CSE 344 System Programming HW4

Yusuf Fatih Şişman 171044017

1 Problem Solving

1.1 Notifying student-for-hired Threads

I used array of unnamed POSIX semaphores which size is equal to number of students. Main thread increases the relative semaphore of chosen student. Since both students and semaphores in arrays, selecting relative semaphore is handled with index of student.

1.2 Accessing Empty Queue

I used an unnamed POSIX semaphore which represent size of queue for prevent accessing empty queue.

1.3 Waiting For Detached Thread

I used an unnamed POSIX semaphore for wait detached thread ending. So ı can safely free or closed used things.

1.4 Waiting Busy Students

I used an unnamed POSIX semaphore which is initialized with a value equals to number of students. When a student started to homework, value of semaphore is decreased; after a student finished homework, value of semaphore is increased. So if all students is busy, value of semaphore will be 0 and this cause to main thread waits.

1.5 Finding Proper Student

Program search for the best fitting, available and affordable student for current homework. If the current money can not afford any available students, program ends.

2 Design Decisions

2.1 Global Variables Used For Thread Communication

- hireds : It is used for store hired students. It is array of Hired structure. Hired structure include name, quality, speed, cost and busy state of hired students. It is created dynamically after read the file which includes students informations.
- solveCounts : It is used for store which students solve how many homeworks. It is an integer array. It is created dynamically after read the file which includes students informations.
- homeworks : It is a queue which hold homeworks. I took linked list implementation of queue from geeksforgeeks and modify it.
  `https://www.geeksforgeeks.org/queue-linked-list-implementation/`

2.2 Semaphores

- full : It is used to prevent accessing empty queue. It is initialized with zero.
- busy : It is used for waiting when all the hired students is busy. It is initialized with a value equals to number of hired students.
- queueLock : It is used to prevent multiple accessing to queue from threads. (creates critical section)
- hiredLock : It is used to prevent multiple accessing to hired students array from threads. (creates critical section)
- wait_h : It is used to wait for ending of detached thread(h thread). It is initialized with zero.
- wait_student(array of semaphores) : It is used to notify the proper student for solve the homework. It is initialized with zero.

3 Achieved Requirements

- Homeworks is given proper students.
- If money can not afford any available students, program ends.
- If no more homeworks left, program ends.
- Synchronization between threads handled successfully. There is no deadlock.
- There is no memory leak after program ends.
- In case of CTRL-C process terminates gracefully.