



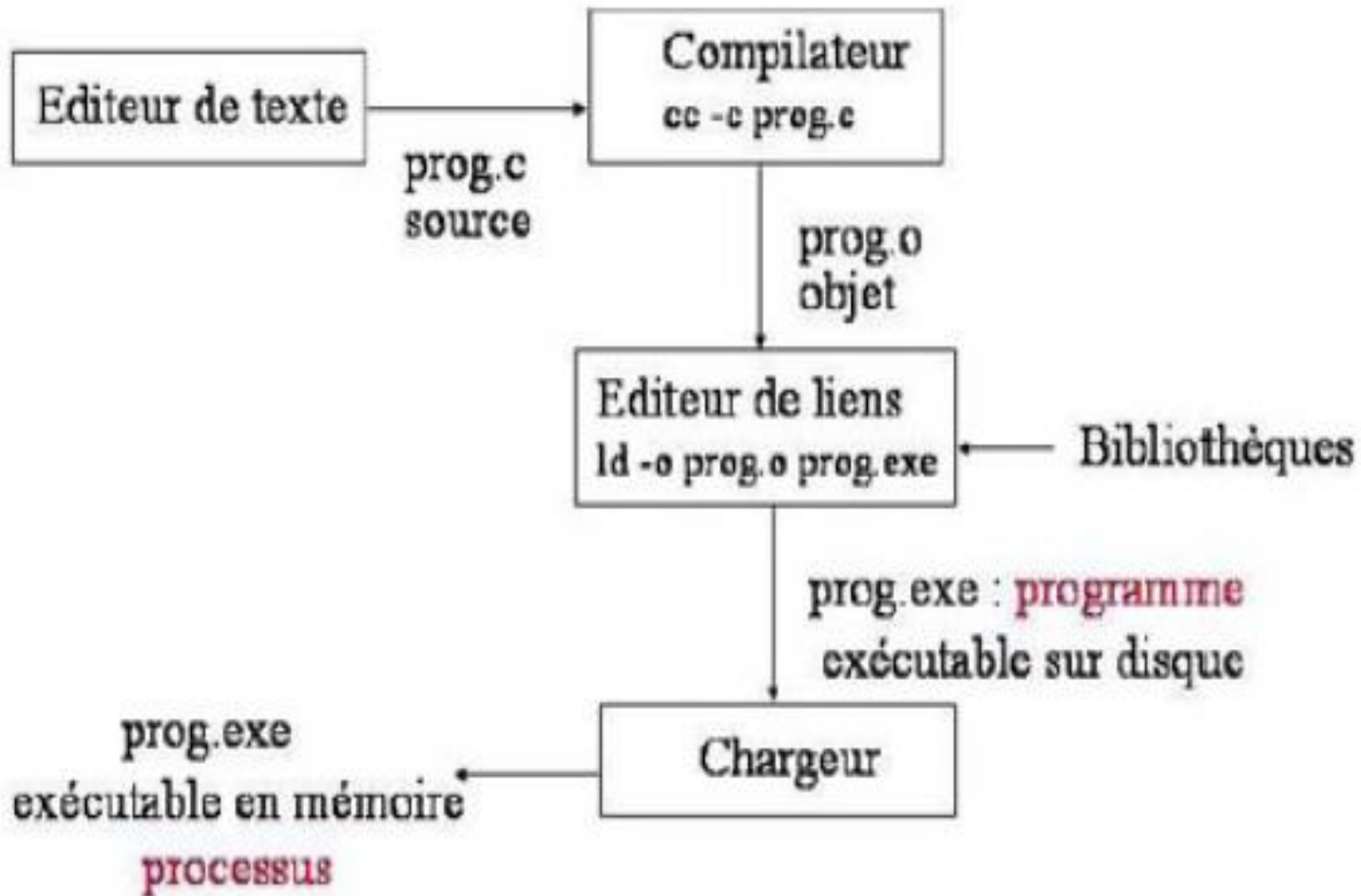
GESTION DES PROCESSUS

M.BOUABID, Avril 2017

PRODUCTION D'UN PROGRAMME EXÉCUTABLE(1/2)

- Ecriture du programme dans un langage haut niveau (programme écrit en langage c par exemple)
 - Fichier source
- Compilation : générer l'équivalent du programme source en langage machine
 - Fichier objet
- Éditions des liens : résoudre les références externes
 - Fichier exécutable
- Exécution du programme
 - Programme (code + données) est chargé en mémoire centrale

PRODUCTION D'UN PROGRAMME EXÉCUTABLE (2/2)



NOTION DE PROCESSUS (1/3)

- Un processus est un programme en cours d'exécution
- Dans un système multitâches , plusieurs processus s'exécutent « simultanément »
- Ils doivent donc se partager l'accès au processeur
- Plusieurs processus peuvent exécuter simultanément des copies (ou instances) d'un même programme.
- un processus est caractérisé par son contexte
 - valeur du compteur ordinal, contenu des différents registres, informations sur l'utilisation des ressources

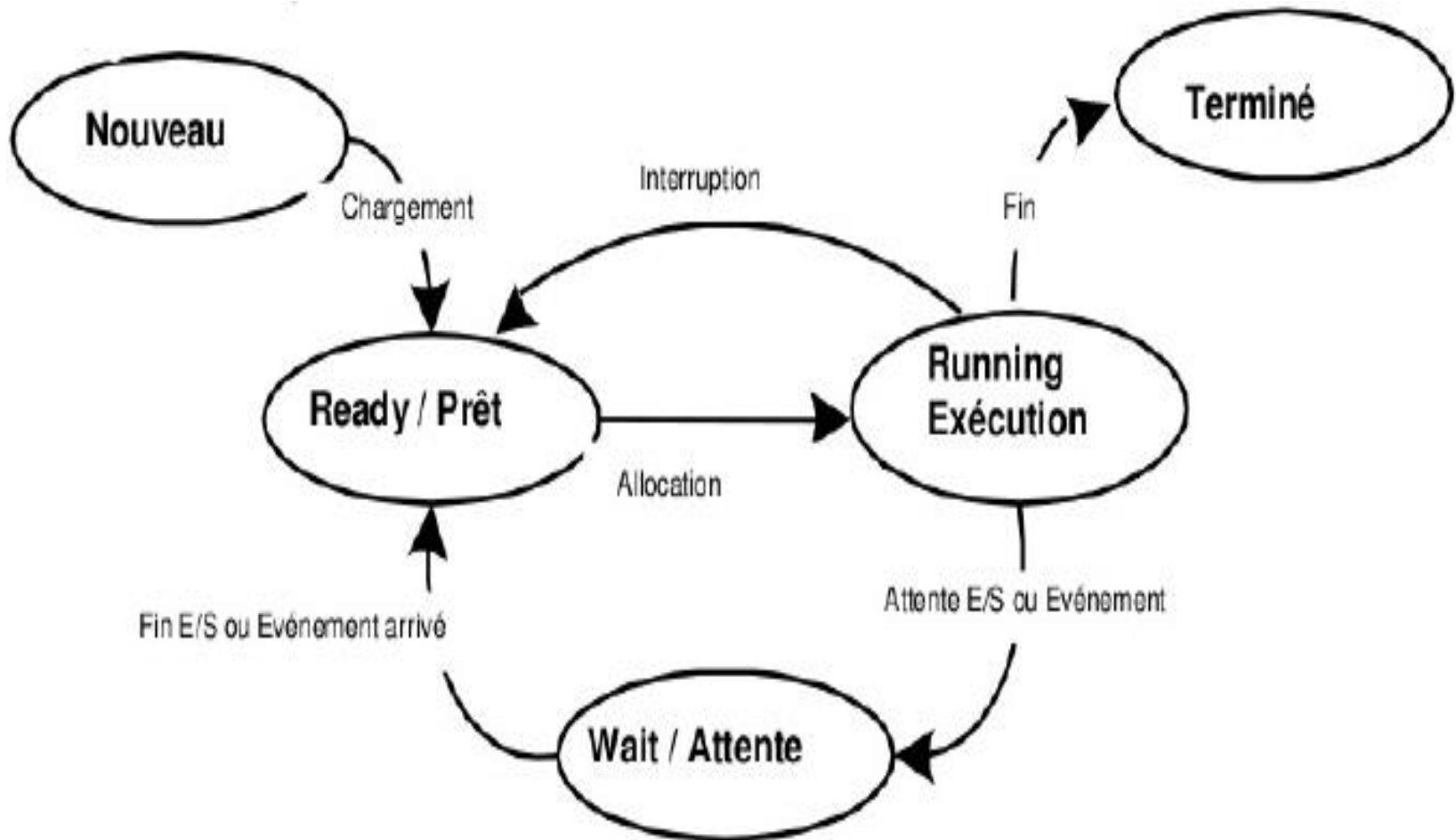
NOTION DE PROCESSUS (2/3)

- Pour cette raison le SE maintient une table de processus contenant des informations sur chaque processus
- Chaque entrée correspond à un processus et comprend généralement :
 - l'état du processus : prêt à être exécuté, en attente, suspendu,
 - des informations sur les ressources utilisées
 - des informations sur mémoire principale : table de pages,...
 - périphériques d'E/S : liste de fichiers ouverts...
 - files d'attente dans lesquelles le processus est inclus, etc...
 - et toutes les informations nécessaires pour assurer la reprise du processus en cas d'interruption

NOTION DE PROCESSUS (3/3)

- Plusieurs processus peuvent se trouver simultanément en cours d'exécution
- si un système informatique ne comporte qu'un seul processeur, alors, a un instant donné, un seul processus aura accès à ce processeur
- En conséquence, un programme en exécution peut avoir plusieurs états.
 - Etat **actif** ou élu (running): le processus utilise le processeur.
 - Etat **prêt** (ready: le processus pourrait utiliser le processeur si il était libre (et si c'était son tour).
 - Etat **en attente** ou bloqué (wait) : le processus attend une ressource

ETATS D'UN PROCESSUS



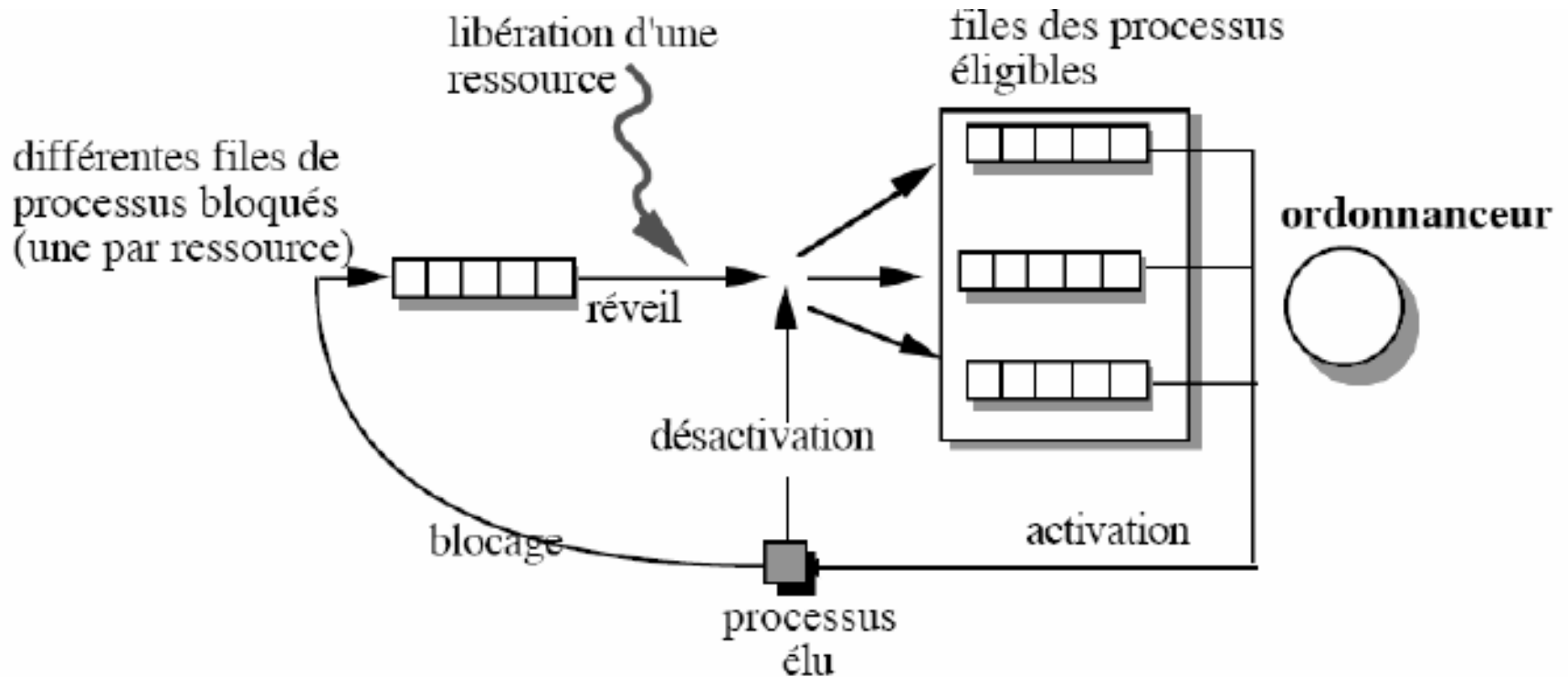
CRITÈRE D'ÉVALUATION DE PERFORMANCES

- Rendement
 - Nombre de travaux exécuter par unité de temps.
- Temps de service
 - Temps qui s'écoule entre le moment où un travail est soumis et où il est exécuté (temps d'accès mémoire + temps d'attente dans la file des processus éligibles + temps d'exécution dans l'unité centrale + temps d'attente + temps d'exécution dans les périphériques d'entrée/sortie).
- Temps d'attente
 - Temps passé dans la file des processus éligibles.
- Temps de réponse
 - Temps qui s'écoule entre la soumission d'une requête et la première réponse obtenue.

ORDONNANCEMENT (1/2)

- Le système d'exploitation gère les transitions entre les états
- Pour ce faire il maintient une liste de processus prêts et une liste de processus en attente
- Il doit également avoir une politique d'activation des processus prêts (ou politique d'allocation du processeur aux processus)
 - parmi les processus éligibles, le quel faut-il activer ?
- Le partage de CPU entre les différents processus en attente pour s'exécuter est appelé ordonnancement des processus.

ORDONNANCEMENT (2/2)



- But de l'ordonnancement : Equitabilité, Temps d'exécution, Rendement, Temps de réponse

ALGORITHMES D'ALLOCATION DU PROCESSEUR (1/3)

- *First Come First Serve (FCFS) : Premier Arrivé Premier Servi*

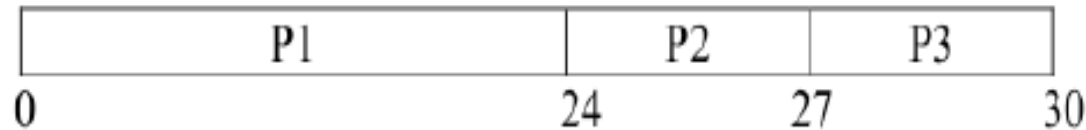
- Gestion des processus à l'aide d'un FIFO
- Ordonnancement non-préemptif
- Facile à implémenter
- Critère de choix : Ordre d'arrivée des processus

FCFS : EXEMPLE DE DÉROULEMENT

Exemple :

Processus	Temps d'exécution en ms
P1	24
P2	3
P3	3

Diagramme d'exécution si ordre d'arrivée P1, P2, P3 :



temps d'attente : P1 = 0, P2 = 24, P3 = 27 \Rightarrow temps moyen d'attente = $(24+27)/3 = 17$ ms.

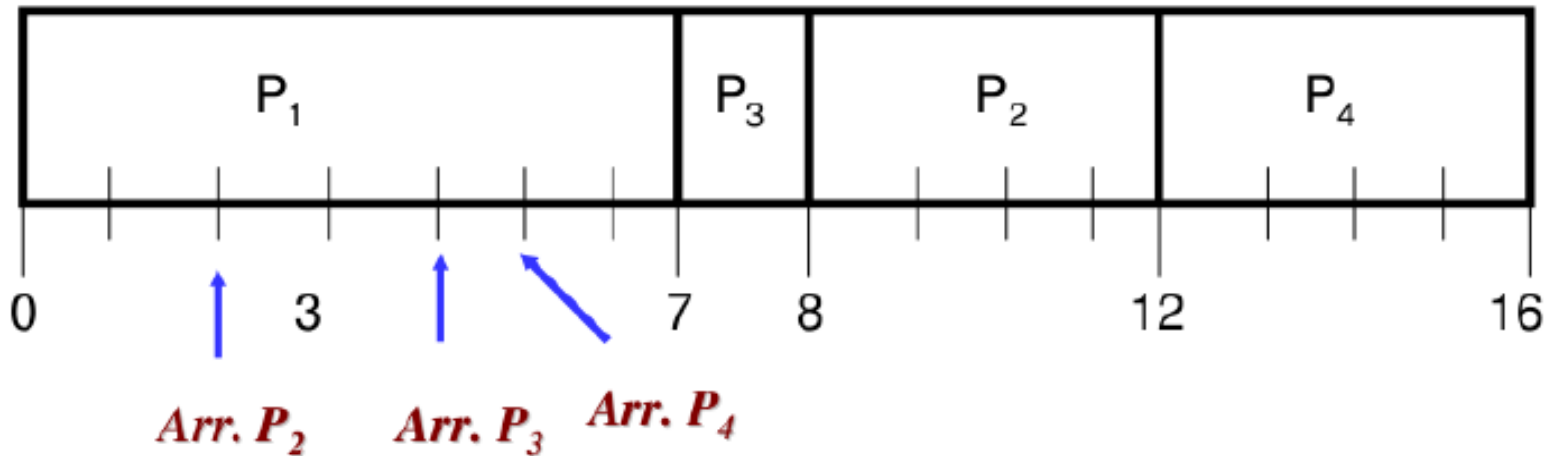
Si ordre P2, P3, P1 \Rightarrow temps d'attente P1 = 6, P2 = 0, P3 = 3 \Rightarrow temps moyen = $9/3 = 3$ ms.

ALGORITHMES D'ALLOCATION DU PROCESSEUR (2/3)

- ***Shortest Job First (SJF), Le Plus Court d'Abord***
 - On exécute d'abord les processus qui ont le temps d'exécution le plus court.
 - Avec ou sans préemption
 - Sans préemption : Sélection du prochain processus à la fin du processus courant
 - Avec préemption:
 - Sélection du prochain processus à l'arrivée d'un nouveau processus
 - Préemption ou non en comparant le temps restant du processus courant avec le temps du nouveau processus
 - Risque de famine (les processus de longue durée peuvent n'avoir jamais accès au processeur si des processus de courte durée arrivent en permanence).

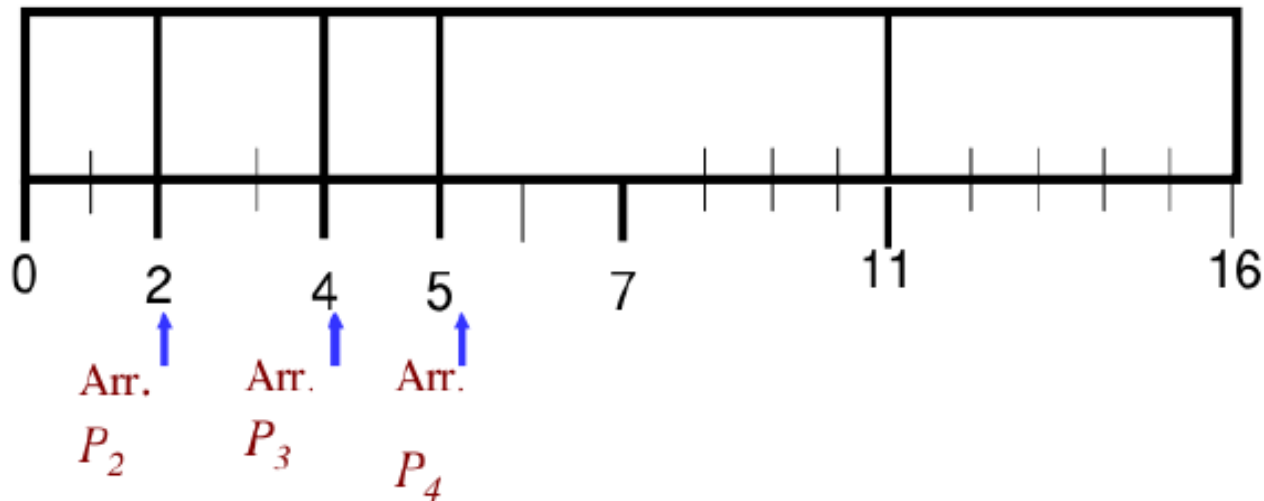
SGF : UN SCÉNARIO DE DÉROULEMENT SANS INTERRUPTION

<u>Processus</u>	<u>Arrivée</u>	<u>Temps d'exécution</u>
<i>P1</i>	0	7
<i>P2</i>	2	4
<i>P3</i>	4	1
<i>P4</i>	5	4



SGF : UN SCÉNARIO DE DÉROULEMENT AVEC INTERRUPTION

<u>Processus</u>	<u>Arrivée</u>	<u>Temps d'exécution</u>
<i>P1</i>	0	7
<i>P2</i>	2	4
<i>P3</i>	4	1
<i>P4</i>	5	4



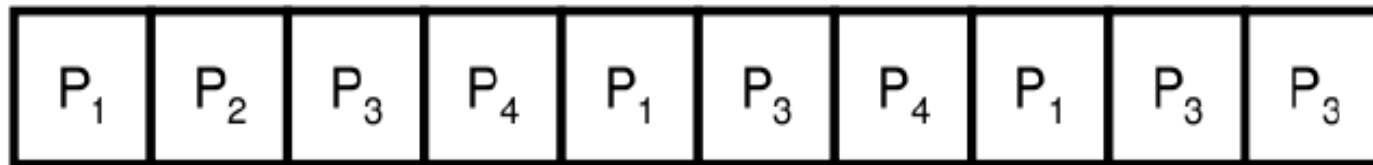
ALGORITHMES D'ALLOCATION DU PROCESSEUR (3/3)

○ *Round Robin ou Tourniquet*

- Processus organisés dans une file d'attente selon leur ordre d'arrivée .
- Utilisation de la notion de QUANTUM : *Unité de temps d'utilisation du processeur, Atomique – In interruptible – Indivisible*
- Principe :
 - Allocation par multiple de Quantums
 - Fin du quantum :
 - P_i terminé -> Quitte la file d'attente
 - P_i non terminé → Revient à la fin de la file, Allocation du processeur au processus P_{i+1}
 - Fin processus < Fin Quantum :
 - Allocation du processeur au processus P_{i+1}
- Ordonnancement préemptif

ROUND ROBIN : UN PREMIER EXEMPLE DE DÉROULEMENT

<u>Processus</u>	<u>Burst Time</u>
P_1	53
P_2	17
P_3	68
P_4	24



0 20 37 57 77 97 117 121 134 154 162

ROUND ROBIN : UN DEUXIÈME EXEMPLE DE DÉROULEMENT

Processus	Durée
P ₁	24
P ₂	3
P ₃	5

