

IN405 – Feuille de TD #8

Communication II

Objectif : manipulation des communications à d'autres fins.

Instructions :

- vous aurez besoin de l'archive `td8-contents.tar`, la description de l'archive est donnée dans le fichier `README.md`, à lire avant de vous lancer dans l'implémentation ;
- l'exercice 8.1 est un TP noté :
 - il doit être fait en monôme et rendu **au plus tard le dimanche 25 avril, 23h59 (heure française métropolitaine)** sur moodle, dans la section 'Travaux Dirigés notés' ;
 - le rendu consiste en une archive `.tar` constituée des fichiers sources, du matériel de compilation et d'un fichier `README.md` décrivant comment utiliser votre programme ;
 - un malus de point sera attribué pour tout retard ou pour toute correspondance suspecte avec les autres rendus.

Exercice 8.1 - Dictionnaire Pascal

Votre objectif est d'implémenter un serveur : programme exécuté en tant que tâche de fond, communiquant avec des clients par messages. Ce serveur garde dans sa mémoire un ensemble de constantes numériques, avec laquelle on souhaite interagir : ajout de constante, suppression, demande de la valeur d'une constante. Vous devez également développer les fonctions du client qui s'occupent de la communication avec le serveur.

Le programme se découpe en deux parties distinctes, client et serveur, qui s'enverront des messages par l'intermédiaire de tubes.

Le client envoie au serveur diverses commandes pour agir sur le dictionnaire du serveur :

- ajout de constante `./client dict + name value` : la constante `name` de valeur `value` est ajoutée au dictionnaire `dict`.
- suppression de constante `./client dict - name` : la constante `name` est supprimée du dictionnaire `dict`.

- demande de la valeur d'une constante `"/client dict ? name"` : retour du serveur de la valeur de la constante `name` stockée dans le dictionnaire `dict`.
- extinction du serveur `"/client dict !"` : le serveur possédant le dictionnaire `dict` est éteint.

Le client renvoie un code de retour nul si tout s'est bien passé, non nul sinon (généralement 1). Pour rappel, vous pouvez vérifier la valeur du code de retour d'un programme en affichant la variable `$?`.

Le serveur reçoit les requêtes des différents clients et met à jour son dictionnaire. Il peut être amené à répondre à un client si ce dernier questionne la valeur d'une constante que le serveur possède.

```
$ ./server dict size
```

La commande est constituée d'un nom de dictionnaire `dict`, permettant aux clients de communiquer avec le serveur et d'une taille `size` max d'entrées que ce dictionnaire peut posséder.

Les messages transmis entre le client et le serveur sont expliqués ci-dessous :

- Addition : caractère opérateur `+`, nom de la constante et nom de la valeur ;
- Suppression : caractère opérateur `-` et nom de la constante ;
- Recherche : caractère opérateur `?` et nom de la constante ;
- Extinction : caractère opérateur `!`.

D'autres informations :

- accéder à un dictionnaire qui n'est pas accessible est considéré comme une erreur de la part du client ;
- ajouter une constante déjà présente dans le dictionnaire modifie sa valeur ;
- retirer une constante qui n'existe pas dans le dictionnaire ne retourne pas d'erreur de la part du client ;
- demander la valeur d'une constante qui n'existe pas retourne une erreur auprès du client, qui doit donc afficher `undefined` et retourner un code d'erreur non nul ;
- vous ne devez pas gérer les conditions de concurrence : on suppose ici qu'un seul client interagit à la fois avec un serveur.