TD n°9

**Exercice 1** :

Soit 3 processus :

P1 :

P1A

P1B (accès fic 1)

P1C

P1D (accès fic 3)

P2 :

P2A (accès fic 2)

P2B

P2C (accès fic 3)

P2D

P3 :

P3A (accès fic 3)

P3B (accès fic 2)

P3C

P3D (accès fic 1)

1. Donner les portions de code en exclusion mutuelle
2. Donner le code de chaque processus

**Exercice 2** :

Soit 2 fichiers toto.txt et truc.txt

Ces 2 fichiers se sont accédés par des processus

* Ecrire le code c permettant d'initialiser la gestion de la concurrence autours de ces 2 fichiers en utilisant des sémaphores
* Ecrire le code C de suppression de ces mécanismes
* Ecrire le code C permettant d'accéder au fichier

**Exercice 3** :

Soit A et B 2 variables partagées.

L'accès concurrent à ces variables s'effectue en utilisant 2 sémaphores (S et S')

Montrer que, sans autre hypothèse, de système comporte une situation d'interblocage.

**Exercice 1 :**

P1B – P3D exclusion mutuelle

P1D – P2C – P3A exclusion mutuelle

P2A – P3B exclusion mutuelle

S1, S2, S3 initialisé à 1

**P1** : P1A, PS1, P1B, VS1, P1C, PS3, P1D, VS3

**P2** : PS2, P2A, VS2, P2B, PS3, P2C, VS3, P2D

**P3** : PS3, P3A, VS3, PS2, P3B, VS2, P3C, PS1, P3D, VS1

**Exercice 2 :**

int main(int argc, char \* argv[]){

    int s= semget(350,2,IPC\_CREAT);//(référencement (clé des sémaphores), nombre de sémaphore, code de création)

    union senum s;

    s.val =1;

    semctl(s,0,SETVAL,s);

    semctl(s,1,SETVAL,s);

    exit(0);

}

int main(int argc, char \* argv[]){

    int s=semget(350,2,0);

    semctl(s,0,IPC\_RMID,0);

    exit(0);

}

int main(argc, char \* argv[]){

    int s= semget(350,2,0);

    struct sembuf b;

    b.sem.num=0;

    b.sem.op=1;

    semop(s,&b,1)//p(s)

    //Accès à toto.txt

    b.sem\_op.op=1;

    semop(s,&b,1);//v(s)

    b.sem\_num=1;

    b.sem\_op=-1;

    semop(s,&b,1);//p(s)

    //Accès truc.txt

    b.sem\_op=1;

    semop(s,&b,1);//v(s')

    exit(0);

}

**Exercice 3 :**

P1 : P(s), P(s'), accès A et B, V(s), V(s')

P2 : P('), P(s), accès A et B, V(s'), V(s)

Exécution :

P1.P(S), P2 P(S'), P1 P(S') 🡪 mise en attente, P2 P(S)🡪 mise en attente, …