**Lab06 Documentation:**

**Unix programming exercise:**

1. **How many child processes are created?**

Try to work out the answer of this question mentally: what would happen if you call fork twice in the following way:

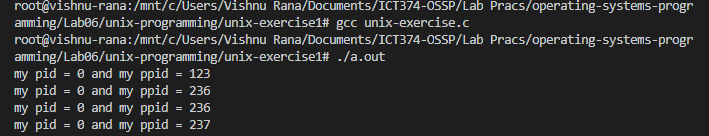
fork();

fork();

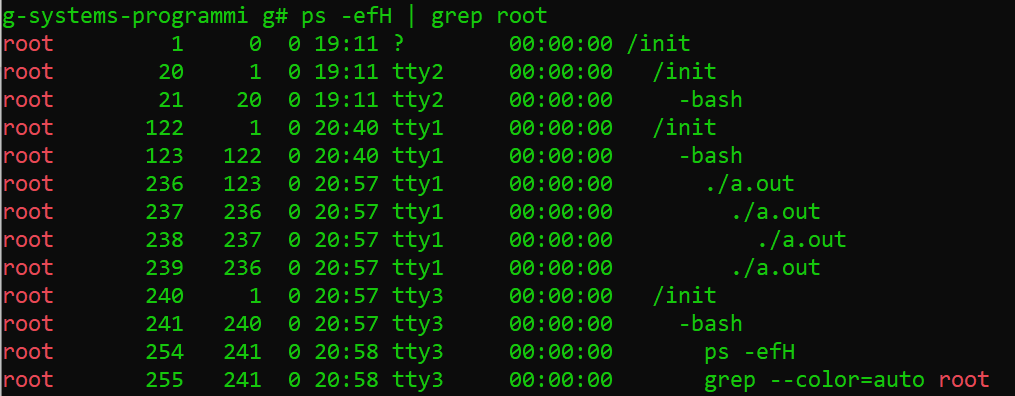
How many child processes would be created? What are the relationships between these processes? Complete the following program:

While running the above program in one terminal, display the processes in another terminal. What ps options would you need to show these processes and their parent-child relationship?

**Program output:**



**Another terminal output:**



1. **Creating multiple direct child processes**

Write a program that creates *n* direct child processes. For *i*th child, it prints "Child i" and its pid and ppid numbers. It then sleeps for *i* minutes before termination.

In creating the child processes you must make sure that all children are the direct children of the original process. Use command ps to verify that this is indeed the case.

Hint: consider the following program structure:

for (i=1; i<=n; ++i) {

pid = fork();

if (pid==0) { // child

print Child i: my pid and my ppid

sleep (i\*60); // sleep for i minutes

exit(0);

}

}

Make sure you genuinely understand how the above code works.

Test your program with different *n* values (e.g., 3, 4, 5) and verify the parent-child relationship with the program outputs as well as the output of the ps command.