

TP Réseau n°1

Table des matières

Client UDP

Serveur UDP

Scanner de ports

Client UDP

Récupérez la version du serveur compilée pour votre système d'exploitation.

Lancez ce serveur sur votre machine en lui indiquant un numéro de port. Par exemple :

```
% ./serveurDate 8000
```

SHELL

Écrire un client UDP en Python (ou en C ou Java, si vous préférez) permettant de dialoguer avec ce serveur UDP.

Au minimum, ce client doit vérifier ses paramètres, envoyer les commandes au serveur et afficher les réponses de celui-ci.

Exemple de session (on suppose que le serveur tourne sur la machine locale, sur le port 8000) :

```
% python3 client-date.py localhost 8000
Entrez une commande (help pour la liste, quit pour quitter) : date
100 12/02/2018
Entrez une commande (help pour la liste, quit pour quitter) : jour
100 lundi
Entrez une commande (help pour la liste, quit pour quitter) : help
100 Commandes possibles : date, heure, jour, mois, annee, paques, ascension
Entrez une commande (help pour la liste, quit pour quitter) : mois
100 février
Entrez une commande (help pour la liste, quit pour quitter) : heure
100 21:23:28
Entrez une commande (help pour la liste, quit pour quitter) : paques
100 01/04/2018
Entrez une commande (help pour la liste, quit pour quitter) : ascension
100 11/05/2018
Entrez une commande (help pour la liste, quit pour quitter) : bla
200 Requête incorrecte : BLA
Entrez une commande (help pour la liste, quit pour quitter) : quit
%
```

SHELL

Serveur UDP

Écrire en Python le serveur de date dont on rappelle ici le protocole :

- DATE : renvoie la date du jour au format JJ/MM/AAAA .
- JOUR : renvoie le nom du jour de la semaine (en français).
- MOIS : renvoie le nom du jour du mois (en français).
- HEURE : renvoie l'heure au format HH:MM:SS .

- PAQUES : renvoie la date du dimanche de Pâques de l'année courante.
- ASCENSION : renvoie la date du lundi de l'Ascension de l'année courante.
- HELP : affiche la liste des commandes reconnues par le serveur.



Paques et Ascension ont uniquement été choisies car ce sont les seules fêtes du calendrier grégorien dont la date est calculable... N'y voyez aucune connotation religieuse : trouver la date du 14 Juillet ou du 11 Novembre aurait été un peu trop simple...

Le serveur n'est pas sensible à la casse : toutes ces commandes peuvent donc être reçues en minuscules et/ou majuscules.

Toutes les réponses du serveur à des requêtes correctes sont précédées du code 100 . Les commandes incorrectes provoquent une réponse du serveur précédée du code 200 .

Le serveur doit "logger" toutes les commandes qu'il reçoit dans un fichier `serveurDate.log` dont voici un extrait correspondant à la session cliente de la question 1 :

```
% cat serveurDate.log
2018/02/12 21:22:28 Server started
2018/02/12 21:22:28 Listen on :8000
2018/02/12 21:23:13 Received date from 127.0.0.1:65482
2018/02/12 21:23:17 Received jour from 127.0.0.1:65482
2018/02/12 21:23:21 Received help from 127.0.0.1:65482
2018/02/12 21:23:24 Received mois from 127.0.0.1:65482
2018/02/12 21:23:28 Received heure from 127.0.0.1:65482
2018/02/12 21:23:32 Received paques from 127.0.0.1:65482
2018/02/12 21:23:37 Received ascension from 127.0.0.1:65482
2018/02/12 21:23:42 Received bla from 127.0.0.1:65482
2018/02/12 21:26:05 Server stopped...
```

SHELL

Pour le tester, si vous utilisez mi-chelle, vous devrez le lancer sur un port différent de celui des autres : utilisez la formule $7000 + N$ où N est le numéro qui vous sera affecté par l'enseignant.



- Le module `datetime` (<https://docs.python.org/3/library/datetime.html>) de Python fournit toutes les classes nécessaires à la gestion du temps.
- Il est souhaitable d'écrire un module `paques.py` que vous importerez dans le code du serveur. Pour le calcul de cette date, utilisez l'un des algorithmes mentionnés sur [cette page](http://fr.wikipedia.org/wiki/Calcul_de_la_date_de_P%C3%A2ques) (http://fr.wikipedia.org/wiki/Calcul_de_la_date_de_P%C3%A2ques).

Vous pouvez évidemment améliorer ce protocole.

Scanner de ports

Pour ceux qui ont terminé les deux questions précédentes :

Écrire un scanner de port, c'est-à-dire un programme permettant de tester les ports ouverts sur une machine donnée...

Ce scanner, par défaut, teste tous les ports (UDP et TCP) mais dispose d'une option permettant de restreindre la recherche aux seuls ports UDP ou aux seuls ports TCP.



Les serveurs considèrent généralement comme hostile un scan de leurs ports. Je vous suggère donc de tester sur vos propres machines et de ne faire un test sur une machine extérieure qu'à la fin...

Exemple d'utilisation :

```
% python3 scan.py mi-chelle          # scanne tous les ports ouverts de mi-chelle
% python3 scan.py -t mi-chelle        # scanne tous les ports TCP ouverts de mi-chelle
% python3 scan.py -u localhost        # scanne tous les ports UDP ouverts de la machine locale
```

SHELL



Si vous avez trouvé un port ouvert, c'est donc que vous avez pu établir une liaison avec lui. N'oubliez pas de la fermer...

Le résultat devra afficher les ports ouverts (un bonus serait d'afficher pour chacun d'eux le protocole correspondant s'il s'agit d'un "well-known protocol"). Pour cela, vous pouvez utiliser `/etc/services`.

Vous pouvez vous inspirer du résultat de la commande `nmap` (voir <http://nmap.org/man/fr/>)

Last updated 2019-01-30 13:03:09 +0100