


Examen de Système d'Exploitation : Session Normale
Durée : 2h00mn

Documents non autorisés

Exercice 1 (5 points)

Dessiner le diagramme simplifié d'états des processus en multiprogrammation non préemptive et pour chacune des transitions entre deux états, indiquer un exemple d'élément qui pourrait en être l'origine.

(Voir cours le diagramme à 3 (Prêt-Elu-Bloqué) ou 5 (Création-Prêt-Elu-Bloqué-Terminé) états sans transition de Elu vers prêt (interruption système).

Dans un système à un processeur, on a les 5 processus suivant A₂, B₆, C₃, D₁, E₂ dont les numéros en indice représentent leur temps d'exécution respectif.

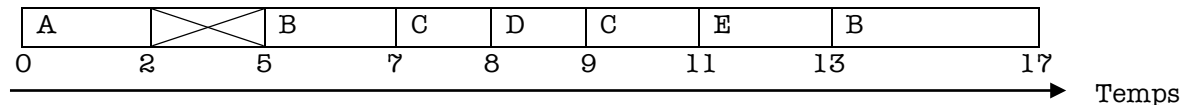
1) Si les processus C et D sont à l'état bloqué, E à l'état d'exécution, et les processus restant à l'état prêt : on applique l'algorithme SJF, donner l'ordre des processus dans la file d'attente?

Avec SJF on a : B -- A ---> tête de la file d'attente

Les processus bloqués ne sont pris en compte par l'ordonnanceur

2) Si on suppose que le processus A se présente en premier, à l'instant 0, B se présente à l'instant 5, C se présente à l'instant 7, D est débloquent à l'instant 8 et E est débloquent à l'instant 9, dessinez un schéma illustrant l'occupation du processeur en fonction du temps en utilisant SRTF.

Avec SRTF on a :


Exercice 2 (6points)

Soit un système de segmentation avec pagination de 4 segments, et une mémoire physique de 16 octets. Chaque segment est paginé (page de taille 4 octets).

Une variable globale X est décalée de 3 dans la 2^{ème} page (chargée dans le 3^{ème} cadre) du segment 1 de taille 32 octets.

1) Sachant que l'adressage est faite par octet et le décompte des numéros commence par 1 et l'adressage par 0, déterminer :

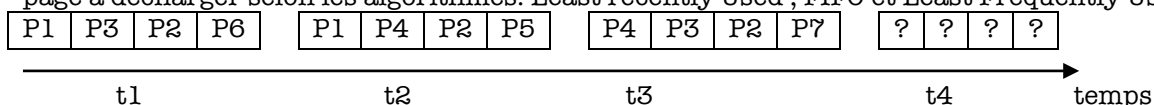
- L'adresse logique de la variable?
- L'adresse physique de la variable?

$$\begin{aligned} \text{Adresse logique} &= \text{N}^\circ\text{segment} - \text{N}^\circ\text{Page} - \text{Offset} \\ &= 0000111 \end{aligned}$$

$$\begin{aligned} \text{Adresse Physique} &= \text{N}^\circ\text{cadre} - \text{Offset} \\ &= 1011 \end{aligned}$$

1) Si l'utilisation de la mémoire physique en fonction du temps (t₁, t₂, t₃) est représentée à travers le schéma suivant par les cadres et les pages qui y sont chargés.

Si on souhaite charger la page X à t₄, en remplaçant les ? à t₄; indiquer en cas de défaut de page, la page à décharger selon les algorithmes: Least recently Used, FIFO et Least Fréquently Used.



Pour LRU : la page la moins récemment utilisée est déchargée. A t₃ on a 4 pages en mémoire utilisées dans le temps selon l'ordre P4, P3, P2, P7, donc la page P4 est la moins récemment utilisée, par conséquent c'est celle qui sera déchargée en cas de défaut de page.

Pour FIFO: parmi les page présentes en mémoire à $t3$, la page $P2$ est plus ancienne, donc c'est elle qui sera déchargée en cas de défaut de page.

Pour LFU: la fréquence d'utilisation des pages ($P4, P3, P2, P7$) residentes en mémoire à $t3$ est:

- Pour $P4=2$ depuis son dernier chargement en mémoire à $t2$
- Pour $P3=1$ depuis son dernier chargement en mémoire à $t3$
- Pour $P2=3$ depuis son dernier chargement en mémoire à $t1$
- Pour $P7=1$ depuis son dernier chargement en mémoire à $t3$

Par conséquent $P7$ ou $P3$ sera déchargée, et ce sera $P3$ si on utilise FIFO en cas d'égalité.

Exercice 3 (6points)

Soient 3 processus $P1, P2, P3$ chargés de calculer $T5 = (a + b) * (b + c) - (c/a)$

$P1$ calcule $T1 = (a + b)$, $P2$ calcule $T2 = (b + c)$, et $P3$ calcule $T3 = c/a$. $P2$ calcule $T4 = T1 * T2$ et $P3$ calcule le résultat en faisant $T5 = T4 - T3$. En utilisant les sémaphores, proposer un schéma permettant de gérer l'exclusion mutuelle des sections critiques et l'interdépendance entre processus.

Les variables partagées sont $a, b, c, T1, T4$

On associe les les sémaphores $S0$ à a , $S1$ à b , $S2$ à c pour organiser l'exclusion mutuelle, et $S3$ à $T1$, et $S4$ à $T4$ pour organiser l'interdépendance

On initialise les sémaphores $S0=S1=S2=1$; et $S3=S4=0$;

P1	P2	P3
P(S1)	P(S2)	P(S0)
P(S0)	P(S1)	P(S2)
$T1 = a + b$	$T2 = b + c$	$T3 = c/a$
V(S0)	V(S1)	V(S2)
V(S1)	V(S2)	V(S0)
V(S3)	P(S3)	
	$T4 = T1 * T2$	
	V(S4)	P(S4)
		$T5 = T4 - T3$

NB: comme $P2$ fait $T2 = b + c$ et $T4 = T1 * T2$, il ne pourra jamais utiliser $T2$ en même temps, par conséquent on n'a pas besoin d'une exclusion mutuelle sur $T2$. Idem pour $P3$ qui utilise $T3$ pour $T3 = c/a$ et $T5 = T4 - T3$.

On constate que si l'un des processus entre dans sa section critique pour executer $(a + b)$ ou $(b + c)$ ou (c/a) les autre sont bloqué, donc on peut simplifier en utilisant un seul semaphore à la place de $S0, S1, S2$ (voir schema suivant).

P1	P2	P3
P(S0)	P(S0)	P(S0)
$T1 = a + b$	$T2 = b + c$	$T3 = c/a$
V(S0)	V(S0)	V(S0)
V(S3)	P(S3)	
	$T4 = T1 * T2$	
	V(S4)	P(S4)
		$T5 = T4 - T3$

Exercice 4 (4points)

1) Quel est le nombre de processus créé par le programme suivant ?

2) Donner un des affichages possibles en fonction du choix de l'ordonnanceur.

Commentaire du code

```
#include <stdio.h> /*pour printf()*/
#include <unistd.h> /*pour fork()*/
int main(int argc, char **argv) {
    int var=20;
    if (fork() == 0 ) { /*Notre programme P0 crée un processus fils P1*/
        var=var+10; } /*Continue l'exécution de P0 et commence celle de P1 (qui considère que var=20)*/
    if (fork() == 0) /*Notre programme et le premier processus fils P1 créent chacun un processus respectivement fils (P2) et petit fils(P3) */
        {var=var+30;} /*Continue l'exécution de P0, de P1 et commence celle de P2 et P3 (qui considèrent que var=20)*/
    else
        {var=var+30;} /*si la creation des processus est réussie, cette instruction ne sera jamais executée*/
    printf ("la variable vaut %d\n", var);
    return 0;}
```

1) Nous avons 4 processus créés (notre programme P0, P1, P2, P3)

2) Quelques affichages possible

la variable vaut 50 -> affichage de P2 ou P3

la variable vaut 60 -> affichage de P0 ou P1

la variable vaut 60 -> affichage de P0 ou P1

la variable vaut 50 -> affichage de P2 ou P3

la variable vaut 50 -> affichage de P0 ou P1

la variable vaut 50 -> affichage de P0 ou P1

la variable vaut 60 -> affichage de P0 ou P1

la variable vaut 60 -> affichage de P0 ou P1

etc.....