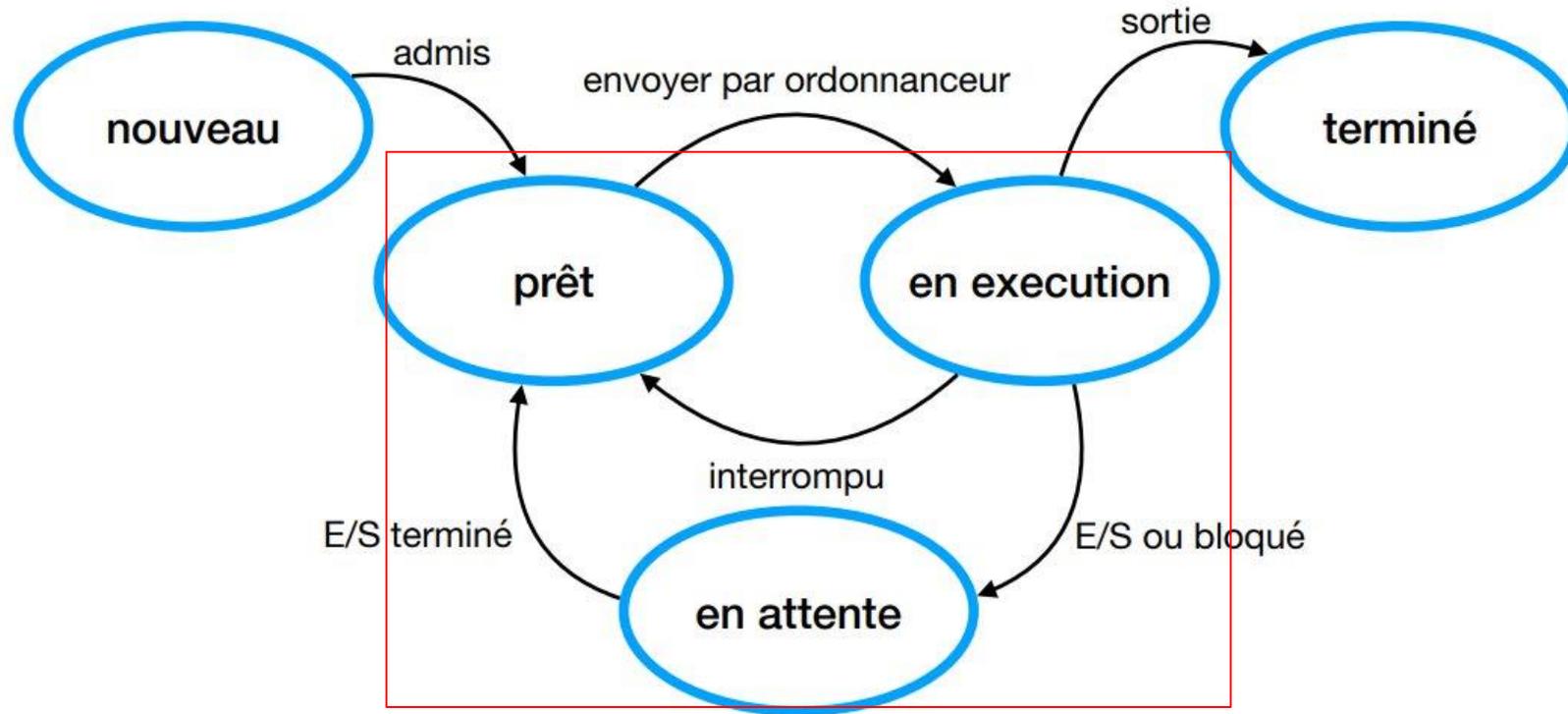


Systemes d'exploitation

Demo Ordonancement

Rappels



Rappels

L'ordonnanceur à court terme choisit parmi les processus dans la file d'attente prêts (ready queue).

Les décisions d'ordonnancement se font :

- Lorsqu'un processus passe de l'état d'exécution (running) à l'état d'attente (waiting).
- Lorsqu'un processus passe de l'état d'exécution (running) à l'état prêt (ready) à cause d'une interruption.
- Lorsqu'un processus passe de l'état d'attente (waiting) à l'état prêt (ready) suite à la fin d'une opération d'entrée/sortie.
- Lorsqu'un processus termine son exécution.

L'ordonnancement non-préemptif: Changement d'états sous le contrôle du processus en cours d'exécution.

L'ordonnancement préemptif: Changement d'états sous le contrôle du système d'exploitation.

Définitions, Métriques et Formules

- **Temp Réel (Real Time / Wall-Clock Time)**: Quantité de temp réel .
- **Temp CPU (CPU Time)**: Quantité de temp que le CPU execute du code .

↑ (On veut maximiser)
↓ (On veut minimiser)

- ↑ **Utilisation CPU (CPU utilisation)**: Ratio entre le temp CPU total et le temp réel total
- ↑ **Débit (Throughput)**: Quantité de travail effectuée par unité de temps.

- **Temp d'arrivée (Arrival Time)**: Moment où le processus arrive dans la "ready queue" et attend d'être exécuté.
- **Temp de premiere execution (First Run Time)**: Moment où le processus est executer pour la premiere fois.
- **Temp de complétion (Completion Time)**: Moment où le processus termine son exécution.

$$T_{turnaround} = T_{completion} - T_{arrival}$$

$$T_{wait} = T_{turnaround} - T_{burst}$$

- ↓ **Temps de rafale (Burst Time)**: Quantité de temps CPU qu'un processus prend pour son exécution.
- ↓ **Délai d'exécution (Turnaround Time)**: Temps écoulé entre l'arrivée du processus et sa complétion.
- ↓ **Temps d'attente (Wait Time)**: Quantité de temps dans la "ready queue".
- ↓ **Temps de réponse (Response Time)**: Quantité de temps entre le temps d'arrivée et sa première exécution.

$$T_{response} = T_{firstrun} - T_{arrival}$$

Diagrammes de Gantt

Process	Burst Time
P1	4
P2	2
P3	3
P4	5



First In, First Out (FIFO)

- On execute les processus en ordre d'arriver
- Simple

First In, First Out (FIFO)

Process	Arrival Time	Burst Time
P1	0	4
P2	0	2
P3	0	3
P4	0	5

Questions

- Produire le diagram de gantt
- Trouvé le délai d'execution (turnaround time) moyen
- Trouvé le temp de réponse (response time) moyen

First In, First Out (FIFO)

Process	Arrival Time	Burst Time	Turnaround Time	Response Time
P1	0	4	4	0
P2	0	2	6	4
P3	0	3	9	6
P4	0	5	14	9



Average Turnaround Time = 8.25
Average Response Time = 4.75

Est-ce que FIFO c'est bon?

First In, First Out (FIFO)

- Souffre du **phénomène du convoi (convoy effect)** : lorsque de courtes bursts d'exécution attendent des grosses bursts d'exécution.
- Imaginez des cas extrêmes!

Shortest Job First (SJF)

- On exécute les processus en ordre croissant des burst time.
- Optimal pour le turnaround time
- Simple

Shortest Job First (SJF)

Process	Arrival Time	Burst Time
P1	0	4
P2	0	2
P3	0	3
P4	0	5

Questions

- Produire le diagram de gantt
- Trouvé le délai d'execution (turnaround time) moyen
- Trouvé le temp de réponse (response time) moyen

Shortest Job First (SJF)

Process	Arrival Time	Burst Time	Turnaround Time	Response Time
P1	0	4	9	5
P2	0	2	2	0
P3	0	3	5	2
P4	0	5	14	9



Average Turnaround Time = 7.5
Average Response Time = 4

Simple, optimal, on a fini?

Shortest Job First (SJF)

- Pas tous les processus n'arrivent en même temps. Donc, on peut retomber dans l'effet convoi.
- Nous n'avons pas toujours le burst time d'un processus, donc nous ne savons pas quel job est plus court. (On peut utiliser une approximation)
- Il faut développer des algorithmes plus complexes et non-optimal!

Shortest Remaining Time First (SRTF)

- Si l'on peut se permettre de lancer des interruptions pour forcer un changement de contexte (ordonnanceur préemptif), on peut généraliser le SFJ (Shortest Job First).
- Nous exécuterons les processus qui sont le plus proche de leur complétion.
- La même chose que : Shortest Time-to-Completion First (STCF)
- Optimal pour le turnaround time

Shortest Remaining Time First (SRTF)

Process	Arrival Time	Burst Time
P1	0	6
P2	1	4
P3	2	2
P4	4	4

Questions

- Produire le diagram de gantt
- Trouvé le délai d'execution (turnaround time) moyen
- Trouvé le temp de réponse (response time) moyen

Shortest Remaining Time First (SRTF)

Process	Arrival Time	Burst Time
P1	0	6
P2	1	4
P3	2	2
P4	4	4

Questions

- Produire le diagram de gantt
- Trouvé le délai d'execution (turnaround time) moyen
- Trouvé le temp de réponse (response time) moyen

Shortest Remaining Time First (SRTF)

Process	Arrival Time	Burst Time	Turnaround Time	Response Time
P1	0	6 --> 5	16	0
P2	1	4 --> 3	6	0
P3	2	2	2	0
P4	4	4	7	3



Average Turnaround Time = 7.75
Average Response Time = 0.75

Est-ce que SRTF souffre de problèmes?

Shortest Remaining Time First (SRTF)

- Pourrait avoir des problèmes avec le response time, et on veut des ordinateurs interactifs. (Meme problème avec STF)
- Imaginez des cas extrêmes!

Algorithme du tourniquet (Round Robin)

- On veut avoir un bon temps de réponse ! C'est une métrique très importante pour les systèmes interactifs.
- On va forcer une interruption (ordonnanceur préemptif) à chaque tranche de temps (time slice).
- Permet d'être juste (fairness).
- Fréquemment utilisé en pratique avec d'autres heuristiques

Remarque cool: l'opération `yield` donne le reste de ton time slice à un autre processus.

Algorithme du tourniquet (Round Robin)

Process	Arrival Time	Burst Time
P1	0	8
P2	2	2
P3	6	2
P4	6	4

Questions

- Produire le diagram de gantt
- Trouvé le délai d'execution (turnaround time) moyen
- Trouvé le temp de réponse (response time) moyen

Quantum (time slice) = 2

Algorithme du tourniquet (Round Robin)

Process	Arrival Time	Burst Time	Turnaround Time	Response Time
P1	0	8 --> 6 --> 4 --> 2	16	0
P2	2	2	2	0
P3	6	2	2	0
P4	6	4 --> 2	8	2



Average Turnaround Time = 7
Average Response Time = 0.5

Est-ce que RR souffre de problèmes?

Algorithme du tourniquet (Round Robin)

- Beaucoup de changements de contexte !
- On doit combiner RR avec d'autres heuristiques et algorithmes comme SRTF pour prendre de meilleures décisions.

Autres considérations

- No oracle: SE ne connais presque jamais le burst time.
- Utilisation du CPU et débit
- Intégration de l'E/S (IO)
- Files d'attente multi-niveaux
- Processeurs multi-coeurs
- Coûts des changements de contexte
- Deadlines
- ...

Beaucoup de considérations! Pas de solution "one size fits all"