



Programmation multitâche

TP n°5 - Mini-projet de programmation

Réalisation d'un serveur multi-threads

Principes généraux de l'application

Comment fonctionnera notre serveur ?

Un serveur a plusieurs tâches à exécuter en parallèle : il doit être à l'écoute des clients qui souhaitent se connecter, mais il doit aussi s'occuper des clients déjà connectés. Nous choisissons d'utiliser un contexte multi-threads. Nous pourrions aussi utiliser un contexte multi-processus mais l'utilisation des threads est suffisante et les threads sont moins lourds à gérer pour le système d'exploitation.

Notre serveur fonctionnera de la manière suivante : le processus principal sera en charge d'écouter les connexions entrantes. Pour chacune d'elles, il devra créer un thread qui sera dédié à cette connexion. Il devra aussi créer tous les éléments nécessaires à ces connexions et à leur gestion.

Comment vont se faire les communications ?

Dans le cas habituel d'un serveur public, la communication entre le serveur et les utilisateurs est assurée par le réseau. Dans notre cas, cette communication sera assurée par des tubes nommés, ce qui en limite la puissance, mais ce qui nous permet d'illustrer les concepts étudiés en cours.

Quels seront les services assurés par ce serveur ?

Nous simulerons un certain nombre de services par des appels de fonctions produisant un message réponse quelconque. Nous pouvons imaginer que ce serveur est un serveur web produisant des réponses sous la forme de textes en langage html, mais cela est d'un intérêt mineur pour notre exercice.

Eléments d'implantation

Structure générale de l'application

L'application est constituée de deux programmes :

- Le programme serveur,
- Le programme utilisateur.

La communication entre ces deux programmes est assurée par des tubes nommés.

Le programme serveur

Le serveur doit être activé avant toute autre opération. Le serveur commence par mettre en place le tube nommé avec lequel les utilisateurs vont pouvoir se connecter et soumettre leurs requêtes. Si nécessaire, le serveur crée les objets nécessaires à la bonne gestion des services.

Après cette initialisation, le serveur doit boucler sur la lecture des requêtes dans le tube d'entrée. Pour chaque requête reçue, le serveur va créer un thread qui sera chargé de traiter cette requête. Comme il est probable que la requête nécessite une réponse, le serveur, ou plus exactement le thread dédié à cette requête, va transmettre cette réponse dans un tube nommé, tube nommé qui aura été créé auparavant par l'utilisateur et dont l'identité a été fournie au serveur en même temps que la requête. Le thread disparaît lorsque la requête a été traitée.

Dans un objectif de maintenance, voire d'obligation légale, le serveur conservera une trace de son activité dans un fichier « log ».

Le programme utilisateur

Le programme utilisateur est appelé par tout utilisateur souhaitant émettre une requête au serveur. Le programme utilisateur transmet sa requête et toute information complémentaire indispensable, dans un tube nommé public dont l'identité est connue de tous. Le programme utilisateur a pour charge de créer le tube nécessaire à la transmission de la réponse à sa requête. Ce programme devra indiquer l'identité de ce tube au serveur afin que ce dernier puisse lui transmettre la réponse à sa requête. Le programme utilisateur aura aussi pour charge de supprimer ce tube nommé en fin de traitement.

Le traitement des requêtes.

Comme il a été dit précédemment, le thème des requêtes traitées dans cet exercice est sans intérêt. On simulera ces traitements par des appels de fonctions diverses.

Travail demandé

Ce travail sera réalisé de manière autonome par chaque étudiant.

Chaque étudiant doit produire (répertoire compressé) :

- Un court document d'analyse expliquant et justifiant l'architecture et les choix retenus dans la solution ainsi que les fichiers contenant les sources de la solution en langage C ;
- Des états de sortie montrant une ou plusieurs connexions significatives ainsi que le fichier log du serveur ;
- Une auto-évaluation de la solution proposée.

*Date limite de remise des dossiers : **Vendredi 9 décembre 2011***

A transmettre par mail à votre enseignant responsable de TP.