PROJET DE SESSION

INF-1009 Réseaux D'ordinateurs 1

Session HIV-22

Fait Par :

GANSONRE ISMAEL GANI06069700,

SEMANOU RICHAR SEM0349500,

RAMI ISKANDER BERLAT BERR80110000,

DIALLO MARIAMA DIAM67590000,

DIALLO FATIMA DIAF06629900,

AUDREY BONI BONIA90560000

1) Problèmes ou difficultés rencontrées

Le principal problème rencontré a été la compréhension de l'énoncé pour la réalisation du travail, il a fallu plusieurs jours pour analyser le problème à proprement dire. Un autre problème a été de faire fonctionner les Threads, étant à ma première expérience avec ces derniers sous Java, ce fut un bon gros défi de faire fonctionner le tout.

2) Description du logiciel

Notre logiciel est une simulation de connexion et transfert des données vers un ordinateur distant. On a le système local qui est devisé en deux couches une de transport et une autre couche réseau. Du coup la liaison des données et de l'ordinateur distant sont simulés complètement par la couche réseau, Ses composantes seront expliquées plus en détail ultérieurement. Nous avons chacune de c'est dernières couches communique avec sa couche supérieure et inferieur. La couche

transport lis une liste d’instruction dans un fichier pour la faire passer ensuite à des threads de la couche réseau qui fabrique des paquets afin de les envoyer au distant simulé par le réseau. Puis le distant prends les paquets, faire inscrire la donnée reçue dans un fichier et répond adéquatement à la couche réseau en gardant un log des transactions.

Enfin, le réseau décide de conserver ou fermer les connexions avec le distant au besoin et indique au transport lorsqu’une connexion est fermée. Le transport vas noter aussi les ouvertures et fermetures de connexion dans un fichier.

3) Instructions spéciales

On n’a pas beaucoup d’instructions spécialisé pour exécuter notre programme du coup il suffira regarder le terminal pour avoir une

confirmation de chaque pression sur le bouton lancer à partir de

la classe main et de choisir l'option « démo » file qui vas se

charger dans la vue « S_lec ». Puis d'appuyer sur le bouton

« Commencer » pour que tout se fait automatiquement

Ensuite on pourra constater ce qui s’est passé à l’aide des différents fichiers et de leur vue dans l'interface. (Il est normal que le programme prends une dizaine de secondes à s’effectuer, car il doit parfois attendre qu’un thread se finir).

Les fichiers textes sont situés sous :

4) Description des fichiers

Les différents fichiers textes que nous utilisons sont les suivants :

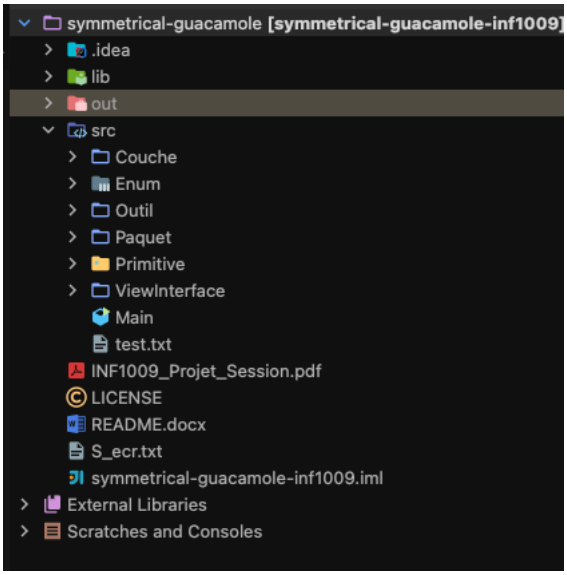
- S_lec : Contient les différentes instructions que nous souhaitons envoyer au distant.
- S_erc : Contient les différentes instructions que nous souhaitons envoyer au distant.

- L_ecr : Ce fichier correspond à ce que reçoit l'application sur la machine distante, Il n'inscrit que le contenu des paquets de data reçus, précédés du numéro de la connexion qui les a envoyés

- L_lec : Ce fichier correspond au fichier log de la liaison de données. Le distant note toutes les transactions dans ce fichier. Il commence encore une fois par le type de paquet suivit par le numéro de la connexion, et de la raison.

5) Vue d'ensemble du projet

Comme vous pouvez voir sur cette capture on trouve la vue de l'ensemble du projet implémente :



6) Description des classes

- Classe main

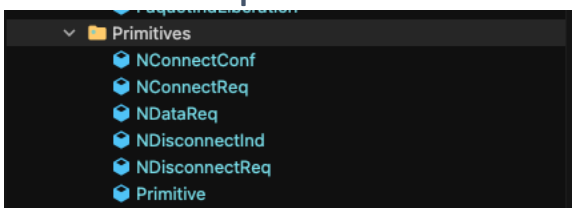
Le programme commence par L'ouverture de l'interface utilisateur

```java
import Couche.CTransport.TransportCouche;
import ViewInterface.guix;
Gansonre Ismael
public class Main {
    Gansonre Ismael
    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(() -> {

            new guix().setVisible(true);
            TransportCouche transport = new TransportCouche();
            transport.DemarrerCommunication();
        });


    }
}
```

- Classe de paquet

Tous les types de paquet sont représentés par une classe distincte qui hérite de la classe générale Paquet se trouvent dans le dossier Paquet

- Classe des primitives



La classe Primitives contient toutes les constantes nécessaires au bon fonctionnement du programme et se trouvent dans le dossier Primitives

- Le Dossier CTransport

Il contient les classes nécessaires pour le transport

1) class AdresseGestionnaire :

Il permet la gestion des adresses class Communication

2) class Communication:

il représente une communication entre deux extrémité (les deux extrémités peuvent être déjà connecté ou non connecté)

- Le Dossier CReseau

1. class Connexion

Une classe qui présenter un connexion entre deux extrémité p(s) et p(r) sont les numéros de séquence des paquets représentant respectivement le numéro ici, je déclare les numéro de séquence en type décimal, Et je vais les convertir en binaire quand je construire les paquets, du paquet envoyé et le numéro du prochain paquet attendu en réception.

2. Classe Réseaux

Une classe qui présente la couche Réseaux,

Une Liste de contrôlé de couche Réseaux

La méthode qui permet de lire les primitive envoyé par la couche transport :

lire_de_transport si c'est un primitive *NConnectReq*

si c'est un primitive *NDataReq* lancer le processus de transfert des données si c'est un primitive *NDisconnect.Req* lancer le processus de transfert des données si c'est un primitive *NDisconnectReq* si c'est un primitive *NDisconnect.Req* envoyer le paquet Indication Libération avec le message suivant : "ET terminer la liaison de donne

Une méthode qui envoie les primitive et id vers la couche transport par *ecrire_vers_transport*

le Fournisseur réseau peut refuser ou accepter le demande de connexion de la couche transport

ce processus est aléatoire par la méthode :

decisionAleatoire. Le Fournisseur réseau peut refuser ou accepter la demande de connexion de la couche transport.

Si l’adresse de la station source est un multiple de 27, le fournisseur refuse la sinon accepter , et envoyer le paquet appel , créer un nouveau Object Connexion, et l'ajouter dans la table de contrôlé

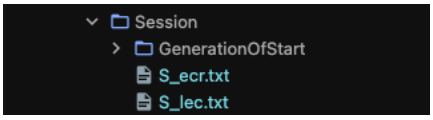
envoyer le paquet appel, en utilisant le service liaison donnée(écriture dans le fichier L_ecr)

une méthode qui gérer les transfert des données gestionTransfertDonnee une méthode qui envoyer un paquet en utilisant le service de liaison de données envoyerPaquet. une méthode qui permet de recevoir le paquet de distant(en simulant par une processus

une méthode qui permet de trouver un Connexion par numéro de connexion.

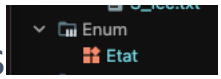
- Le Dossier Liaison de données
  classe qui présenter la couche de liaison de donnée

- Le Dossier Session



GenerationOfStart permet de générer les donnes aleatoires pour tester l’application

- Le Dossier Enum

Il contient les États 

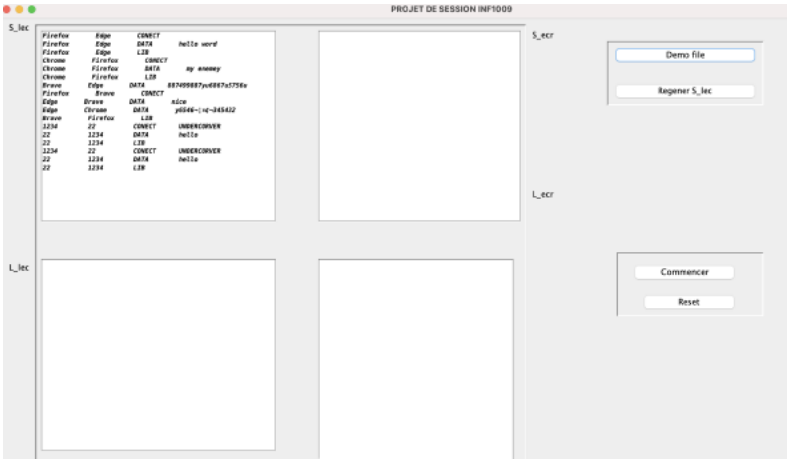
- Le Dossier ViewInterface

Contient la classe liée à l'interface

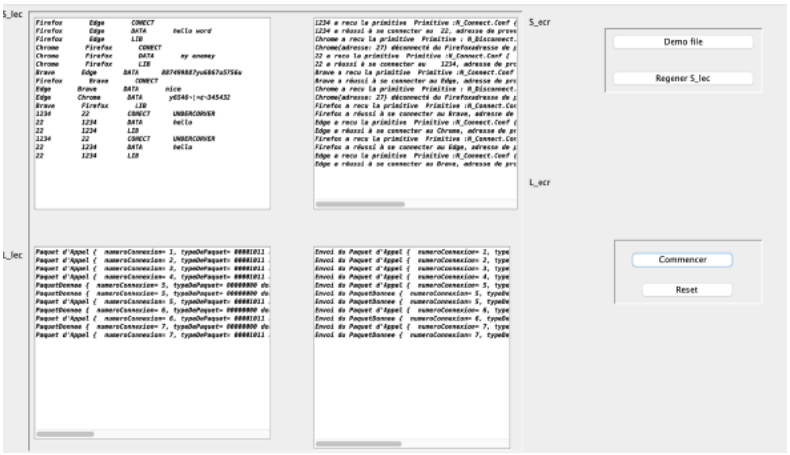
7) Mode d'emploi

L'application se lance a partir du fichier main :

Appuyez sur démo file qui vas charger dans la vue du fichier démo qui est S_lec



Et ensuite appuyez sur commencer pour voir la suite :



Le bouton « reset » permet d’effacer toute les vues et éventuellement les fichiers Le bouton « Régénérer » permet de Générer de nouvelles données S_lec

Cette section permet d’ouvrir dans l’éditeur de texte et d’afficher les traces