The Sleeping Teaching Assistant

Computer Engineering 142 Operating Systems, San Jose State University, Fall 2017

Problem Statement:

A teaching assistant has an office with a single chair and the hallway has N number of chairs. The teaching assistant helps a student in their office and when they have finished, will go out to the hallway and check to see if any other students are waiting. If a student is waiting, the teaching assistant will help that student in their office; if no student is waiting, the teaching assistant will return to the office and sleep.

If a student goes to the teaching assistant's office and finds the teaching assistant sleeping, they will wake the assistant up and be helped; if the student finds another student being helped in the office, they will sit in an empty chair in the hallway and wait for their turn. If there are no empty chairs in the hallway to wait in, the student will come back later.

The students and teaching assistant take an undetermined amount of time to complete each of their functions. For example, the teaching assistant can finish helping a student and leave the office empty while they go to check the hallway for waiting students, and at the same time a student could check the office to see if the teaching assistant is busy. Because no one is in the room the student goes to the hallway to wait. If the teaching assistant doesn't have any students waiting for them, then they will go back to their office and sleep. If these both happen at the same time, the teaching assistant will sleep forever, and the student will wait outside of the office forever. Another problem comes from when there is only one chair available in the hallway and two students are trying to wait for the teaching assistant.

How to compile source code and run:

Navigate to directory and compile using GNU C++ Compiler

```
g++ -o sleeping_ta src/main.cpp -std=c++11 -pthread
```

After the executable is compiled, run it on the command line. You will be asked how many students you want to simulate, and it will go through and output each of the states for every student and what the TA is currently doing.

Sample Output:

Below is the output of running the program with 5 students and 4 waiting chairs. The number of students and waiting chairs is adjustable by passing two numbers for the arguments. The first number is the number of students and the second number is the number of chairs. This allows for the testing of different numbers of students and number of waiting chairs easily.

```
./sleeping_ta 5 4

Student #1 is programming
Student #2 is programming
Student #2 looking for TA's help
Student #2 waiting for TA in chair
Student #1 looking for TA's help
Student #1 waiting for TA in chair
Student #3 is programming
Student #2 goes into TA office
Student #3 looking for TA's help
Student #3 waiting for TA in chair
TA is helping Student #2
TA is done helping Student #2
TA is checking for students
Student #1 goes into TA office
Student #5 is programming
Student #5 looking for TA's help
TA is helping Student #1
TA is done helping Student #1
TA is checking for students
Student #4 is programming
Student #5 waiting for TA in chair
Student #4 looking for TA's help
Student #4 waiting for TA in chair
Student #3 goes into TA office
TA is helping Student #3
TA is done helping Student #3
TA is checking for students
Student #5 goes into TA office
TA is helping Student #5
TA is done helping Student #5
TA is checking for students
Student #4 goes into TA office
TA is helping Student #4
TA is done helping Student #4
TA is checking for students
```