

Expr 5: System Calls Programming

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h> // For fork(), getpid(), getppid(), execlp()

int main() {
    int pid;

    pid = fork(); // Create a new process

    // Step 4: This line runs in both processes
    printf("THIS LINE EXECUTED TWICE\n");

    // Step 5: Fork failed
    if (pid == -1) {
        printf("CHILD PROCESS NOT CREATED\n");
        exit(0);
    }

    // Step 6: Child process
    if (pid == 0) {
        printf("Child Process:\n");
        printf("Process ID (PID): %d\n", getpid());
        printf("Parent Process ID (PPID): %d\n", getppid());

        // Uncomment below if you want child to replace itself with another
        // process (e.g., `ls`)
        // execlp("ls", "ls", "-l", NULL);
    }

    // Step 7: Parent process
    if (pid > 0) {
        printf("Parent Process:\n");
        printf("Process ID (PID): %d\n", getpid());
        printf("Parent's Parent ID (PPID): %d\n", getppid());
    }

    // Step 8: Final line
    printf("IT CAN BE EXECUTED TWICE\n");

    return 0;
}
```

Output:

THIS LINE EXECUTED TWICE

Parent Process:

Process ID (PID): 12345

Parent's Parent ID (PPID): 6789

IT CAN BE EXECUTED TWICE

Child Process:

Process ID (PID): 12346

Parent Process ID (PPID): 12345

IT CAN BE EXECUTED TWICE

Result:

Thus the System Calls ProgrammingCode is implemented in fedora using the C language