

Cinquième Labo UNIX 2018.

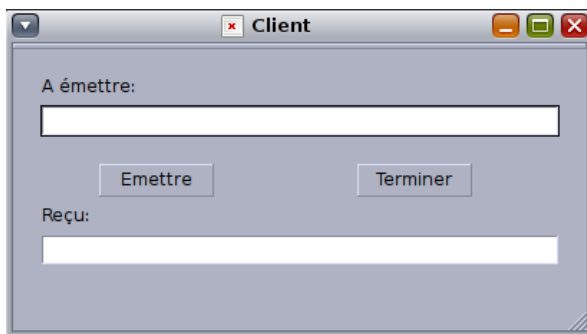
Exercice sur les files de messages :

L'écriture du fichier mafefile est encore et toujours la même, je n'en parle donc plus.

L'objectif de ce dossier est de se familiariser avec les files de messages, c'est-à-dire la création, l'écriture et la lecture,

Ici, il s'agit d'effectuer un transfert d'une chaîne de caractères entre 2 processus en la faisant transiter par un **Serveur**.

Un interface Qt vous est donné.



Vous ne devez travailler que sur le fichier **fenetreclient.cpp, main.cpp** et **Serveur.cpp**.

3 étapes pour y arriver.

Première étape :

Il s'agit de transférer un message de la zone **A émettre** dans la zone **Reçu** par une file de messages. Le **Serveur** crée la file de Messages.

Le processus **fenetreclient** s'y rattache. (Dans le programme main.cpp)

Dans le programme fenetreclient.cpp , la fonction on_BouttonEmettre_clicked() lit le message dans la zone lineMessageEmettre, l'écrit sur la file et le lit **directement** sur la file et le lit pour l'afficher .

Pour cela, vous disposez des éternelles méthodes setMessageEmettre(), setMessageRecu(), getMessageEmettre() et getMessageRecu() .

2^{ème} étape :

Le message à émettre doit passer par le processus **Serveur**.

Celui-ci lit le message et le retransmet immédiatement.

Même programme. Mais pour bien montrer que le message passe par le Serveur, celui-ci fait précéder le message par la chaîne «(Serveur) » avant de le transmettre.

3^{ème} étape.

Ici, on demande au Serveur d'émettre un **SIGUSR1** à la fenêtre pour lui indiquer qu'un message est émis.

La fenêtre lira le message dans un Handler.

4^{ème} étape.

On lance 2 fenêtres qui communiquent entre elles.

Le message émis par une fenêtre est donc affiché sur l'autre.

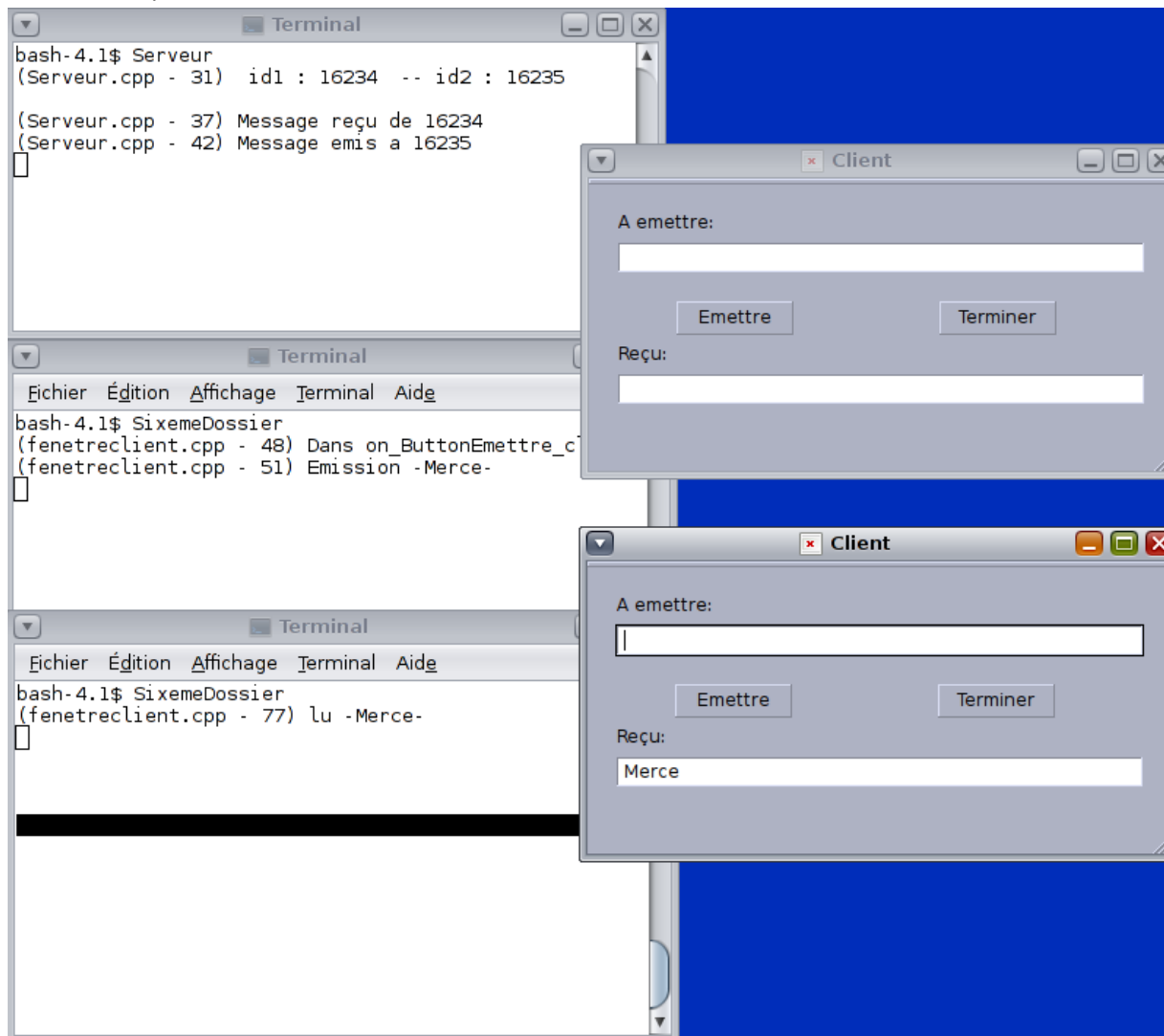
Le Serveur doit donc mémoriser le pid des 2 fenêtres pour faire le transfert.

Il peut le connaître, car il suffit, avant le lancement de la fenêtre, de lui communiquer son pid.

(Dans le programme main.cpp, juste après la récupération de la file)

Le Serveur n'aura qu'à le stocker dans une variable.

Le reste est plus ou moins la même chose.



Noms des variables utilisées :

lineMessageRecu

lineMessageEmettre pas de commentaire.

BouttonEmettre

BouttonTerminer

Et les fonctions correspondantes

Pour les plus avancés. **FACULTATIF**

Doubler le Serveur.

En effet, on peut lancer 1 ou 2 **Serveur. Pas 3.**

C'est le premier qui lit la requête qui la traite.

Ainsi, si on en stoppe 1, l'application continue sans problème.

Et on peut en relancer un autre.