

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



HỆ ĐIỀU HÀNH – THỰC HÀNH

BÁO CÁO LAB04

Lớp: IT007.N12.KHCL

Tên: Lê Gia Kiệt

MSSV: 21522255

BÀI LÀM

- Soạn thảo và biên dịch giải thuật FCFS bên dưới
- Bổ sung code để tính average waiting time và average turnaround.

```
1 #include<stdio.h>
2 void main()
3 {
4     int arr[10], bur[10], star[10], finish[10], tat[10], wt[10], i, n;
5     double totwt=0, tottat=0;
6     printf("Enter the number of processes:");
7     scanf("%d", &n);
8     for(i=0; i<n; i++) {
9         printf("Enter the Process Name, Arrival Time & Burst Time:");
10        scanf("%s%d%d", &pn[i], &arr[i], &bur[i]);
11    }
12    for(i=0; i<n; i++) {
13        if(i==0) {
14            star[i]=arr[i];
15            wt[i]=star[i]-arr[i];
16            finish[i]=star[i]+bur[i];
17            tat[i]=finish[i]-arr[i];
18        } else {
19            star[i]=finish[i-1];
20            wt[i]=star[i]-arr[i];
21            finish[i]=star[i]+bur[i];
22            tat[i]=finish[i]-arr[i];
23        }
24    }
25    printf("\nPName Arrtime Burtine Start TAT Finish");
26    for(i=0; i<n; i++) {
27        printf("\n%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d", pn[i], arr[i], bur[i], star[i], tat[i], finish[i]);
28        totwt+=wt[i];
29        tottat+=tat[i];
30    }
31    printf("\n\nThoi gian cho trung binh: %f\n", totwt/n);
32    printf("Thoi gian hoan thanh trung binh: %f\n", tottat/n);
33 }
```

giakiet@giakiet-virtual-machine: ~/Desktop

```
giakiet@giakiet-virtual-machine:~/Desktop$ ./FCFS
Enter the number of processes:4
Enter the Process Name, Arrival Time & Burst Time:0
0
Enter the Process Name, Arrival Time & Burst Time:1
1
Enter the Process Name, Arrival Time & Burst Time:2
2
Enter the Process Name, Arrival Time & Burst Time:1
2
3
PName Arrtime Burtine Start TAT Finish
0 0 2 0 2 2
1 1 3 2 4 5
2 2 4 5 7 9
1 2 3 9 10 12
Thoi gian cho trung binh: 2.750000
Thoi gian hoan thanh trung binh: 5.750000
giakiet@giakiet-virtual-machine:~/Desktop$ ./FCFS
Enter the number of processes:3
Enter the Process Name, Arrival Time & Burst Time:0
0
Enter the Process Name, Arrival Time & Burst Time:1 1 3
Enter the Process Name, Arrival Time & Burst Time:2 2 4
PName Arrtime Burtine Start TAT Finish
0 0 2 0 2 2
1 1 3 2 4 5
2 2 4 5 7 9
Thoi gian cho trung binh: 1.333333
Thoi gian hoan thanh trung binh: 4.333333
giakiet@giakiet-virtual-machine:~/Desktop$
```

1. Viết chương trình mô phỏng giải thuật SJF với các yêu cầu sau:

- ❖ Nhập số lượng process
- ❖ Nhập process name, arrival time, burst time
- ❖ In ra Process name, response time, waiting time, turnaround time, average waiting time, average turnaround time.

```
1 #include <stdio.h>
2
3 void swap(int *a, int *b)
4 {
5     int temp = *a;
6     *a = *b;
7     *b = temp;
8 }
9
10 void deleteThenInsert(int a[], int n, int cu, int moi)
11 {
12     int temp = a[cu];
13     if (cu < 0 || moi < 0 || cu >= n || moi >= n || moi == cu)
14         return;
15     else if (moi < cu)
16     {
17         for (int i = cu; i > moi; i--)
18             a[i] = a[i - 1];
19         a[moi] = temp;
20     }
21     else
22     {
23         for (int i = cu; i <= moi; i++)
24             a[i] = a[i + 1];
25         a[moi] = temp;
26     }
27 }
28
29 int minAtIndex(int a[], int n, int start, int end)
30 {
31     if (start > end || start < 0 || end >= n)
32         return -1;
33     int index = start;
34     for (int i = start + 1; i <= end; i++)
35         if (a[i] < a[index])
36             index = i;
37     return index;
38 }
39
40 void main()
41 {
42     int pn[20], arr[20], bur[20], star[20], finish[20], tat[20], wt[20], i, n;
```

giakiet@giakiet-virtual-machine: ~/Desktop

```
giakiet@giakiet-virtual-machine:~/Desktop$ ./SJF
Enter the number of processes: 4
Enter the Process Name, Arrival Time & Burst Time: 0 5 15
Enter the Process Name, Arrival Time & Burst Time: 1 2 8
Enter the Process Name, Arrival Time & Burst Time: 2 6 9
Enter the Process Name, Arrival Time & Burst Time: 3 7 25
PName ArrTime Burtine Start Finish TAT WT
1 2 8 2 10 12 0
2 6 9 10 19 9 4
0 5 15 19 34 15 14
3 7 25 34 59 25 27
Average waiting time: 11.250000
Average turn around time: 15.250000
giakiet@giakiet-virtual-machine:~/Desktop$
```

```

40 void main()
41 {
42     int pn[20], arr[20], bur[20], star[20], finish[20], tat[20],
43     wt[20], i, j, k, n;
44     int totwt = 0, tottat = 0;
45     float awt, atat;
46
47     printf("Enter the number of processes: ");
48     scanf("%d", &n);
49     for (i = 0; i < n; i++)
50     {
51         printf("Enter the Process Name, Arrival Time & Burst Time: ");
52         scanf("%s%d%d", &pn[i], &arr[i], &bur[i]);
53     }
54     for (i = 0; i < n - 1; i++)
55     for (j = i + 1; j < n; j++)
56     if (arr[i] > arr[j])
57     {
58         swap(&arr[i], &arr[j]);
59         swap(&bur[i], &bur[j]);
60         swap(&pn[i], &pn[j]);
61     }
62     i = 1;
63     while (i < n && arr[i] == arr[0])
64         i++;
65     j = minAtIndex(bur, n, 0, i - 1);
66
67     if (j != 0)
68     {
69         deleteThenInsert(arr, n, j, 0);
70         deleteThenInsert(bur, n, j, 0);
71         deleteThenInsert(pn, n, j, 0);
72     }
73
74     star[0] = arr[0];
75     wt[0] = 0;
76     finish[0] = star[0] + bur[0];
77     tat[0] = finish[0] + arr[0];
78
79     for (k = 1; k < n; k++)
80     {
81         if (arr[k] > finish[k - 1])
82         {
83             i = k + 1;
84             while (i < n && arr[i] == arr[k])
85                 i++;
86             j = minAtIndex(bur, n, k, i - 1);
87             if (j != k)
88             {
89                 deleteThenInsert(arr, n, j, k);
90                 deleteThenInsert(bur, n, j, k);
91                 deleteThenInsert(pn, n, j, k);
92             }
93             star[k] = arr[k];
94         }
95         else
96         {
97             i = k + 1;
98             while (i < n && arr[i] <= finish[k - 1])
99                 i++;
100             j = minAtIndex(bur, n, k, i - 1);
101             if (j != k)
102             {
103                 deleteThenInsert(arr, n, j, k);
104                 deleteThenInsert(bur, n, j, k);
105                 deleteThenInsert(pn, n, j, k);
106             }
107             star[k] = finish[k - 1];
108         }
109         finish[k] = star[k] + bur[k];
110         wt[k] = star[k] - arr[k];
111         tat[k] = finish[k] - star[k];
112     }
113
114     printf("\nPN\tArrTime\tBurTime\tStart\tFinish\tTAT\tWT");
115     for (i = 0; i < n; i++)
116     {
117         printf("\n%s\t%d\t%d\t%d\t%d\t%d\t%d", pn[i],
118             arr[i], bur[i], star[i], finish[i], tat[i], wt[i]);
119         totwt += wt[i];
120         tottat += tat[i];
121     }
122
123     awt = (float)totwt / n;
124     atat = (float)tottat / n;
125     printf("\n\nAverage waiting time: %f\n", awt);
126     printf("Average turn around time: %f\n", atat);
127 }

```

```

giakiet@giakiet-virtual-machine: ~/Desktop$ ./SJT
Enter the number of processes: 4
Enter the Process Name, Arrival Time & Burst Time: 0 5 15
Enter the Process Name, Arrival Time & Burst Time: 1 2 8
Enter the Process Name, Arrival Time & Burst Time: 2 6 9
Enter the Process Name, Arrival Time & Burst Time: 3 7 25

PN\tArrTime\tBurTime\tStart\tFinish\tTAT\tWT
1\t2\t8\t2\t10\t12\t0
2\t6\t9\t10\t19\t9\t4
0\t5\t15\t19\t34\t15\t14
3\t7\t25\t34\t59\t25\t27

Average waiting time: 11.250000
Average turn around time: 15.250000
giakiet@giakiet-virtual-machine:~/Desktop$ S

```

```

giakiet@giakiet-virtual-machine:~/Desktop$ ./SJT
Enter the number of processes: 4
Enter the Process Name, Arrival Time & Burst Time: 0 5 15
Enter the Process Name, Arrival Time & Burst Time: 1 2 8
Enter the Process Name, Arrival Time & Burst Time: 2 6 9
Enter the Process Name, Arrival Time & Burst Time: 3 7 25

PN\tArrTime\tBurTime\tStart\tFinish\tTAT\tWT
1\t2\t8\t2\t10\t12\t0
2\t6\t9\t10\t19\t9\t4
0\t5\t15\t19\t34\t15\t14
3\t7\t25\t34\t59\t25\t27

Average waiting time: 11.250000
Average turn around time: 15.250000
giakiet@giakiet-virtual-machine:~/Desktop$ SSS

```

2. Viết chương trình mô phỏng giải thuật SRT với các yêu cầu sau:

- ❖ Nhập số lượng process
- ❖ Nhập process name, arrival time, burst time
- ❖ In ra Process name, response time, waiting time, turnaround time, average waiting time, average turnaround time.

```

exercise4.c  SRT.c  RoundRobin.c
1 #include <stdio.h>
2
3 void main()
4 {
5     int a[10], b[10], x[10];
6     int waiting[10], turnaround[10], completion[10];
7     int i, j, smallest, count = 0, time, n;
8     double avg = 0, tt = 0, end;
9     printf("\nEnter the number of Processes: ");
10    scanf("%d", &n);
11    for (i = 0; i < n; i++)
12    {
13        printf("Enter arrival, burst time of process %d: ", i + 1);
14        scanf("%d", &a[i]);
15        scanf("%d", &b[i]);
16    }
17    for (i = 0; i < n; i++)
18    {
19        x[i] = b[i];
20    }
21    b[9] = 9999;
22    for (time = 0; count != n; time++)
23    {
24        smallest = 9;
25        for (i = 0; i < n; i++)
26        {
27            if (a[i] <= time && b[i] < b[smallest] && b[i] > 0)
28                smallest = i;
29        }
30        b[smallest]--;
31        if (b[smallest] == 0)
32        {
33            count++;
34            end = time + 1;
35            completion[smallest] = end;
36            waiting[smallest] = end - a[smallest] - x[smallest];
37            turnaround[smallest] = end - a[smallest];
38        }
39    }
40    printf("pid \t burst \t arrival \t waiting \t turnaround \t completion");
41    for (i = 0; i < n; i++)
42    {
43        printf("\n %d \t %d \t %d \t %d \t %d \t %d", i + 1,
44            x[i], a[i], waiting[i], turnaround[i], completion[i]);
45        avg = avg + waiting[i];
46        tt = tt + turnaround[i];
47    }
48    printf("\n\nAverage waiting time: %lf\n", avg / n);
49    printf("Average finish time: %lf\n", tt / n);
50 }

```

```

giakiet@giakiet-virtual-machine: ~/Desktop$ gcc SRT.c -o SRT
SRT.c: In function 'main':
SRT.c:43:16: warning: too many arguments for format [-Wformat-extra-args]
43 |         printf("\n %d \t %d \t %d \t %d \t %d \t %d", i + 1,
    |
giakiet@giakiet-virtual-machine: ~/Desktop$ gcc SRT.c -o SRT
SRT.c: In function 'main':
SRT.c:43:16: warning: too many arguments for format [-Wformat-extra-args]
43 |         printf("\n %d \t %d \t %d \t %d \t %d \t %d", i + 1,
    |
giakiet@giakiet-virtual-machine: ~/Desktop$ gcc SRT.c -o SRT
giakiet@giakiet-virtual-machine: ~/Desktop$ ./SRT

Enter the number of Processes: 6
Enter arrival, burst time of process 1: 0 20
Enter arrival, burst time of process 2: 25 25
Enter arrival, burst time of process 3: 20 25
Enter arrival, burst time of process 4: 35 15
Enter arrival, burst time of process 5: 10 35
Enter arrival, burst time of process 6: 15 50

pid    burst    arrival    waiting    turnaround    completion
1       20       0         0          20           20
2       25       25        35         60           85
3       25       20         0          25           45
4       15       35         10         25           60
5       35       10         75        110          120
6       50       15        105        155          170

Average waiting time: 37.500000
Average finish time: 65.833333
giakiet@giakiet-virtual-machine: ~/Desktop$

```

3. Viết chương trình mô phỏng giải thuật RR với các yêu cầu sau (giả sử tất cả các tiến trình đều có arrival time là 0):
- ❖ Nhập số process, Nhập quantum time, Nhập process name, burst time
 - ❖ In ra Gantt chart với các thông số: process name, start processor time, stop processor time.
 - ❖ In ra average waiting time và average turnaround time.

```

RoundRobin.c
1 #include <stdio.h>
2 int main()
3 {
4     int count, j, n, time, remain, flag = 0, time_quantum;
5     int wait_time = 0, turnaround_time = 0;
6     int at[10], bt[10], rt[10];
7     printf("Enter number of processes:\t ");
8     scanf("%d", &n);
9     remain = n;
10    for (count = 0; count < n; count++)
11    {
12        printf("Enter Burst time for Process Number %d: ", count + 1);
13        at[count] = 0;
14        scanf("%d", &bt[count]);
15        rt[count] = bt[count];
16    }
17    printf("Enter Time quantum:\t");
18    scanf("%d", &time_quantum);
19    printf("\n\nProcess\t| Stop Time | Start Time\n\n");
20    for (time = 0, count = 0; remain != 0;)
21    {
22        if (rt[count] <= time_quantum && rt[count] > 0)
23        {
24            int startime;
25            startime = time;
26            time += rt[count];
27            rt[count] = 0;
28            flag = 1;
29            printf("P[%d]\t|\t%d\t|\t%d\n", count + 1, time - at[count], startime);
30        }
31        else if (rt[count] > 0)
32        {
33            rt[count] -= time_quantum;
34            time += time_quantum;
35            printf("P[%d]\t|\t%d\t|\t%d\n", count + 1, time - at[count], time_quantum);
36        }
37        if (rt[count] == 0 && flag == 1)
38        {
39            remain--;
40            wait_time += time - at[count] - bt[count];
41            turnaround_time += time - at[count];
42            flag = 0;
43        }
44        if (count == n - 1)
45            count = 0;
46        else if (at[count + 1] <= time)
47            count++;
48        else
49            count = 0;
50    }
51    printf("\n\nAverage waiting time: %f\n", wait_time * 1.0 / n);
52    printf("Average turn around time: %f\n", turnaround_time * 1.0 / n);
53    printf("\n\n");
54    return 0;
55 }

```

```

giakiet@giakiet-virtual-machine: ~/Desktop$ gcc RoundRobin.c -o RR
giakiet@giakiet-virtual-machine: ~/Desktop$ ./RR
Enter number of processes: 6
Enter Burst time for Process Number 1: 20
Enter Burst time for Process Number 2: 25
Enter Burst time for Process Number 3: 25
Enter Burst time for Process Number 4: 15
Enter Burst time for Process Number 5: 35
Enter Burst time for Process Number 6: 50
Enter Time quantum: 10

Process | Stop Time | Start Time
P[1] | 10 | 10
P[2] | 20 | 10
P[3] | 30 | 10
P[4] | 40 | 10
P[5] | 50 | 10
P[6] | 60 | 10
P[1] | 70 | 60
P[2] | 80 | 10
P[3] | 90 | 10
P[4] | 95 | 90
P[5] | 105 | 10
P[6] | 115 | 10
P[2] | 120 | 115
P[3] | 125 | 120
P[5] | 135 | 10
P[6] | 145 | 10
P[5] | 150 | 145
P[6] | 160 | 10
P[6] | 170 | 160

Average waiting time: 93.333333
Average turn around time: 121.666667
giakiet@giakiet-virtual-machine: ~/Desktop$

```