
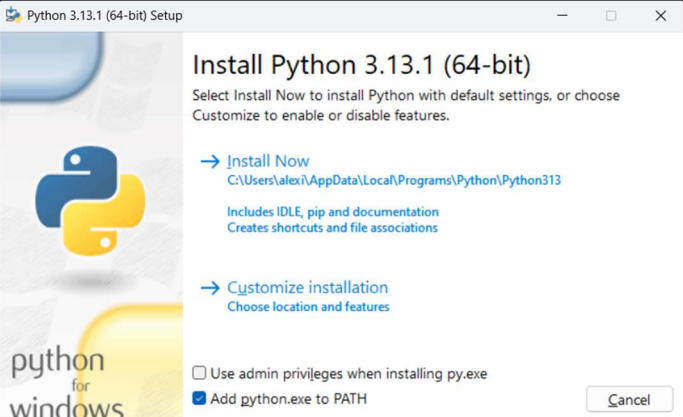


Documentation d'installation :

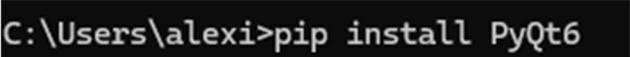
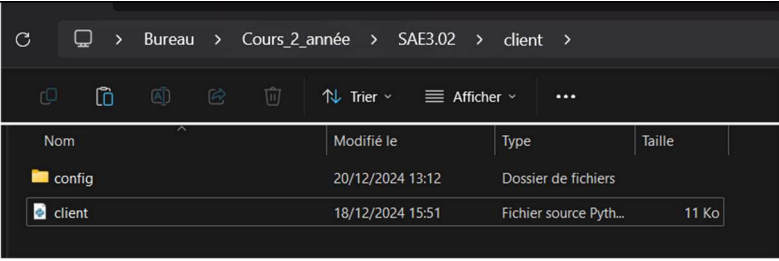
Point Important : chaque serveur doit avoir une adresse IP différente. Si deux serveurs partagent la même adresse IP, cela entraînera un conflit côté client lors de la lecture du fichier de configuration.

Installation Windows :

Installer Python :

Télécharger l'installateur Python :	Python
Une fois le fichier .exe téléchargé, double-cliquez dessus pour lancer l'installation	
Pendant l'installation : <ul style="list-style-type: none">- Cocher la case "Add Python to PATH".- Cliquez sur "Install Now" pour continuer.	

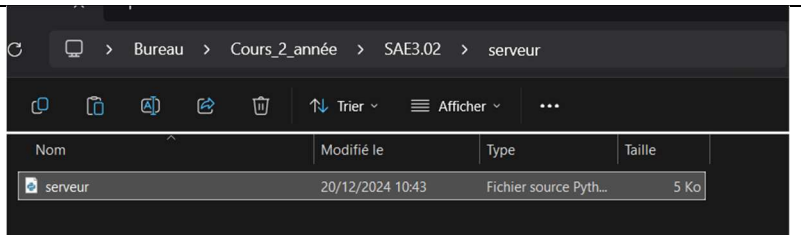
Mise en place du Client :

Ouvrir un terminal et entrez la commande suivante pour installer PyQt6 « pip install PyQt6 » :	
Crée un dossier nommé client et copiez le programme à l'intérieur	Programme client : client.py
Puis crée un dossier nommé « config » avec cette syntaxe, puis crée fichier texte à l'intérieur nommé « config.txt »	

Mise en place du Serveur :

Programme du serveur :

Crée un dossier serveur et copie le programme serveur : [serveur.py](#)



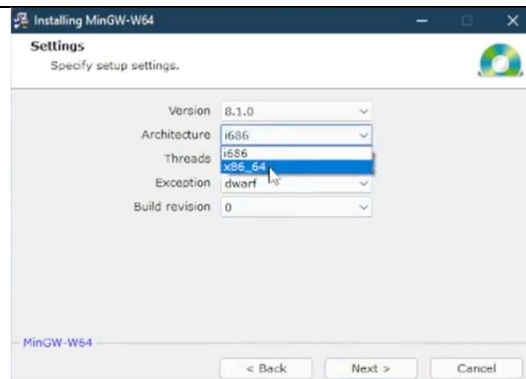
Installation de GCC nécessaire pour le bon fonctionnement serveur :

Source : [Vidéo](#)

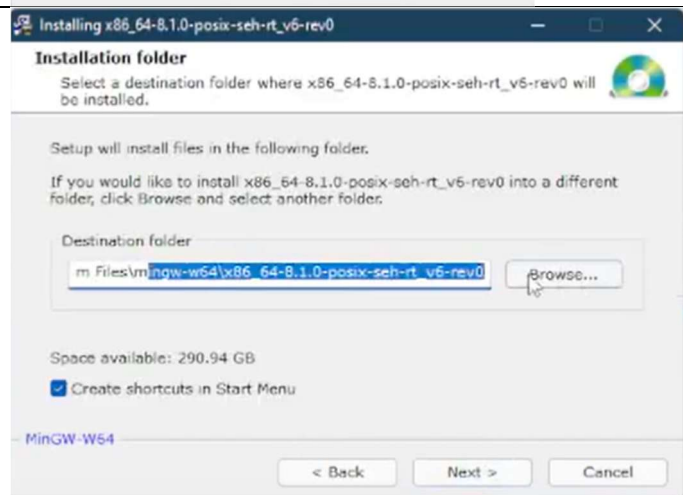
Télécharger le logiciel pour pouvoir compiler du C : [GCC](#)

Une fois téléchargé lancer le logiciel

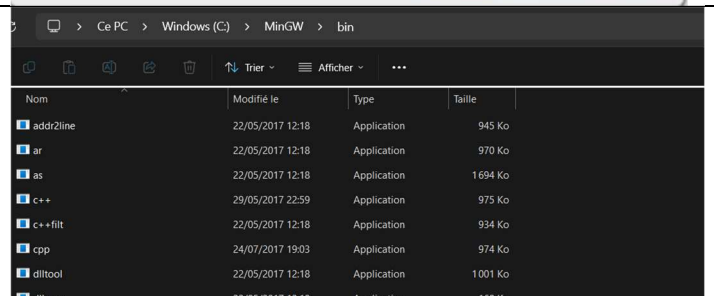
Dans settings sélectionner x86_64 puis cliquer sur Next

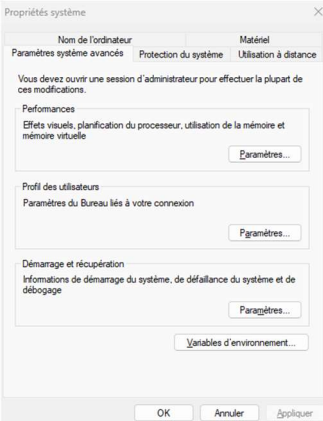


Copiez le destination folder :



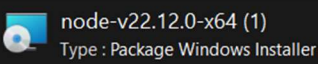
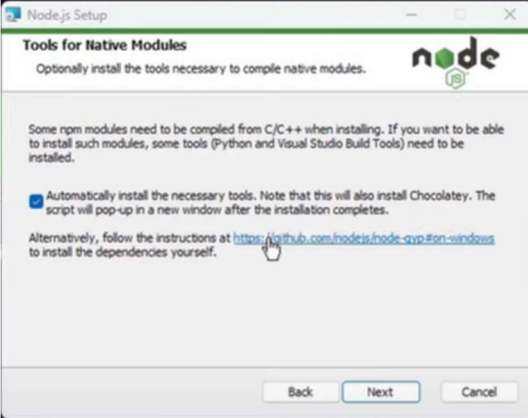
Puis une fois l'installation terminée, aller dans l'explorateur de fichiers, copier le dossier de destination dans l'explorateur de fichiers, aller dans le fichier mingw64, puis dans le fichier bin, puis copier le chemin du fichier.



<p>Aller dans variables d'environnement systèmes et cliquer sur variables d'environnement :</p>	
<p>Cliquer sur variables d'environnement, dans variable utilisateur, double cliquer sur path, nouveau, et copier-coller le chemin, puis cliquer sur ok et fermer</p>	

Installation de NodeJS (nécessaire pour le bon fonctionnement serveur) :

Source : [lien](#)

Installer le logiciel NodeJS	lien
Lancer le logiciel	
<p>Cliquer sur Next jusqu'à « Tools for Native Modules » et cliquer sur « Automatically install ... »</p>	
Puis finir l'installation	

Installation Linux :

Installation de python :

Mettez-vous en superutilisateur dans un terminal	Su puis entrez le mot de passe
Mettre à jour apt	root@dEBIAN:/home/vboxuser# apt update
Installer Python	root@dEBIAN:/home/vboxuser# apt install python3

Mise en place du client :

Crée un dossier client et copiez le programme	Programme client : client.py
Ouvrir un terminal de commande et lancer la commande	Sudo apt install python3-pyqt6

Mise en place du serveur :

Crée un dossier serveur et copiez le programme serveur	lien
Installer GCC	root@dEBIAN:/home/vboxuser# apt install build-essential
Installer NodeJS	root@dEBIAN:/home/vboxuser# apt install build-essential
Naviguer jusqu'au dossier où est contenu le programme serveur dans le terminal et exécuter la commande suivante	root@dEBIAN:/home/vboxuser/Downloads# nano serveur.py
Modifier la troisième ligne « python » en « python3 » après le commentaire Exécution selon l'extension et enregistrer le programme	<pre># Exécution selon l'extension if extension == 'py': resultat_execution = subprocess.run(["python3", nom_fichier_temp], capture_output=True, text=True</pre>

Installation MacOS :

J'ai rencontré des difficultés pour obtenir une machine MacOS, ce qui fait que ma documentation d'installation ne restera que purement théorique

Installation de Python :

Installer Homebrew (pour la gestion des paquets) taper la commande suivante dans un terminal	<code>/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"</code>
Ajouter Homebrew au PATH : lancer les deux commande suivante	<code>echo 'eval "\$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile eval "\$(/opt/homebrew/bin/brew shellenv)"</code>
Installer python	<code>brew install python3</code>

Installation client :

Crée un dossier client et copiez le programme	Programme client : client.py
Ouvrir un terminal de commande et lancer la commande	<code>Pip3 install PyQt6</code>

Installation serveur :

Crée un dossier serveur et copiez le programme serveur	serveur.py
Installer GCC Lancer un terminal et entrer les commandes suivantes : Verifications de l'installation :	<code>« xcode-select –install » et « brew install gcc » gcc-<version> --version</code>
Installer NodeJS Lancer un terminal et entrer les commandes suivantes : Verifications de l'installation :	<code>« brew install node » et « npm install -g npm@latest » node --version npm --version</code>
Naviguer jusqu'au dossier où est contenu le programme serveur et choisir d'ouvrir avec TextEdit	

Modifier la troisième ligne
« python » en « python3 »
après le commentaire
Exécution selon l'extension
et enregistrer le
programme

```
# Exécution selon l'extension
if extension == 'py':
    resultat_execution = subprocess.run(
        ["python3", nom_fichier_temp],
        capture_output=True,
        text=True
```

Lancement du projet :

Lancer le serveur (à lancer avant le client) :

<p>Windows :</p> <ul style="list-style-type: none">- Lancer un terminal et aller dans le dossier serveur grâce à la commande cd, puis lancer la commande- Puis lancer la commande ipconfig et relever l'IP de votre machine	<p>« python serveur.py <port> » Remplacer <port> par le port sur lequel vous voulez lancer le serveur</p> <p><i>Conseille : le port doit être supérieur à 1000 pour éviter le conflit avec d'autres ports utilisés</i></p>
<p>Linux :</p> <ul style="list-style-type: none">- Lancer un terminal et aller dans le dossier serveur grâce à la commande cd, puis lancer la commande- Puis lancer la commande ip a et relever l'IP de votre machine	<p>« python3 serveur.py <port> » Remplacer <port> par le port sur lequel vous voulez lancer le serveur</p> <p><i>Conseille : le port doit être supérieur à 1000 pour éviter le conflit avec d'autres ports utilisés</i></p>
<p>MacOs :</p> <ul style="list-style-type: none">- Lancer un terminal et aller dans le dossier serveur grâce à la commande cd, puis lancer la commande- Puis lancer la commande ifconfig pour trouver l'interface réseau active et ipconfig getifaddr <int> int: active et relever l'IP de votre machine	<p>« python3 serveur.py <port> » Remplacer <port> par le port sur le quelle vous voulez lancer le serveur</p> <p><i>Conseille : le port doit être supérieur à 1000 pour éviter le conflit avec d'autres ports utilisés</i></p>

Lancer le client :

<p>Configuration du fichier config.txt : (Cette opération est primordiale avant le lancement d'un client)</p> <p>Aller dans le fichier config .txt crée précédemment</p>	<p>Rentrer l'ip avec le port annoncé précédemment pour chaque serveur en suivant la syntaxe suivante : ip_serveur port</p> <p>Contrainte :</p> <ul style="list-style-type: none">- Il ne doit pas avoir d'espace entre « » et l'ip et le port doit être celui noté précédemment.- Entre chaque annonce de serveur un retour à la ligne est requis sauf pour la dernière ip annoncée où il ne doit pas y avoir d'espace
--	---

	<p>Exemple :</p> <p>192.168.1.1 1001</p> <p>192.168.1.2 1002</p>
<p>Sur Windows :</p> <p>Lancer un terminal et naviguer jusqu'au dossier contenant le programme client grâce à la commande « cd » et lancer la commande suivante</p>	<p>« python client.py »</p>
<p>Sur Linux et MacOS :</p> <p>Lancer un terminal et naviguer jusqu'au dossier contenant le programme client grâce à la commande « cd » et lancer la commande suivante</p>	<p>« python3 client.py »</p>
<p>Utilisation de l'interface graphique :</p> <ul style="list-style-type: none"> - IP serveur : rentrer l'ip du serveur - Bouton fichier : Choisir le programme à exécuter. Il doit être au format « .js », « py » et « .c » sinon il ne sera pas visible dans l'explorateur de fichiers - Menu déroulant (à côté du bouton fichier) : sélectionner le bon format de fichier <i>(si le format n'a pas été indiqué correctement le serveur exécutera le programme mais cela renverra une erreur systématiquement car l'extension ne sera pas bonne)</i> - Bouton Upload : Si des erreurs ont été relevées, affiche les erreurs rencontrées. Se connecte au serveur, exécute le programme si le serveur est occupé et joignable. Enfin cela renvoie le résultat du programme s'il a été exécuté. 	