



- Partage de mémoires
- Synchronisation par sémaphores
- Communication par tubes.
- Communication par messages(files de messages)

Structure générale du programme

```
void Créer-et-initialiser_sémaphores(nv, mutex)
{ ....}
void Créer_files_messages (Qcr, Qncr; Qimp)
{ ....}
void Créer_et_attacher_tampon(Torgane)
{ ....}
void Créer_tube (tube){ ....}

void MaladeCrl()
{.....
exit(0);}
void MaladeNCrt()
{ .....
exit(0)}
void Chirurgien()
{ .....
exit(0)}
void Donneur()
{.....
exit(0)}

void Detruire_sémaphores(nv, mutex) {.....}
void Detruire_files_messages ((Qcr, Qncr,
Qimp){.....}
void Detruire_tampon (Torgane) {....}
void Detruire_tube (tube);
```

```
main()
{
Créer-et-initialiser_sémaphores(nv, mutex);
Créer_files_messages (Qcr, Qncr, Qimp);
Créer_et_attacher_tampon (Torgane);
Créer_tube (tube);

id=fork();
if (id==0) MaladeCr();
id=fork();
if (id==0) MaladeNCr();
id=fork();
if (id==0) Chirurgien();
id=fork();
if (id==0) Donneur();
}
for (int i, i++, i<=4) wait(0);

Detruire_sémaphores(nv, mutex);
Detruire_files_messages (Qcr, Qncr, Qimp);
Detruire_tampon (Torgane);
Detruire_tube (tube);

}
```

- **Structures de données communes**
 - N: taille du tampon
 - M1: Nombre de malades critiques représentés.
 - M2: Nombre de malades non critiques représentés
 - N1: nombre d'organes critiques
 - N2: Nombre d'organes non critiques
- **Structure des files de messages Qcr et Qncr:** (num_malade, num_organe)
- **Structure de la file de messages Qimp:** (type_malade, num_malade, num_organe)
- **Structure du tube :** (type_malade, num_malade, num_organe)
- **Structure du tampon:** (type_malade, num_malade, num_organe)

où type_malade prend 1 pour le malade critique, 2 pour le malade non critique.

Comportement des processus

Processus MaladeCr ()

```
mess1: structure de format : (num_malade, num_organe)
mess2: structure de format: (type_malade, num_malade, num_organe)
entier i:=0; // numéro malade
entier nb_rep:=0; // nombre d'organes reçus
liste de format: (num_malade, num_organe) // enregistre les organes commandés et non encore reçus
repeter
    Si (i<M1) & Agenerer() Alors // la fonction retourne une valeur booléenne
        i:=i+1; // prochain numéro de malade pour lequel on doit générer un organe
        j:= rand() mod N1+1;// numéro organe généré aléatoirement
        mess1.num_malade:=i; mess1.num_organe:=j; envoyer (Qcr, mess1);
        enregistrer( liste, mess1) // enregistre localement les informations sur la commande faite
    Fsi;
    Si recu(Qimpl, mess2) Alors
        Supprimer (liste, mess2); // Supprime mess2 de la liste locale
        nb_rep:=nb_rep+1
    Fsi;
    Jusqu'à (nb_rep=M1);
```

Le texte du processus Malade_NCr () est symétrique au texte du processus Malade_CR. Il suffit de remplacer:

- M1 par M2
- N1 par N2
- Qcr par Qncr

Comportement des processus

Processus Chirurgien ()

```
mess1: structure de format : (num_malade, num_organe)
mess2: structure de format: (type_malade, num_malade, num_organe)
entier i1:=0; // nombre de requêtes reçues du processus malade critique
entier i2:=0; // nombre de requêtes reçues du processus malade non critique
entier nb_rep:=0; //nombre de commandes honorées
repeter
    Si (i1<M1) & recu(Qcr, mess1) Alors
        i1:=i1+1; mess2.type_malade:=1; mess2.num_malade:=mess1.num_malade;
        mess2.num_organe:= mess1.num_organe); envoyer (tube, mess2)
    Fsi;
    Si (i2<M2) & recu(Qncr, mess1) Alors
        i2:=i2+1; mess2.type_malade:=2; mess2.num_malade:=mess1.num_malade;
        mess2.num_organe:= mess1.num_organe); envoyer (tube, mess2)
    Fsi;
    P(mutex);
    Si (cpt<>0) Alors V(mutex); Prelever(Torgane, mess2); P(mutex); cpt:=cpt-1; V(mutex); V(nvide);
        envoyer (Qimp, mess2); nb_rep:=nb_rep+1
    Sinon V(mutex)
Fsi
Jusqu'à (nb_rep=M1+M2);
```

Comportement des processus

Processus Donneur ()

Tcr: Tableau [1,,M1]de structure valide: booléen; num_organes: entier fstructure;

Tncr: Tableau [1,,M2]de structure valide: booléen; num_organes: entier fstructure;

mess2: structure de format: (type_malade, num_malade, num_organes)

entier i:=0; /* nombre de commandes impliquant les deux types de malades reçues du processus chirurgien */

entier nb_rep:=0; //nombre de commandes honorées

repeter

Si (i<M1+M2) Alors

 Si recu (tube, mess2) Alors enregistrer (Tcr, Tncr, mess2); i:=i+1 Fsi

Fsi;

Tmalade:= rand() mod 2+1;

Cas Tmalade de

1:Nmalade:=rand() mod M1+1; Norgane:=rand() mod N1+1;

 Si trouve(Tcr, Nmalade, Norgane) Alors // la fonction désactive la case correspondant à la commande trouvée

 P(nvide); Deposer(Torgane, (Tmalade, Nmalade, Norgane)); P(mutex); cpt:=cpt+1; V(mutex);

 nb_rep:=nb_rep+1; Tcr[Nmalade].valide:=0;

Fsi;

2: Nmalade:=rand() mod M2+1; Norgane:=rand() mod N2+1;

 Si trouve(Tncr, Nmalade, Norgane) Alors // la fonction désactive la case correspondant à la commande trouvée

 P(nvide); Deposer(Torgane, (Tmalade, Nmalade, Norgane)); P(mutex); cpt:=cpt+1; V(mutex);

 nb_rep:=nb_rep+1; Tncr[Nmalade].valide:=0;

Fsi;

Fcas

Jusqu'à (nb_rep=M1+M2);