TP N°1 : Compilation et mise en œuvre d'un serveur web à partir des sources. Programmation cgi

Le but de ce tp est de compiler une application à partir de ces sources et d'écrire un programme cgi en script bash puis en C.

1/ Compilation et installation du serveur web boa.

Vérifier qu'aucun serveur web ne tourne actuellement sur votre machine (sur le port 80).
Récupérer les sources de la dernière version de boa (boa-0.94.14rc21) (par wget).
Générer boa : ./configure, make... make instal :-)
Installer boa (lire la documentation et voir boa.conf).
Créer un fichier index.html simple et vérifier que vous y avez accès.
Utilisez les commandes ps , grep et kill pour arrêter boa.
Créez un script de démarrage de boa et placez-le dans /etc/init.d puis faire update-rc.d
Vérifier que boa démarre au lancement de la machine.
Expliquer succinctement le mécanisme de démarrage debian :
pistes : man init , man update-rc.d et cherchez sur internet "runlevel debian"
examiner le contenu des répertoires /etc/init.d et /etc/rcX.d

2/ CGI en shell bash
Modifiez boa.conf pour que les cgi puissent s'exécuter depuis un répertoire que vous choisirez.
Créez le fichier (cgi.sh) ci-dessous et donnez lui les droits d'exécution :

```
#! /bin/sh
echo "Content-type: text/plain\n\n"
echo cgi en shell bash
```

- Exécutez ce fichier depuis le navigateur : localhost/cgi-bin/cgi.sh
- Modifiez ce script pour afficher l'occupation du disque (utiliser df -H, head et tail) et le nombre de processus en cours d'exécution (utiliser ps et wc).

3/CGI en C
Le langage C n'est pas le plus indiqué pour générer des pages html. Cependant lorsqu'on se contente de pages simples et que l'on doit accéder directement au matériel, il se révèle tout à fait utilisable.

```
#include<stdio.h>
main()
{
    printf("Content-type: text/html\n\n");
    printf("<html>\n");
    printf("<head><title>Mon premier script CGI</title></head>\n");
    printf("<body bgcolor=\"#FFFFFF\">\n");
    printf("<br><br><br><br>\n");
    printf("<center>\n");
    printf("<h1>Salut à tous<br>voici mon premier script CGI</h1>\n");
    printf("</center>\n");
    printf("</body>\n");
    printf("</html>\n");
}
```

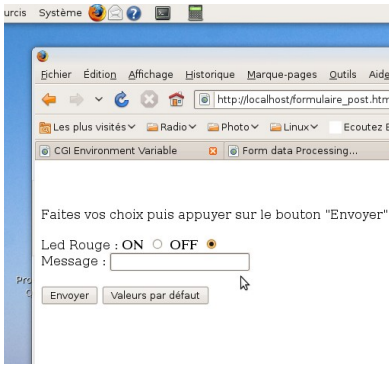
- En utilisant la fonction getenv() , renvoyez au client la valeur de quelques variables d'environnement (REMOTE_ADDR,SEVER_SOFTWARE,etc...)
- Vérifiez que vous pouvez transmettre des données en utilisant une url étendue (localhost/cgi-bin/moncgi?bonjour+monsieur). Quelle est la méthode http utilisée dans ce

cas ?

- Renvoyez au client les données réçues (REQUEST_METHOD, QUERY_STRING)
- Expliquer succinctement le principe de fonctionnement de l'interface cgi.

4/ Création d'un formulaire html et exploitation des données coté serveur.

Créez une page html contenant un formulaire comportant deux boutons radio, une zone de saisie de texte et deux boutons permettant de remettre les valeurs d'origines et d'envoyer les données. Utilisez la méthode POST.



Créer un programme cgi qui exploitant les données formulaire.

Il peut être judicieux de créer des fonctions de bases qui seront réutilisées par la suite :

html_begin : Envoie les en-têtes http et html. On peut spécifier un titre et une couleur de fond en paramètre.

html_end : termine le fichier html

send_error : permet de formater un message d'erreur

send_error_and_quit : send_error et fin du programme

read_post_data : alloue la quantité de mémoire nécessaire pour stocker les données reçues et forme une chaîne avec les données reçues.

Retrive_Value_Of : Permet de récupérer, dans les données reçues, la valeur d'une paire nom=valeur en lui fournissant le nom. Retourne 0 si trouvé et -1 sinon

Exemple :

```
char Value[50];
if(Retrive_Value_Of("OnOff",Value,Data) ) {
        send_error_and_quit("OnOff not found");
}
printf("OnOff=%s\n",Value);
```