Ex. No: 5

Date: 8/2/25

System Calls Programming

AIM:

To experiment system calls using fork(), execlp() and pid() functions.

ALGORITHM:

1. **Start**
2. **Include Header Files**
    - Include stdio.h for input/output functions
    - Include stdlib.h for general utility functions
3. **Variable Declaration**
    - Declare an integer variable pid to store the process ID returned by fork()
4. **Create a New Process**
    - Call the fork() function and assign its return value to pid
        - If fork() returns:
            - -1: Process creation failed
            - 0: This is the **child** process
            - A positive integer: This is the **parent** process
5. **Print Statement Executed by Both Processes**
    - Print: "THIS LINE EXECUTED TWICE"
6. **Check for Process Creation Failure**
    - If pid == -1:
        - Print: "CHILD PROCESS NOT CREATED"

        - Exit the program using exit(0)

7. **Child Process Execution Block**
    - If pid == 0:
        - Print:
            - "Process ID of child: " followed by getpid()
            - "Parent Process ID of child: " followed by getppid()
8. **Parent Process Execution Block**
    - If pid > 0:
        - Print:
            - "Process ID of parent: " followed by getpid()
            - "Parent’s Parent Process ID: " followed by getppid()
9. **Final Print Statement (Executed by Both Processes)**

- Print: objectives

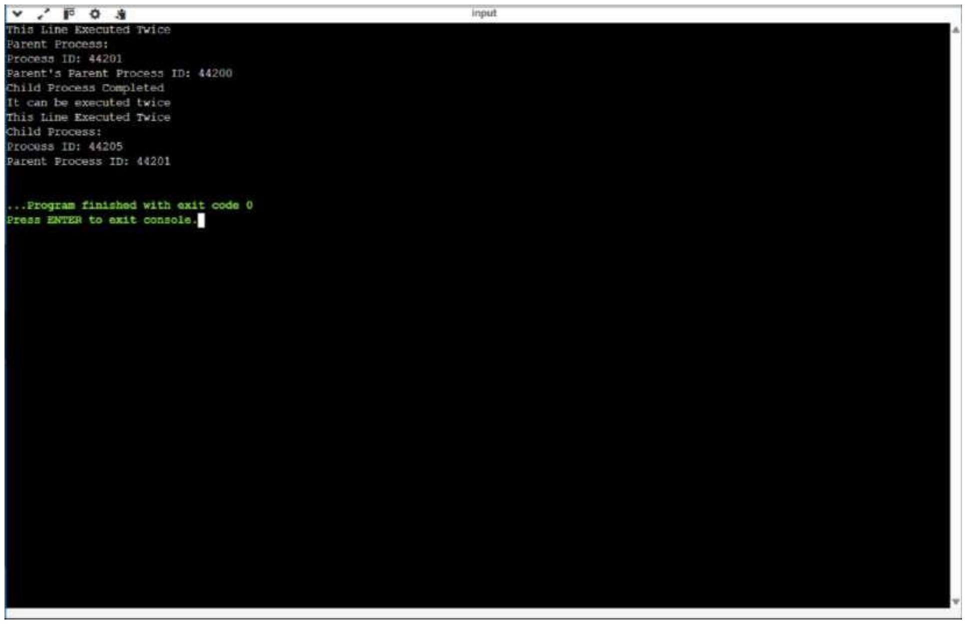
IT CAN BE EXECUTED TWICE

10. **End**

PROGRAM:

```
#include <stdio.h> #include <stdlib.h> #include <unistd.h>

int main() {
int pid;
pid = fork();
printf("This Line Executed Twice\n");

if (pid < 0) {
printf("Child Process Not Created\n"); exit(1);
}

if (pid == 0) {
printf("Child Process:\n"); printf("Process ID: %d\n", getpid());
printf("Parent Process ID: %d\n", getppid()); execlp("/bin/ls", "ls", NULL); perror("execlp failed");
exit(1);
} else { // Parent process
printf("Parent Process:\n"); printf("Process ID: %d\n", getpid());
printf("Parent's Parent Process ID: %d\n", getppid()); printf("Child Process Completed\n");
}
```

```c
printf("It can be executed twice\n");


return 0;

}
```



```
This Line Executed Twice
Parent Process:
Process ID: 44201
Parent's Parent Process ID: 44200
Child Process Completed
It can be executed twice
This Line Executed Twice
Child Process:
Process ID: 44205
Parent Process ID: 44201


...Program finished with exit code 0
Press ENTER to exit console.
```