

## TP Exclusion mutuelle

### Objectif :

Satisfaire les demandes de différents sites dans l'ordre des demandes d'entrée en section critique après accord de tous les autres sites.

### Principe :

Distribution d'une file d'attente associé à un mécanisme d'estampillage des messages. Basé sur un jeton circulant dans un anneau logique de processus.

### Hypothèse :

- Canaux de communication FIFO
- Horloges logiques scalaire HLi sur chaque site i
- Trois types de messages diffusés à tous :
  - Requête : envoyé quand site veut SC
  - Réponse : envoyé pour répondre à requête
  - Libération : envoyé par site qui sort de SC
- Une file FILEi de message ordonné par leur estampille ELM par site i

### Algorithme :

Si un site Si veut entrer en SC :

- Il place sa requête dans FILEi et envoie message requête à tous les autres
- Si entre en SC quand reçu réponse et quand sa requête est en tête de FILEi

Quand un site Sj reçoit requête de Si :

- Met à jour HLj (en utilisant HLj et ELM)
- Place requête dans FILEj et envoie sa réponse réponse à Si

Quand Si sort de SC il envoie message libération à tous les sites et retire sa requête de FILEi

Si le site Sj reçoit un message libération de S'il met à jour son horloge et enlève requête de FILEj

### Implémentation :

On décide arbitrairement que dans 2% des cas le site génère un évènement aléatoire ou envoie une demande de rentrée en section critique (SC).

On a créé une structure clock qui se compose de l'estampille temporelle « time » et du site i associé à cet évènement.

```
rd=rand();
/* rien faire dans X% des cas */
if(rd%50==1)
{
    //evenement locale
    hl.time++;
    printf("Evenement local %d",hl.time);
}
else if(rd%50==0 && !demande)
{
```

```

hl.time++;
printf(" evenement %i : ",hl.time);
//demande d'entree en Section critique si pas de demande déjà
en cours
demande=1;
file[nb_file]=hl;
nb_file++;
qsort(file, nb_file, sizeof(struct clock), compare);
accord[PortBase-site0]=1;

for(i=site0;i<site0+NSites;i++)
{
    if(i!=PortBase)
    {
        //envoyer aux autres serveurs la demande
        sprintf(chaine,"(%i,%i) REQ", hl.time, hl.si);
        printf(" j'envoie a %i la requete de SC ", i);
        SendMsg(i, chaine);
    }
}
}

```

La variable « demande » nous indique si une demande en SC a déjà été faite pour ce site. Si ce n'est pas le cas, alors on rajoute dans la file la demande faite puis on ordonne ensuite la file temporellement pour que les demandes soient traitées de manière chronologique puis on rajoute l'accord de son propre site pour finir par envoyer la requête aux autres sites. Lorsque l'on reçoit un message on actualise notre propre clock selon la clock du message reçu :

```
hl.time=max(hl.time,new_clock.time)+1;
```

Si le message reçu est une requête :

```

if(strncmp("REQ", chaine2, 3)==0)
{
    //le message est une requete de SC
    file[nb_file]=new_clock;
    nb_file++;
    qsort(file, nb_file, sizeof(struct clock), compare);

    //renvoyer reponse positive au site
    hl.time++;
    printf(" evenement %i : ",hl.time);

    sprintf(chaine, "(%i,%i) autorise l'entree en SC",
hl.time, hl.si);
    printf(" j'envoie a %i l'autorisation d'entree en SC ",
new_clock.si);
    SendMsg(new_clock.si, chaine);
}

```

On place la clock reçue à la fin de notre propre file que l'on trie ensuite pour assurer la chronologie. Puis on renvoie l'accord du site pour la rentrée en SC du demandeur.

```

else if(strncmp("autorise", chaine2, 8)==0)
{
    //le message est une autorisation

```

```

        accord[new_clock.si-site0]=1;
    }

```

On actualise sa liste d'accord lorsque l'on reçoit une autorisation.

On rentre en SC si on a tous les accords et que le premier évènement dans la liste vient de notre site.

Si c'est le cas, on reset les accords de notre site, on envoie un message de libération aux autres sites, on supprime le premier élément dans notre file et on reset la valeur de demande à 0.

```

//SC
if(nb_file!=0)
{
    if(testAccord(accord, NSites) && PortBase==file[0].si)
    {
        //entre en SC
        printf("\nJE SUIS EN SC !\n");

        resetAccord(accord, NSites);
        //envoyer les messages de liberation
        hl.time++;
        printf(" evenement %i : ",hl.time);
        for(i=site0;i<site0+NSites;i++)
        {
            if(i!=PortBase)
            {
                //envoyer aux autres sites la demande
                sprintf(chaine,"(%i,%i) LIBERATION", hl.time,
hl.si);
                printf(" j'envoie a %i la demande de
liberation ", i);
                SendMsg(i, chaine);
            }
        }

        //supprimer l'element de sa propre liste
        size=sizeof(file)/sizeof(int);
        end=file+size;

        for(d=file,p=file+1;p<end;++p,++d)
        {
            *d=*p;
        }
        nb_file--;
        demande=0;
    }
}

```

Pour finir, si le message est une libération, on enlève le premier élément de notre file pour qu'il soit bien retiré dans les files de tous les sites.

```

else if(strncmp("LIBERATION", chaine2, 10)==0)
{
    //le message est une demande de liberation. Il faut
enlever le premier element de file
    size=sizeof(file)/sizeof(int);
    end=file+size;
}

```

```

        for (d=file,p=file+1;p<end;++p,++d)
        {
            *d=*p;
        }
        nb_file--;
    }

```

## Résultats :

Se compile avec : gcc TP-DIST.c -Wall -o serveur

### Site 7000 :

```

vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$ gcc TP-DIST.c -Wall -o serveur.exe
vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$ ./serveur.exe 7000 7000 3
Numero de port de ce site 7000
WaitSync : **SYNCHRO**
WaitSync : **SYNCHRO**
Site 0 : toutes les synchros des autres sites recues
..... evenement 1 : j'envoie a 7001 la requete de SC j'envoie a 7002 la requete de SC ..Message recu :
(3,7002) autorise l'entree en SC : evenement 4 .Message recu : (3,7001) autorise l'entree en SC : evenement
5
JE SUIS EN SC !
..... evenement 6 : j'envoie a 7001 la demande de liberation j'envoie a 7002 la demande de liberation .....
.....Message recu : (10,7001) REQ : evenement 11 evenement 12 : j'envoie a 7001 l'autorisatio
n d'entree en SC ..Message recu : (15,7001) LIBERATION : evenement 16 ..... evenement 17 : j'envoie a 7
001 la requete de SC j'envoie a 7002 la requete de SC ..Message recu : (19,7001) autorise l'entree en SC :
..... evenement 20 .Message recu : (19,7002) autorise l'entree en SC : evenement 21
JE SUIS EN SC !
..... evenement 22 : j'envoie a 7001 la demande de liberation j'envoie a 7002 la demande de liberation ..Messa
ge recu : (20,7002) REQ : evenement 23 evenement 24 : j'envoie a 7002 l'autorisation d'entree en SC ...Me
ssage recu : (27,7002) LIBERATION : evenement 28 ...Evenement local 29.....^C
vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$

```

### Site 7001 :

```

vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$ ./serveur.exe 7000 7001 3
Numero de port de ce site 7001
Wait Synchro du Site 0
WaitSync : **SYNCHRO**
Synchro recue de Site 0
.....Message recu : (1,7000) REQ : evenement 2 evenement 3 : j'envoie a 7000 l'autorisation d'entree en SC ...Mess
age recu : (6,7000) LIBERATION : evenement 7 .....Evenement local 8.Evenement local 9..... evenement 10 : j'e
nvoie a 7000 la requete de SC j'envoie a 7002 la requete de SC ..Message recu : (12,7002) autorise l'entree en SC : eve
nement 13 .Message recu : (12,7000) autorise l'entree en SC : evenement 14
JE SUIS EN SC !
..... evenement 15 : j'envoie a 7000 la demande de liberation j'envoie a 7002 la demande de liberation .Evenement local 16.
.....Message recu : (17,7000) REQ : evenement 18 evenement 19 : j'envoie a 7000 l'autorisation d'entree en SC ...M
essage recu : (22,7000) LIBERATION : evenement 23 .Message recu : (20,7002) REQ : evenement 24 evenement 25 : j'envoie
a 7002 l'autorisation d'entree en SC ..Message recu : (27,7002) LIBERATION : evenement 28 .....^C
vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$

```

### Site 7002 :

```
vassia@DESKTOP-4VTU3HC: /mnt/e/DL/E4/P2/HPC 2/TP2$ ./serveur.exe 7000 7002 3
Numero de port de ce site 7002
Wait Synchro du Site 0
WaitSync : **SYNCHRO**
Synchro recue de Site 0
.....Message reçu : (1,7000) REQ : evenement 2 evenement 3 : j'envoie a 7000 l'autorisation d'entree en SC ...Message
reçu : (6,7000) LIBERATION : evenement 7 .....Evenement local 8.....Message reçu : (10,7001) REQ : even
ement 11 evenement 12 : j'envoie a 7001 l'autorisation d'entree en SC ...Message reçu : (15,7001) LIBERATION : eveneme
nt 16 .....Message reçu : (17,7000) REQ : evenement 18 evenement 19 : j'envoie a 7000 l'autorisation d'entree en S
C .. evenement 20 : j'envoie a 7000 la requete de SC j'envoie a 7001 la requete de SC .Message reçu : (22,7000) LIBERA
TION : evenement 23 .Message reçu : (24,7000) autorise l'entree en SC : evenement 25 .Message reçu : (25,7001) autorise
l'entree en SC : evenement 26
JE SUIS EN SC !
evenement 27 : j'envoie a 7000 la demande de liberation j'envoie a 7001 la demande de liberation .....
^C
```

Les sites rentrent chacun leur tour en SC sans se gêner et en respectant un ordre chronologique.

### Commentaire sur l'unité :

Les Tps sont une bonne application du cours et ne sont pas trop difficiles. Par contre, maintenir un contrôle en présentiel en plein covid alors que tous les autres professeurs trouvent une autre solution, c'est très frustrant pour l'élève surtout que le code sur papier n'est pas la meilleure méthode d'évaluation selon moi.