Project 6　The Sleeping Teaching Assistant

程式說明

學生的thread

```
void* stu_process(void* data)
{
        int id = *(int*) data;

        while(1)
        {
                random_delay();
                pthread_mutex_lock(&mutex);

                if(count < SEAT_AMOUNT && !stuStatus[id-1])
                {
                        seat[next_seat] = id;
                        count++;
                        printf("student %d is waiting now!\n",id);
                        next_seat = (next_seat + 1) % SEAT_AMOUNT;

                        pthread_mutex_unlock(&mutex);

                        sem_post(&sem_stu);
                        sem_wait(&sem_ta);
                }
                else if(stuStatus[id-1])
                {
                        pthread_mutex_unlock(&mutex);
                        printf("student %d is going home!\n",id);
                        pthread_exit(NULL);
                }
                else
                {
                        pthread_mutex_unlock(&mutex);
                        printf("There's no available seat now for student %d!\
n",id);
                }
        }
}
```

在學生的**thread**執行時，會先隨機延遲一段時間，然後再去搶**mutex**，搶到之後有**3**個判斷式，分別是等待區有位子、該學生結束輔導可以回家以及等待區沒有位子。

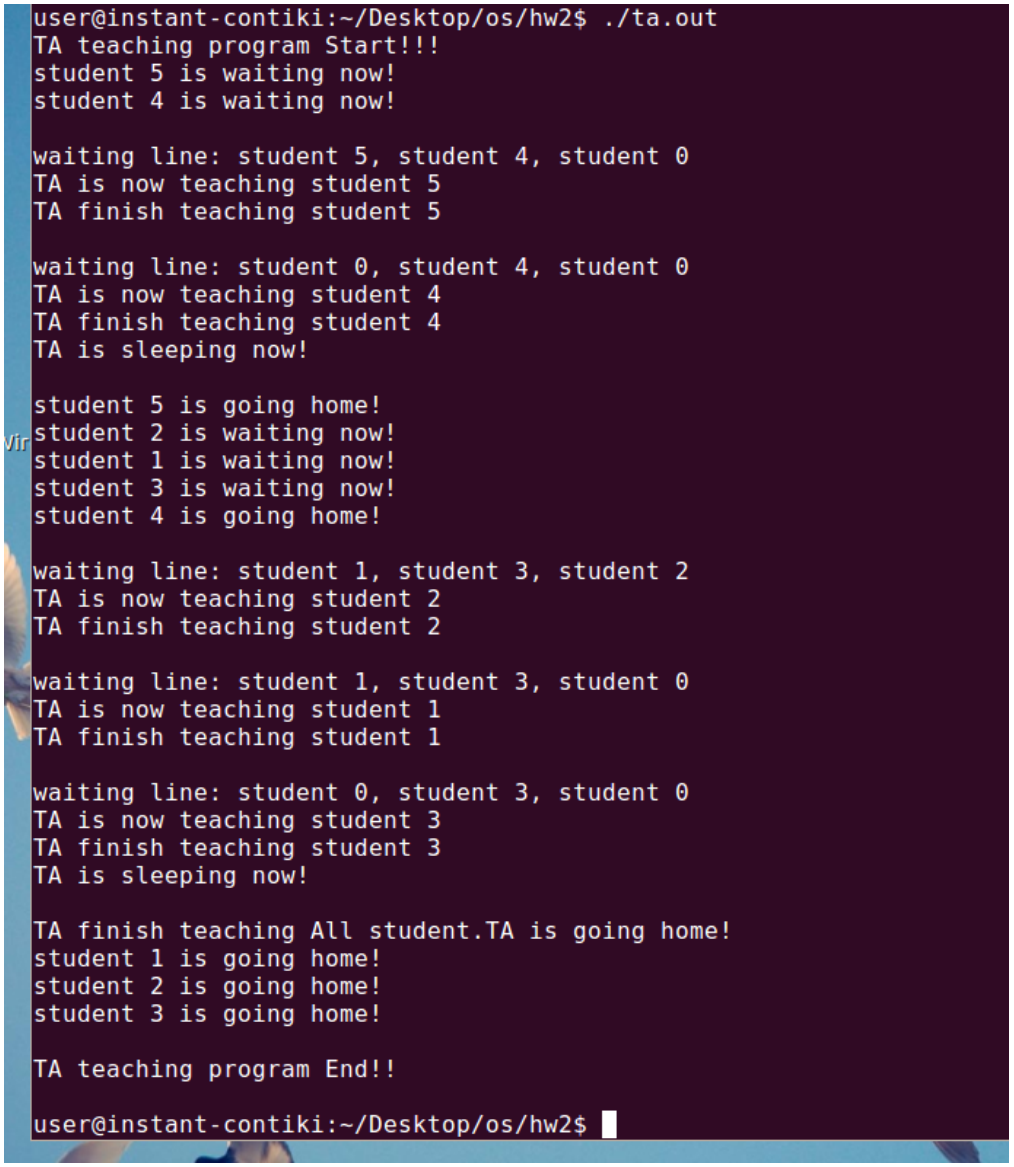
原本好像沒位子的時候就會讓學生回家，不過我這裡選擇不**exit**，讓學生繼續搶**mutex**並且等待位子被釋出。

TA的thread

```
void* ta_teaching()
{
        int tmp;
        while(1)
        {
                sem_wait(&sem_stu);

                pthread_mutex_lock(&mutex);

                printf("\nwaiting line: student %d, student %d, student %d\n",
seat[0],seat[1],seat[2]);
                printf("TA is now teaching student %d\n",seat[next_teach]);

                count--;
                tmp = seat[next_teach];
                stuStatus[tmp-1] = 1;
                seat[next_teach] = 0;

                next_teach = (next_teach + 1) % SEAT_AMOUNT;

                random_delay();
                printf("TA finish teaching student %d\n",tmp);

                if(!count) printf("TA is sleeping now!\n\n");

                if(checkStatus())
                {
                        printf("TA finish teaching All student.TA is going hom
e!\n");
                        pthread_mutex_unlock(&mutex);
                        break;
                }

                pthread_mutex_unlock(&mutex);
                sem_post(&sem_ta);
        }
        pthread_exit(NULL);
}
```

在TA執行前，會先等待stu的semaphore被post，代表目前已經有學生在等待中，TA就可以開始教人。
如果count == 0 代表目前等待區是空的，TA就會睡覺。
checkStatus用迴圈判斷學生狀態，來決定TA是否可以回家了(break and exit)。

結果展示


user@instant-contiki:~/Desktop/os/hw2$ ./ta.out
TA teaching program Start!!!
student 5 is waiting now!
student 4 is waiting now!
waiting line: student 5, student 4, student 0
TA is now teaching student 5
TA finish teaching student 5
waiting line: student 0, student 4, student 0
TA is now teaching student 4
TA finish teaching student 4
TA is sleeping now!
student 5 is going home!
student 2 is waiting now!
student 1 is waiting now!
student 3 is waiting now!
student 4 is going home!
waiting line: student 1, student 3, student 2
TA is now teaching student 2
TA finish teaching student 2
waiting line: student 1, student 3, student 0
TA is now teaching student 1
TA finish teaching student 1
waiting line: student 0, student 3, student 0
TA is now teaching student 3
TA finish teaching student 3
TA is sleeping now!
TA finish teaching All student.TA is going home!
student 1 is going home!
student 2 is going home!
student 3 is going home!
TA teaching program End!!
user@instant-contiki:~/Desktop/os/hw2$