Devoir surveillé ☑	Examen ☐	Session : principale ☑ de contrôle ☐
Matière : **Systèmes d'Exploitation** Enseignant(s) : **Lilia SFAXI** Filière(s) : **GL2/IMI2** Nombre de pages : 7		Semestre: **1** Date: **Octobre 2018** Durée: **1h30** Documents : autorisés ☐ non autorisés ☑

EXERCICE 1. Pagination & Segmentation (10 pts)

Soit un système d'exploitation qui gère la mémoire en utilisant le principe de segmentation paginée. Ce système respecte les critères suivants :

- Architecture : 32 bits
- Taille de la RAM : 1 Go
- Taille d'une page : 8 Ko
- Une adresse permet d'indexer un octet.

Soit un processus T disposant de trois segments : S0, S1 et S2. Ces segments ont pour tailles effectives (taille du contenu du segment) respectives 20 Ko, 36 Ko et 2 Ko.

1. Calculer la fragmentation interne de ce processus.

S0 : 20 div 8 = 2 => 3 pages, avec une fragmentation de 4 Ko

S1 : 36 div 8 = 4 => 5 pages, avec une fragmentation de 4 Ko

S2 : 2 div 8 = 0 => 1 page, avec une fragmentation de 6 Ko

➔ Fragmentation (T) = 4+4+6 = 14Ko

2. Quelle est la taille de la table de segments du processus T ?

Taille de la table de sègments = Nombre de segments (h) * taille d'une adresse (l)

h = 3, l= 32 bits (architecture 32bits)

Taille TS = 3*32 bits = 96 bits = 12 o

3. Quelle est la taille maximale possible d'une table de pages selon cette configuration ?

Taille max TP = nb pages max (H) * (nb de bits pour coder un num de case (L)+ 1)

Nb Cases = Taille RAM / Taille Case = $2^{30}/2^{13} = 2^{17}$

→ L = 17 bits

H = taille mémoire logique / Taille page = $2^{32}/2^{13} = 2^{19}$ pages

➔ Taille max TP = 2^{19} * 18 bits = 1.125 Mo

4. Quelles sont les tailles effectives des tables de pages du processus T ?

Taille effective TP = nb pages (n) * (nb de bits pour coder un num de case (L)+ 1)

L = 17 bits (tel qu'expliqué précédemment)

Taille TP0 = 3 * 18 bits = 54 bits = 6.75 o

Taille TP1 = 5 * 18 bits = 90 bits = 11.25 o

Taille TP2 = 1 * 18 bits = 18 bits = 2.25 o

5. En supposant les affectations suivantes (Si-Pj → Ck veut dire que la page j du segment i est chargée dans la case k) :

S0-P1 → C19 S1-P2 → C12 S1-P3 → C3 S2-P0 → C2

a. Donner les tables de pages de T

S0

	Case	*BP*
0	0	0
1	19	1
2	0	0

S1

	Case	*BP*
0	0	0
1	0	1
2	12	1
3	3	1
4	0	0

S2

	Case	*BP*
0	2	1

b. Donner l'adresse physique associée à l'adresse logique décimale (S1, 30720)

30720 div 8192 = 3 → Page 3 de S 1 → case 3

30720 mod 8192 = 6144 → déplacememt

→ @physique = 3*8192 + 6144 = 30720

6. Supposons que nous conservons les mêmes données (et le même processus T), mais que nous utilisons maintenant la pagination simple.
   a. Donner la valeur de la fragmentation interne de T

Taille T : 20 + 36 + 2 = 58 Ko = 59392 o

Taille T div taille page= 58 div 8 = 7 → 8 pages avec une fragmentation de 6 Ko

   b. Calculer la taille effective de la table de pages.

Taille effective TP = nb pages * (nb de bits pour coder une case +1) = 8*18 bits = 18 octets

   c. Comparer les deux techniques, et justifier pourquoi est-ce que les systèmes actuels utilisent la segmentation paginée plutôt que la pagination.

Technique 1 = gaspillage d'espace, table de pages + seg plus grande, nb de pages plus grand, fragmentation plus grande

Mais elle permet la réutilisation grâce à la division en sègments.

7. Supposons maintenant qu'une adresse indexe 2 octets. Quelle sera alors la taille maximale de la table de pages ?

Taille TP = nb pages (n) * (nb bits pour coder un num de cadre(L) + 1)

n = Taille ML / Taille page = $2*2^{32} / 2^{13} = 2^{20}$

nombre de cases = taille RAM / taille case = $2^{30}/2^{13} = 2^{17}$

Taille TP = 2^{20} * 18 bits = 2.25 Mo

EXERCICE 2. Ordonnancement et Mémoire contiguë (10 pts)

Soit un système utilisant une gestion de mémoire contiguë à partition fixe. La mémoire est partitionnée en deux partitions de taille 5Ko, trois partitions de taille 8 Ko, et une partition de taille 10 Ko.

Le système utilise plusieurs files d'attente pour gérer la mémoire. L'ordonnanceur à long terme utilise l'algorithme FIFO pour gérer les processus en attente de la mémoire.

L'ordonnanceur à court terme utilise un algorithme à priorité avec plusieurs files d'attente. Les priorités des processus sont définies de la plus haute priorité (valeur 0) à la plus faible (valeur 2). Chaque file d'attente utilise un algorithme d'ordonnancement différent pour gérer les processus prêts :

- Pour la priorité 0, on utilise l'algorithme du tourniquet avec Q=2
- Pour la priorité 1, on utilise l'algorithme du tourniquet avec Q=3
- Pour la priorité 2, on utilise l'algorithme FCFS

Les processus suivants se présentent au système :

Nom	***Date d'arrivée***	***Priorité***	***Taille (Ko)***	***Séquence d'exécution***
P0	0	1	3	2 CPU + 3 E/S + 2 CPU
P1	0	2	5	4 CPU + 2 E/S + 3 CPU + 1 E/S + 1 CPU
P2	2	0	7	7 CPU + 1 E/S + 1 CPU
P3	2	0	3	1 CPU + 8 E/S + 1 CPU + 2 E/S + 1 CPU
P4	3	1	10	3 CPU + 1 E/S + 3 CPU
P5	3	1	9	10 CPU
P6	4	2	2	4 CPU

1- Donner le diagramme de GANTT ainsi que les états de la mémoire, dans l'annexe, si le système utilise les critères et hypothèses suivantes :
   a. P0, P1 et P2 utilisent la même E/S (ES1) alors que P3 et P4 utilisent une autre E/S (ES2). L'ordonnancement sur les deux E/S est FIFO.
   b. Si deux processus ayant la même priorité, arrivent en même temps, ils seront classés dans la file d'attente par ordre alphanumérique.
   c. Si un processus à faible priorité, qui utilise l'algorithme du tourniquet, est en cours d'exécution, et qu'un processus prioritaire arrive dans la file d'attente, **le processus en cours doit terminer son quantum** avant de laisser le processus prioritaire s'exécuter.
2- Calculer le rendement du processeur.

Rendement du processeur = temps d'exécution effectif / temps total de traitement = 43/44 = 97.72%

3- Calculer le temps de rotation moyen en précisant la formule utilisée. Analyser le résultat.

Temps de rotation d'un processus = temps d'exécution + temps d'attente dans la file prêt = date de sortie – date d'entrée – temps d'E/S et attente d'E/S

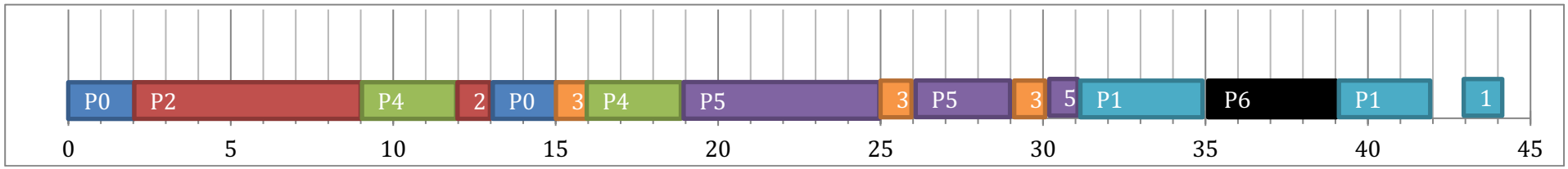
TR(P0) = 15 – 0 – 3 = 12	TR(P1) = 44 – 0 – 3 = 41	TR(P2) = 13 – 2 – 2 = 10
TR(P3) = 30 – 15 – 10 = 5	TR(P4) = 19 – 3 – 1 = 15	TR(P5) = 31 – 19 = 12
TR(P6) = 39 – 30 = 9		

TRM = (12+ 41 + 9 + 5 + 15 + 12 + 9)/ 7 = 14.857

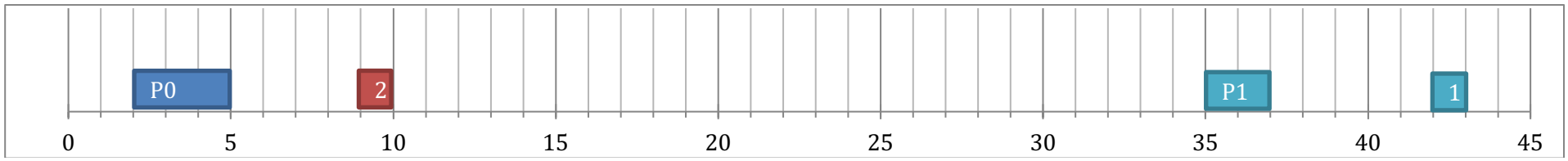
Nom et Prénom …….. CIN …… Filière ……

Annexe (Exercice 2)

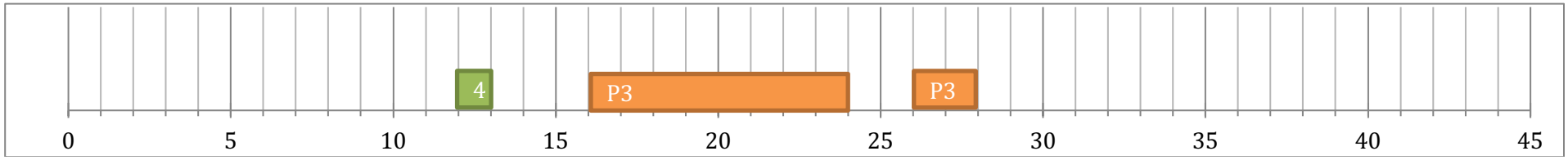
Processeur



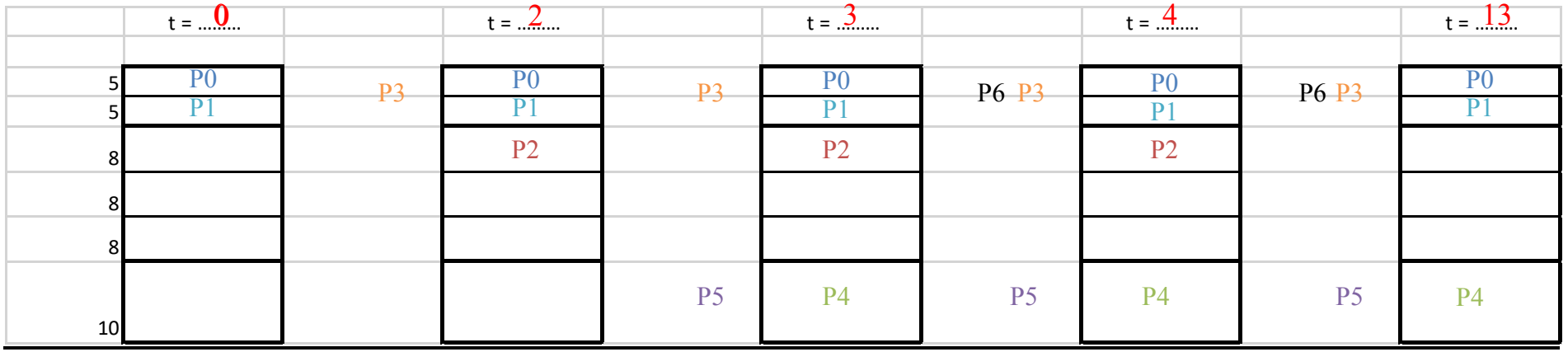
ES1

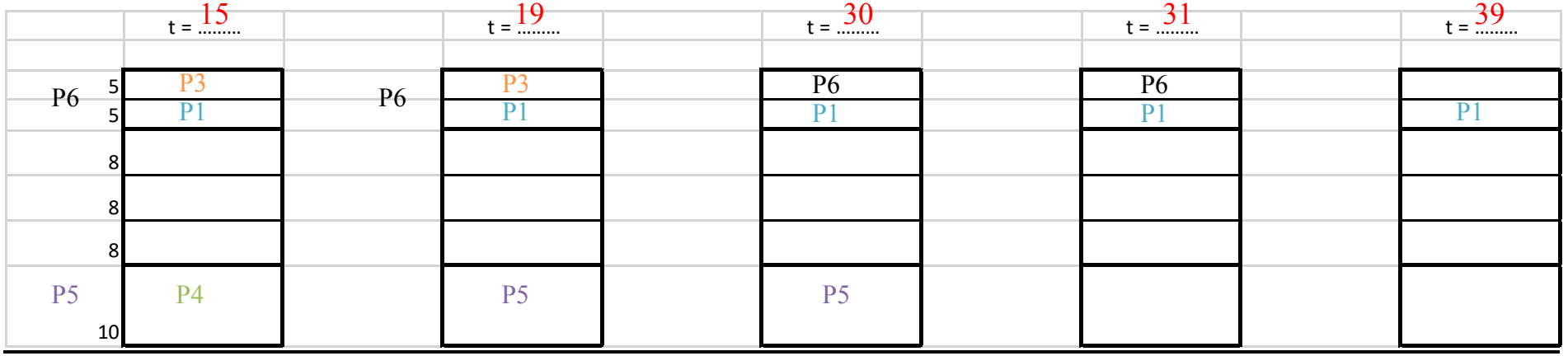


ES2



Nom et Prénom …….. CIN …… Filière ……





Mémoire vide à t=44