

1/ La commande netcat établit une connexion avec un serveur. Tout ce qu'il lit sur son entrée standard, il l'envoie au serveur, et tout ce que le serveur lui envoie, il l'écrit sur sa sortie standard.

netcat offre aussi une fonctionnalité "serveur". Dans ce mode d'utilisation, netcat attend sur un port qu'un client se connecte, puis tout ce qu'il lit sur son entrée standard il l'envoie au client, et tout ce que le client lui envoie, il l'écrit sur sa sortie standard. Le mode "listen" est sélectionné grâce à l'option -l ; il faut aussi l'option -p suivi du numéro de port sur lequel écouter.

Dans un terminal, lancer netcat en mode serveur sur le port 8080 :

```
netcat -l -p 8080
```

Dans un autre terminal, lancer un client qui se connecte à ce serveur :

```
netcat localhost 8080
```

Maintenant, les lignes que vous entrez dans le terminal serveur apparaissent aussi sur le terminal client et toutes les lignes que vous entrez dans le terminal client apparaissent aussi sur le terminal serveur.

2/ Nous allons réaliser un serveur grâce à un script bash.

Etudiez le code du script « serveur » pour le comprendre.

Pour pouvoir interagir avec le client, il faut pouvoir lire la sortie standard de netcat pour obtenir les requêtes du client, et pour chaque requête on veut envoyer une réponse sur l'entrée standard de netcat pour qu'il la transmette au client.

On peut résoudre une partie de notre problème en utilisant un tube :

```
interaction | netcat -l -p PORT
```

où interaction est une fonction qui écrit des réponses sur sa sortie standard.

On veut également que cette fonction reçoive sur son entrée standard les requêtes du client: c'est à dire ce qui est écrit sur la sortie standard de netcat. Il faut faire une sorte de boucle pour connecter la sortie standard du netcat sur la droite à l'entrée standard d'interaction sur la gauche. Ceci peut être réalisé grâce à un tube nommé. Le tube nommé est créé par la commande mkfifo. Supposons que \$FIFO soit le chemin d'un tube nommé, alors on peut l'utiliser pour rediriger la sortie de netcat dans l'entrée d'interaction:

```
interaction < "$FIFO" | netcat -l -p PORT > "$FIFO"
```

Notez que lorsque "interaction" se termine, l'entrée standard de netcat est alors fermée, mais netcat ne clos pas pour autant la connexion internet. Pour forcer netcat à clore cette connexion dès que son entrée standard est fermé, il faut ajouter l'option -c:

```
interaction < "$FIFO" | netcat -c -l -p PORT > "$FIFO"
```

3/ 2 fonctions sont déjà écrites (la fonction « commande-non-comprise » et la fonction « quit »)

Vous devez créer une nouvelle fonction « convert » qui transforme un texte en majuscule. Quand le client envoie la commande « convert » avec un argument, l'argument sera converti en majuscule (utiliser la commande « tr »).

Vous testerez ce serveur en vous y connectant grâce à netcat. Si vous installez votre serveur sur une machine distante, il suffit de remplacer « localhost » dans la commande netcat du client en lui donnant l'adresse du serveur.