NE302 – Projet Réseau
Projet « HTTP Server » - part 4

1 Objectifs de la séance 2 – Projet HTTP

Cette séance 2 va permettre :

- d'expérimenter la configuration du serveur PHP en mode fastCGI, de prendre aussi des captures réseaux servant de références pour la compréhension des spécifications et la réalisation de la partie 4.

Pour les captures réseaux utilisez wireshark et sauvegardez les packet au format pcap.
N'hésitez pas à poser des questions pendant la séance….

2 Installation d'un serveur web « lourd »

Installez le serveur web avec le package apache2 avec les commandes suivantes :

```
#apt-get update
#apt-get install apache2
```

Vérifiez que le serveur écoute sur le port 80, avec la commande

```
# netstat -antp
```

3 Installation de PHP en mode fastCGI

Installez le serveurPHP7 en mode fpm (FastCGI Process Manager) :

```
#apt-get install php-fpm
```

Modifiez la ligne 36 du fichier /etc/php/7.x/fpm/pool.d/www.conf

```
listen = 9000
```

Relancez le processus, et vérifiez qu'il ecoute sur le port 9000

```
# systemctl restart php7.3-fpm
#netstat -antp
```

Le processus PHP ecoute sur le port 9000 les requêtes que devra relayer le serveur web pour les scripts CGI. Quand le serveur relaye les requêtes, il reprend les informations émises par le client HTTP dans les requêtes (POST ou GET), et les remet en forme en suivant les spécification FastCGI pour les envoyer via la connexion TCP sur le port 9000.

Vous allez capturer ces échanges sur le port 9000. Mais avant cela il faut configuer le serveur apache pour rediriger les requêtes PHP. Ceci necessite un module spécifique…

```
# a2enmod proxy_fcgi
# systemctl restart apache2
```

Configurez apache en ajoutant dans le fichier /etc/apache2/sites-enabled/000-default (dans la section virtualhost), les lignes suivantes :

```
<FilesMatch "\.php$">
        SetHandler "proxy:fcgi://127.0.0.1:9000/"
</FilesMatch>
```

Créez le script /var/www/html/info.php, contenant:

```
<?php phpinfo(); ?>
```

Bon tout ça pour ça…. Maintenant appelez http://127.0.0.1/info.php depuis un navigateur et prenez les captures réseaux sur le port 9000 pour observer les formats de messages fastCGI….

Pour cela allez dans wireshark, et sur un paquet sur le potrt TCP 9000 appliquer :

decode As… et choisissez FCGI.

Ce lien peut vous être utile.

https://wiki.wireshark.org/FastCGI

4 Les spécification fastCGI

A l'aide des captures réseau essayer de bien comprendre les spécifications fastCGI, L'interface fast-CGI est spécifiée sur le site www.fastcgi.com (semble actuellement non joignable) –> essayer http://www.mit.edu/~yandros/doc/specs/fcgi-spec.html

Vous pourrez aussi vous aider du fichier fastcgi.h (sur chamilo)

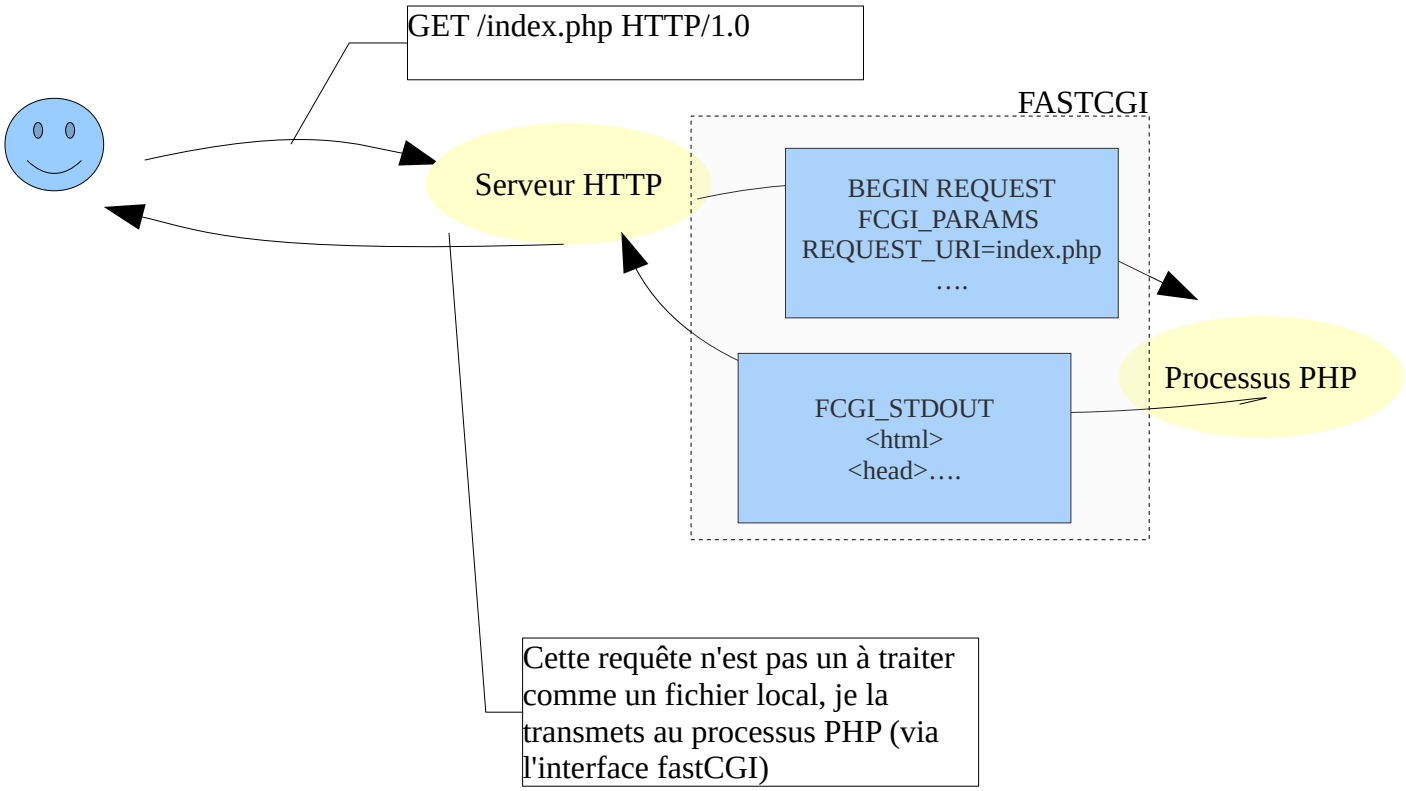
5 Détail du travail demandé

Votre serveur sait maintenant gérer des requêtes pour des fichiers locaux, mais des requêtes pour /index.php ne sont pas encore gérées.

Votre serveur devra pouvoir détecter que ces requêtes doivent être traitées par le processus php, et non pas par le serveur web, cela vous ouvrira des perspectives avancées comme la gestion des sessions utilisateurs, l'accès à une base de données, etc. mais avant d'en arriver là vous devez gérer l'interface fastCGI.

Cette fois-ci seule les primitives de connexions TCP sont déjà écrite, et vous devrez gérer toute la communication entre le serveur web et le processus PHP (gestion des buffers d'emission et reception, etc..)

Schéma du fonctionnement de cette partie :



6 Utilisation de l'API des sockets

La création de la communication se fait via la fonction suivante, les lectures se font avec l'appel à la fonction read (vous pourrez utiliser uen version bloquante), l'écriture via l'appel à la fonction write sur le descripteur retourné par la fonction suivante. (Cf manpages)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
```

```
static int createSocket(char *ip,int port)
{
	int fd;
	struct sockaddr_in serv_addr;
	int enable = 1;

	if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
		perror("socket creation failed\n");
		return (-1);
	}

	bzero(&serv_addr, sizeof(serv_addr));

	serv_addr.sin_family = AF_INET;
	inet_aton(ip,(struct sockaddr *)&serv_addr.sin_addr.s_addr);
	serv_addr.sin_port = htons(port);

	if (connect(fd, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0) {
		perror("connect failed\n");
		return (-1);
	}
	return fd;
}
```

7 Tests

Vous développerez quelques pages php (pas plus de 2 ou 3), afin de montrer le carctère dynamique du site. La démonstration de la fin d'étape 4 se fera avec votre site, l'évaluation pourra se faire avec d'autres pages php.