Prénom :
Nom :
Groupe TD :
Date :

Programmation répartie

M4102

François Merciol, Didier Lesage

DUT Informatique – 2nd année

Vannes – 2020/2021

Support de cours **étudiant**

http://m4102.merciol.fr/

(page gauche vide)

Table des matières

Table des figures

Liste des tableaux

Listings

1 TD 2 : Mandataire

Rappels :

- N'oubliez pas qu'il y a un contrôle de présence. Les absences sont remontées et gérés par le secrétariat. Les retards seront comptabilisés pour être intégrés dans la note de contrôle continue.
- Après l'appel, c'est l'occasion de poser les questions concernant des éléments non compris du cours.
- N'oubliez pas qu'un compte-rendu sera demandé en fin de dernier créneau de la semaine (donc vendredi).
- Ce travail est à faire en binôme (un seul compte-rendu par binôme) dans le cas de groupe TD (environt 28).
- Si le travail est réalisé en TP (environt 14 étudiants) vous aurez
  - un bonus à rendre seul
  - un malus si des éléments sont copier-coller d'une autre copie.

  En d'autres termes si votre travail est collectif, assumez le en rendant en groupe, mais vous êtes invité à travailler seul pour mieux vous investir.
  Dans les circonstances actuelles de travail à domicile, il ne sera peut-être pas facile d'établir des binômes. Nous savons que vous ferez de votre mieux et ferons preuve de bienveillance en tenant compte des conditions.
- Vous ne devez pas rendre un simple listing, mais un document répondant aux questions posées (qui peut contenir des parties de listing). Nous vous invitons à joindre les sources à l'archive de votre compte-rendu.
- Vous rappellerez vos prénoms et noms, groupe TD, date et identifierez clairement le TP.
- Vous recopierez le texte des questions (et numéro) à chaque fois (pensez au copier-coller depuis l'énoncer).
- Vous répondrez par des phrases.
- Vous recopierez les commandes que vous avez saisies pour obtenir un résultat et plus généralement ce que vous saisissez dans un terminal.
- Quand c'est nécessaire, vous pouvez faire une copie d'écran (pas l'écran entier qui serait illisible, mais uniquement la partie que illustrant votre propos).
- Les modalités pour rendre votre travail pourront être précisées par votre encadrant (mel...). Dans tous les cas, vous devez privilégier un lien de téléchargement qui vous sera indiqué plutôt qu'une pièce-jointe à un courriel.

Ce TD porte sur la réalisation d'un programme faisant communiquer des clients et des serveurs sur un réseau. Pour ce faire, nous utiliserons des sockets au-dessus d'IP. Le programme consistera en une communication permettant de tracer les échanges entre un client et un serveur.

De plus, il vous servira de boîte à outils pour l'analyse de protocole que vous aurez à mettre au point dans l'avenir.

1.1 Complément sur les sockets

Commençons par un complément de cours. La semaine dernière nous avons vu la mise en place d'un serveur de sockets en Java. Cette semaine nous examinons la connexion depuis un programme Java vers un serveur déjà existant. Voici un exemple de code qui crée un client de socket.

```
    Socket clientSocket =
  new Socket (args [0], Integer.parseInt (args [1]));
    PrintStream out =
  new PrintStream (clientSocket.getOutputStream (), true);
    out.println (args[2]);
    out.flush ();
    clientSocket.shutdownOutput ();
    BufferedReader in =
  new BufferedReader
  (new InputStreamReader (clientSocket.getInputStream ()));
    for (;;) {
  String line = in.readLine ();
  if (line == null)
      break;
  System.out.println (line);
  System.out.flush ();
  out.close ();
    }
```

Listing 1 – Code Java d'un client de socket

Comme nous connaissons maintenant les deux facettes de la communication avec sockets (client et serveur), nous allons réaliser un programmme qui met en œuvre les deux aspects synultanément.

1.2 Compréhension du sujet

Le logiciel résultant de ce TD possède les propriétés suivantes :

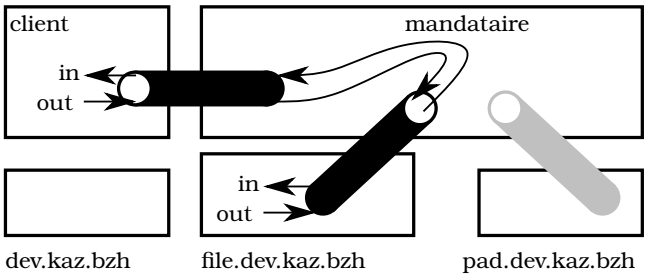
- un comportement de serveur avec un port de connexion paramétrable,
- un comportement de client vers d'autres services (machine et port paramétrables),
- une retransmission "full duplex", c'est-à-dire simultanée et dans les deux sens de transmission entre le client et le serveur,
- une indépendance du protocole retransmis,
- l'acceptation de plusieurs connexions simultanées,
- la possibilité de traces (montante ou descendante)
- la possibilité de modification à la volée (dans le sens descendant) d'une chaîne de caractères par une autre

Nous ajoutons une caractéristique supplémentaire :

- Il est possible de considérer la première ligne et de l'annalyser pour choisir le serveur que l'on va relier.

Ceci nous permet de réaliser un mandataire web ("Proxy" en anglais).

Le schéma général est le suivant :



Le client est par exemple un navigateur "firefox". Vous le configurez pour qu'il passe par un mandataire (proxy) comme indiqué dans l'énoncé du TP.

Lorsque vous chargez une page comme "dev.kaz.bzh" :

- Votre navigateur va se connecter avec un socket vers votre mandataire.
- Ce dernier lit le début du flut pour extraire la première ligne :

  `GET http://dev.kaz.bzh/xxx HTTP/1.1`
- Il va extraire le serveur visé "dev.kaz.bzh".
- Il va ouvrir à son tour un socket vers ce serveur et recopier la première ligne altérée de la façon suivante :

  `GET /xxx HTTP/1.1`
- Il va ensuite recopier les données qui provienne du client vers le serveur et les données qui viennent du serveur vers le client.

Plusieurs remarques :

- La copie des données du client vers le serveur ne doivent pas être bloquantes pour le reste du programme (donc une tâche (Thread) est nécessaire)
- La copie des données du serveur vers le client ne doivent pas être bloquantes pour le reste du programme (donc une tâche (Thread) est nécessaire)
- Les données copiées sont brutes et de n'importe quels formats : images, sons... (non textuelle ou binaire si vous préférez). Il ne sera pas possible d'utiliser les classes de la semaine dernière, mais les class "DataInputStream" et "DataOutputStream".
- À une entrée du client correspond une sortie vers un serveur.
- À une sortie du client correspond une entrée du serveur.
- Le mécanisme de copie est semblable dans un sens et dans l'autre (mutualisez le code).
- Si une source se tarit, elle doit fermer le puit (le consommateur) correspondant (l'autre socket)

Liaison non chiffrée

Les exemples que nous allons montrer sont en “HTTP” et non “HTTPS”. Il est important que les échanges soient en clair pour que nous puissions analyser le contenu. C’est pour cela que nous vous conseillons d’utiliser le site “dev.kaz.bzh”.

1.3 Analyse du logiciel

Question : Donnez un schéma d’utilisation d’un mandataire (par exemple web). Décrivez la communication entre le navigateur et le serveur au travers du mandataire. Expliquez comment notre mandataire vient s’insérer.

Réponse : ✍

Question : Comment paramétrer le mandataire ?

Réponse : ✍

Question : Pour une requête combien y a-t-il de canaux et de sockets chez le mandataire ?

Réponse : ✍

Question : Dans quel ordre les informations arrivent-elles sur les canaux ? Connaît-on leurs tailles ? Que doit faire le mandataire avec ces données ? Y a-t-il un ordre séquentiel de traitement ?

Réponse : ✍

Question : Écrivez une méthode “read” qui alloue et rempli un tampon mémoire avec l’entrée et une méthode “write” qui copie ce tempon dans la sortie. Déclarez les attributs nécessaires. Pour information :

- La méthode in.read (b) retourne le nombre d’octets lus dans un tableau
- La méthode out.write (b, 0, n) écrit les n premiers octets d’un tableau

Réponse : ✍

Question : Écrire une méthode qui copy le flux d’entrée sur celui de sortie en utilisant les méthdoe précédentes.

Réponse : ✍

Question : Donnez l’expression régulière permettant l’analyse de la ligne HTTP du client.

Indiquez également comment elle est écrite dans le fichier source Java.

Réponse : ✍

Question : Que faire lors d'une demande de requête d'un client ? (fournir le code correspondant)

- On utilisera les classes DataOutputStream, DataInputStream

Réponse : ✍

Question : Ecrivez une méthode qui met à jour le tampon mémoire déjà lu.

Réponse : ✍

Question : Que faire lors d'une réponse du serveur ? (fournir le code correspondant)

Réponse : ✍

Question : Que faut-il faire pour disposer d'une trace des requêtes émises ?

Réponse : ✍

Question : Que faut-il faire pour réaliser de la désinformation en ligne ?

Réponse : ✍

Question : Que faire pour en faire un outil d'espionnage de mot de passe de messagerie.

Réponse : ✍

2 TP2 : Mandataire

Cette section porte sur le compte-rendu que vous avez à fournir. Repportez-vous aux rappels du début de la section du TD.

Que dois-je faire ?

Vous retrouverez une partie des questions du TD, mais pas nécessairement dans le même ordre. Il est donc important dans votre compte-rendu de recopier les questions. Ce sont ces questions et ces réponses que l'on attend de vous.

Format des archives

Le nom de l'archive doit être informatif (année, module, nom du TP, groupe TD, tous les noms des auteurs) comme :

`2021-M4102-Clavardage-GrpA-X.Dupond-Y.Durand.tar.bz2`

L'archive doit contenir un répertoire avec le même format de nom :

`2021-M4102-Clavardage-GrpA-X.Dupond-Y.Durand.`

L'encadrant doit pouvoir extraire toutes les archives dans le même espace sans que les fichiers ne se mélangent.

2.1 Validation des outils

Question : Reprenez le programme `recvcmd.java` de la dernière séance avec le port 8080. Téléchargez le programme `sendcmd.java` et testez-le avec le serveur précédent.

Réponse : ✍

2.2 Compréhension du sujet

Question : Donnez un schéma d'utilisation d'un mandataire (par exemple web). Décrivez la communication entre le navigateur et le serveur au travers du mandataire. Expliquez comment notre mandataire vient s'insérer.

Réponse : ✍

Question : Pour une requête combien y a-t-il de canaux et de sockets chez le mandataire ?

Réponse : ✍

2.3 Réalisation du logiciel

Consultez l'API Java concernant les classes : String, Integer, ServerSocket, Socket, IOException, DataInputStream, DataOutputStream, Thread

Question : Faire un lanceur qui prend les paramètres nécessaires en arguments.

Réponse : ✍

Question : Ecrire le ou les contsructeurs de la classe du Mandataire et l'activité d'attente des clients.

Réponse : ✍

Question : Écrivez une méthode “read” qui alloue et rempli un tampon mémoire avec l’entrée et une méthode “write” qui copie ce tempon dans la sortie. Déclarez les attributs nécessaires. Pour information :

- La méthode in.read (b) retourne le nombre d’octets lus dans un tableau
- La méthode out.write (b, 0, n) écrit les n premiers octets d’un tableau

---

Réponse : ✍

Question : Écrire une méthode qui copy le flux d’entrée sur celui de sortie en utilisant les méthdoe précédentes.

Réponse : ✍

Question : Donnez l’expression régulière permettant l’analyse de la ligne HTTP du client.

Indiquez également comment elle est écrite dans le fichier source Java.

Réponse : ✍

Question : Que faire lors d’une demande de requête d’un client ? (fournir le code correspondant)

- On utilisera les classes DataOutputStream, DataInputStream

Réponse : ✍

Question : Ecrivez une méthode qui met à jour le tampon mémoire déjà lu.

Réponse : ✍

Question : Que faire lors d’une réponse du serveur ? (fournir le code correspondant)

Réponse : ✍

Question : Que faut-il faire pour disposer d’une trace des requêtes émises ?

Réponse : ✍

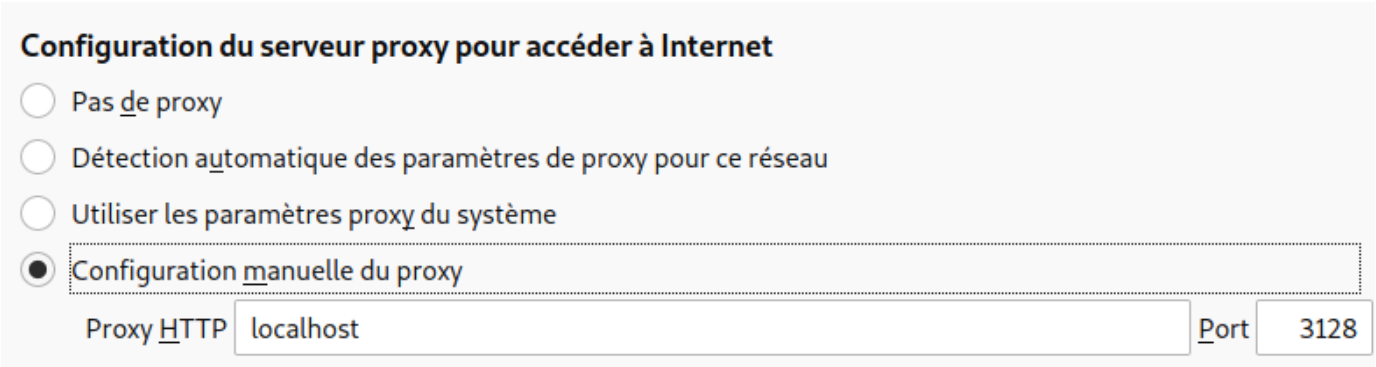
Question : Que faut-il faire pour réaliser de la désinformation en ligne ?

Réponse : ✍

2.4 Test du logiciel

Configuration du navigateur

Pour le test il faudra configurer votre navigateur pour utiliser votre mandataire. Pour ce faire allez dans le menu de votre navigateur firefox : “Edition” => “Préférences”. Puis dans la page “about :preferences” choisissez l’onglet “Général”. En bas de page, dans la section “Paramètres réseau”, cliquez sur le bouton “Paramètres...”. Choisissez la “Configuration manuelle du proxy” pour renseigner les informations “localhost” et “3128”. Puis validez. Vous devez obtenir un formulaire comme suit :



Question : Donnez des exemples de fonctionnement.

- lancez votre mandataire sur le port 3128
- configurez votre navigateur
- allez sur http ://dev.kaz.bzh :80/

Recherchez le mot “KAZ”.

Réponse : ✍

Configuration du navigateur

Pensez à réinitialiser la configuration avec “Utiliser les paramètres proxy du système” de votre navigateur. Sinon, dès que votre programme Java sera arrêté vous ne pourrez plus naviguer.

Question : Quelles sont les limites ? Quelles sont les améliorations à apporter à votre logiciel de “désinformation” en ligne ?

Réponse : ✍

Question : Que faire pour en faire un outil d’espionnage de mot de passe de messagerie.

Réponse : ✍