ESISAR

NE449 - Programmation répartie

TDM numéro 10

Table des matières

1 Objectifs

Le TDM10 va permettre la découverte du protocole HTTP.

2 Exercice 1 : Découverte HTTP avec netcat

Lisez la documentation de l'outil netcat (dans un terminal, faire man netcat).

Le protocole HTTP fonctionne au dessus du protocole TCP, il permet de manipuler des ressources distantes. Il est principalement utilisé pour le web. Le client envoie des requêtes avec le format suivant :

```
Ligne de commande (Commande, URL, Version de protocole)
En-tête de requêtes
[Ligne vide]
Corps de requête
```

La ligne de commande indique ce que le client attend. Par exemple, pour obtenir la page d'accueil, on utilisera la commande :

```
GET /index.html HTTP/1.0
```

L'entête de requête contient une succession de couples (variable : valeur) , par exemple `User-Agent: Mozilla`. Elle permet au client de donner des informations à son propos ou de

préciser la requête.

La ligne vide est très importante elle distingue le header HTTP du reste du message.
Le corps de la requête peut contenir des données additionnelles (dans le cas de messages POST par exemple)

Le serveur répond avec un format similaire :

```
Ligne de statut (Version, Code réponse, Texte réponse)
En-tête de réponse
[Ligne vide]
Corps de réponse
```

La ligne de status indique au client si sa requête a réussi. Parmi les réponses possibles:

- HTTP/1.0 200 OK indique le succès de l'opération
- HTTP/1.0 404 Not Found, indique que la page demandé n'existe pas.
- HTTP/1.0 418 I'm a teapot, indique que le serveur n'est pas un serveur HTTP mais une théière (RFC 2324).

L'entête permet au serveur de préciser la réponse par exemple en indiquant la taille et le type de la ressource demandée.
La ligne vide sépare l'entête de la ressource demandée dans le corps du message.

Naviguez sur le serveur de la salle avec un navigateur classique. L'URL est

```
http://192.168.130.202/index.html
```

Faites ensuite la même navigation avec l'utilitaire netcat, en tapant directement à la main les commandes HTTP (vous resterez avec le protocole HTTP/1.0).

Prenez le temps de bien observer le contenu des échanges, et de comprendre comment fonctionne votre navigateur Web : combien de requêtes HTTP doit il faire pour afficher une page, ...?

3 Exercice 2 : Réaliser son premier client HTTP, ou savoir prendre le bus ...

Vous êtes étudiants ou apprentis, vous habitez dans le centre ville de Valence et vous venez tous les jours à l'ESISAR en bus. Vous prenez le bus au pôle bus à Valence et vous descendez à l'arrêt Briffaut en face de l'ESISAR.

En examinant le site http://citea.info/, vous remarquerez qu'il y a plusieurs bus pour faire ce trajet:

- le bus Citéa4 (toutes les 10 minutes)

- le bus 20 (toutes les 30 minutes)
- le bus 13 (toutes les 20 minutes)

Par ailleurs, en consultant la version mobile du site (www.citea.info/mobile), vous constatez qu'il est possible de visualiser **les horaires réels** de passage des bus.

Voir par exemple :

http://www.citea.info/mobile/horaires/index.asp?rub_code=23&lign_id=52&sens=1&part_id=&network_id=&typeSearch=line&pa_id=&stopPoint=10314$POLE%20BUS%20Quai%20B$0

En fait, tous les bus sont équipés en GPS et remontent leur position en temps réel au central informatique de Citéa. L'information est alors aussi publiée en temps réel sur ce site, par la page que vous venez de voir.

Vous détestez particulièrement attendre le bus dans le froid, vous vous décidez alors à développer une application informatique pour résoudre ce problème. Votre objectif est de partir de chez vous en sachant exactement quand le bus va passer, et quel est son numéro et quel est son quai.

Cette application informatique affichera la liste des bus qui vont passer au pôle bus et qui pourront vous emmener à l'ESISAR. Pour chaque bus, vous indiquerez son numéro de ligne (Cité4 ou 13 ou 20) , le quai de départ (A ou B), et dans combien de temps il sera à quai.

Vous n'affichez que **les bus dont l'horaire est réel** (vous n'affichez pas les horaires théoriques). Bien sûr, la liste des bus est triée par heure de départ (le premier qui part en haut).

Exemple de présentation :

```
Cité4       Quai B              2 min
20          Quai A              3 min
13          Quai X              5 min
Cité4       Quai B             12 min
```

Votre écran se rafraîchira toutes les 10 secondes en automatique.

Vous trouverez ci dessous un exemple de code permettant de télécharger une page WEB :

```
import java.io.IOException;
import java.io.InputStream;
import java.net.InetSocketAddress;
import java.net.Proxy;
import java.net.URL;
import java.net.URLConnection;

public class Bus
{

	public static void main(String[] args) throws IOException
	{
		Bus bus = new Bus();
		System.out.println("Debut de connexion ");
		bus.doIt();
	}

	private void doIt() throws IOException
	{
		URL url = new URL("http://www.citea.info/mobile/horaires/index.asp?
rub_code=23&typeSearch=stop&stopPoint=229%24POLE+BUS
%242%2426362%24&lign_id=52&pa_id=10314&sens=1");

		URLConnection connection = url.openConnection();

		connection.connect();

		InputStream is = connection.getInputStream();

		// A vous de jouer : lire dans is pour récupérer le contenu de la page WEB

	}

}
```