

Compte-rendu de l'exercice de synthèse

Date du CR : 09/12/2021

Section : LA1

Auteur : WALID MOUDDEN

Matière : Programmation Systèmes

TABLE DES MATIÈRES

1- Introduction	3
2- Problème	3
3- Résolution du problème	3
5- Démonstration	5
6- Conclusion	7

1 Introduction

L'objectif de cet exercice est d'utiliser différentes notions acquises en cours de programmation systèmes de LA1 cette année pour réaliser une application client/serveur de « chat » de type broadcast en utilisant principalement : la mémoire partagée, les signaux, les tubes nommés et les threads.

2 Problème

Il s'agit d'écrire une application client/serveur de « chat » de type broadcast.

Le serveur doit pouvoir répondre à des sollicitations de nombreux clients. Il doit recevoir des messages de la part d'un client, et re-transmettre ce message à tous les clients en ligne.

Le client peut envoyer des messages au serveur, saisi par l'utilisateur à la console.

Le serveur est lancé par un utilisateur qui lui communique une valeur de clef. Grâce à cette valeur de clef, le serveur va chercher quelque part les informations nécessaires aux clients pour se connecter, si ce client connaît lui-même la clef du jour, qui lui est passée en argument au lancement.

Exemple :

lancement du serveur :

`./serveur 1234`

Lancement d'un client :

`./client 1234`

1234 étant la « clef du jour ».

3 Résolution du problème

Tout d'abord, dans le programme serveur j'ai commencé par créer une mémoire partagée qui contient son PID à partir du clé passée dans la ligne de commande pour permettre aux clients de lui envoyer des signaux

```
pidServeur = shmat(shmId, NULL, 0);
```

```
    if (pidServeur == (char *) -1)
```

```
    {
```

```
        printf("Impossible de s'attacher au segment de mémoire %d \n", shmId);  
        exit(-1);
```

```
    }
```

printf(pidServeur, "%d", getpid()); // ecrire dans la memoire le pid du serveur
puis dans le programme client, j'accède au mémoire partagée et j'envoie un signal
pour demander la connexion au serveur

et puis dans le serveur j'ajoute le pid du client qui a envoyé le signal au liste des
clients et j'ouvre un tube avec ce client pour lui envoyer des messages

au client, j'ouvre un autre tube et je donne la saisi à l'utilisateur pour envoyer des
messages du client au serveur et ce dernier ouvre ce tube dans lecture pour
pouvoir lire les messages et il l'affiche et envoie ce message a tous les clients
(broadcast)

pour terminer le programme, le client doit saisir AU REVOIR pour qu'il se
déconnecte, quand tous les clients sont déconnecté le serveur termine à son tour.

```
while(strncmp(messageClient,"AU REVOIR",9) != 0)
{
    printf("Votre message ? \n");
    fgets(messageClient,256,stdin);
    write(entreeTube,messageClient,TAIILE_MESSAGE);
    printf("Message envoye au serveur \n");
}
```

précision par rapport au makefile

pour avoir l'exécutable que du serveur : make serveur

pour avoir l'exécutable que du client : make client

pour les deux : make

pour supprimer les deux : make clean

4- Demonstration :

Début du programme

```
110 moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2
111 moudden@moudden-g3-walid: ./serveur 1
112 Création de la clé en cours ...
113 Clé crée !
114 Mon PID dispo dans la mémoire partagée !
115 Nouveau client avec le pid 11888
116 SERVEUR - Ouverture du tube pour le client OK
117 SERVEUR - Ouverture du FIFO OK
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135 printf("Impossible d'ouvrir le tube en lecture \n");
136 else
137 {
138     moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2
139     Fichier Edition Affichage Rechercher Terminal Aide
140     moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2$ ./client 1
141     Pdenande de connexion en cours ....
142     Connexion réussi au serveur
143     // on pas CLIENT - Ouverture du tube écriture OK
144     pthread_CLIENT - Ouverture du tube lecture OK
145     // tant q
146     while(str
147     {
148         pi
149         fi
150         wi
151         pi
152     }
153     CHECK(kil
154     ret = shm
155     if(ret ==
156     {
157         ei
158         pi
159         e:
160     }
161     close(ent)
162     close(sor
163     printf("CLIENT - fin de programme (1)");
164 }
```

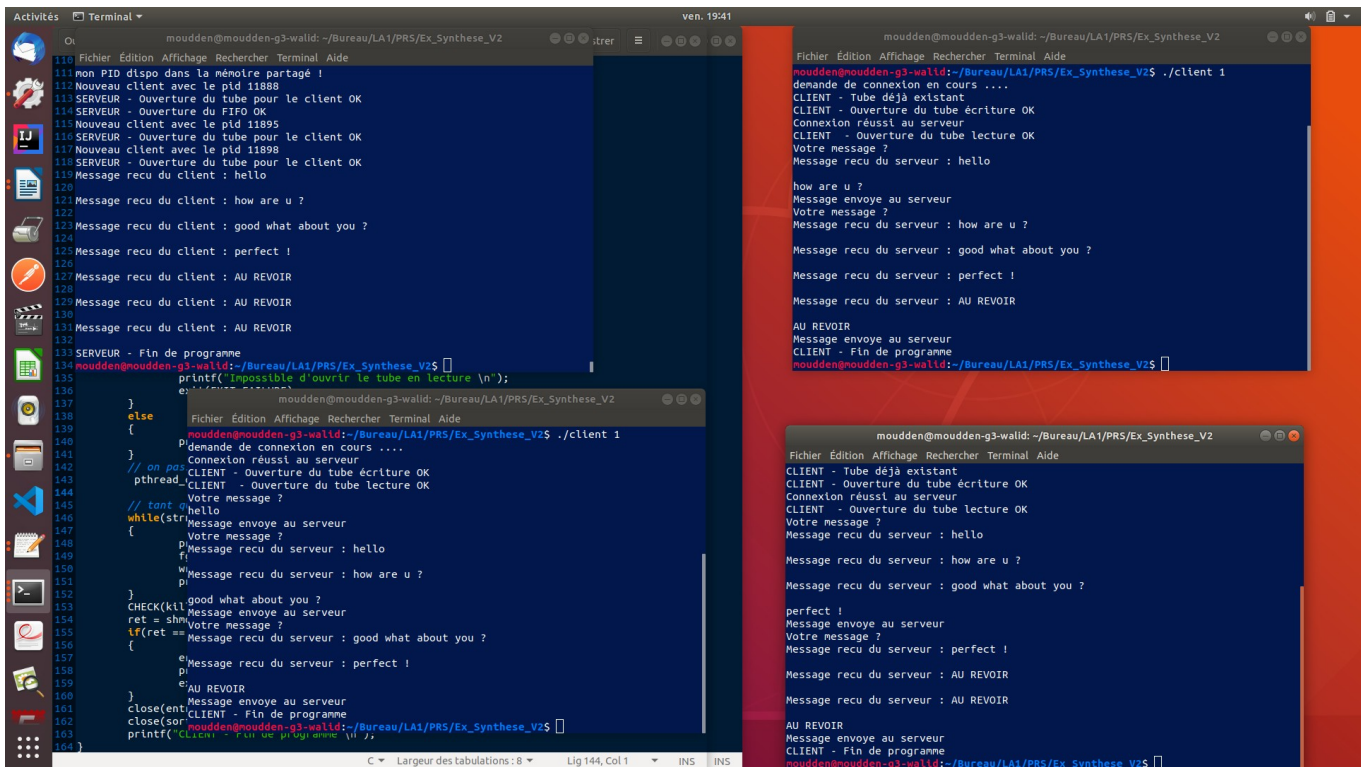
communication entre differents processus

```
Activités Terminal ven. 19:40
moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2
110 Fichier Édition Affichage Rechercher Terminal Aide
111 moudden@moudden-g3-walid:~/Bureau/LA1/PRS/Ex_Synthese_V2$ ./serveur 1
112 Création de la clé en cours ...
113 Clé créée !
114 non PID dispo dans la mémoire partagée !
115 Nouveau client avec le pid 11888
116 SERVEUR - Ouverture du tube pour le client OK
117 SERVEUR - Ouverture du FIFO OK
118 Nouveau client avec le pid 11895
119 SERVEUR - Ouverture du tube pour le client OK
120 Nouveau client avec le pid 11898
121 SERVEUR - Ouverture du tube pour le client OK
122 Message reçu du client : hello
123
124 Message reçu du client : how are u ?
125
126 Message reçu du client : good what about you ?
127
128
129
130
131
132
133
134
135 printf("Impossible d'ouvrir le tube en lecture \n");
136 exit(EXIT_FAILURE);
137
138 }
139 else
140 {
141     moudden@moudden-g3-walid:~/Bureau/LA1/PRS/Ex_Synthese_V2$ ./client 1
142     demande de connexion en cours ....
143     Connexion réussit au serveur
144     CLIENT - Ouverture du tube écriture OK
145     pthread CLIENT - Ouverture du tube lecture OK
146     Votre message ?
147     // tant q'hello
148     while(str)
149     {
150         Message envoyé au serveur
151         Votre message ?
152         Message reçu du serveur : hello
153         f
154         Message reçu du serveur : how are u ?
155         pi
156     }
157     CHECK(kl) good what about you ?
158     Message envoyé au serveur
159     ret = shm
160     Votre message ?
161     if(ret ==
162     Message reçu du serveur : good what about you ?
163     {
164         e
165         p
166         e
167     }
168     close(ent
169     close(sor
170     printf("CLIENT - Fin de programme \n");
171 }
```

Fin d'un programme Client

```
Activités Terminal ven. 19:41
moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2
118 Fichier Édition Affichage Rechercher Terminal Aide
119 moudden@moudden-g3-walid:~/Bureau/LA1/PRS/Ex_Synthese_V2$ ./serveur 1
120 Création de la clé en cours ...
121 Clé créée !
122 non PID dispo dans la mémoire partagée !
123 Nouveau client avec le pid 11888
124 SERVEUR - Ouverture du tube pour le client OK
125 SERVEUR - Ouverture du FIFO OK
126 Nouveau client avec le pid 11895
127 SERVEUR - Ouverture du tube pour le client OK
128 Nouveau client avec le pid 11898
129 SERVEUR - Ouverture du tube pour le client OK
130 Message reçu du client : hello
131
132 Message reçu du client : how are u ?
133
134 Message reçu du client : good what about you ?
135
136 Message reçu du client : perfect !
137
138 Message reçu du client : AU REVOIR
139
140
141
142
143 printf("Impossible d'ouvrir le tube en lecture \n");
144 exit(EXIT_FAILURE);
145
146 }
147 else
148 {
149     moudden@moudden-g3-walid:~/Bureau/LA1/PRS/Ex_Synthese_V2$ ./client 1
150     demande de connexion en cours ....
151     Connexion réussit au serveur
152     CLIENT - Ouverture du tube écriture OK
153     pthread CLIENT - Ouverture du tube lecture OK
154     Votre message ?
155     // tant q'hello
156     while(str)
157     {
158         Message envoyé au serveur
159         Votre message ?
160         Message reçu du serveur : hello
161         f
162         Message reçu du serveur : how are u ?
163         pi
164     }
165     CHECK(kl) good what about you ?
166     Message envoyé au serveur
167     ret = shm
168     Votre message ?
169     if(ret ==
170     Message reçu du serveur : good what about you ?
171     {
172         e
173         p
174         e
175     }
176     close(ent
177     close(sor
178     printf("CLIENT - Fin de programme \n");
179 }
```

fin du programme serveur



```
110 Fichier Edition Affichage Rechercher Terminal Aide
111 mon PID dispo dans la mémoire partagée !
112 Nouveau client avec le pid 11888
113 SERVEUR - Ouverture du tube pour le client OK
114 SERVEUR - Ouverture du FIFO OK
115 Nouveau client avec le pid 11895
116 SERVEUR - Ouverture du tube pour le client OK
117 Nouveau client avec le pid 11898
118 SERVEUR - Ouverture du tube pour le client OK
119 Message reçu du client : hello
120
121 Message reçu du client : how are u ?
122
123 Message reçu du client : good what about you ?
124
125 Message reçu du client : perfect !
126
127 Message reçu du client : AU REVOIR
128
129 Message reçu du client : AU REVOIR
130
131 Message reçu du client : AU REVOIR
132
133 SERVEUR - Fin de programme
134 moudden@moudden-g3-walid: ~/Bureau/LA1/PRS/Ex_Synthese_V2
135 printf("Impossible d'ouvrir le tube en lecture \n");
136
137 }
138
139 {
140     // on pas
141     pthread CLIENT - Ouverture du tube écriture OK
142     pthread CLIENT - Ouverture du tube lecture OK
143     Votre message ?
144     // tant q'hello
145     while(str)
146     {
147         Message envoye au serveur
148         Votre message ?
149         Message reçu du serveur : hello
150         Message reçu du serveur : how are u ?
151         Message reçu du serveur : good what about you ?
152         Message reçu du serveur : perfect !
153         Message envoye au serveur
154         ret = shm
155         if(ret == Message reçu du serveur : good what about you ?
156         {
157             Message reçu du serveur : perfect !
158             p
159             AU REVOIR
160         }
161         Message envoye au serveur
162         close(ent)
163         close(ent)
164         printf("CLIENT - Fin de programme \n");
165     }
```

5 Conclusion

Je trouve que c'est un exercice très important, il m'a permis d'appliquer les différentes connaissances que j'ai acquies en PRS cette année et d'utiliser mon esprit algorithmique pour résoudre les problèmes que j'ai rencontré et d'améliorer ma manière de résoudre les problèmes en mettant tout d'abord une architecture avant de commencer le code.

Merci pour cet exercice intéressant !