Master 1 Informatique
Programmation Répartie et Architecture N Tiers

Examen – Session 1 – mercredi 6 avril 2011

Denis Caromel, Brian Amedro

Université de Nice - Sophia Antipolis, Département Informatique

---

Tous les documents sont autorisés
Prenez le temps de bien lire et de bien comprendre les questions !
Écrivez *lisiblement* en notant *clairement* sur votre feuille les numéros des questions (exemple : 2. 3.).
Durée : 2H00

1 (1 point)

Expliquez Distribution (Répartition) de Situation et Distribution (Répartition) de Résolution.

2 (1 point)

Que signifie DSM (Distributed Shared Memory) ?

3 (2 points)

Dans un programme réparti, lorsque nous devons rechercher et nous connecter sur un objet distant, quels sont les catégories de mécanismes dont nous disposons ?

4 (2 points)

Dans le cadre des programmes répartis, nous qualifions du terme général "proxy" un représentant local d'un objet distant. Cependant, celui-ci réalise au moins 3 fonctionnalités différentes, quelquefois implémentées par 3 objets différents. Donnez le principe de ces 3 fonctionnalités.

5 (2 points)

Dans le cadre de ProActive, lorsque l'on exécute le code suivant :

```
    V1 = ao1.foo ();
V2 = ao2.foo ();
V2 = V1;
V3 = ao3.foo (V1);
V3 = V1;
```

```
V1.print ();
```

Que va-t-il se passer ?
Quels sont les instructions où il risque de se produire une attente ?
Quelle valeur de V1 sera imprimée ?

6 (2 points)

Expliquer les principales différences entre une programmation par objets répartis et par Composants, en particulier au niveau des dépendances, du déploiement, et du cycle de vie.

7 (3 points)

Voici une classe `TestImpl` qui utilise RMI pour exposer une méthode `foo()` sur la machine A :

```
public class TestImpl implements Test {
    public void foo(char c1, char c2 ) {
      System.out.print(c1);
      Thread.sleep(5000); // Pause de 5 secondes
      System.out.print(c2);
    }
}
```

Voici maintenant une class `Main`, exécutée sur la machine B :

```
public class Main {
    public static void main(String[] args) {
      Registry registry = LocateRegistry.getRegistry("A");
      Test remoteObject = (Test) registry.lookup("Test");
      long start = System.currentTimeMillis(); // Récupère le nombre de millisecondes écoulées
depuis le 1/1/1970
      remoteObject.foo('a', 'b');
      remoteObject.foo('c', 'd');
      long end = System.currentTimeMillis();
      System.out.println(" Elapsed time = " + (end-start));
    }
}
```

Tout est correctement configuré et il n'y a pas d'erreur.

1. Quel sera l'affichage sur la console de la machine A (où se trouve l'objet Test)
2. Quel sera l'affichage sur la console de la machine B (où exécutée la classe Main)

8 (3 points)

La méthode suivante est définie dans un objet accessible à distance :

```
public void compute(int val) {
    this.value += val;
    this.value *= val;
}
```

La valeur initiale de la variable d'instance `value` est 5. Deux clients invoquent alors la méthode `compute()` en même temps. Le premier avec pour paramètre 2, et le second avec pour paramètre

3.
Quel sera le contenu de la variable `value`, après les deux appels à `compute()` :
- avec ProActive ?
- avec RMI ?
(Dans les deux cas, justifiez votre réponse).

9 (5 points)

Nous voulons distribuer un test de primalité avec RMI. Ce test sera effectué sur l'intervalle $\left[2; \left\lceil \sqrt{n} \right\rceil\right]$. La distribution consistera à tester en parallèle, depuis une machine M, des sous-intervalles différents sur 3 machines : W_1, W_2 et W_3. Pour ce faire, nous utiliserons un test très simple :

```
isPrime(long candidate, long begin, long end) {
    for (long divider = begin; divider < end; divider++) {
       if ((candidate % divider) == 0) {
          return false;
       }
    }
    return true;
}
```

Questions
Vous devez fournir le code de 3 fichiers :

- une interface et son implémentation qui expose la méthode $isPrime()$ permettant de vérifier la primalité d'un nombre sur intervalle donné.
- une classe $Main$ (avec une méthode $main$) qui prend en argument un nombre et en distribue le test de primalité sur les 3 machines.

Écrivez l'ensemble des commandes que vous lancerez sur les machines M, W_1, W_2 et W_3 pour lancer un test de primalité.