ZiaaiZ

FICHE PROJET

Ceci est notre fiche projet officielle du server web http/1.1

* Développeurs promotions Epitech - 2010s :
* Caner Candan
* Florent Hochwelker
* Majdi Toumi
* Morgan Armand

Description

Une description juste d’un serveur http serait celle-ci :

« Un [**serveur**](http://fr.wikipedia.org/wiki/Serveur_informatique)**HTTP** (…) est un [logiciel](http://fr.wikipedia.org/wiki/Logiciel) servant des requêtes respectant le [protocole de communication](http://fr.wikipedia.org/wiki/Protocole_de_communication) [client-serveur](http://fr.wikipedia.org/wiki/Client-serveur) [HyperText Transfer Protocol](http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP), qui a été développé pour le [World Wide Web](http://fr.wikipedia.org/wiki/World_Wide_Web). » *[Wikipedia]*

Par mis les serveurs qui sont les plus utilises, nous comptons :

* Apache http Server (Apache Software Foundation)
* Internet Information Services ou ISS (Microsoft)
* Sun Java System Web Server (Sun Microsystems)

Notre objectif est de crée un serveur HTTP capable de gérer les requêtes classique de demandes de pages et de documents HTTP, les exécutions d‘interface passerelle commune (CGI), les connexions sécurisées et il devra intégrer un dispositif de gestion de modules…

Nous développons notre serveur HTTP nomme Zia dans l’optique de proposer aux utilisateurs un serveur aussi fiable que ceux déjà présents sur le marché prenant en charge les mêmes fonctionnalités : interprétations de langage, interface de passerelle commune, un système de configuration intuitif, …

Contraintes et recommandations

**1). Contraintes :**

Pour mener a bien notre projet, certaines contraintes nous son imposes, notre serveur doit :

* Etre implémenté en langage C++.
* Comprendre les connexions sécurisées par SSL.
* Etre une implémentation exhaustive du protocole HTTP/1.1, respectant strictement la RFC 2616 (sauf exception du support des proxys).
* Etre complètement configurable par un fichier de configuration XML.
* Etre multiplateforme, c'est-à-dire qu’il doit fonctionner sous Windows, Linux/FreeBSD.
* Ne pas utiliser de librairies extérieurs (mis a par les librairies XML ou OpenSSL).
* Exécution de CGI.
* Supporter les clients :
  + HTTP/HTTPS : Netscape (>= 4.x), Internet Explorer (>= 4.x), Firefox (>= 1.5.\*) et Opera (>= 7)
  + HTTP: Lynx (>= 2.8.\*) et Telnet

**1). Recommandations :**

Dans le même esprit, certaines recommandations nous on été faites, notamment :

* L’utilisation des threads (pour les environnements Unix et Win32 seulement).
* L’utilisation de la « Standard Template Library »
* La création de test unitaire.

Moyen et organisation

Voici les différents moyen de mis en œuvre ainsi que l’organisation des taches au sein du groupe concernant le développement du projet.

**1). Moyens :**

* Station de travail :
  + Environnement Windows (architecture i386)
  + Environnement Linux (architecture i386)
  + Environnement FreeBSD/NetBSD (architecture i386)
* Outils d’édition:
  + Emacs
  + Microsoft Visual Studio 2008
* Outil de compilation et débogage:
  + Linux/FreeBSD :
    - g++ (GCC) version 3.4.2
    - GNU gdb version 6.6
  + Windows :
    - Visual c++
    - Débogueur Windows
* Système de gestion de versions :
  + Linux/FreeBSD :
    - Subversion (svn) version 1.4.6
  + Windows :
    - TortoiseSVN version 1.5.0
* Bureautique :
  + Microsoft Word, Open Office, Adobe Reader, PowerPoint.
* Environment quotidien:
  + Parc informatique Epitech

En ce qui concerne la communication au sein du groupe et avec les utilisateurs, un site internet sera mis en place. Ce site a pour but de mettre à jour de tenir informe les utilisateurs de notre avancement, des nouveautés, …

**1). Organisation :**

* Organigramme général du groupe :

Bilan

1). Objectifs :

Au vue de notre première échéance, notre objectif était de créer un serveur basique qui supporte les méthodes GET, HEAD, POST, les exécutions CGI, le parcours de répertoire, …

1). Résultats obtenus :

Au vue de notre première échéance, notre objectif était de créer un serveur basique qui supporte les méthodes GET, HEAD, POST, les exécutions CGI, le parcours de répertoire, …

1).Evaluation :

Au vue de notre première échéance, notre objectif était de créer un serveur basique qui supporte les méthodes GET, HEAD, POST, les exécutions CGI, le parcours de répertoire, …

Team Zia.