

SIF1015 - Devoir #2

Manuel utilisateur

Projet réalisé par Julio Bangadebia et Quentin Anière

Description

Ce projet est la suite logique du devoir 1. La gestion de la liste chaînée des VMS est la même. Ce qui change c'est la façon dont les transactions sont lues : avant c'était à partir d'un fichier texte, maintenant c'est à partir de sockets.

Un client, via son interface graphique compose une transaction qui est constituée d'opérations. La transaction est envoyée au serveur qui la traite et retourne le résultat à l'utilisateur.

Compilation

Serveur

Avec makefile

```
make server
```

Sans makefile

```
gcc -o server server.c
```

Client

Avec makefile

```
make client
```

Sans makefile

```
gcc -o client client.c
```

Utilisation

Serveur

Avec makefile

```
make launch-server
```

Sans makefile

```
./server
```

Client

Avec makefile

```
make launch-client
```

Sans makefile

```
./client
```

Options

Pour choisir le port (client et serveur) :

Avec makefile

```
make launch-< server | client > port=< port >
```

Sans makefile

```
./< server | client > < port >
```

Opérations possibles

- A - Ajouter une VMS
- L x - y - Lister les VMS avec un identifiant compris entre x et y
- E x - Éliminer une VMS avec un identifiant x
- X x y - Exécuter le code binaire compris dans le fichier y sur la VMS x

Fonctionnement

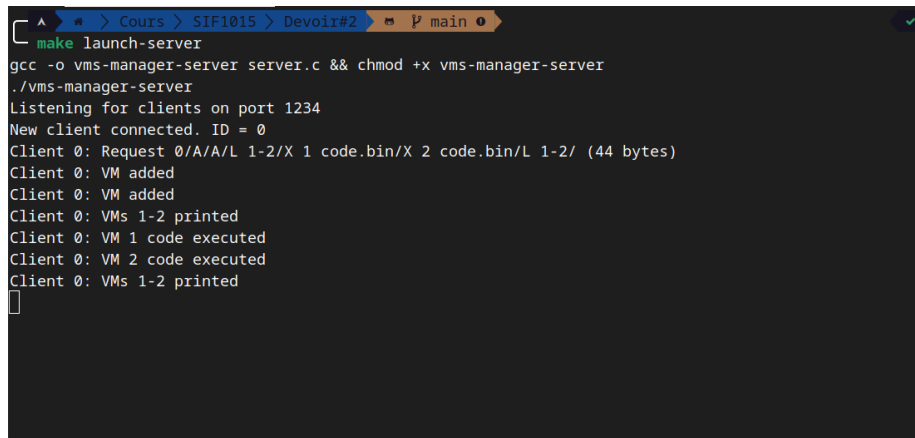
Client

1. Le client affiche une interface graphique
2. L'utilisateur compose une transaction à l'aide de l'interface graphique (Les opérations sont définies plus haut)
3. Le client envoie la transaction au serveur (via un socket)
4. Le client affiche le résultat de la transaction

Serveur

1. Le serveur attend une transaction (via un socket)
2. Le serveur l'ajoute à une liste FIFO (First In, First Out) de transaction
3. Le serveur traite la première transaction de la liste
4. Le serveur démarre un thread pour chaque opération qui compose la transaction
5. Le serveur attend que tous les threads aient terminé
6. Le serveur retourne le résultat de la transaction au client

Captures d'écran



```
^ * > Cours > SIF1015 > Devoir#2 * P main o
make launch-server
gcc -o vms-manager-server server.c && chmod +x vms-manager-server
./vms-manager-server
Listening for clients on port 1234
New client connected. ID = 0
Client 0: Request 0/A/A/L 1-2/X 1 code.bin/X 2 code.bin/L 1-2/ (44 bytes)
Client 0: VM added
Client 0: VM added
Client 0: VMs 1-2 printed
Client 0: VM 1 code executed
Client 0: VM 2 code executed
Client 0: VMs 1-2 printed
□
```

Figure 1: Le serveur affiche des logs dans la sortie standard

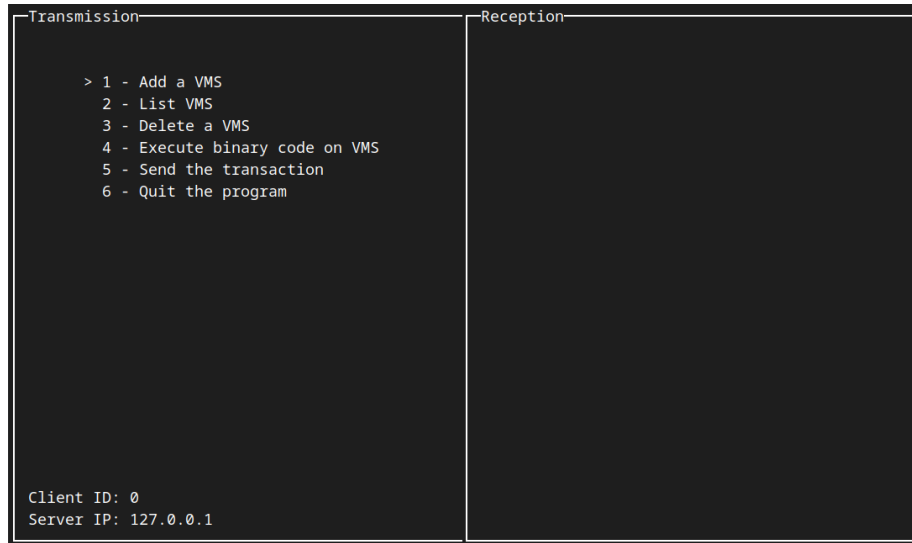


Figure 2: Le client affiche une interface graphique

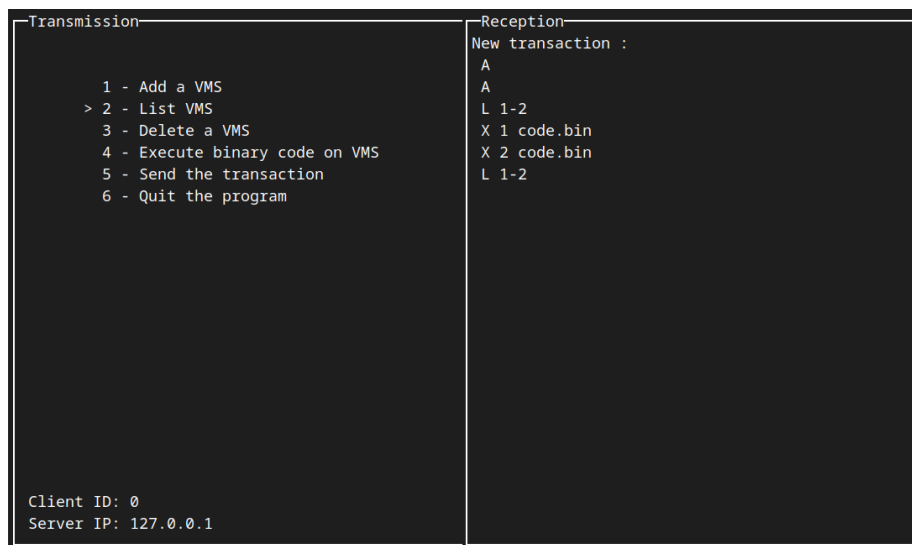


Figure 3: Le client viens de composer une transaction

Transmission	Reception
<pre>1 - Add a VMS 2 - List VMS 3 - Delete a VMS 4 - Execute binary code on VMS > 5 - Send the transaction 6 - Quit the program</pre>	<pre>Server response : VM 1 added VM 2 added VMs 1-2: - VM #1: Busy = 0 - VM #2: Busy = 0 VMs 1-2: - VM #1: Busy = 1 - VM #2: Busy = 1 VM 1 code executed VM 2 code executed</pre>
<pre>Client ID: 0 Server IP: 127.0.0.1</pre>	

Figure 4: Le client vient de recevoir la réponse du serveur