Université Paris Cité

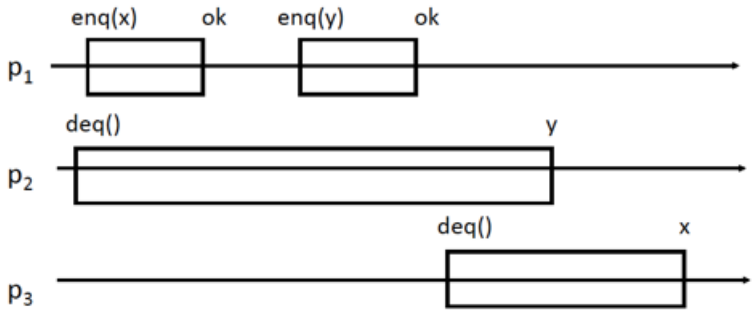
Master 2 - Année 2023-24

Programmation répartie

TP1-objet concurrent

Exercice 1.—

- L'histoire ci dessous est-elle linéarisable par rapport à la specification séquentielle d'une file



- p_2 et p_3 aurait-il pu obtenir une autre réponse à leur demande de *deq*?

Exercice 2.— On considère une pile concurrente (i.e qui peut être utilisée par plusieurs threads) d'entier.

1. En vous inspirant de la définition classique d'une pile donnez la définition et la spécification séquentielle de cet objet
2. On considère l'implémentation suivante

```
public class PileSync {
    int t[];
    int sommet=-1;
    PileSync(int n){
     t= new int[n];
    }

    synchronized void empiler(int j)
    {sommet=sommet+1;
    t[sommet]=j;
    }

    synchronized int depiler(){
        if (sommet==-1)return -1;
        sommet=sommet-1;
        return t[sommet+1];
    }
}
```

Que l'on peut utiliser avec :

```
public class AjoutThread2 extends Thread{
    public PileSync p;
    int id;
    public AjoutThread2 ( PileSync p, int id){
        this.p=p;
        this.id=id;
    }

    public void run(){
        for(int i=1;i<11;i++)
        {
          p.empiler(i+100*id);
          try{Thread.sleep(10);}
          catch(Exception e) { System.out.println( "probleme" );}
          this.yield();
          }

        for(int i=1;i<11;i++)
        {
          System.out.println( "Thread  "+id+"  retire " +p.depiler());
          try{Thread.sleep(10);}
          catch(Exception e) { System.out.println( "probleme" );}
          this.yield();
          }
    }
}
=====
public class Main2 {
     public static void main( String [] v)
    {
    Thread[] Th = new Thread[3];
    PileSync f=new PileSync(100);
    for (int i=0;i<3;i++)
    {
         Th[i]= new AjoutThread2(f,i);
         Th[i].start();
    }
   for (int i=0;i<3;i++)
   {    try{Th[i].join();}
         catch(Exception e) {System.out.println( "probleme" );}
   }
}
}
```

Cette implémentation réalise-t-elle une implémentation linéarisable d'une pile ? Si oui indiquer les points de linéarisation.

3. On supprime les synchronized, l'implémentation réalise-t-elle une implémentation linéarisable d'une pile dans les cas suivants: (Si non indiquez quelles propriétés sont perdues, si oui indiquez les points de linéarisation)

   (a) Il n'y a pas de concurrence entre les appels des threads.

(b) Quand il y a concurrence entre deux threads, c'est 2 threads qui appellent *empiler*

(c) Quand il y a concurrence entre deux threads, c'est 2 threads qui appellent *dépiler*

(d) Quand il y a concurrence entre deux threads, c'est une thread qui appelle *emplier* et l'autre *dépiler*