

System Calls Programming

Program:

```
#include <stdio.h>

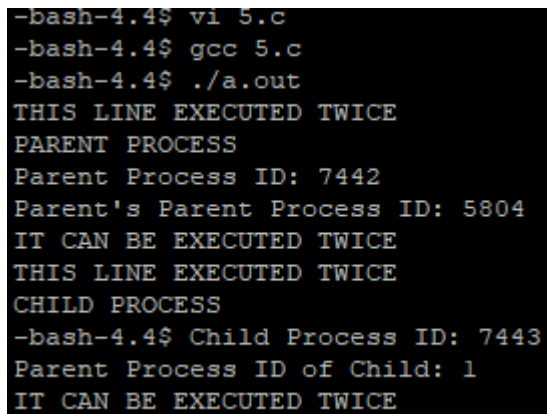
#include <stdlib.h>

#include <unistd.h>

int main() {
    pid_t pid;
    pid = fork();
    if (pid == -1) {
        printf("CHILD PROCESS NOT CREATED\n");
        exit(0); // Exit if forking failed
    }
    printf("THIS LINE EXECUTED TWICE\n");
    if (pid == 0) {
        // Child process execution
        printf("CHILD PROCESS\n");
        printf("Child Process ID: %d\n", getpid());
        printf("Parent Process ID of Child: %d\n", getppid());
    }
    else {
        printf("PARENT PROCESS\n");
        printf("Parent Process ID: %d\n", getpid());
        printf("Parent's Parent Process ID: %d\n", getppid());
    }
}
```

```
printf("IT CAN BE EXECUTED TWICE\n");  
  
return 0;  
  
}
```

OUTPUT :

A terminal window with a black background and white text. The text shows the execution of a C program. The user runs 'vi 5.c', 'gcc 5.c', and './a.out'. The program outputs 'THIS LINE EXECUTED TWICE' twice. Then it prints 'PARENT PROCESS', 'Parent Process ID: 7442', and 'Parent's Parent Process ID: 5804'. It then prints 'IT CAN BE EXECUTED TWICE' and 'THIS LINE EXECUTED TWICE'. Next, it prints 'CHILD PROCESS', 'Child Process ID: 7443', and 'Parent Process ID of Child: 1'. Finally, it prints 'IT CAN BE EXECUTED TWICE' again. A green cursor is visible at the end of the last line.

```
-bash-4.4$ vi 5.c  
-bash-4.4$ gcc 5.c  
-bash-4.4$ ./a.out  
THIS LINE EXECUTED TWICE  
PARENT PROCESS  
Parent Process ID: 7442  
Parent's Parent Process ID: 5804  
IT CAN BE EXECUTED TWICE  
THIS LINE EXECUTED TWICE  
CHILD PROCESS  
-bash-4.4$ Child Process ID: 7443  
Parent Process ID of Child: 1  
IT CAN BE EXECUTED TWICE
```