Clément CHARPENTIER

Rapport projet en programmation réseau

Programmation

- Lecture STDIN

Au début du projet réseau, nous avions alors que les fonctions attach et detach, pour pouvoir structurer notre arbre avec nos différents noeuds, nous passions toutes les commandes en arguments. Il fallait donc pouvoir lire sur l'entrée standard les différentes commandes sans que cela n'empêche les fonctions réseaux. Je suis d'abord parti sur un read() de la sortie standard mais le faite que la fonction soit bloquante m'a rapidement posé problème. J'ai essayé de contourner le problème de différente façon surtout en recherchant comment rendre le read non bloquant ou en utilisant une fonction similaire sans se problème. Puis ensuite j'ai eu l'idée d'utiliser le select() déjà présent car lors des TD3 et TD4 de réseaux on avait vu que les numéros de descripteurs commençait à 3, les 3 premiers étant réservés à STDIN, STDOUT et STDERR. J'ai implémenté un parser pour pouvoir exécuter nos premières commandes, on l'a remplacé par celui de Théo beaucoup plus performant.

- Debug taille paquet (réalisé avec Clément Foissard)

A ce moment-là, notre programme ne pouvait qu'envoyer des « attach » et « detach » et pourtant, nous avions des problèmes d'envoi/réception de paquets TCP. On s'était rendu compte qu'en s'amusant a modifié BUFFSIZE les programmes ne recevaient plus rien. La variable BUFFSIZE affectant directement la taille des paquets qu'on envoyait. On alors lancé Wireshark sur nos PC respectif avec Clément et on a répéter le bug en jouant sur la taille de BUFFSIZE. Le problème était une petite erreur à l'élaboration du programme, nous avions utilisé « read() » qui va prendre en compte uniquement le premier paquet qui arrive car TCP découpe les paquets trop gros. Alors qu'il fallait qu'on utilise le flag « MSG_WAITALL » disponible uniquement avec la fonction « recv() ». Ainsi, à la suite de ce debug, on a décidé de tester la robustesse de notre architecture en transmettant le plus gros paquet possible : nous avons envoyé un paquet de 4Go qui a été reçu avec succès (après 15-20 minutes d'attente). On a aussi corriger à ce bug un autre problème. A chaque commande envoyé on envoyait un ACK de la commande, TCP le fait déjà, cela était donc inutile et cela ralentissait notre programme, on a donc enlevé nos ACK.

- Debug fermeture propre des descripteurs (réalisé avec Clément Foissard)

Dans notre programme lorsque 2 processus sont connectés l'un à l'autre si un des deux souhaitent se déconnecter et se terminer il peut faire un /exit ou /detach puis /exit pour se déconnecter proprement. Or lors de notre développement, par rapidité, on utilisait Ctrl-C pour couper le programme et cela avait pour effet de faire dégénérer l'autre processus (alors inutilisable) qui avait perdu la connection mais sans fermer le descripteur. Cela faisait même swapper l'ordinateur et ça met arrivé personnellement plusieurs dizaines de fois. On a donc Wireshark avec Clément, vu que même en faisant un Ctrl-C il y a un paquet qui indique que la connection a été perdu et donc qu'on peut savoir qu'elle descripteur fermer, ce qu'on a fait.

- Fonction Getway (ou recurway, réalisé avec Clément Foissard)

La fonction getWay() permet d'en un arbre de trouver une chemin passant par route entre 2 noeuds. Cette dernière remonte donc jusqu'au groupe « ROOT », le groupe le plus haut placé et stocke son trajet. La fonction recurWay() est simplement une fonction récursive de parcours d'arbre, inspiré du parcours en profondeur classique. Elle retrace le trajet de « ROOT » vers la destination et stocke son trajet.

- Portabilité Windows

La portabilité Windows était une des consignes, après plusieurs recherche je me suis aperçu que 2 options s'offraient à moi. La première utiliser la librairie des socket Windows, ce qui semblait compliqué sachant que nous étions à un niveau très avancé du programme car il fallait retouché à chaque fonctions réseaux. La deuxième, celle pour laquelle on a opté, était d'utiliser les librairies POSIX qu'on a utilisé sous Windows grâce à un logiciels, Cygwin. Après une installation compliquée, ça marche vraiment bien et ça évite de devoir trop penser à la portabilité qu'en on a codé qu'avec des librairies POSIX pendant tout le projet.

Clément CHARPENTIER

Gestion du groupe

On a choisi de se mettre tout les quatres car nous sommes vraiment très TRÈS proche dans la vie privée et cela nous a beaucoup aidé pour le projet. Au départ le groupe a un niveau en programmation C et réseau très hétérogène, on a d'un côté Théo qui est très à l'aise avec le C et qui tout au long du projet a énormément contribué de l'autre moi et Pierre beaucoup moins à l'aise. C'était donc très enrichissant pour nous 2 car cela nous a permis d'énormément apprendre durant ce projet aussi bien en C qu'en réseau. Globalement on est tous très fier du travail fourni et des compétences acquises.

Pour ce qui est du travail, on a beaucoup codé les uns chez les autres en faisant des sessions a 4, ce qui permet d'avancer rapidement et de suivre l'avancé globale du projet, tout le monde c'est ce que chacun développe et dés qu'on rencontre une difficulté les autres n'ont pas besoin de se plonger dans notre code car ils le suivent déjà ce qui est très pratique.

Pour l'organisation du travail j'ai créer un document répertoriant toute les fonctionnalités à implémenter (disponible sur le git "ProjetReseauTaches") j'ai essayé tout du long de faire travailler tout le monde sur différent point en fonction de leur niveau, de ce qu'il avait déjà fait dans le projet et ce qui les intéressait.

Un autre point qui a aussi été enrichissant, sont les différents phase de debug réalisé par des personnes qui n'avaient pas développés cette partie du programme. Ça permettait de vraiment comprendre parfaitement comment cela marchait d'apporter un point de vue extérieur et c'était très enrichissant.

Globalement pour choisir comment on allait implémenté les différentes choses on décidait tous ensemble en définissant la structure en amont quel paramètre ça prend, qu'est ce que ça retourne, quelles autres fonctions ça utilise. Cependant ayant lu plusieurs fois le sujet, dont il est assez difficile de tirer le cahier des charges précis, j'ai plusieurs fois "imposé" mon idée mes camarades me faisant confiance, c'est notamment le cas sur la partie groupe, qu'elle noeuds peut ou ne peut pas communiquer avec tels noeuds.

Pour ce qui est de la position de chef de projet j'ai été élu à la courte paille.