Manuel technique de l’application SystemInfoDaemon

Tables des matières

Description

L'architecture de l'application permettant la communication entre le démon et le client a pour obligation de respecter ce schéma suivant :

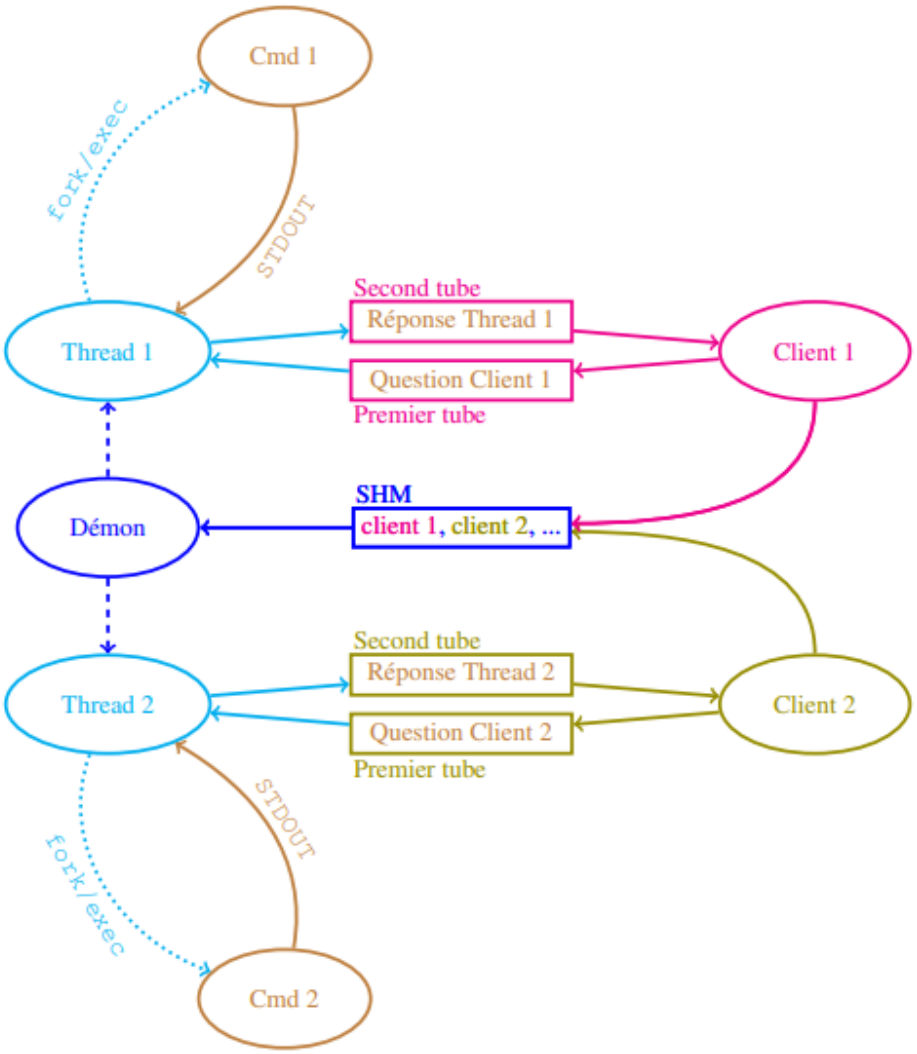

FIGURE 1 – Schéma de l'architecture client/démon

Nous pouvons voir qu' un client se place dans le SHM, puis le démon alloue un thread pour que le client puisse communiquer avec ce thread. Lorsque le client envoie une requête vers le premier tube (question), le thread crée un processus qui exécute la commande et ce processus retourne le résultat dans la sortie standard du thread. Par la suite, la réponse est envoyé au client par le second tube (réponse).

Traitement de la zone de mémoire partagée

La zone de mémoire partagée se représente par une file de clients qui sont représentés par l'identifiant du processus client (PID). Cette file de clients possède une taille maximum. Ainsi si la taille est dépassée, le client souhaitant entrer dans la file sera mis en attente. Lorsqu'un client sort de la file, il libère une place pour un autre client en attente.

Ce traitement utilise l'algorithme du producteur-consommateur afin de synchroniser l'accès à la file qui est la ressource partagée.

Traitement de l'allocation d'une ressource et de la requête

Le démon qui récupère le pid du client dans la file du shm va donc créer un thread pour le client. Le thread en question va ouvrir les tubes que le client aura préalablement créé. C'est à cette étape que la communication étant ouverte, le client peut donc envoyer ses requêtes.

Lorsqu'un client envoie une requête, le thread va créer une tube anonyme permettant au processus enfant qui va être créer d'exécuter la requête avec un execvp. De plus, le processus enfant redirige sa sortie d'erreur et standard dans ce tube anonyme. Pendant ce temps-là, le processus parent se met en attente d'un résultat que l'enfant va lui communiquer ainsi le thread lit l'entrée du tube pour avoir le résultat de la commande. Enfin, il envoie la commande dans le tube réponse du client. Cette opération peut se répéter autant de fois que le client le souhaite.

Libération des ressources

Pour savoir quels sont les clients qui occupent une ressource, il y a une structure qui va enregistrer le pid du client avant que son thread ne soit créé. Ainsi lorsqu'un client doit s'éteindre, il va libérer ses ressources grâce à son pid enregistré puis cette structure réinitialisera l'identifiant afin de signifier qu' une place est à nouveau libre.