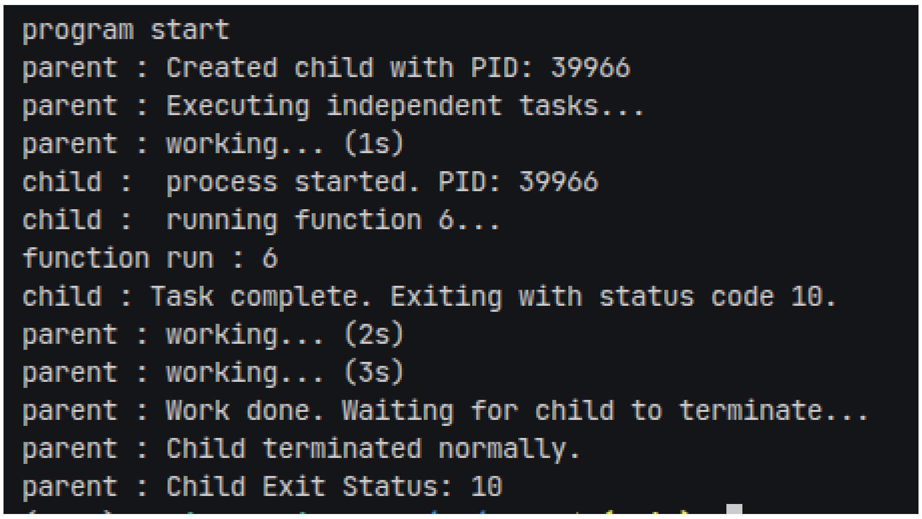
Assignment 2

```
Name : Ravish Ranjan
Course : MCA
Semester : 2nd semester
```

```
Write a C++ program to demonstrate process creation and synchronization
using fork(). After
creating a child process, the parent process should continue executing
independently and
display a message every second for a fixed number of times. The child
process should compute
and display the factorial of a given number and then terminate using exit()
with an appropriate
status code. Ensure that the parent process waits for the child process
using wait() and prints
whether the child terminated normally along with its exit status.
```

Ans

```cpp
#include <iostream>
#include <unistd.h>
#include <sys/wait.h>
#include <cstdlib>
using namespace std;

void func(int num){
    std::cout << "function run : " << num << std::endl;
}

int main(){
    int number = 6;
    int waitStatus;
    cout << "program start" << endl;

    pid_t pid = fork();
    if (pid < 0) {
        cerr << "Fork failed!" << endl;
        return 1;
    }
    else if (pid == 0) {
        cout << "child :  process started. PID: " << getpid() << endl;
        cout << "child :  running function " << number << "..." << endl;

        func(number);

        cout << "child : Task complete. Exiting with status code 10." <<
endl;
        exit(10);
    }
    else {
        cout << "parent : Created child with PID: " << pid << endl;
        cout << "parent : Executing independent tasks..." << endl;

        for (int i = 1; i <= 3; ++i) {
            cout << "parent : working... (" << i << "s)" << endl;
            sleep(1);
        }

        cout << "parent : Work done. Waiting for child to terminate..." <<
endl;

        wait(&waitStatus);
        if (WIFEXITED(waitStatus)) {
            int exitCode = WEXITSTATUS(waitStatus);
            cout << "parent : Child terminated normally." << endl;
            cout << "parent : Child Exit Status: " << exitCode << endl;
        } else {
            cout << "parent : Child terminated abnormally." << endl;
        }
    }
    return 0;
}
```



```
program start
parent : Created child with PID: 39966
parent : Executing independent tasks...
parent : working... (1s)
child :  process started. PID: 39966
child :  running function 6...
function run : 6
child : Task complete. Exiting with status code 10.
parent : working... (2s)
parent : working... (3s)
parent : Work done. Waiting for child to terminate...
parent : Child terminated normally.
parent : Child Exit Status: 10
```