TÖL401G Assignment 13

Stefán Hermundsson, Sigríður Birna Matthíasdóttir

TOTAL POINTS

10 / 10

QUESTION 1

1 Assignment 13 **10 / 10**

✓ **+ 10 pts** *100% correct solution*

+ 2 pts For correct useToilet(): sem.init(1), wait & signal around useToilet()

+ 2 pts For correct breakfastFinishedToMother():

sem.init(0), 1 signal in child, 2 waits in mother

+ 2 pts For correct foodReady(): sem.init(0), 1 wait in child, 2 signal in mother

+ 2 pts For correct drinkReady(): sem.init(0), 1 wait in child, 2 signals in father

+ 2 pts For correct breakfastFinishedToFather(): sem.init(0), 1 signal in child, 2 waits in father

+ 6 pts Max 6. If solution is almost correct, but a deadlock may occur or wrong order. (E.g. using only one semaphore to signal from child to mother and father)

+ 2 pts For correct breakfastFinishedToMother() and correct breakfastFinishedToFather(): But sem.init(-1), 1 signal in child, 2 waits in father and sem.init(-1), 1 signal in child, 2 waits in mother is wrong

+ 1 pts Could work with six semaphores but wrong init value=2

+ 1 pts Six semaphores could work, but missing one breakfast rady.wait in child

- 1 pts Correct use of semaphores but forgot to declare semClear/drink(0) semaphore

- 2 pts Wrong use of S_CLEAN.signal between mother and father, also no waiting for father S_Clean signal at the end

- 1 pts Wrong use of sem.clean.signal() at the end of father

- 1 pts Missing one sem_Food.signal()/wait() or sem_DriveToschool.wait() in mother()

- 1 pts Missing one sem_Drink.signal()/wait() in father()

- 1 pts Wrong use of sem_food.wait()/sem_food.signal or sem drink_wait()&sem_dring.signl() in child

+ 4 pts Solution needs large modification

Háskóli Íslands
Vormisseri 2023
TÖV401G

Stefán Hermundsson
Sigríður Birna Matthíasdóttir

Heimaverkefni 13

```
// Definition of shared semaphores

sem toiletIsFree
sem foodReady
sem drinkReady
sem driveReady
sem cleanReady

//initilize semaphores
toiletIsFree.init(1) //Mututal exclusion
foodReady.init(0)    //Conditon
drinkReady.init(0)   //Conditon
driveReady.init(0)   //Conditon
cleanReady.init(0)   //Conditon

// Run 4 person family in parallel
parallel {child(), child(),
    mother(), father()}

child() {
    toiletIsFree.wait()    //check if toilet is free
    useToilet()
    toiletIsFree.signal()  //toilet is free now

    foodReady.wait()       //child waits for food ready condition
    drinkReady.wait()      //child waits for drink ready condition

    haveBreakfast()
    cleanReady.signal()    //child is done eating
    driveReady.signal()    //child is ready to go
}
```

```
mother() {
    toiletIsFree.wait()    //check if toilet is free
    useToilet()
    toiletIsFree.signal()  //toilet is free now

    prepareFood()
    foodReady.signal()     //food for first child is ready
    foodReady.signal()     //food for second child is ready

    driveReady.wait()      //first child is ready to go
    driveReady.wait()      //second child is ready to go

    takeAndDriveToSchool()
}

father() {
    toiletIsFree.wait()    //check if toilet is free
    useToilet()
    toiletIsFree.signal()  //toilet is free now

    prepareDrinks()
    drinkReady.signal()    //drink for first child is ready
    drinkReady.signal()    //drink for second child is ready

    cleanReady.wait()      //first child is done eating
    cleanReady.wait()      //second child is done eating

    clearTable()
}
```

Lausn forrituð í Java 😊:

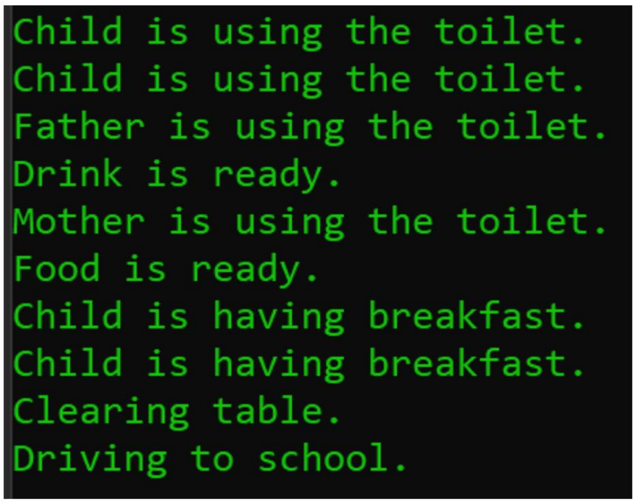
```
import java.util.concurrent.Semaphore;

public class Verkefni13 extends Thread {
    private Semaphore toiletIsFree = new Semaphore(1);
    private Semaphore foodReady = new Semaphore(0);
    private Semaphore drinkReady = new Semaphore(0);
    private Semaphore driveReady = new Semaphore(0);
    private Semaphore cleanReady = new Semaphore(0);

    public class Child extends Thread {
        public void run() {
            try {
                toiletIsFree.acquire();
                useToilet("Child");
                toiletIsFree.release();

                foodReady.acquire();
                drinkReady.acquire();

                haveBreakfast();
                cleanReady.release();
                driveReady.release();
            } catch (InterruptedException ie) {
                //Error
            }
        }
    }
```

```java
public class Mother extends Thread {
    public void run() {
        try {
            toiletIsFree.acquire();
            useToilet("Mother");
            toiletIsFree.release();

            prepareFood();
            foodReady.release();
            foodReady.release();

            driveReady.acquire();
            driveReady.acquire();

            takeAndDriveToSchool();
        } catch (InterruptedException ie) {
            //Error
        }
    }
}

public class Father extends Thread {
    public void run() {
        try {
            toiletIsFree.acquire();
            useToilet("Father");
            toiletIsFree.release();

            prepareDrinks();
            drinkReady.release();
            drinkReady.release();

            cleanReady.acquire();
            cleanReady.acquire();

            clearTable();
        } catch (InterruptedException ie) {
            //Error
        }
    }
}
```

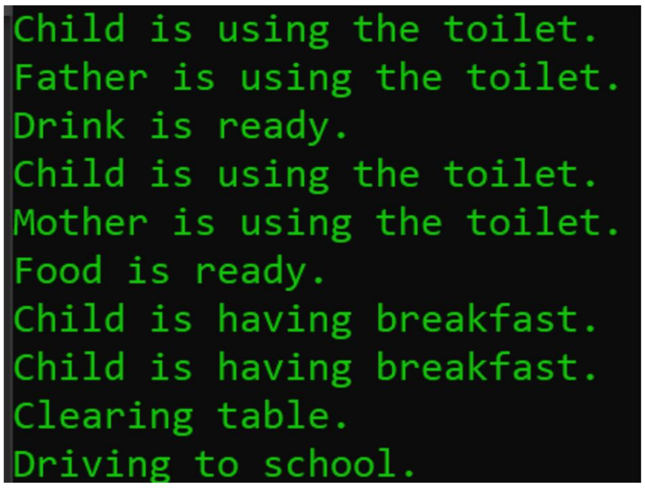
```java
    public static void main(String[] args) {
        Thread thread = new Verkefni13();
        thread.start();
    }

    public void run() {
        Thread child1 = new Child();
        Thread child2 = new Child();
        Thread mother = new Mother();
        Thread father = new Father();

        child1.start();
        child2.start();
        mother.start();
        father.start();
    }

    public void useToilet(String fMember) {
        System.out.println(fMember + " is using the
                                toilet.");
    }

    public void haveBreakfast() {
        System.out.println("Child is having breakfast.");
    }

    public void prepareFood() {
        System.out.println("Food is ready.");
    }

    public void prepareDrinks() {
        System.out.println("Drink is ready.");
    }

    public void takeAndDriveToSchool() {
        System.out.println("Driving to school.");
    }

    public void clearTable() {
        System.out.println("Clearing table.");
    }
}
```

Stefán Hermundsson
Sigríður Birna Matthíasdóttir

Dæmi þegar það er keyrt í cmd:



```
Child is using the toilet.
Child is using the toilet.
Father is using the toilet.
Drink is ready.
Mother is using the toilet.
Food is ready.
Child is having breakfast.
Child is having breakfast.
Clearing table.
Driving to school.
```



```
Child is using the toilet.
Father is using the toilet.
Drink is ready.
Child is using the toilet.
Mother is using the toilet.
Food is ready.
Child is having breakfast.
Child is having breakfast.
Clearing table.
Driving to school.
```

1 Assignment 13 10 / 10

✓ **+ 10 pts** *100% correct solution*

+ 2 pts For correct useToilet():
sem.init(1), wait & signal around useToilet()

+ 2 pts For correct breakfastFinishedToMother():
sem.init(0), 1 signal in child, 2 waits in mother

+ 2 pts For correct foodReady():
sem.init(0), 1 wait in child, 2 signal in mother

+ 2 pts For correct drinkReady():
sem.init(0), 1 wait in child, 2 signals in father

+ 2 pts For correct breakfastFinishedToFather():
sem.init(0), 1 signal in child, 2 waits in father

+ 6 pts Max 6. If solution is almost correct, but a deadlock may occur or wrong order. (E.g. using only one semaphore to signal from child to mother and father)

+ 2 pts For correct breakfastFinishedToMother() and correct breakfastFinishedToFather(): But sem.init(-1), 1 signal in child, 2 waits in father and sem.init(-1), 1 signal in child, 2 waits in mother is wrong

+ 1 pts Could work with six semaphores but wrong init value=2

+ 1 pts Six semaphores could work, but missing one breakfast rady.wait in child

- 1 pts Correct use of semaphores but forgot to declare semClear/drink(0) semaphore

- 2 pts Wrong use of S_CLEAN.signal between mother and father, also no waiting for father S_Clean signal at the end

- 1 pts Wrong use of sem.clean.signal() at the end of father

- 1 pts Missing one sem_Food.signal()/wait() or sem_DriveToschool.wait() in mother()

- 1 pts Missing one sem_Drink.signal()/wait() in father()

- 1 pts Wrong use of sem_food.wait()/sem_food.signal or sem drink_wait()&sem_dring.signl() in child

+ 4 pts Solution needs large modification