

Cours : (3 pts)

1. Quel est l'avantage d'une table de pages multi-niveaux par rapport à une table de pages simple 1pt
2. Expliquer le déroulement d'une E/S depuis l'utilisateur jusqu'à la fin de l'E/S 1pt
3. Citez les conditions de Kaufmann (conditions des interblocages) 1pt
4. Donner un script qui prend un nom comme argument, et le nom est répertoire, il liste son contenu, si le nom est un fichier il affiche son contenu. 1pt

Processus (4 pts):

Soit le tableau suivant :

Processus	Temps d'arrivés	Temps CPU	Mémoire	E/S
1	8 :00	5	1MB	(1,3)
2	8 :05	4	2MB	(2,2), (4,2)
3	8 :10	4	2MB	(1,2), (3,2)
4	8 :10	5	1MB	(2,2)

La mémoire réservée aux processus étant de 3 MB, le système utilise le SETF avec $q=1$ et les E/S de la dernière colonne du tableau, les E/S se font en FCFS.

Donner la séquence des événements en prenant en compte qu'un processus peut suspendre un autre pour défaut de mémoire. Le processus qui demande une E/S est toujours suspendu jusqu'à la fin de son E/S. Le processus suspendu parmi les prêts est celui ayant pris le moins de quantum. Si la mémoire reste insuffisante on suspendra autant de processus jusqu'à ce la mémoire soit disponible. Le processus sélectionné pour retourner en mémoire est celui qui a le plus de quantum.

Mémoire (4 pts):

Soit le mécanisme de pagination et soit la description de la table des pages :

V	PROT	M	R	NPP	C _{chargement}	C _{dernier accès}
1	3	1	1	2	2	2

Le bit V indique si la présence de la page en mémoire. Le bit M indique la modification de la page. Le bit R indique si la page a été dernièrement référencée et C_{dernier accès} est le compteur de cette référence. C_{chargement} est le compteur de son chargement en mémoire. La PROT 000 est une page verrouillée contre la lecture et l'écriture, 001 contre la lecture, 010 contre l'écriture, 100 est une page non exécutable, 111 est totalement accessible. Une page est de 1 KB. La table des pages a 64 entrées. Les premières entrées de la table des pages ont les contenus suivants :

N° d'entrée	Description de la page
0	A02
1	FA5
2	612
3	F3F
4	E4A
5	
...	

1. Donner la taille en bits des champs :

Adresse virtuelle, Adresse physique, le déplacement, le NPV et le NPP. Donner la taille des espaces virtuels et physique ainsi que la taille de la table des pages.

2. Pour les adresses virtuelles. Dire si l'accès est possible, calculer l'adresse physique et indiquer aussi les modifications de la table des pages.

- a. (031C)₁₆ en lecture
- b. (05AA)₁₆ en écriture
- c. (0856)₁₆ en lecture
- d. (0FFF)₁₆ en lecture

3. Soit la liste des adresses virtuelles en hexa suivantes :

(0856)_E, (0134)_L, (104E)_L, (096A)_L, (0481)_E, (0C54)_E, (027C)_L, (12B9)_L, (071A)_L

Trouver la chaîne ω correspondante à cette suite d'adresses.

Déterminer le nombre de défaut de pages par : OPT, NRU, LRU avec compteur et la 2^{ème} chance en utilisant la tables des pages ci-dessus. Modifier la table des pages à chaque référence. Le C_{dernier accès} s'incrémente de 1 à chaque accès pour toutes les pages. Seul C_{chargement} de la page chargée qui devient 11 pour les autres il est décrémenté de 1.

Fichiers (4 pts):

On considère un système d'allocation de blocs par indexage multi-niveaux. Le 1^{er} bloc contient X pointeurs directs, et le système dispose de plusieurs pointeurs d'indirection. Supposons que plusieurs disques sont rattachés à la machine, que le même système de fichier s'étant sur tous les disques pour former un volume cohérent, et qu'un bloc de données d'un fichier peut être alloué sur n'importe quel disque. On considère que le nombre de pointeurs par bloc est de 256. La taille maximale d'un disque est de 16GB. La taille maximale d'un volume est 4096GB. La taille maximale d'un fichier est de 17 247 251 456 Bytes. On considère aussi que les Y bits les plus à droite d'un pointeur contiennent l'adresse d'un bloc de données et que les Z bits restant (les Z bits les plus à gauche d'un pointeur) contiennent l'adresse du disque sur lequel se trouve le bloc de données.

1. Quel est le nombre Z de bits servant de pointeur aux disques.
2. Quel est le nombre Y de bits servant de pointeur de données d'un disque,
3. Quelle est la taille d'un bloc.
4. Quelle est la valeur de X et combien il y a-t-il de pointeurs d'indirection.
5. Dites où se trouve dans votre système le byte 361245.

Interblocages (4 pts) :

Soit l'état suivant :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	0	2	0	3	3	2
P1	3	2	2	2	0	0	1	0	2			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	2	3	0	0	2	2	1	0			

On désire appliquer une méthode (Éviter) les interblocages sans retarder pour autant les requêtes. Pour cela, on accepte la requête, on effectue les mises à jour des matrices et on vérifie si l'état est sûr si oui on procède normalement. Si la requête est supérieure à Available ou que l'état n'est pas sûr, on retire des ressources aux processus d'indice supérieur à celui du processus ayant émis la requête dans un ordre croissant. Sinon on vérifie s'il y a des processus qui ont atteint leur Max auxquels on restitue toutes les ressources puisqu'ils vont terminer normalement.

- a. L'état du système est-il sûr ?
- b. Les requêtes sont émises dans l'ordre P0, P1, P3 traiter chaque cas ?

Cours :

1. Avantage d'une table de pages multi-niveaux par rapport à une table de pages simple (cocher la réponse correcte)

- ☒ diminue le surcoût en mémoire, diminue le temps d'accès
- ☐ diminue le surcoût en mémoire, augmente le temps d'accès
- ☐ augmente le surcoût en mémoire, diminue le temps d'accès
- ☐ augmente le surcoût en mémoire, augmente le temps d'accès

2. Expliquer le déroulement d'une E/S depuis l'utilisateur jusqu'à la fin de l'E/S.

1. L'utilisateur émet une E/S (commande Shell ou appel à partir d'un programme)
L'utilisateur initie son E/S
Son processus passe à l'état bloqué en attente de fin d'E/S.
2. Exécution de l'appel système (changement de mode).
On change du mode utilisateur (user) au mode noyau (kernel)
On accède à la table des primitives système.
On exécute la primitive en question qui initialise le pilote (driver)
L'appel système se bloque en attente de la fin du pilote.
3. Exécution du pilote (driver)
Le pilote traite la requête en initialisant le contrôleur
Le pilote se bloque en attente du contrôleur
4. Le contrôleur traite l'E/S
Le contrôleur initialise l'E/S
Il émet une interruption pour prévenir le système de la fin de l'E/S.
5. Le Scheduler débloquent le pilote qui s'acquiesce de l'E/S (1^{ère} vérification des erreurs de transfert)
6. Le Scheduler débloquent l'appel système qui s'acquiesce de l'E/S (2^{ème} vérification)
7. Le Scheduler débloquent le processus utilisateur qui continue en séquence.

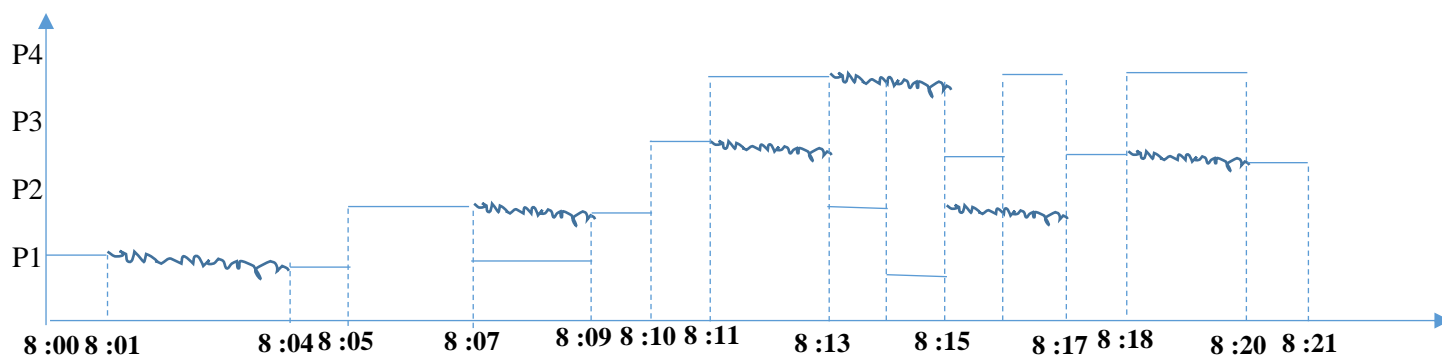
3. Citez les conditions de Kaufmann (conditions des interblocages)

- ☐ Exclusion mutuelle
- ☐ Détection et attente
- ☐ Pas de préemption
- ☐ Attente circulaire

4. Script :

```
for $1 in $* do
  if [ -d $1 ] then
    echo "using ls"
    ls $file
  else cat $1e
fi
done
```

Processus :



Mémoire : 1. Taille des champs, des espaces et de la table des pages

Adresse	Adresse	NPV	NPP	Déplacement	Espace	Espace	Table des
16 bits	12 bits	6	2	10 bits	64KB	4KB	96B

2. calcul des adresses physiques et Modification de la table des pages:

Adresse Virtuelle	Possible	NPV	V	PROT	Déplacement	NPP	Adresse Physique
031C en lecture	oui	0	1	010	11 0001 1100	00	31C
05AA en écriture	oui	1	1	111	01 1010 1010	10	9AA
0856 en lecture	non	2	0				
0FFF en écriture	oui	3	1	111	11 1111 1111	11	FFF

N° d'entrée	Description de la page
0	A93
1	FE6
2	612
3	FFF
4	E4A

3. Chaîne ω :

(0856) _E	(0134) _L	(104E) _L	(096A) _L	(0481) _E	(0C54) _E	(027C) _L	(12B9) _L	(071A) _L
2	0	4	2	1	3	0	4	1

OPT:

ω	2	0	4	2	1	3	0	4	1
0		0					0		
1					1				1
3	2			2		3			
4			4					4	
	D				D				

NRU:

ω	2	0	4	2	1	3	0	4	1
0	2			2					
1					1				1
3		0					0	4	
4			4			3			
	D	D				D		D	

LRU:

ω	2	0	4	2	1	3	0	4	1
3					1				1
0		0				3			
4			4				0		
1	2			2				4	
	D				D	D	D	D	

2^{ème} chance:

	2	0	4	2	1	3	0	4	1
3						3			
4			4					4	
1		0			1				1
0	2			2			0		
	D	D					D		

Fichiers :

Z : sert de pointeurs de disques, on doit donc calculer le nombre de disque.

Nombre de disques = volume/disque = $4096\text{GB}/16\text{GB} = 2^{42}/2^{34} = 2^8 = 256$ disques et $Z = 8$

Y : sert de pointeur de données dans un disque. Donc :

$2^Y \times \text{taille d'un bloc} = 16\text{GB}$ et un pointeur est un multiple du Byte. Taille du pointeur = $Z + Y$

Si taille du pointeur = 8 donc $Y=0$, impossible.

Si taille du pointeur = 16 donc $Y=8$, et taille du bloc = $256 \times 2\text{ B} = 512\text{ B}$. Le disque = $2^8 \times 512\text{ B} \neq 16\text{GB}$

Si taille du pointeur = 24 donc $Y=16$, et taille du bloc = $256 \times 3\text{ B} = 768\text{ B}$. Le disque = $2^{16} \times 768\text{ B} \neq 16\text{GB}$

Si taille du pointeur = 32 donc $Y=24$, et taille du bloc = $256 \times 4\text{ B} = 1\text{KB}$. Le disque = $2^{24} \times 1\text{KB} = 16\text{GB}$

Donc $Y = 24$ bits.

X : Taille du fichier = 17 247 251 456 =

$$= X \times 1\text{KB} + 256 \times 1\text{KB} + 256 \times 256 \times 1\text{KB} + 256 \times 256 \times 256 \times 1\text{KB} = X \times 1024 + 262\,144 + 67\,108\,864 + 17\,179\,869\,184 = X \times 1024 + 17\,247\,240\,192$$

Donc $X = (17\,247\,251\,456 - 17\,247\,240\,192) / 1024 = 11$

Le byte 361245 se trouve : $361\,245 / 1024 = 352$ avec un reste de 797. Ce reste est le déplacement dans le bloc de données en question. On commence par soustraire les 11 pointeurs simples = $352 - 11 = 341$.

Comme $352 > 256$, on retire les pointeurs de 1^{ère} indirection et on obtient $341 - 256 = 85$.

Donc, on est au 1^{er} pointeur de 2^{ème} indirection à l'entrée 85.

Résultats:

Z	Y	X	Taille d'un bloc	Nombre d'indirection	Le byte 361245
8	24	11	1K	3 (Simple, double, triple)	85 ^{ème} entrée du 1 ^{er} pointeur de 2 ^{ème} indirection avec un déplacement = 797

Interblocages

a. État sûr : OUI

b. Requêtes :

1. P0 : (0 2 0)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	0	0	0	3	1	2
P1	3	2	2	2	0	0	1	0	2			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	2	3	0	0	2	2	1	0			

Etat sûr : OUI

P3	P1	P0	P2	P4
----	----	----	----	----

2. P1 : (1 0 2)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	0	0	0	2	1	0
P1	3	2	2	3	0	2	0	0	0			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	2	3	0	0	2	2	1	0			

Etat sûr : NON

P	P	P	P	P
---	---	---	---	---

3. P3 : (0 2 1) = (0 1 1) + (0 1 0)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	0	0	0	2	2	0
P1	3	2	2	3	0	2	0	0	0			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	0	1	0	2	1			
P4	4	2	3	0	0	2	2	1	0			

État sûr : NON

P	P	P	P	P
---	---	---	---	---

(0 2 1) ≠ Available, on doit chercher C de P4

P1	P3	P0	P2	P4
----	----	----	----	----

Remède : Retirer Ressources :

Ressources	Nombre	Processus

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0												
P1												
P2												
P3												
P4												

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

Remède : Retirer Ressources :

Ressources	Nombre	Processus
B	1	P3

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	0	0	0	2	2	0
P1	3	2	2	3	0	2	0	0	0			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	0	1	0	2	1			
P4	4	2	3	0	0	2	2	1	0			

Etat sûr : OUI

P1	P3	P0	P2	P4
----	----	----	----	----

Remède : Retirer Ressources :

Ressources	Nombre	Processus
C	1	P4

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	0	0	0	2	0	0
P1	3	2	2	3	0	2	0	0	0			
P2	9	0	2	3	0	2	0	0	0			
P3	2	2	2	2	2	2	0	0	0			
P4	4	2	3	0	0	1	2	1	1			

Etat sûr : OUI

P3	P1	P0	P2	P4
----	----	----	----	----

NOM :PRENOM :GR :.....

Cours :

1. Avantage d'une table de pages multi-niveaux par rapport à une table de pages simple (cocher la réponse correcte)

- ☐ diminue le surcoût en mémoire, diminue le temps d'accès
- ☐ diminue le surcoût en mémoire, augmente le temps d'accès
- ☐ augmente le surcoût en mémoire, diminue le temps d'accès
- ☐ augmente le surcoût en mémoire, augmente le temps d'accès

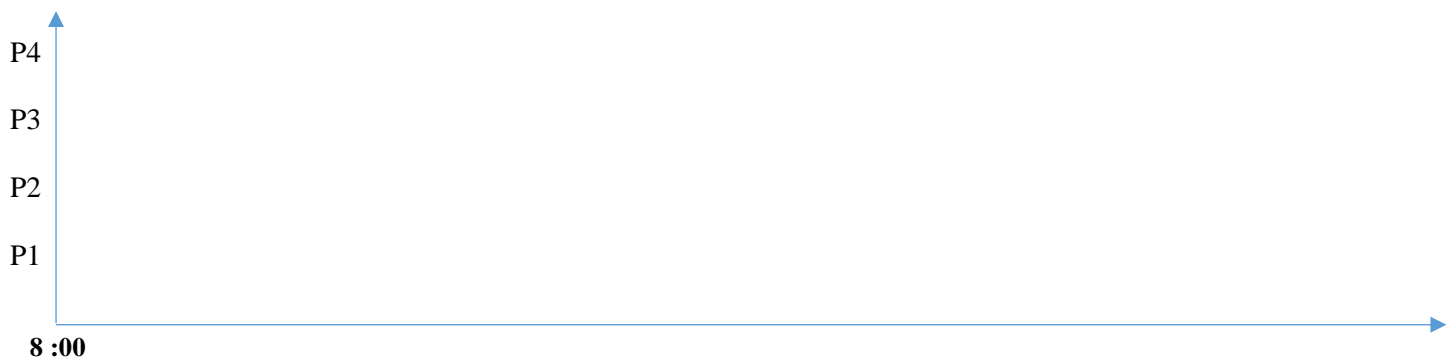
2. Expliquer le déroulement d'une E/S depuis l'utilisateur jusqu'à la fin de l'E/S.

3. Citez les conditions de Kaufmann (conditions des interblocages)

- ☐ Exclusion mutuelle
- ☐ Pas de préemption
- ☐ Détention et attente
- ☐ Attente circulaire

4. Script :

Processus :



Mémoire : 1. Taille des champs, des espaces et de la table des pages

Adresse Virtuelle	Adresse Physique	NPV	NPP	Déplacement	Espace Virtuel	Espace physique	Table des pages
bits	bits	bits	bits	bits	64KB	4KB	96B

2. calcul des adresses physiques :

Adresse Virtuelle	Possible(oui/non)	NPV	V	PROT	Déplacement (bits)	NPP	Adresse Physique (hexa)
031C en lecture							
05AA en écriture							
0856 en lecture							
0FFF en écriture							

Modification de la table des pages

N° d'entrée	Description de la page (hexa)
0	
1	
2	
3	
4	

4. Chaîne ω :

(0856) _E	(0134) _L	(104E) _L	(096A) _L	(0481) _E	(0C54) _E	(027C) _L	(12B9) _L	(071A) _L

OPT:

ω									
0								..	
1								..	
3								..	
4								..	

NRU:

ω									

LRU:

ω									

2^{ème} chance:

ω									

Fichiers :

Calcul

Resultats:

Z	Y	X	Taille d'un bloc	Nombre d'indirection	Le byte 361245

Interblocages

a. État sûr : OUI / NON

b. Requêtes :

1. P0 : (0 2 0)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3				0	0	0			
P1	3	2	2									
P2	9	0	2									
P3	2	2	2									
P4	4	2	3									

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

2. P1 (1 0 2)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3				0	0	0			
P1	3	2	2				0	0	0			
P2	9	0	2									
P3	2	2	2									
P4	4	2	3									

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

3. P3 (0 2 1)

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3				0	0	0	2	2	0
P1	3	2	2				0	0	0			
P2	9	0	2				0	0	0			
P3	2	2	2				0	0	0			
P4	4	2	3									

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

P	P	P	P	P
---	---	---	---	---

Remède : Retirer Ressources :

Ressources	Nombre	Processus

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0												
P1												
P2												
P3												
P4												

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

Remède : Retirer Ressources :

Ressources	Nombre	Processus

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3				0	0	0			
P1	3	2	2				0	0	0			
P2	9	0	2									
P3	2	2	2									
P4	4	2	3									

Etat sûr : OUI / NON

P	P	P	P	P
---	---	---	---	---

Remède : Retirer Ressources :

Ressources	Nombre	Processus

Nouvel état :

job	Max			Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3				0	0	0			
P1	3	2	2				0	0	0			
P2	9	0	2				0	0	0			
P3	2	2	2				0	0	0			
P4	4	2	3									

