

Project 2 Report

Name: chen yongjun Student id: 522031910203

1 lcm

i use four sem to control larry curly moe's action:

```
sem_t shovel;           // 铲子信号量
sem_t hole_to_seed;     // 未种的坑
sem_t hole_to_fill;     // 种了但未填坑
sem_t unfilled_avalid;  // 所有未填的坑
```

Figure 1: sems

larry: wait(unfilledavalid); wait(shovel); post(shovel); post(holetoseed);

moe: wait(holetoseed); post(holetofill);

curly: wait(holetofill); wait(shovel); post(shovel); post(unfilledavalid);

2 party

2sem,1mutex control dean and student

```
sem_t studentnum; //0
sem_t waitingstudent;
pthread_mutex_t room_mutex = PTHREAD_MUTEX_INITIALIZER;
```

Figure 2: party sem

dean will repeatly check studentnum, and take action according it. to avoid dean repeatly take the lock, use semgetalve funtiion to check student num, and get lock only when it is 0 or 10;

```
void *dean() {
    for (int i = 0; i < N; i++) {
        usleep((rand() % 500 + 200) * 1000);
        int stnum=0;
        while(true)
        {
            sem_getvalue(&studentnum,&stnum);
            if(stnum>10||stnum==0)
            {
                pthread_mutex_lock(&room_mutex);
                sem_getvalue(&studentnum,&stnum);
                if(stnum==0)
                {
                    dean_enter();
                    search();
                    dean_leave();
                    pthread_mutex_unlock(&room_mutex);
                    break;
                }
                else if(stnum>10)
                {
                    dean_enter();
                    breakup();
                    while(stnum!=0)
                    {
                        sem_getvalue(&studentnum,&stnum);
                    }
                    dean_leave();
                    pthread_mutex_unlock(&room_mutex);
                    break;
                }
            }
            else
            {
                pthread_mutex_unlock(&room_mutex);
                continue;
            }
        }
    }
}
```

Figure 3: dean

student will check dean, if it is not in room, student will enter randomly. use sleep to solve the problem student num can't reach 10; to avoid student continually have lock, student have another queue to wait waitingstudent sem, only one student can wait mutex together with dean, so that dean can get in room immediately.

```
void *student(void *arg) {
    int id = *(int *)arg;
    sem_wait(&waitingstudent);
    pthread_mutex_lock(&room_mutex);
    student_enter(id);
    sem_post(&studentnum);
    //printf("student in the room:%d\n",++stnum);
    pthread_mutex_unlock(&room_mutex);
    sem_post(&waitingstudent);
    usleep((rand() % 500 + 1000) * 1000);
    party(id);
    usleep((rand() % 500 + 1000) * 1000);
    student_leave(id);
    //printf("student in the room:%d\n",--stnum);
    sem_wait(&studentnum);
    // some code goes here
    pthread_exit(0);
}
```

Figure 4: student