NAME: G. GANESH

REG NO: 192373008

EXERICSE-1

Create a new process by invoking the appropriate system call. Get the process identifier of the currently running process and its respective parent using system calls and display the same using a C program.

AIM:

To create a new process using the fork() system call in C, retrieve the process identifier (PID) of the currently running process and its parent process, and display these identifiers**.**

Algorithm:

1. Create a new process using the fork() system call.
2. Retrieve the PID of the current process using the getpid() system call.
3. Retrieve the PPID of the current process using the getppid() system call.
4. Display the PID and PPID values.

Procedure:

1. Use the fork() system call to create a new process.
2. The fork() function returns a value:
   - in the child process.
   - Positive value (child's PID) in the parent process.
3. In both processes, use getpid() to retrieve the PID and getppid() to retrieve the PPID.
4. Display the PID and PPID values**.**

Code:

```
#include <stdio.h>
#include <unistd.h>
int main() {
   pid_t pid = fork();
   if (pid < 0) {
      printf("Fork failed\n");
      return 1;
   }
   printf("PID: %d\n", getpid());
```

```
    printf("PPID: %d\n", getppid());
    return 0;
}
```

Result:

1. The program successfully creates a new process using fork().
2. It displays the process ID (PID) and parent process ID (PPID) of both the parent and child processes.
3. The output confirms the relationship between the parent and child processes.

Output:

```
PID: 3361
PPID: 3354
PID: 3362
PPID: 3361
```