



Réseaux pour ingénieurs

TP3

Programmation et transmission via socket

Professeur responsable : Ronald Beaubrun

Courriel : Ronald.Beaubrun@ift.ulaval.ca

Responsable des TP : Julien Duchesne et Antoine Lefrançois

Courriel : glo2000a15@gmail.com

Automne 2015

Ce troisième TP a pour but de vous familiariser avec la programmation socket, ainsi que la transmission de données via le réseau. Vous programmerez un petit logiciel en Perl qui exploite certaines fonctionnalités de la transmission de paquets sur le réseau. Pour plus d'informations sur le langage Perl, visitez le site <http://perldoc.perl.org>.

Question 1 – Serveur calculatrice (12 points)

Vous devez programmer une application utilisant les sockets. Celle-ci a pour mission d'envoyer des opérations mathématiques qui seront reçues et effectuées par une autre station. L'application comporte 2 modes, le mode calculatrice (exécuté par la station en écoute) et le mode utilisateur (exécuté par la station émettrice).

Ce programme devra répondre aux caractéristiques suivantes

1. Le mode TCP doit être utilisé par défaut. (0,5 point)
2. Utiliser l'option « -p » ou « -port » pour désigner le port d'écoute. (1 point)
3. Utiliser l'option « -c » ou « -calc » pour démarrer l'application en mode calculatrice (recevoir des paquets), sinon le mode de utilisateur (envoyer des paquets) est utilisé par défaut. (1 point)
4. En mode calculatrice, le programme attend de recevoir trois lignes distinctes soit deux nombres et un opérateur (« + », « - », « * », « / ») puis effectue l'équation et renvoie à utilisateur la réponse. La calculatrice se remet en mode écoute par la suite. Par exemple, utilisateur inscrit « 3 », « 5 », « - », la calculatrice va renvoyer « -2 ». (2.5 points)
5. Si l'utilisateur entre un opérateur qui n'est pas « + », « - », « * » ou « / », l'application doit renvoyer « Mauvais opérateur. Recommencez » au lieu de la réponse et recommencez. (1 point)

6. La station calculatrice doit renvoyer un message après la réception de chacun des deux nombres pour indiquer que le nombre a bien été reçu et la station utilisateur doit l'afficher. Vous n'avez pas à faire de traitement d'erreur sur les nombres. (0.5 point)
7. Utiliser l'option « -d » ou « -destination » permet de désigner l'adresse IP du destinataire (1 point)
8. En utilisant l'option « -h » ou « -help », le programme doit afficher l'aide, soit la façon dont le programme s'utilise. La forme d'affichage n'est pas imposée. Vous pouvez donc vous fier à l'aide affichée par d'autres programmes sous Linux. (1 point)
9. Le programme s'exécutera en ligne de commande.
Par exemple, « Perl TP3.pl -c -p 5678 » (lancement de l'application en mode calculatrice sur le port 5678).
Vous devez vérifier que l'appel contient obligatoirement l'option « -p ». (1 point)
10. Vous devez vérifier que les options « -d » et « -c » ne sont pas dans le même appel, car une station ne peut pas être à la fois en mode écoute et réception.
Par exemple, l'appel suivant est invalide :
« Perl TP3.pl -c -d 192.168.0.123 -p 3333 ». (1 point)
11. En cas d'erreur, le programme doit retourner un message d'erreur significatif et terminer son exécution. Vous pouvez également afficher l'aide à la suite du message. (0.5 point)

«Erreur : L'application ne peut pas utiliser -d et -c simultanément»

«Erreur : L'option -p est obligatoire»

...
12. Chaque message d'erreur devra aussi être conservé dans un fichier de log « Error.log ». Celui-ci doit contenir le message d'erreur ainsi que la date et l'heure à laquelle s'est produite l'erreur en question. Celui-ci est conservé sur la station sur laquelle l'erreur s'est produite. (1 point)

Exemple d'exécution côté serveur (rien n'est affiché, mais vous êtes libre de le faire):

```
> perl tp3.pl -p 5000 -c
```

Exemple d'exécution côté utilisateur (entre [] sont les actions de l'utilisateur) :

```
> perl tp3.pl -p 5000 -d localhost
```

```
[7] [Entrée]
```

```
7
```

```
Reçu: 7
```

```
[76] [Entrée]
```

```
76
```

```
Reçu: 76
```

```
[-] [Entrée]
```

```
-
```

```
Réponse: -69
```

```
[3] [Entrée]
```

```
3
```

```
Reçu: 3
```

```
[5] [Entrée]
```

```
5
```

```
Reçu: 5
```

```
[*] [Entrée]
```

```
*
```

```
Réponse: 15
```

```
[5] [Entrée]
```

```
5
```

```
Reçu: 5
```

```
[6] [Entrée]
```

```
6
```

```
Reçu: 6
```

```
[7] [Entrée]
```

```
7
```

```
Mauvais opérateur. Recommencez
```

Question 2 - Utilisation du serveur calculatrice (3 points)

En utilisant votre programme Perl ainsi que WireShark, donnez-moi les informations suivantes :

1. Démarrez Wireshark et reniflez le réseau sur localhost, c'est-à-dire sur votre propre machine. Utilisez votre programme Perl en mode de calculatrice et écoutez sur le port 42000 TCP de votre serveur. Par la suite, utilisez votre programme Perl en mode opérateur dans une autre console (autre Shell), connectez-vous sur ce port et tapez les lignes suivantes :

« 66 »

« 6 »

et

« + »

Vous devez nous remettre le fichier contenant la trace des paquets que vous avez envoyé. Vous devez faire les mêmes opérations que dans le TP2 pour imprimer le paquet dans un fichier texte. Nommez le fichier « Q2_1 ». (1 point)

Nous devons être capables de reproduire le comportement lors de la correction. (2 points)

Modalités de remise

Veuillez prendre note que les travaux seront corrigés à l'aide du système Kali. Si votre programme ne fonctionne pas sous Kali, vous ne recevrez pas de points.

Vous devez remettre un document compressé « TP3_matricule_matricule.zip » contenant les éléments suivants :

- Votre programme nommé TP3.pl
- Le fichier contenant la trace
- Un fichier PDF contenant vos noms et matricules ainsi que tout commentaire que vous voulez nous faire parvenir.
- Une pénalité sera attribuée de 10% si le document envoyé n'est pas de format PDF