Ex. no: 5)

Name : Ashish P Shaji

Roll NO : 230701041

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:

Objective:

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 <sys/types.h>
#include <unistd.h>
int main(){
  pid_t pid;
  pid=fork();
  printf("This line executed twice\n");
  if( pid == -1){
    printf("Child proceess not created\n");
    exit(0);
}
 if(pid == 0){
   printf("parent: getpid:%d,getppid= %d\n",getpid(), getppid());
}
  else{
   printf("child: getpid=%d, getppid=%d\n",getpid(),getppid());
}
printf("IT can be executed twice\n");
}
```

OUTPUT :

```
This line executed twice
child: getpid=2945, getppid=1373
IT can be executed twice
This line executed twice
parent: getpid:2946,getppid= 1
IT can be executed twice
```