

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

MÔN HỆ ĐIỀU HÀNH

BÁO CÁO THỰC HÀNH LAB 4

Lớp IT007.P11

23520552 – PHAN CẢNH ĐĂNG HUÂN

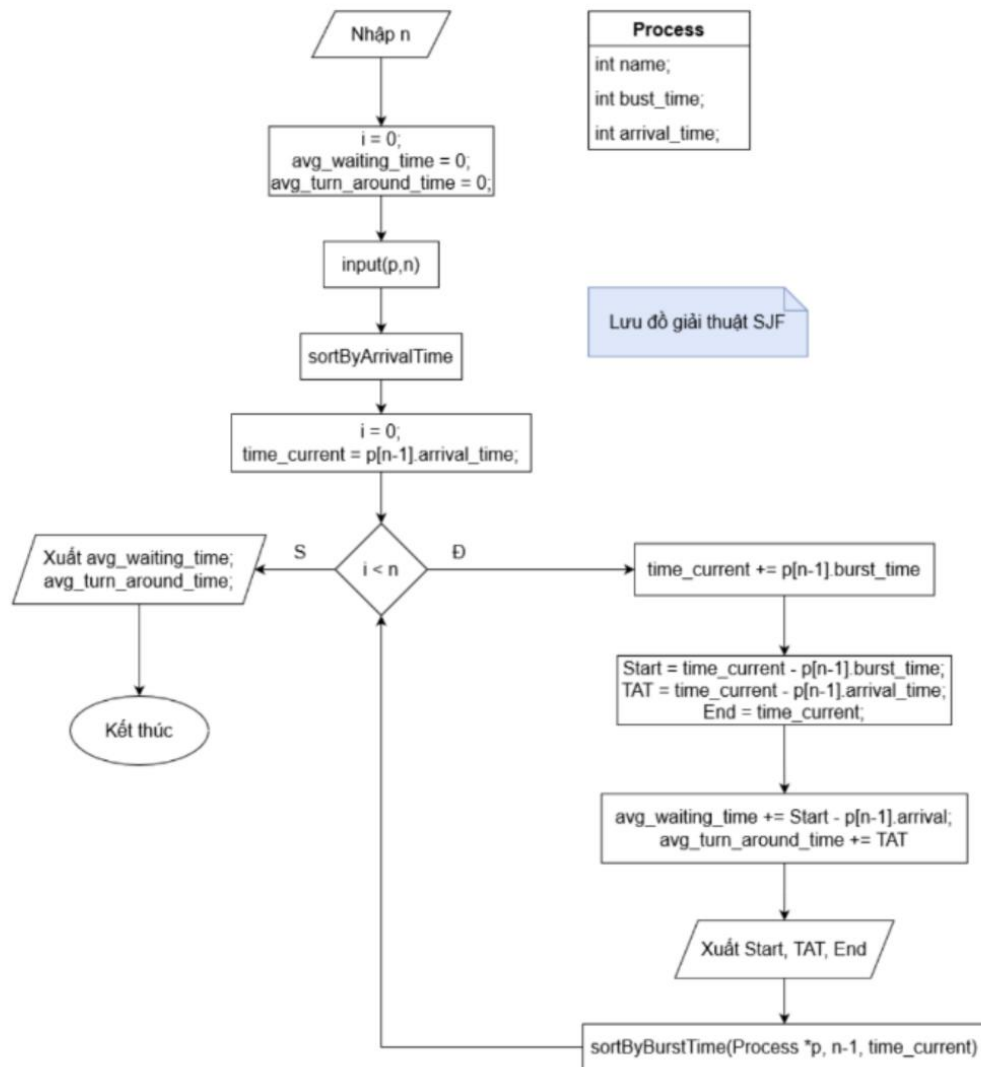
NỘI DUNG BÁO CÁO

Cho các yêu cầu sau:

- Vẽ lưu đồ giải thuật.
 - Trình bày tính đúng đắn của lưu đồ bằng cách chạy tay ít nhất 01 test case có ít nhất 05 tiến trình.
 - Thực hiện code cho giải thuật, trong đó, Arrival Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[0, 20]$, Burst Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[2, 12]$.
 - Trình bày tính đúng đắn của code bằng cách chạy ít nhất 03 test case, mỗi test case có ít nhất 5 tiến trình, so sánh kết quả chạy tay và chạy code.
-

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

a. Lưu đồ giải thuật



b. Trình bày tính đúng đắn của lưu đồ bằng cách chạy tay ít nhất 01 test case có ít nhất 05 tiến trình.

Process	Arrival Time	Burst Time
P3	11	10
P1	18	2
P4	19	2
P2	16	6
P5	17	7

1. Thời gian đáp ứng (Response Time):

- P3: 0
- P1: 10
- P4: 12
- P2: 14
- P5: 17

Thời gian đáp ứng trung bình: 10.6

2. Thời gian đợi (Waiting Time):

- P3: 0
- P1: 9
- P4: 10
- P2: 12
- P5: 14

Thời gian đợi trung bình: 9

3. Thời gian hoàn thành (Turnaround Time):

- P3: 10
- P1: 12
- P4: 14
- P2: 16
- P5: 21

Thời gian hoàn thành trung bình: 14.6

c. Thực hiện code cho giải thuật, trong đó, Arrival Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[0, 20]$, Burst Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[2, 12]$.

```

1  #include <iostream>
2  #include <queue>
3  #include <cstdlib>
4  #include <ctime>
5
6  using namespace std;
7
8  struct Process {
9      int name;
10     int burst_time;
11     int arrival_time;
12 };
13
14 static double avg_turn_around_time = 0;
15 static double avg_waiting_time = 0;
16
17 void swap(Process &p1, Process &p2) {
18     Process tmp = p1;
19     p1 = p2;
20     p2 = tmp;
21 }
22
23 void sortByArrivalTime(Process *p, int n) {
24     for (int i = 0; i < n - 1; i++) {
25         for (int j = i + 1; j < n; j++) {
26             if (p[i].arrival_time > p[j].arrival_time) {
27
28                 swap(p[i], p[j]);
29             }
30         }
31     }
32 }
33
34 void Input(Process *p, int n) {
35     for (int i = 0; i < n; i++) {
36         cout << "-----\n";
37         cout << "Nhập Process ID: "; cin >> p[i].name;
38         cout << "Nhập Arrival Time: "; cin >> p[i].arrival_time;
39         cout << "Nhập Burst Time: "; cin >> p[i].burst_time;
40     }
41 }
42
43 void sortByBurstTime(Process *p, int n, int time_current) {
44     for (int i = 0; i < n - 1; i++) {
45         for (int j = i + 1; j < n; j++) {
46             if (p[i].burst_time > p[j].burst_time && p[j].arrival_time <= time_current)
47
48             {
49
50                 swap(p[i], p[j]);
51             }
52         }
53     }
54 }
55
56 void SelectionFunction(Process *p, int n) {
57     int time_current = 0;
58     sortByArrivalTime(p, n);
59
60     for (int i = 0; i < n; i++) {
61         if (p[i].arrival_time > time_current) {
62             time_current = p[i].arrival_time;
63         }
64     }
65
66     int start_time = time_current;
67     int finish_time = time_current + p[i].burst_time;
68     int turn_around_time = finish_time - p[i].arrival_time;
69     int waiting_time = start_time - p[i].arrival_time;
70
71     avg_waiting_time += waiting_time;
72     avg_turn_around_time += turn_around_time;

```

```

73 cout << p[i].name << "\t\t" << p[i].arrival_time << "\t\t"
74 << p[i].burst_time << "\t\t" << start_time << "\t\t"
75 << turn_around_time << "\t\t" << finish_time << endl;
76
77 time_current = finish_time;
78 sortByBurstTime(p + i + 1, n - i - 1, time_current);
79 }
80 }
81
82 void inputRandom(Process *p, int n) {
83 srand(time(0));
84 for (int i = 0; i < n; i++) {
85 p[i].name = i + 1;
86 p[i].arrival_time = rand() % 21;
87 p[i].burst_time = 2 + rand() % 11;
88 }
89 }
90
91 int main() {
92 Process *p = new Process[100];
93 int n;
94
95 cout << "Nhap so luong tien trinh: ";
96 cin >> n;
97
98 inputRandom(p, n);
99
100 cout << "PName\t\tArrtime\t\tBurtime\t\tStart\t\tTAT\t\tFinish\n";
101 SelectionFunction(p, n);
102
103 cout << "Thoi gian cho trung binh: " << avg_waiting_time / n << endl;
104 cout << "Thoi gian hoan thanh trung binh: " << avg_turn_around_time / n <<
105 endl;
106 return 0;
107 }

```

d. Trình bày tính đúng đắn của code bằng cách chạy ít nhất 03 test case, mỗi test case có ít nhất 5 tiến trình, so sánh kết quả chạy tay và chạy code.

```

(kali@kali)-[~/Desktop/HDH/Lab4]
$ cd "/home/kali/Desktop/HDH/Lab4/" && g++ sjf.cpp -o sjf && "/home/kali/Desktop/HDH/Lab4/"sjf
Nhap so luong tien trinh: 5
PName    Arrtime    Burtime    Start    TAT    Finish
5         1           5          1         5         6
1         3           9          6        12        15
4         18          9          18         9        27
3         18          3          27        12        30
2         20          10         30        20        40
Thoi gian cho trung binh: 4.4
Thoi gian hoan thanh trung binh: 11.6

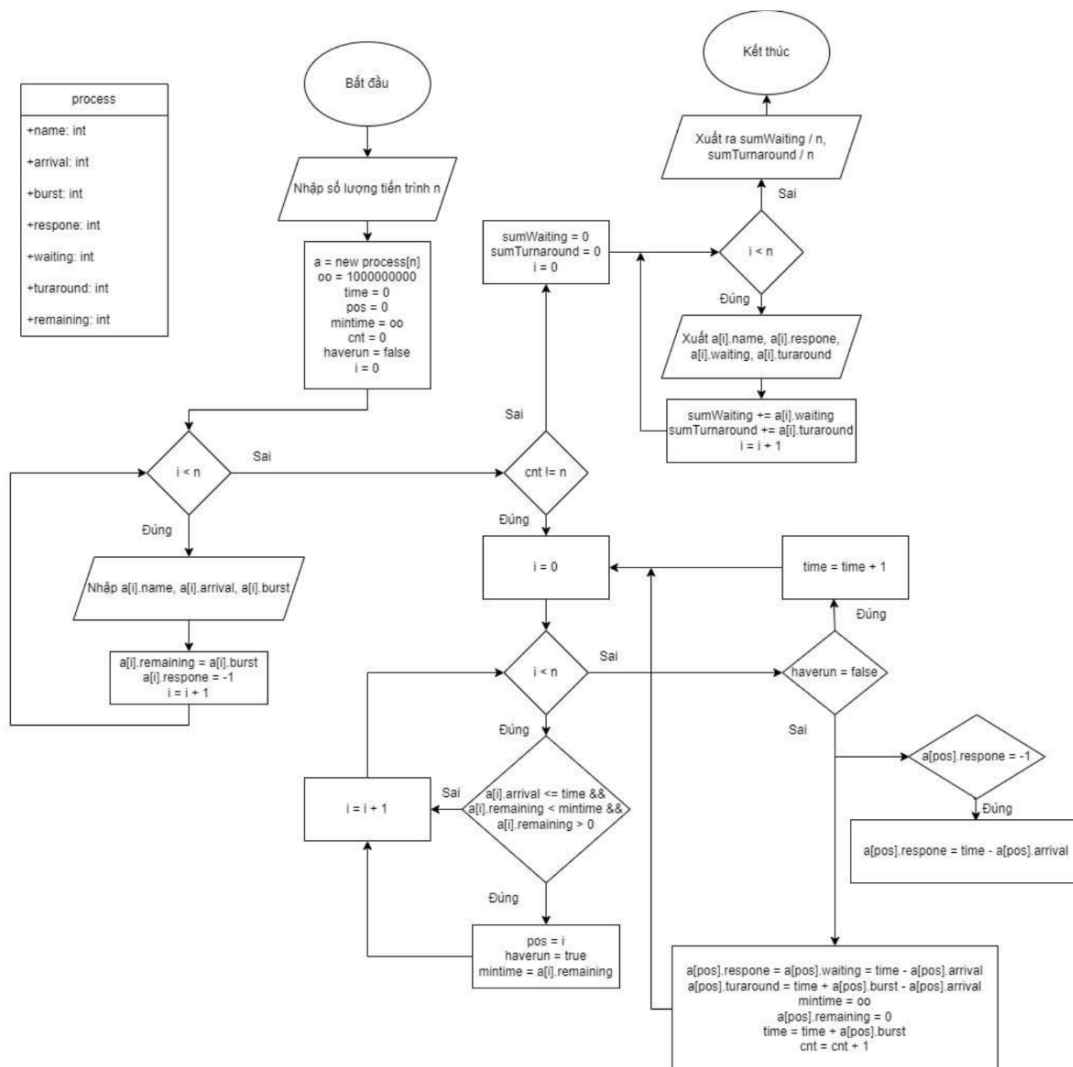
(kali@kali)-[~/Desktop/HDH/Lab4]
$ cd "/home/kali/Desktop/HDH/Lab4/" && g++ sjf.cpp -o sjf && "/home/kali/Desktop/HDH/Lab4/"sjf
Nhap so luong tien trinh: 5
PName    Arrtime    Burtime    Start    TAT    Finish
1         9           3          9         3         12
5         9           9          12        12        21
4         15          3          21         9        24
3         12          10         24        22        34
2         20          11         34        25        45
Thoi gian cho trung binh: 7
Thoi gian hoan thanh trung binh: 14.2

(kali@kali)-[~/Desktop/HDH/Lab4]
$ cd "/home/kali/Desktop/HDH/Lab4/" && g++ sjf.cpp -o sjf && "/home/kali/Desktop/HDH/Lab4/"sjf
Nhap so luong tien trinh: 5
PName    Arrtime    Burtime    Start    TAT    Finish
3         5          11         5         11        16
2         16          3          16         3        19
1         6           4          19         17        23
5         11          6          23         18        29
4         5           9          29         33        38
Thoi gian cho trung binh: 9.8
Thoi gian hoan thanh trung binh: 16.4

```

2. Chọn một trong hai giải thuật gồm SRTF và RR để thực hiện các yêu cầu trên.

a. Lưu đồ giải thuật (SRTF)



b. Trình bày tính đúng đắn của lưu đồ bằng cách chạy tay ít nhất 01 test case có ít nhất 05 tiến trình.

Process	Arrival Time (AT)	Burst Time (BT)
P1	0	8
P2	1	4
P3	2	9
P4	3	5
P5	4	2

1. Thời gian đáp ứng (Response Time):

- **P1:** 0
- **P2:** 1
- **P3:** 19
- **P4:** 7
- **P5:** 5

Thời gian đáp ứng trung bình: 6.4

2. Thời gian đợi (Waiting Time):

- **P1:** 12
- **P2:** 0
- **P3:** 17
- **P4:** 4
- **P5:** 1

Thời gian đợi trung bình: 6.8

3. Thời gian hoàn thành (Turnaround Time):

- **P1:** 19
- **P2:** 4
- **P3:** 26
- **P4:** 9
- **P5:** 3

Thời gian hoàn thành trung bình: 12.2

c. Thực hiện code cho giải thuật, trong đó, Arrival Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[0, 20]$, Burst Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn $[2, 12]$.

```

1  #include <iostream>
2  #include <queue>
3  #include <cstdlib>
4  #include <ctime>
5  #include <climits>
6
7  using namespace std;
8
9  struct Process {
10 int name;
11 int burst_time;
12 int arrival_time;
13
14 int remaining_time;
15 int finish_time;
16 int start_time = -1;
17 };
18
19 static double avg_turn_around_time = 0;
20 static double avg_waiting_time = 0;
21
22 void inputRandom(Process *p, int n) {
23 srand(time(0));
24 for (int i = 0; i < n; i++) {
25 p[i].name = i + 1;
26 p[i].arrival_time = rand() % 21;
27 p[i].burst_time = 2 + rand() % 11;
28 p[i].remaining_time = p[i].burst_time;
29 }
30 }
31
32 void SRTF(Process *p, int n) {
33 int time_current = 0;
34 int completed = 0;
35
36 while (completed < n) {
37 int shortest = -1;
38
39 while (completed < n) {
40 int min_remaining_time = INT_MAX;
41
42 for (int i = 0; i < n; i++) {
43 if (p[i].arrival_time <= time_current && p[i].remaining_time > 0 &&
44 p[i].remaining_time < min_remaining_time) {
45 min_remaining_time = p[i].remaining_time;
46 shortest = i;
47 }
48 }
49
50 if (shortest == -1) {
51 time_current++;
52 continue;
53 }
54
55 if (p[shortest].start_time == -1) {
56 p[shortest].start_time = time_current;
57 }
58
59 p[shortest].remaining_time--;
60 time_current++;
61
62 if (p[shortest].remaining_time == 0) {
63 completed++;
64 p[shortest].finish_time = time_current;
65
66 int turn_around_time = p[shortest].finish_time - p[shortest].arrival_time;
67 int waiting_time = turn_around_time - p[shortest].burst_time;
68
69 avg_turn_around_time += turn_around_time;
70 avg_waiting_time += waiting_time;
71
72 cout << p[shortest].name << "\t\t" << p[shortest].arrival_time << "\t\t"
73 << p[shortest].burst_time << "\t\t" << p[shortest].start_time << "\t\t"
74 << turn_around_time << "\t\t" << p[shortest].finish_time << endl;

```

```

73 }
74 }
75 }
76
77 int main() {
78     Process *p = new Process[100];
79     int n;
80
81     cout << "Nhap so luong tien trinh: ";
82     cin >> n;
83
84     inputRandom(p, n);
85
86     cout << "PName\t\tArrtime\t\tBurtime\t\tStart\t\tTAT\t\tFinish\n";
87     SRTF(p, n);
88
89     cout << "Thoi gian cho trung binh: " << avg_waiting_time / n << endl;
90     cout << "Thoi gian hoan thanh trung binh:5 " << avg_turn_around_time / n <<
91     endl;
92
93     delete[] p;
94     return 0;
95 }

```

d. Trình bày tính đúng đắn của code bằng cách chạy ít nhất 03 test case, mỗi test case có ít nhất 5 tiến trình, so sánh kết quả chạy tay và chạy code

```

• $ cd "/home/kali/Desktop/HDH/lab4/" && g++ srtf.cpp -o srtf && "/home/kali/Desktop/HDH/lab4/"srtf
Nhap so luong tien trinh: 5
PName      Arrtime      Burtime      Start      TAT      Finish
3           4           6           4           6           10
5           10          2           10          2           12
1           14          2           14          2           16
4           16          8           16          8           24
2           8           11          12          25          33
Thoi gian cho trung binh: 2.8
Thoi gian hoan thanh trung binh:5 8.6

(kali@kali) - [~/Desktop/HDH/lab4]
• $ cd "/home/kali/Desktop/HDH/lab4/" && g++ srtf.cpp -o srtf && "/home/kali/Desktop/HDH/lab4/"srtf
Nhap so luong tien trinh: 5
PName      Arrtime      Burtime      Start      TAT      Finish
1           1           4           1           4           5
4           16          4           16          4           20
2           17          5           20          8           25
5           20          8           25          13          33
3           15          11          15          28          43
Thoi gian cho trung binh: 5
Thoi gian hoan thanh trung binh:5 11.4

(kali@kali) - [~/Desktop/HDH/lab4]
• $ cd "/home/kali/Desktop/HDH/lab4/" && g++ srtf.cpp -o srtf && "/home/kali/Desktop/HDH/lab4/"srtf
Nhap so luong tien trinh: 5
PName      Arrtime      Burtime      Start      TAT      Finish
2           7           8           7           8           15
3           12          3           15          6           18
5           8           7           18          17          25
1           11          11          25          25          36
4           8           11          36          39          47
Thoi gian cho trung binh: 11
Thoi gian hoan thanh trung binh:5 19

```