**PRACTICAL NO.2**

Aim :- To Write a C Program to implement System calls getpid() & getppid() to return ID.

Theory:- There are 2 functions to get process id in C:

1. getpid;
2. getppid;

every process that gets executed in computer has its process id which can be obtained by using this functions.

Both getppid() and getpid() are inbuilt functions defined in unistd.h library.

**getppid()** : returns the process ID of the parent of the calling process. If the calling process was created by the [fork()](https://www.geeksforgeeks.org/fork-system-call/) function and the parent process still exists at the time of the getppid function call, this function returns the process ID of the parent process. Otherwise, this function returns a value of 1 which is the process id for init process.

**Return type:** getppid() returns the process ID of the parent of the current process. It never throws any error therefore is always successful.

**getpid() :** returns the process ID of the calling process. This is often used by routines that generate unique temporary filenames.

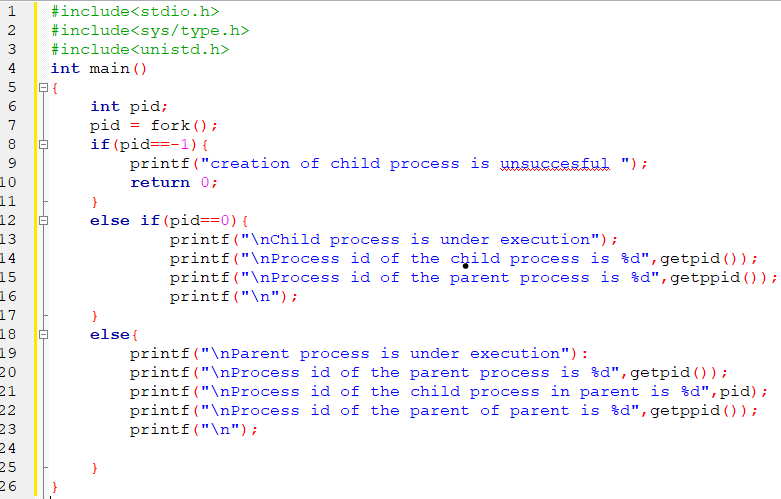
**Return type:** getpid() returns the process ID of the current process. It never throws any error therefore is always successful.

Fork system call is used for creating a new process, which is called *child process*, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call. A child process uses the same pc(program counter), same CPU registers, same open files which use in the parent process.

It takes no parameters and returns an integer value. Below are different values returned by fork().

*Negative Value*: creation of a child process was unsuccessful.  
*Zero*: Returned to the newly created child process.  
*Positive value*: Returned to parent or caller. The value contains process ID of newly created child p

*Program:*

**

*Output:*

Parent process is under execution. Process id of the parent process is 14251

Process id of the child process is 14252

lindro

Process id of the parent of parent is 14228

Child process is under execution

Process id of the child process is 14252 Process id of the parent process is 14251