**Department of Computer Science and Engineering**

**Midterm Examination Fall 2022**

**CSE 321: Operating Systems**

**[Set A]**

|  |  |
| --- | --- |
| **Duration:** 1 Hour 10 Minutes | **Total Marks:** 25 |

Answer the following questions.

Figures in the right margin indicate marks.

|  |  |  |
| --- | --- | --- |
| 1.  **CO1** | a) **State** time-sharing system with an example.  b) **Mention** the roles of the system call interface. | [3]  [2] |
| 2.  **CO2** | a) **Distinguish** between CPU scheduler and job scheduler.  b) **Find** the output of the following code snippet.   |  | | --- | | int main(){  int id;  static int x = 10;  int y = 5;  id = fork();  if (id < 0){  printf("fork failed\n");  }  else if(id == 0){  printf("child started\n");  printf("child finished\n");  }  else{  wait(NULL);  printf("parent started\n");  x=x-2;  y=y+5;  printf("values of x: %d & y: %d\n",x,y);  printf("parent finished\n");  }  x=x+5;  y=y-5;  printf("values of x: %d & y: %d\n",x,y);  printf("terminating\n");  return 0;  } | | [3]  [4] |
| 3.  **CO3** | a) **When** is CPU scheduling required?  b) **Draw** a Gantt chart and illustrate the execution of the process using the **Round Robin** scheduling algorithm **(time quantum = 11 units)**. **Calculate** the **average waiting time** and **number of context switching**.   |  |  |  | | --- | --- | --- | | **Processes** | **Arrival Time** | **Burst Time** | | P1 | 3 | 37 | | P2 | 12 | 17 | | P3 | 58 | 28 | | P4 | 59 | 21 | | P5 | 68 | 19 | | [2]  [3+2+1] |
|  |
|  | c) Consider the following set of processes with the length of the CPU-burst time given in milliseconds. **Draw** the Gantt Charts illustrating the execution of these processes using **preemptive priority** (the lowest number implies a higher priority). **Calculate** the **average turnaround time** for the below data set.   |  |  |  |  | | --- | --- | --- | --- | | **Processes** | **Priority** | **Arrival Time** | **Burst Time** | | P1 | 12 | 0 | 4 | | P2 | 8 | 1 | 2 | | P3 | 6 | 2 | 3 | | P4 | 2 | 3 | 5 | | P5 | 4 | 4 | 1 | | P6 | 1 | 5 | 4 | | P7 | 3 | 6 | 6 | | [3+2] |