M1 - Outils de l'Internet (Année 2010/2011)

Le massacre de Fort Apache

L'objet de ce deuxième TD est de comprendre comment le serveur HTTP Apache httpd traite les requêtes.

Exercice 1. *Requête HTTP*

✎ Sans entrer trop dans les détails du serveur HTTP, expliquer les différentes opérations qu'il exécute lorsqu'il reçoit une requête de type GET. On pourra détailler ce qui se passe pour la requête suivante :

```
GET /test/toto.html HTTP/1.1
User-Agent:  Opera/9.02 (Macintosh; Intel Mac OS X; U; en)
Accept: text/html, image/jpeg, image/png, text/*, image/*, */*
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: iso-8859-1, utf-8;q=0.5, *;q=0.5
Accept-Language: en,fr
Authorization: Basic Z3V5YnJ1c2g6bW9ua2V5
Host: www.monsite.fr
Connection: Keep-Alive
```

Exercice 2. *Apache Inside – Cycle de traitement d'une requête*

Le traitement d'une requête en Apache s'effectue en plusieurs points, ici classés par ordre alphabétique.

Check Access	Check Auth	Check User Id
Child exit	Child initialization	Handlers
Header Parser	Logger	Post Read Request
Prerun Fixups	Translate Name	Type Checker

✎ Proposer une signification pour chacun de ces points, et dans quel ordre les exécuter. Expliquer ce qui se passe sur la requête ci-dessus.

Exercice 3. *Apache Hacked – Modules*

Un module est un programme qui se greffe sur l'une des phases du cycle de traitement d'une requête.
CGI : Le module CGI est un module qui permet l'exécution de programmes. Par exemple la requête `http://www.google.com/search?q=treasure` de google est typiquement un appel à un programme qui s'appelle (sans doute) `search` avec comme argument `q=treasure`.

1. Où se situe a priori un tel module dans le cycle de traitement ?

En pratique, le résultat d'un script CGI correspond à la réponse du serveur, donc doit commencer par le type de la réponse (content-type), puis son contenu

2. Écrire un script `toto` de sorte que `http://localhost/toto?blop` vous affiche une page XHTML contenant la liste des 10 premiers entiers.

Les paramètres sont transmis au script CGI par des variables d'environnement. Voilà les variables d'environnement que reçoit le script `toto` lorsqu'on appelle
`http://www.monsite.fr/toto?blop`
(on négligera dans ce TD les problèmes évidents de codages sous-jacents).

```
SERVER_SOFTWARE = Apache/2.0.54 (Fedora)
SERVER_NAME = www.monsite.fr
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.1
SERVER_PORT = 80
REQUEST_METHOD = GET
HTTP_ACCEPT = 'text/html, image/jpeg, image/png, text/*, image/*, */*'
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/toto
QUERY_STRING = blop
REMOTE_HOST =
REMOTE_ADDR = 10.50.13.99
REMOTE_USER =
CONTENT_TYPE =
CONTENT_LENGTH =
```

Remarque : en python la variable `environ` du module `os` est un dictionnaire contenant la valeur de toutes les variables d'environnement.

3. Écrire un script `fact` de sorte que `http://localhost/fact?n` vous affiche une page XHTML contenant la factorielle de n.

Exercice 4. *Autres modules*

1. Expliquer où il faut greffer, d'après vous, les modules suivants dans le cycle de traitement :
- Un module qui permet de faire du cache ;
- Un module qui n'autorise l'accès au fichier `toto` que depuis l'ordinateur `toto.cmi.univ-mrs.fr` ;
- Un module qui permet d'exécuter du php ;
- Un module qui transforme les URI commencant par `/toto` en `/old/toto/`.