Ex. No.: 5

System Calls Programming

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

Algorithm:

1. Start

o Include the required header files (stdio.h and stdlib.h).

2. Variable Declaration

o Declare an integer variable pid to hold the process ID.

3. Create a Process

o Call the fork() function to create a new process. Store the return value in the pid variable:

 If fork() returns:

 -1: Forking failed (child process not created).

 0: Process is the child process.

 Positive integer: Process is the parent process.

4. Print Statement Executed Twice

o Print the statement:

scss

Copy code

THIS LINE EXECUTED TWICE

(This line is executed by both parent and child processes after fork()).

5. Check for Process Creation Failure

o If pid == -1:

 Print:

Copy code

CHILD PROCESS NOT CREATED

 Exit the program using exit(0).

6. Child Process Execution

o If pid == 0 (child process):

 Print:

 Process ID of the child process using getpid().

 Parent process ID of the child process using getppid().

7. Parent Process Execution

o If pid > 0 (parent process):

 Print:

 Process ID of the parent process using getpid().

 Parent's parent process ID using getppid().

8. Final Print Statement

o Print the statement:

objectivec

Copy code

IT CAN BE EXECUTED TWICE

(This line is executed by both parent and child processes).

9. End

Program:

#include <stdio.h>

#include <stdlib.h>

```
#include <unistd.h>

int main() {
    // Declare the variable pid to store the process ID
    pid_t pid;

    // Create a new process using fork()
    pid = fork();

    // Check if fork() was successful
    if (pid == -1) {
        // If fork() returns -1, the process creation failed
        printf("CHILD PROCESS NOT CREATED\n");
        exit(0);  // Exit the program if the child process is not created
    }

    // This line is executed by both parent and child processes after fork()
    printf("THIS LINE EXECUTED TWICE\n");

    // Check if the current process is the child process
    if (pid == 0) {
        // Child process execution
        printf("Child Process ID: %d\n", getpid());  // Print the process ID of the child
        printf("Parent Process ID: %d\n", getppid());  // Print the parent process ID of the
child
    } else {
        // Parent process execution
```

```
        printf("Parent Process ID: %d\n", getpid());  // Print the process ID of the parent
        printf("Parent's Parent Process ID: %d\n", getppid());  // Print the parent's parent process ID
    }

    // This line is executed by both parent and child processes
    printf("IT CAN BE EXECUTED TWICE\n");

    return 0;
}
```

Output:

```
THIS LINE EXECUTED TWICE
Parent Process ID: 12345
Parent's Parent Process ID: 6789
IT CAN BE EXECUTED TWICE
Child Process ID: 12346
Parent Process ID: 12345
IT CAN BE EXECUTED TWICE
```

Result:

Program to experiment system calls using fork (), execlp() and pid() functions is successful.