Université Yahia Farès de Médéa
Département: GEI
2ème année Informatique (L.M.D)
Option : RSI/SIIA

Gestion du Processeur central

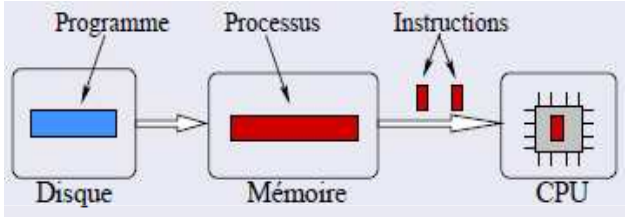
A. KHELDOUN
Émail: univ.medea@gmail.com

Année Universitaire: 2013-2014

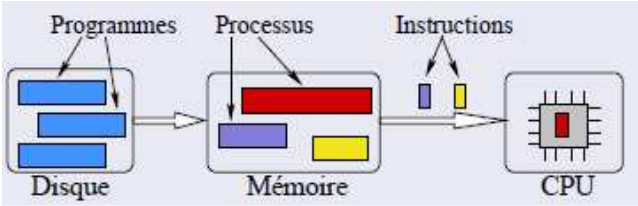
Problématique

Exécution de processus

Monoprogrammation



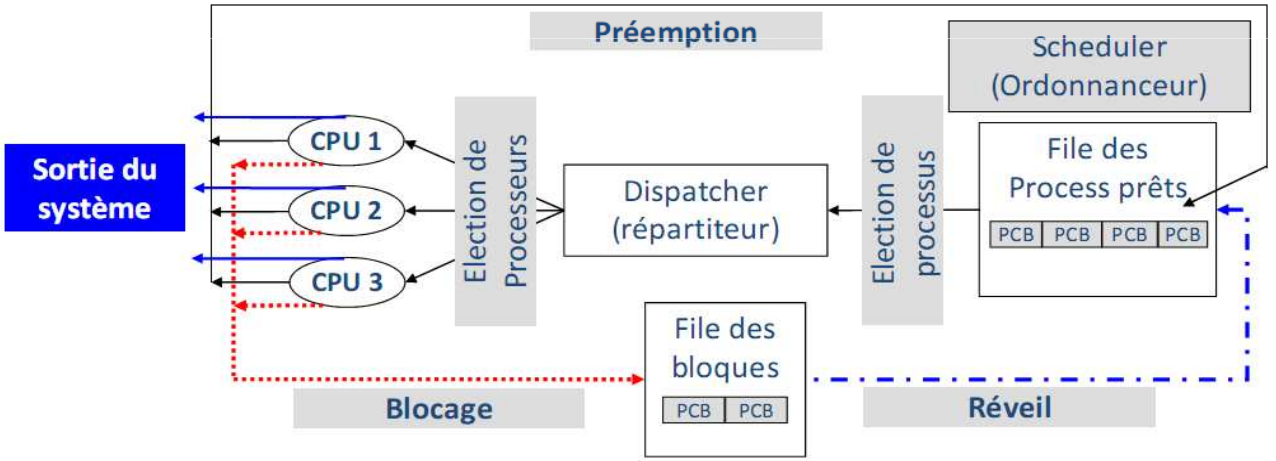
Multiprogrammation



- Cas d'un système multiprogramme
  - ✓ Plusieurs processus en mémoire, un seul à la fois accède au CPU;
  - ✓ Comment choisir le processus qui a accès au CPU? Comment Changer cet accès au cours de l'exécution ?

Scheduling/Scheduler

C'est un module (programme) du système d'exploitation qui attribue le contrôle de l'unité centrale, à tour de rôles, aux différents processus en compétition suivant une politique définie à l'avance par les concepteurs du système.



Objectifs de scheduling

- **L'équité**
  - ✓Le scheduler doit servir les programmes de même priorité d'une manière juste et équitable.

- **Le rendement**
  - ✓Désigne le taux d'occupation de l'unité centrale (i.e. Le nombre de travaux réalisés par unité de temps).
  - ✓Il minimise le temps de réponse des travaux.

- **L'utilisation des ressources**
  - ✓Assurer une occupation maximale des ressources de la machine
  - ✓ Minimiser le temps d'attente pour l'allocation d'une ressource à un processus.

Critères de scheduling

- **La disponibilité des ressources**
  - ✓Planifier les exécutions des programmes selon les ressources demandées et celles qui existent.
- **La classe des programmes**
  - ✓Programmes orientés calcul ou E/S.
  - ✓Programmes interactifs
  - ✓Programmes temps réel
- **Scheduling avec ou sans préemption**
  - ✓**Non prémption**: une fois le processeur central est alloué à un processus ; il le gardera jusqu'à sa terminaison.
- **Scheduling avec ou sans priorité**
  - ✓***Priorité statique*** : Une fois allouée à un processus, elle restera inchangée jusqu'à sa terminaison.
  - ✓***Priorité dynamique*** **:** Elle change en fonction de l'environnement d'exécution du processus (ex : ancienneté, …).

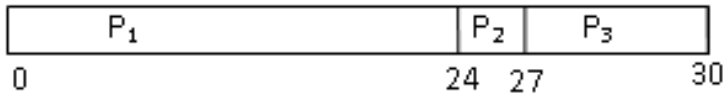
Politiques (Algorithmes) de scheduling

Politiques de schudeling sans préemption

- ***Temps d'attente*** : Temps passé à attendre dans la file d'attente des processus prêts.
- ***Temps de réponse*** ($T=T_{Fin} - T_{Arrivé}$) où:
  - ✓T_{Fin} est le temps de fin d'exécution du processus
  - ✓$T_{Arrivé}$ est son temps d'arrivé

Premier arrive, premier servi (*FIFO;FCFS*)

- ✓ La queue (file d'attente) des processus prêts gérée en FIFO
- ✓ Exemple (***03 processus P_1, P_2 et P_3***):
  - ➢ Ordre d'arrivé (t=0)
  - ➢ Durées d'exécution sont respectivement : 24, 3, 3 ms
- ✓ ***Diagramme de Gantt***

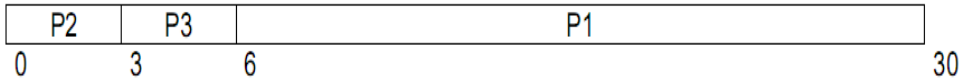


- ✓ Temps d'attente: $P_1 = 0$ ms; $P_2 = 24$ ms; $P_3 = 27$ ms
- ✓ Temps d'attente moyen: (0 + 24 + 27) / 3 = 17 ms

Politiques (Algorithmes) de scheduling

Politiques de schudeling sans préemption

- **Premier arrive, premier servi (*FIFO;FCFS*)**
  - ✓ Exemple (***03 processus P_1, P_2 et P_3***):
    - ➢ Ordre d'arrivée: P_2 , P_3 , P_1.
  - ✓ ***Diagramme de Gantt***



  - ✓ Temps d'attente: P_1 = 6 ms; P_2 = 0 ms; P_3 = 3 ms
  - ✓ Temps d'attente moyen: (6 + 0 + 3) / 3 = 3 ms

  - ✓ ***Critique de la méthode***

    La politique FIFO **désavantage les processus courts** s'ils ne sont pas exécutés en premier.

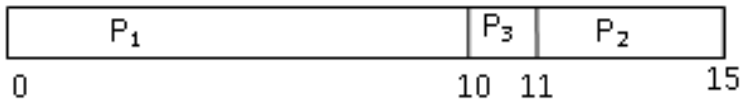
Politiques (Algorithmes) de scheduling

Politiques de schudeling sans préemption

- **Algorithme du Plus Court d'abord(*SJF*)**
  - ✓ Affecter le processeur au processus possédant le temps d'exécution le plus court
  - ✓Exemple :

Processus	Temps d'arrivé	Temps d'exécution
P1	0	10
P2	3	4
P3	5	1

  - ✓ ***Diagramme de Gantt***



  - ✓Temps d'attente moyen: (0 + 8 + 5)/3 = 4,33 ms
  - ✓ Temps d'attente moyen (FCFS): (0+7+9) = 5,33 ms
  - ✓ ***Critique de la méthode***
    - ➢ Avantage les processus les plus courts→risque de privation
    - ➢ Comment connaître le temps d'exécution d'un processus à l'avance ?

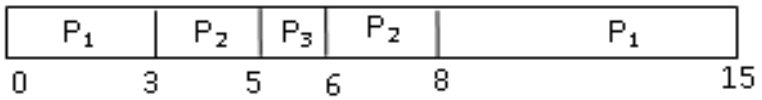
Politiques (Algorithmes) de scheduling

Politiques de schudeling avec préemption

- **Le plus court temps restant le premier (*SRTF*)**
  - ✓ SJF + mécanisme de réquisition.
    - ➢ Mêmes contraintes que SJF.
    - ➢ On choisit le processus qui a le temps restant d'exécution le plus petit.
    - ➢ L'arrivée d'un nouveau processus peut interrompre le processus courant.
  - ✓ Exemple

Processus	Temps d'arrivé	Temps d'exécution
P1	0	10
P2	3	4
P3	5	1

  - ✓ ***Diagramme de Gantt***



  - ✓ Temps d'attente moyen: (5 + 1 + 0)/3 = 2 ms

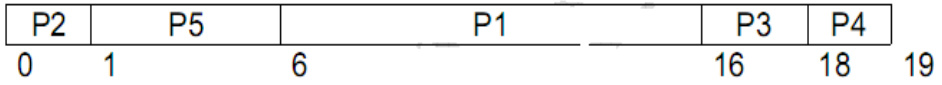
Politiques (Algorithmes) de scheduling

Politiques de schudeling avec préemption

- **Scheduling avec priorité**
  - ✓ Associer à chaque processus une priorité (un entier).
  - ✓ Affecter le processeur au processus de plus haute priorité.
  - ✓Exemple

Processus	Temps d'exécution	Priorité
P1	10	2
P2	1	4
P3	2	2
P4	1	1
P5	5	3

  - ✓ ***Diagramme de Gantt***



  - ✓Temps d'attente moyen: (0+1+6+16+18)/5=8.2 ms
  - ✓ ***Critique de la méthode***

    Risque de privation.

Politiques (Algorithmes) de scheduling

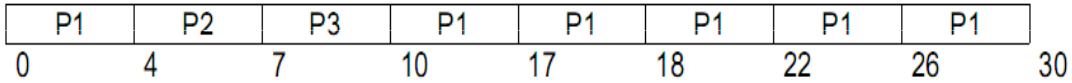
Politiques de schudeling avec préemption

- **Round Robin (*Tourniquet*)**
  - ✓ FIFO + mécanisme de réquisition (durée Quantum)
    - ➢ le temps d'allocation du processeur est découpé en tranches, nommées quantum (typiquement 10 ms – 100 ms)
    - ➢ L'arrivé d'un nouveau processus prêt est inséré à la fin de la file d'attente des processus prêts.
    - ➢ À la fin de son quantum, le processus élu est interrompu,
      a. Insertion de processus courant le fin de la file d'attente
      b. Allocation de processeur au prochain processus (FIFO)

Politiques (Algorithmes) de scheduling

Politiques de schudeling avec préemption

- **Round Robin (*Tourniquet*)**
  - ✓Exemple (***03 processus P_1, P_2 et P_3***):
    - ➢ Ordre d'arrivé (t=0)
    - ➢ Durées d'exécution sont respectivement : 24, 3, 3 ms
  - ✓***Diagramme de Gantt***



  - ✓Temps d'attente moyen: (6+4+7)/3 = 17/3 = 5.66 ms
  - ✓***Remarque***

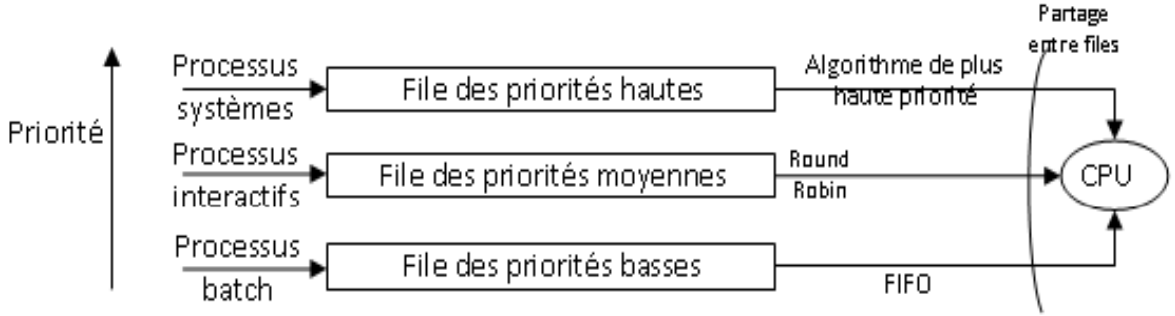
    Attention au choix du quantum relativement au coût du dispatcher.

✓***Exercice :*** On dispose d'un processus P dont le temps d'exécution est de 10 ms. Calculons le nombre de commutations de contexte nécessaires pour un quantum égal respectivement à : 12, 6 et 1.

Politiques (Algorithmes) de scheduling

Politiques de schudeling avec préemption

- **Scheduling avec files d'attente multi-niveaux**
  - ✓ Ordonnanceur gère plusieurs politiques (c-à-d plusieurs queues)

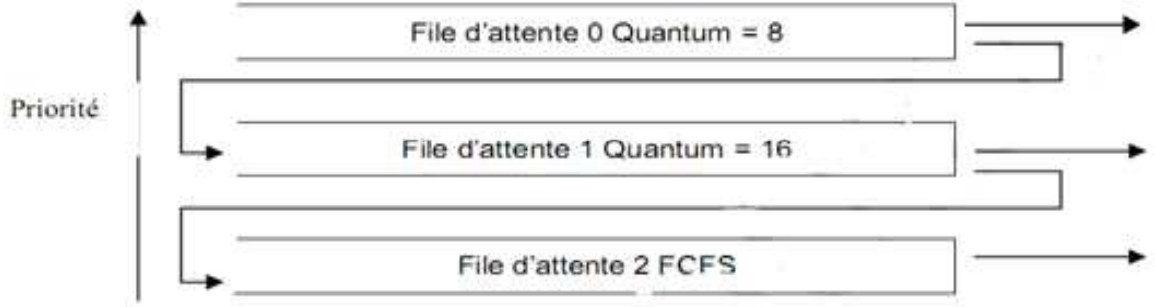


  - ✓ Allocation du CPU au processus de la queue j uniquement si la queue i (i<j) est **vide**.

Politiques (Algorithmes) de scheduling

Politiques de schudeling avec préemption

- **Scheduling avec files d'attente multi-niveaux et feedback**
  - ✓ Ordonnanceur gère plusieurs politiques (c-à-d plusieurs queues)
  - ✓ les processus peuvent passer d'une queue à une autre



  - ✓ Remarque
    Avantager les processus courts

Ordonnancement linux

- **Norme POSIX: trois types de politiques :**
  - ✓ SCHED_FIFO : l'algo à priorités fixes non préemptif.
    → threads temps réel non préemptibles
  - ✓ SCHED_RR : l'algo de tourniquet
    → thread temps réel préemptibles
  - ✓SCHED_OTHER : ordonnancement en temps partagé
    → threads ordinaires (non temps réel)

- **Remarque**

  Pas de temps réel strict en Linux, extensions RTLinux (ordonnancement préemptif avec priorités fixes)