TD6 : serveur HTTP

1 Découverte du protocole HTTP

Avant de commencer, lisez attentivement la page wikipedia consacrée au protocole HTTP : `http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol`. Notez que le paragraphe sur la version 1.0 du protocole donne un exemple de requête HTTP avec la réponse du serveur.

Les 3 exercices de cette partie sont à réaliser avec le logiciel `telnet`.

Exercice 1 : Méthode `HEAD`

Quels sont les entêtes de la page de garde de www.labri.fr, www.inria.fr, www.essi.fr ?

Exercice 2 : Classes de réponse - En utilisant la méthode `HEAD`, essayez d'obtenir les différentes classes de réponse.

1. Succès. `/* facile */`
2. Erreur client. `/* vraiment facile */`
3. Inchangé. `/* plus dur */`

Exercice 3 : Méthode `GET` simple et entêtes

Reprenez la question précédente, pour obtenir le contenu du document des pages `index.php`. Obtenez vous la même chose que sous Firefox ?

2 Programmation d'un serveur web de pages statiques

L'archive `www.zip` contient un mini site web. Extrayez cette archive et parcourez ce site avec `firefox`, sans utiliser le protocole HTTP.

Les liens de la section "Gazette" ne fonctionnent pas car les fichiers sont (volontairement) manquants. Expliquez ce qui se passe lorsque vous cliquez sur le lien "Votre configuration".

L'objectif de cette séance est de réaliser un mini serveur web en java, permettant de parcourir ce site : vous pourrez donc utiliser un navigateur sur la machine de votre voisin pour tester le bon fonctionnement de votre serveur.

Cette partie est à réaliser par binôme : vous devez remettre votre code par mail à la fin de la séance. Le langage de programmation doit être `Java` (préconisé) ou `C++`. Il n'est pas possible de tout faire en 2h : faites simplement le maximum ...

2.1 Démonstration

L'archive `demo.zip` contient une version de démonstration. Pour l'exécuter :

1. `java HTTPD` (cela lance le serveur sur le port 1234)
2. ouvrez votre navigateur à l'URL `http://localhost:1234/index.xhtml`

Le contrôle d'accès implémenté porte sur l'ensemble du site (pas seulement l'intranet). Cette version contient également les rudiments d'un serveur PHP. Les entêtes HTTP retournés par le serveur sont stockés dans des fichiers du répertoire `reponse`.

Deux fenêtres permettent de suivre le fonctionnement du serveur :
– "Mon serveur web : dialogue" : cette fenêtre retrace l'ensemble des dialogues HTTP ; "Mon serveur web : identification du client" : cette fenêtre donne des informations sur les adresses IP des clients, les pages demandées, et la présence éventuelle de code PHP dans la page à servir.

2.2 Code de départ

Nous vous fournissons un code de départ sous la forme de la classe `MonServeurWeb`. Cette classe écoute sur le port 1234, récupère une requête `GET` et récupère la valeur des paramètres `nom` et `prenom`. Elle retourne ensuite une réponse sous la fome d'une page HTML.

Pour tester, utiliser l'URL : `http://localhost:1234/index.html?nom=bob&prenom=edouard`

```
import java.net.*;
import java.io.*;
import java.util.*;

public class MonServeurWeb{

public static void main( String[] args ) throws Exception{
        ServerSocket ss = new ServerSocket(1234);
        System.out.println("Mon serveur web en ecoute");
        Socket s = ss.accept();
        System.out.println("******* Connexion etablie
           ************");
        InputStream is = s.getInputStream();
        LineNumberReader lnr = new LineNumberReader( new
           InputStreamReader( is ));
        // lecture ligne par ligne de la requete
        String ligne = " ";
        // analyse premiere ligne
        ligne = lnr.readLine();
        String nom=""; String prenom="";
        if (ligne.toUpperCase().indexOf("GET")!=-1){ // ligne
           contient GET
             System.out.println("Serveur: methode GET
                utilisee");
             ligne = ligne.substring(ligne.indexOf("?"));
                // on enleve le debut
```

```
                ligne = ligne.substring(0,ligne.indexOf("HTTP"
                    )); // on enleve la fin
                StringTokenizer st = new StringTokenizer(ligne
                    , " ?&"); // on decoupe variable par
                    variable
                while (st.hasMoreElements()){
                        String variable = st.nextElement().
                            toString(); // recuperation d'une
                            variable
                        System.out.println("serveur: "+
                            variable);
                        if (variable.startsWith("nom=")) nom=
                            variable.substring(4); // valeur de
                            nom
                        if (variable.startsWith("prenom="))
                            prenom=variable.substring(7); //
                            valeur de prenom
                }
        }

        // autres lignes
        while (ligne.compareTo("")!=0){
                ligne = lnr.readLine();
                System.out.println("Client: "+ligne);
        }
        System.out.println("envoi de la reponse");
        // reponse
        String page = "<HTML><BODY> <H1> Nom saisi: "+nom+" </
            H1><H2> Prenom saisi: "+prenom+"</H2> </BODY></HTML>
            ";
        String entete = "HTTP/1.1 200 voici la page!!\nDate:
            Tue, 25 Oct 2005 19:57:48 GMT\nServer: MonServeurWeb
            \nAccept-Ranges: bytes\nContent-Length: "+page.
            length()+"\nContent-Type: text/html\n\n";

        OutputStream os =s.getOutputStream();
        PrintWriter pw = new PrintWriter(os);
        pw.println(entete);
        pw.println(); // ligne blanche
        pw.println(page);

        pw.flush();
        s.close();
}

}
```

2.3 Fonctionnalités du serveur à implémenter

Il faut que le serveur Web soit capable d'héberger un site simple. Les fonctionnalités requises sont donc les suivantes :

Interprétation des requêtes HTTP 1.0

Le serveur doit être capable de comprendre et répondre aux requêtes HEAD et GET. Pour les spécifications, consulter la RFC 1945 disponible sur :
`http://www.ietf.org/rfc/rfc1945.txt`

Envoi de pages statiques

Le serveur doit être capable d'envoyer des pages Web, des feuilles de styles, des images, etc ...Plus précisément, pour des requêtes de type GET, il faut que le navigateur reçoive, en plus de l'entête HTTP, le document demandé.

Un peu d'aide ?

```
private byte[] getReponse( Socket s, String requete ) throws
   IOException{
        String page = ""; byte[] image; String entete ="";
        // test autorisation
        if (!autorisation(s)){ // si echec autorisation, on
           retourne l'erreur 403
                page = getPage( new File("./pages/403.html"));
                entete = getEntete( 403, page.length(), "text/
                   html" );
                return (entete+page).getBytes("UTF8");
        }
        if (requete.startsWith("GET")){
                // on retourne la page
                StringTokenizer st = new StringTokenizer(
                   requete, " ");
                String mot = st.nextToken(); // on saute GET
                mot = st.nextToken(); // on recupere le nom du
                    fichier demande
                pageDemandee = mot;
                if (p!=null) p.append(IPresentation.client, "
                   Serveur : demande du fichier "+mot+" par le
                   client\n");
                File f = new File("./www"+mot);
                if (f.exists()&&(!f.isDirectory())){
                        entete="";
                        if (mot.endsWith(".png"))
                                {
                                if (p!=null) p.append(
                                   IPresentation.client, "
```

```
                                        Chargement de l\'image "+mot
                                        +"\n");
                                    image = getImage(new File("./
                                        www"+mot));
                                entete = getEntete( 200, image.length
                                    , "image/png");
                                return concat(entete.getBytes("UTF8")
                                    , image);
                                    }
                                page = getPage(new File("./www"+mot));
                                if (mot.endsWith(".css"))
                                        entete = getEntete( 200, page.
                                            length(), "text/css");
                                if (entete.equals(""))
                                        entete = getEntete( 200, page.
                                            length(), "text/html" );
                                reponse = entete+page;
                                return (entete+page).getBytes("UTF8");
                        }
                        // page non trouvee
                        page = getPage(new File("./pages/404.html"));
                        entete = getEntete( 404, page.length(), "text.
                            html" );
                        reponse = entete+page;
                        return (entete+page).getBytes("UTF8");
                }
                reponse="";
                return ("").getBytes("UTF8");
        }

        ...

        private String getEntete(int code, int taille, String type)
            throws IOException{
                // entete HTTP de la reponse
                String valeur = new Integer(taille).toString(); //
                    recuperation de la taille de la page
                String reponse = "";
                File entete = new File("./reponses/"+code+".http");
                InputStream is = new FileInputStream(entete);
                LineNumberReader lnr = new LineNumberReader( new
                    InputStreamReader(is) );
                String ligne = " ";
                while ((ligne=lnr.readLine())!=null)
                        // lecture ligne par ligne de du fichier
                        reponse+=ligne.replaceFirst("VALEUR",valeur).
                            replaceFirst("TYPE",type)+"\n";
```

```
        lnr.close();
        return reponse;
}

private String getPage( File html ) throws IOException{
        InputStream is = new FileInputStream(html);
        LineNumberReader lnr = new LineNumberReader( new
           InputStreamReader(is) );
        String ligne = " "; String page="";
        while ((ligne = lnr.readLine())!=null)
                // lecture ligne par ligne de du fichier
                page+=ligne+"\n";
        lnr.close();
        return page;
}
```

Contrôle d'accès

Un contrôle d'accès (dans le style htaccess) doit être implémenté. C'est à dire que si l'adresse IP du client ne fait pas partie des adresses autorisées dans le fichier .htaccess, le navigateur doit demander un nom et un mot de passe à l'utilisateur avant d'accéder aux répertoires protégés, afficher les pages correspondantes si le login/mot de passe est correct, et une page d'échec sinon. L'utilisation de htaccess est expliquée sur : `http://www.securiteinfo.com/conseils/htaccess.shtml`.