

Sioux

Serveur web pour téléphone mobile

Par **Vial Loïc**

I - Description

Sioux est un serveur web pouvant traiter des requêtes HTTP/1.0 standard de la forme "GET <fichier>", où <fichier> représente le chemin absolu depuis la racine du serveur vers un fichier présent sur le serveur.
Le serveur va répondre aux requêtes en envoyant au client le fichier demandé, dans le respect de la norme HTTP/1.0.

Sioux s'accompagne d'une gestion en parallèle d'un programme de GPS qui met régulièrement à jour les coordonnées du serveur, et qui les affiche dans un fichier "/position.html".

II - Compilation

- "make all" : va générer le serveur "sioux", ainsi que deux programmes "sioux_gps" et "sioux_gps_google" pour la gestion des coordonnées GPS.
- "make clean" : supprime les fichiers .o générés par gcc.

III - Demarrage

- "./sioux" : lance le serveur web Sioux

IV - Configuration

Le serveur Sioux est paramétrisable depuis son fichier de configuration "sioux.conf".
Le fichier de configuration doit respecter le squelette suivant :

```
ip_address   ____ . ____ . ____ . ____
port         ____
web_root_dir  _____
log_file     _____
gps_software  _____
frequency    ____
```

- **ip_address**: l'adresse IP du serveur, généralement 127.0.0.1 en local, ou bien l'adresse LAN pour une utilisation externe.
- **port** : le port d'écoute du serveur, par défaut 2222.
- **web_root_dir** : le repertoire racine du serveur, généralement là où se trouvent les fichiers "index.html" et "position.html", par défaut ".".
- **log_file** : le chemin vers le fichier de log du serveur, par défaut "./sioux.log".
- **gps_software** : le chemin vers le programme executable chargé de générer le fichier "position.html" qui contient les coordonnées GPS du serveur.
- **frequency** : la fréquence en secondes à laquelle on appelle le programme GPS, par défaut 3.

Exemple de fichier de configuration (fichier par défaut) :

```
ip_address    127.0.0.1
port          2222
web_root_dir  .
log_file      ./sioux.log
gps_software  ./sioux_gps_google
frequency     3
```

Attention : le fichier de configuration "sioux.conf" doit être accessible dans le répertoire de lancement du programme.

V - Test

- pour tester le serveur, utilisez un client web standard (Firefox, Chrome, etc.) et connectez vous sur "**http://ip_address:port**", avec l'adresse IP et le port décrits dans le fichier de configuration.

VI - Notes

1) A propos du serveur HTTP

- il traite la racine "/" comme si c'était le fichier "/index.html", comme un serveur web habituel.
- il renvoie une erreur "404 Not Found" avec comme message "Not Found" dans le cas où il ne trouve pas un fichier demandé.
- il renvoie une erreur "400 Bad Request" avec comme message "Bad Request" dans le cas où la requête client est mal formée.
- il peut traiter plusieurs clients à la fois grâce à l'utilisation de threads.

2) A propos des programmes GPS fournis

- sioux_gps genere une page HTML simple avec les deux coordonnées GPS (latitude et longitude).
- sioux_gps_google genere la même page HTML, avec en plus une carte Google montrant graphiquement le point sur la carte du monde.
- les deux programmes generent aleatoirement des coordonnées GPS, de n'importe où sur le globe, ce ne sont donc pas les vraies coordonnées GPS du serveur qui sont affichées.

3) A propos du code

- on utilise la librairie POSIX threads (pthreads) pour la gestion des threads, il faut donc compiler avec l'option -lpthread pour utiliser la librairie.
- les commentaires explicatifs à propos des signatures des fonctions du fichier "function.c" sont dans l'entête "function.h".

4) A propos de l'utilisation du programme

- le programme envoie un message en console à chaque fois qu'on écrit dans le fichier de log, et à chaque fois que le programme GPS est lancé.
- si vous ne voulez pas être ennuyé par ces messages, lancez `"/sioux > /dev/null"` au lieu de `"/sioux"`.
- chaque démarrage du serveur, connexion d'un client, requête d'un client et fermeture du serveur est enregistré avec la date et l'heure dans le fichier de log.

5) Point faible du serveur

- lors de l'arrêt du programme avec les signaux 2 (SIGINT) ou 15 (SIGTERM), le serveur attend une dernière requête client avant de se fermer.
- si on veut arrêter immédiatement le programme, on peut toujours envoyer un signal 9 (SIGKILL).