

Cahier des charges d'une interface de surveillance de serveurs ou de machines clients

Contexte

A l'image du protocole SNMP, nous souhaitons réaliser un système client-serveur permettant d'envoyer des commandes systèmes permettant de superviser et de diagnostiquer des machines ou serveurs à distance.

Fonctionnalités

Nous souhaitons une interface graphique permettant de réaliser l'ensemble des fonctionnalités suivantes. Sur la base d'un système client-serveur, vous devrez pouvoir interroger des serveurs sur leur état et leur fonctionnement et même détecter lorsqu'ils sont déconnectés du réseau. Nous souhaitons à partir avoir une liste d'IP ou de machines sous la forme de fichier d'interroger ces mêmes machines sur leur état.

L'interface graphique permettra d'effectuer les commandes suivantes :

- Lire le fichier de machines
- Pouvoir ajouter des machines et enregistrer ces nouvelles machines dans le fichier
- Envoyer des commandes simples à l'ensemble des machines :
 - OS : demande l'OS et le type d'OS par exemple MacOS Darwin ou Linux Ubuntu 21.4
 - RAM : mémoire totale, mémoire utilisée et mémoire libre restante
 - Disconnect : déconnexion de l'interface permettant de libérer la machine monitorée pour permettre de libérer le serveur pour d'autres requêtes
 - Connexion information : IP, nom de la machine
 - Kill : tue le serveur
 - Reset : reset du serveur
 - CPU : utilisation de la CPU
- De plus, l'interface permettra d'envoyer des commandes données par l'utilisateur, par exemple :
 1. DOS:dir
 2. DOS:mkdir toto
 3. Linux:ls -la
 4. Powershell:get-process
 5. python --version
 6. ping 192.157.65.78

Pour chacune des commandes lorsque le type de système est spécifié par exemple « DOS: », « Linux: », « Powershell: », l'OS sera vérifié pour compatibilité et la commande exécutée et renvoyer au système de monitoring. Lorsque d'aucun système n'est spécifié (commandes 5 et 6), la commande est lancée quel que soit le système.

Quelle que soit la commande utilisée, l'interface devra afficher le résultat de la commande et garder en historique les commandes exécutées sur l'ensemble des machines. Les machines où sont lancées les commandes pourront être choisies par une simple sélection entre 1 et l'ensemble des machines. Il serait également intéressant de savoir si une machine est connectée (et ne pas lancer de commandes si la machine n'est pas en ligne) et l'OS de la machine.

Nous n'avons pas spécifiquement de prérequis sur l'interface mais elle se doit d'être ergonomique et facile d'utilisation tant pour un administrateur système que pour un débutant averti sur l'administration système.

Points d'attention

Vous devrez bien sûr gérer l'ensemble des erreurs possibles et éviter tous les crashes de votre application.

Deux documentations de qualité en anglais devront être rédigées pour l'utilisateur de l'interface et sur un document du développement effectué afin de pouvoir ajouter de nouvelles fonctionnalités (ou commandes) si nécessaire.

Options supplémentaires (à chiffrer) :

- Filtrage des commandes exécutées sur une ou plusieurs machines pour connaître l'état.
- Un graphique sur l'utilisation de la CPU ou la mémoire.
- Nous nous posons également la question sur une connexion sécurisée et chiffrée entre le client et le serveur, il serait intéressant de voir si sur le réseau les commandes peuvent apparaître à l'aide d'un outil tel que Wireshark. Si les commandes ne sont pas sécurisées, il serait donc intéressant de les sécuriser.