

Devoir 1 DU COURS IFT **585**

*TITRE*

f**onctionnement d’un protocole bidirectionnel à reprises multiples (sélectives)**

|  |  |
| --- | --- |
| *PAR*  BILODEAU CEDRIC 15070012 CISSE OUSMANE COMEAU SAMUEL NGUIFFO MAMNO PATRICIA 16155893 ROY MICHAEL | *Professeur***:**  **Bessam Abdulrazak** |

Table des matières

[Mise en contexte 3](#_Toc484959048)

[Contenue du zip 3](#_Toc484959049)

[Utilisation 3](#_Toc484959050)

[Procédure d’exécution 3](#_Toc484959051)

[Méthode de compilation et d’exécution 3](#_Toc484959052)

[Explication du programme 3](#_Toc484959053)

[Main 3](#_Toc484959054)

[Fonction emetteurRecepteur 3](#_Toc484959055)

[Fonction supportTransmission 3](#_Toc484959056)

[Class TamponCirculaire 3](#_Toc484959057)

[Class Trame 3](#_Toc484959058)

[Namepace Hamming 3](#_Toc484959059)

[Class SortieFichier et EntreFichier 3](#_Toc484959060)

[Étapes de conception 3](#_Toc484959061)

[Degré de fonctionnement 4](#_Toc484959062)

# Mise en contexte

Dans le cadre de notre premier devoir de télématique, nous avons implémenter en C++ le rejet sélectif comme Protocol de liaison et Hamming pour la correction/détections des erreurs.

# Contenue du zip

# Utilisation

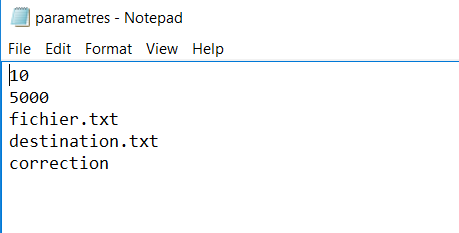
## Procédure d’exécution

## Méthode de compilation et d’exécution

## Le fichier de paramètre

Ce fichier doit se nommer « parametres.txt ». Le fichier de paramètre est composé de 5 paramètre. Chaque paramètre prend une ligne. Le premier paramètre est la taille des tampons en nombre de trame qui peuvent prendre. Le second est le délai de temporisation en millisecondes. Le troisième est le chemin du fichier à copier. Le quatrième est le chemin de destination pour le fichier à copier. Le dernier est si le programme roule en mode détection d’erreur ou en mode correcteur d’erreur avec Hamming. Pour être en mode correction, la ligne doit exactement avoir les caractères suivant : « correction », sinon il est en mode détection automatiquement.

Exemple



# Explication du programme

## Main

Le main ne fait que lire le fichier de paramètre et partir les trois thread. Il y a deux thread qui roule sur la fonction (emetteurRecepteur) et une autre qui est la couche physique (supportTransmission).

## Fonction emetteurRecepteur

Cette fonction est l’implication du rejet sélectif vue dans le cours.

## Fonction supportTransmission

Cette fonction ne fait que passer les données entre le thread émetteur et le thread récepteur.

## Class TamponCirculaire

Class gérant l’ensemble des fonctionnalités du tampon nécessaire au rejet sélectif. Il utilise la class trame.

## Class Trame

Class représentant notre trame. Celle-ci est composée de 3 bits pour numéro de séquence, de 2 bits pour le type et de 16 bits pour les données. Il y a aussi 5 autre bits pour Hamming. Cette trame est contenue dans un uint32\_t, mais il y a quelques bits qui ne servent à rien.

## Namepace Hamming

Le namespace Hamming comprend l’ensemble des fonctions nécessaire pour l’encodage, décode, détection et correction de la trame.

## Class SortieFichier et EntreFichier

Ces deux class permette l’écriture et la lecture de fichier à

# Étapes de conception

Pour la conception de devoir, nous avons commencé par faire tous les class qui seront utile à la conception du rejet sélectif. Nous avons implémenté Hamming au début aussi. Par la suite, nous avons intégré les différentes class. Par la suite, nous avons tenté de faire la synchronisation, mais malheureusement celle-ci était mal faite. Par la suite, nous avons fait le code de la fonction émetteur-récepteur. Finalement, nous avons fait beaucoup de débogages, car nous avions beaucoup de bug. Au début, les bugs étaient dans la synchronisation par la suite dans le code de l’émetteur-récepteur et dans le tampon.

# Degré de fonctionnement