PG208, Projet n°3 : Serveur HTTP évolué

Bertrand LE GAL[*], Serge BOUTER[†] et Clément VUCHENER[‡]

Filière électronique 2^{eme} année - Année universitaire 2011-2012

1 Introduction

1.1 Objectif du projet

L'objectif du projet est de mettre en œuvre les notions appréhendées durant l'enseignement de de PG208. Durant cet enseignement vous avez normalement acquis les notions de base de la programmation orientée objets appliquées au langage C++. Le point de départ du projet est constitué du présent document. Ce dernier correspond au cahier des charges spécifié par le client (votre enseignant). Afin de mener à bien ce projet, vous allez devoir :

- lire le cahier des charges,
- identifier les besoins exprimés par le client,
- comprendre ce que vous devez développer,
- élaborer un modèle de solution,
- développer et mettre au point votre application.

Pour mener à bien les quatre premières étapes, il vous est demandé de mettre en pratique les notion d'UML vues en cours. Cela vous permettra de structurer vos idées et de les partager (avec votre binôme et votre enseignant). De plus, étant donné que le projet est un projet de conception objets, il vous est demandé dans les deux dernière étapes de mettre en œuvre les notions vues en cours (héritage, classes, polymorphisme, surcharge, etc.).

1.2 Déroulement du projet

La durée estimée du projet est d'environ 18 heures. Six séances de TP de 3 heures sont programmés dans votre emploi du temps. Toutefois, elles ne sont pas toutes encadrées : les 2 premières et les 2 dernières

[*]bertrand.legal(at)ims-bordeaux.fr
[†]serge.bouter(at)u-bordeaux1.fr
[‡]clement.vuchener(at)inria.fr

séances sont encadrées tandis que les 2 séances intermédiaires sont planifiées mais se dérouleront sans encadrement.

Le sujet du projet a été écrit de manière à prendre en considération les différences de niveaux entre les groupes. Pour cela le sujet du projet est écrit sous forme de cahier des charges à tiroir. Cette particularité vous permettra d'avancer à votre rythme.

1.3 Évaluation de votre travail

La notation du projet se basera sur :

- La qualité de l'analyse du cahier des charges. Cette analyse doit être réalisée à l'aide d'UML.
- L'utilisation adéquate des possibilités offertes par le langage C++ (héritage, surcharge, ...).
- Le respect des fonctionnalités spécifiées dans le cahier des charges.

L'évaluation du projet sera réalisée en deux parties :

- Lors d'une présentation orale de votre projet.
- Dans un rapport papier détaillant le développement de votre application.

Lors de la présentation orale de votre application, vous êtes en charge de présenter à votre enseignant (le client) l'application développée. Cette présentation d'une durée approximative de 10 minutes vise à démontrer au client que vous avez développé ce qu'il vous a commandé. Pour cela, vous présenterez vos choix de conception, vous expliquerez les différentes étapes par lesquelles vous êtes passées et enfin une réaliserez démonstration des fonctionnalités de l'outil (démonstration que vous aurez pris soin de préparer).

En ce qui concerne le rapport écrit, une attention particulière sera portée à la présentation des choix d'implantation que vous aurez réalisés lors de votre développement. Ces choix ainsi que votre cheminement devront être détaillés clairement à l'aide du langage UML. De plus une analyse des avantages et inconvénients du langage C++ par rapport au langage C devra être réalisée.

2 Cahier des charges

Durant ce projet vous allez devoir concevoir un serveur HTTP évolué. Pour réaliserce développement vous devrez développer certains objets et en réutiliser d'autres. L'objectif pédagogique principal de ce projet est de vous faire réutiliser les classes proposées par la bibliothèque objets QT dont l'utilisation aisée nécessite toutefois un petit temps d'adaptation.

Le serveur internet que vous allez développer devra permettre à tout client internet de venir lire des pages internet se trouvant sur le poste où le serveur (votre application est lancée). Pour des raisons de simplicité, dans un premier temps, l'outil possèdera une interface en ligne de commande.

Voici la liste des fonctionnalités que devra supporter votre serveur internet :

- Traitement des requêtes de lecture de fichier,
- Traitement des requêtes de lecture de fichier (scripts),
- Traitement des requêtes de lecture du contenu d'un répertoire (listing du contenu des fichiers et répertoires contenus à une adresse),
- Traitement des erreurs (page non présente, pas de droit d'accès, etc.),
- Mémorisation des statiques d'utilisation,
- Génération d'une page d'information à la volée sur l'identité du serveur (nom des concepteurs, date et heure de compilation, etc.),
- Génération d'une page d'information à la volée sur les statistiques du serveur (combien de requêtes ont été reçues, traitées, etc.),
- Un gestionnaire de cache des fichiers récemment accédés : afin de réduire la consommation d'énergie, les accès disques, les calculs a réaliser tout en augmentant le débit du serveur, nous souhaitons mémoriser en mémoire RAM les pages récemment accédées afin d'évider de devoir les relire sur le disque dur ou le réseau à chaque requête.

2.1 Les statistiques d'utilisation

Les statistiques d'utilisation de votre serveur internet devront vous permettre de connaître à tout moment les informations suivantes :

- **Le nombre de requêtes reçues** par le serveur,
- **Le nombre de requêtes traitées** (en partie et jusqu'au bout) ;
- **Le nombre d'erreurs** de chaque type ;
- **Le nombre de clients** s'étant connectés,
- **Le nombre d'octets transmis et reçu** du réseau,
- **L'ensemble des requêtes reçues** par le serveur,
- **Le nom des fichiers** téléchargés ainsi que ne nombre de fois où ils l'ont été,

La liste des informations énoncée ci-dessus peut être complétée par d'autres points que vous trouverez pertinents ?

2.2 Les statistiques d'utilisation

Afin de rendre la vie de l'administrateur de notre entreprise la plus simple possible, nous souhaitons que ce dernier puisse analyser à distance ce qui se passe sur le serveur. Pour cela, il a été décidé que lorsqu'un utilisateur demande à accéder à la page internet nommée `./private/statistiques.html` alors vous devez générer une page internet à partir des données courantes. Ces données seront affichées de manières claires et concises.

2.3 Le gestionnaire de cache

Le gestionnaire de cache a pour rôle de minimiser l'accès aux périphériques de masse (disque dur, carte mémoire, clef usb, réseau, etc.). Pour cela, les fichiers récemment utilisés sont temporairement stockés en mémoire, disponibles en cas de nouvelle demande. Ainsi, le serveur lorsqu'il reçoit une requête va regarder en mémoire si le fichier est présent, si oui alors il l'envoie. Dans le cas contraire, ce dernier est chargé depuis le périphérique de stockage sur lequel il se situe et est placé en mémoire si cela est possible (le résultat de l'exécution des scripts pouvant varier, ces derniers ne sont évidemment pas placés en mémoire).

Le pendant de cette technique est la saturation inévitable de la mémoire du système embarqué. Afin d'éviter cela, on limite l'espace mémoire disponible pour le cache. Lorsqu'un fichier est chargé dans le cache, on regarde si l'espace mémoire occupé est supérieur à la borne fixée, si c'est le cas alors on doit supprimer des fichiers s'y trouvant. La politique de suppression que vous adopterez est nommée « Last Recently Used (LRU) » et vise à supprimer les fichiers les plus anciens.

2.4 Pages de configuration

Afin de faciliter encore la vie de l'administrateur qui sera bien souvent éloigné de ses systèmes embarqués, nous souhaitons que ce dernier puisse à distance effectuer des opérations de maintenance élémentaires. Pour cela nous voulons qu'il puisse : Consulter l'état du cache (nombre de fichiers, noms des fichiers, mémoire occupée, taux d'utilisation des fichiers et statistiques du cache) en se connectant à l'adresse « ./private/cache.html ».

Vider le cache des fichiers qu'il contient à l'aide de l'adresse `./private/clear_cache.html` Désactiver ou activer la réponse du serveur aux clients « Service unavailable » à l'aide des pages `./private/(des) activate.html`

3 Point de départ de la conception

Le point de départ de votre conception vous sera fourni par votre enseignant lors de la première séance de projet. Le code source qui vous sera distribué assure les fonctions suivantes :

- Création d'un serveur réseau écoutant sur le port 8080 en mode TCP/IP,
- Gestion des connexions réseaux entrantes à chaque connexion entrante, un objet MyClientSocket est crée et sa méthode run est invoquée.
- L'affichage à l'écran des requêtes provenant du client,
- Un exemple de réponse au client.

Basez-vous sur l'ensemble de ce travail afin d'étendre les capacités de l'outil dans le but de répondre à toutes les attentes du client.

4 Extension de l'application

Maintenant que nous avons développé un serveur internet embarqué, nous souhaitons développer une légère interface graphique basée sur la bibliothèque QT afin d'afficher en temps réel et les statistiques du serveur. Mettez au point une telle interface à l'aide de QT et faite en sorte que les informations soient rafraîchies toutes les secondes à l'écran. Pour réaliser cette tâche, vous pourrez utiliser les nombreux tutoriaux présentant la programmation graphique à l'aide de QT présent sur internet.

5 Outils de développement

Afin de mener à bien ce projet, vous allez devoir changer d'environnement de développement. La bibliothèque objets QT nécessite des scripts de compilation (makefile) que le logiciel éclipse ne sait pas gérer, nous allons donc nous rabattre sur QtCreator qui est développé par TrollTech (maison mère de Qt).

Vous pourrez télécharger une version gratuite des bibliothèques QT, du compilateur g++ ainsi que de l'éditeur à l'adresse suivante :

`http://www.qtsoftware.com/downloads`

6 Classes utiles

La bibliothèque QT offre des centaines de classes toutes plus utiles les unes que les autres ? Ci-dessous je vous en cite un certains nombre qui peuvent vous servir dans le cadre de ce projet, mais beaucoup d'autres sont aussi utilisables (cette liste est non exhaustive) :

- QString : Gestion des chaines de caractères
- QHash : Structure de données associative (pratique pour le cache)
- QVector : Structure de données itérative
- QFile : Objet permettant de lire, d'écrire et de vérifier l'existence d'un fichier (QFileInfoList + QFileInfo),
- QByteArray : Object permettant de manipuler des tableaux de données 8 bits en mémoire,
- QDir : Object représentant un répertoire (consultation des fichiers présents, vérification de l'existante du répertoire, etc.).