

# MINI-PROJET RÉSEAU

## *Programmation socket en C*

---

## 1 Objectif

L'objectif de ce mini-projet est de réaliser un système d'affichage partagé. Il est donc constitué d'un serveur et d'un ou plusieurs clients.

Le service permettra la mise à jour d'une fenêtre partagée en :

- effaçant la fenêtre,
- traçant une ligne,
- traçant un rectangle,
- incrustant une image. (Binômes et Trinômes)
- incrustant une vidéo, (Trinômes)
- affichant plusieurs fenêtres (serveurs) (Trinômes)

## 2 Moyens

Afin de simplifier le travail, plusieurs unités de compilation C sont fournies afin de gérer l'affichage (en utilisant la lib-SDL2), et la lecture d'images (en utilisant les lib-jpeg, lib-png, et video4linux).

Des fichiers de test sont fournis pour illustrer l'usage de ces unités.

## 3 Protocole

Je vous propose le protocole suivant, de façon à vous simplifier le travail. Comme à l'accoutumée, vous pouvez concevoir votre protocole vous-même si vous pensez arriver à un système plus simple... Je vous demanderai alors de décrire précisément le protocole que vous aurez élaboré.

### 3.1 La base

Le protocole proposé est basé sur UDP.

- effacer la fenêtre : 'C', Couleur (5 octets)
- tracer une ligne : 'L', X1, Y1, X2, Y2, Couleur (21 octets)
- tracer un rectangle : 'R', X, Y, Largeur, Hauteur, Couleur (21 octets)
- incruster une image : 'i', X, Y, Largeur, Hauteur, PortTCP (19 octets)
- incruster une vidéo : 'v', X, Y, Largeur, Hauteur, PortTCP (19 octets)

Chaque message est donc constitué d'une lettre ASCII, suivie d'un ou plusieurs arguments entiers. Les coordonnées (X, Y, L, H) seront fournies sur 32 bits en notation Big-Endian.

Les couleurs seront fournies au format ARGB sur 32 bits, avec d'abord la composante Alpha (opacité, 0=transparent, 255=opaque) sur 8 bits, suivi de Rouge sur 8 bits, puis Vert sur 8 bits, et enfin Bleu sur 8 bits.

Le port TCP sera fourni sur 16 bits en notation Big-Endian.

### 3.2 Pour les binômes

L'envoi d'images se fera sur un port TCP dédié, ouvert par le client. En effet, l'envoi de trames UDP est limité à 64Ko de données, mais le taux de perte de telles trames est très élevé (fragmentation en trames IP de 1480 octets, sans aucune garantie).

L'image sera envoyée au format ARGB, sans entête : Le premier octet envoyé correspond au premier pixel de l'image, et la connexion est fermée dès que le dernier pixel est envoyé. Ainsi une image 320x200 sera envoyée comme un flux de  $320 \times 200 \times 4$  octets, soit 256Ko.

### 3.3 Pour les trinômes

L'envoi de vidéo se fera sur le même principe que l'envoi d'une image, excepté que la connexion ne sera pas fermée à la fin de l'envoi d'une image. La seconde image de la vidéo sera envoyée à la suite, puis la troisième, et ainsi de suite...

## 4 Evaluation

Chaque groupe d'étudiants enverra par e-mail une archive au format ZIP **au plus tard le 31 décembre 2016**, heure de Paris. L'archive sera envoyée à L. Jeanpierre ou à P. Lefebvre selon votre groupe de TP. Vous devrez inclure (Tout autre fichier sera refusé!) :

- les fichiers C, H,
- le script de compilation (makefile, cmake, shell-script, ...),
- un fichier README donnant le nom de tous les étudiants du groupe et expliquant la compilation et le protocole réseau si vous l'avez modifié.

Au moins deux exécutables *LINUX* doivent être créés par les sources rendus : un serveur et un ou plusieurs clients (aucune interface graphique n'est demandée pour le client ; un exécutable par 'dessin' est acceptable)

Le serveur doit accepter les ordres d'un ou plusieurs clients et les exécuter dans l'ordre sans bloquer. Ainsi, l'envoi d'une image puis le tracé d'une ligne dessinera la ligne par dessus l'image. Une vidéo ne doit pas empêcher le dessin par d'autres clients, mais les images successives peuvent l'effacer... (par exemple, un dessin par dessus une vidéo sera effacé par l'image suivante de la vidéo)

Le protocole réseau décrit doit être scrupuleusement respecté. Un JAR implantant le protocole décrit dans ce document sera mis à disposition ultérieurement. Il vous aidera à valider votre implantation du protocole.

Le ou les clients doivent permettre de choisir l'adresse IP du serveur et les caractéristiques du dessin (coordonnées, image, couleur)

La qualité de la programmation (commentaires, nom des variables et des fonctions, propreté, indentation, gestion des erreurs, ...) sera prise en compte dans la notation.

Une mini-soutenance de groupe sera planifiée en janvier. Dans cette soutenance rapide, vous ferez une démonstration de votre travail, et chaque étudiant devra détailler sa participation au projet. L'enseignant posera alors quelques questions à chacun.

**Bon travail !**