Name: Nguyễn Đại Hưng

ID: 24520601

Class: IT007.Q15.1

# OPERATING SYSTEM
# LAB 4'S REPORT

**SUMMARY**



**Self-scrores: 10/10**

**Note: Export file to **PDF** and name the file by following format:*
***Student ID_LABx.pdf***

# Section 4.5

## 1. Task 1: Viết chương trình mô phỏng thuật toán SJF.

### A. Ý tưởng thuật toán.

*Phần Khởi tạo:*

- Tạo Class **Process** chứa các thuộc tính **name** kiểu string và **arrival_time, brust_time, start_time, finish_time, waiting_time, turnaround_time, response_time** kiểu int.
- Nhập số lượng **process** từ người dùng.
- Tạo **vector process** để lưu trữ thông tin của các **process**.
- Với mỗi **process**, nhập thông tin bao gồm **name, arrival_time, brust_time,** và thêm nó vào **vector process**.

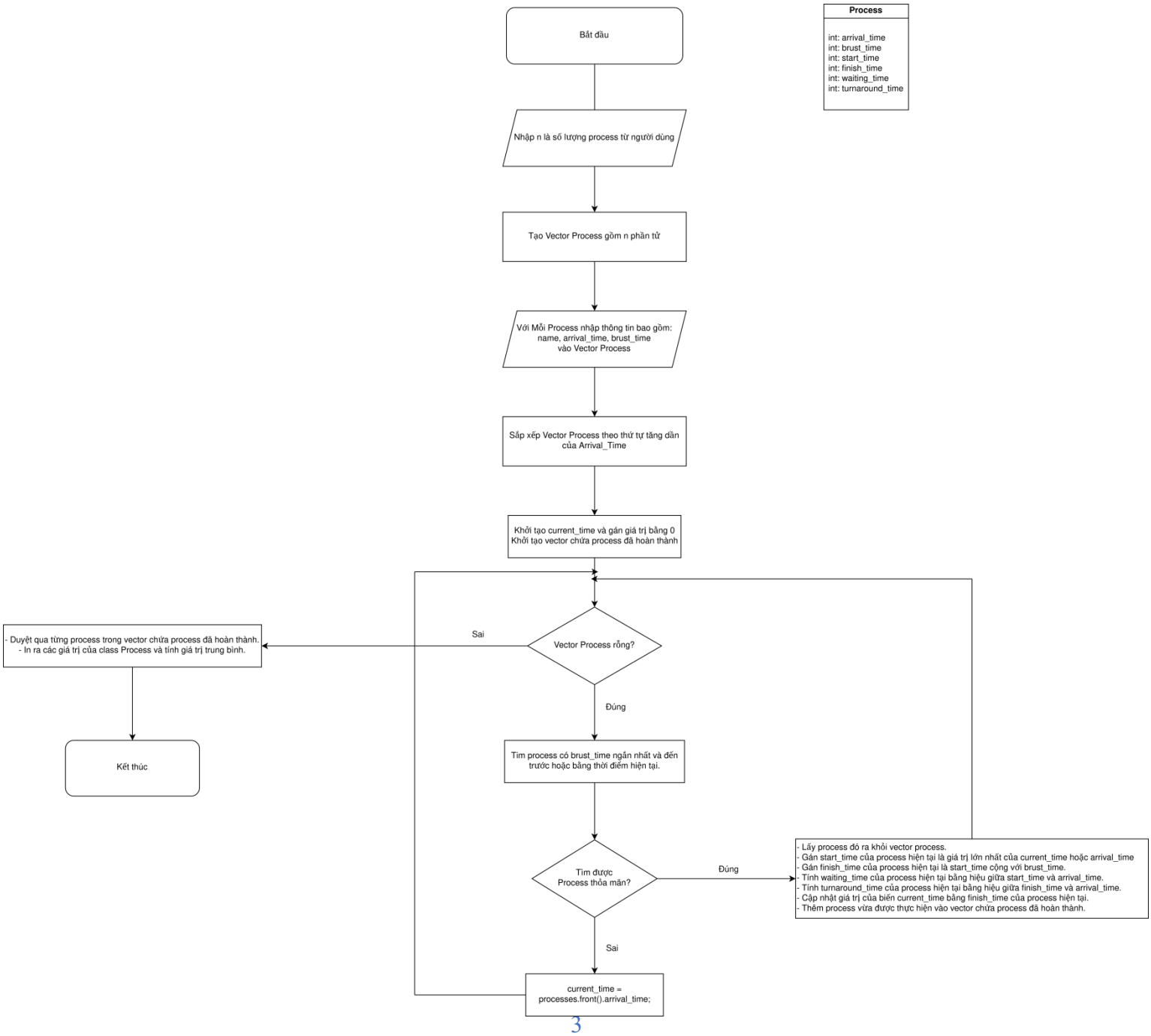
*Phần Xử lý:*

- Sắp xếp **vector process** theo thứ tự tăng dần **arrival_time**.
- Khởi tạo biến **current_time** kiểu int và gán giá trị **arrival_time** nhỏ nhất.
- Khởi tạo một **vector** chứa **process đã hoàn thành.**
- Trong khi **vector process** không **rỗng**:
  - Tìm process có **burst_time** ngắn nhất và đến trước hoặc bằng thời điểm hiện tại.
  - Nếu không tìm thấy process nào thỏa mãn, cập nhật **current_time** bằng **arrival_time** của process đầu tiên trong vector chứa process và tiếp tục vòng lặp.
  - Nếu tìm thấy **process** thỏa mãn, thực hiện các bước sau:
    - Xóa **process** đó ra khỏi **vector process**.
    - Gán **start_time** của **process** hiện tại là giá trị lớn nhất của **current_time** hoặc **arrival_time**
    - Gán **finish_time** của **process** hiện tại là **start_time** cộng với **brust_time**.
    - Tính **waiting_time** của **process** hiện tại bằng hiệu giữa **start_time** và **arrival_time**.
    - Tính **turnaround_time** của **process** hiện tại bằng hiệu giữa **finish_time** và **arrival_time**.
    - Tính **response_time** của **process** hiện tại bằng hiệu giữa **start_time** và **arrival_time**.
    - Cập nhật giá trị của biến **current_time** bằng **finish_time** của process hiện tại.
    - Thêm **process** vừa được thực hiện vào **vector** chứa **process đã hoàn thành**.

*Phần In kết quả:*

• Duyệt qua từng process trong **vector** chứa **process đã hoàn thành**. In ra các giá trị của class Process và tính giá trị trung bình.

## B. FlowChart của thuật toán.



## C. Code của thuật toán

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <climits>

using namespace std;

class Process {
public:
    int id;
    int arrival_time;
    int burst_time;
    int start_time;
    int finish_time;
    int waiting_time;
    int turnaround_time;
    int response_time;

    Process(int id, int arrival_time, int burst_time) :
        id(id), arrival_time(arrival_time), burst_time(burst_time) {}

    void execute(int time_current) {
        start_time = max(time_current, arrival_time); <- (a, b)
        finish_time = start_time + burst_time;
        waiting_time = start_time - arrival_time;
        turnaround_time = finish_time - arrival_time;
        response_time = start_time - arrival_time;
    }
};

bool compareByArrivalTime(Process& p1, Process& p2) {
    return p1.arrival_time < p2.arrival_time;
}

void SJF(vector<Process>& processes) {
    vector<Process> finished_processes;

    sort(processes.begin(), processes.end(), compareByArrivalTime); <- (first, last, comp)
    int time_current = processes[0].arrival_time;

    while (!processes.empty()) {
        int shortest_burst_time = INT_MAX;
        int shortest_process_index = -1;

        for (int i = 0; i < processes.size(); i++) {
            if (processes[i].arrival_time <= time_current && processes[i].burst_time < shortest_burst_time) {
                shortest_burst_time = processes[i].burst_time;
```

sjf1.cpp clangd

s, 3185 bytes

```
        if (shortest_process_index == -1) {
            time_current = processes.front().arrival_time;
            continue;
        }

        Process shortest_process = processes[shortest_process_index];
        processes.erase(processes.begin() + shortest_process_index); <- (position)

        shortest_process.execute(time_current);
        time_current = shortest_process.finish_time;
        finished_processes.push_back(shortest_process); <- (x)
    }

    // Print the results
    int total_waiting = 0;
    int total_turnaround = 0;
    cout << "PName\tArrival\tBurst\tStart\tTurnaround\tFinish\tWaiting\n";
    for (auto process : finished_processes) { => Process
        cout << "P" << process.id << "\t";
        cout << process.arrival_time << "\t";
        cout << process.burst_time << "\t";
        cout << process.start_time << "\t";
        cout << process.turnaround_time << "\t\t";
        cout << process.finish_time << "\t";
        cout << process.waiting_time << "\n";
        total_waiting += process.waiting_time;
        total_turnaround += process.turnaround_time;
    }
    cout << "\nAverage Waiting Time: " << (float)total_waiting / finished_processes.size() << " (ms)";
    cout << "\nAverage Turnaround Time: " << (float)total_turnaround / finished_processes.size() << " (ms)";
    cout << "\n";
}

int main() {
    int n;
    cout << "Enter the number of processes: ";
    cin >> n;

    vector<Process> processes;
    for (int i = 0; i < n; i++) {
        int arrival_time, burst_time;
        cout << "Enter the arrival time and burst time of process " << i + 1 << ": ";
        cin >> arrival_time >> burst_time;
        processes.push_back(Process(i + 1, arrival_time, burst_time)); <- (x, id)
    }

    SJF(processes); <- (&)
```

## D. Test Case

**Test 1**

| Process | Arrival Time | Burst Time |
|---|---|---|
| 1 | 16 | 29 |
| 2 | 18 | 19 |
| 3 | 11 | 26 |
| 4 | 12 | 25 |
| 5 | 11 | 22 |
| 6 | 6 | 15 |

**Kết quả chạy tay**

| Name | Arr | Brust | Start | TAT | Finish |
|---|---|---|---|---|---|
| P1 | 6 | 15 | 6 | 15 | 21 |
| P2 | 11 | 26 | 21 | 102 | 113 |
| P3 | 11 | 22 | 40 | 51 | 62 |
| P4 | 12 | 25 | 62 | 75 | 87 |
| P5 | 16 | 29 | 87 | 126 | 142 |
| P6 | 18 | 19 | 113 | 22 | 40 |
| **Average waiting time: 42.5 (ms)**<br>**Average turnaround time: 65.167 (ms)** | | | | | |

**Kết quả chạy chương trình**

```
Enter the number of processes: 6
Enter the arrival time and burst time of process 1: 16 29
Enter the arrival time and burst time of process 2: 18 19
Enter the arrival time and burst time of process 3: 11 26
Enter the arrival time and burst time of process 4: 12 25
Enter the arrival time and burst time of process 5: 11 22
Enter the arrival time and burst time of process 6: 6 15
PName   Arrival Burst   Start   Finish  Turnaround      Waiting Response
P6      6       15      6       21      15              0       0
P2      18      19      21      40      22              3       3
P5      11      22      40      62      51              29      29
P4      12      25      62      87      75              50      50
P3      11      26      87      113     102             76      76
P1      16      29      113     142     126             97      97

Average Waiting Time: 42.5 (ms)
Average Turnaround Time: 65.1667 (ms)
```

**Test 2**

| Process | Arrival Time | Burst Time |
|---|---|---|
| 1 | 65 | 100 |
| 2 | 21 | 2 |
| 3 | 14 | 10 |
| 4 | 9 | 9 |
| 5 | 99 | 12 |
| 6 | 1 | 23 |

**Kết quả chạy tay**

| Name | Arr | Brust | Start | TAT | Finish |
|---|---|---|---|---|---|
| P1 | 1 | 23 | 1 | 23 | 24 |
| P2 | 21 | 2 | 24 | 5 | 26 |
| P3 | 9 | 9 | 26 | 26 | 35 |
| P4 | 14 | 10 | 35 | 31 | 45 |
| P5 | 65 | 100 | 65 | 100 | 165 |
| P6 | 99 | 12 | 165 | 78 | 177 |
| **Average waiting time: 17.833 (ms)**<br>**Average turnaround time: 43.833 (ms)** | | | | | |

**Kết quả chạy chương trình**

```
Enter the number of processes: 6
Enter the arrival time and burst time of process 1: 65 100
Enter the arrival time and burst time of process 2: 21 2
Enter the arrival time and burst time of process 3: 14 10
Enter the arrival time and burst time of process 4: 9 9
Enter the arrival time and burst time of process 5: 99 12
Enter the arrival time and burst time of process 6: 1 23
PName   Arrival Burst   Start   Finish  Turnaround      Waiting Response
P6      1       23      1       24      23              0       0
P2      21      2       24      26      5               3       3
P4      9       9       26      35      26              17      17
P3      14      10      35      45      31              21      21
P1      65      100     65      165     100             0       0
P5      99      12      165     177     78              66      66

Average Waiting Time: 17.8333 (ms)
Average Turnaround Time: 43.8333 (ms)
```

**Test 3**

| Process | Arrival Time | Burst Time |
|---|---|---|
| 1 | 100 | 21 |
| 2 | 9 | 7 |
| 3 | 21 | 5 |
| 4 | 23 | 32 |
| 5 | 56 | 11 |
| 6 | 7 | 89 |

**Kết quả chạy tay**

| Name | Arr | Brust | Start | TAT | Finish |
|---|---|---|---|---|---|
| P1 | 7 | 89 | 7 | 89 | 96 |
| P2 | 21 | 5 | 96 | 80 | 101 |
| P3 | 9 | 7 | 101 | 99 | 108 |
| P4 | 56 | 11 | 108 | 63 | 119 |
| P5 | 100 | 21 | 119 | 40 | 140 |
| P6 | 23 | 32 | 140 | 149 | 172 |
| **Average waiting time: 59.167 (ms)**<br>**Average turnaround time: 86.667 (ms)** | | | | | |

## 2. Task 2: Viết chương trình mô phỏng thuật toán SRTF.

### A. Ý tưởng thuật toán.

*Phần Khởi tạo:*

- Tạo Class **Process** chứa các thuộc tính **name** kiểu string và **arrival_time, brust_time, start_time, finish_time, waiting_time, turnaround_time, remaining_time** kiểu int.
- Nhập số lượng **process** từ người dùng.
- Tạo **vector process** để lưu trữ thông tin của các **process**.
- Với mỗi **process**, nhập thông tin bao gồm **name, arrival_time, brust_time,** và thêm nó vào **vector process**.

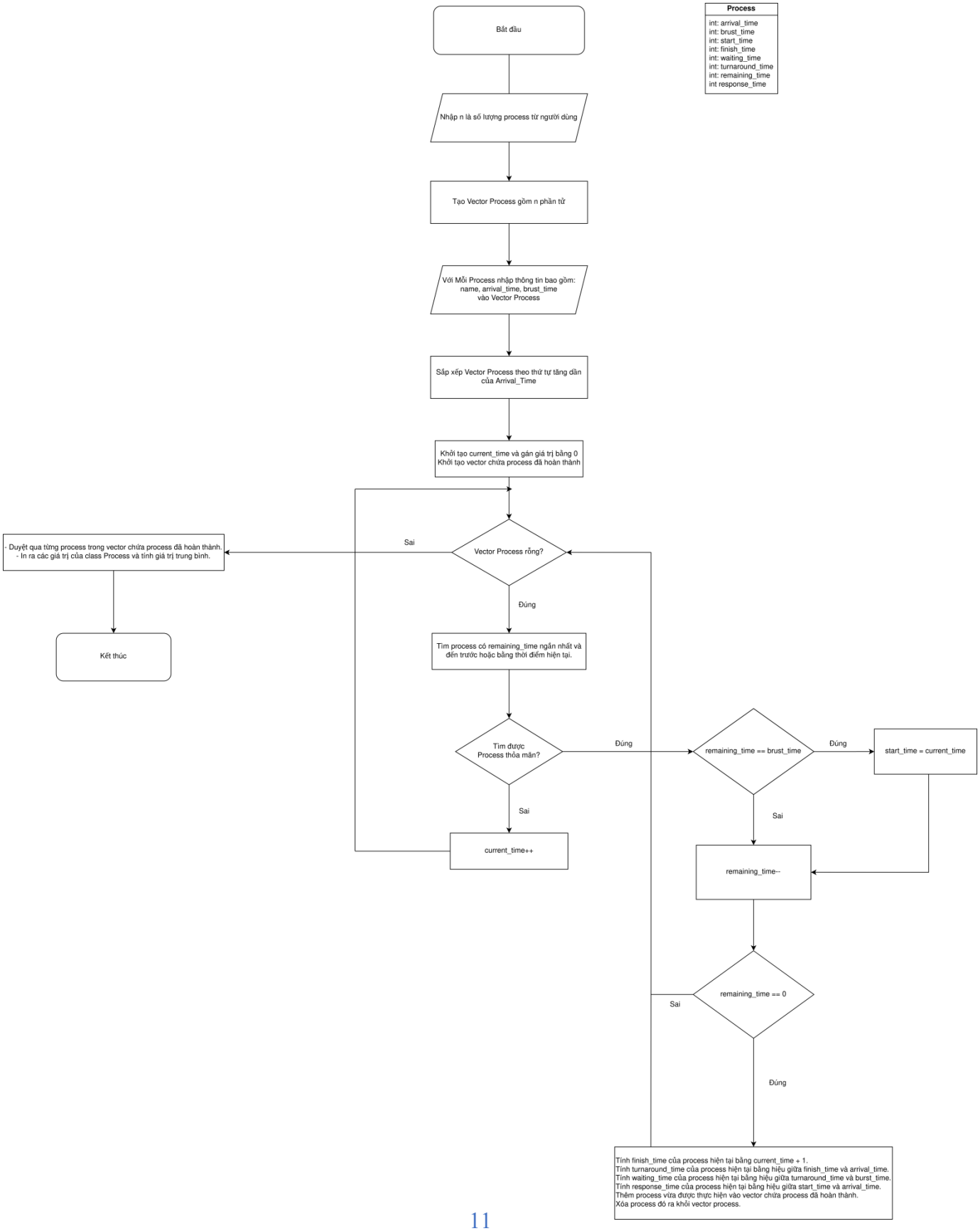
*Phần Xử lý:*

- Sắp xếp **vector process** theo thứ tự tăng dần **arrival_time**.
- Khởi tạo biến **current_time** kiểu int và gán giá trị **arrival_time** nhỏ nhất.
- Khởi tạo một **vector** chứa **process đã hoàn thành.**
- Trong khi **vector process** không **rỗng**:
    - Tìm process có **remaining_time** ngắn nhất và đến trước hoặc bằng thời điểm hiện tại.
    - Nếu không có **process** nào thỏa mãn, chúng ta sẽ tăng **current_time** lên 1 đơn vị và tiếp tục tìm kiếm.
    - Nếu tìm thấy **process** thỏa mãn, thực hiện các bước sau:
        - Nêu **process** chạy lần đầu tiên ta gán **start_time** bằng **current_time.**
        - Giảm **remaining_time** đi một đơn vị.
        - Nếu **remaining_time = 0** (chương trình đã hoàn thành):
            - Tính **finish_time** của **process** hiện tại bằng **current_time + 1.**
            - Tính **turnaround_time** của **process** hiện tại bằng hiệu giữa **finish_time** và **arrival_time**.
            - Tính **waiting_time** của **process** hiện tại bằng hiệu giữa **turnaround_time** và **burst_time**.
            - Tính **response_time** của **process** hiện tại bằng hiệu giữa **start_time** và **arrival_time**.
            - Thêm **process** vừa được thực hiện vào **vector** chứa **process đã hoàn thành**.
            - Xóa **process** đó ra khỏi **vector process**.

*Phần In kết quả:*

• Duyệt qua từng process trong **vector** chứa **process đã hoàn thành**. In ra các giá trị của class Process và tính giá trị trung bình.

## B. FlowChart của thuật toán.



## C. Code của thuật toán.

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

class Process {
public:
    int id;
    int arrival_time;
    int burst_time;
    int start_time;
    int finish_time;
    int waiting_time;
    int turnaround_time;
    int remaining_time;
    int response_time;

    Process(int id, int arrival_time, int burst_time) :
        id(id), arrival_time(arrival_time), burst_time(burst_time), remaining_time(burst_time) {}

    void execute(int time_current) {
        if (remaining_time == burst_time)
        {
            start_time = time_current;
        }
        remaining_time--;
        if (remaining_time == 0) {
            finish_time = time_current + 1;
            turnaround_time = finish_time - arrival_time;
            waiting_time = turnaround_time - burst_time;
            response_time = start_time - arrival_time;
        }
    }
};


bool compareByArrivalTime(Process& p1, Process& p2) {
    return p1.arrival_time < p2.arrival_time;
}

void SRTF(vector<Process>& processes) {
    vector<Process> order;

    sort(processes.begin(), processes.end(), compareByArrivalTime); <- (first, last, comp)
    int time_current = processes[0].arrival_time;

    while (!processes.empty()) {
```

srtf.cpp

```cpp
        auto it = min_element(processes.begin(), processes.end(), <- (first, last)
                              [&](const Process& p1, const Process& p2) { => -> bool
                                  return p1.remaining_time < p2.remaining_time &&
                                         p1.arrival_time <= time_current &&
                                         p2.arrival_time <= time_current;
                              }); <- (comp)

        if (it == processes.end()) {
            time_current++;
            continue;
        }

        Process& p = *it;
        if (time_current >= it->arrival_time) {
            p.execute(time_current);
            time_current++;

            if (p.remaining_time == 0) {
                order.push_back(p); <- (x)
                processes.erase(it); <- (position)
            }
        }
        else {
            time_current++;
        }
    }

    double average_waiting_time = 0;
    double average_turnaround_time = 0;
    for (const Process& p : order) {
        average_waiting_time += p.waiting_time;
        average_turnaround_time += p.turnaround_time;
    }
    average_waiting_time /= order.size();
    average_turnaround_time /= order.size();

    cout << "PName\tArrival\tBurst\tStart\tFinish\tTurnaround\tWaiting\tResponse\n";
    for (auto process : order) { => Process
        cout << "P" << process.id << "\t";
        cout << process.arrival_time << "\t";
        cout << process.burst_time << "\t";
        cout << process.start_time << "\t";
        cout << process.finish_time << "\t";
        cout << process.turnaround_time << "\t\t";
        cout << process.waiting_time << "\t";
        cout << process.response_time << "\n";
    }

```

srtf.cpp

## D. Test case.

**Test case 1:**

| Process | Arrival time | Burst time |
|---|---|---|
| **1** | 11 | 4 |
| **2** | 12 | 6 |
| **3** | 12 | 12 |
| **4** | 45 | 8 |
| **5** | 19 | 7 |

**Kết quả chạy tay:**

| Name | Arr | Burst | Start | TAT | Finish | Waiting |
|---|---|---|---|---|---|---|
| **P1** | 11 | 4 | 11 | 4 | 15 | 0 |
| **P2** | 12 | 6 | 15 | 9 | 21 | 3 |
| **P3** | 12 | 12 | 28 | 28 | 40 | 16 |
| **P4** | 45 | 8 | 45 | 8 | 53 | 8 |
| **P5** | 19 | 7 | 21 | 9 | 28 | 2 |
| **Average waiting time: 4.2 (ms)**<br>**Average turnaround time: 11.6 (ms)** | | | | | | |

**Kết quả chương trình:**

```
Enter the number of processes: 5
Enter the arrival time and burst time of process 1: 11 4
Enter the arrival time and burst time of process 2: 12 6
Enter the arrival time and burst time of process 3: 12 12
Enter the arrival time and burst time of process 4: 45 8
Enter the arrival time and burst time of process 5: 19 7
PName   Arrival Burst   Start   Finish  Turnaround      Waiting Response
P1      11      4       11      15      4               0       0
P2      12      6       15      21      9               3       3
P5      19      7       21      28      9               2       2
P3      12      12      28      40      28              16      16
P4      45      8       45      53      8               0       0


Average Waiting Time: 4.2
Average Turnaround Time: 11.6
```

**Test case 2:**

| Process | Arrival Time | Burst Time |
|---|---|---|
| **1** | 15 | 32 |
| **2** | 12 | 41 |
| **3** | 6 | 26 |
| **4** | 9 | 28 |
| **5** | 2 | 40 |
| **6** | 10 | 32 |

**Kết quả chạy tay:**

| Name | Arr | Brust | Start | TAT | Finish | Waiting |
|---|---|---|---|---|---|---|
| **P1** | 15 | 32 | 92 | 109 | 124 | 77 |
| **P2** | 12 | 41 | 160 | 189 | 201 | 148 |
| **P3** | 6 | 26 | 6 | 26 | 32 | 0 |
| **P4** | 9 | 28 | 32 | 51 | 60 | 23 |
| **P5** | 2 | 40 | 2 | 158 | 160 | 118 |
| **P6** | 10 | 32 | 60 | 82 | 92 | 50 |
| **Average waiting time: 69.333 (ms)**<br>**Average turnaround time: 102.5 (ms)** | | | | | | |

**Kết quả chương trình:**

```
Enter the number of processes: 6
Enter the arrival time and burst time of process 1: 15 32
Enter the arrival time and burst time of process 2: 12 41
Enter the arrival time and burst time of process 3: 6 26
Enter the arrival time and burst time of process 4: 9 28
Enter the arrival time and burst time of process 5: 2 40
Enter the arrival time and burst time of process 6: 10 32
PName   Arrival Burst   Start   Finish  Turnaround      Waiting Response
P3      6       26      6       32      26              0       0
P4      9       28      32      60      51              23      23
P6      10      32      60      92      82              50      50
P1      15      32      92      124     109             77      77
P5      2       40      2       160     158             118     0
P2      12      41      160     201     189             148     148


Average Waiting Time: 69.3333
Average Turnaround Time: 102.5
```

**Test case 3:**

| Process | Arrival Time | Burst Time |
|---|---|---|
| **1** | 2 | 1 |
| **2** | 100 | 10 |
| **3** | 100 | 10 |
| **4** | 50 | 7 |
| **5** | 66 | 16 |
| **6** | 22 | 1 |

**Kết quả chạy tay:**

| Name | Arr | Brust | Start | TAT | Finish | Waiting |
|---|---|---|---|---|---|---|
| **P1** | 2 | 1 | 2 | 1 | 3 | 0 |
| **P2** | 100 | 10 | 100 | 10 | 110 | 0 |
| **P3** | 100 | 10 | 110 | 20 | 120 | 10 |
| **P4** | 50 | 7 | 50 | 7 | 57 | 0 |
| **P5** | 66 | 16 | 66 | 16 | 82 | 0 |
| **P6** | 22 | 1 | 22 | 1 | 23 | 0 |
| **Average waiting time: 1.667 (ms)**<br>**Average turnaround time: 9.167 (ms)** | | | | | | |

**Kết quả chương trình:**

```
Enter the number of processes: 6
Enter the arrival time and burst time of process 1: 2 1
Enter the arrival time and burst time of process 2: 100 10
Enter the arrival time and burst time of process 3: 100 10
Enter the arrival time and burst time of process 4: 50 7
Enter the arrival time and burst time of process 5: 66 16
Enter the arrival time and burst time of process 6: 22 1
PName   Arrival Burst   Start   Finish  Turnaround      Waiting Response
P1      2       1       2       3       1               0       0
P6      22      1       22      23      1               0       0
P4      50      7       50      57      7               0       0
P5      66      16      66      82      16              0       0
P2      100     10      100     110     10              0       0
P3      100     10      110     120     20              10      10


Average Waiting Time: 1.66667
Average Turnaround Time: 9.16667
```