Université Constantine 2 Faculté des NTIC Département d'Informatique Fondamentale et ses Applications	1ère Année Master Réseaux et Systèmes Distribués Module : Algorithmes Distribués (ALDI) Année 2021/2022

TP N°5 : Implémentation du 2ème algorithme distribué en utilisant la plateforme JADE

Enoncé :

Soit un système distribué composé de **N sites** reliés par un anneau unidirectionnel. On suppose que les liaisons sont fiables et chaque **site** connait seulement le nom de son **successeur**.

On considère l'algorithme de **Le Lann** qui permet de résoudre le problème de la section critique. Ce dernier est basé sur l'existence d'un message appelé **Jeton** qui est en mouvement circulaire sur l'anneau. Le principe de fonctionnement de cet algorithme est le suivant :

- Initialement, le **Jeton** est placé sur un **site** et tous les **sites** sont dans l'état **dehors**.
- Lorsqu'un site **Pi** désire utiliser la section critique (**SC**), il passe à l'état **demandeur** et attend l'arrivée du **Jeton**.

Lorsque le **Jeton** arrive au **site Pi** alors **Pi** passe à l'état **dedans** et accède à la **SC**.

- Au bout d'un temps fini, **Pi** libère la **SC** en passant vers l'état **dehors** et en envoyant le **Jeton** à son **successeur**.
- Lorsqu'un **site** reçoit le **Jeton** et il est dans l'état **dehors** alors il envoie le **Jeton** à son **successeur**.

Variables locales pour un processus Pi :

- **suivant** : Nom du site successeur dans l'anneau;
- **état = {dehors, demandeur, dedans}** initialisé à **dehors**;
- **jetonPrésent** : booléen initialisé à **faux** sauf sur le site **j** sur lequel est initialement placé le jeton;

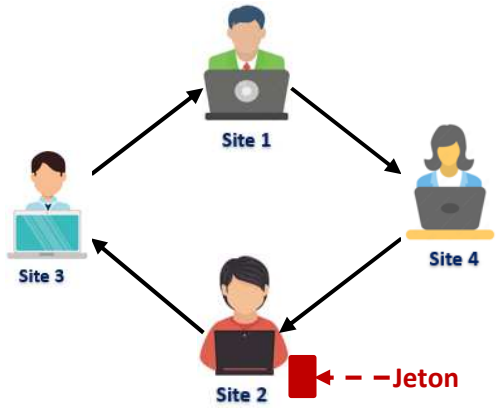
Procédures du processus Pi :

Lors d'un appel à acquérir
état = demandeur;
Attendre (jetonPrésent == vrai);
état = dedans;

Lors d'un appel à libérer
état = dehors;
jetonPrésent = faux;
Envoyer ***Jeton*** au **suivant**;

Lors de la réception du *jeton*
Si (état == dehors) **alors**
Envoyer ***Jeton*** au **suivant**;
Sinon
jetonPrésent = vrai;
FinSi

Exemple d'un anneau composé de 4 sites



Questions :

1. Tracer un graphe qui représente les comportements de chaque **site** et les relations qui existent entre les différents comportements
2. Quel(s) est(sont) le(s) message(s) échangé(s) entre les **sites** ? Donner leur contenu.
3. Quels sont les arguments qui seront passés à chaque **site** ? Donner leur signification.
4. Implémenter la classe **site** (la méthode setup et les différents comportements).
5. Exécuter le programme en utilisant 3 **sites**.

Extrait du code du TP n°4 :

```
public class Site extends Agent{
  String nomControleur;
  ……
  public void setup(){
    System.out.println("Agent " + getLocalName());
    Object [] args = getArguments();
    if (args != null) {
      nomControleur = args[……].toString();
      ……
    }
    FSMBehaviour fsm = new FSMBehaviour(this);
    fsm.registerFirstState(new liberer(), "dehors");
    fsm.registerState(new acquerir(), "demandeur");
    ……
    fsm.registerTransition("dehors", "dehors", 0);
    ……
    fsm.registerDefaultTransition("dedans", "dehors");
    ParallelBehaviour parallel = new ParallelBehaviour(ParallelBehaviour.WHEN_ALL);
    parallel.addSubBehaviour(fsm);
    ……
    addBehaviour(parallel);
  }
  public class acquerir extends OneShotBehaviour {
    public void action(){
      ……
      ACLMessage msgEnvoi  = new ACLMessage (ACLMessage.INFORM);
      msgEnvoi.addReceiver(new AID(……, AID.ISLOCALNAME));
      msgEnvoi.setContent(……);
      send(msgEnvoi);
    }
    public int onEnd(){
      ……
    }
  }
  ……
  public class consulterBoite extends CyclicBehaviour {
    public void action(){
      ACLMessage msgRecu = receive();
      if (msgRecu != null){
        String msgContenu = msgRecu.getContent();
        String nomEmetteur = msgRecu.getSender().getLocalName() ;
        ……
      }
    }
  }
}
```

Autres fonctions que vous pouvez utiliser en cas de besoin :

```
String chaine = ……;
String [] tab = chaine.split("#");
```

```
int nb = ……;
int val = (int) (Math.random() * nb);
```

```
String st = ……;
int val = Integer.parseInt(st);
```

```
public class Test {
  public static void main(String[] args) {
    String [] commande = new String[3];
    String argument = "";
    argument = argument+"s1:site(……)";
    ……
    commande[0] = "-cp";
    commande[1] = "jade.boot";
    commande[2] = argument;
    jade.Boot.main(commande);
  }
}
```