

Série N° 4 - Corrigé -
(Gestion du Processeur Central)

Exercice 1

F1 (Q=2)					
-------------	--	--	--	--	--

F2 (Q=4)			P3	P1
-------------	--	--	----	----



FB					
----	--	--	--	--	--

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Periph																		P2	P2	P2		
CPU	P1	P1	P1	P1	P2	P2	P2	P2	P3	P3	P3	P3	P1	P1	P1	P1	P2	P3	P3	P3	P3	P2
F0 (P1)		P2	P2 P3	P2 P3	P3 P1	P3 P1	P3 P1	P3 P1	P1 P2	P1 P2	P1 P2	P1 P2	P2 P3	P2 P3	P2 P3	P2 P3	P3 P1	P1	P1	P1	P1	P1 P3

F1

																				P2	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----	--

Periph
CPU

F0

F1

23	24	25	26	27	28	29	30	31	32												
P2	P1	P1	P3	P3	P3	P3	P2	P2	P2												
P1 P3	P3 P2	P3 P2	P2	P2	P2	P2															

Les programmes sont:

- Scheduler + SVC(dem E/S, arrivée, fin de processus)+ Routine fin d'E/S+ Routine fin de Q+ Init (TC1).

Scheduler ()

Debut

etiq :

Si (F1 est non vide)

Alors m_pactif =<0 ,0 ,Co_ pactif >// preparer le mot d'état

Q=2;

activer (F1, pactif) ;

Sinon

Si (F0 est non vide)

Alors m_pactif =<0 ,0 ,Co_ pactif >// preparer le mot d'etat

Q=4;

activer (F0, pactif) ;

Sinon Jump etiq // les deux vides

Fsi ;

<p>Init ()</p> <p><SC> préparer les mots d'état des pgmme connus m_h=<1 ,1 ,Co_RH > m_svc=<1 ,1 ,Co_svc > // m_Rf=<1 ,1 ,Co_Rf > // m_sch=<0 ,0 ,Co_sch > Lpsw(m_ sch); <RC></p>	<p>Routine horloge () // s incrémente chaque 1 ms</p> <p><SC> Q1--; Q2--; Si Q1=0 alors Pactif->etat="prêt"; Enfiler(F0, pactif) Lpsw(m_ sch);</p> <p>Si Q2=0 alors Pactif->etat="prêt"; Enfiler(F0, pactif) Lpsw(m_ sch);</p> <p><RC></p>	<p>SVC()</p> <p><SC> Switch case..... Dem E/S: si periph est libre (+0.25)... Arrivée : P=allouer(PCB); P->nom=nomE; P->idf= allouer-id(); P->etat="prêt"; P->mot_etat=<0, 0, adr>; Enfiler(F0, P); Fin pgme (): librer_ ressources Lpsw(m_ sch);</p> <p><RC></p>
<p>Routine fin E/S ()</p> <p><SC> Pactif->etat="prêt"; Enfiler(F1, pactif): Débloquer les demandes en attente ; <RC> // reprise de scheduler ()</p>		

--

Exercice 2

1. Cas non- Préemptif:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Periph																						
CPU	P1	P1	P2	P2	P2	P4	P5	P5	P3	P3	P6	P6	P6	P7	P7	P7	P6	P3	P3	P8	P8	P9
Ftr (F1)	-	-	-	P4	P4, P5	P5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	P9	-
Fsys (F2)	P2	P2	-	-	-	-	-	-		P6	P7	P7	P7	P6	P6	P6	-	-	-	-	-	--
Fuser (F3)	-	-	P3	P3	P3	P3	P3	P3	-	-	-P3	-P3	-P3	P3	P3	P3, P8	P3, P8	P8	P8	P3	P3	P3, P8

[illegible]

CPU	P9	P9	P10	P10	P10	P3	P3	P8	P8	P8											
Ftr (F1)	-	-	-	-	-	-	-	-	-	-											
Fsys (F2)	-P10	P10	-	-	-	-	-	-	-	-											
Fuser (F3)	P3, P8	P3, P8	P3, P8	P3, P8	P3, P8	P8	P8	-	-	-											

Les programmers sont : scheduler, RfinQ, SVC (arrivée, fin pgmme) .

Scheduler ()

Debut

Attendre :

Si \neg vide(FTR)

alors

Défiler (P-actif, FTR) ;

P-actif.etat := "Actif" ;

Lpsw(P-actif) ;// le rendre actif

sinon si \neg vide(Fsys)

Défiler (P-actif, Fsys) ;

P-actif.etat := "Actif" ;

Q1= 3;

Lpsw(P-actif) ;

sinon si \neg vide(Fuser)

```

        Défiler (P-actif, Fuser) ;
        P-actif.etat := "Actif" ;
        Q2= 2;
        Lpsw(P-actif) ;
    sinon  Jump Attendre ;
Fsi
fsi

Fin

```

SVC (cause,)

Debut

<S.G.CXT>

Cas cause vaut

Creation : Créer le PCB;

PCB.etat := "Prêt";

PCB.File := File_correspondante () ;

PCB.priorité := prio ; // la priorité de La FILE (0, 1, 2),

si PCB.File= F1 Alors Enfiler (PCB, F1) ;

si PCB.File= F2 Alors Enfiler (PCB, F2) ;

```
        si PCB.File= F3 Alors Enfiler (PCB, F3) ;  
// ici Préemptive (Pt) si la politique est Préemptive  
    Dem E/S: .... // connu  
Fin processus: ... // connu // liberer processus+ appeler  
    <R.G .CXT>
```

Routine d'it Horloge() // la fréquence est 1 ms.

```
    <S.G.CXT> // à rectifier , prendre en compte l'origine de processus actif  
    Q1- -, Q2- -;  
    Si si pactif ∈ à F2 et Q1=0 alors          P-actif.etat := "Prêt";  
                                                Enfiler (P, F2);  
                                                Fsi;  
    Si si pactif ∈ F3 et Q2=0 alors          P-actif.etat := "Prêt";  
                                                Enfiler (P, F3);  
                                                Fsi;
```

```
    Lpsw(scheduler);
```

```
    <R.G.CXT>
```


2. Cas Préemptif:// jusqu'a 10 ms

	1	2	3	4	5	6	7	8	9	10
Periph										
CPU	P1	P1	P2	P4	P5	P5	P2	P2	P3	P6
Ftr (F1)	-	--	-	-	-	-	-	-	-	-
Fsys (F2)	P2	P2	-	P2	P2	P2	-	-	-	-
Fuser (F3)	-	-	P3	P3	P3	P3	P3	P3	-	P3

Préemptive (Pt) // on fait appelle à cette procedure ds le SVC

Debut

```

Si (P-actif ≠ Null) et (Pt.priorité > P-actif.priorité)
    Alors P-actif.etat := "Prêt";
           Enfiler (P-actif, File correspondante());
           Lpsw(scheduler);
Fsi;

```

Fin.

