COMP 2004 - Fall 2021
Instructor: Dr Vinicius Prado da Fonseca
Assignment 4

1. (40%) Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst time	Priority
P_1	5	4
P_2	3	1
P_3	1	2
P_4	7	2
P_5	4	3

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

   a. (20%) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
   b. (8%) What is the turnaround time of each process for each of the scheduling algorithms in part a?
   c. (8%) What is the waiting time of each process for each of these scheduling algorithms?
   d. (4%) Which of the algorithms results in the minimum average waiting time (overall processes)?

2. (30%) The following processes are being scheduled using a pre-emptive, priority-based, round-robin scheduling algorithm.

Process Time	Priority	Burst	Arrival
P_1	8	15	0
P_2	3	20	0
P_3	4	20	20
P_4	4	20	25
P_5	5	5	45
P_6	5	15	55

Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. The scheduler will execute the highest priority process.

For processes with the same priority, a round-robin scheduler will be used with a time quantum of 10 units. If a process is pre-empted by a higher-priority process, the pre-empted process is placed at the end of the queue.

a. (16%) Show the scheduling order of the processes using a Gantt chart.
b. (8%) What is the turnaround time for each process?
c. (8%) What is the waiting time for each process?

3. (30%) Design a program using ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and sends it back to the first process. For example, if the first process sends the message Hi There, the second process will return hI tHERE. This will require using two pipes, one for sending the original message from the first to the second process and the other for sending the modified message from the second to the first process. Write this program using UNIX pipes.

   The code should read the string message as a single line from a file name provided on the command line. The output should be the original string on a line and the processed string on the second line. All output should be done in the "first process". i.e., the process that modifies the string should not print any output. For example:

   ```
   $ ./a.out string.txt
   Hi There
   hI tHERE
   ```

   Contents of string.txt file:

   ```
   Hi There
   ```

   The file provided "a4q3_sample.c" may be used as a starting point. (Sample code extracted from Section 3.7.4.1 from the textbook or Section 3.6.3.1 from 9e)