

Dans ce TP, vous explorerez les protocoles HTTP, SMTP et POP3 de manière interactive en utilisant le programme `telnet` pour vous connecter à des serveurs.

Exercice 1. SMTP

- connectez-vous au serveur SMTP mis en place sur `info2`. On rappelle qu'un serveur SMTP est conventionnellement à l'écoute sur le port 25.
- utilisez la commande `HELP` (si celle-ci est implémentée par le serveur) pour obtenir des infos sur les commandes possibles.
- en vous connectant au serveur SMTP sur `info2`, envoyez-vous un email.
- en vous connectant par `ssh` à `info2`, vous trouverez votre boîte de courriel dans le répertoire `/var/mail`. Utilisez la commande `cat` pour voir ce qu'il y a dedans.

<code>HELO domain</code>	permet au client de s'identifier
<code>MAIL FROM: adresse</code>	indique l'expéditeur
<code>RCPT TO: adresse</code>	indique le destinataire
<code>DATA</code>	indique que les lignes suivantes forme le contenu du message. La fin du message est signalée par une ligne contenant juste un point.
<code>QUIT</code>	termine la session et ferme la connection
<code>RSET</code>	annule la transaction en cours
<code>VERFY nom</code>	vérifie qu'un utilisateur est géré par le serveur SMTP
<code>EXPN liste</code>	affiche les membres d'une liste de diffusion
<code>HELP [command]</code>	affiche des infos sur les commandes disponibles, ou sur une commande particulière

Exercice 2. POP3

- consultez, sur `http://abcdrfc.free.fr/` le RFC définissant le protocole POP3.
- connectez-vous au serveur POP3 mis en place sur `info2`. On rappelle qu'un serveur POP3 est conventionnellement à l'écoute sur le port 110.
- utilisez cette connection pour lire le message que vous vous êtes envoyé.

<code>USER nom</code>	fournit le nom de l'utilisateur
<code>PASS secret</code>	fournit le mot de passe
<code>LIST</code>	listing bref des messages présents
<code>RETR n</code>	obtient le message numéro <i>n</i>
<code>DELE n</code>	supprime le message numéro <i>n</i>
<code>QUIT</code>	termine la transaction et ferme la connection

Exercice 3. HTML connectez-vous à un serveur HTTP de votre choix, et utilisez la commande `GET` pour obtenir un document hébergé sur ce serveur. Par exemple, pour obtenir la page principale de l'université :

```
telnet www.univ-orleans.fr 80
GET / HTTP/0.9
```

Malheureusement, l'université nous force à passer par un proxy pour accéder aux sites web externes : dans ce cas, il faut se connecter au proxy et lui soumettre la requête. Par exemple, pour obtenir la même page que ci-dessus, mais en passant par le proxy :

```
telnet wwwcacheetu.univ-orleans.fr 3128
GET http://www.univ-orleans.fr/ HTTP/0.9
```

Exercice 4. Développement d'une application *Couriel*

(début du TP2)

Dans les scripts ci-dessous, vous utiliserez le programme `netcat` au-lieu de `telnet`. Ces programmes s'invoquent de la même façon et remplissent le même rôle, mais `telnet` est plutôt conçu pour une interaction sur un terminal, alors que `netcat` est plutôt conçu pour être utilisé dans des scripts. Comme ce programme ne semble pas être installé par défaut, j'en ai placé une version compilée dans le répertoire `/pub/2A/ASR-Reseau`.

- (a) écrivez et testez un script bash appelé `send.sh` pour vous envoyer un email sur le serveur `info2`.
- (b) écrivez et testez un script bash appelé `fetch.sh` pour afficher le premier email de votre boîte de courriel sur `info2` et le supprimer.

Dans ces scripts, vous mettrez directement tout "en dur". Nous généraliserons par la suite.