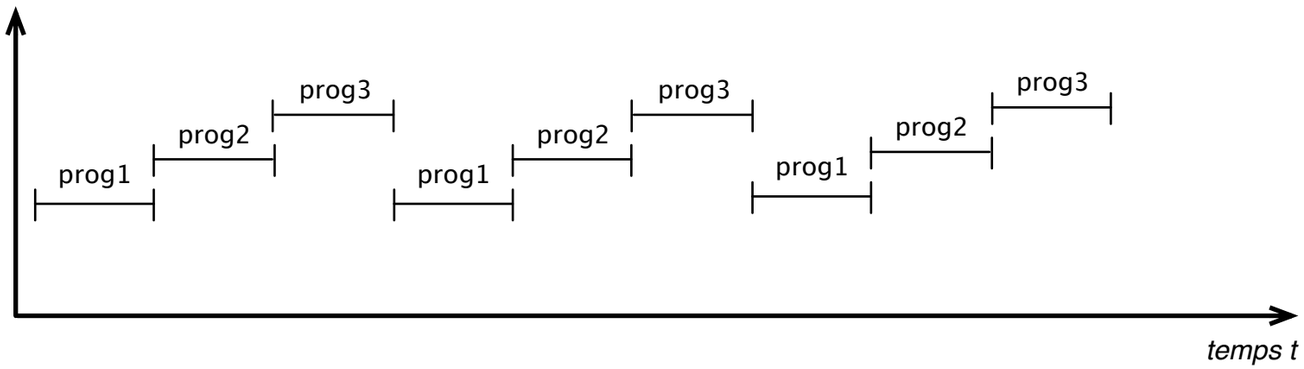
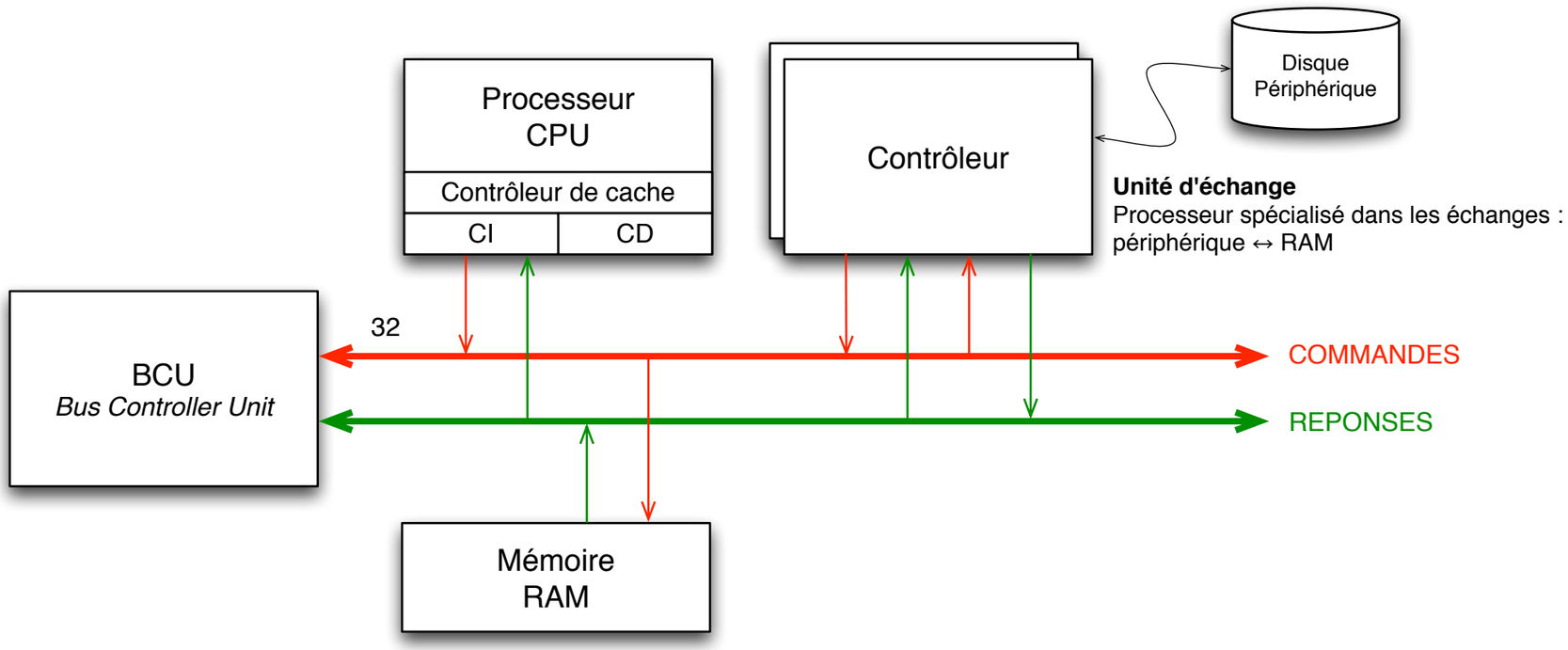
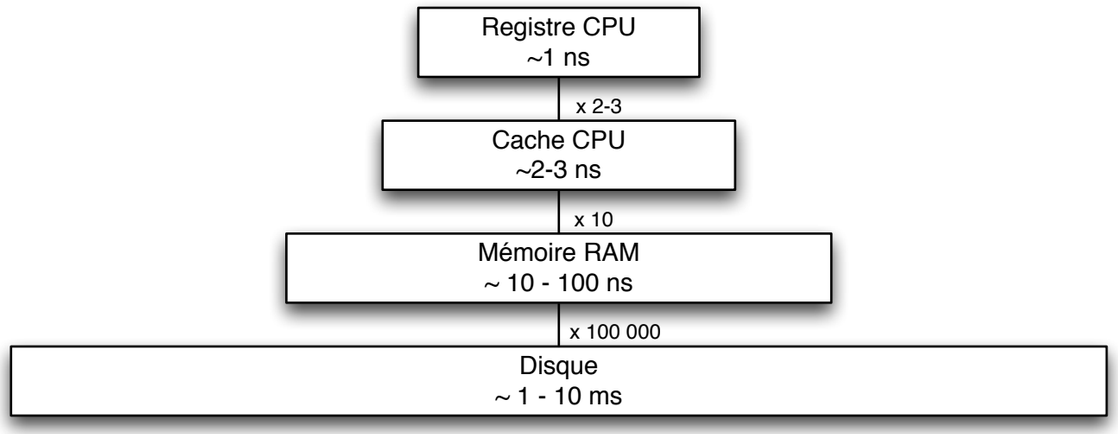
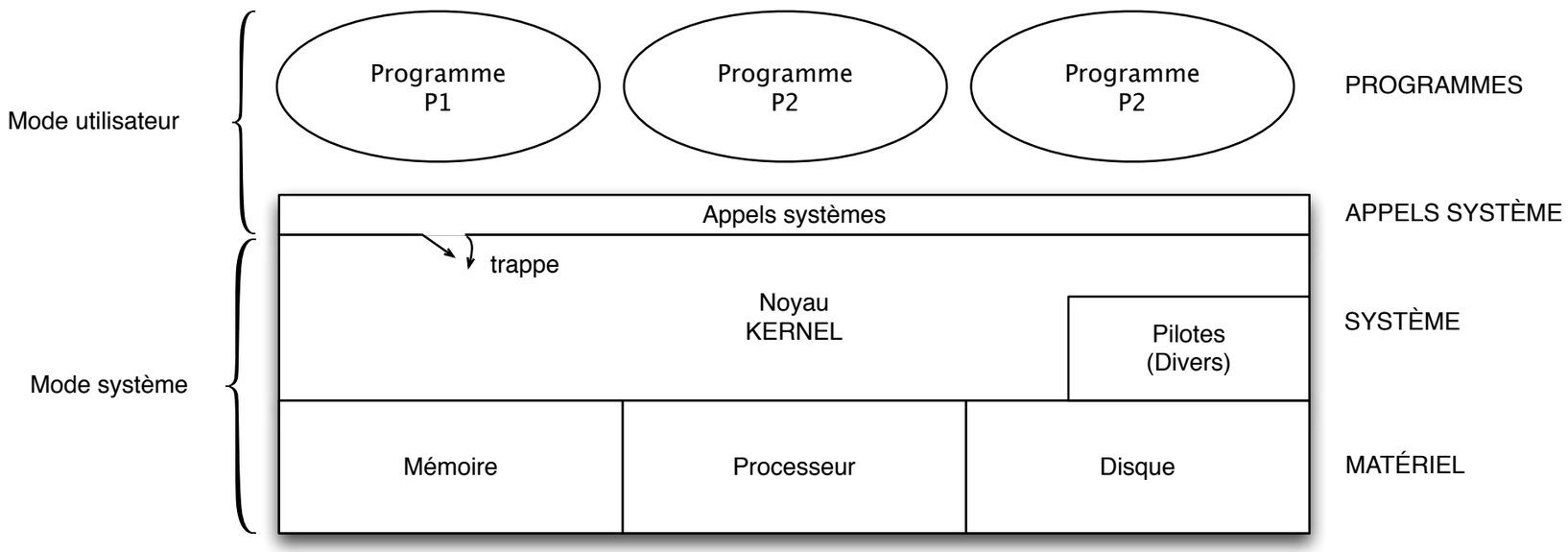
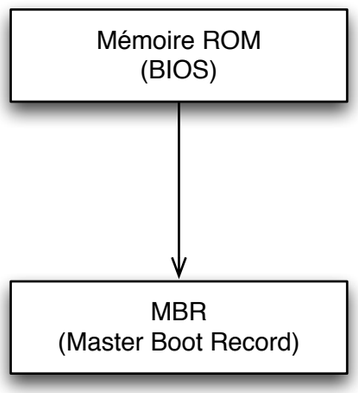
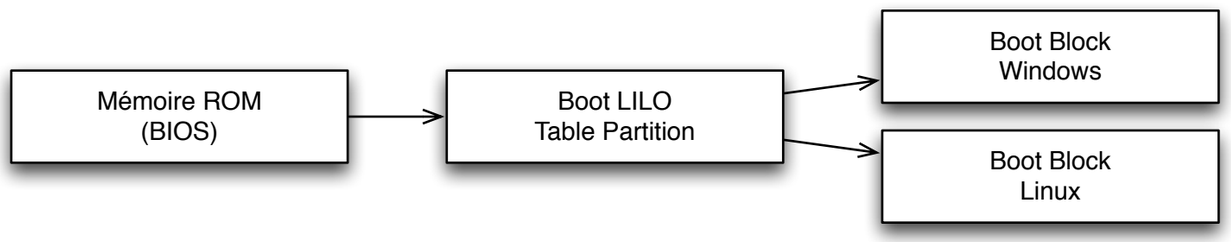

prog1
prog2
prog3
prog1
prog2
prog3
prog1
prog2
prog3
temps t

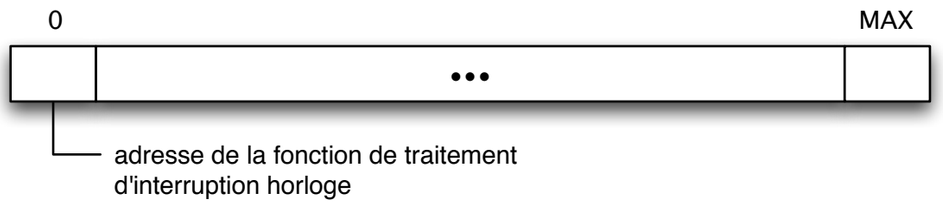

Processeur
CPU
Contrôleur de cache
CI
CD
Contrôleur
Disque
Périphérique
Unité d'échange
Processeur spécialisé dans les échanges :
périphérique ↔ RAM
BCU
Bus Controller Unit
32
COMMANDES
REPONSES
Mémoire
RAM

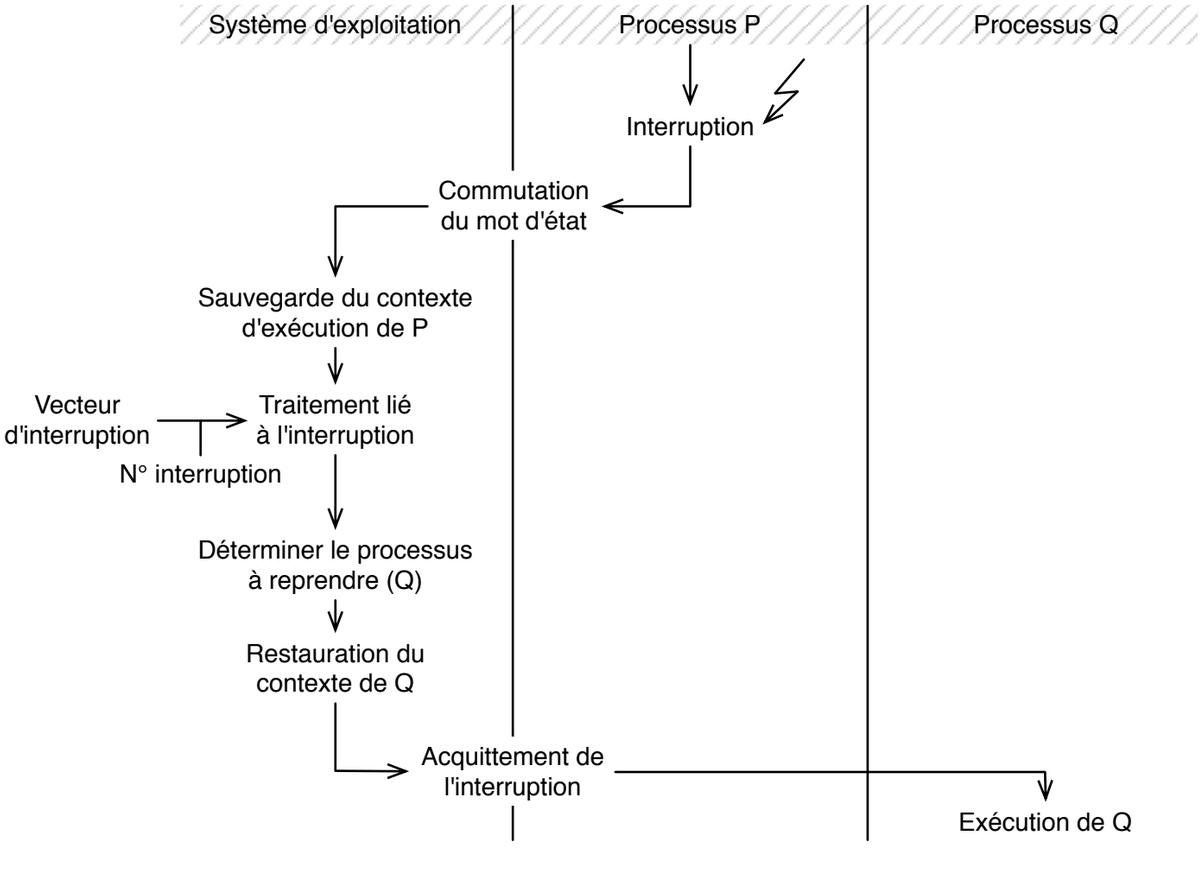

Registre CPU
~1 ns
x 2-3
Cache CPU
~2-3 ns
x 10
Mémoire RAM
~ 10 - 100 ns
x 100 000
Disque
~ 1 - 10 ms

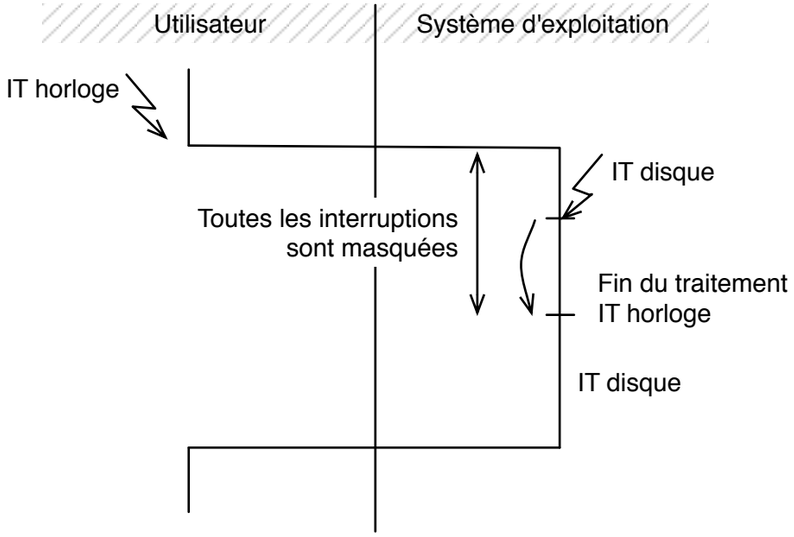

Mode utilisateur
Mode système
Programme P1
Programme P2
Programme P2
Appels systèmes
trappe
Noyau KERNEL
Pilotes (Divers)
Mémoire
Processeur
Disque
PROGRAMMES
APPELS SYSTÈME
SYSTÈME
MATÉRIEL

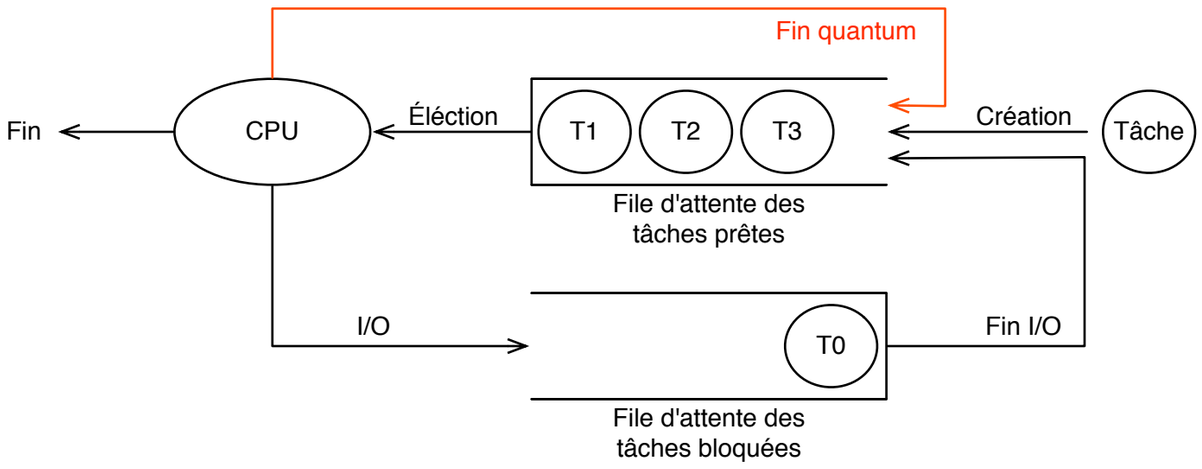

Mémoire ROM
(BIOS)
MBR
(Master Boot Record)

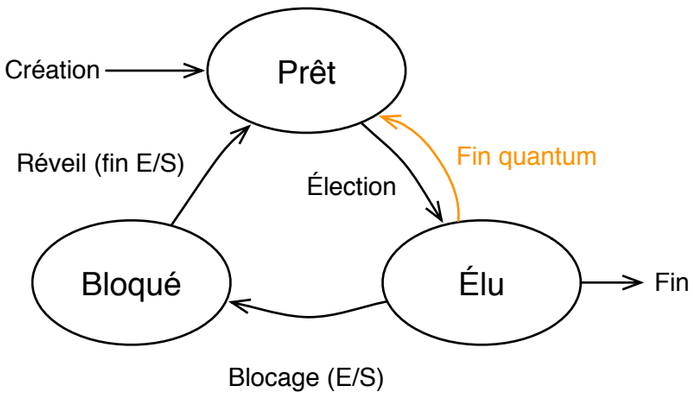

Mémoire ROM
(BIOS)
Boot LILO
Table Partition
Boot Block
Windows
Boot Block
Linux

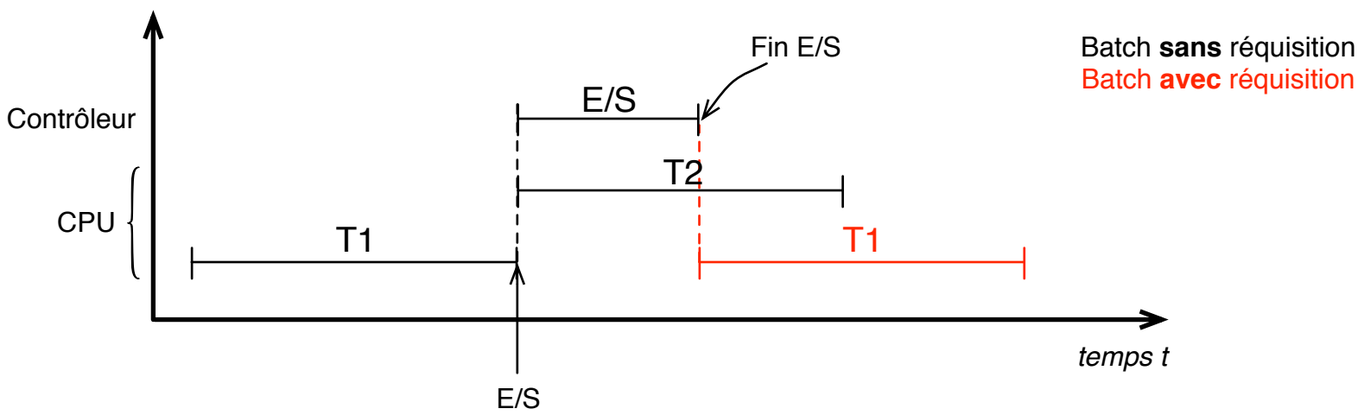

0
MAX
adresse de la fonction de traitement
d'interruption horloge


Système d'exploitation
Processus P
Processus Q
Interruption
Commutation
du mot d'état
Sauvegarde du contexte
d'exécution de P
Vecteur
d'interruption
Traitement lié
à l'interruption
N° interruption
Déterminer le processus
à reprendre (Q)
Restauration du
contexte de Q
Acquittement de
l'interruption
Exécution de Q

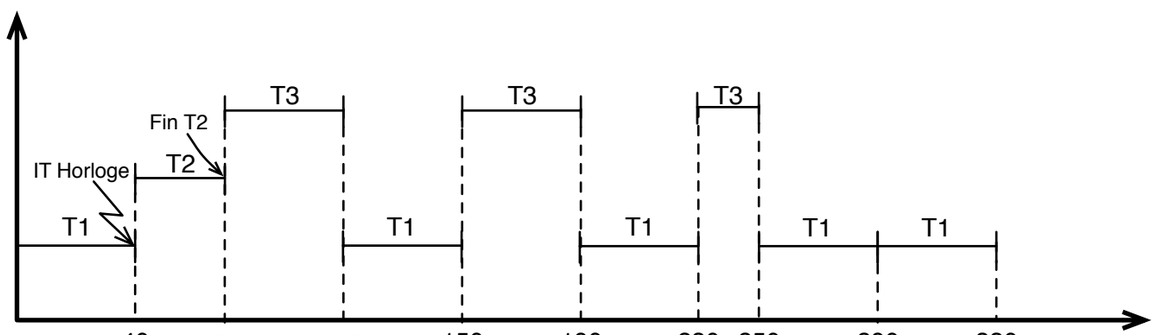

Utilisateur
Système d'exploitation
IT horloge
IT disque
Toutes les interruptions
sont masquées
Fin du traitement
IT horloge
IT disque

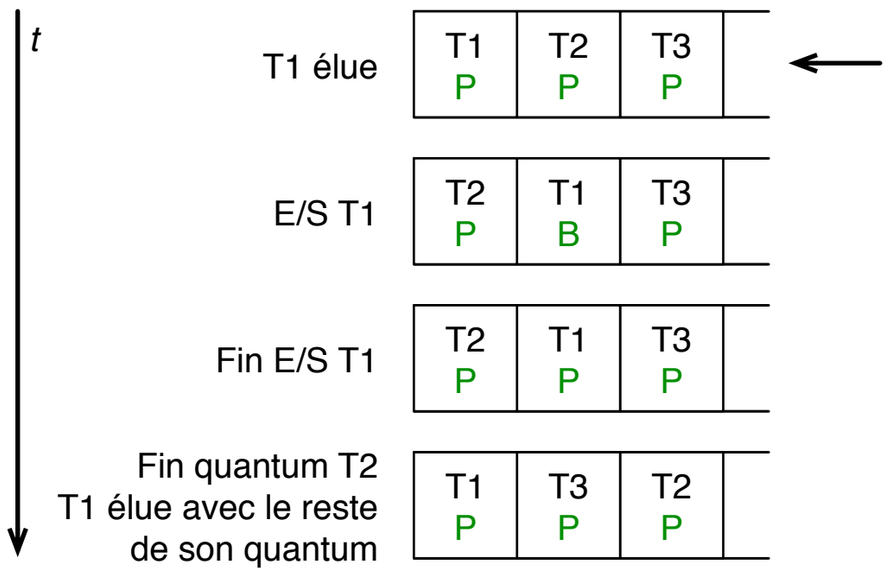

Fin quantum
Fin
CPU
Éléction
T1
T2
T3
Création
Tâche
File d'attente des
tâches prêtes
I/O
T0
Fin I/O
File d'attente des
tâches bloquées

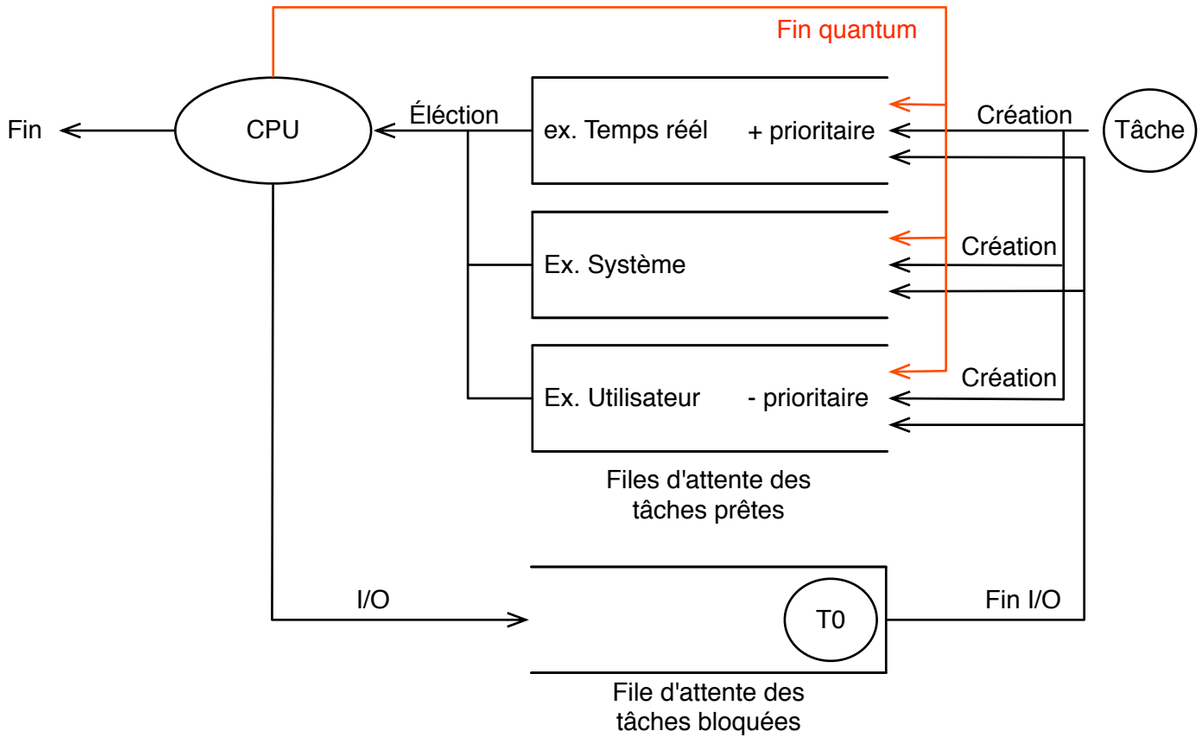

Création
Prêt
Réveil (fin E/S)
Élection
Fin quantum
Bloqué
Élu
Fin
Blocage (E/S)

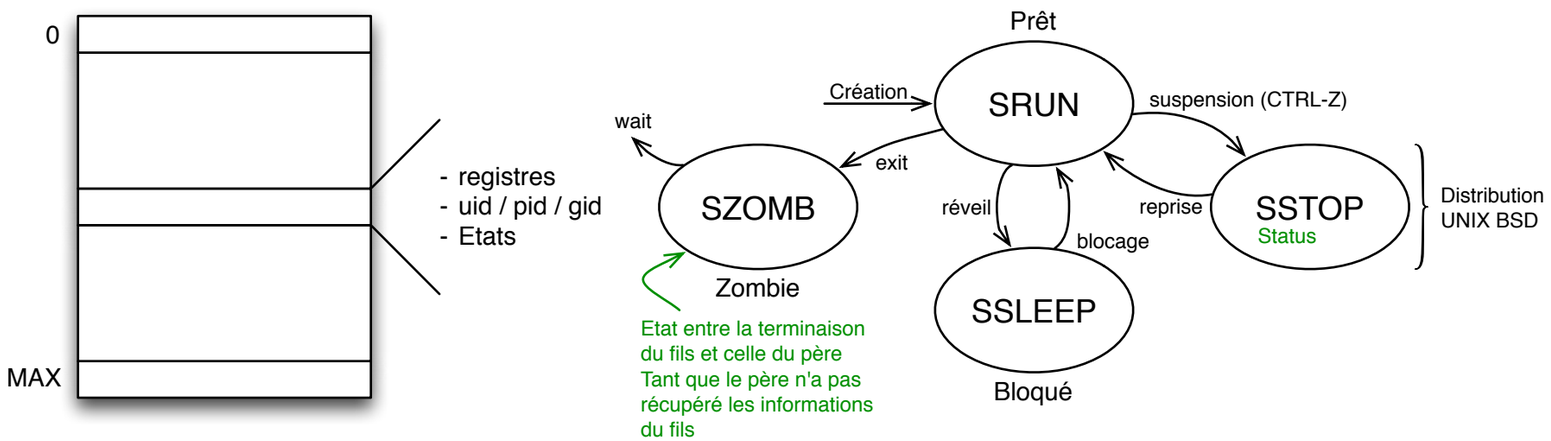


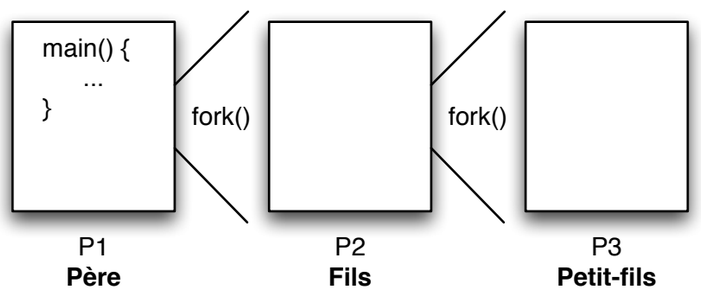
Batch **sans** réquisition
Batch **avec** réquisition

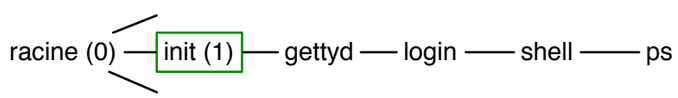

T3
T3
T3
Fin T2
T2
IT Horloge
T1
T1
T1
T1
T1
40
70
110
150
190
230
250
290
330
temps t
T1 T2 T3
T2 T3 T1
T3 T1
T1 T3
T3 T1
T1 T3
T3 T1 T1
T1

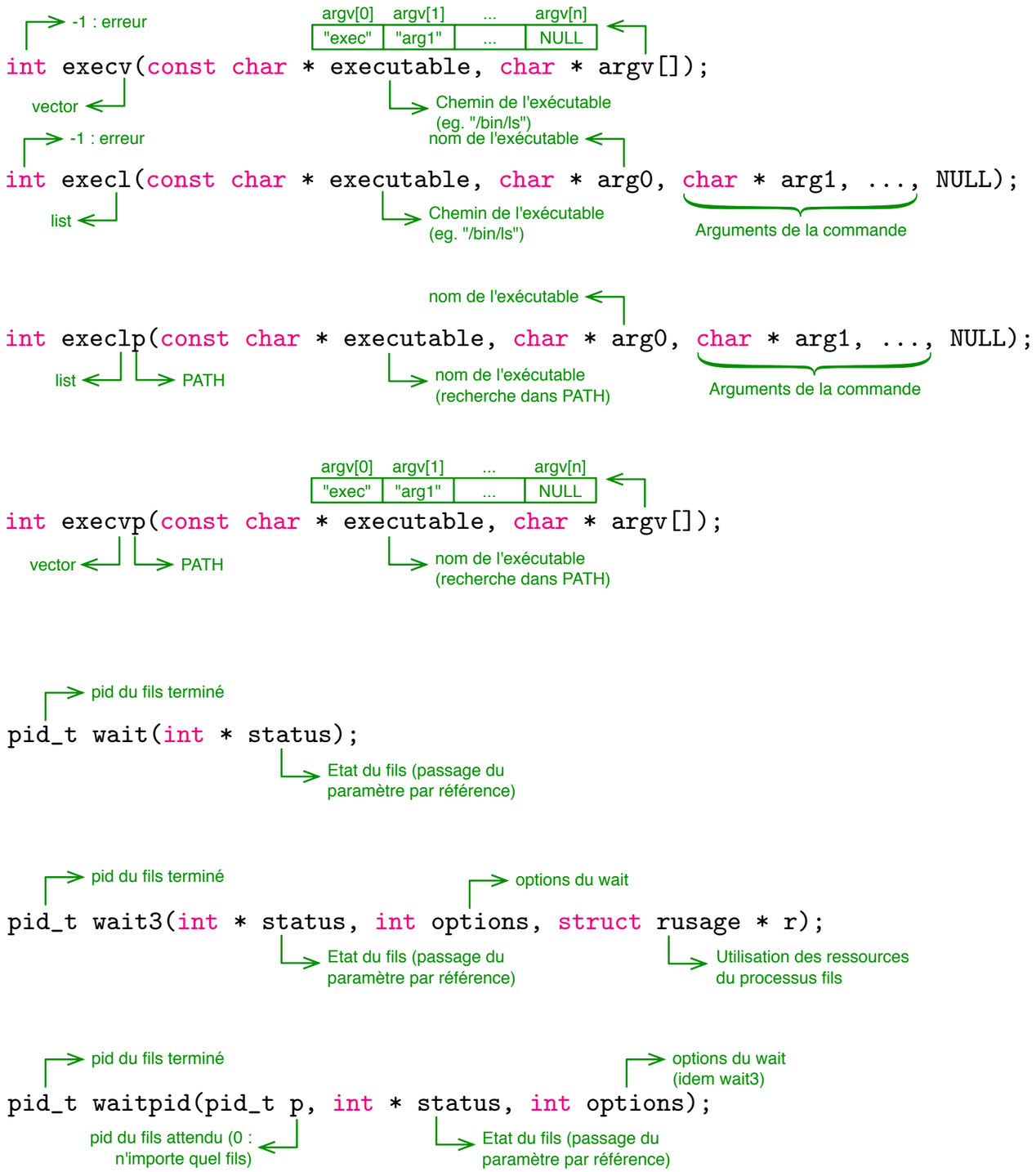

t
T1 élue
T1 P
T2 P
T3 P
E/S T1
T2 P
T1 B
T3 P
Fin E/S T1
T2 P
T1 P
T3 P
Fin quantum T2
T1 élue avec le reste
de son quantum
T1 P
T3 P
T2 P


Fin quantum
Fin
CPU
Éléction
ex. Temps réél
+ prioritaire
Création
Tâche
Ex. Système
Création
Ex. Utilisateur
- prioritaire
Création
Files d'attente des
tâches prêtes
I/O
T0
Fin I/O
File d'attente des
tâches bloquées

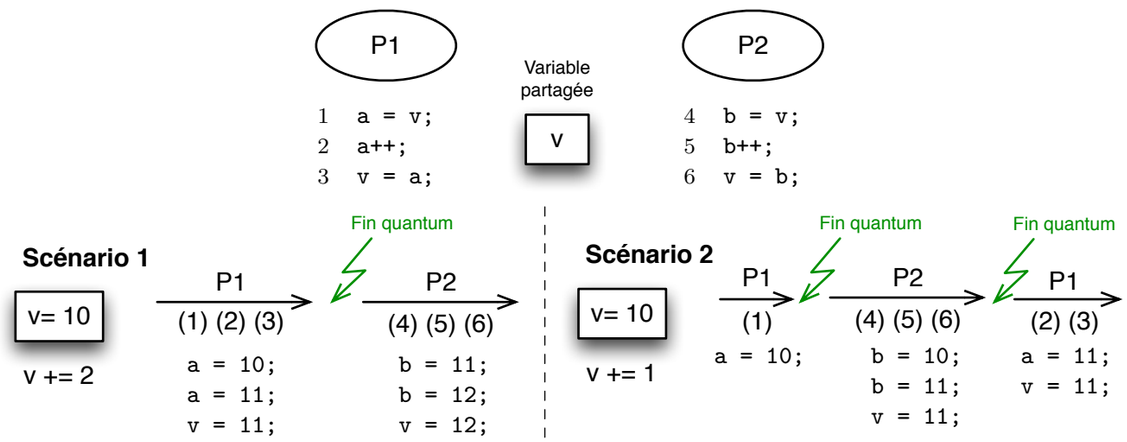

0
MAX
- registres
- uid / pid / gid
- Etats
Prêt
SRUN
Création
suspension (CTRL-Z)
wait
exit
SZOMB
Zombie
réveil
blocage
reprise
SSTOP
Status
Distribution
UNIX BSD
SSLEEP
Bloqué
Etat entre la terminaison
du fils et celle du père
Tant que le père n'a pas
récupéré les informations
du fils


main() {
...
}
fork()
fork()
P1
Père
P2
Fils
P3
Petit-fils


racine (0)
init (1)
gettyd
login
shell
ps


-1 : erreur
argv[0] argv[1] ... argv[n]
"exec" "arg1" ... NULL
int execv(const char * executable, char * argv[]);
vector
Chemin de l'exécutable
(eg. "/bin/ls")
-1 : erreur
nom de l'exécutable
int execl(const char * executable, char * arg0, char * arg1, ..., NULL);
list
Chemin de l'exécutable
(eg. "/bin/ls")
Arguments de la commande
nom de l'exécutable
int execlp(const char * executable, char * arg0, char * arg1, ..., NULL);
list
PATH
nom de l'exécutable
(recherche dans PATH)
Arguments de la commande
argv[0] argv[1] ... argv[n]
"exec" "arg1" ... NULL
int execvp(const char * executable, char * argv[]);
vector
PATH
nom de l'exécutable
(recherche dans PATH)

pid du fils terminé
pid_t wait(int * status);
Etat du fils (passage du
paramètre par référence)
pid du fils terminé
options du wait
pid_t wait3(int * status, int options, struct rusage * r);
Etat du fils (passage du
paramètre par référence)
Utilisation des ressources
du processus fils
pid du fils terminé
options du wait
(idem wait3)
pid_t waitpid(pid_t p, int * status, int options);
pid du fils attendu (0 :
n'importe quel fils)
Etat du fils (passage du
paramètre par référence)


P1
Variable partagée
v
P2
1 a = v;
2 a++;
3 v = a;
4 b = v;
5 b++;
6 v = b;
Scénario 1
v= 10
v += 2
P1
(1) (2) (3)
Fin quantum
P2
(4) (5) (6)
a = 10;
a = 11;
v = 11;
b = 11;
b = 12;
v = 12;
Scénario 2
v= 10
v += 1
P1
(1)
a = 10;
Fin quantum
P2
(4) (5) (6)
b = 10;
b = 11;
v = 11;
Fin quantum
P1
(2) (3)
a = 11;
v = 11;

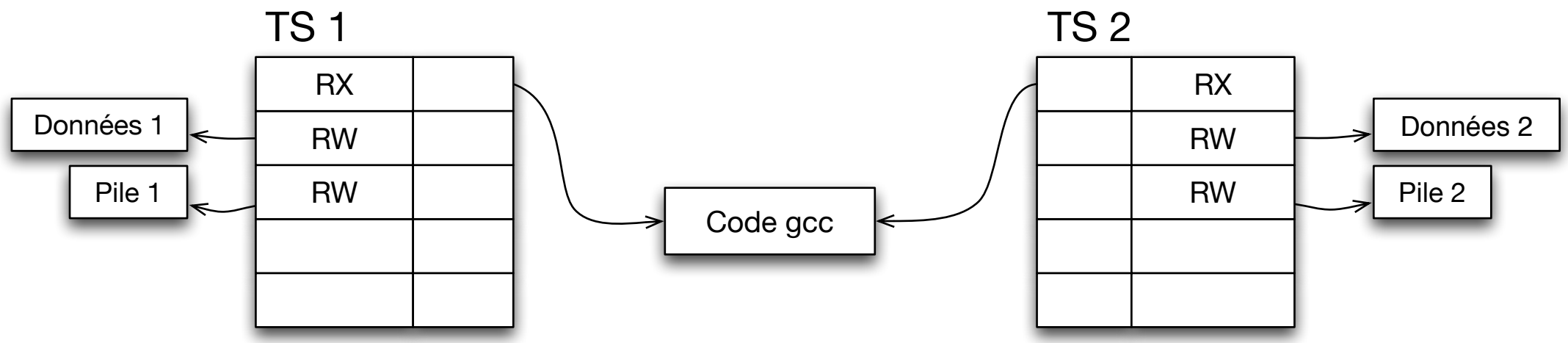
```
bool flag[2] = {false, false};
int tour;
```

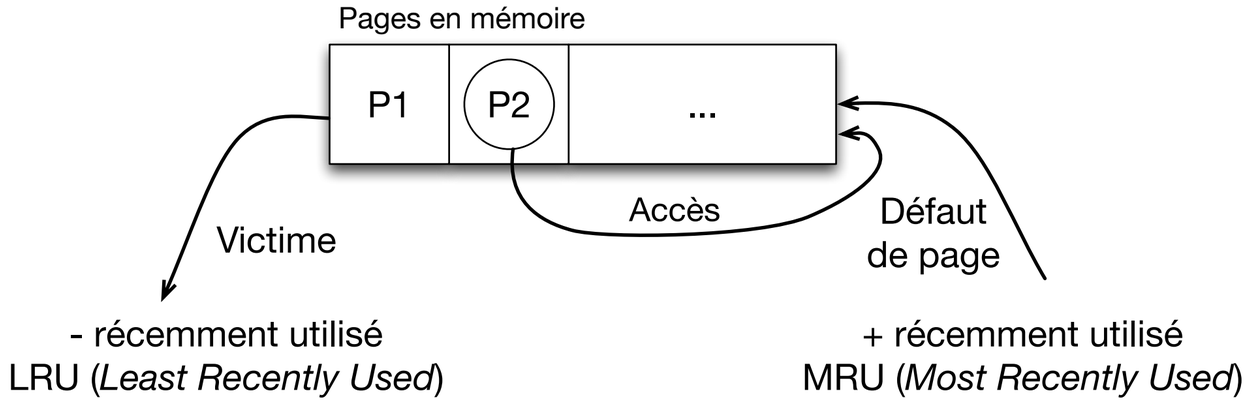
EnterSC()
Attente active

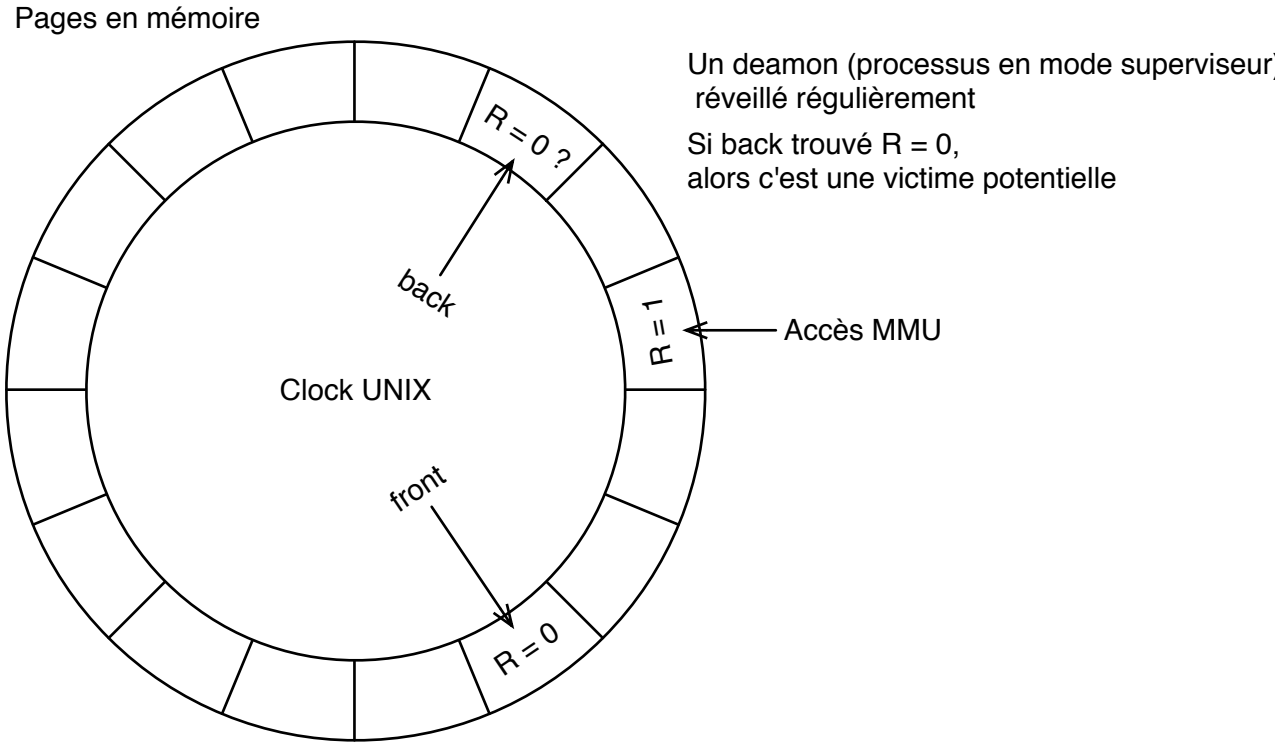
```
flag[i] = true;
tour = j;
while(flag[j] == true && tour == j);

        /* Section Critique */
```

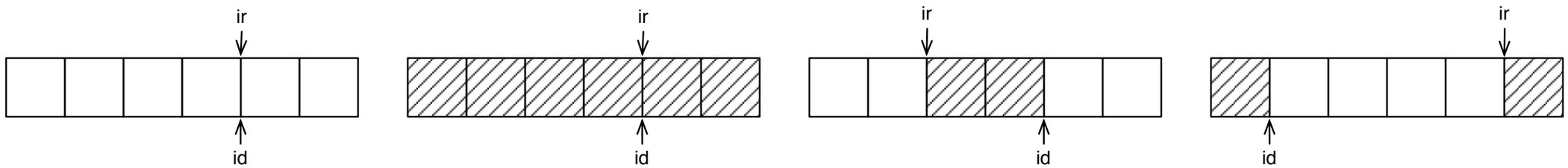
SortirSC()

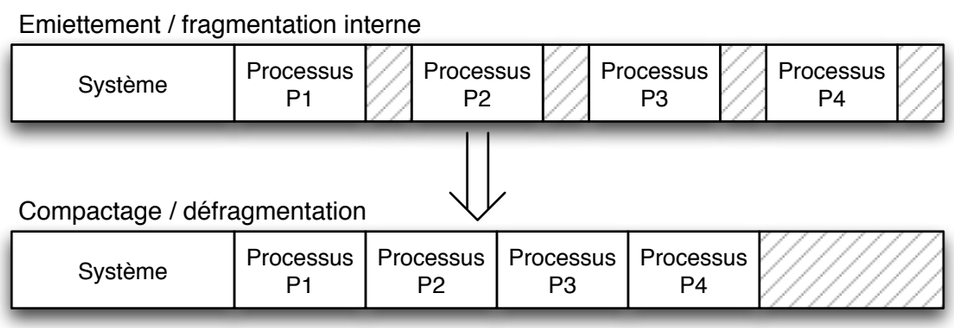
```
flag[i] = false;
```

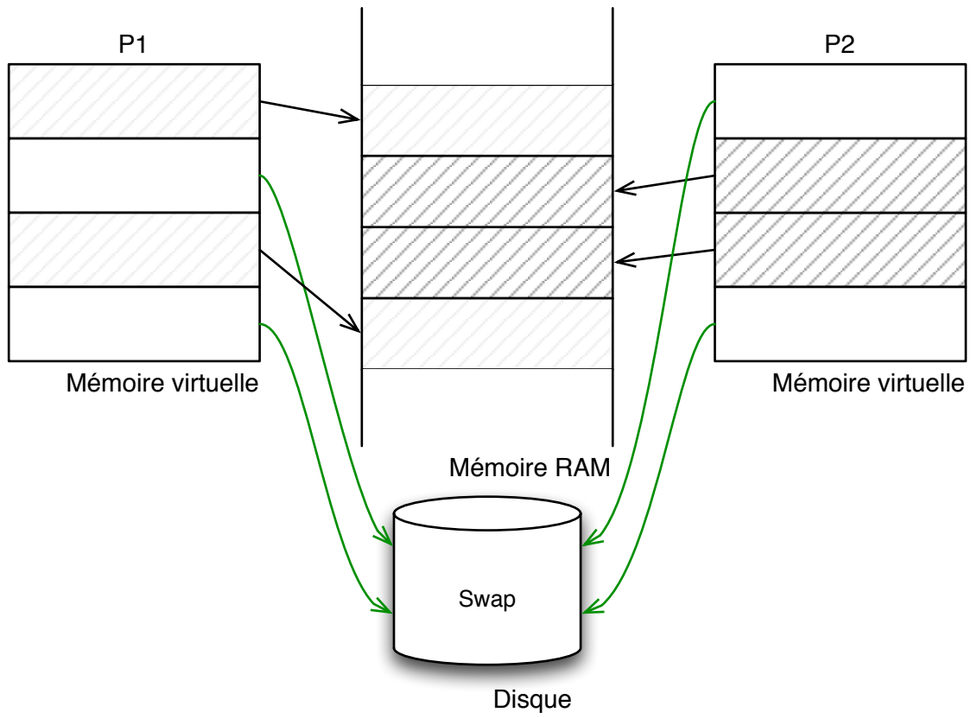

TS 1
RX
RW
RW
Données 1
Pile 1
Code gcc
TS 2
RX
RW
RW
Données 2
Pile 2

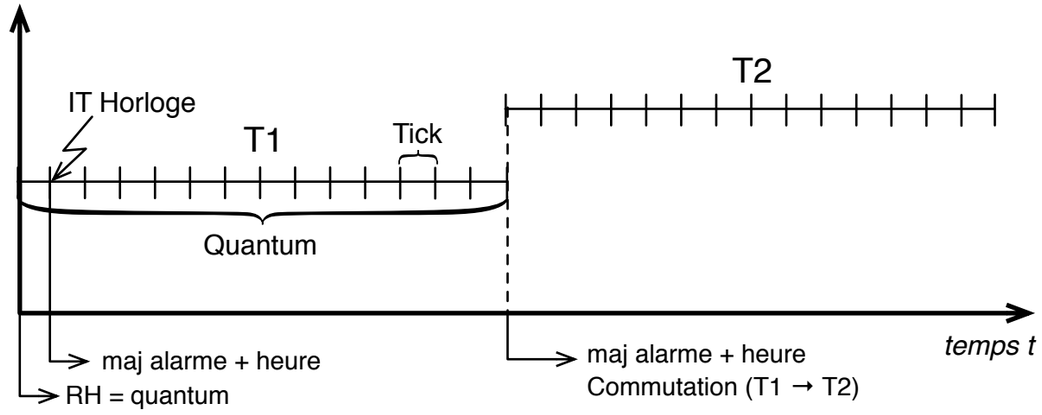

Pages en mémoire
P1
P2
...
Accès
Victime
Défaut
de page
- récemment utilisé
LRU (*Least Recently Used*)
+ récemment utilisé
MRU (*Most Recently Used*)


Pages en mémoire
Un deamon (processus en mode superviseur)
réveillé régulièrement
Si back trouvé R = 0,
alors c'est une victime potentielle
R = 0 ?
back
R = 1
Accès MMU
Clock UNIX
front
R = 0

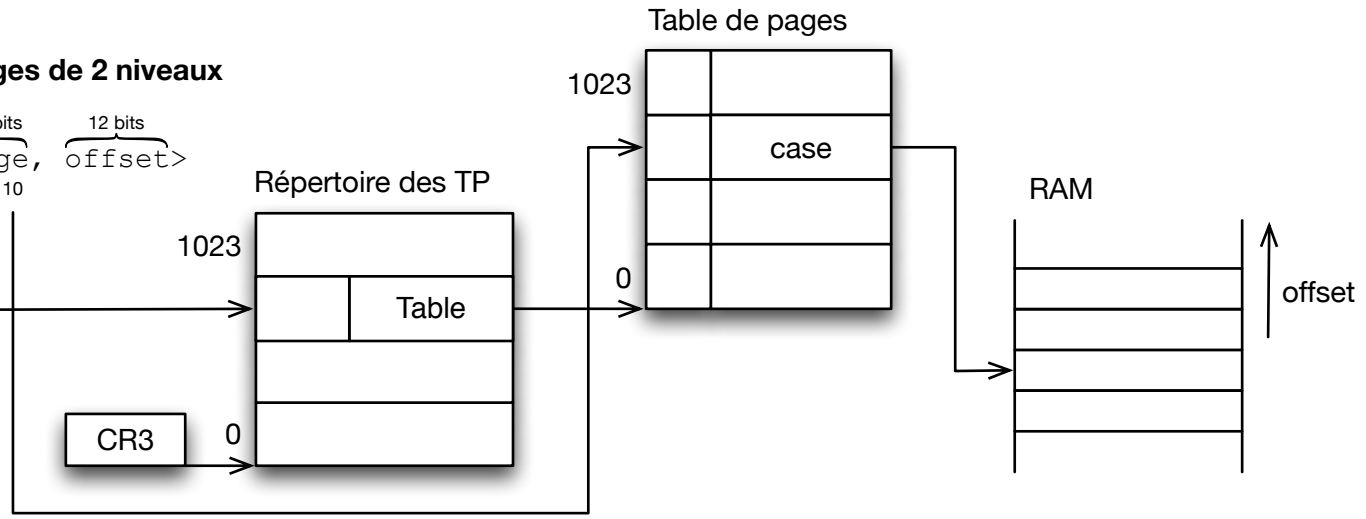

ir
id
ir
id
ir
id
ir
id


Emiettement / fragmentation interne
Système
Processus P1
Processus P2
Processus P3
Processus P4
Compactage / défragmentation
Système
Processus P1
Processus P2
Processus P3
Processus P4

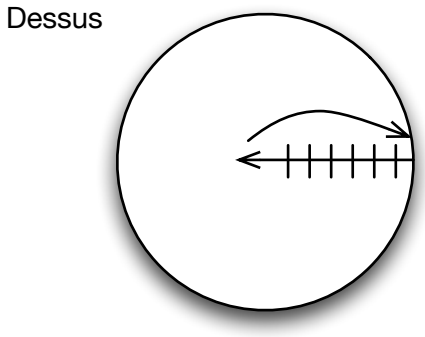

P1
P2
Mémoire virtuelle
Mémoire virtuelle
Mémoire RAM
Swap
Disque


T2
IT Horloge
T1
Tick
Quantum
maj alarme + heure
RH = quantum
maj alarme + heure
Commutation (T1 → T2)
temps t

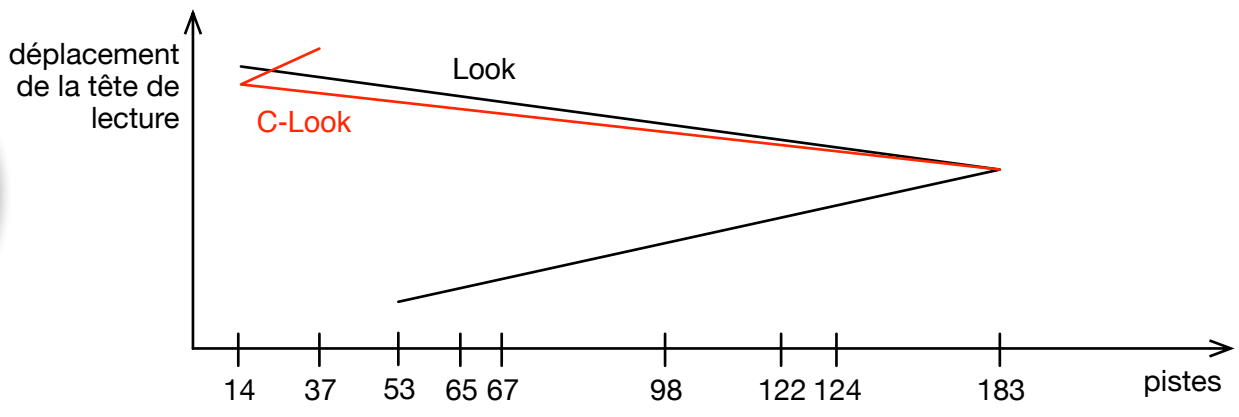
Pages de 2 niveaux



Variante : **C-Look** (*Circular Look*)



Look (balayage)



bloc 0
bloc 1
bloc N-1
bloc N
DATA
NEXT
...
-1
Répertoire
"f1"
20
130
taille
50
130