1 Transfert de fichier

L'objectif de cet exercice est d'apprendre à réaliser des transferts de fichiers en TCP. En d'autres termes, vous allez programmer un transfert FTP et donc comprendre les bases sur lesquelles s'appuie l'implémentation d'un tel protocole applicatif.
Dans un premier temps, nous envisageons le transfert d'un fichier de petite taille, i.e. dont la taille ne dépasse pas la taille du buffer/tampon de la socket de réception. Ce transfert est à réaliser par une application avec :

- un programme client qui 1) lit **EN UNE SEULE FOIS** le contenu d'un fichier de taille inférieure à la taille du buffer de réception du serveur et 2) envoie ce contenu **EN UNE SEULE FOIS** au serveur. Le fichier à envoyer (comme tout autre fichier à transférer) est à ranger dans un répertoire "./emission" et son nom sera passé en paramètre de votre programme.
- un programme serveur qui reçoit **EN UNE SEULE FOIS** le contenu d'un fichier et le stocke dans un fichier à ranger dans un répertoire "./reception".

1. définir un protocole d'échanges entre le client et le serveur pour réaliser ce transfert.
2. écrire les deux programmes client et serveur. Le programme serveur devra afficher le nombre d'octets reçus et correspondant au contenu du fichier transféré.
3. exécuter l'application (en utilisant deux machines différentes) et s'assurer que le fichier reçu par le serveur est bien une copie du fichier envoyé par le client.
4. tester votre application sur différents exemples de fichiers en respectant une taille inférieure à celle du buffer de réception. **Remarque** : vous ne devez pas avoir besoin de modifier votre programme pour tester différents scénarios, corriger sinon.
5. pour chaque exemple de fichier testé, le serveur reçoit-il le contenu en entier du fichier ? Quelle est votre conclusion ?

A présent, tester votre application sur des fichiers dont la taille est supérieure à la taille du buffer de réception du serveur.

1. Que se passe-t-il ? **Remarque** : ces tests ne doivent pas nécessiter autre modification que la taille des tampons stockant vos messages qui, à présent, peuvent avoir une taille qui dépasse la taille du buffer de réception de la socket du serveur.
2. Expliquer le problème et corriger vos programmes. Le programme client devra afficher le nombre d'octets effectivement envoyés et correspondant au contenu du fichier transféré.
3. Est ce raisonnable de continuer à lire le contenu d'un fichier en une seule fois ? Expliquer pourquoi et améliorer votre programme client.

Donner une conclusion globale sur le comportement de TCP.

Aide

Pour la manipulation d'un fichier, vous pouvez utiliser les fonctions suivantes :

- pour l'ouverture d'un fichier :

  FILE * fopen(const char *restrict filename, const char *restrict mode) ;

- pour connaitre la taille d'un fichier :

  int stat(const char *path, struct stat *buf) ;

- pour lire dans un fichier :

  size_t fread(void *restrict ptr, size_t size, size_t nitems, FILE *restrict stream) ;

- pour écrire dans un fichier :

  size_t fwrite(const void *restrict ptr, size_t size, size_t nitems, FILE *restrict stream) ;

- pour fermer un fichier :

  int fclose(FILE *stream) ;

Enfin, pour connaitre la taille du buffer de réception d'une socket, vous pouvez :

- soit consulter le contenu de : /proc/sys/net/ipv4/tcp_rmem

- soit utiliser la fonction
  int getsockopt(int socket, int level, int option_name, void *restrict option_value, socklen_t *restrict option_len) ;
  en passant en paramètre SOL_SOCKET pour level, SO_SNDBUF pour option_name, l'adresse d'un entier pour option_value et la taille d'un entier pour option_len. option_value permet de récupérer la taille du buffer en réception.
- soit une autre solution de votre choix.