Les processus dans les systèmes d'exploitation

INF34207 – Séance du

24 janvier 2024

Lise Boudreault, chargée de cours

Plan du cours:

UQAR
Rimouski | Lévis

1. Le processus
2. État d’un processus
3. Cycle d’exécution d’un processus
4. Blocs de contrôle des processus (BCP)
5. Routine d’ordonnancement des processus

Le processus séquentiel et/ou itératif

Entrée→Traitement → Sortie

Source: https://fr.dreamstime.com/photo-stock-entr%C3%A9e-processus-r%C3%A9sultats-image43936667

1. Les processus

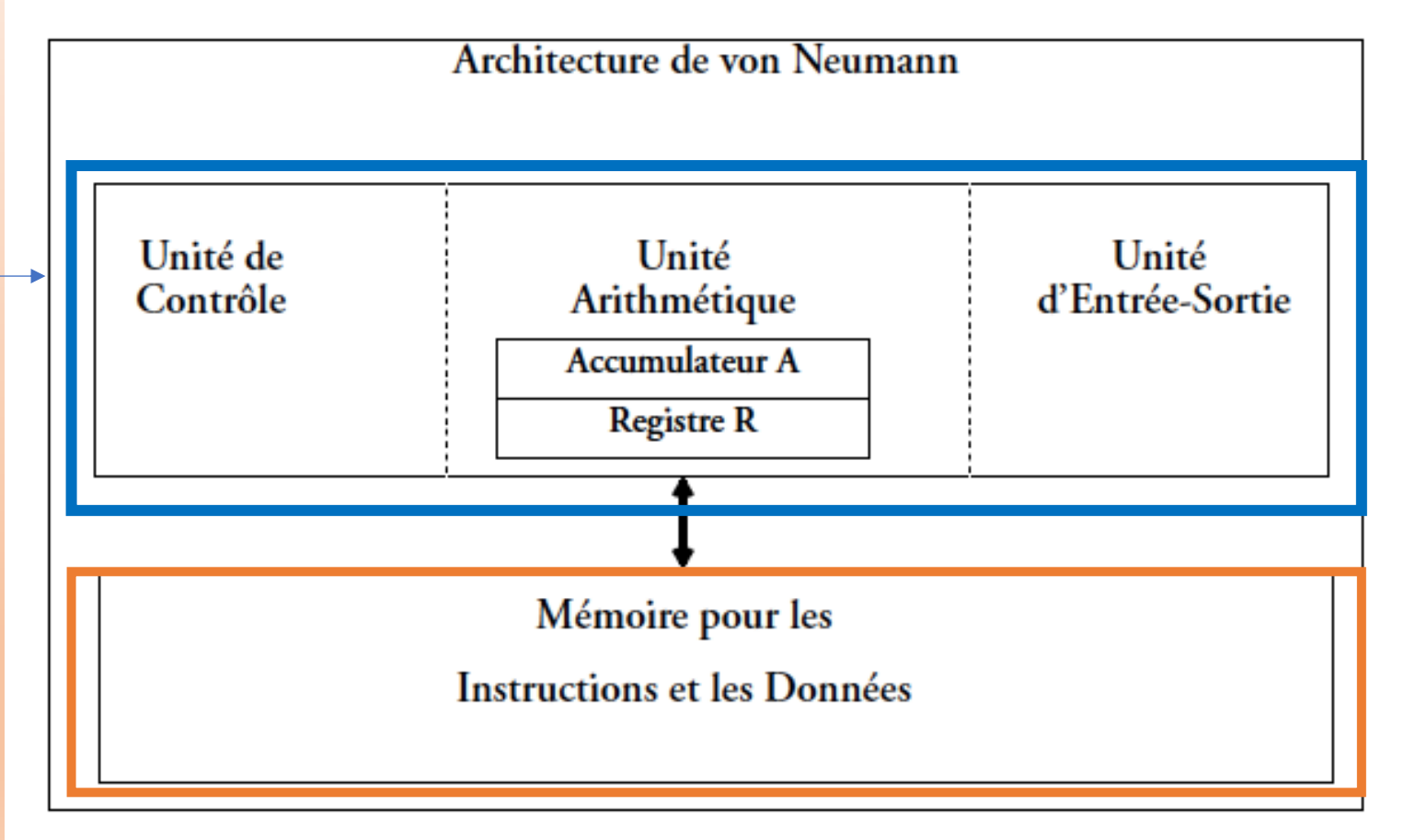
Le concept du processus n’est pas exclusif au domaine de l’informatique, elle existe dans plusieurs autres domaines comme la gestion: processus d’affaires, processus de gestion des achats, etc.

Toutefois, peu importe le domaine d’application, le concept du processus implique un ensemble d’actions et d’activités à traiter (exécuter)

1. Le processeur =
la ressource la plus importante d'une machine

UQAR
Rimouski | Lévis

Processeur/
Microprocesseur
(3 composantes)



Source: Bloch, L. p. 23

1. Les processus en informatique

UQAR
Rimouski | Lévis

- Un processus est un programme **en cours d'exécution** avec ses propres ressources (zone mémoire, données, etc).
- Le système d'exploitation manipule deux types de processus:
  - **Processus système:** processus lancé par le système, et
  - **Processus utilisateur:** processus lancé par l'utilisateur.

Source: Marwa Attafi, chargé de cours, hiver 2023

1. Les processus en informatique

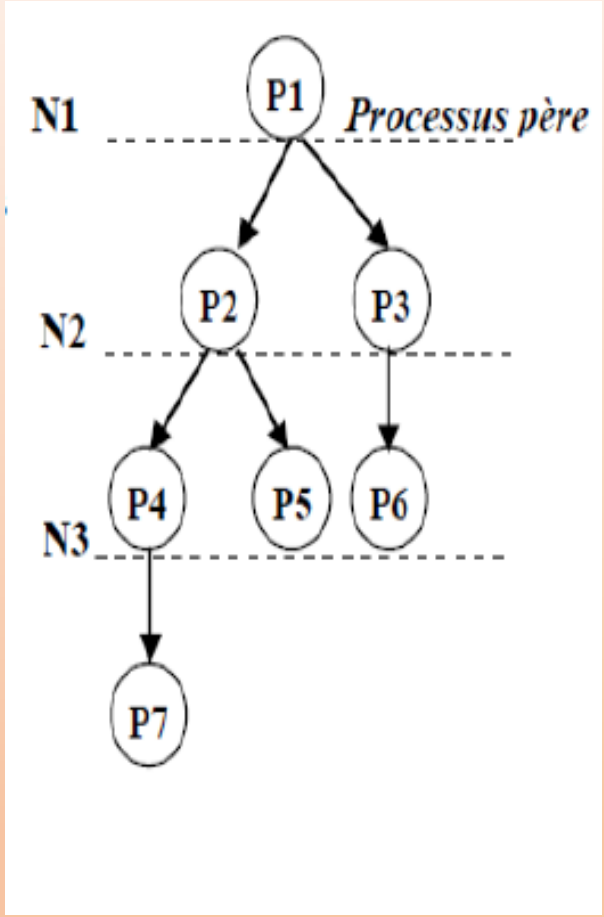
UQAR
Rimouski | Lévis

Dès sa création, un processus reçoit les paramètres suivants:

- **PID:** identificateur du processus (numéro unique),
- **PPID:** identificateur du processus père,
- **UID:** identificateur de l'utilisateur qui a lancé le processus,
- **GID:** identificateur du groupe de l'utilisateur qui a lancé le processus.

Source: Marwa Attafi, chargé de cours, hiver 2023

1. Les processus en informatique



Source: Marwa Attafi, chargé de cours, hiver 2023

1. Les processus en informatique

UQAR
Rimouski | Lévis

Un processus **(père)** peut créer d'autres processus **(fils)** qui héritent les descripteurs de son père. Ce dernier à son tour peut créer d'autres processus. Un processus a un seul père mais peut avoir plusieurs fils.

Source: Marwa Attafi, chargé de cours, hiver 2023

1. Les processus en informatique

UQAR
Rimouski | Lévis

Les processus peuvent **se terminer** ou ils peuvent **être éliminés** par d'autres processus (exemple: ***kill*** en Unix). A la destruction d'un processus, on **libère** toutes les ressources qu'il avait.

1. Les processus en informatique

UQAR
Rimouski | Lévis

Dans certains cas, la destruction d'un processus entraîne l'élimination de ses descendants. Cette opération n'affecte pas les processus qui peuvent continuer indépendamment de leur père (**processus orphelins**).

2. État d’un processus

UQAR
Rimouski | Lévis

Dans les systèmes **mono-programmés (mono-tâches)**, un programme ne quitte pas le processeur avant de terminer son exécution. Pendant cette période, il dispose de **toutes les ressources** de la machine.

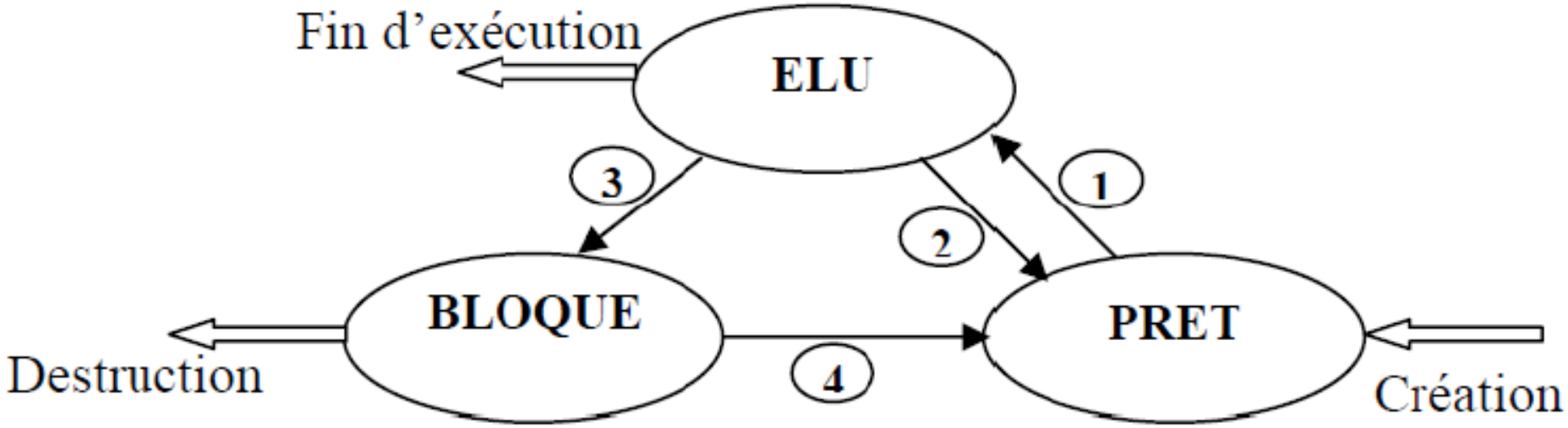
2. État d’un processus

UQAR
Rimouski | Lévis

Par contre, dans les systèmes **multi-programmés (multi-tâches)**, un processus peut se trouver dans l’un des états suivants:

- **Élu:** si le processus est en cours d'exécution;
- **Bloqué:** si le processus est en attente d’un événement à se produire ou bien d’une ressource à se libérer pour pouvoir continuer;
- **Prêt:** si le processus dispose de toutes les ressources nécessaires à son exécution à l'exception du processeur.

2. État d'un processus



(1): Allocation du processeur au processus sélectionné;

(2): Réquisition du processeur après expiration de la tranche du temps par exemple;

(3): Blocage du processus élu dans l'attente d'un événement;

(4): Réveil du processus bloqué après disponibilité de l'événement bloquant.

Figure 1. Diagramme de transition du processus entre ses différents états.

3. Cycle d'exécution d'un processus

UQAR
Rimouski | Lévis

L'exécution d'un processus peut être vue comme une séquence de phases. Chaque phase comprend deux cycles:

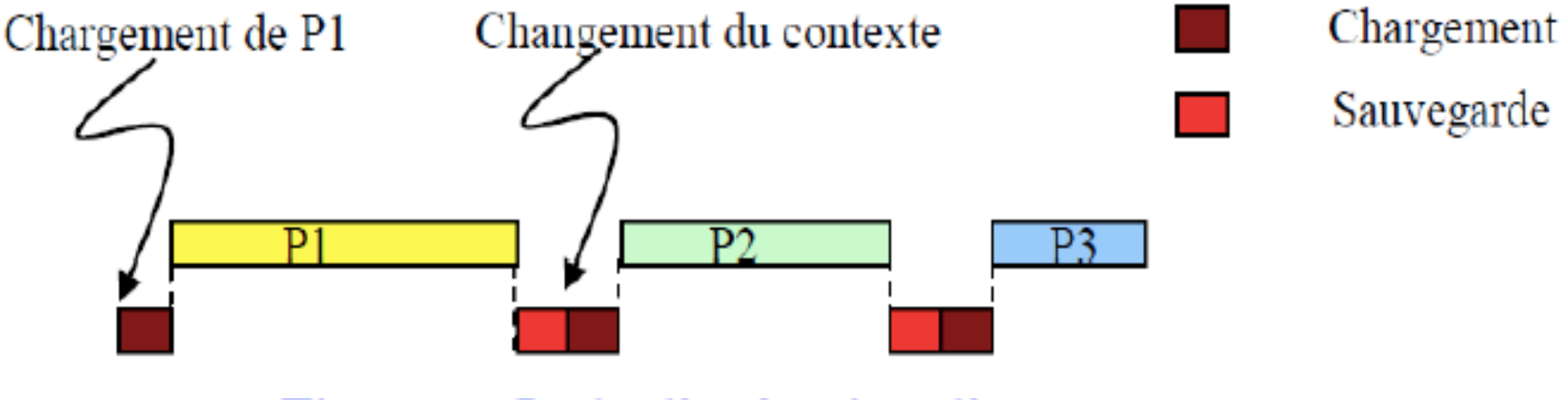
- un cycle d'exécution (ou calcul) réalisé par le processeur, et
- un cycle d'entrée/sortie assuré par le canal.

→ Phase 1 → Phase 2

3. Cycle d'exécution d'un processus

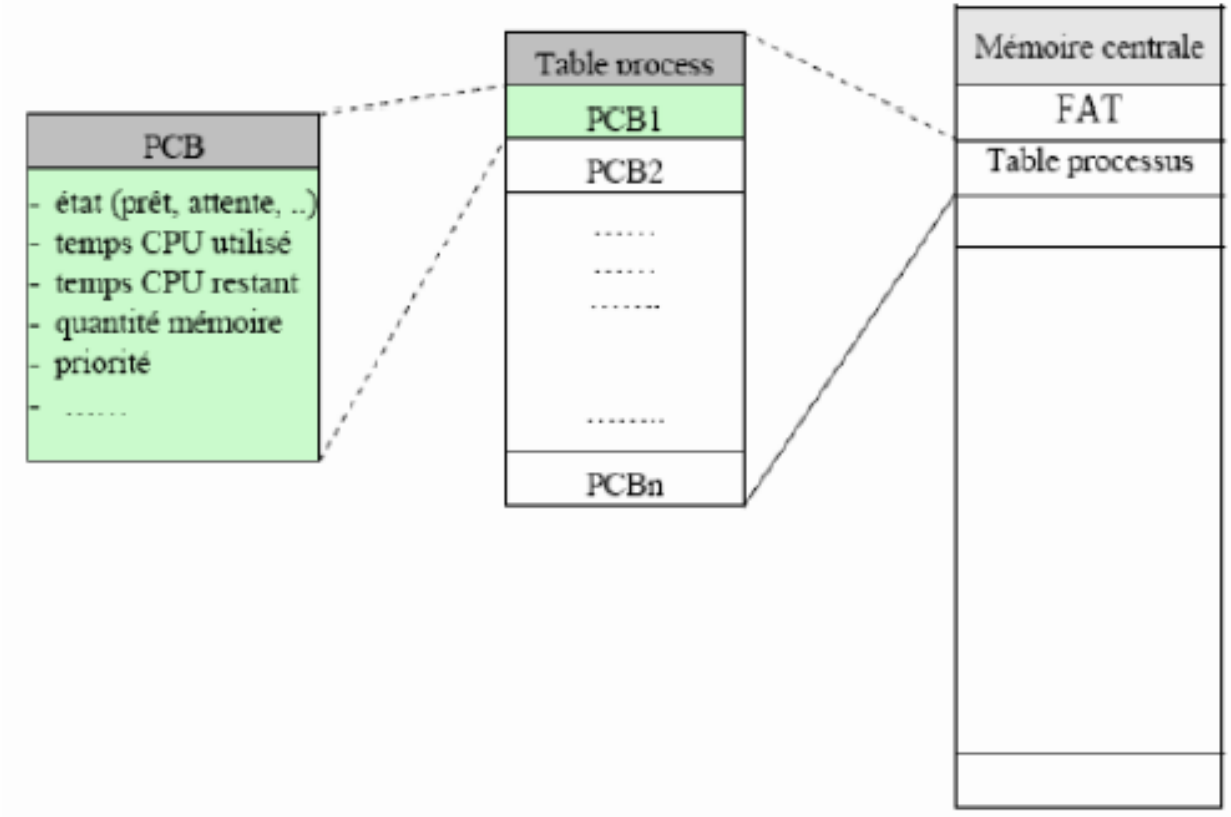
UQAR
Rimouski | Lévis

La dernière phase de tout processus doit comprendre obligatoirement un seul cycle dans lequel sera exécutée la requête informant le système d'exploitation sur la terminaison du processus. Cet appel permet au système de restituer les ressources utilisées par le processus qui vient de terminer.



4. Blocs de contrôle des processus (BCP)

Les BCP sont rangés dans une table (table des processus) qui se trouve dans l'espace mémoire du système.



5. Routine d'ordonnancement des processus

UQAR
Rimouski | Lévis

Chaque fois, que le processeur devient inactif, le système d'exploitation doit sélectionner un processus de la file d'attente des processus prêts, et lui passe le contrôle. D'une manière plus concrète, cette tâche est prise en charge par deux routines système:

- **le Répartiteur**
- **l'Ordonnanceur**.

5. Rôle du répartiteur/ordonnanceur

L’ordonnanceur détermine dans quelle séquence les processus seront exécutés

Le répartiteur, alloue le processeur au processus sélectionné

5. Routine d'ordonnancement des processus

UQAR
Rimouski | Lévis

L'ordonnancement est la partie du système d'exploitation qui détermine **dans quel ordre** les processus **prêts** à s'exécuter (présents dans la file des prêts) seront **élus**.

Ses objectifs sont:

- Assurer le plein usage du CPU (agir en sorte qu'il soit le moins souvent possible inactifs);
- Réduire le temps d'attente des utilisateurs;
- Assurer l'équité entre les utilisateurs.

5. Routine d'ordonnancement des processus

Il existe plusieurs algorithmes d'ordonnancement tels que:

- Premier Arrivé Premier Servi,
- Shortest Job First (Le pus court d'abord),
- Round & Robin (Tourniquet),
- Etc.

5. Routine d'ordonnancement des processus

UQAR
Rimouski | Lévis

L'algorithme FCFS (First Come First Served):

- Cet algorithme est connu aussi sous le nom FIFO (First In First Out) ou encore Premier Arrivé Premier Servi;
- Les processus sont rangés dans la file d'attente des processus prêts selon leur ordre d'arrivée. Les règles régissant cet ordonnancement sont:
  - Quand un processus est prêt à s'exécuter, il est mis en queue de la file d'attente des processus prêts;
  - Quand le processeur devient libre, il est alloué au processus se trouvant en tête de la file d'attente des processus prêts;
  - Le processus élu relâche le processeur s'il se termine ou s'il demande une entrée sortie.

5. Routine d'ordonnancement First Come First Served

UQAR
Rimouski | Lévis

- **Exemple d'application du FCFS**

Processus	Durée d'exécution	Date d'arrivé
P1	24	0
P2	3	1
P3	3	2

Diagramme de Gantt:

- Ce diagramme montre que le processus P1 occupe le processeur de l'instant **0** jusqu'à l'instant **24**. A ce dernier, le processeur devient occupé par le processus P2, puis à l'instant **27** il sera suivi du processus P3.

5. Routine d'ordonnancement First Come First Served

Exemple d'application du FCFS

Processus	Durée d'exécution	Date d'arrivé
P1	24	0
P2	3	1
P3	3	2

Critères d'évaluation de la performance:

- Le **TRM (Temps de Réponse Moyen)** qui décrit la moyenne des dates de fin d'exécution:

$$TRM = \sum_{i=1}^{n} TRi \,/\, n \text{ , avec } TR_i = \text{date fin} - \text{date arrivée.}$$

- Le **TAM (Temps d'Attente Moyen)** qui décrit la moyenne des délais d'attente pour commencer une exécution:

$$TAM = \sum_{i=1}^{n} TAi \,/\, n \text{ , avec } TA_i = TR_i - \text{temps d'exécution}$$

UQAR
Rimouski | Lévis

5. Routine d'ordonnancement First Come First Served

UQAR
Rimouski | Lévis

- **Exemple d'application du FCFS**

Processus	Durée d'exécution	Date d'arrivé
P1	24	0
P2	3	1
P3	3	2

- Critères d'évaluation de la performance:

$$TRM = \sum_{i=1}^{n} TRi \,/\, n$$, avec TR_i=date fin –date arrivée.

TRM = [(24 - 0) + (27 – 1) + (30 – 2)] / 3 = **26**

$$TAM = \sum_{i=1}^{n} TAi \,/\, n$$, avec TA_i =TR_i - temps d'exécution

TAM = [(24 - 24) + (26 – 3) + (28 – 3)] / 3 = 48/3 = **16**

5. Routine d'ordonnancement First Come First Served

UQAR
Rimouski | Lévis

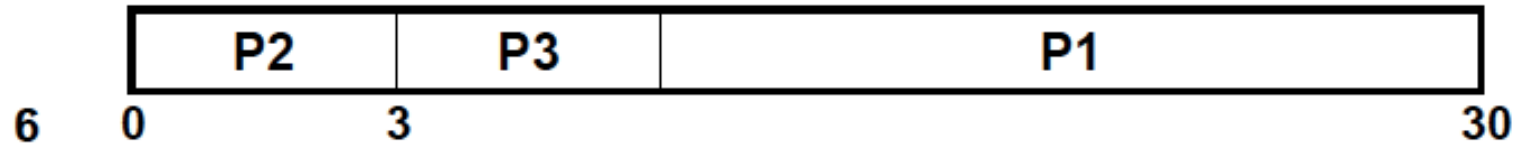
- **Exemple d'application du FCFS**

Supposant maintenant que les processus étaient arrivés dans l'ordre P2, P3 et P1:

Processus	Durée d'exécution	Date d'arrivé
P2	3	0
P3	3	1
P1	24	2

Dessiner le diagramme de Gantt puis calculer TRM et TAM?

Diagramme de Gantt:



TRM =((3-0)+(6-1)+(30-2))/3=12 ; TAM = ((3-3)+(5-3)+(28-24))/3=2

5. Routine d'ordonnancement First Come First Served

UQAR
Rimouski | Lévis

Critiques du FCFS:

- Le temps moyen d'attente avec une politique FCFS n'est généralement **pas minimal** et peut varier substantiellement si les durées d'exécution des processus sont **assez variées**.

5. Routine d'ordonnancement First Come First Served

UQAR
Rimouski | Lévis

Critiques du FCFS:

L'algorithme FCFS est particulièrement **incommode** pour les systèmes **à temps partagé**, où il est important que l'utilisateur obtienne le processeur à des intervalles réguliers. Il peut paraître désastreux de permettre qu'un processus garde le processeur pendant une période **étendue**.

Je vous remercie de votre attention

UQAR
Rimouski | Lévis

Rappel: Travail #1 remise 14 février 23h59

Préparation pour la semaine prochaine:

Système d'exploitation et gestion de la mémoire

-Lecture recommandée:

Laurent Bloch, chapitre 4

Bonne semaine!