

Document d'installation



Rédigé par :

Loan BELTRAN

SOMMAIRE

Table des matières

I.	Client.....	3
1.	Prérequis	3
2.	Installation.....	3
3.	Fonctionnement	4
II.	Serveur Maître.....	5
1.	Prérequis	5
2.	Installation.....	6
3.	Fonctionnement	6
III.	Serveur Esclave.....	8
1.	Prérequis.....	8
2.	Installation.....	9
3.	Fonctionnement	9

I. Client

1. Prérequis

Afin d'installer le client, vous devez avoir en votre possession un OS¹ sur lequel sont installés les paquets **python** et **pyqt5**. Python exécutera le code et PyQt5 permettra d'afficher l'interface graphique.

- Si vous êtes sous Windows

1. Télécharger la version de python 3.13.1 sur <https://www.python.org/downloads/>
2. Lancer l'installation et cocher « **Add python to path** ».

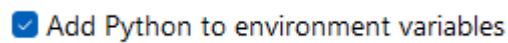


Image 1: Case PATH

3. Vérifier l'installation avec la commande `python --version` dans un cmd.

- Si vous êtes sous Linux :

1. Ouvrir un cmd et y installer python avec les commandes `apt update` et `apt install python3 python3-pqt python3-venv`. Si le téléchargement ne fonctionne pas d'un ou plusieurs paquets ne fonctionne pas, remplacer le répertoire de fichier `/etc/apt/sources.list` par une version miroir.
2. Créer un environnement virtuel : `python3 -m venv sae/`.
3. Activer l'environnement virtuel : `source sae/bin/activate`.
4. Vérifier l'installation avec la commande `python --version` dans un cmd.

Notez qu'il est primordial d'ajouter python au PATH

2. Installation

Dans un premier temps, vous devez récupérer le dossier « **Codes** » sur le github :

<https://github.com/loanbltr/R3.09-SAE3.02/tree/main/SAE3.02/Codes/>

Dans un second temps, récupérer le code « **client.py** » et exécuter le dans un cmd² via la commande `python client.py` sous Windows. Sous Linux, copier le fichier dans le dossier `sae/` préalablement créé et effectuer la commande `python3 client.py` dans l'environnement.

¹ « operating system », il s'agit du système d'exploitation de la machine (windows, debian, macos, ...)

² Invite de commande

3. Fonctionnement

Lors de l'exécution, une interface graphique apparaît sur l'écran de l'utilisateur.

L'utilisateur a la possibilité de choisir l'adresse IP du serveur maître. Il a la possibilité d'entrer manuellement les informations (adresse IP + port) ou de cocher la case *localhost*³ remplissant automatiquement la section *Server IP*.

Notez que l'utilisateur a obligation d'entrer un port ainsi qu'une IP.

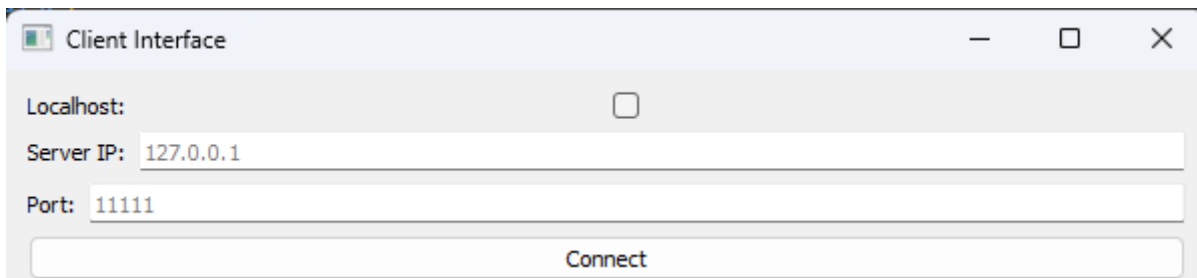


Image 2: Client IP + Port

L'utilisateur doit ensuite sélectionner un code source qu'il souhaite envoyer à l'aide du bouton *Select File*. Le contenu du fichier s'affichera dans la rubrique *File Content*.

Notez que l'utilisateur a la possibilité de modifier le contenu de la rubrique *File Content*.

En cliquant sur le bouton *Send File*, l'utilisateur envoie le contenu au serveur maître.

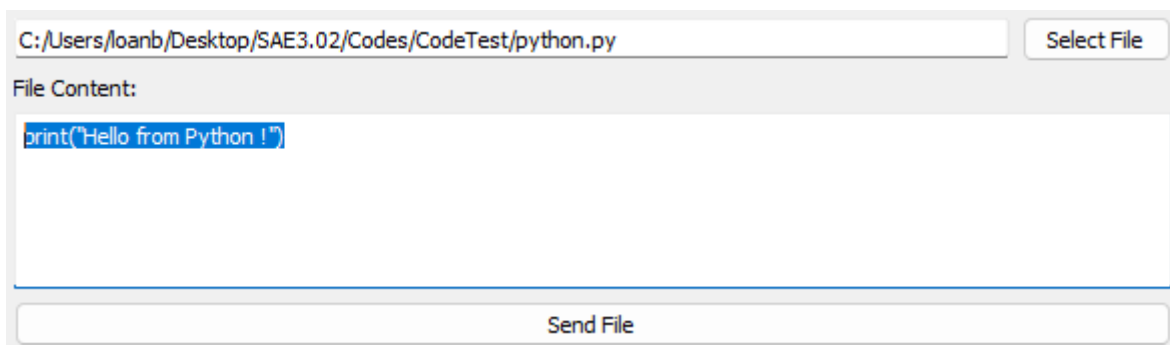


Image 3: Client choix et affichage du fichier

La dernière rubrique affiche le contenu reçu par le serveur. Le contenu retourne le résultat de l'exécution du code.

Notez que si le résultat retourné est « No slave available », aucun serveur esclave n'est actuellement disponible. Patientez quelques secondes et réessayez.

³ Permet de définir l'IP en local sur 127.0.0.1

Messages Received:

Image 4: Reception du message

II. Serveur Maître

1. Prérequis

Afin d'installer le serveur maître, vous devez avoir en votre possession un OS sur lequel sont installés les paquets **python** et **pyqt5**. Python exécutera le code et PyQt5 permettra d'afficher l'interface graphique.

Il est primordial d'installer une VM⁴ Ubuntu sur la version 24.04 pour mener à bien le lancement de l'interface graphique. De plus, il faut que la carte ethernet de la VM soit en mode bridge.

Il faut ensuite télécharger les différents paquets sur la VM.

Installation des paquets python

1. Ouvrir un cmd et y installer python avec les commandes `apt update` et `apt install python3 python3-pyqt5 python3-venv python3-pip`. Si le téléchargement ne fonctionne pas d'un ou plusieurs paquets ne fonctionne pas, remplacer le répertoire de fichier `/etc/apt/sources.list` par une version miroir.
2. Créer un environnement virtuel : `python3 -m venv sae/`.
3. Activer l'environnement virtuel : `source sae/bin/activate`.
4. Vérifier l'installation avec la commande `python --version` dans un cmd.
5. Dans l'environnement virtuel, exécuter la commande `pip install pyqt5`.

Notez qu'il est primordial d'ajouter python au PATH

⁴ Machine virtuelle

2. Installation

Dans un premier temps, vous devez récupérer le dossier « **Codes** » sur le github :

<https://github.com/loanbltr/R3.09-SAE3.02/tree/main/SAE3.02/Codes/>

Dans un second temps, récupérer le code « **ServerMaster.py** », copier le fichier dans le dossier *sae/* préalablement créé et effectuer la commande `python3 ServerMaster.py` dans l'environnement.

Il vous est conseillé d'installer le paquet **SSH** sur votre VM. En effet, Ubuntu ne permet pas de copier-coller des fichiers entre votre machine hôte et votre VM. Avec SSH, il sera donc plus facile de s'y connecter depuis d'autre application comme **Putty** pour pouvoir copier-coller.

3. Fonctionnement

Lors de l'exécution, une interface graphique apparaît sur l'écran de l'utilisateur.


L'utilisateur doit entrer son adresse IP (visible via la commande `ip a`). Il a la possibilité d'entrer manuellement les informations (adresse IP + ports) ou de cocher la case *localhost*⁵ remplissant automatiquement la section *Server IP*.

« Client port » définit le port sur lequel se connectera le client.

« Slave port » définit le port sur lequel se connectera le serveur esclave.

« Slave CPU port » définit le port sur lequel se connectera la partie CPU du serveur esclave.

Notez que l'utilisateur a obligation d'entrer un port ainsi qu'une IP.



The screenshot shows a graphical user interface for the ServerMaster.py application. It features a 'Localhost' checkbox which is currently unchecked. Below it are five input fields: 'IP Master' containing '192.168.1.1', 'Client port' containing '11111', 'Slave port' containing '22222', and 'Slave CPU port' containing '33333'. The fields are arranged vertically with labels to their left.

Image 5: IP et ports du serveur maître

La rubrique « *Messages Received* » affiche le contenu échangé entre les clients et les serveurs esclaves.

⁵ Permet de définir l'IP en local sur 127.0.0.1

Le bouton *Start Server* permet de lancer le serveur une fois que tous les ports et IP sont entrés

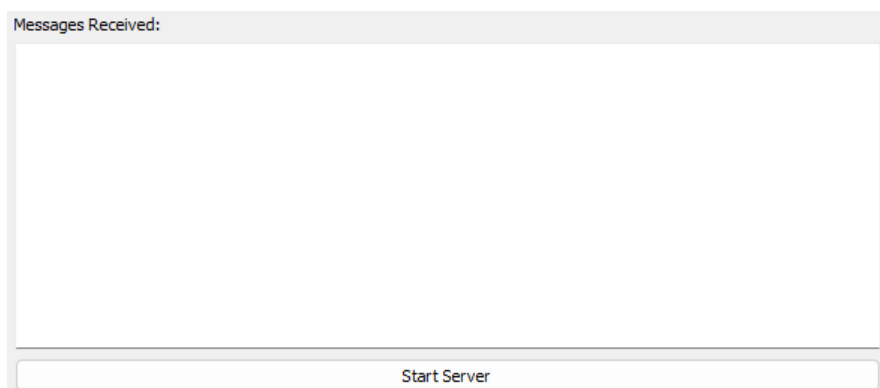


Image 6: Messages reçus du serveur maître

Lorsqu'un ou plusieurs serveurs esclaves sont connectés, l'utilisateur a la possibilité de cliquer sur le bouton « *View Slave Info* » pour afficher toutes les informations liées au serveur esclave.

L'échange de messages CPU et RAM se fait sur un socket différents que l'échange classique afin d'éviter des problèmes de collisions ou de transmissions des messages.

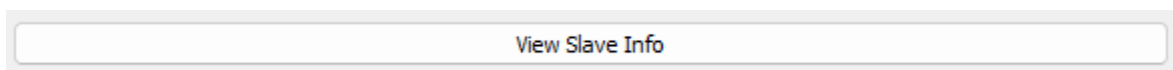


Image 7: Bouton informations serveurs esclaves

A screenshot of a window titled 'Slave Info'. Inside the window, there is a section titled 'Slave Information' which contains a table with five columns: ID, Etat, CPU, RAM, and langage. The table has two rows of data.

ID	Etat	CPU	RAM	langage
5158	False	16.6	58.1	python/java
3284	False	10.1	58.1	python/java

Image 8: Affichage informations esclaves

Ainsi lorsque le serveur maître reçoit un message du client, il vérifie si un serveur esclave est disponible et lui envoie le message en ajoutant le port initial du client au début du message. Il récupère ensuite le message traité et le renvoie au client grâce à son port

III. Serveur Esclave

1. Prérequis

Afin d'installer le serveur maître, vous devez avoir en votre possession un OS sur lequel sont installés les paquets **python** et **psutil**. Python exécutera le code.

Vous pouvez utiliser une VM Debian 12. De plus, il faut que la carte ethernet de la VM soit en mode bridge.

Il faut ensuite télécharger les différents paquets sur la VM.

Installation des paquets python

1. Ouvrir un cmd et y installer python avec les commandes *apt update* et *apt install python3 python3-psutil python3-venv*. Si le téléchargement ne fonctionne pas d'un ou plusieurs paquets ne fonctionne pas, remplacer le répertoire de fichier */etc/apt/sources.list* par une version miroir.
2. Créer un environnement virtuel : *python3 -m venv sae/*.
3. Activer l'environnement virtuel : *source sae/bin/activate*.
4. Vérifier l'installation avec la commande *python --version* dans un cmd.

Notez qu'il est primordial d'ajouter python au PATH

Installation des paquets C

1. Ouvrir un cmd et y installer C avec les commandes *apt update* et *apt install gcc*. Si le téléchargement ne fonctionne pas d'un ou plusieurs paquets ne fonctionne pas, remplacer le répertoire de fichier */etc/apt/sources.list* par une version miroir.

Installation des paquets Java

1. Ouvrir un cmd et y installer java avec les commandes *apt update* et *apt install default-jdk*. Si le téléchargement ne fonctionne pas d'un ou plusieurs paquets ne fonctionne pas, remplacer le répertoire de fichier */etc/apt/sources.list* par une version miroir.

2. Installation

Dans un premier temps, vous devez récupérer le dossier « **Codes** » sur le github :

<https://github.com/loanbltr/R3.09-SAE3.02/tree/main/SAE3.02/Codes/>

Dans un second temps, récupérer le code « **ServerSlave.py** », copier le fichier dans le dossier *sae/* préalablement créé et effectuer la commande `python3 ServerSlave.py` dans l'environnement.

3. Fonctionnement

Afin de lancer le serveur esclave, il faut ouvrir un cmd et y exécuter la commande **python3 ServerSlave.py** accompagné de différentes options :

OBLIGATOIRES :

- **-ip <ip>** : permet de définir l'adresse ip du serveur maître

Exemple d'exécution : -ip 127.0.0.1

- **-l <langage1>/<langage2>/...** : permet de définir les langages que le serveur est capable de traiter.

Exemple d'exécution : -l java/python

- **-ps <port>** : permet de définir le port sur lequel l'échange de code va se faire avec le serveur maître.

Exemple d'exécution : -ps 22222

- **-psc <port>** : permet de définir le port sur lequel l'échange des informations CPU et RAM va se faire avec le serveur maître.

Exemple d'exécution : -psc 33333

OPTIONNELS/

- **-m <taux maximal du CPU (en%)>** : permet de définir la charge maximale du CPU avant que le serveur ne soit plus disponible (par défaut 100).

Exemple d'exécution : -m 60

- **-r <taux maximal de la RAM (en%)>** : permet de définir la charge maximale de la RAM avant que le serveur ne soit plus disponible (par défaut 100).

Exemple d'exécution : -r 70

- **-c <nombre de codes maximale>** : permet de définir le nombre de code maximum que peut effectuer le serveur avant d'être indisponible (par défaut 99).

Exemple d'exécution : -c 4

Une fois le serveur correctement lancé, il envoie toutes les 5 secondes des informations concernant la charge du CPU et de la RAM au serveur maître. Il réceptionne également des codes, les exécute en fonction de ses caractéristiques et les renvoie au serveur maître.

IV. Tables des Illustrations

Image 1: Case PATH..... 3

Image 2: Client IP + Port..... 4

Image 3: Client choix et affichage du fichier 4

Image 4: Reception du message..... 5

Image 5: IP et ports du serveur maître..... 6

Image 6: Messages reçus du serveur maître..... 7

Image 7: Bouton informations serveurs esclaves 7

Image 8: Affichage informations esclaves..... 7