

Q1: Draw a process tree using following program:

```
#include <stdio.h>
#include <unistd.h>
int main()
{
    int i;

    for (i=0; i<4;i++)
        fork();
    return 0;
}
```

Q2. Consider the following program code. Determine about each statement that whether it would be executed in user mode or kernel mode.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main() {
    pid_t forkStatus;

    forkStatus = fork();
    if (forkStatus == 0) {
        printf("Child is running, processing.\n");
        sleep(5);
        printf("Child is done, exiting.\n");
    } else if (forkStatus != -1) {
        printf("Parent is waiting...\n");
        wait(NULL);
        printf("Parent is exiting...\n");
    } else {
        perror("Error while calling the fork function");
    }
    return 0;}

```

Q3. Draw Gantt charts that illustrate the execution of these processes given in Table 1. Calculate average turnaround time and average waiting time for part a and b.

- a) Preemptive Version of Shortest Job First algorithm
- b) Non-Preemptive Version of Shortest Job First algorithm

Table 1 : Process Burst Time

Process	Arrival Time	Burst Time
P1	0	11
P2	5	28
P3	12	2
P4	2	10
P5	9	16

Q4. Answer the Following questions as True and False. You must also specify a one or two sentence explanation of your answer. No point will be awarded if you do not explain.

- a. All instructions execute in kernel space in Microkernel operating system structure.
- b. Asymmetric multiprocessing executes all processes in shared address space.
- c. System calls such as read () and write () execute in user mode.
- d. In message passing, direct communication uses a port to send () and recv () messages.
- e. Clustering provides high availability, fault tolerance, and reliability.

Q5. Answer the following questions:

- 1. Which type of processes demand more I/O resources as compare to CPU resources?
- 2. Suppose a browser is executed with multiple tabs, which type of processes are created. How a cascaded termination can be performed in this example?
- 3. How can we find the process id of a parent process and a child process?
- 4. How can we distinguish between policy and mechanism? Explain with operating system's example.
- 5. Critical processes should be schedule preemptive or non-preemptive?