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

IT CAN BE EXECUTED TWICE

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

9. End

Program:

```c
# include <stdio.h>.
# include < unistd.h>
# include < stdlib.h>
int main()
{
    int pid;
    pid = fork();
    printf("\n 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());
        printf("\n THE CHILD PARENT PROCESS ID IS: %d \n", getppid());
    }
    else
    {   printf("\nI AM PARENT PROCESS AND ID IS %d \n", getpid());
        printf("\n THE PARENTS PARENT PROCESS ID IS: %d \n", getppid());
    }
    printf("\n It CAN BE EXECUTED TWICE ");
    printf("\n");
}.
```

33

Output: With <unistd.h>.

This line executed twice.

I'm parent process and MY ID is : 1829.

the parents parent process ID is : 1582

I can be executed twice

this line executed twice

I am child - process and my ID is : 1829.

the child process ID is : 1828.

Result:

Thus the script for system calls using fork exec()p.

and pid() are executed and obtained.