CS370 Fall 2025
HW4: Programming Assignment v.10/27/2025 10:42 AM

CPU Scheduling Algorithms

In this assignment, you will implement First Come First Serve, Shortest Job First with preemption, and Round Robin to generate a Gantt chart to evaluate measures of effectiveness.

Due Date: Thursday, October 30, 2025, 11:00 pm
Late penalty: 10% per day until Saturday, Nov 1, 2025, **11:00 pm**

This document is available at Canvas (Assignments > HW4)

1. Task Description

You will be developing the three CPU scheduling methods listed below in this project. The first command line argument indicates the name of the file containing the list of processes to be used by your scheduling methods. The next command line argument specifies the time quantum (for Round Robin). You will use Python 3 for the programming language to implement these three scheduling techniques shown below (you are free to use Pandas and NumPy libraries if needed). Assume that all jobs are CPU bound (no I/O blocking) and that the context-switching time is minimal. Assume that in Round Robin, if a new process arrives at the same time as a process is switched out, the new process is placed first in the ready queue.

- First Come First Serve (FCFS)
- Shortest Job First with pre-emption (SJF-P)
- Round Robin (RR) with the specified quantum.

You must produce a Gantt chart for each scheduling algorithm to visualize the sequence of execution for each process (See the example output below). You will analyze the performance of these scheduling algorithms by tracking the turnaround time and wait time for each process by printing: the average waiting time, the average turnaround time, and the throughput after all processes have completed.

- The turnaround time for a process is the difference between a job's submission and completion times. The average turnaround time reports the average for the set of processes that were scheduled.
- The waiting time for a process reflects the total amount of time spent by that process in the ready queue. The average waiting time reports the average for the set of processes that were scheduled.
- The throughput for a scheduling algorithm measures the total number of tasks processes per unit time.

2. **Task Requirements**
   1. Your program must be able to take two command-line parameters. The first argument specifies the name of the process file (for example: processinfo.csv). This file is comma-separated and has four columns (process ID, arrival time, burst time, priority), with each row representing a single process. This file is expected to contain a maximum of 15 processes. The second argument is the time quantum for Round Robin scheduling. Please note that the file that we pass can be named anything. DO NOT JUST LOOK FOR A FILE CALLED "PROCESSINFO.CSV"
   2. Given a set of processes, arrival time, burst time, and priority for each process, your application (scheduler.py) should be able to do FCFS, Shortest Job First with Preemption, and Round Robin scheduling appropriately.
   3. Use the following format to implement FCFS and print a Gantt chart depicting the execution process (align the number in the columns). Report each process's waiting time and turnaround time. Report the average waiting time and turnaround time, as well as the overall throughput for all procedures.
   4. Repeat item 3 above for Shortest Job First with pre-emption (if a process with shorter remaining burst time arrives, it pre-empts the process with longer remaining time currently running).
   5. Repeat item 3 above for RR using the specified quantum.

Note: The processes in the file (first command line argument file) may be specified in such a way that it may result in some IDLE time when there are no processes ready to be executed. During the IDLE time the CPU has no processes to execute and waits for the next process to appear in the ready queue. Your Gantt chart should include these IDLE times. Also, the first process need not start at time 0. At time 0, there can be IDLE time in the Gantt chart till the first process arrives later on in time.

*** Assume that when two new processes arrive at the same time, the process with the shorter burst time enters the Ready Queue first. If burst times are equal, the process with lower process ID goes first. Your program must explicitly handle this ordering rule when retrieving new arrivals at each simulation time step. To ensure this, implement a helper method (for example, get_arrivals(time)) that performs the following actions:
1. Filters the list of processes to select those whose arrival time equals the current simulation time.
2. Sorts these processes by burst time (shorter burst time first).
3. If two or more processes have the same burst time, break ties by process ID (lower process ID first).
The sorted list returned from this helper should then be added to the ready queue in that order. This ensures consistent queue behavior for cases in which multiple processes arrive simultaneously.
Example Outputs:
Input File: processinfo.csv contains the following comma-separated values:

```
ProcessID,Arrival Time,Burst Time
3,0,6
2,0,5
1,12,8
4,13,6
```

Note: The first line in the processinfo.csv file has headers Process ID, Arrival Time, Burst Time, Priority.

```
<system_name>:<folder_path>$ python3 scheduler.py processinfo.csv 3

________________FCFS________________
Process ID 1 2 3 4
Waiting Time 0 0 5 7
Turnaround Time 8 5 11 13

FCFS Gantt Chart
[0]--2--[5]
[5]--3--[11]
[11]--IDLE--[12]
[12]--1--[20]
[20]--4--[26]

Average Waiting Time: 3.0
Average Turnaround Time: 9.25
Throughput: 0.15384615384615385

___Shortest Job First with Preemption_____
Process ID 1 2 3 4
Waiting Time 6 0 5 0
Turnaround Time 14 5 11 6

Shortest Job First with Preemption Gantt Chart
[0]--2--[5]
[5]--3--[11]
[11]--IDLE--[12]
[12]--1--[13]
[13]--4--[19]
[19]--1--[26]

Average Waiting Time: 2.75
Average Turnaround Time: 9.0
Throughput: 0.15384615384615385
```

```
___________Round Robin___________
Process ID 1 2 3 4
Waiting Time 6 3 5 5
Turnaround Time 14 8 11 11

Round Robin Gantt Chart
[0]--2--[3]
[3]--3--[6]
[6]--2--[8]
[8]--3--[11]
[11]--IDLE--[12]
[12]--1--[15]
[15]--4--[18]
[18]--1--[21]
[21]--4--[24]
[24]--1--[26]

Average Waiting Time: 4.75
Average Turnaround Time: 11.00
Throughput: 0.15384615384615385
```

5. What to Submit

Use the CS370 Canvas to submit a single .zip or .tar file that contains:

- All .py files with descriptive comments within,
- a README.txt file containing a description of each file and any information you feel the grader needs to grade your program.
- Please do not include processinfo.csv with your submission.
- Create a zip with the source code files only. Don't put the files under a directory.
- Try to maintain the output strings as exactly as the output format given in Section 4, since our Autograder testing script is case-sensitive.

For this and all other assignments, ensure that you have submitted a valid .zip/.tar file. After submitting your file, you can download it and examine to make sure it is indeed a valid zip/tar file, by trying to extract it.

Filename Convention: The archive file must be called: <FirstName>-<LastName>-HW4.<zip>. E.g. if you are John Doe and submitting for assignment 4, then the zip file should be named John-Doe-HW4.zip.

In the example above, processinfo.csv is the name of the file containing the information about all the processes; your program should work with any name (not just processinfo).

6. Grading

The assignments must compile and function correctly on machines in the CSB-120 Lab. Assignments that work on your laptop on your particular flavour of Linux/Mac OS X but not on the Lab machines are considered unacceptable.

The grading will be done on a 100-point scale. The points are broken up as follows:

Objective	**Points**
FCFS - Gantt Chart, Throughput, Waiting time and Average Waiting time, Turnaround Time and Average Turnaround time	40 points
Shortest Job First with Preemption - Gantt Chart, Throughput, Waiting time and Average Waiting time, Turnaround Time and Average Turnaround time	30 points
Round Robin - Gantt Chart, Throughput, Waiting time and Average Waiting time, Turnaround Time and Average Turnaround time	25 points
Compilation with no warnings or errors	2 points
Suitable documentation of code in code files and README	3 points

You are required to [work alone] on this assignment.

7. Late Policy

Click here for the class policy on submitting late assignments.

Revisions: Any revisions/clarifications in the assignment will be noted below.

No	**Changes**	**Revision**
1	Initial draft	v.10/12/2025 11:58PM
2	• Removed Priority from processinfo.csv • Scheduler's output format has been modified	v.10/17/2025 09:40AM
3	• Example Output format and values corrected and updated • Clearly explained the situation when two new processes arrive at the same time.	v.10/20/2025 11:54 AM
4	Round-Robin output example corrected	v.10/22/2025 10:12 AM
5	Round-Robin Gantt-Chart example corrected	v.10/27/2025 10:42 AM