Département Informatique **Année 2005-2006**
IUT Bordeaux 1 **Année II**

TD PSR3 - Téléinformatique

Programmation à l'aide de l'interface des sockets en TCP

Copiez sur votre compte le contenu du répertoire `/net/exemples/PSR3/seanceTCP`.

Exercice 1 : Expérimentations avec telnet

La commande `telnet` permet d'établir une connexion TCP avec une machine distante, sur un port donné, et peut jouer le rôle d'un "client TCP universel".

1. Les sources TCP BigBen vus en cours vous sont fournies dans `seanceTCP/BigBen`.
   Exécutez un serveur BigBen sur une machine, et connectez vous à ce serveur grâce :
   - au client fourni,
   - à la commande : `telnet machine_serveur num_port`.
   Que remarquez-vous ? Sous quelle forme récupérez-vous les données avec `telnet` ?

Les protocoles "classiques" sont normalisés et décrits par des documents appelés `RFC` (Requests For Comments) qui sont accessibles très facilement sur le Web. Celui pour POP3 (rfc1939) vous est fourni dans `seanceTCP`.

2. Vous pouvez vous connecter avec `telnet` sur divers services : mail, news, ... (voir le fichier `/etc/services` pour les numéros de port). En particulier, essayez le serveur `POP3` dont il sera question dans l'exercice suivant, et dont voici les commandes autorisées :

Commande	Fonction
`USER` *utilisateur*	Nom du compte de l'utilisateur
`PASS` *mot de passe*	Mot de passe
`STAT`	Donne le nombre de messages non lus et leur taille totale
`RETR` n	Récupère le message numéro n
`DELE` n	Supprime le message numéro n
`LAST`	Donne le numéro du dernier message auquel on a accédé
`LIST` *[n]*	Taille de n-ième message ou de tous les messages
`RSET`	Annule la suppression de tous les messages.
`TOP` n k	Affiche les entêtes et k lignes du messages numéro n
`NOOP`	Ne fait rien
`QUIT`	Temine la session POP3

Exercice 2 : Programmation d'un client pour POP3

Les sources de départ se trouvent dans `seanceTCP/POP3`.

L'application fournie est une ébauche d'un client pour `POP3`. Elle réalise la connexion au serveur, l'envoi du nom de l'utilisateur et de son mot de passe, et récupère la réponse du serveur.

Les envois et réceptions sont réalisés à l'aide de **fichiers de haut niveau** (cf cours PSR3 et polycopiés PSR3 et PSR2). Ceci permet en particulier une lecture ligne par ligne des réponses, et facilite énormément leur gestion.

Le but de l'exercice est d'ajouter des fonctionnalités à cette application.

1. Exécutez cette application et familiarisez vous avec les sources (le code contient des commentaires qui complètent la description sommaire précédente et donne la syntaxe d'utilisation). N'oubliez pas d'envoyer des messages dans vos boîtes aux lettres pour qu'il se passe quelque chose ...
2. Rajoutez un menu qui permet de réaliser les fonctionnalités suivantes.
   (**Auparavant, consultez l'annexe ci-dessous !**)
   (a) Affichez le $n$ème message.
   (b) Affichez uniquement le nom de l'expéditeur et le sujet du $n$ème message.
   (c) Affichez le nom de l'expéditeur et le sujet de tous les messages.
3. Imaginez des extensions des fonctionnalités précédentes, et de nouvelles fonctionnalités.
4. **Remettre une version propre et commentée du code la semaine prochaine.**

Annexe : remarques importantes

Voici quelques compléments sur `POP3`, et sur des manipulations de chaînes de caractères.
POP3 :
- Toute réponse d'un serveur `POP3` qui contient plusieurs lignes est terminée par une ligne contenant un caractère '.' tout seul en début de ligne.
- Si, "par malchance", une ligne de données commence par un point, celui-ci est doublé. Ainsi, on ne pourra pas confondre une ligne de données qui ne contiendrait qu'un point (elle sera alors codée "..CR/LF") avec la ligne de fin (".CR/LF")
- Expérimentez avec telnet pour mieux comprendre ces remarques.
- Pour plus d'informations, voir la RFC !

Chaînes de caractères :
- La fonction `strncmp` permet de comparer le début de deux chaînes.
- La fonction `strncasecmp` fait la même chose en ignorant les différences minuscules / majuscules.
- Pour plus d'informations, voir `man` !