

# Sujet du TP n° 6

Placez-vous dans le répertoire **PSE/TP6**.

## Application multithread modèle statique

Copiez dans **TP6** les fichiers **serveur.c** et **client.c** de **TP5**.

Dans le TP précédent, nous avons mis en œuvre un serveur multithread, dans lequel un thread est créé à chaque nouvelle connexion d'un client.

Le but du présent TP est de mettre en œuvre une cohorte de taille fixe de threads *workers*. L'algorithme à implémenter est le suivant :

### THREAD PRINCIPAL

```
...
créer cohorte de threads workers
...
BOUCLE INFINIE
    connexion d'un client (accept)

    sélectionner un worker libre et lui indiquer qu'il a une session
    client à gérer
    (s'il n'y a pas de worker libre : attendre)
FIN BOUCLE
```

### THREAD WORKER

```
BOUCLE INFINIE
    le worker attend d'avoir une session client à gérer

    session de dialogue avec le client

    une fois la session client terminée, le worker indique qu'il est
    de nouveau libre
FIN BOUCLE
```

- Pour les données des threads on n'utilisera plus une liste chaînée de structures DataThread (données utiles DataSpec) mais un tableau de structures DataSpec.
- Pour gérer l'état d'un worker et le synchroniser avec le thread principal, on utilisera sa variable canal (-1 le worker est libre, > 0 le worker a une session client).
- Les attentes (worker libre, session client à gérer) se feront par des boucles dans lesquelles on teste et on temporise. On utilisera la fonction **usleep()** (microsecondes).

## Amélioration du serveur multithread modèle dynamique

Ecrire une version du serveur de TP5 et/ou du module datathread qui libère les maillons de la liste chaînée.

(c) Philippe Lalevée, 2023-2024.