

Name: Vương Đình Thanh Ngân

ID: 20521649

Class: ATCL2022

## OPERATING SYSTEM LAB 1'S REPORT

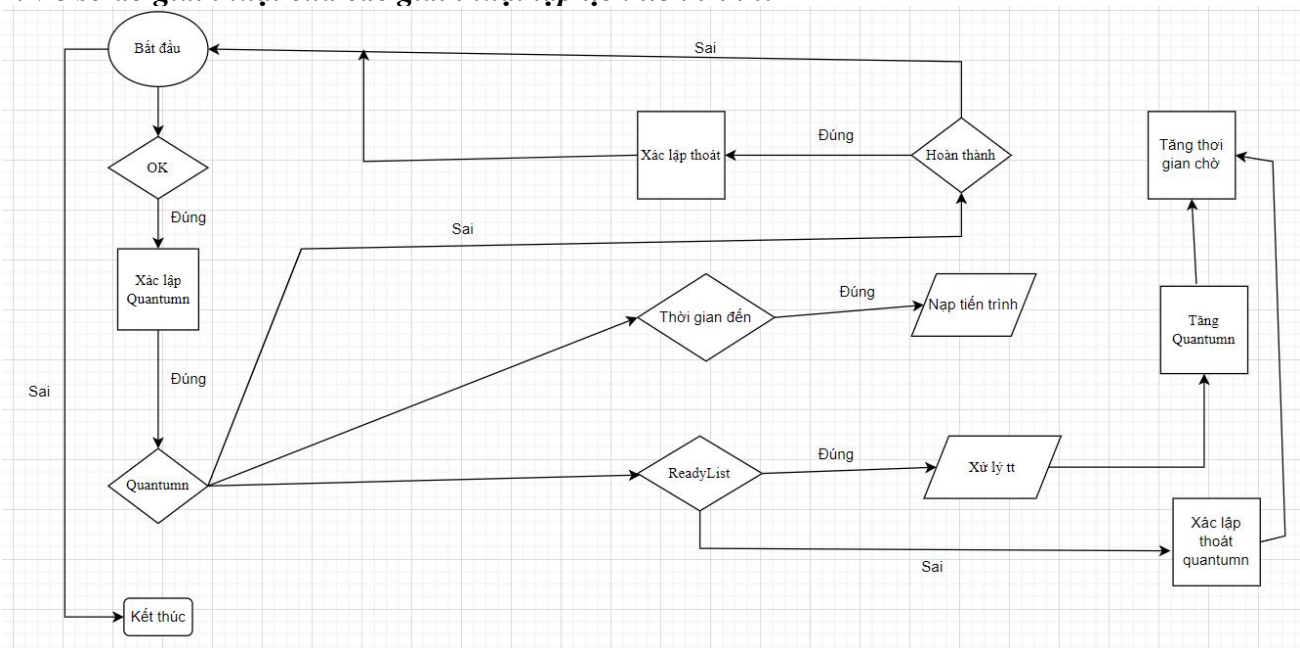
### SUMMARY

	Task	Status	Page
Section 4	Ex 4.3	Done	1, 2, 3, 4
	Ex 4.4	Done	4, 5, 6
	Ex 4.5	Done	6, 7, 8, 9, 10, 11

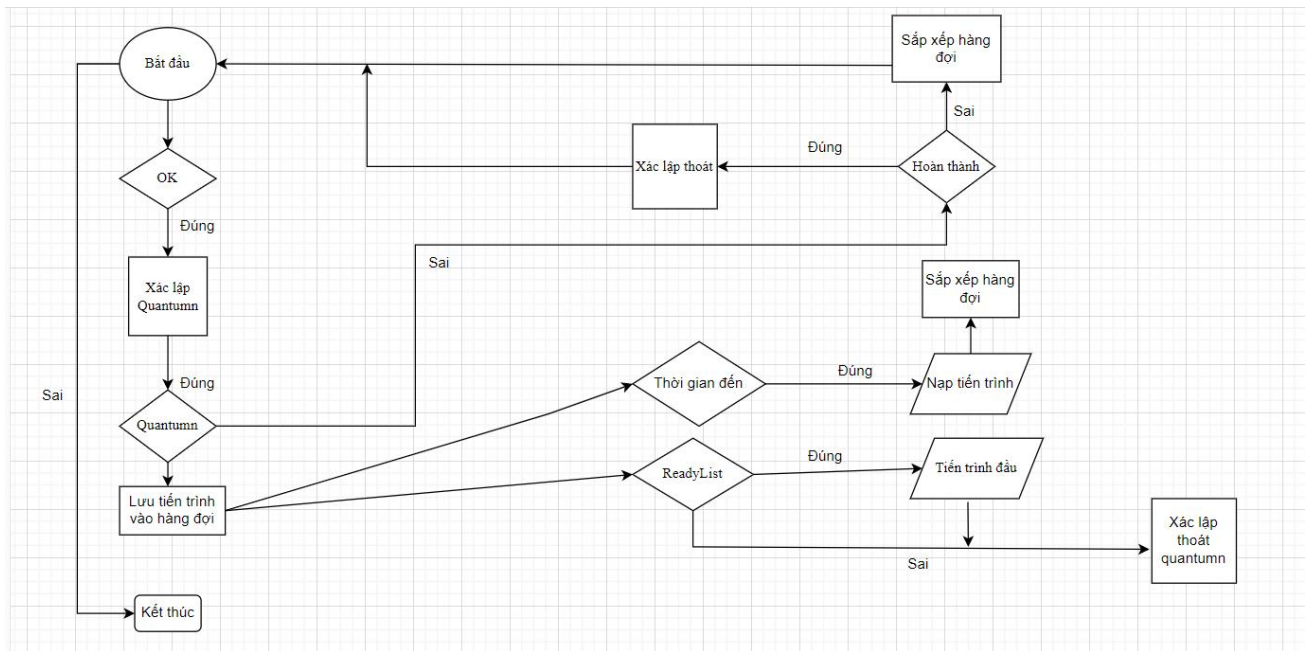
Self-scores:

### 4.3.5:

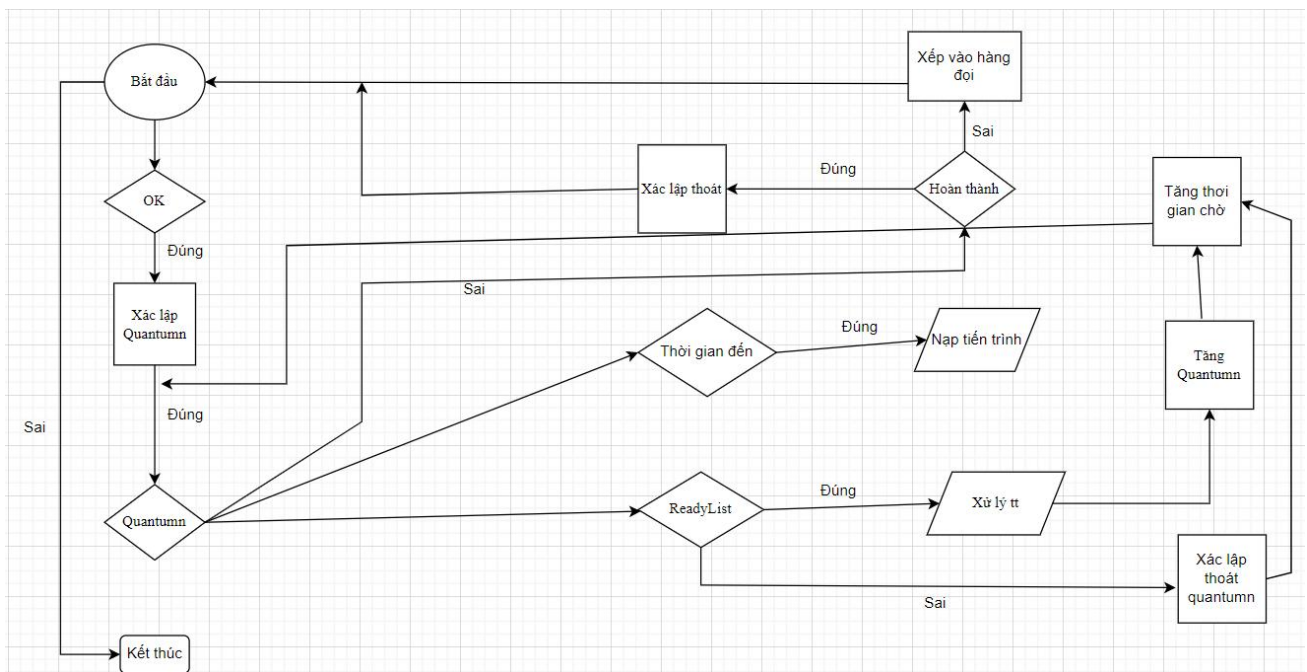
#### 1. Vẽ sơ đồ giải thuật của các giải thuật lập lịch tiến trình:



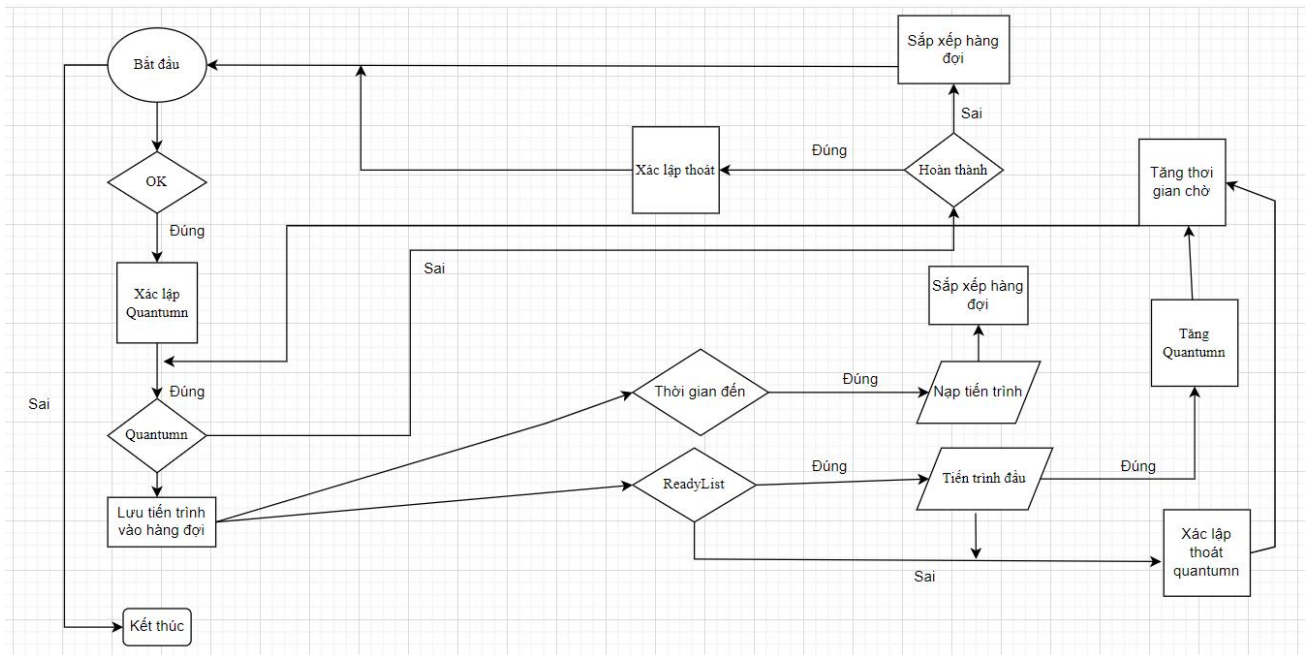
Hình 1.1: Lưu đồ giải thuật FCFS



Hình 1.2: Lưu đồ giải thuật RR



Hình 2.3: Lưu đồ giải thuật SJF



Hình 2.4: Lưu đồ giải thuật SRT

\* Thực hiện vẽ giản đồ Gantt:

Test case:

Process	Arrival time	Burst time
P1	8	11
P2	0	9
P3	4	7
P4	2	5
P5	6	3

+ FCFS

P2	P4	P3	P5	P1
0	9	12	17	24
				35

- Average Turnaround time: 15.4

- Average Waiting time: 8.4

+ RR với quantum = 6

P2	P4	P3	P5	P2	P1	P3	P1
0	6	11	17	20	23	29	30
							35

- Average Turnaround time: 19.8

- Average Waiting time: 12.8

+ SJF

P2	P5	P4	P3	P1
----	----	----	----	----

0                      9                      12                      17                      24                      35

- Average Turnaround time: 15.4

- Average Waiting time: 8.4

+SRT

P1	P4	P5	P2	P3	P1	
0	2	7	10	17	24	35

- Average Turnaround time: 14.6

- Average Waiting time: 7.6

## 2. Giải thích các thuật ngữ sau:

STT	Thuật ngữ	Mô tả
1	Arrival time	Thời gian đến - thời điểm tiến trình vào hàng đợi
2	Burst time	Thời gian thực thi - thời gian mà tiến trình cần để thực hiện xong công việc của nó
3	Quantum time	Khoảng thời gian mà một tiến trình có thể sử dụng CPU
4	Response time	Thời gian đáp ứng - khoảng thời gian process nhận yêu cầu đến khi yêu cầu đầu tiên được đáp ứng
5	Waiting time	Thời gian chờ - thời gian một process đợi trong ready queue
6	Turnaround time	Thời gian quay vòng (hoàn thành) - khoảng thời gian từ lúc một process được nạp vào hệ thống đến khi process đó kết thúc
7	Average waiting time	Thời gian chờ trung bình - khoảng thời gian trung bình tiến trình phải chờ đợi để được thực thi xong
8	Average turnaround time	Thời gian xoay vòng trung bình - khoảng thời gian trung bình chờ đợi để từ khi process được thực thi đến khi hoàn thành (thời gian lưu lại trung bình của tiến trình trong hệ thống)

## 4.4 Soạn thảo và biên dịch giải thuật FCFS bên dưới:

- Đoạn code cho bài

```

1 /*****
2 # University of Information Technology #
3 # IT007 Operating System #
4 10
5 # <Vuong Dinh Thanh Ngan, <20521649> #
6 # File: fcfs.c #
7 *****/
8 #include<stdio.h>
9 void main(){
10 int pn[10];
11 int arr[10], bur[10], star[10], finish[10], tat[10], wt[10], i, n;
12 int totwt=0, tottat=0;
13 printf("Enter the number of processes:");
14 scanf("%d",&n);
15 for(i=0;i<n;i++) {
16 printf("Enter the Process Name, Arrival Time & Burst Time:");
17 scanf("%d%d%d",&pn[i],&arr[i],&bur[i]);
18 }
19 for(i=0;i<n;i++) {
20 if(i==0) {
21 star[i]=arr[i];
22 wt[i]=star[i]-arr[i];
23 finish[i]=star[i]+bur[i];
24 tat[i]=finish[i]-arr[i];
25 } else{
26 11|
27 star[i]=finish[i-1];
28 wt[i]=star[i]-arr[i];
29 finish[i]=star[i]+bur[i];
30 tat[i]=finish[i]-arr[i];
31 }
32 }
33 printf("\nPName Arrtime Burtime Start TAT Finish");
34 for(i=0;i<n;i++) {
35 printf("\n%d\t%d\t%d\t%d\t%d\t%d",pn[i],arr[i],bur[i],star
36 [i],tat[i],finish[i]);
37 totwt+=wt[i];

```

Hình 4.4.1: Đoạn code theo yêu cầu

- Ta dùng câu lệnh gcc 4\_4.c -o4\_4 và ./4\_4 để thực thi
- Chạy theo yêu cầu của chương trình với các process cho sẵn:
  - + Tiến trình đầu tiên : 0 0 2
  - + Tiến trình thứ hai: 1 1 3
  - + Tiến trình thứ ba: 2 2 4

```

VuongDinhThanhNgan_20521649@ubuntu:~$ ./4_4
Enter the number of processes:3
Enter the Process Name, Arrival Time & Burst Time:0 0 2
Enter the Process Name, Arrival Time & Burst Time:1 1 3
Enter the Process Name, Arrival Time & Burst Time:2 2 4

PName Arrtime Burtime Start TAT Finish
0          0          2          0          2          2
1          1          3          2          4          5
2          2          4          5          7          9
VuongDinhThanhNgan_2052164
9@ubuntu:~$

```

Hình 4.4.2: Chương trình sau khi được sửa lỗi và chạy



- Để tính average waiting time và average turnaround ta cần thêm đoạn code như dòng 40 và 41 trong hình sau:

```

36 [i],tat[i],finish[i]);
37 totwt+=wt[i];
38 tottat+=tat[i];
39 }
40 printf("\nAVG wt = %f", (float)totwt/n);
41 printf("\nAVG tat = %f", (float)tottat/n);
42 }

```

Hình 4.4.3: Đoạn code được thêm vào để tính average waiting và average turnaround time

- Sau đó ta thực hiện chạy lại đoạn code trên và kết cho cho ra average waiting time và average turnaround.

```

VuongDinhThanhNgan_20521649@ubuntu:~$ gcc 4_4.c -o 4_4
VuongDinhThanhNgan_20521649@ubuntu:~$ ./4_4
Enter the number of processes:3
Enter the Process Name, Arrival Time & Burst Time:0 0 2
Enter the Process Name, Arrival Time & Burst Time:1 1 3
Enter the Process Name, Arrival Time & Burst Time:2 2 4

PName Arrtime Burtime Start TAT Finish
0      0      2      0      2      2
1      1      3      2      4      5
2      2      4      5      7      9
9AVG wt = 1.333333AVG tat = 4.333333

```

Hình 4.4.6: Chương trình được chạy lại sau khi thêm code

## 4.5:

### 4.5.1 Viết chương trình mô phỏng giải thuật SJF với các yêu cầu:

```

tina@ubuntu: ~
VuongDinhThanhNgan_20521649@ubuntu:~$ vim sjf.c++
VuongDinhThanhNgan_20521649@ubuntu:~$

#include<iostream>
#include<sstream>
#include<string>
using namespace std;
struct Process
{
int name, arr, bur, star, finish, tat, wt;
};
void SortBurtime(Process* &Prc, int m, int n)
{
for (int i = m; i < n -1; i++)
for (int j = m +1; j <n ; j++)
{
if (Prc[i].bur > Prc[j].bur)
{
Process temp = Prc[i];
Prc[i] = Prc[j];
Prc[j] = temp;
}
}
}
int main()
{
"/sjf.c++" 84L, 1838C 1,1 Top

```

Hình 4.5.1.1: Code chương trình giả thuật SJF

```

Enter the number of Process: 5
Enter the Process Name, Arrival Time & Burst Time: 1 0 4
Enter the Process Name, Arrival Time & Burst Time: 2 1 4
Enter the Process Name, Arrival Time & Burst Time: 3 2 3
Enter the Process Name, Arrival Time & Burst Time: 4 12 1
Enter the Process Name, Arrival Time & Burst Time: 5 9 2
PName  Arrtime  Burtime  Start  WAT  TAT  Finish
1      0       4       0      0    4    4
3      2       3       4      2    5    7
2      1       4       7      6   10   11
5      9       2      11      2    4   13
4     12       1      13      1    2   14
AVG wt = 2.2
AVG tat = 5

```

Process	Arrival time	Burst time
P1	0	4
P2	1	4
P3	2	3
P4	12	1
P5	9	2

Hình 4.5.1.2: Hình sau khi thực thi code giải thuật SJF

#### 4.5.2 Viết chương trình mô phỏng giải thuật SRT với các yêu cầu:

- Đầu tiên tạo file và viết code thực thi

```

VuongDinhThanhNgan_20521649@ubuntu:~$ vim srt.c

```

```

struct process_struct{
    int pid;
    int at;
    int bt;
    int ct, wt, tat, rt, start_time;
}ps[100];
int main()
{
    int n;
    float bt_remaining[100];
    bool is_completed[100]= {false}, is_first_process = true;
    int current_time = 0;
    int completed = 0;
    float sum_tat = 0, sum_wt = 0, sum_rt = 0, prev = 0;
    printf("Enter total number of processes: ");
    scanf("%d", &n);
    for(int i = 0; i < n; i++)
    {
        printf("Enter processes %d Arrival time: ", i + 1);
        scanf("%d", &ps[i].at);
        ps[i].pid = i;
    }
    for(int i=0; i < n; i++)
    {
        printf("Enter processes %d Burst time: ", i + 1);
        scanf("%d", &ps[i].bt);
        bt_remaining[i] = ps[i].bt;
    }
    while(completed!=n)
    {
        int min_index = -1;
        int minimum = INT_MAX;
        for(int i = 0; i<n; i++)
        {
            if(ps[i].at <= current_time && is_completed[i] == false)
            {
                if(bt_remaining[i] < minimum)
                {
                    minimum = bt_remaining[i];
                    min_index = i;
                }
            }
        }
    }
}

```



```

        {
            if(ps[i].at < ps[min_index].at)
            {
                minimum = bt_remaining[i];
                min_index = i;
            }
        }
    }
    if(min_index == -1)
    {
        current_time++;
    }
    else
    {
        if(bt_remaining[min_index] == ps[min_index].bt)
        {
            ps[min_index].start_time = current_time;
            is_first_process = false;
        }
        bt_remaining[min_index] -= 1;
        current_time++;
        prev = current_time;
        if(bt_remaining[min_index] == 0)
        {
            ps[min_index].ct = current_time;
            ps[min_index].tat = ps[min_index].ct - ps[min_index].at;
            ps[min_index].wt = ps[min_index].tat - ps[min_index].bt;
            ps[min_index].rt = ps[min_index].start_time - ps[min_index].at;
            sum_tat += ps[min_index].tat;
            sum_wt += ps[min_index].wt;
            sum_rt += ps[min_index].rt;
            completed++;
            is_completed[min_index] = true;
        }
    }
}
printf("\nProcess No. \tAT\tCPU Burst Time\tCT\tTAT\tWT\tRT\n");
for(int i=0; i<n; i++)
printf("%d\t%d\t%d\t%d\t%d\t%d\t%d\n", ps[i].pid + 1, ps[i].at, ps[i].bt, ps[i].ct, ps[i].tat, ps[i].wt, ps[i].rt);
printf("\n");
printf("\nAvr TAT: %f", (float)sum_tat/n);

```

Hình 4.5.2.1: Code thực thi giải thuật SRT

```

Enter total number of processes: 5
Enter Process 1 Arrival Time: 0
Enter Process 2 Arrival Time: 1
Enter Process 3 Arrival Time: 4
Enter Process 4 Arrival Time: 6
Enter Process 5 Arrival Time: 7
Enter Process 1 Burst Time: 11
Enter Process 2 Burst Time: 9
Enter Process 3 Burst Time: 10
Enter Process 4 Burst Time: 6
Enter Process 5 Burst Time: 13

Process No.   AT   CPU Burst Time   CT   TAT   WT   RT
1             0    11             26   26    15    0
2             1     9             10    9     0    0
3             4    10             36   32    22   22
4             6     6             16   10     4    4
5             7    13             49   42    29   29

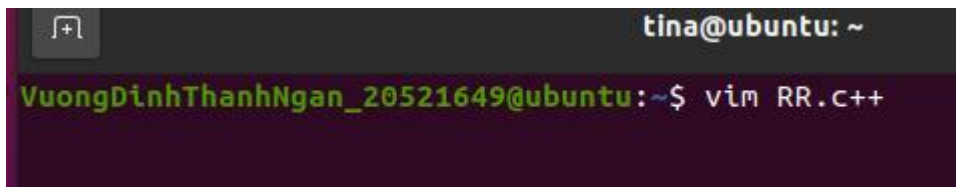
Average Turn Around time= 23.799999
Average Waiting Time= 14.000000

```

Process	Arrival time	Burst time
P1	7	13
P2	0	9
P3	4	3
P4	2	12
P5	1	7

Hình 4.5.2.1: Hình sau khi thực thi code giải thuật SRT

### 4.5.3: Viết chương trình mô phỏng giải thuật RR với các yêu cầu



The screenshot shows a terminal window with the prompt 'tina@ubuntu: ~'. The user has entered the command 'virm RR.c++'. The output of the command is 'VuongDinhThanhNgan\_20521649@ubuntu:~\$ virm RR.c++'.

Hình 4.5.3.1: Thực hiện mở vim tạo file c++

```

#include<iostream>
#include<queue>
#include<algorithm>
using namespace std;
struct Process
{
int name, arr, bur, star, finish, tat, wt, tempbur;
};
bool compare(Process p1, Process p2)
{
return p1.arr < p2.arr;
}
int main()
{
int n, qt;
int totwt = 0, tottat = 0;
cout << "Enter the number of Process: ";
cin >> n;
cout << "Enter the quantum time of Process: ";
cin >> qt;
Process* Prc = new Process[n];
bool* Check = new bool[n];
for (int i = 0; i < n; i++)
{
Check[i] = false;
}
for (int i = 0; i < n; i++)
{
cout << "Enter the Process Name, Arrival Time & Burst Time: ";
cin >> Prc[i].name >> Prc[i].arr >> Prc[i].bur;
Prc[i].tempbur = Prc[i].bur;
}
sort(Prc, Prc + n, compare);
14
queue <int> q;
q.push(0);
Check[0] = true;
int time = 0, index;
int count = 0;
while (count != n)
{
index = q.front();

```

```

Prc[index].finish = time;
Prc[index].tat = Prc[index].finish - Prc[index].arr;
Prc[index].wt = Prc[index].tat - Prc[index].bur;
}
for (int i = 1; i < n; i++)
{
if (Prc[i].arr <= time && Prc[i].tempbur > 0 && Check[i] != true)
{
q.push(i);
15
Check[i] = true;
}
}
if (Prc[index].tempbur > 0)
q.push(index);
if (q.empty())
{
for (int i = 1; i < n; i++)
{
if (Prc[i].tempbur > 0)
{
q.push(i);
Check[i] = true;
break;
}
}
}
cout << "\nPName Arrtime Burttime Start WAT TAT Finish";
for (int i = 0; i < n; i++)
{
cout << endl;
cout << " " << Prc[i].name << "\t " << Prc[i].arr << "\t " << Prc[i].bur << "\t "
<< Prc[i].star << "\t " << Prc[i].wt << "\t " << Prc[i].tat << "\t " << Prc[i].finish;
totwt += Prc[i].wt;
tottat += Prc[i].tat;
}
cout << endl;
cout << "AVG wt = " << (float)totwt/n << endl;
cout << "AVG tat = " << (float)tottat/n << endl;
return 0;
}

```

Hình 4.5.3.2: Code giải thuật RR

Process	Arrival time	Burst time
P1	0	13
P2	4	8
P3	6	4
P4	7	20
P5	12	10

Hình 4.5.3.3: Test case cho RR

```

Enter the number of Process: 5
Enter the quantum time of Process: 5
Enter the Process Name, Arrival Time & Burst Time: 1 0 13
Enter the Process Name, Arrival Time & Burst Time: 2 4 9
Enter the Process Name, Arrival Time & Burst Time: 3 6 4
Enter the Process Name, Arrival Time & Burst Time: 4 7 20
Enter the Process Name, Arrival Time & Burst Time: 5 12 10

PName      Arrtime    Burttime   Start    WAT     TAT     Finish
1          0         13         0        23      36      36
2          4         9          5        15      24      28
3          6         4          15       9       13      19
4          7        20         19       29      49      56
5         12        10         28       24      34      46
AVG wt = 20
AVG tat = 31.2

```

Hình 4.5.3.4: Thực thi code giải thuật RR