# CS342 Operating Systems - Fall 2021
# Project #2 – CPU Scheduling

Assigned: November 3, 2021, Tuesday.
Due date: November 18, 2021, 23.59. Document version: 1.1

*This project will be done in groups of two students. You can do it individually as well. You will use the C programming language in Linux OS.*

In this project you will implement a CPU scheduling simulator (schedule.c).

You will simulate the following scheduling algorithms: FCFS, SJF, SRTF, and RR(q). Min value of q (time quantum) can be 10 ms, and max value can be 300 ms. The workload will be specified in an input file as a sequence of CPU bursts. Number of such bursts can be large (maximum 1000 bursts). In the input file, for each burst, burst number, the arrival time and burst length is specified (in ms). For example:

```
1  0   60
2  10  120
3  15  50
4  35  100
5  45  60
6  46  60
.....
```

Above, for example, for burst 2, 10 ms is the arrival time and 120 ms is burst length.

Max burst length is 400 ms and min burst length is 5 ms.

In case of a tie, give priority to the burst that has the smaller number (for example, put such a burst earlier into the queue, or run it earlier).

As the output, measure the avg turnaround time.

The program will be run as follows:

./schedule <inputfile> <quantum>

<inputfile> will contain the bursts. <quantum> is time quantum used in RR scheduling. Output will be average turnaround time expressed as an integer in a separate line of output for each scheduling algorithm, in the following format. You will round the average turnaround time to the closest integer.

```
FCFS  100
SJF   100
SRJF  100
RR    100
```

Example, invocation of the program can be:

./schedule 100

## Experiments and Report (30 pts).

Run your program with a fixed input file but with various time quantum values. Plot the results.

## Submission

Submit a pdf file as your report presenting and discussing your experiments. Your report will include the results, your interpretations and conclusions. Put your report.pdf file and all other files (including a Makefile) into a directory named with your ID (one of the students' ID is enough). Then tar and gzip the directory. For example, a student (of a group) with ID 21404312 will create a directory named "21404312" and will put the files there. Then he/she will tar the directory (package the directory) as follows:

tar cvf 21404312.tar 21404312

Then he/she will gzip the tar file as follows:

gzip 21404312.tar

In this way he/she will obtain a file called 21404312.tar.gz. Then he/she will upload this file into Moodle.

## Tips and Clarification

- Starting early is highly recommended.
- Work incrementally.