UNIVERSITY OF LAYYAH

DEPARTMENT OF COMPUTER SCIENCE

Operating system lab manual

Name : Bakhtawar Saleem

Roll no. : 31

Semester : 5th sec(A)

Course Title : Operating System

Course Instructor: Sir Umar Daraz

Experiment Title : Round Robin in Linux(Ubantu)

Round Robin CPU Scheduler in C++ (Ubuntu Guide)

1. Introduction

This manual guides you through:

- Creating a C++ source file
- Writing the Round Robin Scheduler code
- Compiling the program using g++
- Running the final executable
- Understanding the output

2.System Requirements

Before starting, make sure your Ubuntu system has: • Ubuntu OS (any LTS version recommended)

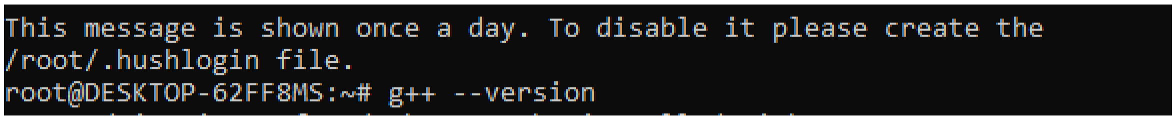
• Terminal (default GNOME Terminal is fine)

• G++ compiler

• A text editor (nano or VS Code)

To check if g++ is installed,

run:



```
This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@DESKTOP-62FF8MS:~# g++ --version
```

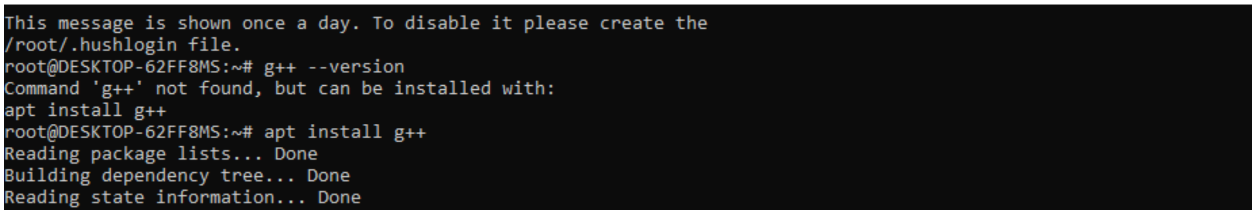
g++ --version

If it shows a version, you are ready.

If not, install it using:

sudo apt update

sudo apt install g++



```
This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@DESKTOP-62FF8MS:~# g++ --version
Command 'g++' not found, but can be installed with:
apt install g++
root@DESKTOP-62FF8MS:~# apt install g++
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

3. Creating the Project Directory

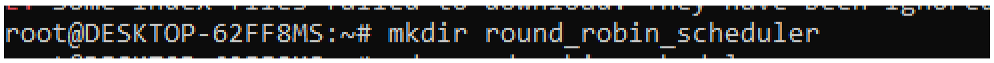
It is recommended to store your project in a separate folder

Step 1: Open the Terminal Press:

CTRL + ALT + T

Step2: Create a directory

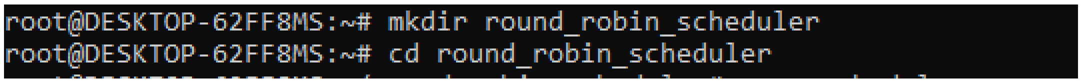
mkdir round_robin_scheduler



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
```

Step 3:

Enter the directory cd round_robin_scheduler



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
```

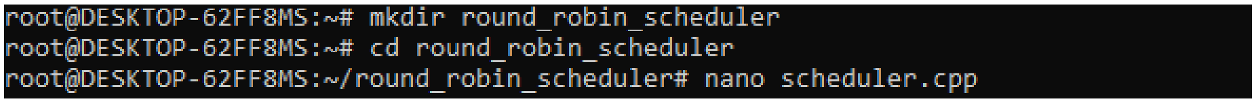
4. Creating the C++ Source File

Step 1:

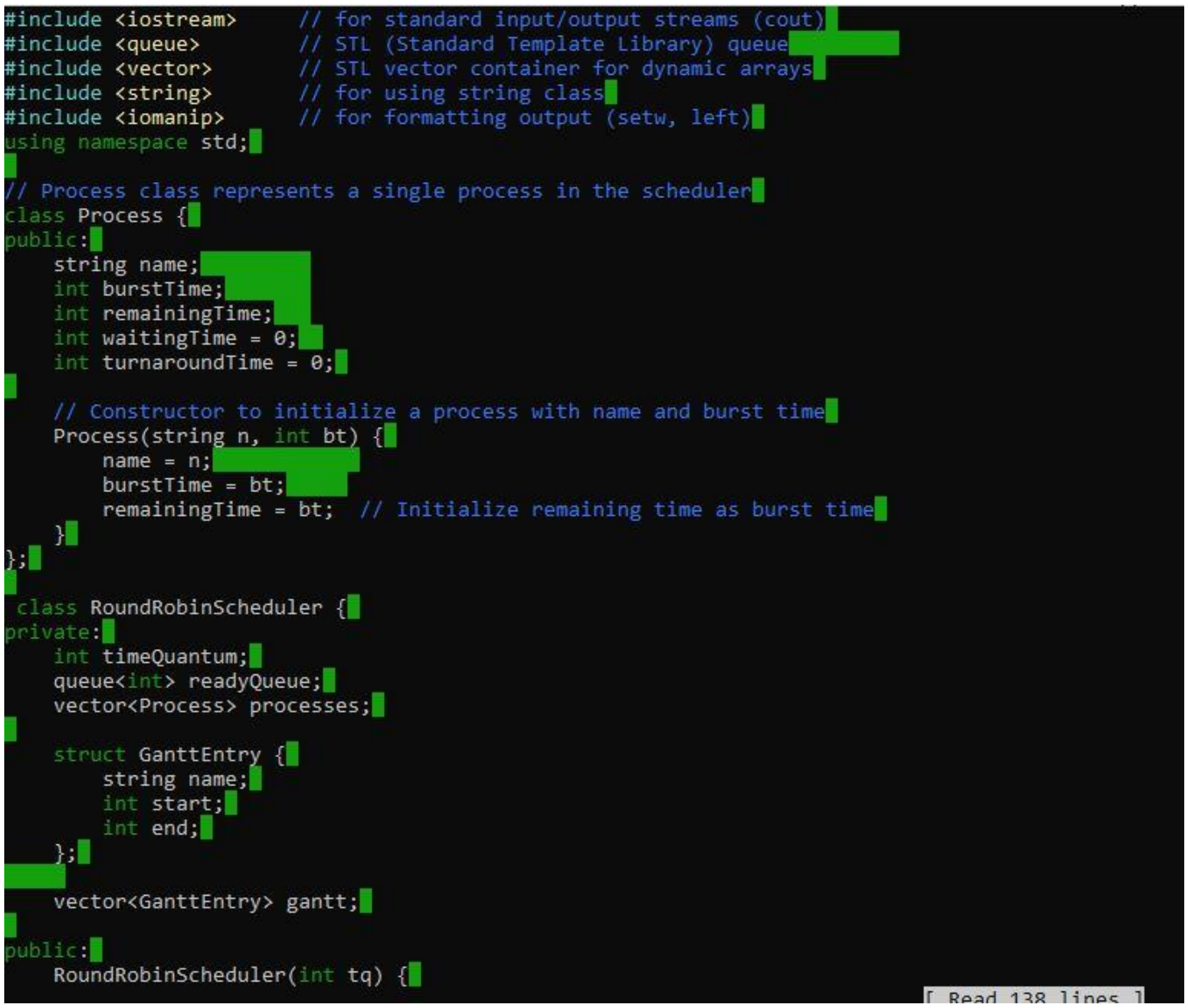
Create a new file Use nano:

Compiling the Program Step 1: Compile using g++ Inside the same folder, run:

nano scheduler.cpp

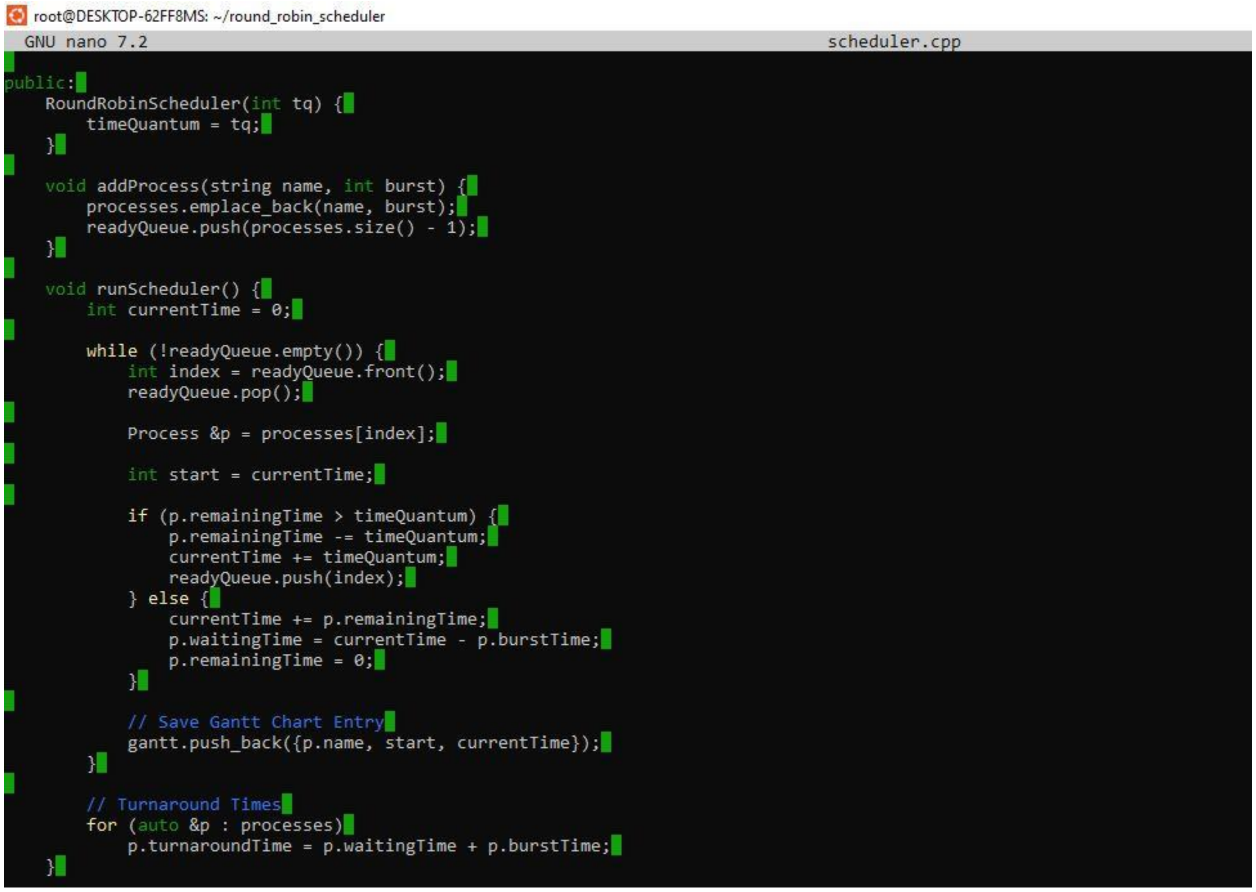


```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# nano scheduler.cpp
```

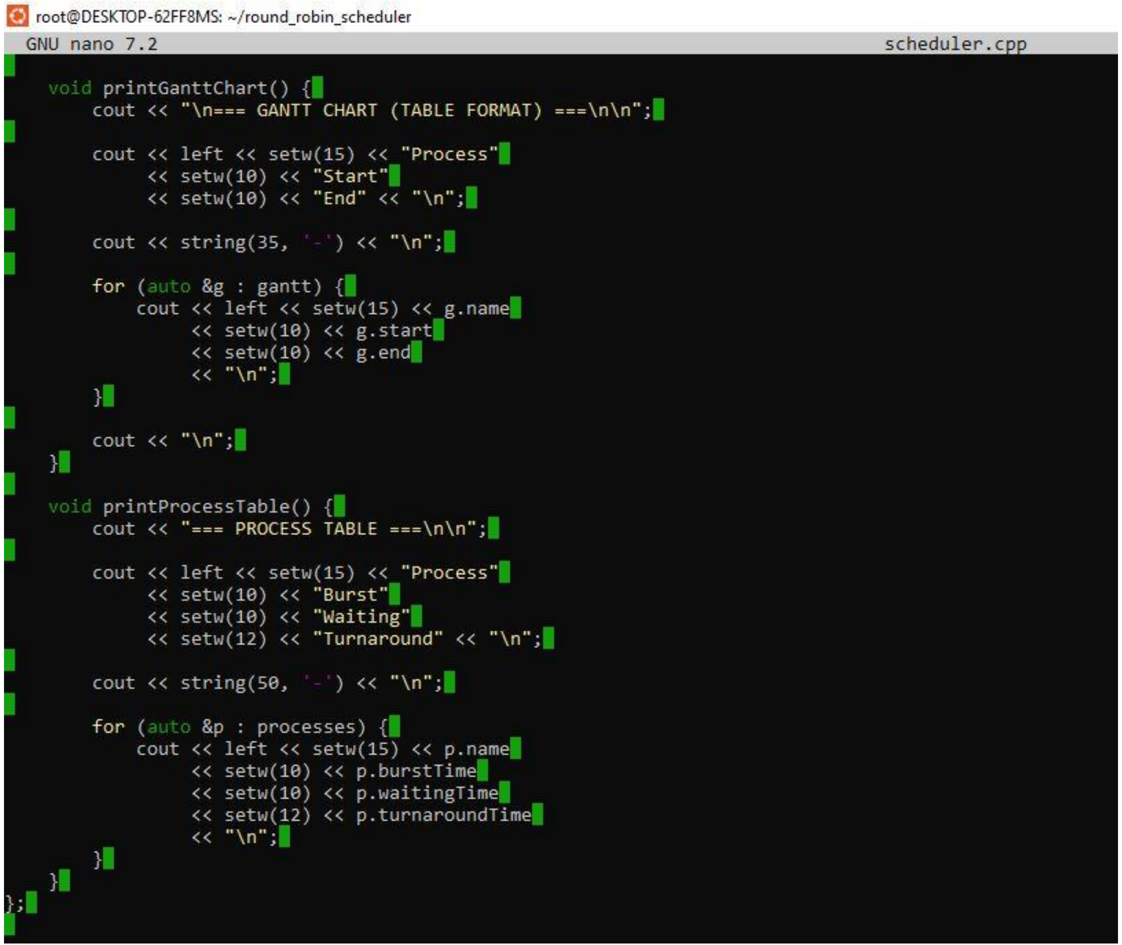


```
#include <iostream>     // for standard input/output streams (cout)
#include <queue>        // STL (Standard Template Library) queue
#include <vector>       // STL vector container for dynamic arrays
#include <string>       // for using string class
#include <iomanip>      // for formatting output (setw, left)
using namespace std;

// Process class represents a single process in the scheduler
class Process {
public:
    string name;
    int burstTime;
    int remainingTime;
    int waitingTime = 0;
    int turnaroundTime = 0;

    // Constructor to initialize a process with name and burst time
    Process(string n, int bt) {
        name = n;
        burstTime = bt;
        remainingTime = bt;  // Initialize remaining time as burst time
    }
};

 class RoundRobinScheduler {
private:
    int timeQuantum;
    queue<int> readyQueue;
    vector<Process> processes;

    struct GanttEntry {
        string name;
        int start;
        int end;
    };

    vector<GanttEntry> gantt;

public:
    RoundRobinScheduler(int tq) {
                                                           [ Read 138 lines ]
```

Code


root@DESKTOP-62FF8MS: ~/round_robin_scheduler

```
  GNU nano 7.2                                                     scheduler.cpp

public:
    RoundRobinScheduler(int tq) {
        timeQuantum = tq;
    }

    void addProcess(string name, int burst) {
        processes.emplace_back(name, burst);
        readyQueue.push(processes.size() - 1);
    }

    void runScheduler() {
        int currentTime = 0;

        while (!readyQueue.empty()) {
            int index = readyQueue.front();
            readyQueue.pop();

            Process &p = processes[index];

            int start = currentTime;

            if (p.remainingTime > timeQuantum) {
                p.remainingTime -= timeQuantum;
                currentTime += timeQuantum;
                readyQueue.push(index);
            } else {
                currentTime += p.remainingTime;
                p.waitingTime = currentTime - p.burstTime;
                p.remainingTime = 0;
            }

            // Save Gantt Chart Entry
            gantt.push_back({p.name, start, currentTime});
        }

        // Turnaround Times
        for (auto &p : processes)
            p.turnaroundTime = p.waitingTime + p.burstTime;
    }
```

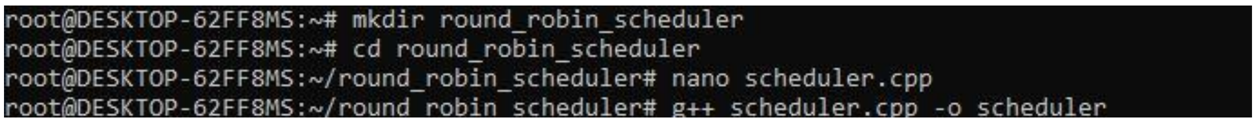

root@DESKTOP-62FF8MS: ~/round_robin_scheduler

```
  GNU nano 7.2                                                     scheduler.cpp

    void printGanttChart() {
        cout << "\n=== GANTT CHART (TABLE FORMAT) ===\n\n";

        cout << left << setw(15) << "Process"
             << setw(10) << "Start"
             << setw(10) << "End" << "\n";

        cout << string(35, '-') << "\n";

        for (auto &g : gantt) {
            cout << left << setw(15) << g.name
                 << setw(10) << g.start
                 << setw(10) << g.end
                 << "\n";
        }

        cout << "\n";
    }

    void printProcessTable() {
        cout << "=== PROCESS TABLE ===\n\n";

        cout << left << setw(15) << "Process"
             << setw(10) << "Burst"
             << setw(10) << "Waiting"
             << setw(12) << "Turnaround" << "\n";

        cout << string(50, '-') << "\n";

        for (auto &p : processes) {
            cout << left << setw(15) << p.name
                 << setw(10) << p.burstTime
                 << setw(10) << p.waitingTime
                 << setw(12) << p.turnaroundTime
                 << "\n";
        }
    }
};
```

Compiling the Program

Step 1: Compile using g++ Inside the same folder,

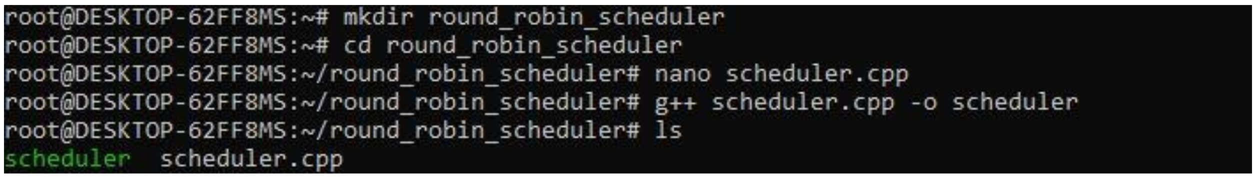
run:



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# nano scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# g++ scheduler.cpp -o scheduler
```

Step 2:

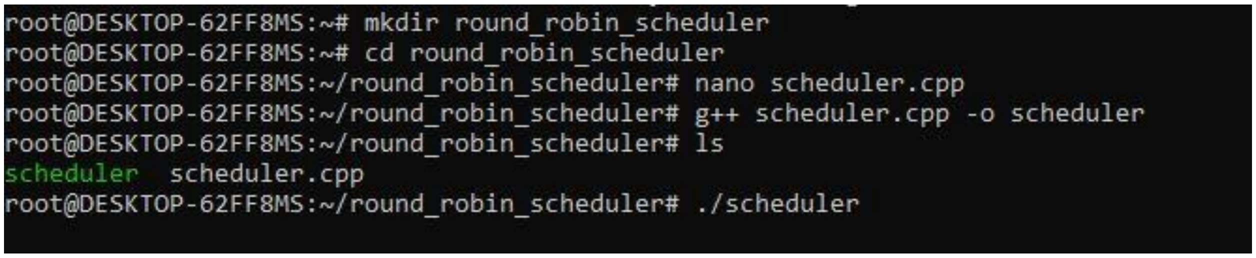
Check if executable was created Run:



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# nano scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# g++ scheduler.cpp -o scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# ls
scheduler  scheduler.cpp
```

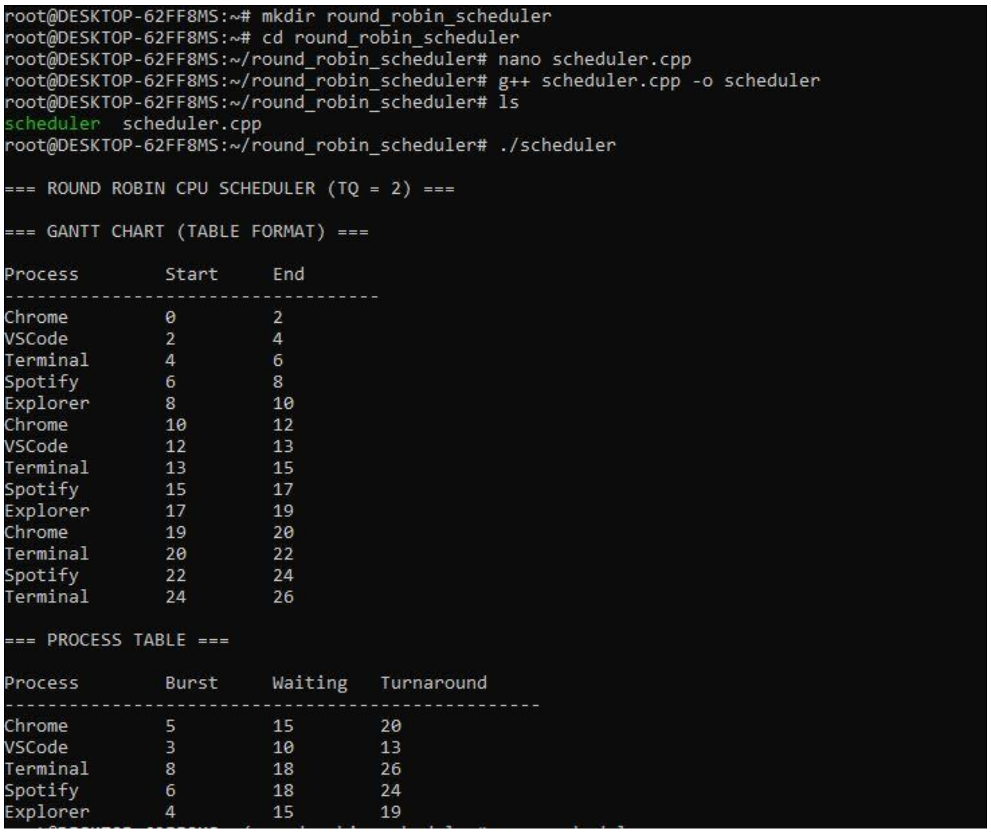
Running the Program

Step 1: Execute the compiled program



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# nano scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# g++ scheduler.cpp -o scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# ls
scheduler  scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# ./scheduler
```

Output :



```
root@DESKTOP-62FF8MS:~# mkdir round_robin_scheduler
root@DESKTOP-62FF8MS:~# cd round_robin_scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# nano scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# g++ scheduler.cpp -o scheduler
root@DESKTOP-62FF8MS:~/round_robin_scheduler# ls
scheduler  scheduler.cpp
root@DESKTOP-62FF8MS:~/round_robin_scheduler# ./scheduler

=== ROUND ROBIN CPU SCHEDULER (TQ = 2) ===

=== GANTT CHART (TABLE FORMAT) ===

Process        Start     End
-----------------------------------
Chrome         0         2
VSCode         2         4
Terminal       4         6
Spotify        6         8
Explorer       8         10
Chrome         10        12
VSCode         12        13
Terminal       13        15
Spotify        15        17
Explorer       17        19
Chrome         19        20
Terminal       20        22
Spotify        22        24
Terminal       24        26

=== PROCESS TABLE ===

Process        Burst     Waiting   Turnaround
--------------------------------------------------
Chrome         5         15        20
VSCode         3         10        13
Terminal       8         18        26
Spotify        6         18        24
Explorer       4         15        19
```