

Nom :	Prénom :
	Matricule :

USTHB
 Faculté d'Electronique et Informatique
 Département Informatique

Année 2015/2016, 09/01/2016
 SYS02
 Section Lic. Acad. A,B,C

EMD (Durée : 1h30)

Exercice 1 (4 pts)

On désire transférer un fichier de 3000000 octets, se trouvant en mémoire centrale, vers un disque dur. A chaque transfert de 5000 octets, le disque émet une interruption.

1. Quel est le mode d'E/S utilisé ? Justifiez votre réponse.
2. Calculer le nombre d'interruptions générées.
3. Sachant que le registre de données a une taille de 5000 octets, programmer le pilote.

Exercice 2 (8 pts)

Considérons un système d'exploitation doté d'une politique de scheduling préemptive, à base de priorité. La valeur 0 correspond à la **plus faible** priorité.

Sous ce système, les processus sont répartis en trois classes :

- **Classe HP (Haute Priorité)** : dans cette classe, le processeur est donné au processus de plus haute priorité. Ce processus peut être préempté par un processus de la même classe ayant une priorité supérieure.
- **Classe TP (Temps Partagé)** : dans cette classe, le processeur est donné au processus de plus haute priorité pour un **quantum** de temps égal à **20 ms**. La politique appliquée est celle du tourniquet. A l'arrivée, tous les processus de la **classe TP** ont la même priorité initiale **pri₀**. Le système recalcule la nouvelle priorité d'un processus P_i dans deux cas :
 1. A la fin de son quantum de temps, en utilisant la formule suivante : $Pri(P_i) = Pri(P_i) - \alpha$.
 2. Après Fin d'E/S, en utilisant la formule suivante : $Pri(P_i) = Pri(P_i) + \beta$.
- **Classe PF (Priorité Fixe)** : dans cette classe, les processus possèdent une **priorité fixe**. Le processeur est donné au processus de plus haute priorité.

Les processus de la classe **HP** sont toujours plus prioritaires que les processus des autres classes. Les processus de la classe **PF** ne seront servis que si toutes les autres classes sont vides.

Supposons que le processeur exécute le jeu de processus présenté dans la table suivante :

Processus	Classe	Instant d'arrivée (ms)	Durée d'exécution (ms)	Priorité
P1	Classe TP	0	30(10)30	20
P2	Classe TP	0	70	20
P3	Classe HP	10	30(20)10	25
P4	Classe HP	30	30	28
P5	Classe PF	30	30	10
P6	Classe PF	50	20(10)10	11

Nom :

Prénom :

Matricule :

1. Sachant que $\alpha = 2$ et $\beta = 2$, donner le diagramme d'exécution pour l'ordonnancement de ces processus.
2. Donner l'état des files d'attente aux instants : 20 ms, 80 ms, 110 ms, 155 ms, et 240 ms.
3. Pour chacun des processus de la classe TP, donner les différents instants de mise à jour de la priorité ainsi que sa valeur.
4. Donner les pseudo-codes du scheduler, de la routine IT horloge, et le SVC.

Nom : _____ Prénom : _____ Matricule : _____

Exercice 2

1. Diagramme d'exécution : = 10ms

Periph							3	3																									
CPU	1	3	3	4	4	4	3	2	2	3	1	1	2	2	2	2	1	1	1	2	6	6	5	5	5	6	X	X					

2. Etat des files d'attente :

	A l'instant 20ms	A l'instant 80ms	A l'instant 110ms	A l'instant 155ms	A l'instant 240ms
HP	∅	∅	∅	∅	∅
TP	→ P2 → P1	→ P1	→ P2	→ P1	∅
PF	∅	→ P6 → P5	→ P6 → P5	→ P6 → P5	→ P6
E/S	∅	→ P3	∅	∅	∅

3. Instants de mise à jour de la priorité et sa valeur pour les processus de la classe TP

P1	Instant	120ms	150ms	180ms		
	Valeur	18	20	18		
P2	Instant	90ms	140ms	160ms		
	Valeur	18	16	14		

Exo1:

① Le mode d'EIS utilisé est le mode direct asynchrone car le disque émet une interruption à chaque transfert d'un bloc de 5000 octets.

② le nombre d'its générées est.

$$\text{Nb_it} = \frac{\text{Taille_Fichier}}{\text{Taille_bloc}} = \frac{3000000}{5000} = \boxed{600 \text{ its}}$$

③ Pilote Asynchrone :

le sens du transfert est une sortie.

Les registres :

RD : Registre de Données d'une taille de 5000 octets

RE : Registre d'état à deux bits.

RE [0] : Bit prêt.

RE [1] : Bit erreur.

RC : Registre de commande à deux bits.

RC [0] ← sens du transfert (1: sortie, 0: entrée)

RC [1] ← pour commander le stoppage.

sortie

"
Pilote Asynchrone. Init (Nb, adr); // Nb = $\frac{3000000}{5000} = 600$

Debut

si RE[0] = 1 Alors // lancer le transfert du 1^{er} bloc

 Mov R, [adr];

 RD ← R;

 RE[0] ← 0;

 RE[0] ← 1; // sens

 RC[1] ← 1;

 sinon // mettre en attente la Dem EIS.

 enfiler (FEIS, sem) =

fin; fin;

①

^{= sortie}
Pilote - Asynchrone. Rit (Nb, adr);

Debut

<S, G, Cxt>

"RE"

si Erreur

Alors Traiter_Erreur();

si non Nb -- ;

si Nb <> 0

Alors // Lancer le transfert d'un autre bloc

adr ← adr + Taille (RD);

Mov R, [adr];

RD ← R;

RE[0] ← 0;

si non Défiler (FPB, P);

P. etat ← prêt;

Enfiler (Fp, P);

si Vide (FEIS)

Alors // Lancer une autre Demande ELS;

Défiler (FEIS, dem);

si dem sens = 0

Alors Entrée-Asynchrone. Init (dem. Nb, dem. adr)

si non sortie-Asynchrone. Init (dem. Nb, dem. adr)

~~si~~ si non RC[1] ← 0;

fin;

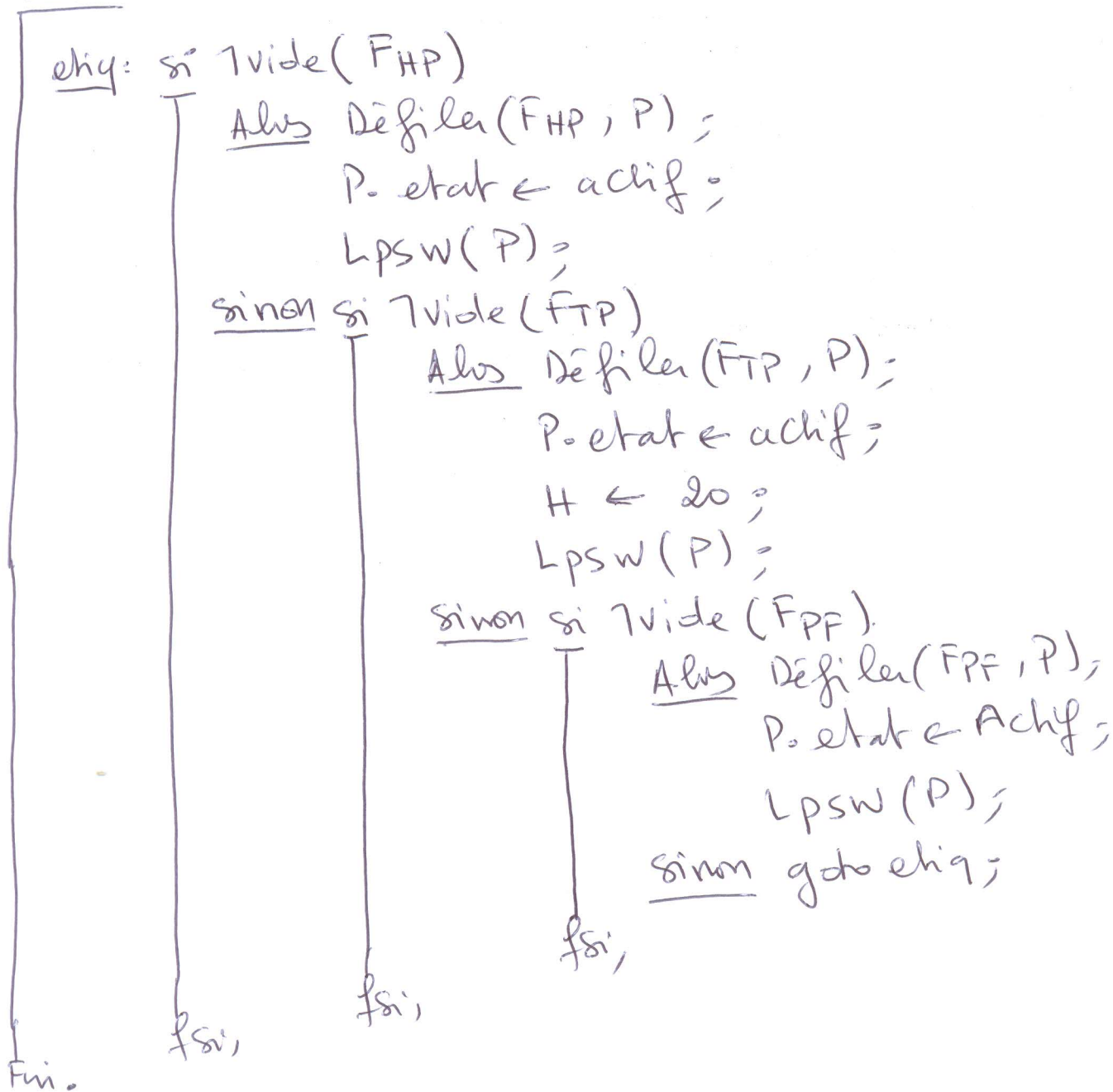
fin;

~~fin~~
fin < R, G, Cxt >

Exo 2 :

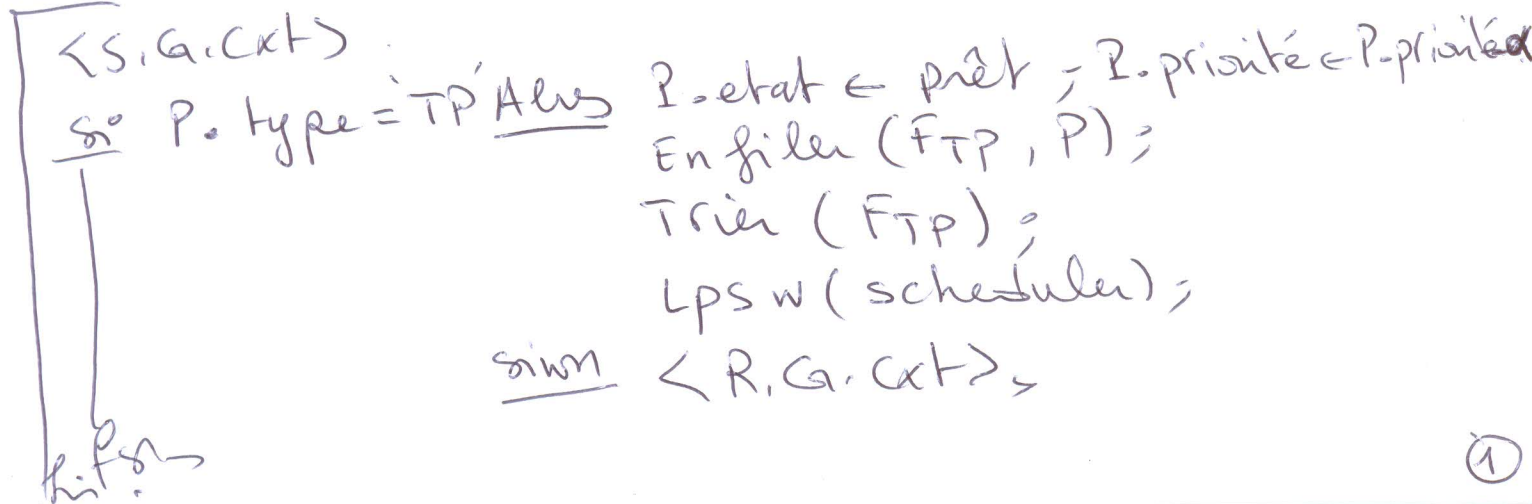
Scheduler () ;

Debut



Routine d'it-Horloge () ;

Debut



SVC (cause, - - - -)

Debut

<S.G. ext>

Case cause of

Arrivée : créer PCB

PCB. etat ← prêt ;

PCB. priorité ← Calcul-priorité (PCB);

PCB. type ← Determiner-type (PCB. priorité);

si PCB. type = 'HP'

Alors Enfiler (FHP, PCB); Trier (FHP);

si (PCB. priorité > P. priorité)

Alors P. etat ← prêt ;

si P. type = 'HP'

Alors Enfiler (FHP, P); Trier (FHP);

sinon

si P. type = 'TP'

Alors Enfiler (FTP, P); Trier (FTP);

sinon Enfiler (FPF, P); Trier (FPF);

fin;

fin;

LPSW (scheduler);

fin;

sinon si PCB. type = 'TP'

Alors Enfiler (FTP, PCB); Trier (FTP);

~~Alors~~ si (PCB. priorité > P. priorité)

Alors si P. type = PF

Alors enfiler (FPF, P);

Trier (FPF);

fin;

LPSW (scheduler);

fin;

②

①

②

② ① si un Enfiler (FPF, PCB); Trier (FPF) >

fsi;

fsi;

<R.G. act>

DemEIS: - vérifier les droits d'accès
P0 etat ← Bloqué;

Enfiler (FPB, P);

Pilote - Asynchrone. Init (N, adr);

Lpsw (scheduler);

fin pgme: - libérer les ressources allouées à P (PCB, Table -

Lpsw (scheduler);

fin case

fin