Roll Number: Name:

Instructions

1. Total marks – 25
2. For MLFQ, choose the convention discussed in our class. If a lower PID process can run (no I/O pending, time slices left, no higher priority process), then it runs. It can pre-empt if needed.

Q1. Which of these is **not** a process state? a) Ready, b) Running, c) Top, d) Blocked **[0.5]**

Q2. Consider a single CPU (processor). How many programs/processes can it run at a given instance of time? Why? **[0.5]**

Q3. Which of them is **not** a system call? a) wait, b) strlen, c) exec, d) fork **[0.5]**

Q4. Name at least 2 architectures on which Linux can run. Hint: think about your laptop and phone. **[0.5]**

Q5. List any 3 important design goals of operating systems. **[1]**

Q6. List any 5 different pieces of hardware that most computers today come with? **[1]**

Q7. Write a program using fork() (only main function).The child process should print "hello"; the parent process should print "bye". You should ensure that the child process always prints first; do this without calling **wait()** in the parent? Do not use **sleep(). [1]**

Q8. There are 3 components (disk, memory and CPU) associated with the execution of a program (program -> process). Explain briefly the workflow from a .c program to a process being executed. **[2]**

Q9. For what types of workloads and quantum lengths does SJF deliver the same response times as RR? **[1]**

Q10. What happens to response time with RR as quantum lengths increase? Can you write an equation that gives the worst-case response time, given N jobs? **[2]**

Q11. Print the output (with explanation) of the program assuming all correct headers are present. **[1]**

```
int main(int argc, char *argv[]) {
int x = 10;
printf("x = %d\n", x);
int rc = fork();
if (rc <0) { printf("Fork failed"); }
else if (rc ==0) {
    x = x + 10;
    printf("x = %d\n", x);
    }
else {
   wait(NULL);
   printf("x = %d\n", x);
   }
}
```

Q12. Under MLFQ scheme, compare the response time and turnaround time for the following 3 processes: J0 (start = 0, timeunits = 19), J1 (start = 0, timeunits = 24), J2 (start = 0, timeunits = 22).

a) Time quantum of Q2 (highest priority) is 2, Q1 is 10 and Q0 (lowest priority) is 15.
b) Time quantum of Q2 (highest priority) is 15, Q1 is 10 and Q0 (lowest priority) is 2.

What can you conclude? **[3]**

Q13. Assuming 1<= Qi < = 18 for i = 2, 1, 0; such that sum(Qi) = 20. Given the workload in above question, what values of Q2, Q1, and Q0 will you choose to optimize the turnaround time and the response time? **[2]**

Q14. How does MLFQ Rule 4 (over Rule 4a and Rule 4b) prevent scheduler gaming? What aspect of MLFQ helps prevent process starvation? **[1]**

Q15. Calculate the turnaround time and response time under MLFQ for the following load and configuration. Quantum length of Q2, Q1, and Q0 are 5, 10 and 10 respectively. Boost time is 30. Jobs: J0 (start_time = 0, run_time = 40), J1(start_time = 10, run_time = 20), J2 (start_time = 20, run_time = 30). **[4]**

Q16. What happens to response time with SJF as job lengths increase? Hint: Assume N jobs ordered in increasing length such that time(J_i) < time(J_{i+1}). What's the response time of this workload? How would response time vary if: each time(J_i) was changed to time(J_i + K). How would it vary if each time(J_i) was changed to time(J_i * K)? **[2]**

Q17. Explain a scenario where shortest remaining time first is better than shortest job first. **[2]**