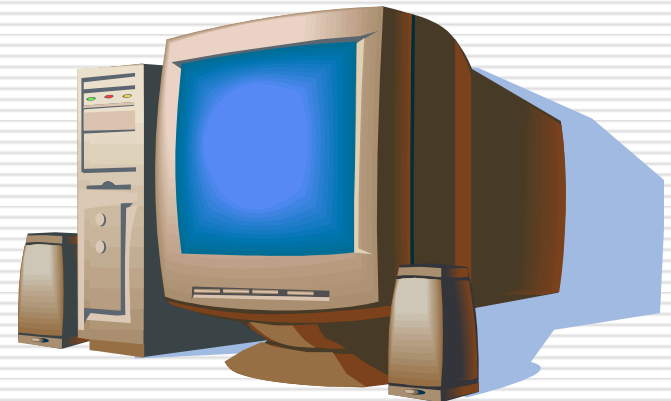
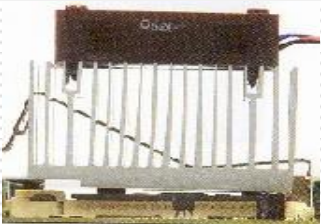
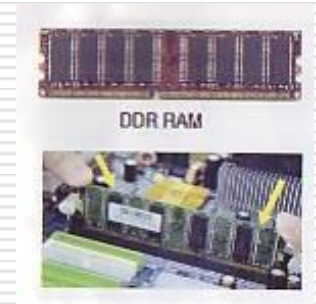
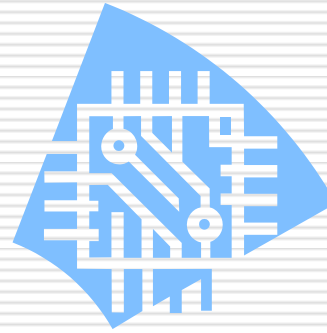
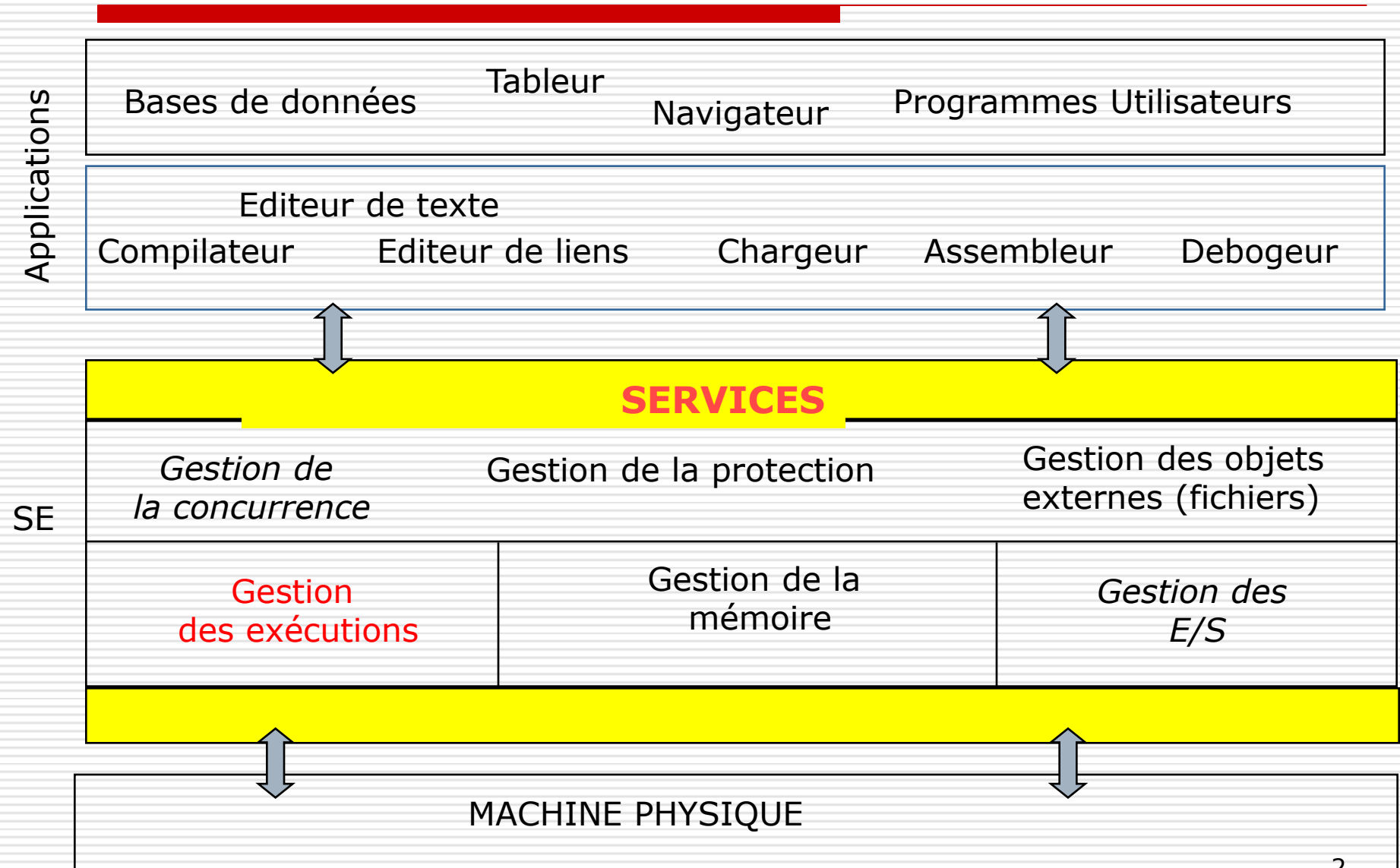


LES PROCESSUS

Structure et commandes



FONCTIONS D'UN SYSTEME D'EXPLOITATION



FONCTION D'EXECUTION

La gestion des exécution de programmes



La gestion des exécutions de programmes : processus et ordonnancement

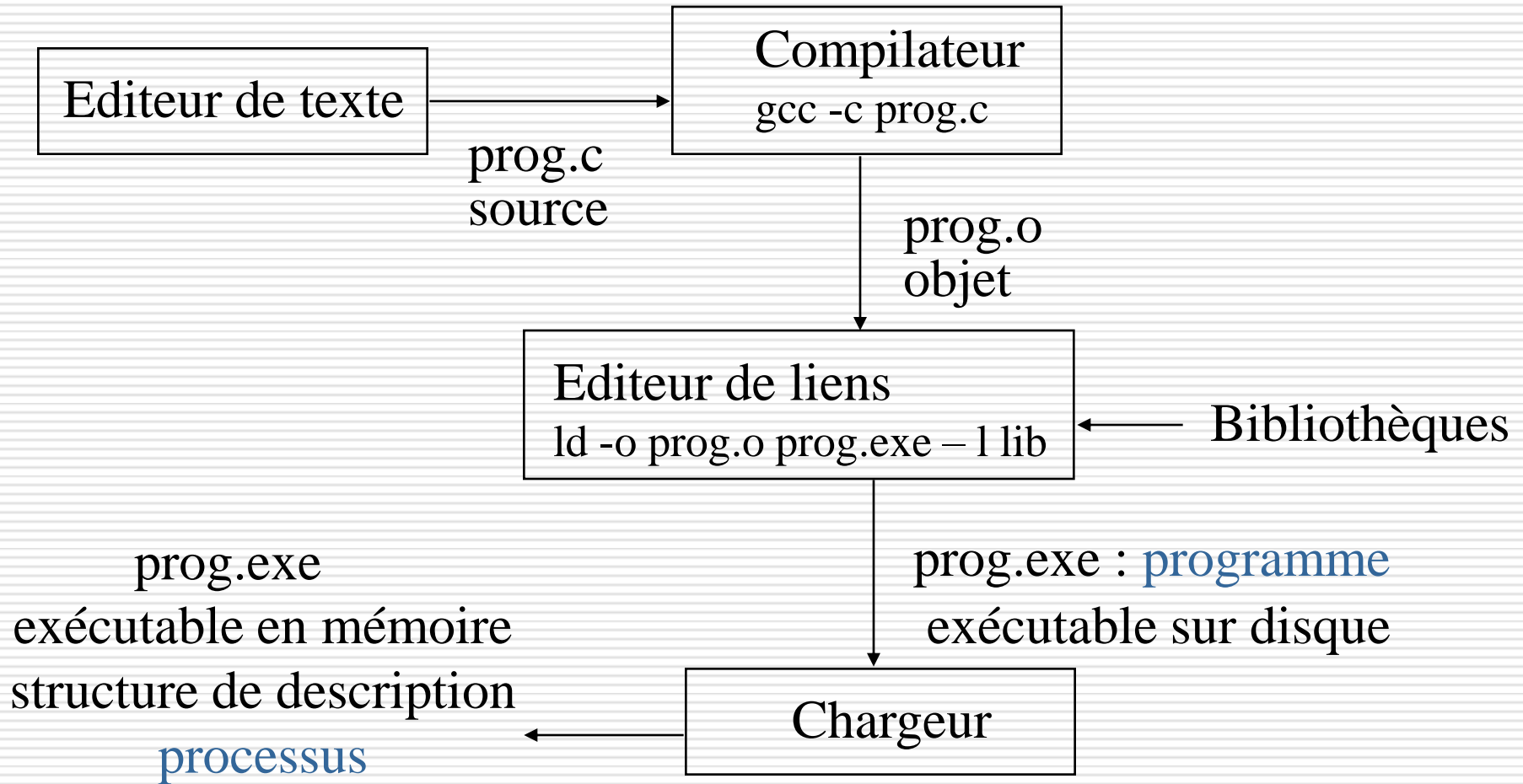
- Notion de processus : un processus est une exécution de programme
- Ordonnancement : planification des exécutions

Notion de processus

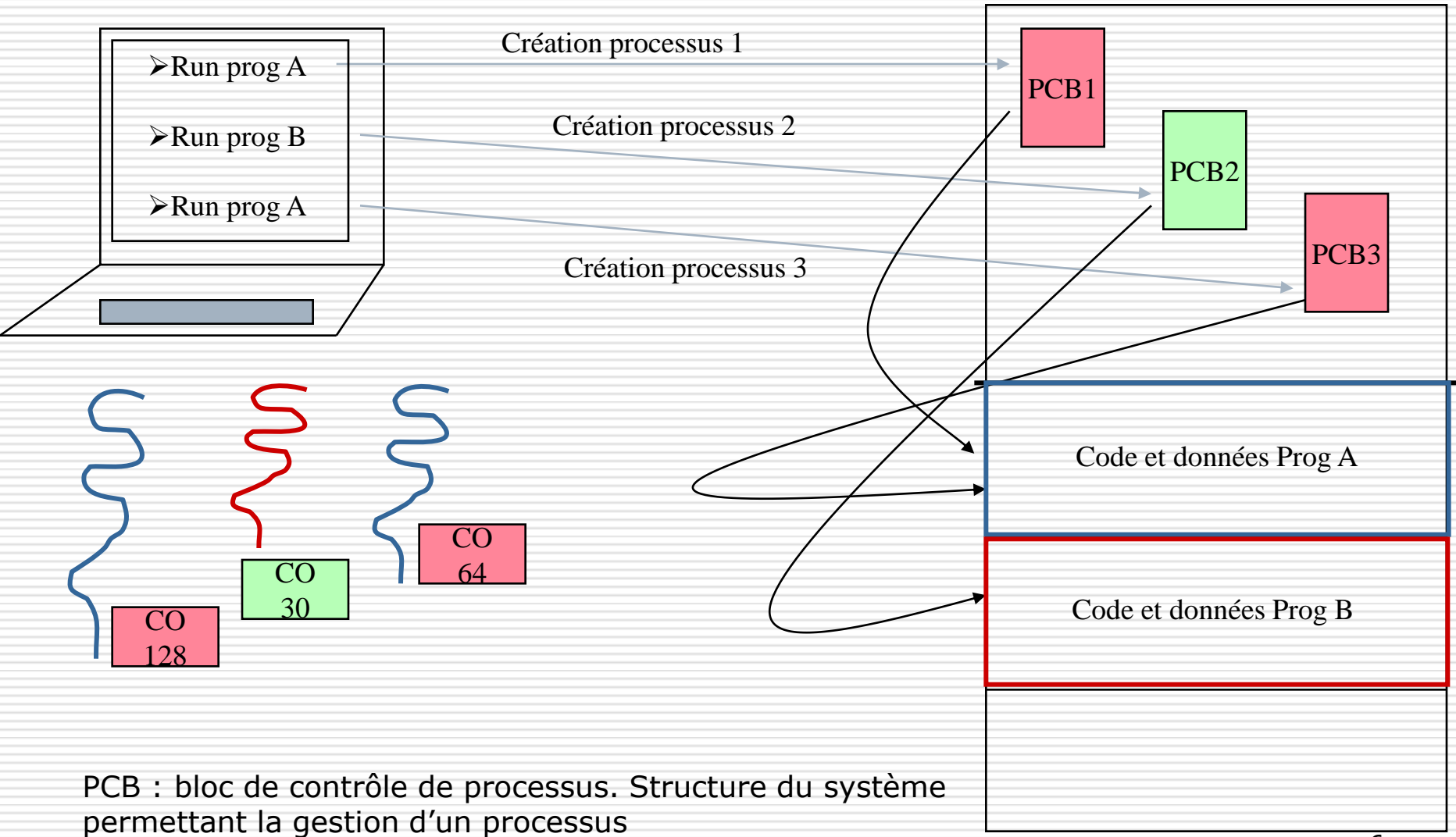
□ Définitions

- Un processus est un programme en cours d'exécution auquel est associé un environnement processeur (CO, PSW, RSP, registres généraux) et un environnement mémoire appelés contexte du processus.
- Un processus est l'instance dynamique d'un programme et incarne le fil d'exécution de celui-ci.

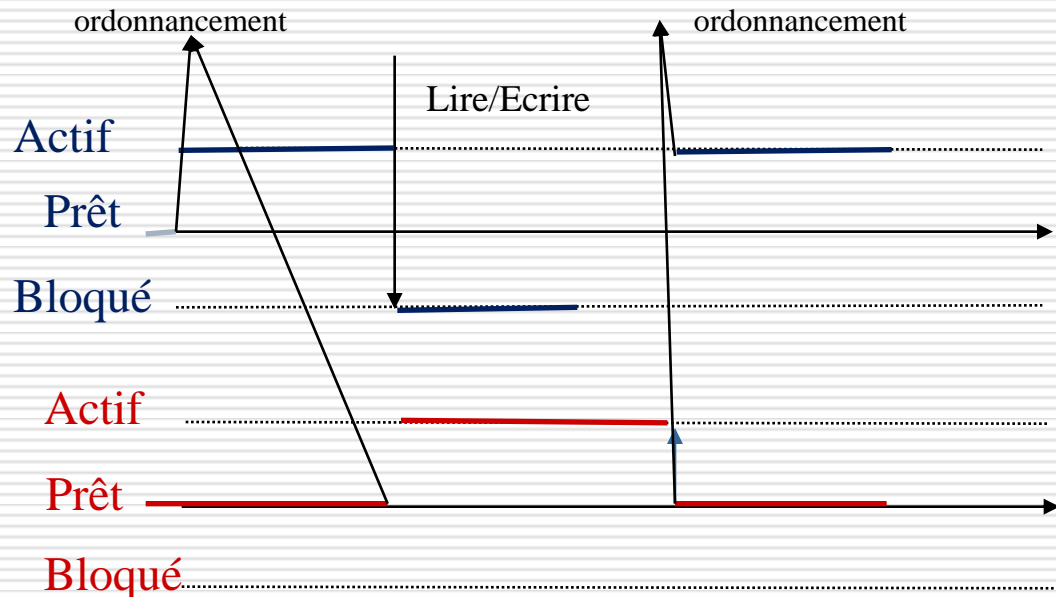
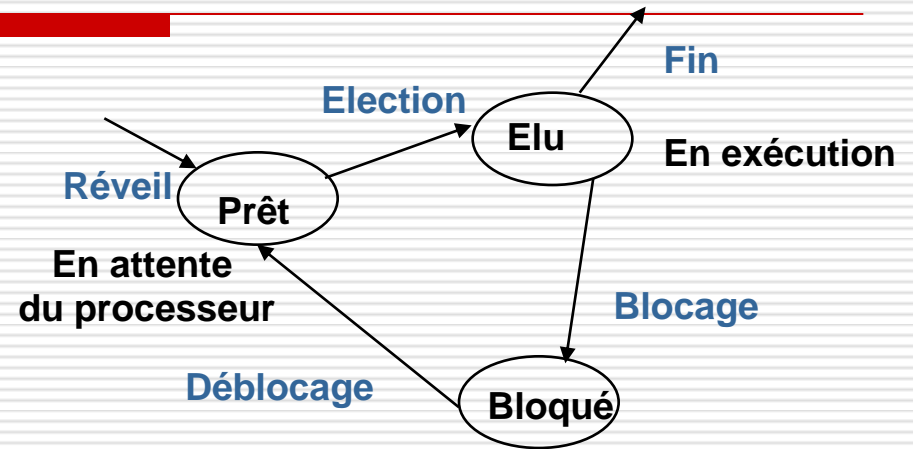
Du programme au processus



Il est important de distinguer la différence entre le programme exécutable et le processus.



Une ressource désigne toute entité (matérielle, logicielle) dont a besoin un processus pour s'exécuter (processeur, variable)



FONCTION D'EXECUTION

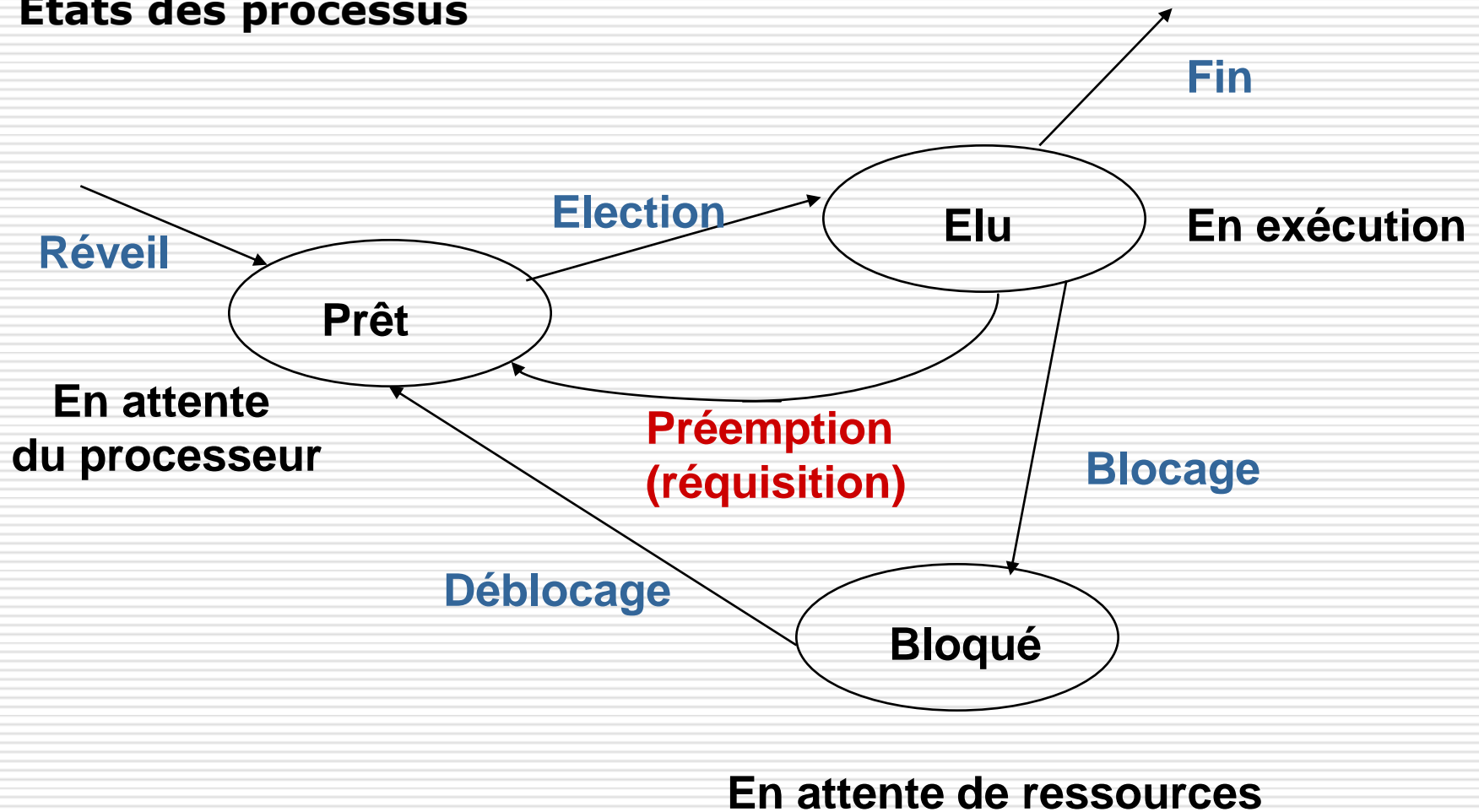
La gestion des exécution de programmes



La gestion des exécutions de programmes : processus et ordonnancement

- Notion de processus
- Ordonnancement : planification des exécutions de processus

Etats des processus



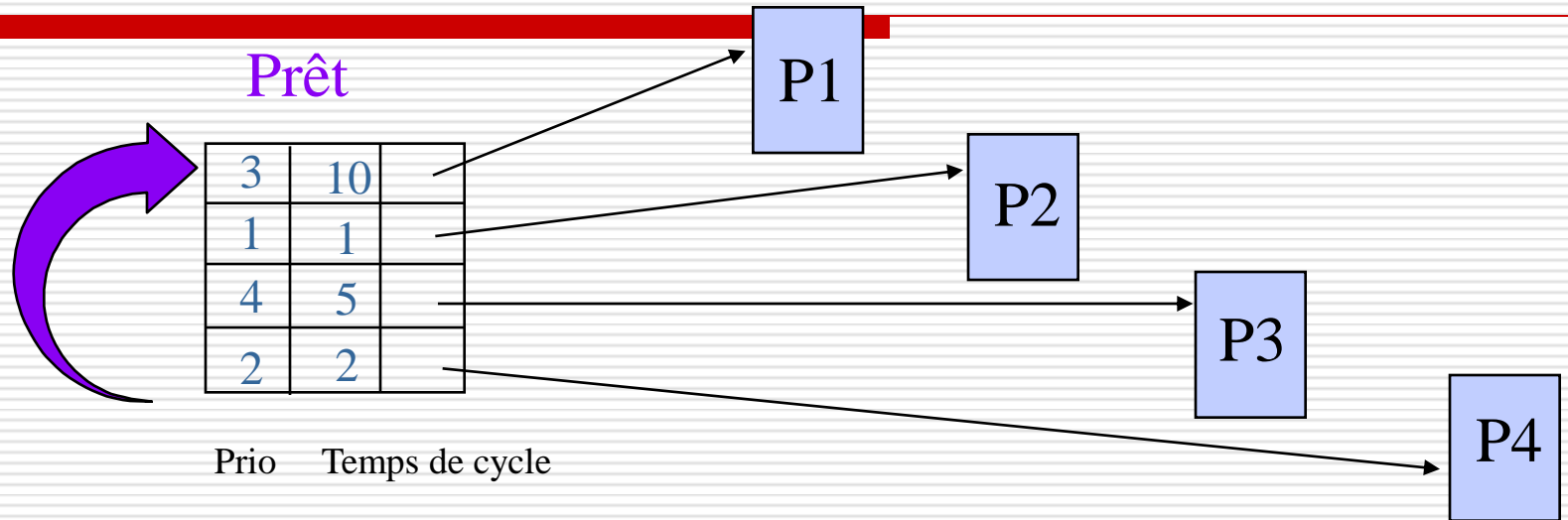
□ Par priorités constantes

- chaque processus reçoit une priorité
- le processus de plus forte priorité est élu

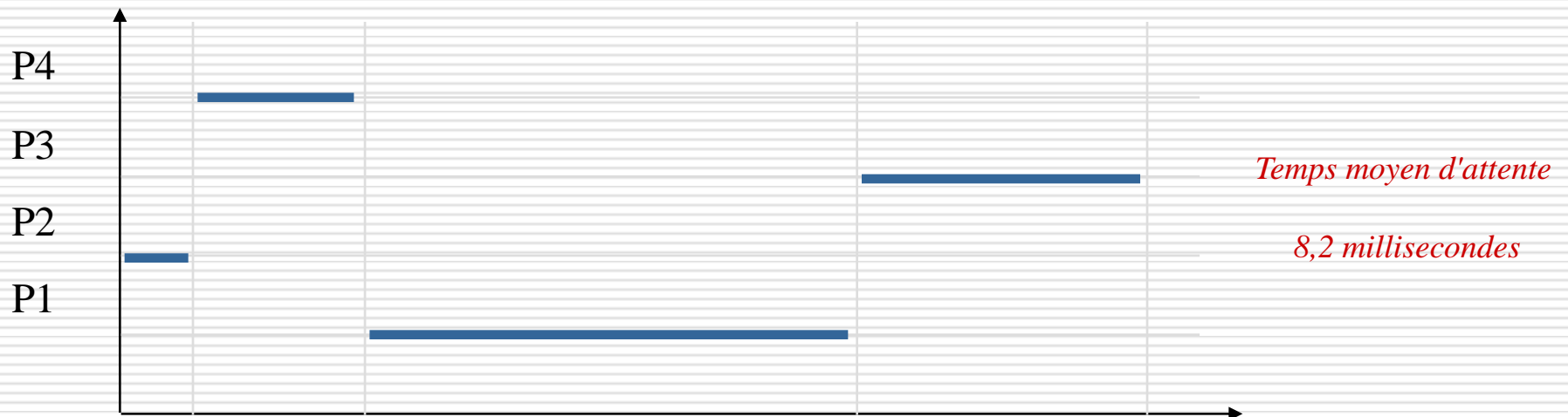
□ Par tourniquet (round robin)

- Définition d'un quantum = tranche de temps
- Un processus élu s'exécute au plus durant un quantum; à la fin du quantum, préemption et réinsertion en fin de file d'attente des processus prêts

Algorithme : avec priorités



↳ Priorité : le plus petite valeur correspond à la plus forte priorité



□ Par priorités constantes

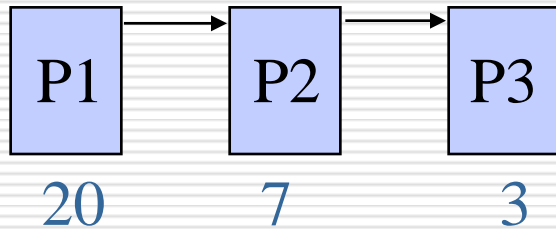
- chaque processus reçoit une priorité
- le processus de plus forte priorité est élu

□ Par tourniquet (round robin)

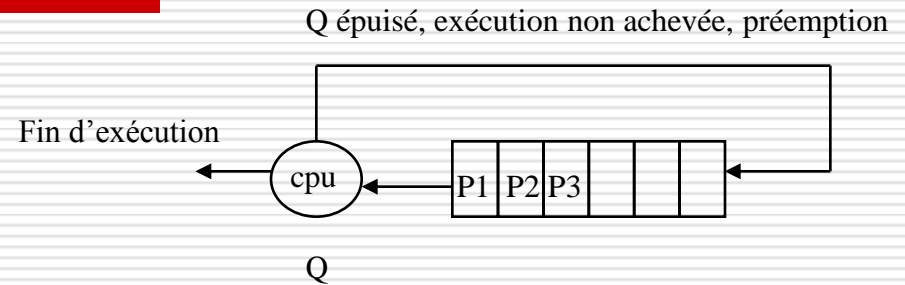
- Définition d'un quantum = tranche de temps
- Un processus élu s'exécute au plus durant un quantum; à la fin du quantum, préemption et réinsertion en fin de file d'attente des processus prêts

Algorithme : tourniquet

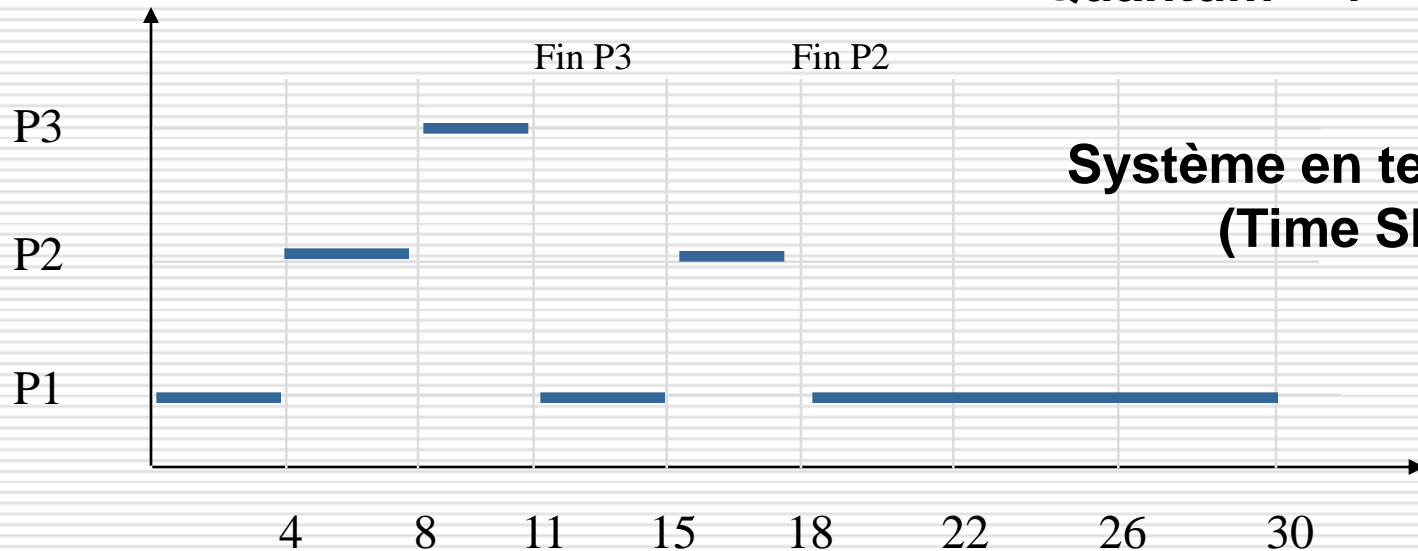
Prêt



Temps de cycle



Quantum = 4



P1	P2	P3	P1	P2	P1
P2	P3	P1	P2	P1	
P3	P1	P2			

**Système en temps partagé
(Time Sharing)**

Algorithme : avec priorités et préemption

Prêt

0	3	10	
5	5	1	
0	4	2	
3	2	5	

Date de soumission Prio Temps de cycle

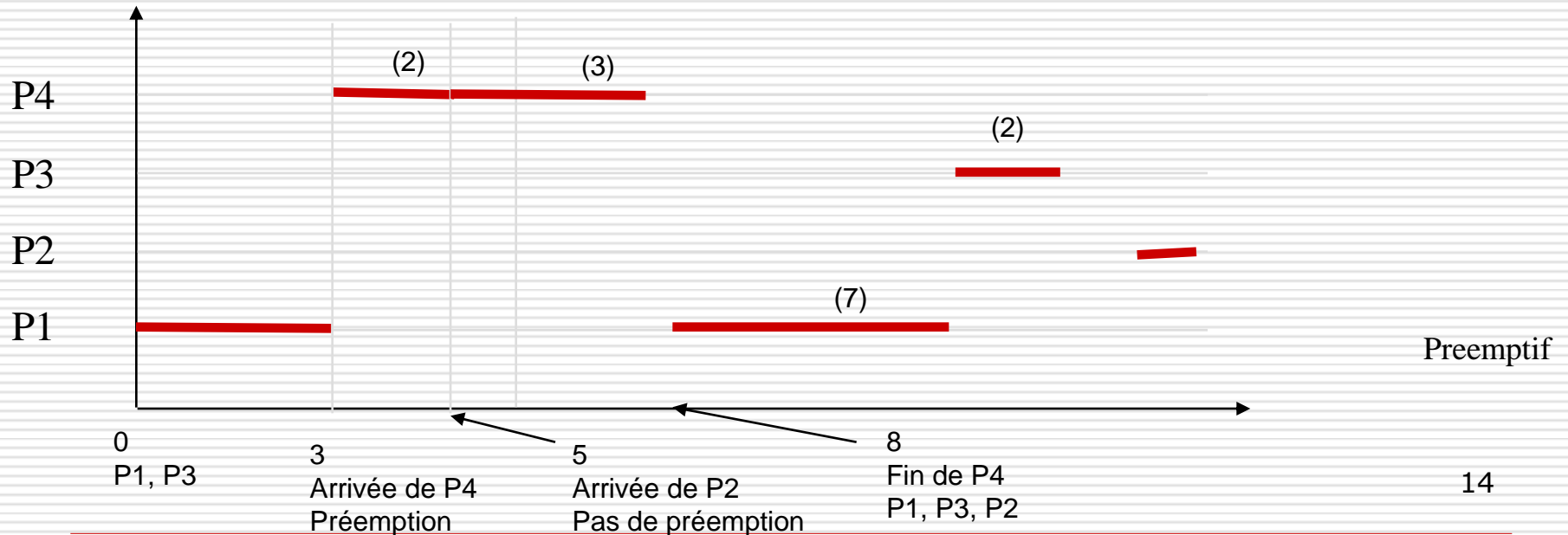
P1

P2

P3

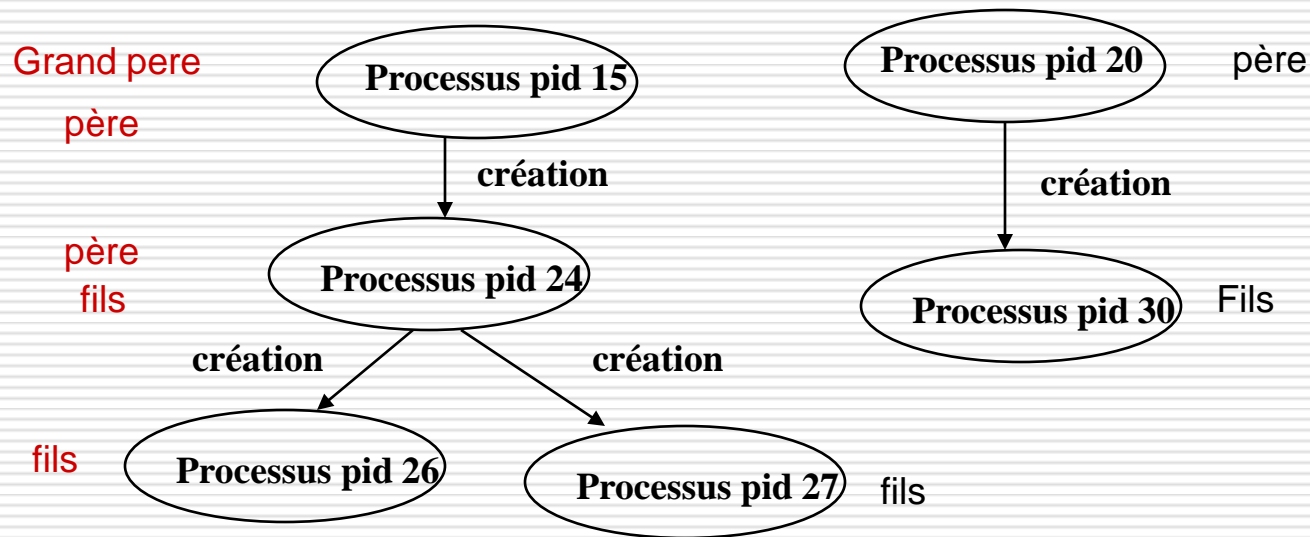
P4

↳ Priorité : le plus petite valeur correspond à la plus forte priorité

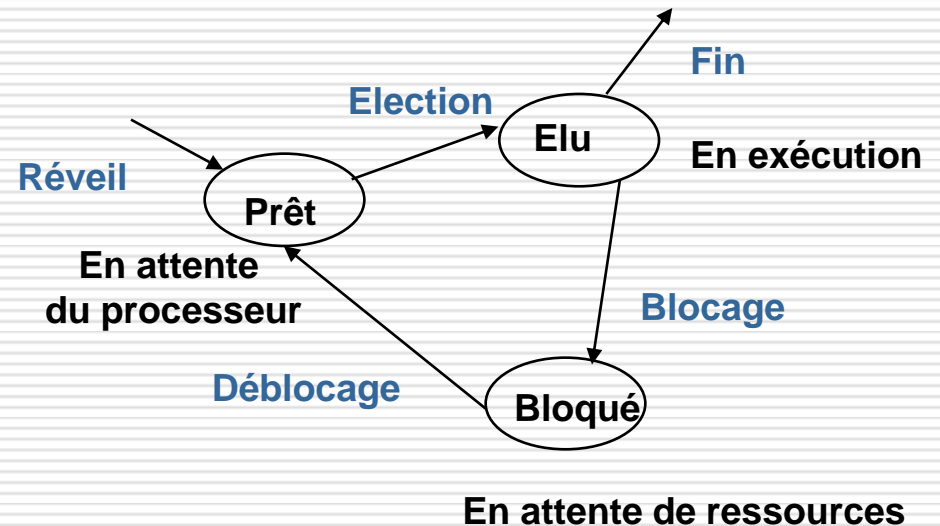
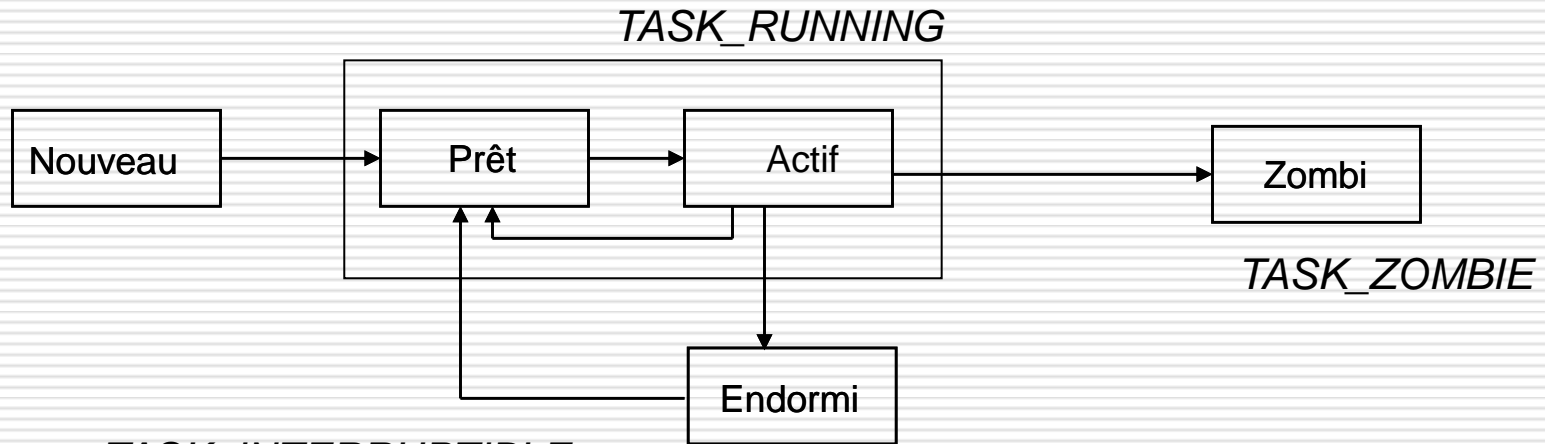


Processus Linux

- ❑ Identifié par un pid
- ❑ Tout processus Linux peut créer un autre processus Linux
 - Arborescence de processus avec un rapport père - fils entre processus créateur et processus crée

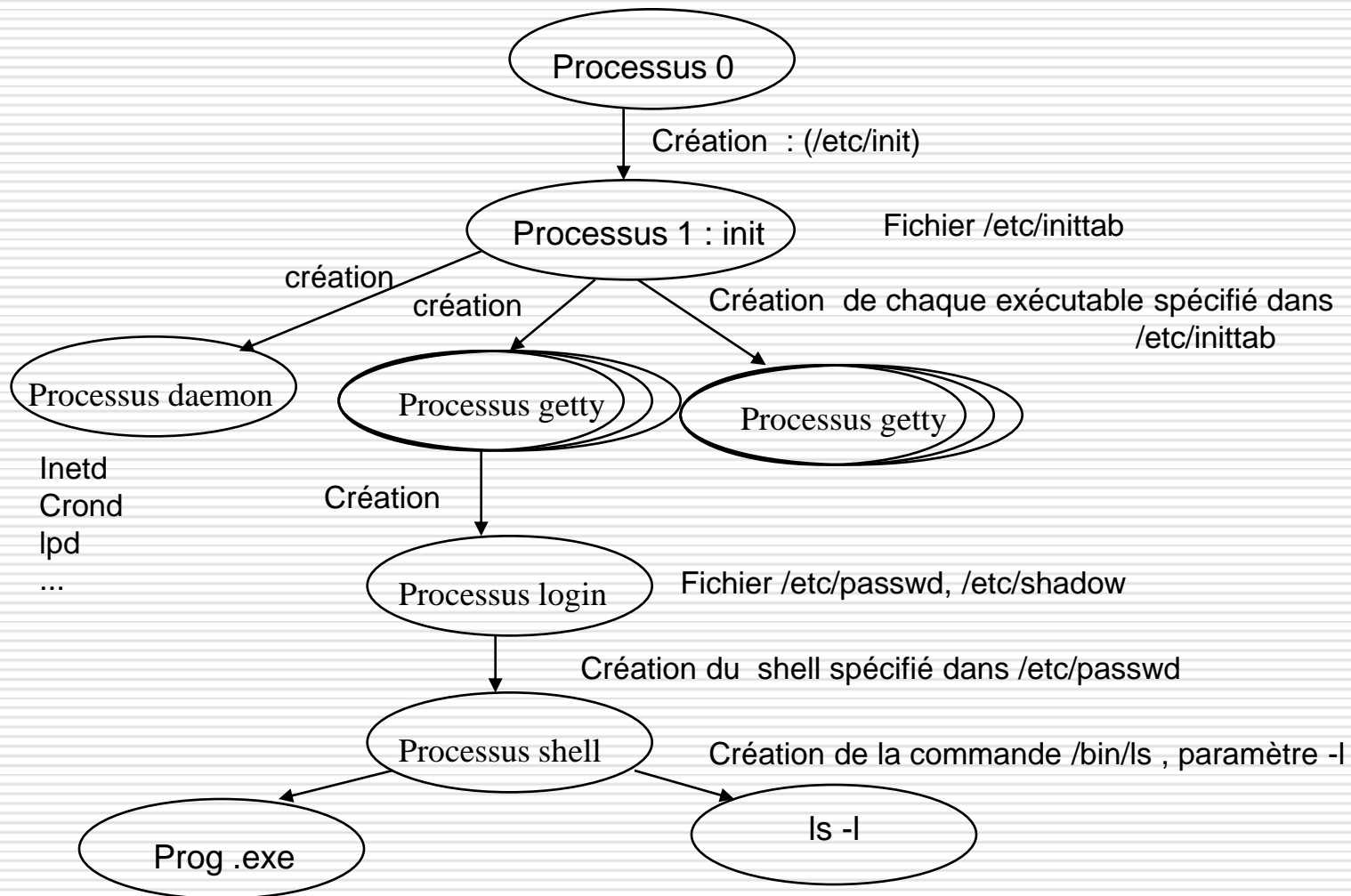


Processus Linux



Processus Linux

- Tout le système Linux repose sur ce concept arborescent



- Politique utilisant à la fois les notions de quantum et de priorité.
- Le processus élu est le prioritaire.
- Il s'exécute durant au plus un quantum de temps
- La priorité des processus est recalculée en fonction de leur utilisation du processeur :
 - Un processus qui a eu le processeur voit sa priorité baisser
 - Un processus qui attend le processeur voit sa priorité augmenter.

□ Lister les processus : ps

- S : état du processus (S Stopped, R Running)
- UID : User Id, nom de l'utilisateur
- PID : Process Id, identifiant du processus
- PPID : Parent Process ID, identifiant du processus père
- PRI : priorité du processus
- TTY : terminal auquel est rattaché le processus
- TIME : durée de traitement du processus
- CMD : commande exécutée

```
linux-9bxb:~/jojo # ps
PID TTY          TIME CMD
3430 pts/1        00:00:00 bash
3866 pts/1        00:00:00 ps
```

```
linux-9bxb:~/jojo # ps -l
F S  UID    PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0     3430  3363  0  80   0  -   3309  wait  pts/1        00:00:00 bash
0 R   0     4015  3430  0  80   0  -   3932  -    pts/1        00:00:00 ps
```

Commandes processus Linux : ps

- ❑ Lister les processus : ps
 - **ps aux** : lister tous les processus présents sur la machine en fournissant le PID, le nom du processus, le nom du propriétaire, le temps d'exécution, l'état du processus ...
 - **ps -u [Utilisateur]** : lister les processus lancés par un utilisateur
- ❑ Lister les processus : top
 - ps donne une liste statique des processus à l'instant du lancement de la commande
 - Top délivre une liste en temps réel.

```
linux-9bxb:~ # top
top - 16:35:04 up 15 min,  3 users,  load average: 1.05, 0.89, 0.52
Tasks: 147 total,  2 running, 145 sleeping,  0 stopped,  0 zombie
%Cpu(s): 25.5 us,  2.1 sy,  0.0 ni, 72.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  5205800 total, 1020828 used, 4184972 free,  35676 buffers
KiB Swap: 8385532 total,  0 used, 8385532 free, 479484 cached
```

□ Commande ps

- délivre la liste des processus avec leur caractéristiques (pid, ppid, état, terminal, durée d'exécution, commande associée...)

S	PID	PPID	PRI	TIME
S	581	579	60	bash
S	592	581	61	essai
S	593	592	61	essai
R	599	580	73	ps



> < >> <<

- Un processus Linux dispose de trois entrées/sorties standard (descripteurs) qui lui permettent de lire /écrire
 - STDIN : entrée standard (le clavier par défaut)
 - STDOUT : sortie standard (l'écran par défaut)
 - STDERR : sortie erreur (pour les messages d'erreur)

- Il est possible de rediriger ces descripteurs par exemple sur un fichier.

- EN SORTIE (STDOUT)

- `$> ls -l > resultat.txt` (le fichier est créé; s'il existe son contenu est effacé)

```
$> cat resultat.txt
```

```
-rw-r--r--  1 delacroix 152 Jan  8  1999 resultat.txt  
-rw-r--r--  1 delacroix 19 Dec 15  1998 essai  
-rw-r--r--  1 delacroix 67 Dec 15  1998 essai.c
```

- `$> date >> resultat.txt` (les données sont ajoutées à la fin du fichier)

```
$> cat resultat.txt
```

```
-rw-r--r--  1 delacroix 182 Jan  8  1999 resultat.txt  
-rw-r--r--  1 delacroix 19 Dec 15  1998 essai  
-rw-r--r--  1 delacroix 67 Dec 15  1998 essai.c
```

Thu Sep 17 15:20:12 CEST 2015

> < >> <<

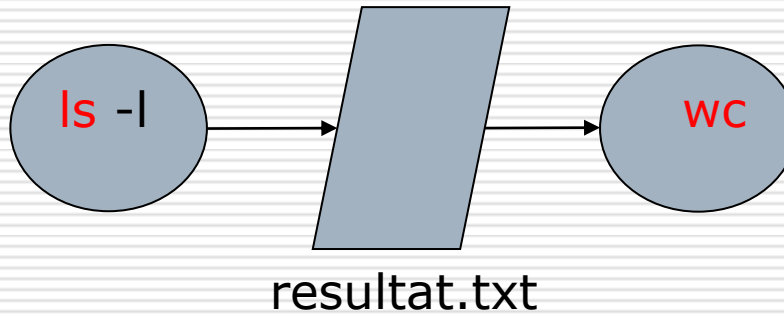
- Un processus Linux dispose de trois entrées/sorties standard (descripteurs) qui lui permettent de lire /écrire
 - STDIN : entrée standard (le clavier par défaut)
 - STDOUT : sortie standard (l'écran par défaut)
 - STDERR : sortie erreur (pour les messages d'erreur)
- Il est possible de rediriger ces descripteurs par exemple sur un fichier.
- EN ENTREE (STDIN)
 - `$> wc < resultat.txt`
5 35 182 (ligne, mots, caractères)

> < >> <<

```
linuxjojo@linuxjojo-VirtualBox:~$  
linuxjojo@linuxjojo-VirtualBox:~$ ls  
Bureau  examples.desktop  Modèles  Public  Vidéos  
Documents  Images  Musique  Téléchargements  
linuxjojo@linuxjojo-VirtualBox:~$ echo "bonjour"  
bonjour  
linuxjojo@linuxjojo-VirtualBox:~$ echo "bonjour" > fichredirect  
linuxjojo@linuxjojo-VirtualBox:~$ ls  
Bureau  examples.desktop  Images  Musique  Téléchargements  
Documents  fichredirect  Modèles  Public  Vidéos  
linuxjojo@linuxjojo-VirtualBox:~$ more fichredirect  
bonjour  
linuxjojo@linuxjojo-VirtualBox:~$ echo "les petits amis" > fichredirect  
linuxjojo@linuxjojo-VirtualBox:~$ more fichredirect  
les petits amis  
linuxjojo@linuxjojo-VirtualBox:~$ echo "bonjour" >> fichredirect  
linuxjojo@linuxjojo-VirtualBox:~$ more fichredirect  
les petits amis  
bonjour  
linuxjojo@linuxjojo-VirtualBox:~$
```


Redirection des entrées – sorties standards les tubes

```
$> ls -l > resultat.txt (le fichier est créé; s'il  
existe son contenu est effacé)  
$> wc < resultat.txt
```



Resultat.txt est un fichier temporaire qui stocke la sortie de « ls-l » pour la donner en entrée à « wc »

→ Un autre outil appelé « tube » peut être utilisé. Il connecte automatiquement la sortie d'un processus sur l'entrée d'un autre processus

```
$> ls -l | wc  
5      35    182 (ligne, mots, caractères)
```

Le nombre de programmes dans un pipeline n'est pas limité.

```
$> ls -l | grep toto | wc -l
```

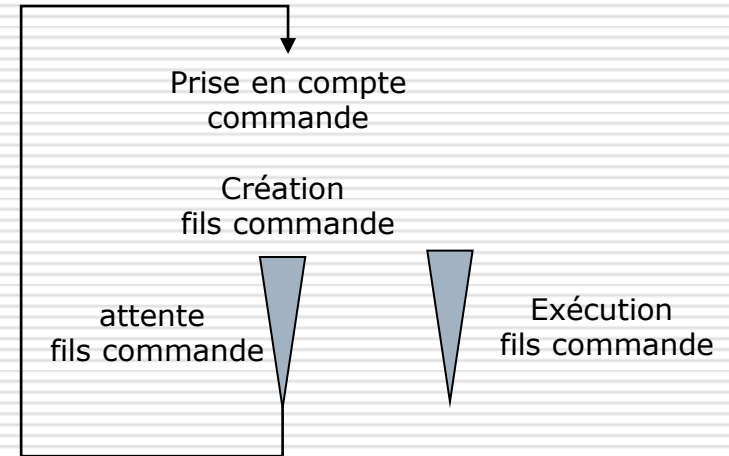
Exécution en arrière plan

- Lorsqu'on lance une commande, par défaut, l'interpréteur de commandes attend la fin du processus créé pour accepter une nouvelle commande.

□ `$> ls -l`
`$> ps`

Est équivalent à

`$> ls -l ; ps`



- Rien n'oblige le shell à attendre la fin d'une commande pour en accepter une nouvelle.

`$> ls -l > listefich &` la redirection évite le mélange des sorties avec ce que l'on fait ensuite

`[1] 3644`
`$>`

Exécution en arrière plan

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l
total 48
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Bureau
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Documents
-rw-r--r-- 1 linuxjojo linuxjojo 8980 juil. 15 18:43 exemples.desktop
-rw-rw-r-- 1 linuxjojo linuxjojo 24 sept. 16 19:26 fichredirect
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Images
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Modèles
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Musique
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Public
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Téléchargements
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Vidéos
linuxjojo@linuxjojo-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2088 pts/14    00:00:00 bash
 2199 pts/14    00:00:00 ps
linuxjojo@linuxjojo-VirtualBox:~$ ls -l & ps
total 48
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Bureau
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Documents
-rw-r--r-- 1 linuxjojo linuxjojo 8980 juil. 15 18:43 exemples.desktop
-rw-rw-r-- 1 linuxjojo linuxjojo 24 sept. 16 19:26 fichredirect
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Images
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Modèles
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Musique
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Public
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Téléchargements
drwxr-xr-x 2 linuxjojo linuxjojo 4096 juil. 15 19:13 Vidéos
  PID TTY          TIME CMD
 2088 pts/14    00:00:00 bash
 2201 pts/14    00:00:00 ps
linuxjojo@linuxjojo-VirtualBox:~$
```

Exécution en arrière plan

```
linuxjojo@linuxjojo-VirtualBox:~$ ./exemple
bonjour les petits amis
la valeur de l'entier est 20
linuxjojo@linuxjojo-VirtualBox:~$ ./exemple&
[1] 3053
linuxjojo@linuxjojo-VirtualBox:~$ bonjour les petits amis

linuxjojo@linuxjojo-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2088 pts/14    00:00:00 bash
 3053 pts/14    00:00:00 exemple
 3054 pts/14    00:00:00 ps
linuxjojo@linuxjojo-VirtualBox:~$ la valeur de l'entier est 20
```

Arrêter un processus : kill

- ❑ La commande `kill` permet d'envoyer un signal à un processus.
- ❑ Un signal est un moyen de communication entre processus; il permet de spécifier à un processus qu'un évènement est arrivé. Chaque signal est identifié par un nom et un numéro
- ❑ Le processus réagit au signal reçu (par exemple en s'arrêtant)
 - `kill - numerosignal pid`
 - `SIGKILL 9` Force le processus à se terminer.
 - `SIGTERM 15` signal par défaut. Termine le processus en « douceur ».

```
linux-9bxb:~ # ./essaibis&
[1] 4052
linux-9bxb:~ # ps
  PID TTY          TIME CMD
 3424 pts/1        00:00:00 bash
 4052 pts/1        00:00:00 essaibis
 4055 pts/1        00:00:00 ps
linux-9bxb:~ # kill 4052
[1]+  Terminated                  ./essaibis
linux-9bxb:~ # ps
  PID TTY          TIME CMD
 3424 pts/1        00:00:00 bash
 4060 pts/1        00:00:00 ps
linux-9bxb:~ #
```

Autres commandes

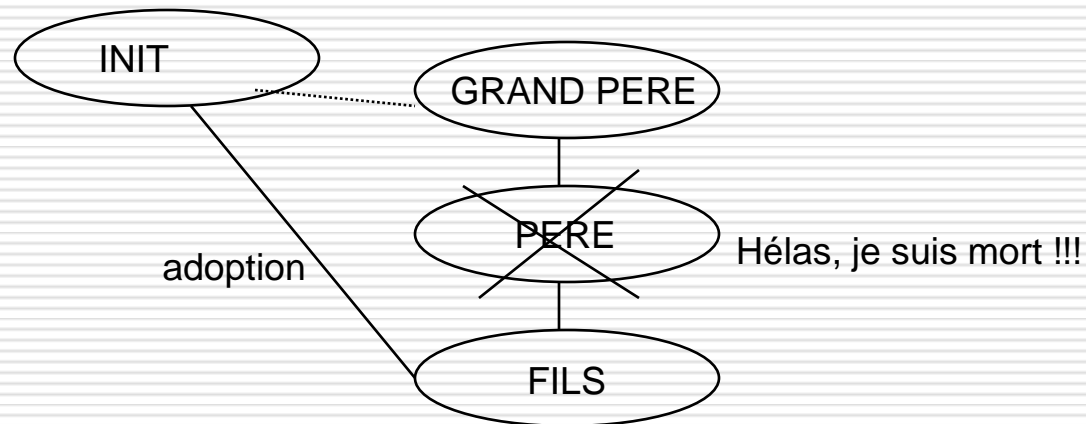
- ❑ `$> kill 0`
 - termine tous les processus de l'utilisateur sauf le shell initial
- ❑ `$> nice commande_longue &`
 - Baisse la priorité du processus « `commande_longue` »

```
linuxjojo@linuxjojo-VirtualBox:~$ ./exemple > fichtrace &
[1] 3099
linuxjojo@linuxjojo-VirtualBox:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  2088  2084  0  80   0 -  6758 wait  pts/14    00:00:00 bash
0 S  1000  3099  2088  0  80   0 -  1051 hrtime pts/14    00:00:00 exemple
0 R  1000  3105  2088  0  80   0 -  3562 -    pts/14    00:00:00 ps
linuxjojo@linuxjojo-VirtualBox:~$ nice ./exemple > fichtrace2 &
[2] 3106
[1] Terminé 29
./exemple > fichtrace
linuxjojo@linuxjojo-VirtualBox:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  2088  2084  0  80   0 -  6758 wait  pts/14    00:00:00 bash
0 S  1000  3106  2088  0  90  10 -  1051 hrtime pts/14    00:00:00 exemple
0 R  1000  3107  2088  0  80   0 -  3562 -    pts/14    00:00:00 ps
linuxjojo@linuxjojo-VirtualBox:~$
```

Génétique des processus Linux

□ Décès et adoption

- 1. Un processus fils défunt reste zombie jusqu'à ce que son contexte soit totalement détruit.
- 2. Un processus fils orphelin, suite au décès de son père (le processus père s'est terminé avant son fils) est toujours adopté par le processus 1 (Init).



Génétique des processus Linux

Cas 1

```
linux-9bxb:~ # ./essai &
[1] 3997
linux-9bxb:~ # ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	3432	3368	0	80	0	-	3309	wait	pts/1	00:00:00	bash
0	S	0	3997	3432	0	80	0	-	1022	-	pts/1	00:00:00	essai
1	S	0	4000	3997	0	80	0	-	1022	-	pts/1	00:00:00	essai
0	R	0	4001	3432	0	80	0	-	3932	-	pts/1	00:00:00	ps

```
linux-9bxb:~ # kill 4000
linux-9bxb:~ # ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	3432	3368	0	80	0	-	3309	wait	pts/1	00:00:00	bash
0	S	0	3997	3432	0	80	0	-	1022	-	pts/1	00:00:00	essai
1	Z	0	4000	3997	0	80	0	-	0	exit	pts/1	00:00:00	essai <defunct>
0	R	0	4006	3432	0	80	0	-	3932	-	pts/1	00:00:00	ps

Génétique des processus Linux

Cas 2

```
linux-9bxb:~ # ./essai &
[2] 3957
linux-9bxb:~ # ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	3432	3368	0	80	0	-	3309	wait	pts/1	00:00:00	bash
0	S	0	3948	3432	0	80	0	-	1021	-	pts/1	00:00:00	essaibis
0	S	0	3957	3432	0	80	0	-	1022	-	pts/1	00:00:00	essai
1	S	0	3960	3957	0	80	0	-	1022	-	pts/1	00:00:00	essai
0	R	0	3961	3432	0	80	0	-	3932	-	pts/1	00:00:00	ps

```
linux-9bxb:~ # kill 3957
[2]+  Terminated                  ./essai
linux-9bxb:~ # ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	3432	3368	0	80	0	-	3309	wait	pts/1	00:00:00	bash
0	S	0	3948	3432	0	80	0	-	1021	-	pts/1	00:00:00	essaibis
1	S	0	3960	1	0	80	0	-	1022	-	pts/1	00:00:00	essai
0	R	0	3967	3432	0	80	0	-	3932	-	pts/1	00:00:00	ps