Homework 2
COM S 352
Fall 2021

1. (10 points) Of these two types of programs:

a. I/O bound (interactive)
b. CPU bound

which is more likely to have voluntary context switches, and which is more likely to have nonvoluntary context switches? Explain your answer.

2. (20 points) OSTEP describes the problem of starvation in section 8.2 and shows how priority boost can be used to avoid it in MLFQ. Consider other scheduler policies. Explain why each of the following policies can or cannot result in starvation?

a. FIFO
b. STCF
c. RR
d. Lottery

3. (10 points) CPU efficiency can be defined as:

```
process_running_time / (process_running_time + os_overhead)
```

Between STCF and RR which do think will typically have better CPU efficiency? What are the overheads involved? Explain your answer.

4. (30 points) Consider the following set of jobs, with times given in milliseconds:

Job	Arrival Time	Runtime
A	0	20
B	2	20
C	5	5
D	10	10

a. Draw four Gantt charts that illustrate the execution of these jobs using the following scheduling algorithms: FIFO, SJF, STCF and RR (with quantum = 5).

The charts must clearly label the job and the start and end time of each run, for example:

```
|    A    |  B  |  C  |  B  |
0         10    15    20    25
```

b. What is the average turnaround time for each of the scheduling policies? Show calculations.

c. What is the average response time for each of these scheduling policies? Show calculations.

5. (20 points) Consider the following set of jobs, with times given in milliseconds.

Job	Arrival Time	Runtime
A	0	200
B	5	20
C	10	10
D	50	10

Assume we are using a MLFQ scheduler with 3 priority levels Q2: quanta=10, Q1: quanta=20, Q3: quanta=50.

The MLFQ follows rules 1-4 from OSTEP, it does not have priority boost rule 5.

Draw a Gantt chart that illustrates the execution of the jobs.

6. (10 points) Assume that two processes, A and B, are running on a Linux system. The nice values of A and B are −5 and +5, respectively. If both A and B have the same positive `vruntime`, and neither has had its `vruntime` reset due to sleeping, which has run the longest actual time on the CPU? Explain your answer.