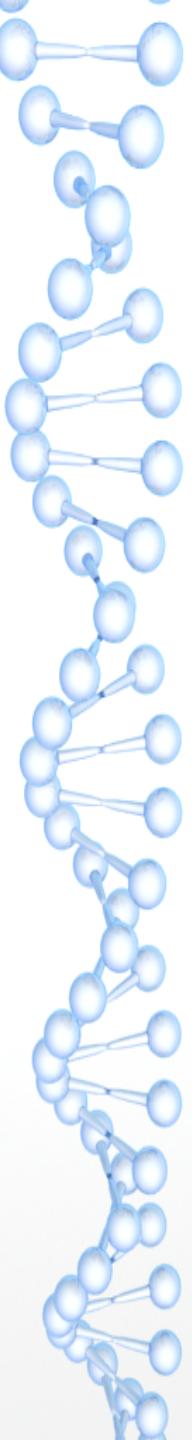


GNU/Linux

## Les processus



# Sommaire

## I. Les processus

Définition

États d'un processus

Gestion des processus et commandes de base

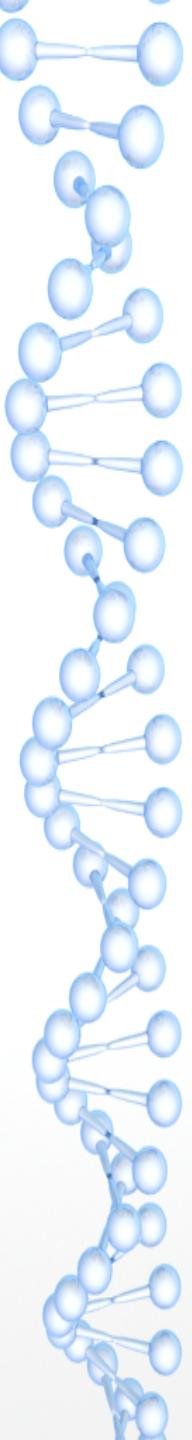
Les redirection I/O

Les tubes

## II. Les filtres

Définition

Exemples



# Les processus

## Définition

Un processus est la représentation en mémoire d'un programme qui est exécuté par un processeur . Il regroupe toutes les informations nécessaires à son contrôle (code machine, état des registres du processeur, mémoire allouée, descripteurs de fichiers ouverts, etc.). Un même programme peut faire l'objet de plusieurs processus.

- Un processus (processus fils) est toujours créé par un autre processus (processus père) sauf le 1<sup>er</sup> processus appelé systemd
- 
- 

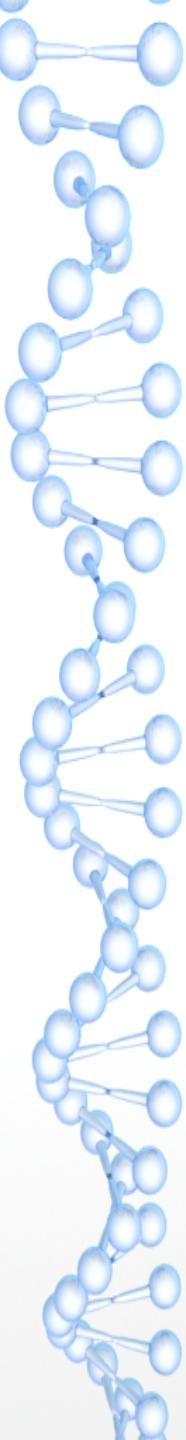
### Principales caractéristiques d'un processus:

- Un identifiant PID
- Un identifiant du processus parent (PPID)
- Son propriétaire
- Sa priorité

...

quelques commandes Pour afficher les processus en cours d'exécution : **ps, top**

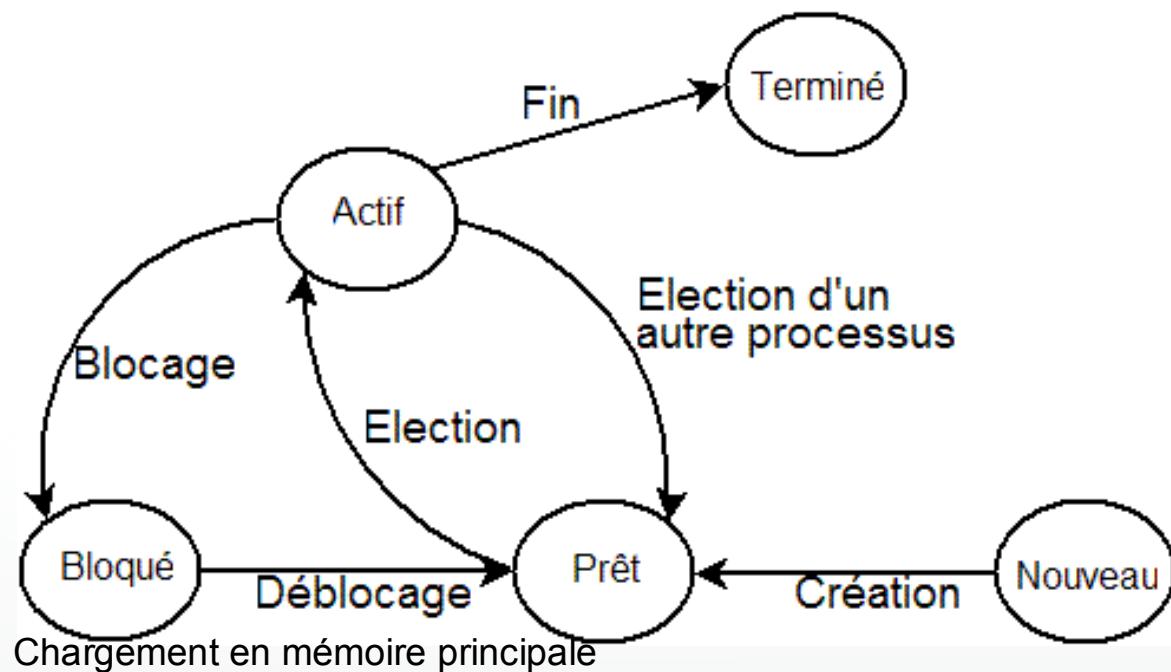
Pour afficher la hiérarchie des processus : **pstree**

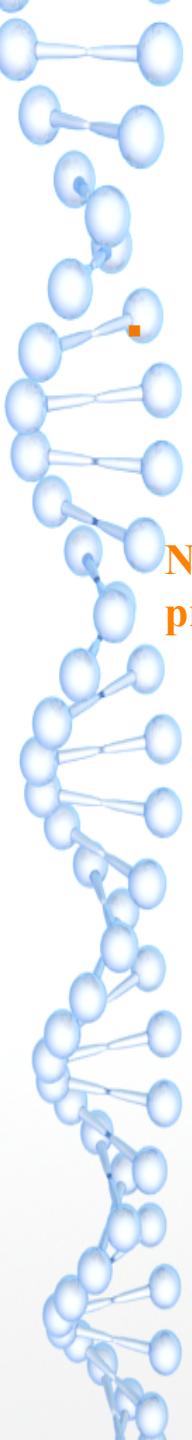


# Les processus

## États d'un processus

Depuis sa création jusqu'à sa destruction (Fin), chaque processus va transiter par un certain nombre d'états.





# Les processus

Quelques informations renvoyées par la commande `ps`:

Numéro de processus

PID  
3899  
4180

TTY pts/1  
pts/1

STAT R  
S

Terminal associé

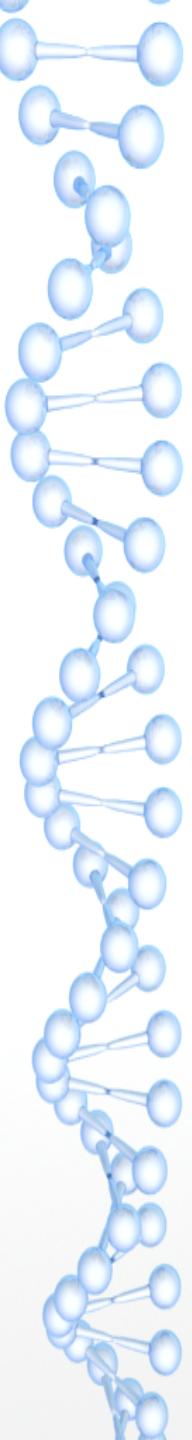
Temps CPU utilisé  
TIME 0:00.08 0:00.04  
CMD bash ps

Commande exécutée

État du processus:

R	Actif
T	Bloqué
S	Endormi
IW	Swappé
Z	Zombie

La commande `ps` sans option affiche seulement les processus attachés au terminal exécuté.



# Les processus

## Gestion des processus et commandes de base

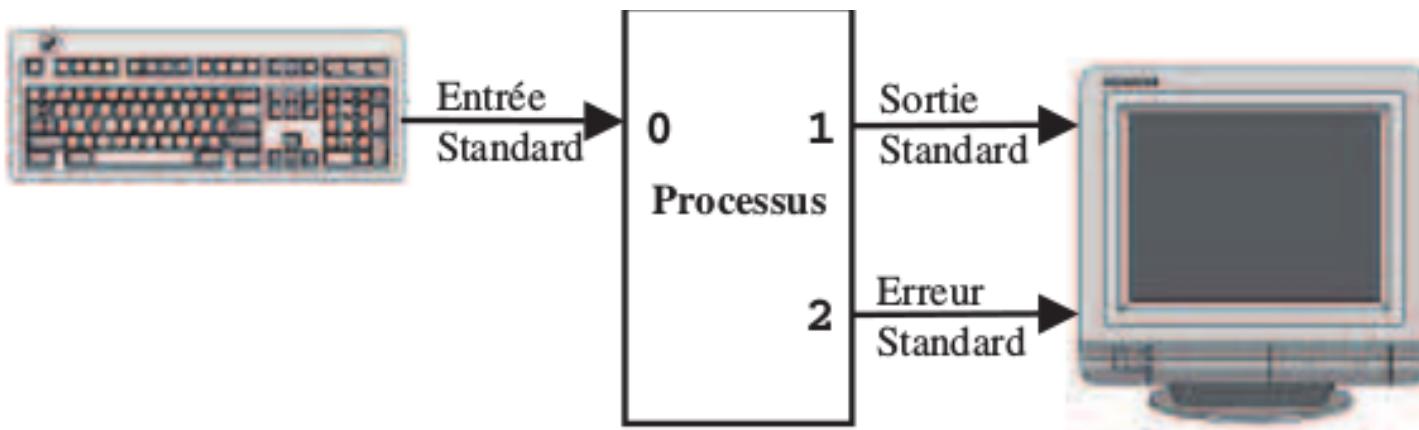
- Pour afficher tous les processus en cours d'exécution on execute : **ps aux**  
Options :
  - a      Liste tous les processus actifs
  - u      Format d'affichage long
  - x      Inclut les processus sans terminal
- Pour Terminer (tuer) un processus: **kill -9 <PID>**
- Exécuter un processus en arrière-plan: **commande &** (le terminal n'est pas bloqué)

# Les processus

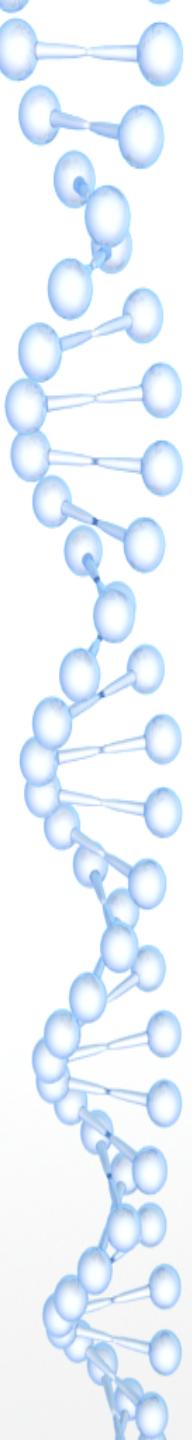
## Les redirection I/O

### Le principe de redirection

On appelle processus, ou tâche, l'exécution d'un programme exécutable. Au lancement de chaque processus, l'interpréteur de commandes ouvre d'office une entrée standard (par défaut le clavier), une sortie standard (par défaut l'écran) et la sortie d'erreur standard (par défaut l'écran)



Ces entrées-sorties standard peuvent être redirigées vers un fichier, un tube, un périphérique. La redirection de la sortie standard consiste à renvoyer le texte qui apparaît à l'écran vers un fichier. Aucune information n'apparaîtra à l'écran, hormis celles qui transitent par la sortie d'erreur standard.



# Les processus

## Les redirection I/O

Symbol	Signification
<	Redirige l'entrée standard
<<	Concatène l'entrée standard
>	Redirige la sortie standard
>>	Concatène la sortie standard
2>	Redirige la sortie d'erreur

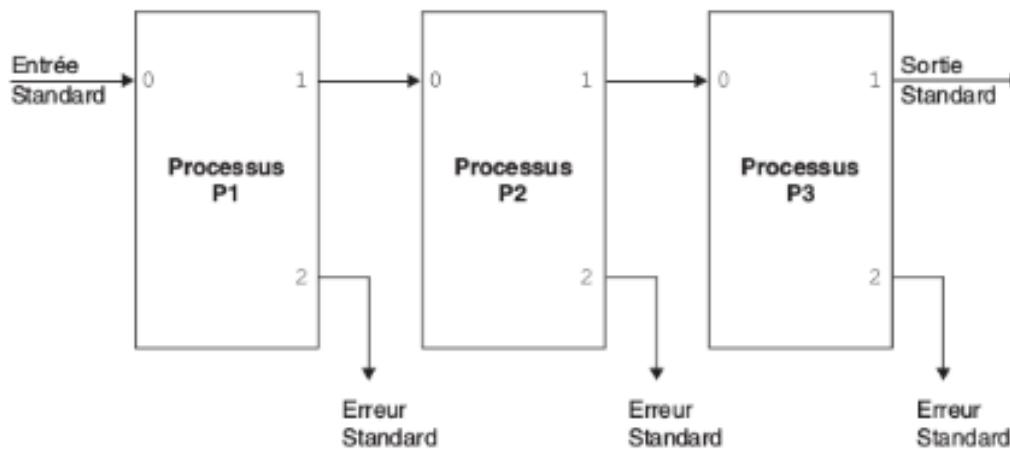
Exemples:

```
ls /home > liste Le résultat de la commande ls est enregistré dans le fichier liste  
date >> liste Ajoute à la fin du fichier liste la sortie de la commande date  
wc -l < liste Envoie comme entrée à la commande wc -l le fichier liste
```

# Les processus

Les tubes: | (AltGr6)

Un tube (`p 1 p *` en anglais) est un flot de données qui permet de relier la sortie standard d'une commande à l'entrée standard d'une autre commande sans passer par un fichier temporaire



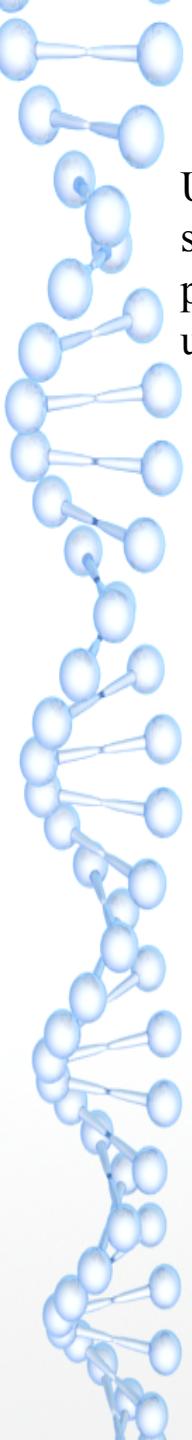
Exemple: Compter le nombre de fichiers dans le répertoire courant.

Sans le pipe:

```
ls > tmp ; wc -l < tmp
```

avec un pipe:

```
ls | wc -l
```



# Les filtres

Un filtre est une commande qui lit les données sur l'entrée standard, les traite et les écrit sur la sortie standard. Le concept tube ou pipe avec sa simplicité devient un outil très puissant dans GNU/Linux et propose un choix très vaste de filtres. Les filtres les plus utilisés sont les suivant :

**cat** Affiche le contenu des fichiers passés en argument (par défaut, stdin)

**more** Affiche page par page les fichier passés en argument (par défaut, stdin)

**tee** Recopie l'entrée standard sur la sortie standard et dans le fichier passé en argument.

**cut** Sélectionne uniquement certaines colonnes du fichier passé en paramètre

**grep** Recherche dans le fichier passé en argument les lignes vérifiant une expression régulière donnée.

**find** Permet de retrouver des fichiers à partir de certains nombres de critères

Et encore plein d'autres...

`which, locate, tail, head, wc, tr ...`

*Processus maîtrisés, ressources sécurisées..*

# Linux Process Control

- > Stop
- > End
- > Kill

