+8/1/18+

Université d'Évry Val d'Essonne
Master 1 MIAGE App
2021-2022

Partiel de Systèmes et Applications Répartis (SAR)

Durée : 2h30

1. Inscrivez ci-contre votre numéro d'étudiant, en écrivant un chiffre par case.
2. Reportez un chiffre / colonne en **noircissant** la case correspondante.

2	0	1	7	5	2	3	3

Répondez directement sur l'énoncé et rendez le. **Lisez attentivement l'énoncé.** Les documents, les calculatrices et les téléphones mobiles sont interdits. Toutes les questions comportent *au moins* une bonne réponse et *au moins* une mauvaise réponse. Il est préférable de **noircir** les cases en les remplissant plutôt que de simplement les cocher. Il n'est pas gênant de déborder. Vous pouvez utiliser du correcteur blanc pour effacer une réponse, il n'est pas gênant alors de masquer complètement la case. Utilisez un stylo à encre, de préférence **noire**. Le code +8/p/x+ en haut de page indique qu'il s'agit de la page p du sujet numéro 8. Vérifiez qu'on vous a bien distribué les 6 pages de votre sujet numéro 8.

1 Question de cours (2pts)

Question 1 ♣ Un système distribué est un :

- ☐ système avec une architecture producteur/consommateur
- ☐ système réactif
- ☐ système dont les données sont distribuées
- ☐ système avec une architecture client/serveur
- ■ système dont les traitements sont distribués
- ☐ système monolithique
- ☐ ensemble de processus complètement indépendants
- ■ ensemble de processus coopératifs
- ☐ système temps réel

2 Processus et Synchronisation (6pts)

Soit un système distribué composé de trois partenaires A, B et C. Chaque partenaire utilise des canaux de communication synchrones pour se synchroniser et communiquer avec les autres partenaires selon le protocole défini par le LTS de la figure 1. Chaque partenaire est représenté par un processus chargé d'appliquer le protocole d'interaction décrit par le LTS de la figure 1. Les processus sont itératifs. Nous considérons dans cette partie que les partenaires A, B et C partagent le même espace mémoire et ont accès aux mêmes canaux.

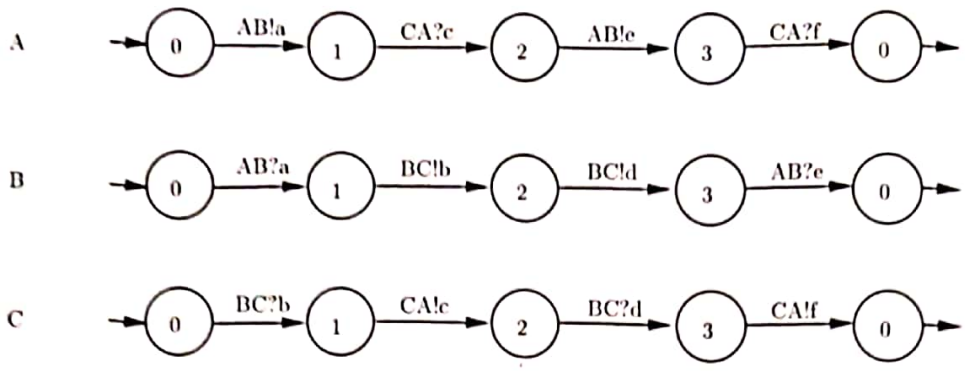

Figure 1: Système de Transition Labellisé des 3 partenaires.

Question 2 ♣ Parmi les états suivants, quels sont ceux qui appartiennent à l'ensemble des état du système distribué décrit ci-dessus ? Un état du système = (état de A, état de B, état de C)

- ☐ (3, 3, 3)
- ☐ (3, 3, 2)
- ☐ (4, 3, 4)
- ☐ (1, 1, 0)
- ☐ (2, 1, 2)
- ☐ (1, 2, 1)
- ☐ (2, 3, 3)
- ☐ (1, 0, 1)
- ☐ (2, 3, 2)
- ☐ (1, 2, 2)
- ☐ (2, 2, 2)
- ☐ (3, 4, 3)

Question 3 Complétez le code ci-dessous de la classe des canaux de communication synchrones qui permettent l'envoi et la réception de messages de type chaîne de caractères.

☐z ☐a ☐b ☐c ☐d *Cadre reservé*

```
public class Canal <String>
{
        private String message=null;

        public void envoyer(String m){
            Synchronized


             message = m;

              NotifyAll();
              wait();

        }

            Synchronized
        public void recevoir(String s){

            if(message==null) wait();
            String tmp =message;
              message = null;

             return(tmp);


        }
}
```

Question 4 Complétez le code ci-dessous de la classe du processus du partenaire C.

☐z ☐a ☐b ☐c ☐d *Cadre reservé*

```
1  public class ProcessusC
2  {
3       private  Canal BC;
4       private Canal CA;
5
6                 public ProcessusC(Canal x, Canal y){
7
8                     this.BC=x;
9                     this.CA=y;
10                }
11
12
13                    public void start()throws RemoteException, InterruptedException {
14                    BC.envoyer();
15                    CA.envoyer();
16                    CA.recevoir(); }
17
18
19
20
21
22
23
24
25
26
27
28
29
30 }
```

3 RMI (7pts)

Nous considérons dans cette partie que les trois partenaires A, B et C sont distribués sur trois sites différents. Nous voulons toujours réaliser le système distribué représenté par la figure 1 en utilisant les canaux de communication comme des objets distribués distants à travers RMI. Chaque partenaire met en place un seul canal de communication sur son site : le canal AB sur le site de A, le canal BC sur le site de B et le canal CA sur le site de C.

Question 5 Complétez l'interface *ICanal* des canaux de communication synchrones des partenaires *A*, *B* et *C*.

z a b c d *Cadre reservé*

```
1  public interface ICanal extends Remote
2  {
3
4
5
6
7
8
9
10         void AB() throws RemoteException;
11         void recevoirAB() throws RemoteException, InterruptedException;
12         void BC() throws RemoteException;
13         void recevoirBC() throws RemoteException, InterruptedException;
14         void CA() throws RemoteException;
           void recevoirCA() throws RemoteException, InterruptedException;
   }
```

Question 6 Complétez la classe *CanalImpl* qui implémente la classe des canaux de communication synchrones sur le site A.

z a b c d e f *Cadre reservé*

```
1  public class CanalImpl extends UnicastRemoteObject implement canalA
2  {
3    private boolen recevoirAB;
4
5    public class canalimpl() throws Remote Exception{
6
7          recevoirAB= false;
8    }
9
10   public Synchronized void AB()Remote Exception{
11         recevoirAB= true;
12         notifyAll();
13   }
14   public Synchronized void recevoirAB() Remote Exception, Interrupted Exception{
15         while(!recevoirAb){
16         waite(); }
17   }}
18
19
20
21
22
23 }
```

Question 7 ♣ On souhaite écrire le serveur du partenaire *A*. Sélectionnez les bonnes instructions qui permettent de réaliser le serveur du partenaire *A*.

- [x] `Naming.bind("//localhost/AB", AB);`
- [ ] `(new ProcessusA(AB, BC)).run();`
- [ ] `AB=(ICanal)registry.lookup("AB");`
- [ ] `new ProcessusA(BC, CA)).start();`
- [x] `ICanal AB = new CanalImpl();`
- [ ] `registry.rebind("CA", CA);`
- [ ] `ICanal BC = new CanalImpl();`
- [ ] `Naming.bind("//localhost/BC", BC);`
- [ ] `ICanal CA = new CanalImpl();`
- [ ] `BC=(ICanal)registry.lookup("BC");`
- [ ] `CA=(ICanal)Naming.lookup("//localhost/CA");`
- [ ] `(new ProcessusA(AB, CA)).go();`

4 gRPC (5pts)

Les trois partenaires *A*, *B* et *C* sont distribués sur trois sites différents et ne sont pas implémentés avec le même langage de programmation. Seul le partenaire *B* utilise le langage Java comme langage de programmation. Nous voulons réaliser le système représenté par la figure 1 en utilisant les canaux distribués distants à travers gRPC.

Question 8 Complétez la description Protocol Buffers (.proto) des opérations des canaux de communication synchrones.

☐z ☐a ☐b ☐c ☐d ☐e ☐f *Cadre reservé*

```
1  syntax="proto3";
2  option java_multiple_files=true;
3  option java_package="gRPC";
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Question 9 Complétez l'implémentation de la classe des canaux de communication (Servants) du partenaire *B*. Respectez les noms des classes, des interfaces et des méthodes.

☐z ☐a ☐b ☐c ☐d *Cadre reservé*

```
public class CanalImpl extends CanalImplBase
{
}
```

```
public interface StreamObserver<V>{
    void onNext(V value) ;
    void onCompleted();
    void onError();
}

Classe o = Classe.newBuilder().setAttribut(val).build();
```

Figure 2: Annexe gRPC.