Week13 Report

姓名：Yitong WANG(王奕童) 11910104@mail.sustech.edu.cn

学号：11910104

实验课时段：周五5-6节

实验课教师：Yun SHEN(沈昀) sheny@mail.sustech.edu.cn

实验课SA:

- Yining TANG(汤怡宁) 11811237@mail.sustech.edu.cn
- Yushan WANG(王宇杉) 11813002@mail.sustech.edu.cn

Q1. local_intr_save(intr_flag);

如课件中代码所示：

```
......
local_intr_save(intr_flag);
{
    临界区代码
}
local_intr_restore(intr_flag);
......
```

可见 `local_intr_save(intr_flag);` 是和 `local_intr_restore(intr_flag);` 一起避免在进程切换过程中处理中断。

Q2. Philosopher problem

(1) 可避免死锁。

原因：可以通过 `sem_init(&s[i], 1);` 确保只有1个哲学家能够拿起筷子，从而将多个线程的执行退化为单线程情况，以避免死锁。

当唯一执行的哲学家将锁释放后，剩余的哲学家尝试同时竞争锁，拿到锁的哲学家分为两种情况：

- 与释放锁的哲学家相邻：这样就会进入等待状态，直到哲学家用餐完毕才继续执行

- 与释放锁的哲学家不相邻：无需等待，直接释放锁即可。

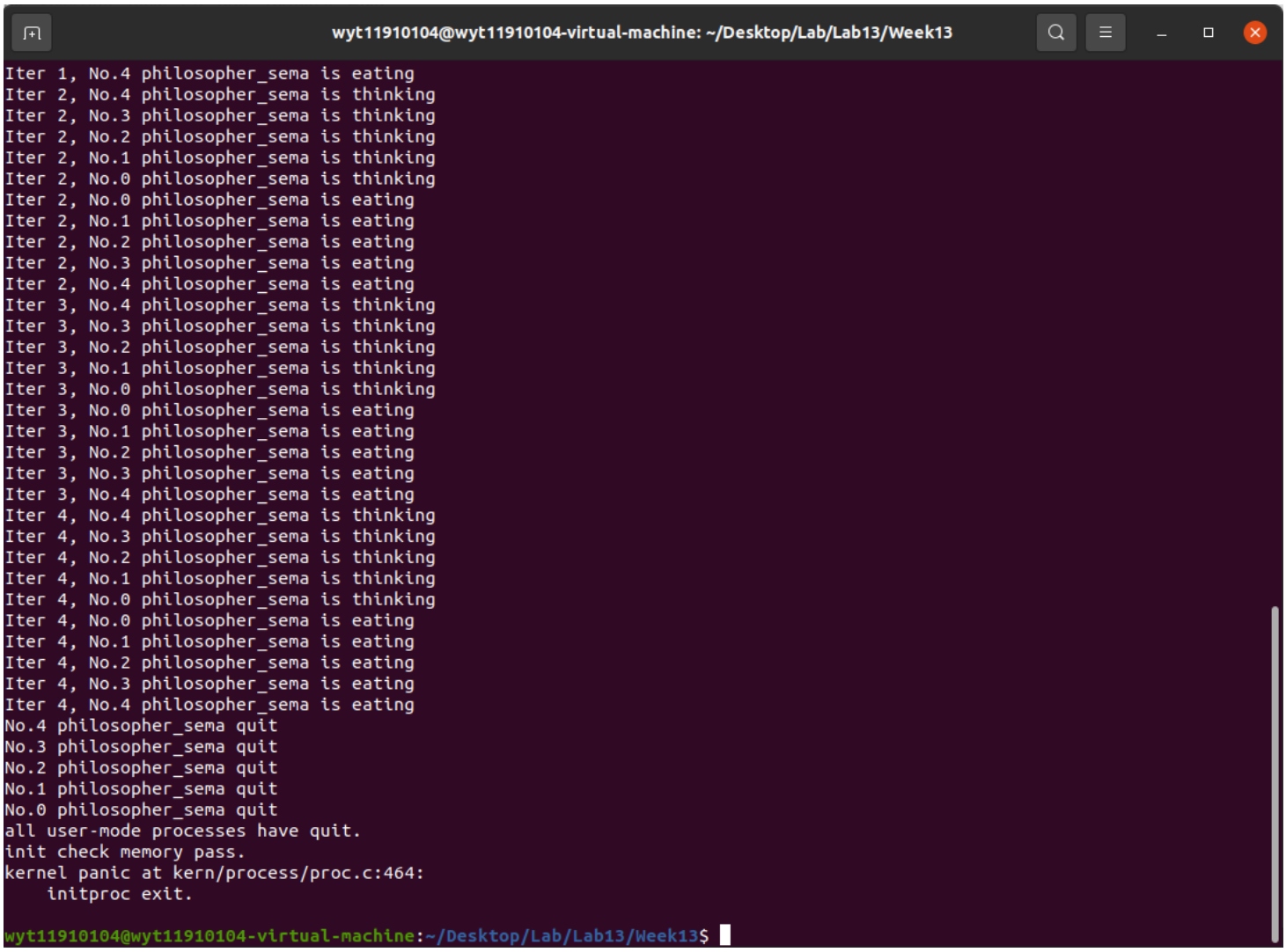
（2）

代码截图：

```
int left(int i){
    return (i - 1 + 5) % 5;
}

int right(int i){
    return (i + 1) % 5;
}

int isRange(int x){
    return 0 <= x && x < 5;
}

void phi_test_sema(int i)
{
    if(!isRange(i)){
        return;
    }
    if(state_sema[i] == HUNGRY){
        if(state_sema[left(i)] != EATING && state_sema[right(i)] != EATING){
            up(&s[i]);
        }
    }

}
```

```
51  void phi_take_forks_sema(int i)
52  {
53      if(!isRange(i)){
54          return;
55      }
56      down(&mutex);
57      state_sema[i] = HUNGRY;
58      phi_test_sema(i);
59      up(&mutex);
60      down(&s[i]);
61  }
62
63  void phi_put_forks_sema(int i)
64  {
65      if(!isRange(i)){
66          return;
67      }
68      down(&mutex);
69      state_sema[i] = THINKING;
70      phi_test_sema(right(i));
71      phi_test_sema(left(i));
72      up(&mutex);
73
74  }
75  //---------------end-----------------
76
```

运行截图：


wyt11910104@wyt11910104-virtual-machine: ~/Desktop/Lab/Lab13/Week13
Iter 1, No.4 philosopher_sema is eating
Iter 2, No.4 philosopher_sema is thinking
Iter 2, No.3 philosopher_sema is thinking
Iter 2, No.2 philosopher_sema is thinking
Iter 2, No.1 philosopher_sema is thinking
Iter 2, No.0 philosopher_sema is thinking
Iter 2, No.0 philosopher_sema is eating
Iter 2, No.1 philosopher_sema is eating
Iter 2, No.2 philosopher_sema is eating
Iter 2, No.3 philosopher_sema is eating
Iter 2, No.4 philosopher_sema is eating
Iter 3, No.4 philosopher_sema is thinking
Iter 3, No.3 philosopher_sema is thinking
Iter 3, No.2 philosopher_sema is thinking
Iter 3, No.1 philosopher_sema is thinking
Iter 3, No.0 philosopher_sema is thinking
Iter 3, No.0 philosopher_sema is eating
Iter 3, No.1 philosopher_sema is eating
Iter 3, No.2 philosopher_sema is eating
Iter 3, No.3 philosopher_sema is eating
Iter 3, No.4 philosopher_sema is eating
Iter 4, No.4 philosopher_sema is thinking
Iter 4, No.3 philosopher_sema is thinking
Iter 4, No.2 philosopher_sema is thinking
Iter 4, No.1 philosopher_sema is thinking
Iter 4, No.0 philosopher_sema is thinking
Iter 4, No.0 philosopher_sema is eating
Iter 4, No.1 philosopher_sema is eating
Iter 4, No.2 philosopher_sema is eating
Iter 4, No.3 philosopher_sema is eating
Iter 4, No.4 philosopher_sema is eating
No.4 philosopher_sema quit
No.3 philosopher_sema quit
No.2 philosopher_sema quit
No.1 philosopher_sema quit
No.0 philosopher_sema quit
all user-mode processes have quit.
init check memory pass.
kernel panic at kern/process/proc.c:464:
initproc exit.
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab13/Week13$