
Cours 3

Ordonnancement de tâches/processus

15/09/2024

Cours 3 : Ordonnancement- DANT

1

1

Définitions

■ Programme

- Expression d'un algorithme à l'aide d'un langage de programmation.

■ Tâche

- Conjonction d'un programme et des données auxquelles il s'applique.

■ Processus

- Un processus est l'exécution d'une tâche. Il est caractérisé par la tâche et son contexte d'exécution
 - contexte d'exécution:
 - ❑ valeur du compteur ordinal, les valeurs des registres, etc.

15/09/2024

Cours 3 : Ordonnancement- DANT

2

2

Définitions (cont).

■ Monoprogramation

Il y a un seul processus à la fois en mémoire.

- Lorsqu'une tâche est soumise et que le processeur est disponible, on la charge en mémoire puis on exécute le processus associé jusqu'à ce qu'il soit terminé. On passe alors à la tâche suivante.

■ Multiprogrammation

- Il peut y avoir plusieurs processus à la fois en mémoire. Une tâche soumise est chargée en mémoire s'il y a de la place et donne naissance à un processus.
- La multiprogrammation peut être utilisée en batch ou en temps partagé.

15/09/2024

Cours 3 : Ordonnancement- DANT

3

3

Ordonnancement de Processus

■ L'objectif de la multiprogrammation est de maximiser l'utilisation de la CPU

- Plusieurs processus en mémoire
- Lorsqu'un processus doit attendre une ressource, le système d'exploitation retire la CPU à ce processus et l'alloue à un autre.

■ Ordonnanceur

- Chaque fois que la CPU est oisive, le ordonnanceur doit choisir un des processus dans la file d'attente des processus prêts en mémoire. La CPU est affectée à ce processus

15/09/2024

Cours 3 : Ordonnancement- DANT

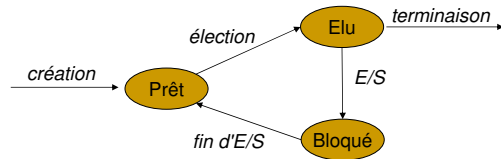
4

4

Multiprogrammation batch

■ Le processus actif rend la main

- lorsqu'il se termine
- lorsqu'il se bloque en attente d'une E/S



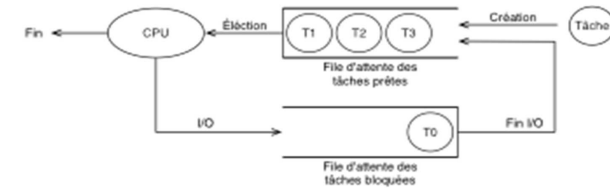
15/09/2024

Cours 3 : Ordonnancement- DANT

5

5

Multiprogrammation batch



15/09/2024

Cours 3 : Ordonnancement- DANT

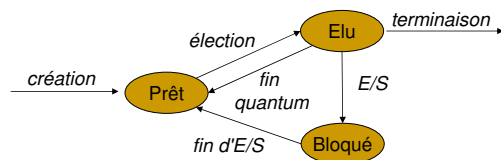
6

6

Multiprogrammation temps partagé

■ Le processus actif rend la main :

- lorsqu'il se termine
- lorsqu'il se bloque en attente d'une E/S
- lorsqu'il a épuisé son quantum de temps
 - Un quantum est une durée élémentaire (ordre de 10 à 100 ms)



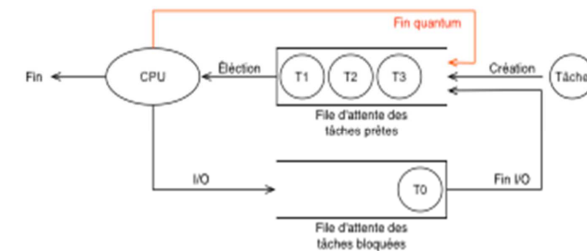
15/09/2024

Cours 3 : Ordonnancement- DANT

7

7

Multiprogrammation temps partagé



15/09/2024

Cours 3 : Ordonnancement- DANT

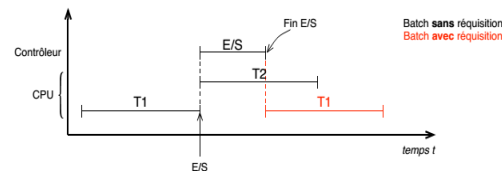
8

8

Requisition (préemption)

- Une tâche plus prioritaire peut prendre le processeur.

- La tâche élue est interrompue.
 - Remise en tête de la file des tâches prêtes.



T1 plus prioritaire que T2

15/09/2024

Cours 3 : Ordonnancement- DANT

9

9

Dispatcheur

- Le dispatcheur est le responsable pour donner le contrôle de la CPU au processus choisi par l'ordonnanceur. Cette tâche comprend:

- Commutation de contexte
- Passer en mode utilisateur
- Branchement au bon emplacement dans le programme élu pour le relancer

- Le dispatcheur doit être le plus rapide possible

- **Overhead du Dispatcheur**

- temps pris par le dispatcheur pour stopper un processus et en redémarrer un autre.

15/09/2024

Cours 3 : Ordonnancement- DANT

10

10

Algorithmes d'ordonnancement

- **Objectif**

- Suivre les critères de manière optimale
 - En particulier :
 - ❑ garantir l'équité
 - ❑ éviter les famines
 - Famine : une tâche n'est jamais élue

- **Un grand nombre d'algorithmes**

- *eg. FIFO, SJF, RR, EDF, ...*
 - Choix du bon algorithme dépend de l'utilisation du système (nombre de tâches, types, ...)

15/09/2024

Cours 3 : Ordonnancement- DANT

11

11

Critères d'ordonnancement

- **Critères :**

- *Utilisation de la CPU*
 - occuper la CPU le maximum possible
- *Débit (Throughput)*
 - Nombre de processus qui terminent leur exécution par unité de temps
- *Temps d'attente*
 - temps d'un processus dans la file d'attente des processus prêts
- *Temps de réponse*
 - temps entre une requête émise et la première réponse

- **But :**

- Maximiser l'utilisation et le débit de la CPU et minimiser les temps de commutation, attente et de réponse.

15/09/2024

Cours 3 : Ordonnancement- DANT

12

12

Décision d'ordonnancement

■ Les décisions d'ordonnancement de la CPU sont prises lors:

1. Du changement d'état "exécution" à "en attente"
Exemple : demande d'un E/S
2. Du changement d'état de "exécution" à "prêt"
Exemple: fin quantum
3. Du changement d'état de "en attente" à "prêt"
Exemple: fin d'une E/S
4. De la terminaison d'un processus

15/09/2024

Cours 3 : Ordonnancement- DANT

13

Algorithmes d'ordonnancement

■ Mode Batch

- Premier arrivé, premier servi
 - FCFS : First come first serviced
- Le travail le plus court d'abord
 - SJF: Shortest job first

■ Temps Partagé

- Tournique
 - Round-robin
- Priorité
 - Statique
 - Dynamique

15/09/2024

Cours 3 : Ordonnancement- DANT

14

13

14

Batch

Premier Arrivé, Premier Servi

■ First-Come, First-Served (FCFS)

- Le premier processus qui demande la CPU, la reçoit en premier.
- Politique facilement gérée avec une file d'attente FIFO
- Lorsque la CPU est libérée, on lui alloue le processus en tête de file.
 - Le processus en exécution est retiré de la file.
- Pas préemptif

15/09/2024

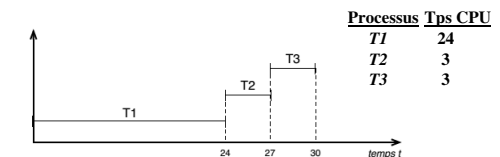
Cours 3 : Ordonnancement- DANT

15

Batch

Premier Arrivé, Premier Servi (cont.)

- Supposons que les processus sont créés dans l'ordre suivant: T1, T2, T3 à t=0



- Temps de réponse : $T_{r_{T_i}}$ = date de fin de T_i - date de création de T_i ;
- Temps d'attente : $T_{a_{T_i}}$ = temps de réponse de T_i - durée de T_i .

Tâche	Temps de réponse	Temps d'attente
T1	$T_{r_{T_1}} = 24 - 0 = 24$ s	$T_{a_{T_1}} = 24 - 24 = 0$ s
T2	$T_{r_{T_2}} = 27 - 0 = 27$ s	$T_{a_{T_2}} = 27 - 3 = 24$ s
T3	$T_{r_{T_3}} = 30 - 0 = 30$ s	$T_{a_{T_3}} = 30 - 3 = 27$ s

Temps d'attente moyen : 17 secondes.

15/09/2024

Cours 3 : Ordonnancement- DANT

16

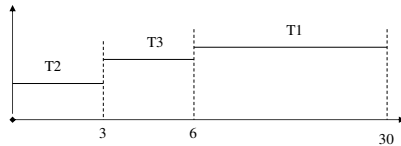
15

16

Batch

Premier Arrivé, Premier Servi (cont.)

- Supposons que les tâches arrivent dans l'ordre $T2$, $T3$, $T1$



- Temps d'attente de $T1 = 6$; $T2 = 0$; $T3 = 3$
- Temps d'attente moyen: $(6 + 0 + 3)/3 = 3$
- Meilleur résultat que le cas précédent
- Effet d'accumulation : un processus court derrière un processus long est pénalisé

15/09/2024

Cours 3 : Ordonnancement- DANT

17

17

Batch

Le travail le plus court d'abord

Shortest-Job-First (SJF)

- Associer à chaque processus la longueur de son prochain temps d'utilisation de la CPU.
 - Lorsque la CPU devient disponible, on lui alloue le processus qui possède le plus court.
- Deux schémas:
 - Non préemptif
 - dès que la CPU est allouée à un processus, ce dernier ne peut pas être interrompu avant la fin de son temps CPU
 - préemptif
 - si un nouveau processus arrive avec un temps CPU plus petit que le reste du temps CPU du processus courant, on commute vers le nouveau processus. Ce schéma est connu sous le nom de Shortest-Remaining-Time-First (SRTF)

SJF est optimal

- Il donne un temps moyen minimal pour un ensemble de processus donnés

15/09/2024

Cours 3 : Ordonnancement- DANT

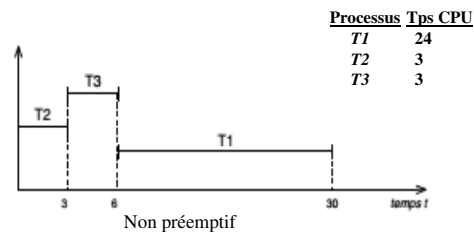
18

18

Batch

Le travail le plus court d'abord (cont.)

- Supposons que les processus sont créés dans l'ordre suivant: $T1$, $T2$, $T3$ à $t=0$



Tâche	Temps de réponse	Temps d'attente
T1	$T_{r_{T1}} = 30 - 0 = 30$ s	$T_{a_{T1}} = 30 - 24 = 6$ s
T2	$T_{r_{T2}} = 3 - 0 = 3$ s	$T_{a_{T2}} = 3 - 3 = 0$ s
T3	$T_{r_{T3}} = 6 - 0 = 6$ s	$T_{a_{T3}} = 6 - 3 = 3$ s

Temps d'attente moyen : 3 secondes.

15/09/2024

Cours 3 : Ordonnancement- DANT

19

19

Algorithme à temps partagé

- Notion de quantum (time-slice)
- Utilisation du registre horloge
- L'horloge (Timer) est un composant externe au processeur

- décrémente le registre horloge RH à une certaine fréquence ($\sim 1\text{MHz}$ – toutes les μs).
 - Si $R_{\text{tempo}} = 0$, alors une interruption horloge est levée. Elle sera ensuite traitée par le système d'exploitation.

15/09/2024

Cours 3 : Ordonnancement- DANT

20

20

Temps partagé Algorithme du tourniquet – (Round Robin)

■ Accès des tâches à tour de rôle au processeur.

- file des tâches prêtes.
 - Création d'une tâche
 - insertion en queue
 - Election d'une tâche (N ticks)
 - choisir la tâche en tête de la file des tâches prêtes
 - fin de quantum (N ticks)
 - Réinsertion de la tâche élue en queue. Celle-ci devient une tâche prête.
 - Entrée / Sortie
 - la tâche passe à l'état bloqué. A la fin de l'E/S, la tâche passe à l'état prêt.

15/09/2024

Cours 3 : Ordonnancement- DANT

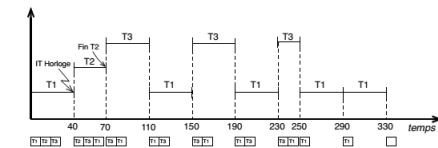
21

21

Temps partagé Algorithme du tourniquet – (Round Robin)

■ Exemple sans E/S

- Trois tâches T1, T2, T3 créées dans cet ordre à $t = 0$
 - T1 dure 200 ms ; T2 dure 30 ms ; T3 dure 100 ms.
 - Quantum = 40ms; temps commutation est négligeable



Temps d'attente des différentes tâches :

- $T_{a_{T_1}} = (330 - 0) - 200 = 130$ ms ;
- $T_{a_{T_2}} = (70 - 0) - 30 = 40$ ms ;
- $T_{a_{T_3}} = (250 - 0) - 100 = 150$ ms ;

15/09/2024

Cours 3 : Ordonnancement- DANT

22

22

Temps partagé Algorithme du tourniquet – (Round Robin)

■ Avantages

- Equitable, simple
- Pas de famine
 - si une tâche T est prête, soit n la taille de la file devant T et q la quantum, alors T attend $n \times q$;

■ Inconvénients

- pas de priorités entre les tâches.

15/09/2024

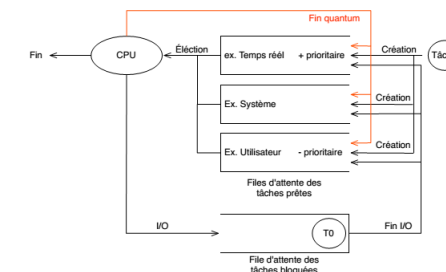
Cours 3 : Ordonnancement- DANT

23

23

Temps partagé avec priorité statique

- Chaque tâche dispose à sa création d'une priorité qui conserve la même valeur tout au long de son exécution.



15/09/2024

Cours 3 : Ordonnancement- DANT

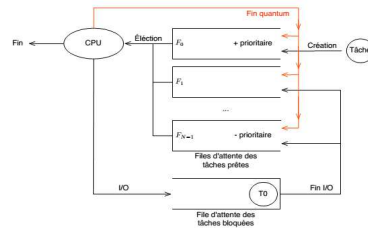
24

24

Temps partagé avec priorité dynamique

■ Les priorités affectées aux tâches changent en cours d'exécution.

- Plusieurs algorithmes sont alors possibles suivant le critère de changement de priorité.
 - Exemple: tâche élue est ré-insérée en queue de la file suivante.



15/09/2024

Cours 3 : Ordonnancement- DANT

25

25

Temps partagé avec priorité

■ Avantages

- Permet de prendre en compte les tâches prioritaires
- Priorité dynamique
 - plus une tâche est élue, moins elle est prioritaire.
 - une tâche ancienne est moins prioritaire

■ Inconvénients

- Pas équitable
- Risque de famine des tâches les moins prioritaires.

15/09/2024

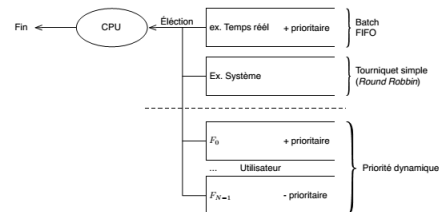
Cours 3 : Ordonnancement- DANT

26

26

Ordonnancement dans les systèmes "modernes"

■ Multi-niveaux



15/09/2024

Cours 3 : Ordonnancement- DANT

27

27