

**EXP NO:** **5 SYSTEM CALLS PROGRAMMING**

**DATE:10/2/25**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main() {

    // Declare an integer variable pid to store the process ID

    pid\_t pid;

    // Create a new process using fork()

    pid = fork();

    // Print the statement executed twice (both parent and child process)

    printf("THIS LINE EXECUTED TWICE\n");

    // Check if fork() failed (pid == -1)

    if (pid == -1) {

        printf("CHILD PROCESS NOT CREATED\n");

        exit(0);  // Exit if the process creation fails

    }

    // If the process is the child (pid == 0)

    if (pid == 0) {

        // Print the process ID of the child and the parent process ID of the child

        printf("Child Process ID: %d\n", getpid());

        printf("Parent Process ID of Child: %d\n", getppid());

        // Optionally, we could use execlp() to execute a new program, e.g., /bin/ls

        // execlp("/bin/ls", "ls", (char \*) NULL);  // Uncomment if you want to run 'ls'

    }

    // If the process is the parent (pid > 0)

    else {

        // Print the process ID of the parent and the parent's parent process ID

        printf("Parent Process ID: %d\n", getpid());

        printf("Parent's Parent Process ID: %d\n", getppid());

    }

    // Final print statement executed by both parent and child

    printf("IT CAN BE EXECUTED TWICE\n");

    return 0;

}

**OUTPUT:**

THIS LINE EXECUTED TWICE

Parent Process ID: 2306

Parent's Parent Process ID: 2299

IT CAN BE EXECUTED TWICE

THIS LINE EXECUTED TWICE

Child Process ID: 2307

Parent Process ID of Child: 2306

IT CAN BE EXECUTED TWICE