



## Rapport Mini Chat

Joel Roman Ky

### 1. Introduction

Dans le cadre du cours de Systèmes d'Exploitation Centralisés, ce projet consistait à réaliser un minichat, application de chat entre plusieurs utilisateurs, mais en utilisant deux versions : ***version Tableau Blanc et version Console/Serveur.***

Chacune de ces deux versions utilisant un choix de conception différents.

### 2. Choix de conceptions

#### 1. Console/Serveur

Cette version du minichat consiste en une application serveur lancé dans un processus **Serveur** qui gère la transmission des messages des processus **Consoles**

lancés dans des processus différents à travers la structure des tubes nommés appelés avec la primitive **mkfifo**.

Cette méthode d'implémentation reposait essentiellement sur les notions de **tubes**, **fichiers** et surtout l'utilisation de la primitive **select** qui consiste à rendre les E/S bloquantes afin que les lectures dans les tubes se fassent au bon moment.

Les commentaires dans le code détailleront les choix de conception.

## 2. Tableau blanc

Cette version du minichat consiste en une version sans serveur, où les messages sont écrits directement sur l'écran grâce à un couplage mémoire partagé avec la primitive **mmap**.

Cependant avec une architecture pareil, les messages ne sont pas actualisés en temps réel car l'écriture dans un fichier partagé en lecture et en écriture l'actualisation du fichier n'est pas systématique.

### 3. Difficultés

Les principales difficultés pour ces versions furent ;

- Serveur/Console
  - Difficultés de gerer le cas ou le nombre de clients est supérieur aux nombres maximum en les faisant attendre.
  - La suppression d'un client ne bloque ni le serveur , ni les autres clients mais empêche la fin du processus qui bug.
- Tableau Blanc
  - Difficultés de gérer l'actualisation périodique de la discussion

### 3. Comparaison

La version Console / Serveur qui est la plus difficile en termes d'implémentation semble etre la plus efficace en ce sens que les messages sont actualisés à chaque fois.

Celle du tableau blanc par contre qui par le principe de couplage mémoire est plus simple présente des difficultés que l'on pourrait peut être resoudre avec des

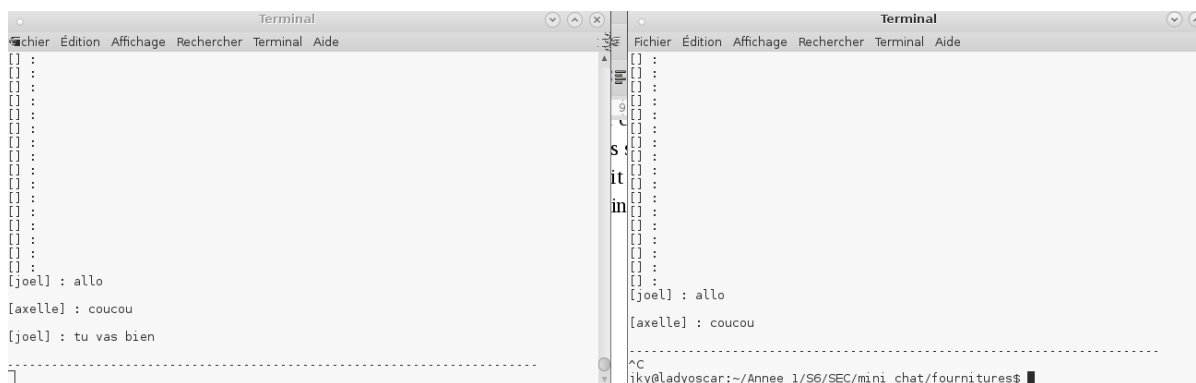
signaux qui seraient redéfinis, mais que je n'ai pas pu implémenter.

## 4. Tests

## 1. Tableau Blanc

Les tests réalisés dans cette version ont été essentiellement l'appui de la commande **CTRL-C**.

L'appui de cette commande dans un des processus fils n'interrompt pas la discussion des autres participants. Et donc elle peut être utilisé pour terminer la conversion. Cependant l'arrivée ou le départ d'un participant ne sont pas notifiés aux autres utilisateurs

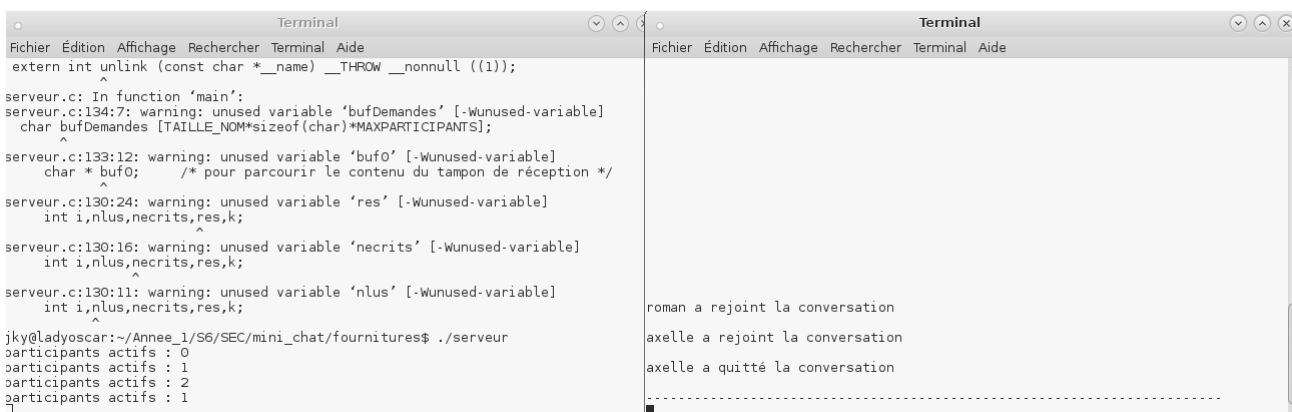


## 2. Console/Serveur

Les principaux tests effectués pour cette version ont été :

- la déconnexion d'un client

Lors de l'exécution de ce test on peut voir que les autres utilisateurs sont notifiés de la déconnexion et que le nombre de participants est actualisé.



The image shows two terminal windows side-by-side. The left window displays C code for a server, with several compiler warnings about unused variables. The right window shows a chat log with messages from 'roman', 'axelle', and 'axelle'.

```
extern int unlink (const char *__name) __THROW __nonnull ((1));

serveur.c: In function 'main':
serveur.c:134:7: warning: unused variable 'bufDemandes' [-Wunused-variable]
  char bufDemandes [TAILLE_NOM*sizeof(char)*MAXPARTICIPANTS];
  ^
serveur.c:133:12: warning: unused variable 'buf0' [-Wunused-variable]
  char * buf0;      /* pour parcourir le contenu du tampon de réception */
  ^
serveur.c:130:24: warning: unused variable 'res' [-Wunused-variable]
  int i,nlus,necrits,res,k;
  ^
serveur.c:130:16: warning: unused variable 'necrits' [-Wunused-variable]
  int i,nlus,necrits,res,k;
  ^
serveur.c:130:11: warning: unused variable 'nlus' [-Wunused-variable]
  int i,nlus,necrits,res,k;
  ^
jky@ladyoscar:~/Annee_1/S6/SEC/mini_chat/fournitures$ ./serveur
participants actifs : 0
participants actifs : 1
participants actifs : 2
participants actifs : 1

roman a rejoint la conversation
axelle a rejoint la conversation
axelle a quitté la conversation
.....
```

Cependant le terminal qui contenait le processus du client qui vient de se déconnecter ne se termine pas proprement et boucle indéfiniment sur l'affichage.

- l'appui de la touche **CTRL-C**

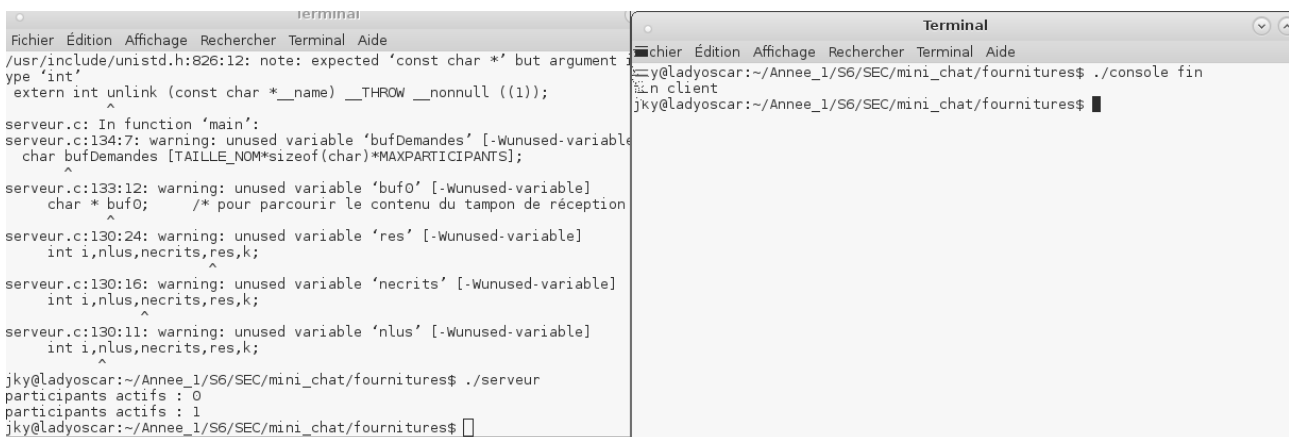
L'appui de cette commande dans le shell entraine un bug complet de l'application en empêchant la

discussion des autres clients qui ne peuvent plus utiliser l'application car leurs processus bouclent indéfiniment sur un affichage des messages.

Une solution possible serait d'ignorer le signal **SIGINT**, mais ne serait pas la meilleure solution

- la fin du serveur par la connexion d'un client de pseudo *fin*.

Ce dernier test fonctionne parfaitement car cela entraine la terminaison propre du serveur



The image shows two terminal windows side-by-side. The left window, titled 'terminal', displays the compilation and execution of a server program. It shows several compiler warnings about unused variables and the successful execution of the server. The right window, titled 'Terminal', shows the execution of a client program. The user enters 'fin' as input, which triggers the termination of the server process in the left window.

```
Fichier Édition Affichage Rechercher Terminal Aide
/usr/include/unistd.h:826:12: note: expected 'const char *' but argument
type 'int'
extern int unlink (const char *__name) __THROW __nonnull ((1));
^
serveur.c: In function 'main':
serveur.c:134:7: warning: unused variable 'bufDemandes' [-Wunused-variable]
char bufDemandes [TAILLE_NOM*sizeof(char)*MAXPARTICIPANTS];
^
serveur.c:133:12: warning: unused variable 'buf0' [-Wunused-variable]
char * buf0; /* pour parcourir le contenu du tampon de réception
^
serveur.c:130:24: warning: unused variable 'res' [-Wunused-variable]
int i,nlus,necrits,res,k;
^
serveur.c:130:16: warning: unused variable 'necrits' [-Wunused-variable]
int i,nlus,necrits,res,k;
^
serveur.c:130:11: warning: unused variable 'nlus' [-Wunused-variable]
int i,nlus,necrits,res,k;
^
jky@ladyoscar:~/Annee_1/S6/SEC/mini_chat/fournitures$ ./serveur
participants actifs : 0
participants actifs : 1
jky@ladyoscar:~/Annee_1/S6/SEC/mini_chat/fournitures$
```

```
Fichier Édition Affichage Rechercher Terminal Aide
jky@ladyoscar:~/Annee_1/S6/SEC/mini_chat/fournitures$ ./console fin
fin client
jky@ladyoscar:~/Annee_1/S6/SEC/mini_chat/fournitures$
```

Cependant, cela entraine une mauvaise terminaison des autres processus clients qui auraient été lancés.