Ex. No.: 5
Date: 13/2/25

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:

```
#include <stdio.h>
#  include <stdlib.h>
#  include <unistd.h>
int main (){
 int pid;
 Pid = fork ();
 Printf ("THIS LINE EXECUTED TWICE");
 if (pid = = -1){
     Printf ("\n CHILD PROCESS NOT CREATED\n");
  exit (0);
 }
 if (Pid==0){
    Printf ("\n I am CHILD PROCESS AND MY ID
                                IS %d\n", getpid ())
    Print ("\n I AM CHILD PARENT PROCESS IDIS
                                %d \n", getppid());
}
Else {
    Printf("\n I AM PARENT PROCESS AND MY
                        ID IS : %d \n", getpid ());
```

```
Printf("\n the parent process ID is : %d\n";
}
Printf("\n It can be executed Twice");
Printf("\n");
}
```

Output:

```
This line executed twice
I am parent process and Pd is : 1718
The parent process Pd is : 1718
It can be executed twice
This line executed twice
I am child process and my Pd is 1719
The child parent process id is : 1719
```

Result:

Hence the program is executed in System call using fork(), execlp() and pid() function