

## TP n°8

### HTTP

#### Exercice 1 [Un serveur web en ligne de commande]

1. Lancez la commande `nc` avec l'option `-l 8080`, cela ouvre un serveur sur votre machine locale sur le port 8080. Connectez-vous à ce serveur avec votre navigateur préféré. Tapez une ligne dans votre terminal. Que se passe-t-il ? Quittez votre serveur `nc`, que se passe-t-il alors dans votre navigateur ?
2. Dans un fichier `test.html` tapez un exemple de code HTML. Lancez de nouveau `nc` comme pour la question précédente en lui donnant en entrée (grâce à un pipe et à la commande `cat`) votre fichier html. Connectez vous de nouveau à votre serveur avec un navigateur web, que se passe-t-il ? Pouvez-vous rafraichir la page ?
3. De façon à pouvoir rafraichir la page depuis la navigateur, écrivez un script bash bouclant à l'infini et exécutant la commande de l'exercice précédent. Testez avec votre navigateur le résultat.

**Exercice 2** [Un serveur simple] Écrivez en C un programme qui écoute sur le port 8080. Lorsqu'il a accepté une connection, votre programme fermera immédiatement cette dernière. Testez votre serveur à l'aide d'un navigateur web. Assurez-vous que votre programme est capable d'accepter plusieurs connections de suite.

**Exercice 3** [Lecture de requêtes] Modifiez votre programme pour que, lorsqu'il reçoit une connection, il lise des données jusqu'à une ligne vide CR-LF-CR-LF. Une fois la ligne vide trouvée, il enverra une réponse de code 501, par exemple la suivante (où chaque ligne se termine par CR-LF) :

```
HTTP/1.0 501 Pas encore implemente
Content-Type: text/html; charset=utf-8

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <body>
    <p>Desole, j'ai pas encore fini.</p>
  </body>
</html>
```

puis ferme la connection. Comme avant, testez votre programme.

**Exercice 4** [Analyse de requêtes] Modifiez votre programme pour qu'il analyse la première ligne de la requête du client. Celle-ci doit être composée de trois champs séparés par des espaces 1 :

- le premier (le type de requête) est la chaîne de caractères «GET» ;
- le deuxième (le chemin) commence par le caractère «/» ;
- le troisième (le protocole) commence par «HTTP/1.».

Si les trois conditions ci-dessus sont remplies, votre serveur donnera une réponse de code 200 ; dans le cas contraire, il donnera une réponse de code 400.