CSE321: Operating Systems
Quiz-1

Name: _____ ID: _____ Section: ____

Q1) Why must the Long term scheduler select wisely? **[3]**

Q2) Find output of the following code: **[6]**

```
int x = 63;
int y = 49;
pid_t pid1 = fork();
if (pid1 == 0) {
    x += 9;
    y -= 8;
    pid_t pid2 = fork();
    if (pid2 == 0) {
        x -= 12;
        y += 10;
        printf("Child 1: x = %d", x);
        printf("Child 1: y = %d", y);
    } else {
        x -= 15;
        y += 8;
        printf("Child 2: x = %d", x);
        printf("Child 2: y = %d", y);
    }
} else {
    wait(NULL);
    x += 8;
    y -= 16;
    printf("Parent: x = %d", x);
    printf("Parent: y = %d", y);
}
```

Q3) Apply Round Robin (RR) scheduling algorithm with quantum = 4 and show the following –

- Gantt Chart **[3]**
- Average Waiting Time & Average Turnaround Time **[2]**
- Number of Context Switching **[1]**

Process ID	Arrival Time	Burst Time
P1	0	8
P2	3	9
P3	2	10
P4	1	3
P5	5	5

CSE321: Operating Systems

Quiz-1

Name: _____ ID: _____ Section: ____

Q1) What are Zombie processes, how can we prevent that? **[3]**

Q2) Find output of the following code: **[6]**

```
int x = 63;
int y = 89;
pid_t pid1 = fork();
if (pid1 == 0) {
    x += 1;
    y -= 13;
    pid_t pid2 = fork();
    if (pid2 == 0) {
        x -= 14;
        y += 17;
        printf("Child 1: x = %d", x);
        printf("Child 1: y = %d", y);
    } else {
        x -= 7;
        y += 2;
        printf("Child 2: x = %d", x);
        printf("Child 2: y = %d", y);
    }
} else {
    wait(NULL);
    x += 7;
    y -= 7;
    printf("Parent: x = %d", x);
    printf("Parent: y = %d", y);
}
```

Q3) Apply Round Robin (RR) scheduling algorithm with quantum = 4 and show the following –

- Gantt Chart **[3]**
- Average Waiting Time & Average Turnaround Time **[2]**
- Number of Context Switching **[1]**

Process ID	Arrival Time	Burst Time
P1	0	9
P2	4	4
P3	9	10
P4	7	2
P5	5	2

CSE321: Operating Systems

Quiz-1

Name: _____ ID: _____ Section: ____

Q1) Explain Long term and Short term scheduler, what is the reason for such naming? **[3]**

Q2) Find output of the following code: **[6]**

```
int x = 66;
int y = 16;
pid_t pid1 = fork();
if (pid1 == 0) {
    x += 15;
    y -= 5;
    pid_t pid2 = fork();
    if (pid2 == 0) {
        x -= 1;
        y += 17;
        printf("Child 1: x = %d", x);
        printf("Child 1: y = %d", y);
    } else {
        x -= 15;
        y += 15;
        printf("Child 2: x = %d", x);
        printf("Child 2: y = %d", y);
    }
} else {
    wait(NULL);
    x += 15;
    y -= 10;
    printf("Parent: x = %d", x);
    printf("Parent: y = %d", y);
}
```

Q3) Apply Round Robin (RR) scheduling algorithm with quantum = 4 and show the following –

- Gantt Chart **[3]**
- Average Waiting Time & Average Turnaround Time **[2]**
- Number of Context Switching **[1]**

Process ID	Arrival Time	Burst Time
P1	0	3
P2	2	10
P3	2	2
P4	1	6
P5	8	5

CSE321: Operating Systems

Quiz-1

Name: _____ ID: _____ Section: ____

Q1) What is context switching? What can be the drawbacks of very frequent context switching? **[3]**

Q2) Find output of the following code: **[6]**

```
int x = 35;
int y = 1;
pid_t pid1 = fork();
if (pid1 == 0) {
    x += 19;
    y -= 13;
    pid_t pid2 = fork();
    if (pid2 == 0) {
        x -= 16;
        y += 8;
        printf("Child 1: x = %d", x);
        printf("Child 1: y = %d", y);
    } else {
        x -= 10;
        y += 12;
        printf("Child 2: x = %d", x);
        printf("Child 2: y = %d", y);
    }
} else {
    wait(NULL);
    x += 12;
    y -= 14;
    printf("Parent: x = %d", x);
    printf("Parent: y = %d", y);
}
```

Q3) Apply Round Robin (RR) scheduling algorithm with quantum = 4 and show the following –

- Gantt Chart **[3]**
- Average Waiting Time & Average Turnaround Time **[2]**
- Number of Context Switching **[1]**

Process ID	Arrival Time	Burst Time
P1	0	7
P2	3	4
P3	6	2
P4	6	8
P5	8	7