

---

## 6G7Z1004 Advanced Computer Networks and Operating Systems

### (Lab 1: Process Scheduling)

This lab was originally designed by Emma Norling, MMU, UK

---

The purpose of this lab is to explore the different process scheduling algorithms discussed in today's lecture. To this end, we will use a Java-based simulator of OS concepts named **OS sim** which can be downloaded from: <https://sourceforge.net/projects/oscsimulator/>. This simulator was written by Alex Macia, and is distributed under a BSD License.

#### **Task 1**

Download `ossim-1.2.tar.gz` and extract the files. Use a terminal window to enter the command

```
./ossim.sh
```

in the directory in which you extracted the files. (Alternatively, double-click on that `ossim.jar` file - this should work on Windows too.)

Click on the top left icon (the CPU) for the process scheduling simulator. You have a choice of four scheduling algorithms, and an option (for some of them) for **pre-emptive scheduling**. Start with FCFS (the default); close the settings window to get back to the main window.

Now, set up a simulation of the processes shown in the table below:

Process	Arrival time	No. bursts
P1	0	3
P2	2	6
P3	4	4
P4	6	5
P5	8	2

To create a process, click the + button near the top of the window. Enter the name (P1 for the first one), leave the priority at the default, set the submission time (default for the first one) and the number of bursts (3 for the first one). Then click Done. Do similar for the other 4 processes.

Now run the simulation. Watch how processes get put into the ready queue, then as the processor (CPU) becomes available, the first process in the ready queue gets moved to the processor. Click on the **Data and Statistics** button (the table-like button next to the + button) to see the statistics of the run.

Now try with different scheduling algorithms: Click the stop button, then the settings button (toolbox-like button in the other side of the + button). Explore the different settings and see what happens in the simulations.

Explore this topic further through the inbuilt exercises at

**Help->Exercises->Process scheduling**

### **Task 2**

Once you have explored your understanding of the scheduling algorithms, test your knowledge by solving these problems **without** using the simulator.

Consider the following set of processes with arrival times (i.e. submission time) and length of CPU burst times given in milliseconds:

Process	Arrival time	No. bursts
P1	9	3
P2	8	9
P3	7	4
P4	6	7
P5	5	9

**Q1:** which CPU scheduling algorithm listed below yields the shortest average **turnaround time**?

- (a) FCFS (First-Come-First-Served)
- (b) SJF (Shortest Job First; without pre-emption)
- (c) SRT (Shortest Remaining Time; same as SJF with pre-emption)
- (d) RR-1 (Round-Robin with a time quantum of 1 millisecond)

**Q2:** which CPU scheduling algorithm listed below yields the shortest average **waiting time**?

- (e) RR-1 (Round-Robin with a time quantum of 1 millisecond)
- (f) RR-2 (Round-Robin with a time quantum of 2 milliseconds)

**Turnaround time** – amount of time needed to execute a particular process

**Waiting time** – amount of time a process has been waiting in the ready queue