Gebze Technical University
Computer Engineering

System Programming
CSE344 – 2021

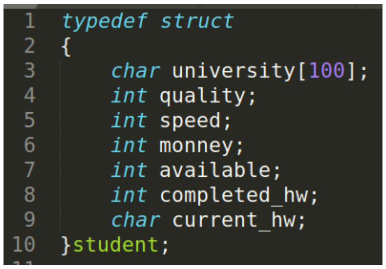
MIDTERM-REPORT

Yusuf Abdullah ARSLANALP
151044046

0.1 How I Solved This Problem

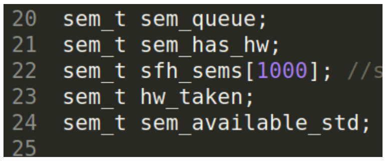
There are three types of thread. These are main thread, H thread and student-for-hire thread. Main thread creates all necessary semaphores and initialize them. After termination of all other threads main thread destroys all created semaphores.

Students for hire are read from the file and filled in to a struct array. The structure of struct as follows:



```
typedef struct
{
    char university[100];
    int quality;
    int speed;
    int monney;
    int available;
    int completed_hw;
    char current_hw;
}student;
```

Semaphores are kept as global. So they can be use from every threads. There are used four semaphore and one semaphore array in the homework. Semaphore names in program as follows:



```
sem_t sem_queue;
sem_t sem_has_hw;
sem_t sfh_sems[1000]; //s
sem_t hw_taken;
sem_t sem_available_std;
```

Every student-for-hire thread has its own semaphore. Initially all student-for-hire threads waits with their own semaphores. When main thread decides to assign a homework to one of the student-for-hire it increment semaphore value of the selected student-for-hire. Selected student does the homework. And waits for another assignment.

0.2 Which requirements I achieved

- When CTRL-C pressed program terminated gracefully.. And all resources are given back to the system.
- No warning with -Wall flag
- If the required command line arguments are missing/invalid, The program prints usage information and exit.
- The report prepeared via latex. (latex folder is in the homework)
- No zombie processes
- No busy waiting. No sleep.(Except for the student-for-hire threads. HW PDF specifies to use sleep for student-for-hire thread.)
- The make file only compiles the program.
- No memory leaks.