SÉANCE 18

INTRODUCTION À L'ORDONNANCEMENT DES PROCESSUS

SUJETS

Ordonnancement de CPU

Ordonnancement « First Come First Serve » (FCFS) (premier arrivé premier servi)

Ordonnancement Round Robin (RR)

Shortest Job First (SJF) (plus court processus en premier)

Estimation du prochain burst

ORDONNANCEMENT DE CPU

uOttawa
L'Université canadienne
Canada's university

Comment est-ce que le système d'exploitation décide quelle tâche peut exécuter parmi plusieurs dans la file des « processus prêts »?

Ordonnancement: décider quels processus possèdent l'accès aux ressources d'un moment à un autre

ASSOMPTIONS CONCERNANT L'ORDONNANCEMENT

uOttawa
L'Université canadienne
Canada's university

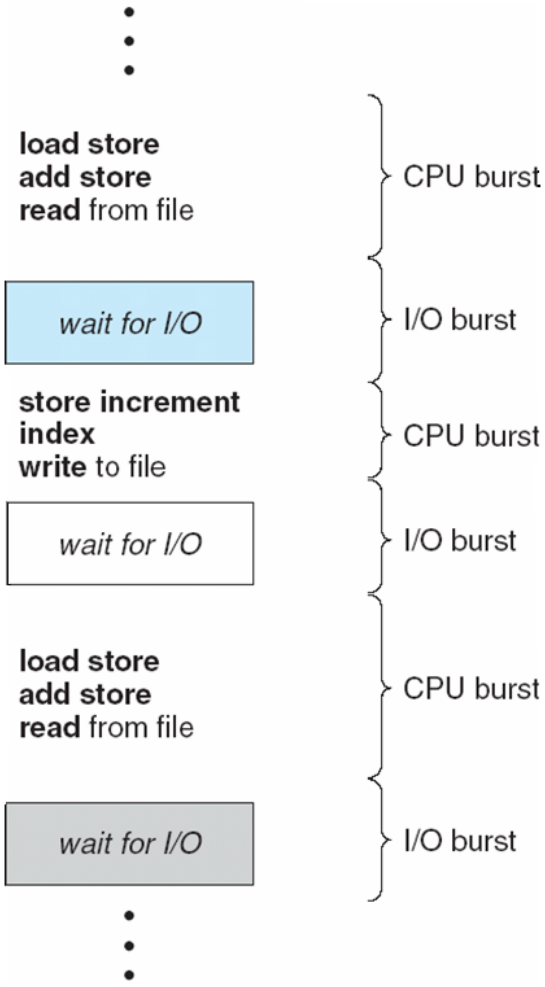
Ordonnancement de CPU était un grand domaine de recherche au début des années 70

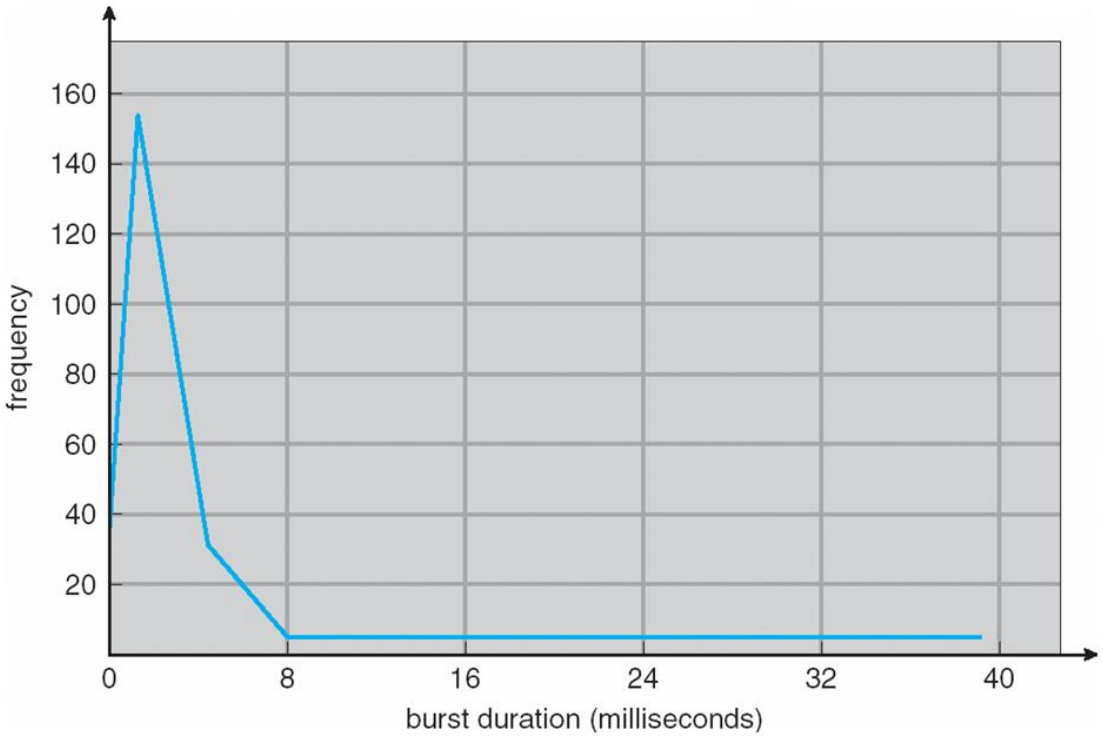
Les études souvent font les hypothèses suivantes

- Un processus par utilisateur
- Un thread par processus
- Les processus sont indépendants

Ce n'est pas réaliste mais ça simplifie le problème

ASSOMPTION: BURSTS DU CPU



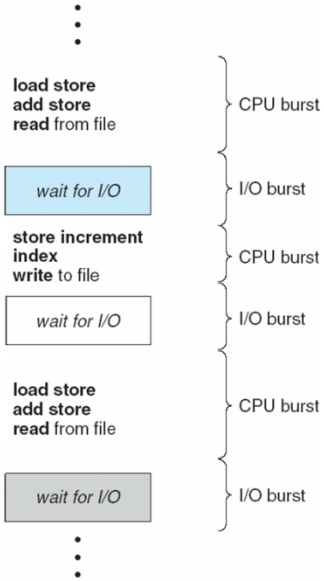


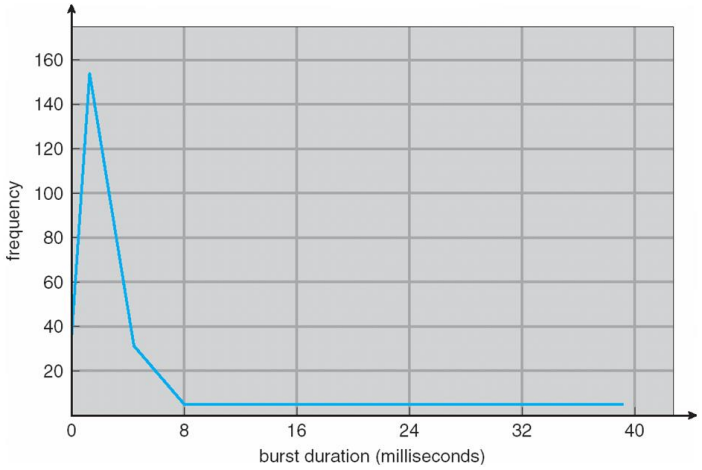
ASSOMPTION: BURSTS DU CPU

uOttawa
L'Université canadienne
Canada's university

Modèle d'exécution: les processus alternent entre les bursts de CPU et les opérations E / S

- Le processus utilise généralement la CPU pendant une certaine période de temps, puis il fait des opérations E / S, puis utilise le CPU de nouveau
- Chaque décision de planification concerne quel processus à attribuer au CPU pour l'exécuter lors du prochain burst de CPU





QU'EST-CE QUI EST IMPORTANT POUR UN ALGORITHME D'ORDONNANCEMENT?

QU'EST-CE QUI EST IMPORTANT POUR UN ALGORITHME D'ORDONNANCEMENT?

uOttawa
L'Université canadienne
Canada's university

Minimiser le temps de réponse

- Temps passé pour effectuer une opération (travail)
- Le temps de réponse que l'utilisateur observe
  - Le temps pour refléter la frappe dans un éditeur
  - Le temps pour compiler un programme
  - Tâches en temps réel: doit respecter les délais imposés par l'environnement

QU'EST-CE QUI EST IMPORTANT POUR UN ALGORITHME D'ORDONNANCEMENT?

Maximiser le throughput (débit)

- Travaux par secondes
- Débit relié au temps de réponse, mais non identique
    - Minimiser le temps de réponse cause plus de commutation de contexte que si vous maximiser le débit seulement
- Minimiser le overhead (temps de commutation de contexte) en plus de l'utilisation efficace des ressources (CPU, disque, mémoire, etc.)

QU'EST-CE QUI EST IMPORTANT DANS UN ALGORITHME D'ORDONNANCEMENT?

Justesse

- Partager le CPU entre les processus d'une manière équitable
- Ne pas seulement minimiser le temps de réponse

ALGORITHMES D'ORDONNANCEMENT: FIRST-COME, FIRST-SERVED (FCFS)

"Run until Done:" algorithme FIFO

Au début, ceci voulait dire qu'un processus coure d'une façon non-préemptif jusqu'à ce qu'il soit terminé

- Incluant tout processus bloqué dans les opérations I/O

Maintenant, FCFS veut dire qu'un processus garde le CPU jusqu'à ce qu'il complète son burst

Exemple: Trois processus arrivent dans l'ordre P1, P2, P3.

- Temps de burst de P1: 24
- Temps de burst de P2: 3
- Temps de burst de P3: 3

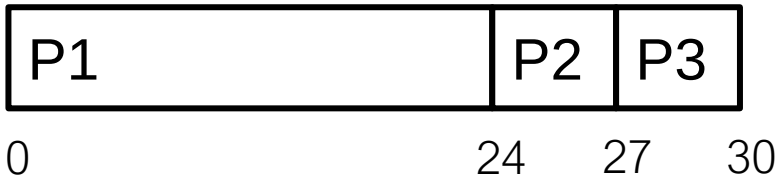
Tracez le diagramme Gantt et calculez la temps moyen d'attente et le temps moyen d'achèvement.

ALGORITHMES D'ORDONNANCEMENT: FIRST-COME, FIRST-SERVED (FCFS)

uOttawa
L'Université canadienne
Canada's university

Exemple: Trois processus arrivent dans l'ordre P1, P2, P3.

- Temps de burst de P1: 24
- Temps de burst de P2: 3
- Temps de burst de P3: 3



Temps d'attente:

- P1: 0
- P2: 24
- P3: 27

Temps d'achèvement:

- P1: 24
- P2: 27
- P3: 30

Temps moyen d'attente: (0+24+27)/3 = 17

Temps moyen d'achèvement: (24+27+30)/3 = 27

ALGORITHMES D'ORDONNANCEMENT: FIRST-COME, FIRST-SERVED (FCFS)

Qu'en est-il si l'ordre fut P2, P3, P1?

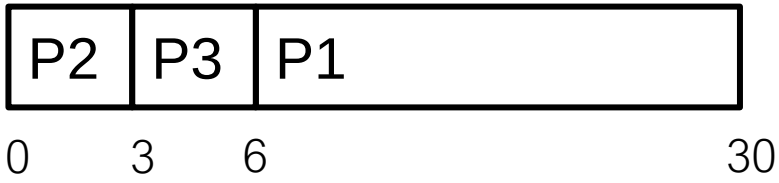
- Temps de burst de P1: 24
- Temps de burst de P2: 3
- Temps de burst de P3: 3

ALGORITHMES D'ORDONNANCEMENT: FIRST-COME, FIRST-SERVED (FCFS)

uOttawa
L'Université canadienne
Canada's university

Qu'en est-il si l'ordre fut P2, P3, P1?

- Temps de burst de P1: 24
- Temps de burst de P2: 3
- Temps de burst de P3: 3



Temps d'attente:

- P2: 0
- P3: 3
- P1: 6

Temps d'achèvement:

- P2: 3
- P3: 6
- P1: 30

Temps moyen d'attente: (0+3+6)/3 = 3 (comparé à 17)

Temps moyen d'achèvement: (3+6+30)/3 = 13 (comparé à 27)

ALGORITHMES D'ORDONNANCEMENT: FIRST-COME, FIRST-SERVED (FCFS)

Temps moyen d'attente: (0+3+6)/3 = 3 (comparé à 17)

Temps moyen d'achèvement: (3+6+30)/3 = 13 (comparé à 27)

Avantages et Désavantages de FCFS:

- Simple (+)
- Les travaux courts se coincent derrière les travaux longs (-)
- La performance est dépendante de l'ordre dans lequel les travaux arrivent (-)

COMMENT PEUT-ON AMÉLIORER CECI?

ORDONNANCEMENT ROUND ROBIN (RR)

uOttawa
L'Université canadienne
Canada's university

Méthode Round Robin

- Chaque processus obtient une petite unité de temps de CPU (quantum de temps)
  - Habituellement 10-100 ms
- Après l'expiration du quantum, le processus est préempté et ajouté à la fin de la file des « processus prêts »
- Soit N processus dans la file des « processus prêts » et le quantum de temps est Q ms:
  - Chaque processus reçoit 1/N du temps de CPU
  - Quel est le temps d'attente maximum pour chaque processus 
    - ***Pas d'attentes de processus plus que (N-1)Q unités de temps***

ORDONNANCEMENT ROUND ROBIN (RR)

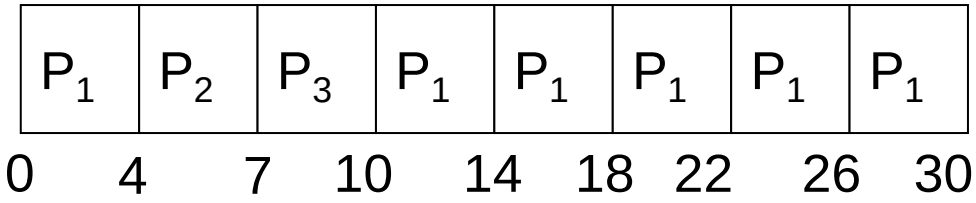
La performance dépend de la valeur de Q

- Petite valeur de Q ➔ beaucoup de overhead
- Grande valeur de Q est comme **FCFS**
- Q doit être grand comparé **au temps de commutation de contexte**, sinon le overhead est trop haut
  - Vous dépensez la plupart de votre temps en faisant la commutation de contexte!

EXEMPLE DE RR AVEC QUANTUM DE TEMPS = 4

Processus	Temps de Burst
P_1	24
P_2	3
P_3	3

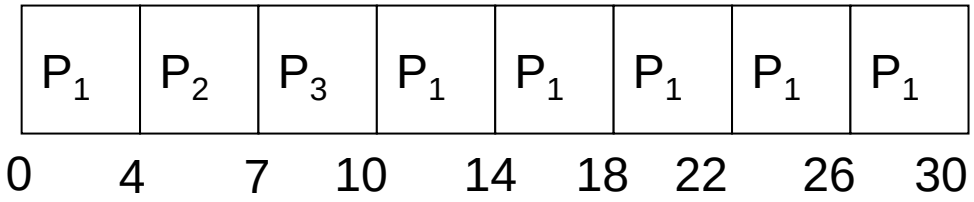
Le diagramme Gantt est:



EXEMPLE DE RR AVEC QUANTUM DE TEMPS = 4

uOttawa
L'Université canadienne
Canada's university

Processus	Temps de Burst
P_1	24
P_2	3
P_3	3



Temps d'attente:

- P1: (10-4) = 6
- P2: (4-0) = 4
- P3: (7-0) = 7

Temps d'achèvement:

- P1: 30
- P2: 7
- P3: 10

Temps moyen d'attente: (6 + 4 + 7)/3= 5.67

Temps moyen d'achèvement: (30+7+10)/3=15.67

EXEMPLE DE RR AVEC QUANTUM DE TEMPS = 20

Processus	Temps de Burst
P1	*53*
P2	*8*
P3	*68*
P4	*24*

EXEMPLE DE RR AVEC QUANTUM DE TEMPS = 20

uOttawa
L'Université canadienne
Canada's university

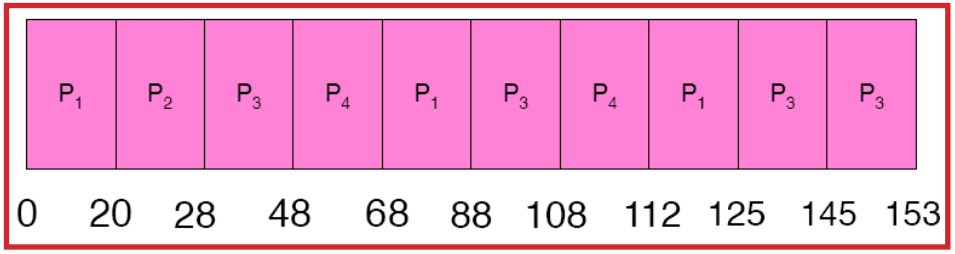
Temps d'attente:

- P1: (68-20)+(112-88) = 72
- P2: (20-0) = 20
- P3: (28-0)+(88-48)+(125-108) = 85
- P4: (48-0)+(108-68) = 88

Process	Temps de Burst
P1	*53*
P2	*8*
P3	*68*
P4	*24*

Temps d'achèvement:

- P1: 125
- P2: 28
- P3: 153
- P4: 112



Temps moyen d'attente: (72+20+85+88)/4 = 66.25

Temps moyen d'achèvement: (125+28+153+112)/4 = 104.5

SOMMAIRE DE RR

uOttawa
L'Université canadienne
Canada's university

Avantages et Désavantages:

- Meilleur pour les travaux courts (+)
- Juste (+)
- Le temps de commutation de contexte augmente pour les travaux longs (-)
    - Les exemples précédents supposent que la commutation de contexte ne prend pas de temps
    - En réalité, la commutation de contexte augmente le temps d'attente et d'achèvement
    - Rappelez-vous: le système d'exploitation utilise des ressources aussi!

Si le quantum choisi est

- Trop grand, le temps de réponse souffre
- Infini, la performance est la même que FCFS
- Trop petit, le throughput (débit) souffre et le pourcentage de overhead grandit

ORDONNANCEMENT SHORTEST-JOB-FIRST (SJF)

uOttawa
L'Université canadienne
Canada's university

Associez à chaque processus la durée de son prochain burst

Choisissez le processus avec le burst le plus court pour exécuter sur le CPU

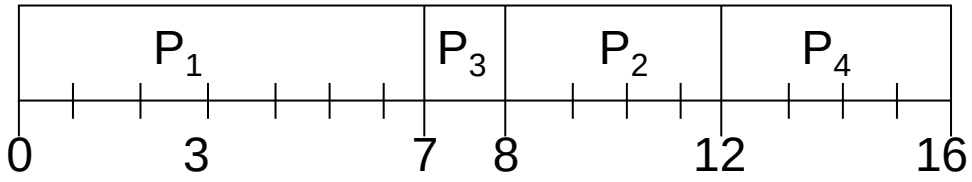
Deux méthodes:

- Non préemptif - une fois que le CPU a été administré au processus, il ne peut pas être préempté jusqu'à ce qu'il ait terminé son burst
- Préemptif - si un nouveau processus arrive avec la durée de de burst moins que le temps du processus qui exécute actuellement, préempte
    - Cette méthodes est appelée ***Shortest-Remaining-Time-First (SRTF)***

SJF est optimal - donne un temps d'attente moyen minimum pour un ensemble de processus donné

EXEMPLE DE SJF NON PRÉEMPTIF

Processus	Temps d'arrivé	Burst
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

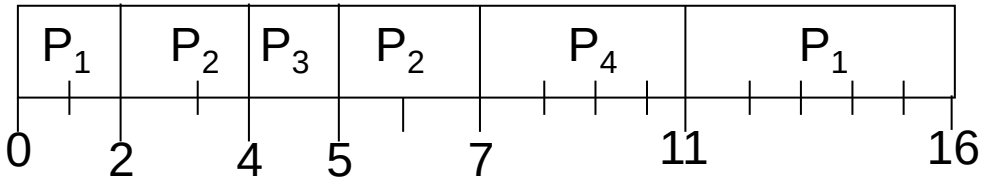


Temps moyen d'attente : (0 + 6 + 3 + 7)/4 = 4

Temps moyen d'achèvement : (7 + 10 + 4 + 11)/4 = 8

EXEMPLE DE SJF PRÉEMPTIF

Processus	Temps d'arrivé	Burst
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4



Temps moyen d'attente : (9 + 1 + 0 +2)/4 = 3

Temps moyen d'achèvement : (16 + 5 + 1 + 6)/4 = 7

SHORTEST JOB FIRST: CRITIQUE

uOttawa
L'Université canadienne
Canada's university

Possibilité de manque de ressources pour des processus plus longs tant qu'il y a un flux régulier de processus plus courts

Le manque de préemption n'est pas adapté à un environnement de partage de temps

- Un processus sans E / S pourrait monopoliser la CPU si c'est le premier à entrer dans le système

SJF intègre implicitement les priorités: les processus les plus courts sont favorisés

ORDONNANCEMENT PRIORITAIRE

Un nombre de priorité (entier) est associé à chaque processus

La CPU est allouée au processus avec la plus haute priorité (le plus petit nombre entier ≡ priorité la plus élevée).

- Préemptif
- Non-préemptif

SJF est une méthode d'ordonnancement Prioritaire où la priorité est le prochain temps de burst de CPU.

Problème ≡ Manque les processus à faible priorité risques ne jamais exécutés

Solution ≡ Vieillissement - au fil du temps, augmenter la priorité du processus

DÉTERMINATION DU BURST SUIVANT

uOttawa
L'Université canadienne
Canada's university

Peut être fait en utilisant la longueur des bursts de CPU précédentes, en utilisant une opération de moyenne exponentielle

1. t_n = longueur réelle du burst n
2. τ_{n+1} = valeur prédite pour le prochain burst
3. $\alpha, 0 \leq \alpha \leq 1$

$$\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n$$

EXEMPLE

Supposons les valeurs actuel pour les bursts pour un processus: 6, 4, 6, 4, 13, 13, 13 et supposons que l'estimation initiale est 10

Calculé l'estimation du prochain burst, α = 0.8

τ_n	10	6.8	4.56	5.71	4.34	11.27	12.49
T_n	6	4	6	4	13	13	13
τ_{n+1}	6.8	4.56	5.71	4.34	11.27	12.49	12.89

$$\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n$$

EXEMPLE

Supposons les valeurs actuel pour les bursts pour un processus: 6, 4, 6, 4, 13, 13, 13 et supposons que l'estimation initiale est 10

Calculé l'estimation du prochain burst, α=0.2 et compare avec α=0.8

τ_n		10	6.8	4.56	5.71	4.34	11.27	12.49
T_n		6	4	6	4	13	13	13
τ_{n+1}	α=0.8	6.8	4.56	5.71	4.34	11.27	12.49	12.89
τ_{n+1}	α=0.2	8.96	7.808	7.206	6.405	7.204	7.843	8.354

MERCI!

QUESTIONS?