

TD Internet et Réseaux partie II

Web, protocole HTTP sur des sockets

Ce TD a pour but de vous faire développer votre propre serveur Web que vous pourrez tester avec un simple telnet localhost 8080 puis via un navigateur Web standard. Vous pourrez vous aider pour cela des informations du cours et/ou celles de :

<http://www.commentcamarche.net/internet/http.php3>

1 Rappel sur les sockets

Vous trouverez les principales fonctions de l'API/C# Socket sur <http://profdinfo.com/web/420-KHH-LG/sockets.html>.

Vous trouverez une série d'exemples détaillés sur <http://www.java2s.com/Code/CSharpAPI/System.Net.Sockets/SocketConnectIPEndPointip.htm>.

Les tests se feront grâce la commande telnet que vous installerez si elle n'est pas encore présente sur votre machine.

Vous trouverez toutes les explications pour cela sur le site du cours sur www.tigli.fr.

2 Description de « l'architecture » d'un serveur Web

Pour vous aider voici quelques informations et questions :

1. Un serveur Web est un serveur socket mode connecté (TCP/IP). Vous pourrez donc utiliser au choix Java ou C/C++ pour vos développements.
2. Le port standard pour un serveur WEB est le port 80. Pourquoi devons nous utiliser ici le port 8080 ?
3. Après avoir crée la connexion TCP/IP un serveur dialogue avec un client en utilisant le protocole HTTP (Cf. cours et/ou RFC 1945).

3 Tests avant de commencer

Commencer par tester à l'aide de la commande telnet la mise en place d'un dialogue avec un serveur existant. Vous pourrez par exemple tester la commande suivante (au cas où, installez telnet sur votre machine) :

```
telnet www.unice.fr 80
GET / HTTP/1.0
```

, et demandez la page :

```
http://www.google.com
```

Quel est le résultat obtenu ? Quel est le serveur Web utilisé ?

De même installez Google Chrome et utilisé "les outils de développement" du browser Web.

TD Internet et Réseaux partie II

Web, protocole HTTP sur des sockets

Combien de requêtes HTTP sont déclenchées pour récupérer l'intégralité de la page ?

Mesurez et comparez les durées de récupération de la page www.google.com avec ou sans utilisation du cache local. Quel est le % de gain de temps ? de bande passante ?

4 Mise en œuvre d'un serveur HTTP basique

Commencer par créer un serveur en C# en vous inspirant des codes de la première section de ce document. Votre serveur doit prendre en compte une partie simple du protocole HTTP du type :

- *GET* <chemin relatif d'un fichier HTML par rapport à la racine du site Web>
- Exemple *GET /index.html http/1.0*, correspond pour le serveur à l'envoi du fichier */www/pub/index.html*

Voici un extrait d'un exemple de fichier index.html pour vos tests :

```
<TITLE> L'exemple HTML le plus simple</TITLE>
<H1> Ceci est un sous-titre de niveau 1</H1>
Bienvenue dans le monde HTML. Ceci est un paragraphe. <P>
Et ceci en est un second. <P>
<A HREF="index.html">cliquez ici</A> pour réafficher
```

Vous veillerez à déclarer une variable HTTP_ROOT qui permette de spécifier le chemin d'accès sur votre machine à l'espace disque qui sera servi par votre serveur Web (répertoire racine des documents accessibles par le serveur).

5 Avancé : Création de pages Web dynamiques

Pour le moment les pages Web récupérées sont dites statiques. En effet les pages renvoyées au client sont celles présentes et occasionnellement modifiées par leur propriétaire.

Lors de la consultation d'une page Web statique, un serveur HTTP renvoie donc le contenu du fichier où la page est enregistrée. Lors de la consultation d'une page Web dynamique, un serveur HTTP transmet la requête au logiciel correspondant à la requête, et le logiciel se charge de générer et envoyer le contenu de la page.

De nombreuses solutions logicielles sont développées pour faciliter et améliorer la génération de pages Web dynamiques. Citons par exemple les langages PHP, JavaServer Pages (JSP) ou Active Server Pages (ASP)...

Pour bien comprendre que ce principe ne dépend pas d'une technologie donnée, nous allons mettre en œuvre une des toutes premières approches pour la création de page web dynamique : les cgi-bin.

Nous allons maintenant ajouter l'appel à un cgi-bin (soit l'exécution autorisée d'un binaire qui pourra générer une page Web dynamique). Les paramètres sont alors passés après le "?", séparés par un "&", dans l'URL.

Exemple : pour l'exécution du programme « toto jean pascal » dont le fichier toto est situé dans \$HTTP_ROOT/cgi-bin, l'appel sera :

```
http://localhost:8080/cgi-bin/toto?jean&pascal
```

La page Web retournée sera alors :

TD Internet et Réseaux partie II

Web, protocole HTTP sur des sockets

```
<HTML>
<HEAD>
<TITLE>Doc. Produit par un CGI</TITLE>
</HEAD>
<BODY>
<H1>Coucou jean pascal !</H1>
</BODY>
</HTML>
```

Dans le cas d'un script shell cgi-bin, nombre de variable d'environnement du processus lancé sont positionnées.

Exemple : HTTP_USER_AGENT, SERVER_ADMIN, SERVER_SOFTWARE, ...

Les deux variables qui permettent la récupérations des paramètre de l'URL sont : REQUEST_URI et QUERY_STRING.

Exemple : QUERY_STRING=namex=valuex&namey=valuey&namez=aluez

6 Avancé : Les formulaires pour faciliter le passage de paramètres dans l'URL.

Jusque là les paramètre sont fournis à la main, dans le bon ordre, dans la syntaxe même de l'URL demandée.

Un formulaire HTML de positionner dans une page Web des zones interactives qui permettent à la fois la saisies des paramètres mais aussi la création de l'URL demnadée.

Exemple :

```
<form action="http://www.example.com/cgi-bin/example.sh" method="get">
Enter a username: <input type="text" name="username"></input><br>
<input type="radio" name="whatToDo" value="remove" checked="checked">Value<br>
<input type="radio" name="whatToDo" value="create">Create<br>
<input type="submit" name="subbtn" value="Submit">
</form>
```

Les différents types d'éléments existant pour une balise <input> sont : buttoncheckbox, file, hidden, image, password, radio, reset, submit, text.

Le type submit effectuera à la fois l'affectation de la valeur "Submit" au paramètre submit mais déclenchera aussi la requête sur l'URL créée et paramétrée.

Nous pouvons donc reprendre l'exemple précédent, en passant les paramètres "Jean" et "Pascal" grâce à un formulaire.

Une fois ce travail réalisé, comparez les performances du serveur de la section 3 et celle de la section 5 dans le cas de multiples requêtes simultanées.