

L'objectif de ce TP est de réaliser des serveurs TCP capables de traiter plusieurs clients. Il s'agit de reprendre des applications déjà réalisées en TP et traitant un seul client pour passer à des applications multi-clients.

## 1 Transfert de fichiers (suite)

Nous souhaitons modifier l'application de transfert de fichiers du TP 4, pour que le serveur puisse recevoir des fichiers en provenance de différents clients.

1. Pour traiter les demandes de plusieurs clients sur le serveur, quel type de traitement serait le mieux adapté :
  - (a) Un traitement itératif?
  - (b) Un traitement concurrent ?
  - (c) Un traitement avec multiplexage ?
2. Modifier le serveur pour réaliser ce traitement.
3. Tester votre application avec plusieurs clients (sur plusieurs machines).

**Note** Pour éviter les conflits sur les noms de fichiers envoyés par différents clients, le serveur peut créer, pour chaque client, un sous-répertoire dédié dans `./reception`

## 2 Application de chat

En partant de l'application réalisée à l'exercice 3 du TP 3 ("les échanges en TCP"), réaliser une application dans laquelle :

- un client permet de saisir des chaînes de caractères à envoyer au serveur et afficher des chaînes de caractères envoyées par ce serveur.
- à la réception d'une chaîne de caractères envoyée par un client, le serveur la retransmet à l'ensemble des autres clients connectés.

1. Pour gérer les différents clients quel type de traitement serait le mieux adapté :
  - (a) Un traitement itératif?
  - (b) Un traitement concurrent ?
  - (c) Un traitement avec multiplexage ?
2. Ecrire le programme client et le programme serveur.
3. Tester votre application avec plusieurs clients (sur plusieurs machines).
4. Que faire pour que le client puisse recevoir/afficher des messages reçus pendant la saisie ? Modifier le programme client pour supporter ce cas.