

# Applications Réparties TD n°1 Serveur Web

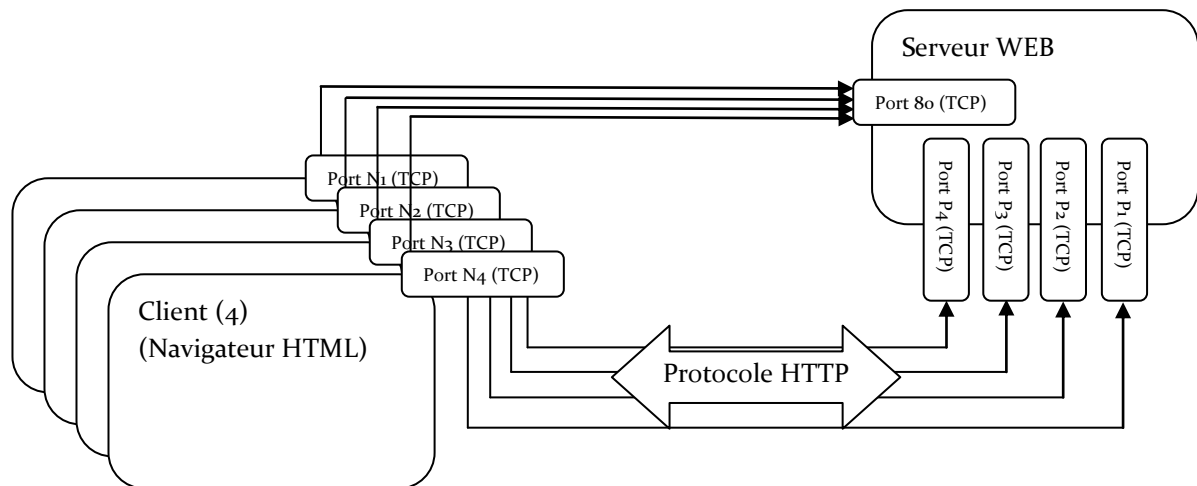
Ce TD a pour but de vous faire développer votre propre serveur Web que vous pourrez tester avec un simple telnet localhost 8080 puis via un navigateur Web standard. Vous pourrez vous aider pour cela des informations du cours et/ou celles de :

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

## 1 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).



## 2 Test 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 :

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

Quel est le résultat obtenu ?

## 3 Mise en œuvre d'un serveur HTTP basique

Commencer par créer un serveur qui prenne en compte une requête HTTP simple 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
```

## Applications Réparties TD n°1

### Serveur Web

---

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).

#### 4 Gestion des pages dynamiques

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 "?" qui suit l'URL, séparés par un "&".

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

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

#### 5 Serveur Web Multithreads :

Les plus avancés pourront modifier le code du serveur développé pour la section 3 afin d'autoriser des requêtes simultanées de plusieurs clients. Pour cela il faudra créer un thread par demande de connexion.

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.