

INTRODUCTION AUX SYSTEMES D'EXPLOITATION

TD5 ***L'ordonnancement***

S O M M A I R E

1. QUELQUES DEFINITIONS RAPIDES.....	1
2. EXERCICE 1: EVALUATION D'ALGORITHMES D'ORDONNANCEMENT SUR SCENARIO DE PROCESSUS	1
3. EXERCICE 2: ORDONNANCEMENT UNIX SVR3 ET 4.3 BSD UNIX.....	2

1. Quelques définitions rapides

Quelques définitions concernant les métriques de base utilisées pour évaluer les algorithmes d'ordonnancement:

- **Temps de rotation/turnaround (T_r):** intervalle de temps entre la soumission d'un processus et sa complétion. Ce temps inclut le temps d'attente et d'exécution. Métrique utilisée principalement pour le traitement batch (par lot).
- **Temps de réponse:** temps entre la soumission d'une tâche et la (première?) réponse. Métrique utilisée principalement pour les tâches interactives.
- **Débit:** nombre de tâches complétées par unité de temps
- **Temps de service (T_s):** temps d'exécution/service d'une tâche.

2. Exercice 1: Evaluation d'algorithmes d'ordonnancement sur scénario de processus

Dans cet exercice on évaluera selon certaines métriques les algorithmes d'ordonnancement suivants:

- **FCFS** (First Come First Served): premier arrivé premier servi, algorithme non préemptif
- **Tourniquet:** ou Round Robin, algorithme préemptif caractérisé par le quantum de temps défini (*timeslice*)
- **SPN** (Shortest Process Next): Processus avec temps d'exécution le plus court d'abord
- **SRT** (Shortest Remaining Time): version preemptive du SPN
- **HRRN** (Highest Response Ratio Next): cet algorithme non préemptif choisit le processus avec le "ratio de réponse R" le plus important. Avec $R = (T_a + T_s) / T_s$ ou T_a est le temps d'attente.
- **FB** (Feedback): si le temps d'exécution total d'un processus n'est pas connu, on peut se baser sur le temps passé à s'exécuter. La priorité du processus baisse au fur et à mesure de son exécution (→ passage à une file différente et de priorité moindre).

Nous supposons le scénario suivant:

Processus	Temps d'arrivée	Temps de service
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Pour ce scénario, remplissez le tableau suivant pour les politiques/algorithmes: FCFS, RR ($q=1$), RR ($q=4$), SPN, SRT, HRRN, FB ($q=1$), FB ($q=2^i$) avec i le numéro de la file d'attente ($i=0$ pour la première file).

	A	B	C	D	E	moyenne
Temps de terminaison						
Temps de rotation Tr						
Tr/Ts						

3.Exercice 2: ordonnancement UNIX SVR3 et 4.3 BSD UNIX

Systèmes à temps partagé et interactif, l'ordonnanceur essaye de donner de bonnes performances aux processus interactifs tout en évitant leur famine. L'ordonnanceur conventionnel utilise le **Feedback multiniveaux** (plusieurs files de priorité) avec un tourniquet sur chaque file de priorité. Le quantum de temps utilisé est de 1 seconde (c'est un temps très important par rapport à ce qui est utilisé dans les systèmes actuels).

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + nice_j$$

Ou:

- $CPU_j(i)$ est la mesure de l'utilisation de CPU du processus j dans l'intervalle i .
- $P_j(i)$ est la priorité du processus j au début de l'intervalle i (valeur basse => priorité haute).
- $Base_j$: valeur de priorité de base du processus j .
- $nice_j$: facteur d'ajustement (contrôlé par l'utilisateur).

La priorité de chaque processus est recalculée au bout d'une seconde.

1. Nous avons 3 processus A, B, et C qui sont créés en même temps avec des priorités de base de 60. Montrez l'exécution des 3 processus. On suppose ici que l'horloge interrompt le système 60 fois par seconde pour incrémenter le compteur d'utilisation de CPU (qui est donc incrémenté 60 fois par seconde).
2. Si l'on met la valeur de $nice = 10$ pour le processus A. Quel effet cela a-t-il sur l'ordonnancement.