

ใบงานที่ 2

เรื่อง Shared Memory

จัดทำโดย

นายวรรต พรมอนันต์ รหัส 67543206020-9

ห้อง ENGCE125_SEC_1

เสนอ

อาจารย์นุรักษ์ ไชยศรี

ใบงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE125

Operating Systems

(ระบบปฏิบัติการ)

หลักสูตร วศ.บ. วิศวกรรมคอมพิวเตอร์

สาขาวิชาวิศวกรรมไฟฟ้า คณะ วิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 2 ปีการศึกษา 2568

Source code

```
package warehousesystems;

import java.util.LinkedList;

public class WareHouseSystems {

    public static void main(String[] args) {

        LinkedList<Integer> wareHouse = new LinkedList<>();

        final int capacity = 5 ;

        //Producer

        Thread t1 = new Thread(() -> {

            int producVal = 0 ;

            while (true) {

                synchronized (wareHouse) {

                    while(wareHouse.size() == capacity ){

                        try { //wait consumer item get out

                            System.out.println("Ware House is Full") ;

                            wareHouse.wait();

                        } catch (InterruptedException e) {

                            Thread.currentThread().interrupt();

                        }

                    }

                    producVal++ ;

                    wareHouse.add(producVal);

                    System.out.println("additem | Produc have " +
wareHouse.size()+" Item");

                    wareHouse.notifyAll(); //is not empty wakeUp

                }

                try {

                    Thread.sleep(2000);

                } catch (InterruptedException e) {
```

```

        Thread.currentThread().interrupt();
    }
}

});

// Consumer

Thread t2 = new Thread(() -> {
    while (true) {
        synchronized (wareHouse) { //wait produc insert item
            while (wareHouse.isEmpty()) { // if empty wait
                try {
                    System.out.println("wait produc insert item");
                    wareHouse.wait();
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt();
                }
            }

            wareHouse.removeFirst();

            System.out.println("out of Warehouse Produc have " +
wareHouse.size()+" Item");

            wareHouse.notifyAll(); //say i removed it
        }

        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }

});

t1.start();

```

```
        t2.start();
    }
}
```

Result

```
additem | Produc have 1 Item
out of Warehouse Produc have 0 Item
additem | Produc have 1 Item
additem | Produc have 2 Item
out of Warehouse Produc have 1 Item
additem | Produc have 2 Item
additem | Produc have 3 Item
out of Warehouse Produc have 2 Item
additem | Produc have 3 Item
additem | Produc have 4 Item
additem | Produc have 5 Item
out of Warehouse Produc have 4 Item
additem | Produc have 5 Item
Ware House is Full
out of Warehouse Produc have 4 Item
additem | Produc have 5 Item
Ware House is Full
out of Warehouse Produc have 4 Item
additem | Produc have 5 Item
Ware House is Full
out of Warehouse Produc have 4 Item
additem | Produc have 5 Item
Ware House is Full
```