Ex. No.: 5
Date:

System Calls Programming

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

Algorithm:

1. **Start**
   - Include the required header files (stdio.h and stdlib.h).
2. **Variable Declaration**
   - Declare an integer variable pid to hold the process ID.
3. **Create a Process**
   - 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**
   - 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**
   - If pid == -1:
     - Print:

       Copy code
       CHILD PROCESS NOT CREATED

     - Exit the program using exit(0).
6. **Child Process Execution**
   - 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**
   - 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**
   - Print the statement:

     objectivec

Copy code
IT CAN BE EXECUTED TWICE

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

9. **End**

Program:

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

int main() {
    int pid;

    pid = fork();

    if (pid == -1) {
        printf("CHILD PROCESS NOT CREATED\n");
        exit(0);
    }

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

    if (pid == 0) {
        printf("Child Process:\n");
        printf("Process ID of the child: %d\n", getpid());
        printf("Parent Process ID of the child: %d\n", getppid());
    }
    else if (pid > 0) {
        printf("Parent Process:\n");
        printf("Process ID of the parent: %d\n", getpid());
        printf("Parent's Parent Process ID: %d\n", getppid());
    }

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

    return 0;
}
```

Output:

```
THIS LINE EXECUTED TWICE
Parent Process:
Process ID of the parent: 65244
Parent's Parent Process ID: 63696
IT CAN BE EXECUTED TWICE
THIS LINE EXECUTED TWICE
Child Process:
Process ID of the child: 65245
Parent Process ID of the child: 1
IT CAN BE EXECUTED TWICE
```

Result:

System calls `fork()`, `exec()`, `getpid()`, `opendir()`, and `readdir()` have been successfully implemented, demonstrating the ability to create processes, execute programs, retrieve process IDs, and navigate directories.