

TP Java Réseau

Fabien Coelho

1 Thread Pompe

Vous disposez d'une *thread* `Pompe.java` qui recopie tout ce qui vient d'un `InputStream` vers un `OutputStream`. Les deux flux lui sont fournis de l'extérieur. Cette pompe vous sera utile dans les prochains exercices.

2 Serveur Echo

À partir du squelette `Echo.java`, réaliser un serveur qui retourne au client tout ce qu'il reçoit. Testez ce serveur avec la commande `netcat`.

3 Serveur HTTP minimal

À partir du squelette `Httpd.java`, réaliser un serveur web minimal qui sert les fichiers contenus dans un répertoire fixe donné en arguments du programme lors de son lancement.

un navigateur

```
GET /chemin/vers/fichier.html HTTP/1.0
```

et vous envoyez en réponse le fichier pris dans un répertoire donné fixe :

```
/repertoire/de/base/chemin/vers/fichier.html
```

La réponse envoyée est alors de la forme :

```
HTTP/1.0 200 OK
```

```
Content-type: text/html
```

le contenu du fichier est afficher là...

Il est obligatoire de fournir le type MIME du fichier, par exemple `text/html` `text/plain` `image/gif` `image/jpeg` `image/png`.

Testez d'abord le serveur avec la commande `netcat`, puis avec un navigateur. Que ce passe-t-il si vous mettez `..` dans un nom de répertoire ?

4 Client HttpSurvey

À partir du squelette `HttpSurvey.java`, réalisez un client HTTP qui récupère les entête `Server:` en faisant une requête HEAD sur des serveurs web dont les noms sont lus sur l'entrée standard.

```
sh> java ServerSurvey < liste_de_serveur.txt
www.ensmp.fr: Apache
www.microsoft.com: IIS/6.0
...
```

Que ce passe-t-il si le serveur n'existe pas ? Si le serveur ne répond pas ? Améliorer la situation.