CS206 Operating System

Winter Semester 2022-23
Mid Semester Examination
Dr. Ashish Phophalia

Feb 28, 2023

1. **[3 Marks]** Shortest remaining time First (SRTF) scheduler behaves like shortest job First (SJF) algorithm if the set of processes are Fixed (no new process is created) and the processes are completely CPU bound. Is statement True or False? Justify.

2. **[3 Marks]** What is the difference between CPU bound Process and I/O bound Process. How Context Switch will be effected by either type of processes?

3. **[3 Marks]** Ten processes share a critical section implemented using a counting semaphore named x. Nine of these processes use the code: wait(x); CriticalSection; signal(x). However, one process erroneously uses the code signal(x); CriticalSection; signal(x). What is the maximum number of processes that can be in the critical section at the same time? Justify. 10

4. **[3 Marks]** Under what circumstances is Rate-Monotonic Scheduling inferior to Earliest Deadline First Scheduling in meeting the deadlines associated with processes?

5. **[3 Marks]** Can a system detect that some of its processes are starving? If you answer 'yes', explain how it can. If you answer 'no', explain how the system can deal with the starvation problem.

6. **[3 Marks]** The following two functions P_1 and P_2 that share a variable B with an initial value of 2 execute concurrently.

PROCESS P_1 CODE
P1()
{
C = B-1;
B = 2*C
}

B = 2
C = 1
B = 2

PROCESS P_2 CODE
P2()
{
D = 2*B;
B = D-1
}

D = 4
B = 3

2
4 5

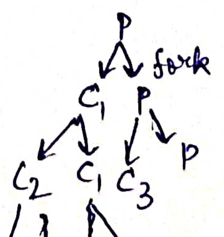
Find out all the possible values for the variable B after the complete execution of both the processes.

7. **[3 Marks]** Consider the following code segment:

pid_t pid;

pid = fork();

6



```
if (pid == 0) { /* child process */
        fork();
        thread_create(...);
}
fork();
```

6 (a) How many unique processes are created?

2 (b) How many unique threads are created?

8. **[5 Marks]** To what degree do the following algorithms favor CPU bound processes?

   (a) First Come First Serve

   (b) Shortest Job First

   (c) Shortest Remaining Time

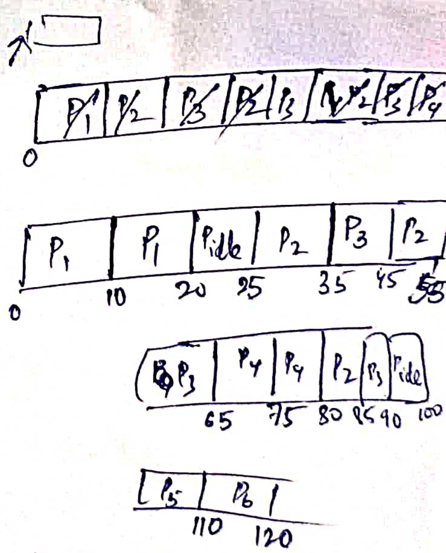
   (d) Round Robin

   (e) Multilevel Feedback Queue

9. **[3+1+1+1=6 Marks]** The following processes are being scheduled using a preemptive, round robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an **idle task** (which consumes NO CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

TAT = CT − BT
WT = TAT − AT

Thread	Priority	Burst	Arrival
P_1	40	~~20~~ 10	0
P_2	30	~~25~~ ~~15~~ 5	25
P_3	30	~~25~~ ~~15~~ 5	30
P_4	35	~~15~~ 5 0	60
P_5	5	10	100
P_6	10	10	105



CT − AT
TAT = CT − AT − BT

5/6

   (a) Show the scheduling order of the processes using a Gantt Chart.

   (b) What is the turnaround time for each process?

   (c) What is the waiting time for each process?

   (d) What is the CPU utilization rate?

10. **[6 Marks]** Consider the following Two Process solution for Critical Section Problem. The two processes, P_0 and P_1, share the following variables:

```
boolean flag[2]; /* initially false */
int turn;
```

The structure of process P_i (i == 0 or 1) is shown below. The other process is P_j (j == 1 or 0). Prove or Disprove that the algorithm satisfies all three requirements for the critical section problem or not.

ME ✓

```
do {
    flag[i] = true;

    while (flag[j]) {
        if (turn == j) {
            flag[i] = false;
            while (turn == j)
                ; /* do nothing */
            flag[i] = true;
        }
    }

    /* Critical Section */

    turn = j;
    flag[i] = false;

    /* Remainder Section */
} while (true);
```

11. **[3+3=6 Marks]** Consider two processes, P_1 and P_2, where period $p_1 = 50$, CPU time $t_1 = 25$, $p_2 = 75$ and $t_2 = 30$ (Assume that the deadline is start of the next period). Can these two processes be scheduled using rate-monotonic scheduling? Justify. Illustrate the scheduling of two processes mentioned using earliest deadline first (EDF) Scheduling.

12. ***[1+1+3=6 Marks]*** Least Laxity First (LLF) is a preemptive real time scheduling algorithm for periodic tasks. Slack time, or laxity, is the amount of the time between when a task would complete if it started now and its next deadline. This is the size of the available scheduling window. Laxity can be expressed as

$$Laxity = DeadlineTime - CurrentTime - ProcessorTimeNeeded$$

LLF selects the task with the minimum laxity to execute next. If two or more tasks have the same minimum laxity value, they are serviced on a FCFS Basis. Consider deadline from start of period, i.e. for Process P_1, at time $t = 20$, Deadline would be $t = 27$.

(a) Suppose a task currently has a laxity of 0. What does this mean?

(b) What does it mean if a task has negative laxity?

(c) Given the following tasks arrived at Time $t = 0$, construct the Gantt chart for time $t = 0$ to $t = 15$. Show laxity values and selection at each step.

Table 1: LLF Scheduling Tasks

Task	Period	Deadline	Execution Time
P_1	20	7	3
P_2	5	4	2
P_3	10	8	1