Task 3

张配天-2018202180

1 5.6

1.1 a

X=7,Y=28

1.2 b

operation	value of X	value of y
LOAD R1,X	2	3
LOAD R2,Y	2	3
MUL R1,R2	2	3
STORE X,R1	6	3
INC R2	6	3
STORE Y,R2	6	4
LOAD R3,X	6	4
INC R3	6	4
LOAD R4,Y	6	4
MUL R4,R3	6	4
STORE X,R3	7	4
STORE Y,R4	7	28

1.3 c

可以用不同的调度使得 X 得到返回值 6,Y 返回值 9

operation	value of X	value of Y
LOAD R3,X	2	3
INC R3	2	3
LOAD R4,Y	2	3
MUL R4,R3	2	3
LOAD R1,X	2	3
LOAD R2,Y	2	3
MUL R1,R2	2	3
STORE X,R3	2	3
STORE X,R1	6	3
INT R2	6	3
STORE Y,R2	6	4
STORE Y,R4	6	9

2 5.26

我们设立 2 个全局变量, 用来记录精灵和驯鹿的个数:

- elves = 0, 精灵数
- deers = 0, 驯鹿数

我们设立 1 个私用信号量, 用来实现进程之间的同步:

- wakeup = 0, 代表圣诞老人是否醒来
- solved = 0, 代表精灵问题被解决

和 4 个公用信号量, 用来实现进程之间的互斥:

- elfmutex = 1, 用于精灵出问题的互斥锁
- elfproblem = 1, 用于三个精灵出问题后的互斥
- deermutex = 1, 用于驯鹿回北极的互斥锁

它们用来 3 个进程 (不分顺序):

- 圣诞老人进程 Santa_Claus()
- 小精灵进程 Elf()
- 驯鹿进程 Reindeer()

这些进程中有函数:

- 睡觉 sleep()
- 起床 wakeup()
- 组装雪橇 sleigh()

- 解决问题 solve()
- 做玩具 toy()
- 遇到问题 problem()
- 出精灵洞 leave()
- 从热带回来 back()
- 到达北极 arrive()

抽象代码如下:

```
Santa\_Claus(){
    sleep();
    semWait(wakeup);
    if(deers == 9){
        sleigh()
        deers = 0;
    }
    else if(elves == 3){
        solve();
        semSignal(solved);
    }
}
Elf(){
    toy();
    if(problem()){
        semWait(elfproblem);
        semWait(elfmutex);
        elves++;
        if(elves == 3){
            semSignal(wakeup);
            leave()
        }
        else{
            semSignal(elfproblem)
        }
    }
    semWait(solved);
    semWait(elfmutex);
    elves--;
    if(elves == 0){
        semSignal(elfproblem);
    }
```

```
33      }
34      Reindeer(){
35          semWait(deermutex);
36          if(back()){
37              deers++;
38              if(deers == 9){
39                  semSignal(wakeup);
40              }
41          }
42      }
```

3 5.28

若有多个读者, 有可能导致写入进程饿死。