

**Travail Pratique 2 (LINUX)**  
**Virtual Machine System (VMS) et ressources partagées**

**1-Amina Guerrou**

**2- Faycal Chefai**

Présenté a **Francois Meunier**

**SESSION AUTOMNE 2023**

Pour accéder a notre projet via linux :

ident : **mi642**

Mdp : **oolseef9**

Il se trouve dans le bureau intitulé « tp »

Ce projet a été conçu pour répondre aux besoins de gestion efficace des machines virtuelles dans un environnement Linux. L'objectif était de créer un système capable de gérer plusieurs VMs, en permettant des opérations telles que l'ajout, la suppression, le listage, et l'exécution de fichiers sur les VMs.

Le système est basé sur une architecture client-serveur utilisant des FIFOs pour la communication et des threads pour le traitement parallèle. Il est développé en C avec l'utilisation de la bibliothèque ncurses pour l'interface utilisateur et pthreads pour la gestion des threads.

Le projet a été réalisé en plusieurs phases : la conception de l'architecture, le développement du code, les tests et la mise en œuvre. Les défis majeurs rencontrés incluaient la gestion de la communication entre les processus et la synchronisation des threads.

### **Gestion des VMs**

Le système permet de réaliser diverses opérations sur les VMs :

**Ajout de VMs** : Création de nouvelles machines virtuelles.

**Suppression de VMs** : Suppression de machines virtuelles existantes.

**Listage de VMs** : Affichage des machines virtuelles disponibles.

**Exécution de fichiers** : Exécution de fichiers spécifiques sur les VMs.

## I. **Fichiers inclus :**

1- Fichier.c :

- gestionVSM\_MAIN.c - gestionListeChaineVMS.c - gestionVMS.c  
- client.c

2- Fichier d'entête :

- gestionListeChaineVMS.h - gestionVMS.h - client.h

3- Fichiers object :

- \_MAIN.o - gestionListeChaineVMS.o  
- gestionVMS.o  
- client.o

4- Makefile

5- Mult-Numbers.olc3

6- Fichier exécutable :

- gestionVSM\_MAIN  
- client

## II. **Fonctionnement**

Chaque client rentre des transactions dans une fenêtre de transmission, sous forme de chaîne de caractères. Celles-ci sont transmises au serveur à travers la FIFO de transmission.

Après le traitement d'une transaction, le serveur transmet au client un retour d'information (feedback), selon la nature de la transaction.

Chaque retour d'information (feedback) est affichée dans la fenêtre de réception.

```
mi642@dmilinux2:~/Desktop/tp$ ls
client.c      fifo          gestionVMS_MAIN.o  Mult-Numbers.olc3  trans2.txt
gestionVMS.h  gestionListeChaineeVMS.c  gestionListeChaineeVMS.h  gestionVMS.c_
client.c~    gestionListeChaineeVMS.c_  'Sum-Numbers (1).olc3'  gestionVMS.c_
gestionVMS_MAIN  gestionVMS.o  Sum-Numbers.olc3  gestionVMS.h
client.h      gestionListeChaineeVMS.c_  Sum-Numbers.olc3  gestionVMS.c_
gestionVMS_MAIN.c  Makefile_  trans1.txt
client.h~    gestionListeChaineeVMS.c_~  gestionListeChaineeVMS.o  gestionVMS.h
gestionVMS_MAIN.c~  Makefile_  trans1.txt
mi642@dmilinux2:~/Desktop/tp$ gcc -o client client.c -Wall -I. -lpthread -lncurses
mi642@dmilinux2:~/Desktop/tp$ ./gestionVMS_MAIN
```

TRANSMISSION  
Entrez une transaction ou 'q' pour quitter : █

