Université Ziane Achour de djelfa
Département d'informatique et mathématique
Module : Système d'exploitation 2.
Niveau : 3ème LMD , Année : 2009/2010

Examen	02H00

Questions : (5 Points)

1. Quelle est la différence principale entre un processus et un thread ?
2. Qu'est ce qu'une section critique ? Donnez un exemple pour les threads. Que peut-il se passer lorsque l'on supprime du programme l'entrée et la sortie d'une section critique ?
3. Quelles sont les conditions que doit vérifier une solution du problème de la section critique ?
4. Qu'est ce qu'un processus zombie ?
5. Qu'est ce qu'un interblocage ? Dans quelles conditions cela se produit-il ?
6. Comment prévenir les interblocages ? Donnez et expliquez trois méthodes.
7. On considère la stratégie suivante d'allocation de ressources. Chaque fois qu'un processus effectue la demande d'une ressource qui n'est pas disponible, tous les processus en attente d'une ressource quelconque sont examinés. Si l'un d'entre eux tient la ressource demandée, elle lui est réquisitionnée et attribuée au processus demandeur. Discuter cette stratégie en termes de blocage et de famine, en vous référant le plus possible au cours.

Exercice n°1: (1.5 Points)

Soit le programme :

```
int main (){
    int i, x, f;
    x = 1;
    for (i = 0; i < 3; i++) {
      pid = fork ();
      x = x * 2;
    printf ("pid %d : %d\n", getpid(), x);
}
```

1. Faire un schéma de l'arborescence de processus créée par ce programme;
2. Combien de processus seront créés lors de l'exécution du programme ?
3. Combien de lignes seront imprimées lors de l'exécution du programme ?

Indiquer une des suites possibles, par exemple :
pid 500 : 2
...

Justifier, en les commentant, les valeurs affichées.

4. Proposer une modification en utilisant le code de retour de la fonction fork afin de créer uniquement 3 processus fils d'un même processus parent.

Exercice n°2: (1.5 Points)

On considère deux processus P0 et P1 dont on désire gérer l'accès à une section critique. On définit le tableau requete, partagé par les deux processus de la façon suivante :

int requete [2] ;

dont les éléments sont initialisés à zéro .

On propose une première solution :

Processus P0	Processus P1

int main(void){
...
requete[0]= 1;
while (requete[1] == 1) {};
/* debut section critique */
...
/* fin section critique */
requete[0]= 0;
...
}
	int main(void){
	...
	requete[1]= 1;
	while (requete[0] == 1) {};
	/* debut section critique */
	...
	/* fin section critique */
	requete[1]= 0;
	...
	}

1. Montrer que cette solution n'est pas correcte.

- On considère maintenant une deuxième solution où l'on introduit une variable partag
tour: int tour ;

Processus P0	Processus P1

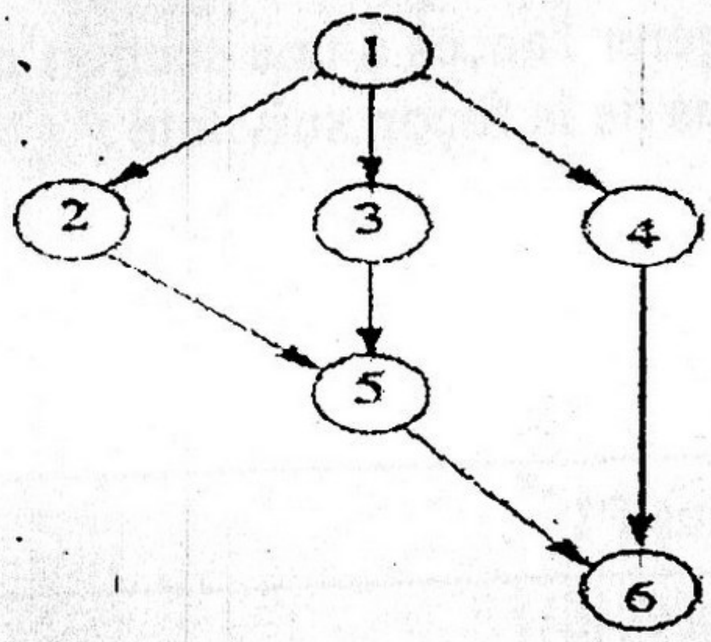
void main (void) {
requete[0]= 1;
tour = ***;
while ((requete[1] == 1)&&(tour == ***))
{};
/* debut section critique */
...
/* fin section critique */
requete[0]= 0;
...
}
	void main (void) {
	requete[1]= 1;
	tour = ***;
	while ((requete[0] == 1)
	&& (tour == ***)) {};
	/* debut section critique */
	...
	/* fin section critique */
	requete[1]= 0;
	...
	}

2. Compléter le programme (uniquement les affectations et les tests sur la variable tour) et montrer que cette solution est correcte.
3. Quel est le défaut de cette solution ?

Exercice n°3: (3 Points)

1. Comment pourrait-on utiliser un sémaphore de sorte à ce que un processus se bloque si un événement n'a pas encore eu lieu, cet événement étant provoqué par un autre processus?

2. Soit l'ensemble de processus suivant, où les contraintes de précédence sont données par le graphe.



(a) Donner une solution utilisant des sémaphores et permettant de synchroniser ces processus de manière à respecter les contraintes de précédence.

(b) Donner une solution de même type n'utilisant pas plus de 3 sémaphores.

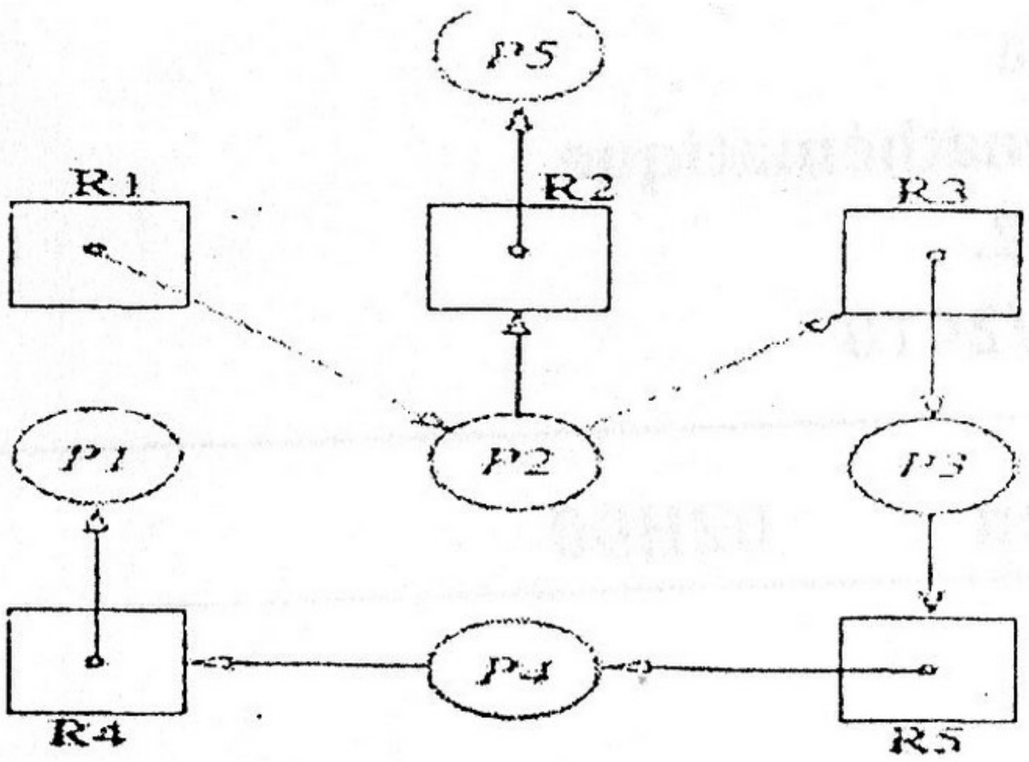
Exercice n°4: (6 Points)

Supposer qu'on dispose d'une seule salle de bain qui peut être utilisée par des hommes ou des femmes mais pas au même moment.

1. Donner une solution à ce problème. Il faut permettre au plus 4 personnes (du même sexe) en même temps d'être dans la salle de bain. La solution doit assurer l'exclusion mutuelle demandée et l'absence d'inter-blocage mais peut ne pas être équitable.
2. Modifier la solution proposée pour assurer l'équité.
3. même question que (2) mais en utilisant un **Moniteur**.

Exercice n°5: (2 Points)

1. Le graphe d'allocation de ressources pour un système à un moment donné est le suivant:



Y a-t-il risque d'inter-blocage en ce moment? Si oui, justifier. Si non, modi
graphe en ajoutant une flèche pour que le risque d'inter-blocage existe.
2 .On considèrent un système composé de 4 ressources identiques qui sont
partagées par 3 processus. Chacun des processus utilise, au plus, 2 ressourc
Montrez qu'un Interblocage est impossible dans un tel système.

Exercice n°6: (3 Points)

4 processus (numérotés de 1 à 4) partagent des ressources de 3 types différe
ou C) ; il existe 2 ressources de type A, et 5 de type B, et 3 de type C.
Au cours de son exécution, chaque processus a besoin d'un nombre de ress
donné

On considère le système dans l'état représenté par les deux matrices :

	A	B	C
1	1	2	1
2	0	1	0
3	0	0	1
4	0	1	1

(a) Ressources allouées

	A	B	
1	2	2	
2	0	3	
3	1	2	
4	0	1	

(b) Besoins maxima

1. En utilisant l'algorithme du banquier, vérifié si l'état courant du sys
   sûr ou risqué. Justifier votre réponse.
2. Pour chaqu'une des demandes suivantes dite qu'elle devrait être acc
   ou mise en attente.
   a. Le processus P1 fait une demande pour une ressource A.
   b. Le processus P3 fait une demande pour une ressource A.
   c. Le processus P4 fait une demande pour une ressource B.