



CHAPITRE 1

L'ENVIRONNEMENT UNIX/LINUX (PART 2: SYSTÈME DE FICHER)

Docteur Ingénieur : Fatma SOMAA

AU: 2023-2024

COMMANDE

- Une commande Unix est un mot ou une phrase = syntaxe bien particulière entrée dans un interpréteur de commandes et donnant l'ordre d'actions à exécuter par l'ordinateur.
- Une commande UNIX se décompose en 3 parties :
 - la commande elle-même (c'est toujours le premier élément sur la ligne),
 - des options commençant par le symbole moins « - » qui, comme le nom l'indique, sont optionnelles (zéro, une ou plusieurs options),
 - des arguments : zéro, un nombre fixe ou variable d'arguments

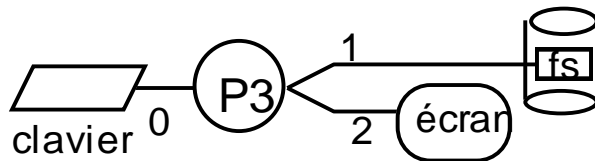
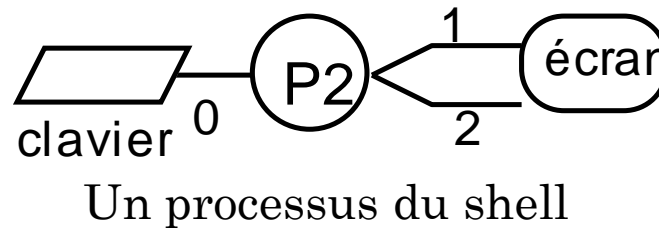
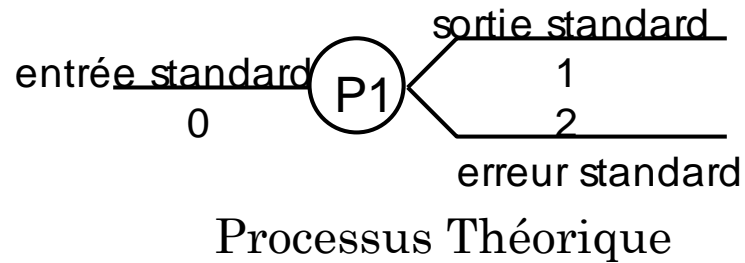
\$ commande -option1 -option2 -option3 argument1 argument2

MANUEL DES COMMANDES

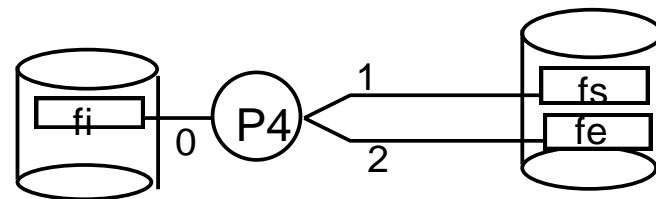
- Pour tout connaître sur une commande, une page de manuel est fournie avec le système. Ces pages sont souvent disponibles dans plusieurs langues, mais la page en anglais est toujours la référence ;

\$ man commande

FICHIERS STANDARDS



sortie standard redirigé



les entrées-sorties
standards redirigées

FICHIERS STANDARDS

Différentes formes de redirection sont possibles:

- `1>f` ou `>f` redirection de la sortie standard sur `f`, avec création;
- `2>f` idem, pour rediriger l'erreur standard;
- `1>>f` ou `>>f` redirection de la sortie standard sur `f`, avec création de `f` s'il n'existait pas;
- `2>>f` idem, pour rediriger l'erreur standard;
- `0<f` ou `<f` redirection de l'entrée standard à partir de `f`;

affiche un message sur la sortie standard

```
$ echo ceci est mon message
```

le message sert à initialiser le fichier `f`.

```
$ echo ceci est mon message >f
```

*écrit la liste des fichiers dans **fichier.liste***

```
$ ls >fichier.liste
```

```
$ cat f1 f2 f3 >>temp
```

*ajouter `f1 f2 f3` dans **temp***

COMMANDE LS

- La commande `ls` (abréviation du mot anglais `list`) vous permet de visualiser le contenu du système de fichiers. Elle affiche les fichiers et le nom des sous-répertoires qui se trouvent dans un répertoire.

COMPOSITION DE PROCESSUS

- composition séquentielle (;): **$C1;C2;C3$**

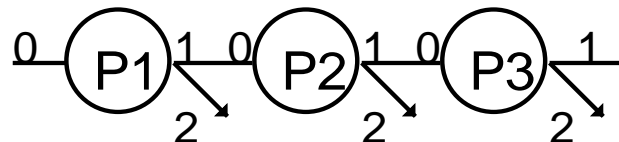
```
$ date; wc <fff; who am i; echo fin de commande>toto
Mon Sep Jan 27 19:08:47 PST 1993
12 33 345
jmr tty9 Jan 27 17:55
$ cat toto
fin de commande composee
```

- composition parallèle de processus indépendants (&):
 $C1&C2&C3$

le commande wc s'exécutera pendant que l'utilisateur à la main

```
$ wc ch* >wc.out &
2145
$ lp ch*& cc code.c
```

- composition parallèle de processus communicants(|):
 $C1|C2|C3$



FICHER

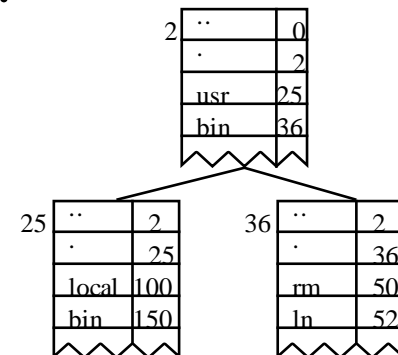
- Un **fichier** est une suite non structurée, stockée sur une mémoire auxiliaire.
- Plusieurs sortes de fichiers:
 - les fichiers ordinaires;
 - les répertoires;
 - les périphériques à accès par caractère (terminal);
 - les périphériques à accès par bloc (disque);
 - les références vers un autre volume (liens symboliques);
 - les tubes mettant en communication deux processus ;
 - les prises réseaux, pour accéder à des fichiers éloignés.
- Un fichier peut avoir plusieurs liens (noms)

CARACTÉRISTIQUES D'UN FICHIER

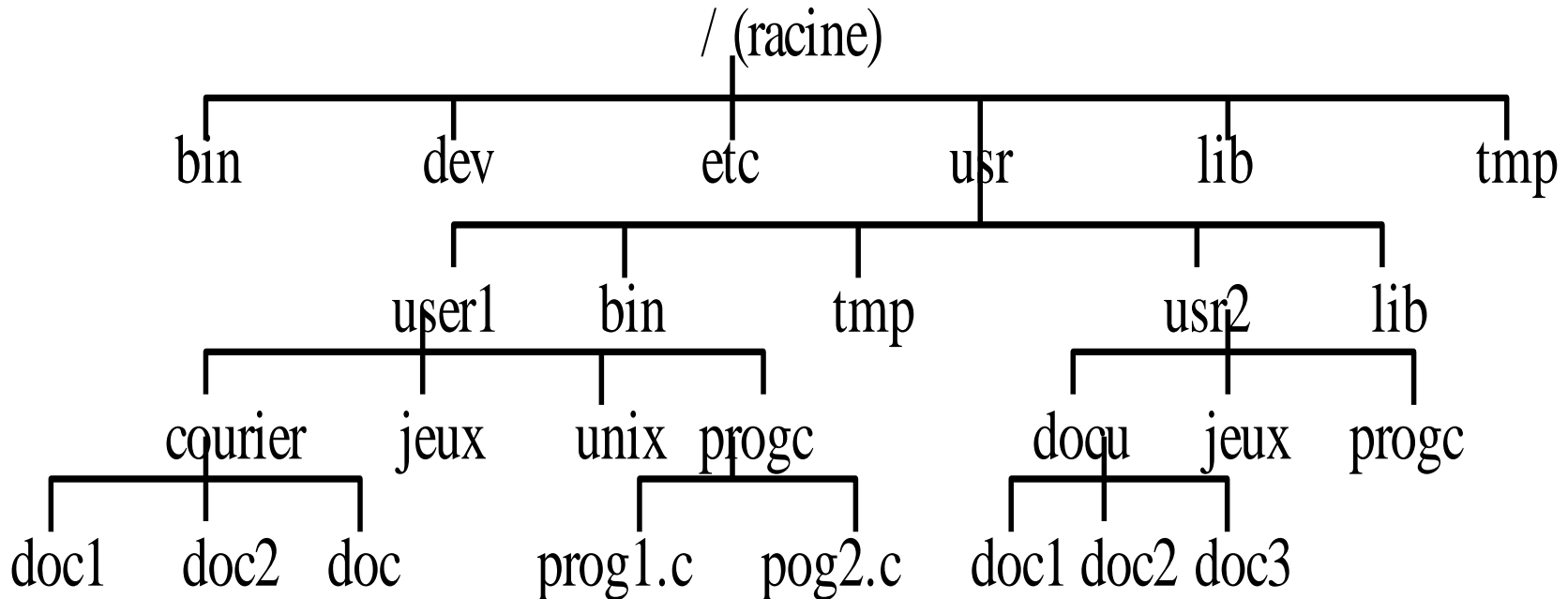
- Les caractéristiques statiques d'un fichier sont stockées dans un descripteur appelé *i-noeud*:
 - le type de fichier (ordinaire, répertoire, ...);
 - sa taille, en octets;
 - le couple UID-GID; le numéro du propriétaire et le numéro de son groupe;
 - date de création, la date de dernière modification et la date de dernière consultation;
 - un compteur de référence sur le i-noeud (nombre de liens);
 - la liste des blocs contenant l'information sur le disque.
 - ...

DÉSIGNATION D'UN FICHIER : RÉPERTOIRES

- La correspondance entre désignation externe (identificateur) et désignation interne (n° de i-noeud) est réalisée par les répertoires .
- Un *répertoire* (catalogue) est un fichier particulier : c'est en fait un tableau à deux colonnes, l'une noms des fichiers, numéro de i-noeud.



RÉPERTOIRE



Le répertoire racine (/) : on parle alors de chemin absolu

Le répertoire courant : on parle alors de chemin relatif

Le chemin complet du fichier **prog1.c** est :

/usr/user1/progc/prog1.c

le chemin relatif à **doc** : **courrier/doc**

LES PERMISSIONS

- Chaque fichiers possède un ensemble de permissions, qui déterminent qui peut faire quoi avec le fichier.
- 3 types de permissions associées à un fichier :
 - **lecture** examen du contenu;
 - **écriture** modification;
 - **exécution** faire exécuter comme un programme;
- L'option **-l** de **ls** imprime les permissions :

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root 5115 Aug 30 10 10:40 /etc/passwd
$ ls -lg /etc/passwd
-rw-r--r-- 1 root 5115 Aug 30 10 10:40 /etc/passwd
$ ls -l /bin/passwd
-rwsr-xr-x 1 root 5115 Aug 30 10 10:40 /etc/passwd
```
- Le **s** à la place du **x** dans les permissions du propriétaire indique que, lorsque la commande est exécutée, elle doit obtenir les permissions associées à son propriétaire (**set-uid**).

Question(s)