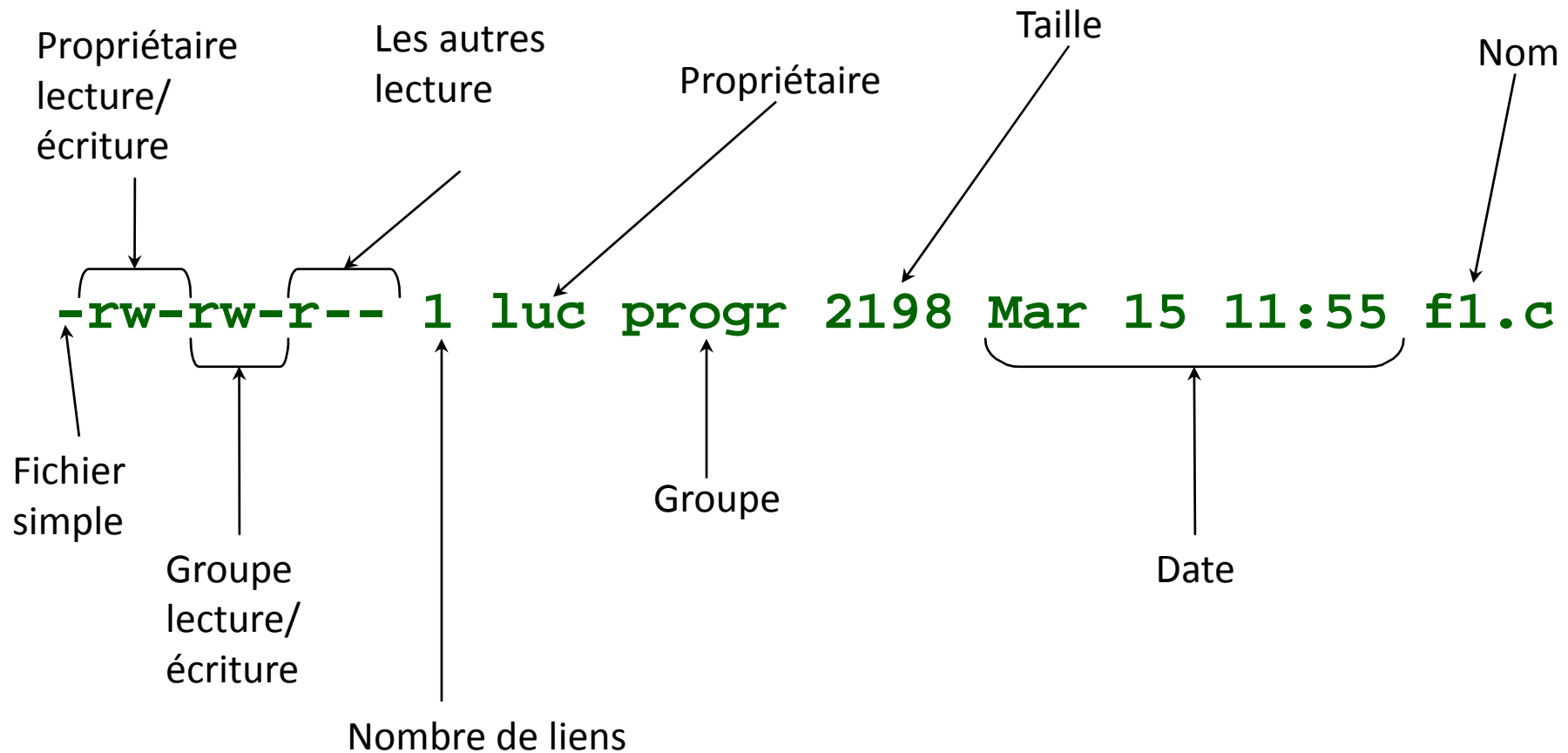
---

- Utilisateur
  - celui qui crée le fichier (généralement)
- Groupe
  - un ensemble d'utilisateurs (par ex. une équipe de programmeurs)
- Autres
  - les utilisateurs qui ne sont ni l'utilisateur ni membres du groupe



# Exemple : fichier simple

```
ls -l
```





# Exemple : executable

Propriétaire  
lecture/  
écriture/  
exécution

Les autres  
lecture/  
exécution

`-rwxr-xr-x 1 luc progr 316 Mar 15 14:55 truc`

Fichier simple

Groupe lecture/  
exécution



# Exemple : répertoire

Propriétaire  
lecture/  
écriture/  
exécution

Les autres  
lecture/  
exécution

```
drwxr-xr-x 3 luc progr 316 Mar 15 14:55 proj
```

Répertoire

Groupe  
lecture/  
exécution



# Changement des droits d'accès

---

□ **chmod MODE fichier**

➤ **MODE**

» ugoa

» +/-

» rwx

□ **chown propriétaire fichier**

□ **chgrp groupe fichier**

(Exemple: installation d'un programme dans **bin** en tant que superutilisateur)



# Autorisation : processus

---

- ❑ Communication par envoi de signaux (**kill**)
  - Protection : propriétaire et superutilisateur
- ❑ Utilisation de l'UC par définition de priorité (**nice**)
  - **nice -n commande [args]**
    - » n [-20, 19]
  - **renice**



# Informations de maintenance

---

- ❑ Fichiers

```
$ ls -i
```

- ❑ Fichiers

```
$ time date
```

- ❑ Utilisateur

```
$ whoami
```

- ❑ Machine

```
$ hostname
```