#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <string.h>

#include <time.h>

#include <semaphore.h>

#define NUM\_SEATING 3

#define SLEEP\_MAXX 5

pthread\_mutex\_t mutex; /\* mutex \*/

sem\_t sem\_stud;

sem\_t sem\_t\_a;

int chairs[3];

int count = 0; //count the waiting students

int nexxt\_bench = 0;

int nexxt\_teach = 0;

void random\_sleeps(void);

void\* stud\_programme(void\* stud\_ids);

void\* ta\_teaches();

int main(int argcs, char \*\*argvs){

pthread\_t \*students;

pthread\_t ta;

int\* students\_id;

int students\_numb;

int i;

printf("Number of Students are : ");

scanf("%d", &students\_numb);

if (students\_numb==0)

{

printf ("Teacher Assistant is sleeping,no students are their\n");

exit(-1);

}

students = (pthread\_t\*)malloc(sizeof(pthread\_t) \* students\_numb);

students\_id = (int\*)malloc(sizeof(int) \* students\_numb);

memset(students\_id, 0, students\_numb);

sem\_init(&sem\_stud,0,0);

sem\_init(&sem\_t\_a,0,1);

//set random

srand(time(NULL));

//create thread

pthread\_create(&ta,NULL,ta\_teaches,NULL);

//sem\_init(&mutex,0,1);

pthread\_mutex\_init(&mutex,NULL);

//create the threadss

for(i=0; i<students\_numb; i++)

{

students\_id[i] = i+1;

pthread\_create(&students[i], NULL, stud\_programme, (void\*) &students\_id[i]);

}

pthread\_join(ta, NULL);

for(i=0; i<students\_numb;i++)

{

pthread\_join(students[i],NULL);

}

return 0;

}

void\* stud\_programme(void\* stud\_ids)

{

int id = \*(int\*)stud\_ids;

printf("[stu] This Student %d is about to leave for TA's' office\n",id);

while(1)

{

random\_sleeps();

pthread\_mutex\_lock(&mutex);

if(count < NUM\_SEATING)

{

chairs[nexxt\_bench] = id;

count++;

printf(" [stu] This Student %d is about to be waiting seated in hallway\n",id);

printf("Wating student are as follows : [1] %d [2] %d [3] %d\n",chairs[0],chairs[1],chairs[2]);

nexxt\_bench = (nexxt\_bench+1) % NUM\_SEATING;

pthread\_mutex\_unlock(&mutex);

sem\_post(&sem\_stud);

sem\_wait(&sem\_t\_a);

}

else //no more chairss

{

pthread\_mutex\_unlock(&mutex);

printf("[stu] Their is no extra chair. Stud %d is now out of the hallway and about to be back later\n",id);

}

}

}

void\* ta\_teaches()

{

while(1)

{

sem\_wait(&sem\_stud);

pthread\_mutex\_lock(&mutex);

printf(" [ta] TA continously teaches the student %d\n",chairs[nexxt\_teach]);

chairs[nexxt\_teach]=0;

count--;

printf("Waiting Students are mentioned here: [1] %d [2] %d [3] %d\n",chairs[0],chairs[1],chairs[2]);

nexxt\_teach = (nexxt\_teach + 1) % NUM\_SEATING;

random\_sleeps();

printf(" [ta] Finally Teaching got finished.\n");

pthread\_mutex\_unlock(&mutex);

sem\_post(&sem\_t\_a);

}

}

void random\_sleeps(void){

int time = rand() % SLEEP\_MAXX + 1;

sleep(time);

}