Département d'Informatique - Faculté d'Electronique et Informatique - USTHB
Master 1 RSD - Module système d'Exploitation

31/01/2010

Examen

Durée : 1h30

Exercice 1 : (7 points)

Nous considérons un algorithme de scheduling basé sur la priorité dynamique. Les processus arrivent dans la file ***F1*** avec une priorité initiale ***k***. Le temps CPU est partagé (round Robin avec priorité) avec un quantum de 10ms et un top horloge toutes les 2ms.

Chaque processus qui libère le CPU avant la fin de son quantum de temps voit sa priorité augmenter de 1 (il devient plus prioritaire).

1 . Quel est l'objectif de cette stratégie (2lignes au maximum)?
2 . Quel est l'inconvénient de cette stratégie (2lignes au maximum)?

Cette stratégie est modifiée comme suit : Chaque processus qui reste en attente dans F1 pour n quantums de temps successifs voit sa priorité augmenter de 1 ?

3 . Quel serait l'objectif dans ce cas (2lignes au maximum)?
4 . Proposer une structure de pcb dans ce cas ?
5 . Ecrire les programmes : svc, scheduler, routine_horloge.

Exercice 2 : (5 points)

Soit la chaine de référence ***C***: 4 3 2 1 4 3 5 4 3 2 1 5

Nous définissons M(P,c, T) comme étant la liste des pages qui doivent être en mémoire pour une chaîne de référence P, une mémoire de taille c et un instant de référence t.

1. Appliquer l'algorithme de remplacement FIFO et LRU pour une mémoire centrale de 3 et 4 pages
2. Calculer M (P, 3, 6) et M(P, 4, 7), comparer les résultats.
3. Que peut-on dire de M(P,c, T) et M(P,c+1, T) ?

Exercice 3: (8 points)

On suppose un fichier de taille importante sous le SGF UNIX. La taille des tables d'index est de 256 entrées. Un fichier est lu bloc entier. Le délai d'un transfert entre le disque et le mémoire centrale est de 40ms.

1- Quelle est la taille maximale d'un fichier ?

On veut accéder au bloc 329 du fichier.

2- Dans le où le système ne dispose pas de buffer cache, quelle est la chaîne des accès à effectu pour accéder à ce bloc (c-à-d la liste des blocs accédés pour pouvoir accéder au bloc 329).
3- Quel est le temps d'accès au bloc 329?

On suppose que le système dispose d'un buffer cache. La taille de ce buffer cache est de 2 blocs et il e géré en LRU.

4- Donner les états successifs du buffer cache.
5- Quel est le temps d'accès au bloc 329 dans ce cas ?
6- Quel est le temps nécessaire pour accéder successivement aux blocs 329 et 330.

Correction Exam

Exercice 1 : (7 points)

Nous considérons un algorithme de scheduling basé sur la priorité dynamique. Les processus arrivent dans l: avec une priorité initiale *k*. Le temps CPU est partagé (round Robin avec priorité) avec un quantum de 10r top horloge toutes les 2ms.
Chaque processus qui libère le CPU avant la fin de son quantum de temps voit sa priorité augmenter de 1 (il plus prioritaire).

1 . Quel est l'objectif de cette stratégie (2lignes au maximum)? (0,5)

Prise en compte de processus prioritaire avec partage équitable de CPU entre processus de m priorité. Compenser les processus orientés E/S par rapport aux retards induits par ces opérations d'I

2 . Quel est l'inconvénient de cette stratégie (2lignes au maximum)? (0,5)

Famine d'autant plus que les processus prioritaire peuvent augmenter encore leur priorité sans limit

Cette stratégie est modifiée comme suit : Chaque processus qui reste en attente dans F1 pour n quantums de te successifs voit sa priorité augmenter de 1 ?

3 . Quel serait l'objectif dans ce cas (2lignes au maximum)? (0,5)

Remédier au problème précédent.

4 . Proposer une structure de pcb dans ce cas ?(1pt)

Id
Etat
Mot Etat
Prio
Nb QuantumAttente
....
SVT

5 . Ecrire les programmes : svc, scheduler, routine_horloge. (1,5 + 1,5 + 1 ,5)

✓**Routine it-Horloge()**

```
{ Save Contexte;
  Q --;
 Si Q = 0
 Alors
      p=teteF1 ;
      Tant que p !=Null faire
        Si p.NbQuantumAttente ++ = n alors
          p.priorité++ ;  p.nbQuantum Attente=0 ;
      fsi ;
      TrierFile(F1) ;
      réquisition = 1;     LPSW(m-sched);
 Sinon                Restaurer contexte;
 Fsi;
}
```

✓**SVC(cause, ...)**

```
{ Save contexte;
  Case cause
        Arrivée:/*Créer  pcb processus et l'initialiser par les paramètres du processus */
              Allouer (pcb);
              pcb→id = id;
              pcb→nom = nom;
              pcb→etat= "prêt";
              pcb→MotEtat = motEtat;
              pcb→cpu = 0;
              pcb→prio= k;
              pcb→NbQuantumAttente= 0;
              pcb→svt = Null;

              /* insérer pcb dans File Prêt */
```

```
                Si TeteF1 = Null
                alors        Tete F1= pcb;
                Sinon                QueueF1-->svt = pcb;
                fsi;
                QueueF1= pcb;
                Si actif=null et les files prêt vides
                Alors  LPSW(m-sched)

Fin:            LibérerRessources();
                Actif →etat ="terminé"; Désarmer-H()
                Libérer(pcb);  /* libération de l'espace alloué au pcb  */
                Réquisition =0;  LPSW(m-sched);
Demande E/S:
                Actif→etat = "bloqué";  Désarmer-H()
                /* Construire requête d'E/S' */
                Actif→prio++
                Actif→ NbQuantumAttente = 0 ;
                Alouer(Req);    /*descripteur requête E/S*/
                Req→op= op;
                Req→adr = adr;  /* adresse source ou destination des données à transférer */
                .....
                /* insérer pcb dans File Bloque  */
                Si TeteBloque = Null
                alors        TeteBloque= pcb;
                Sinon                QueueBloque -->svt = pcb;
                fsi;
                QueueBloque= pcb;

                Si périphérique libre Alors lancer-E/S(Req);  fsi;
                Réquisition =0;  LPSW(m-sched);
        ////
        Fincase;
           RestaurerContexte;
                FinSVC.
```

Scheduling à Temps partagé avec priorité dynamique.

```
Scheduler ()
{               / * Réquisition   */
  Si actif = null
  alors  /* Réquisition CPU  */
          actif→etat ="prêt";
          actif→NbQuantumAttente = 0;

     insérer le processus dans F1 en respectant le tri selon priorité décroissante à partir de la
  tête;
   fsi
                /* Allocation PC  */
  Actif = NULL;
  Si  TeteF1 !=  NULL
   Alors
        Actif = TeteF1;
        actif→etat =actif;
          Si QueueF1 =  TeteF1   alors QueueF1 = NULL;
        Fsi;
          TeteF1 = TeteF1→svt;
        Q = 5;
        LPSW(actif→MotEtat);   /* lancement nouveau processus et sortie de boucle pour*/
```

```
  Fsi;
Si actif = NULL alors  /* F1 vide*/
   Attendre();
Fsi;
}
```

Exercice 2 : (5 points)

Soit la chaine de référence *C*: 4 3 2 1 4 3 5 4 3 2 1 5

Nous définissons M(P,c, T) comme étant la liste des pages qui doivent être en mémoire pour une chaîne de référence P, une mémoire de taille c et un instant de référence t.

1. Appliquer l'algorithme de remplacement FIFO et LRU pour une mémoire centrale de 3 et 4 pages(0,75x 4 pts)

	4	3	2	1	4	3	5	4	3	2	1	5	
CH:	4	3	2	1	4	3	5	4	3	2	1	5	
	4	4	4	1	1	1	5			5	5	5	FIFO
		3	3	3	4	4	4			2	2	2	Taux DP=9/12
			2	2	2	3	3			3	1	1	
	DP	DP	DP	DP	DP	DP	DP			D	DP		
	4	4	4	4	4	4	5	5	5	5	1	1	FIFO
		3	3	3	3	3	3	4	4	4	4	5	Taux
			2	2	2	2	2	2	3	3	3	3	DP=10/12
				1	1	1	1	1	1	2	2	2	
	DP	DP	DP	DP			DP	DP	DP	DP	DP	DP	
	4	4	4	1	1	1	5		5	2	2	2	LRU
		3	3	3	4	4	4			4	1	1	Taux
			2	2	2	3	3			3	3	5	DP=10/12
	DP	DP	DP	DP	DP	DP	DP			D P	DP	DP	
	4	4	4	4	4	4					4	5	FIFO
		3	3	3	3	3	3					3	Taux
			2	2	2	2	5	5	5	5	1	1	DP=8/12
				1	1	1	1	1	1	2	2	2	
	DP	DP	DP	DP			DP			DP	DP	DP	

2. Calculer M (P, 3, 6) et M(P, 4, 7), comparer les résultats. (1pt)

M (P, 3, 6)=1,4,3 M(P, 4, 7)= 5, 3, 2, 1 → FIF0

M (P, 3, 6)=1,4,3 M(P, 4, 7)=4, 3,5, 1 → LRU

M (P, 3, 6)fifo Non Inclu dans M(P, 4, 7)fifo → FIF0

M (P, 3, 6)lru Inclu dans M(P, 4, 7) → LRU

3. Que peut-on dire de M(P,c, T) et M(P,c+1, T) ? (1pt)

M(P,c, T) Inclu M(P,c+1, T) pour LRU

Nb DP de M(P,c, T) >= M(P,c+1, T) pour LRU

M(P,c, T) Non Inclu M(P,c+1, T) pour FIFO

Nb DP de M(P,c, T) ?? M(P,c+1, T) pour FIFO

Exercice 3: (8 points)

On suppose un fichier de taille importante sous le SGF UNIX. La taille des tables d'index est de 256 entrées. Un fichier est lu bloc entier. Le délai d'un transfert entre le disque et le mémoire centrale est de 40ms.

1- Quelle est la taille maximale d'un fichier ?(1pt)

10 + 256 + 256*256 + 256*256*256

On veut accéder au bloc 329 du fichier.

2- Dans le où le système ne dis ɔose r buffer cache, quelle est la chaîne des accès
pour accéder à ce bloc (c-à-d la liste des blocs accédés pour pouvoir accéder au bloc
Inode → Table index double indirection Niv0 → Table index double indirection Niv1 →
3- Quel est le temps d'accès au bloc 329? (1pt) suite au résultat de question précédente

On suppose que le système dispose d'un buffer cache. La taille de ce buffer cache est de 2 est géré en LRU.

4- Donner les états successifs du buffer cache. (1 + 1 pt)

Contenu Buffer Cache

T. IndexDoubleniv0	Bloc329	Bloc329
T. IndexDoubleniv1	T. IndexDoubleniv1	Bloc330

5- Quel est le temps d'accès au bloc 329 dans ce cas ? (1pt) 160ms

6- Quel est le temps nécessaire pour accéder successivement aux blocs 329 et 330. 1pt)