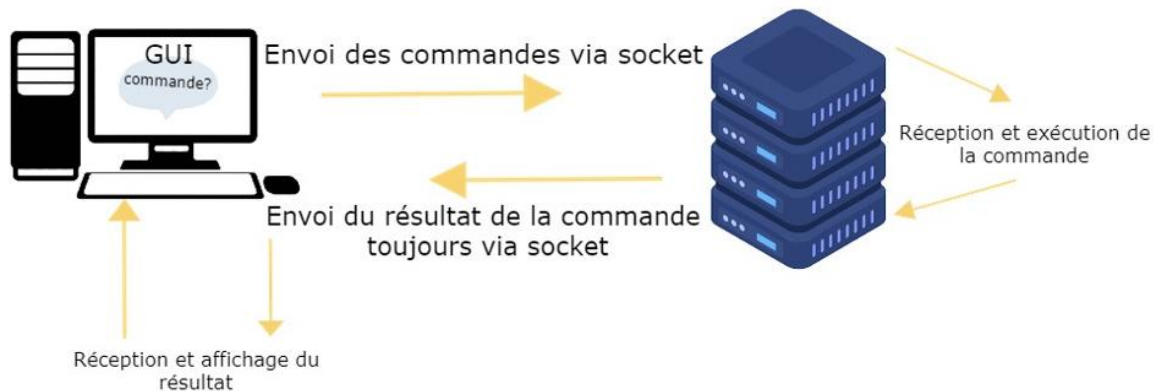


Documentation programmeur

Structure de l'application :



Architecture du code :

J'ai créé deux fichiers python, un pour le serveur et un pour le client.

Dans celui du serveur se trouve le code à faire tourner sur un serveur pour le faire écouter les messages venant des hôtes, et qui, lorsqu'il reçoit un certain message (la commande envoyée par le client), il va faire appel à une fonction qui va envoyer les informations demandées par le client ou le résultat de sa commande.

Dans celui du client, j'ai regroupé directement le GUI et le client. On a donc deux classes ; Client et MainWindow.

La classe client s'occupe uniquement de la configuration de la connexion socket client, et d'envoyer les messages au serveur.

Dans la classe MainWindow, on retrouve la quasi-totalité des fonctionnalités étant donné que nous gérons notre programme depuis le GUI. Dans cette classe est défini le style du GUI, l'organisation de celui-ci (en grid) avec PyQt5. La classe gère aussi les actions liées à l'appuie sur les boutons, qui font généralement appel à une fonction qui va prendre notre 'objet' connexion socket et depuis celui-ci appeler la fonction d'envoi de la classe Client avec le

message/commande sélectionnée et s'occuper d'afficher les résultats dans la box principal, et la commande dans l'historique des commandes, comme ici par exemple :

```
self.__lstcmd.activated.connect(self.othercom)
```

```
def othercom(self):
    if self.__lstcmd.currentText() == 'ping':
        text, ok = QDialog.getText(self, 'IP', 'Adresse IP à ping :')
        if ok and self.is_valid_ip(text):
            try:
                cmd = 'ping ' + text
                self.__logscmd.append(cmd)
                self.conn.envoi(cmd)
                self.__logs.append(self.conn.envoi(cmd))
            except:
                msg2 = QMessageBox()
                msg2.setWindowTitle('Erreur')
                msg2.setIcon(QMessageBox.Critical)
                msg2.setText('Le ping ne passe pas !')
                msg2.exec_()
        else:
            msg2 = QMessageBox()
            msg2.setWindowTitle('Erreur')
            msg2.setIcon(QMessageBox.Critical)
            msg2.setText('Adresse IP invalide !')
            msg2.exec_()
    elif self.__lstcmd.currentText() == 'Commande personnalisée':
        cmdpers, ok = QDialog.getText(self, 'Commande personnalisée', 'Commande')
        if ok:
            try:
                self.__logscmd.append(cmdpers)
                self.conn.envoi(cmdpers)
                self.__logs.append(self.conn.envoi(cmdpers))
            except:
                msg2 = QMessageBox()
                msg2.setWindowTitle('Erreur')
                msg2.setIcon(QMessageBox.Critical)
                msg2.setText('Commande non reconnue par l\'OS !')
                msg2.exec_()
```

On retrouve aussi une fonction de vérification des IP par exemple :

```
def is_valid_ip(self, ip):
    regex = "^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$"
    if re.search(regex, ip):
        numbers = ip.split(".")
        for number in numbers:
            if int(number) > 255:
                return False
        return True
    return False
```

Qui sera sollicitée lorsque l'utilisateur sera mené à renseigner une adresse IP.

Qu'avez-vous fait ? et pourquoi vous n'avez pas tout fait si c'est le cas ?

J'ai fait toutes les commandes demandées, et les commandes personnalisées selon l'OS (même si je n'ai pas eu l'occasion de tester sur des machines avec des OS différents de Windows donc c'est possible qu'il y ai des erreurs sur l'exécution de celles-ci.)

J'ai essayé de gérer la plupart des exceptions possibles mais il est probable que j'en ai oubliée.

Je n'ai pas fait les options supplémentaires et mon programme ne fonctionne qu'avec une seule machine à la fois car je n'ai pas réussi à faire en sorte qu'il se connecte à plusieurs machines en même temps avec les threads.