

Assignment #3

1. Problem

1) Implement a scheduling module for an operating system

One of the main applications of priority queues is in operating systems for scheduling jobs on a CPU. You should build a scheduler for simulated CPU jobs. Your program should run in a loop, each iteration of which corresponds to a time slice for the CPU. Each job is assigned a priority (-50 to 49, the lower number runs first). From among all jobs waiting to be processed in a time slice, the CPU must work on a job with the highest priority. In this simulation, each job will also come with a length value (1 to 50), indicating the number of time slices that are needed to process this job. Your program should be able to add a job into the priority queue at any time during the simulation. Processed job priority is always increased by its unit.

The conditions given are:

- CPU performs one length of work in one loop.
- The priority of jobs processed by CPU always increases by its unit (Priority does not exceed 49).
- The job priority is between -50 and 49 (-50 is the highest priority).
- The job unit is between 1 and 100.
- The job length is between 1 and 50.
- There is no duplicate job ID.
- Be sure to use a priority queue to solve this problem
- Return the time each job ended.
- Print job scheduling log.
- If there are multiple jobs with the same priority, print the lowest job ID first.

Example of input (scheduler_input1.txt)

2 (number of jobs)

-7 5 2 (job_id = 1, priority = -7, unit = 5, length = 2)

-6 3 2 (job_id = 2, priority = -6, unit = 3, length = 2)

The return value of the above input (end time of each job)

- output_list[0] = 3
- output_list[1] = 4

The output log of the above input

- 1: Job ID 1 with priority -7 has been processed, remaining time length: 1
- 2: Job ID 2 with priority -6 has been processed, remaining time length: 1
- 3: Job ID 2 with priority -3 has been processed, the job is done
- 4: Job ID 1 with priority -2 has been processed, the job is done