

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**







**HỆ ĐIỀU HÀNH**

**LỚP: IT007.O18**

**Tên: Lê Minh Nhựt**

**MSSV: 22521060**

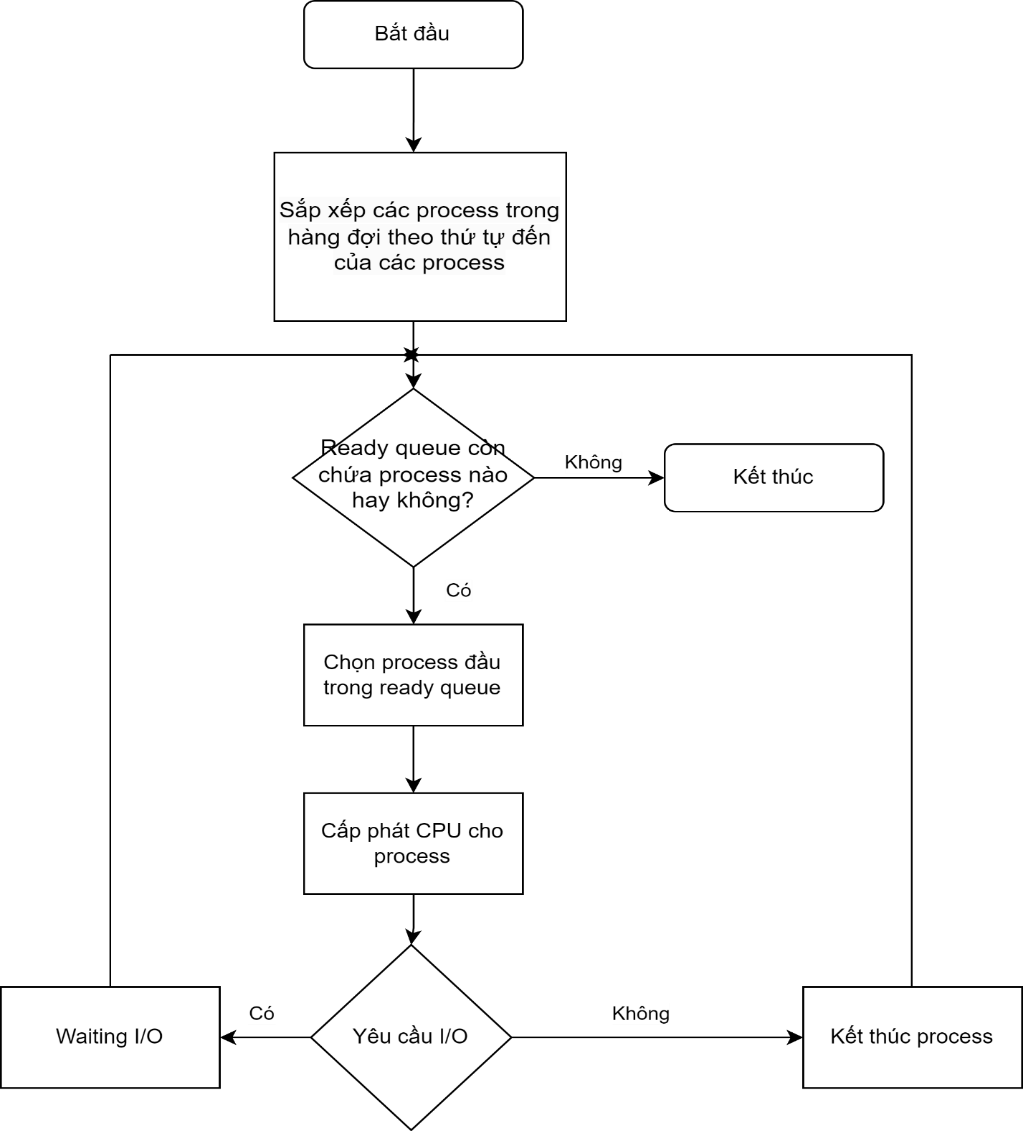


**THỰC HÀNH LAB 4**

**Lý thuyết**

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

* FCFC (First Come First Served)



|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian thực hiện |
| P1 | 0 | 12 |
| P2 | 2 | 7 |
| P3 | 5 | 8 |
| P4 | 9 | 3 |
| P5 | 12 | 6 |

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

0 12 19 27 30 36

* RR (Round Robin)

A screenshot of a computer screen

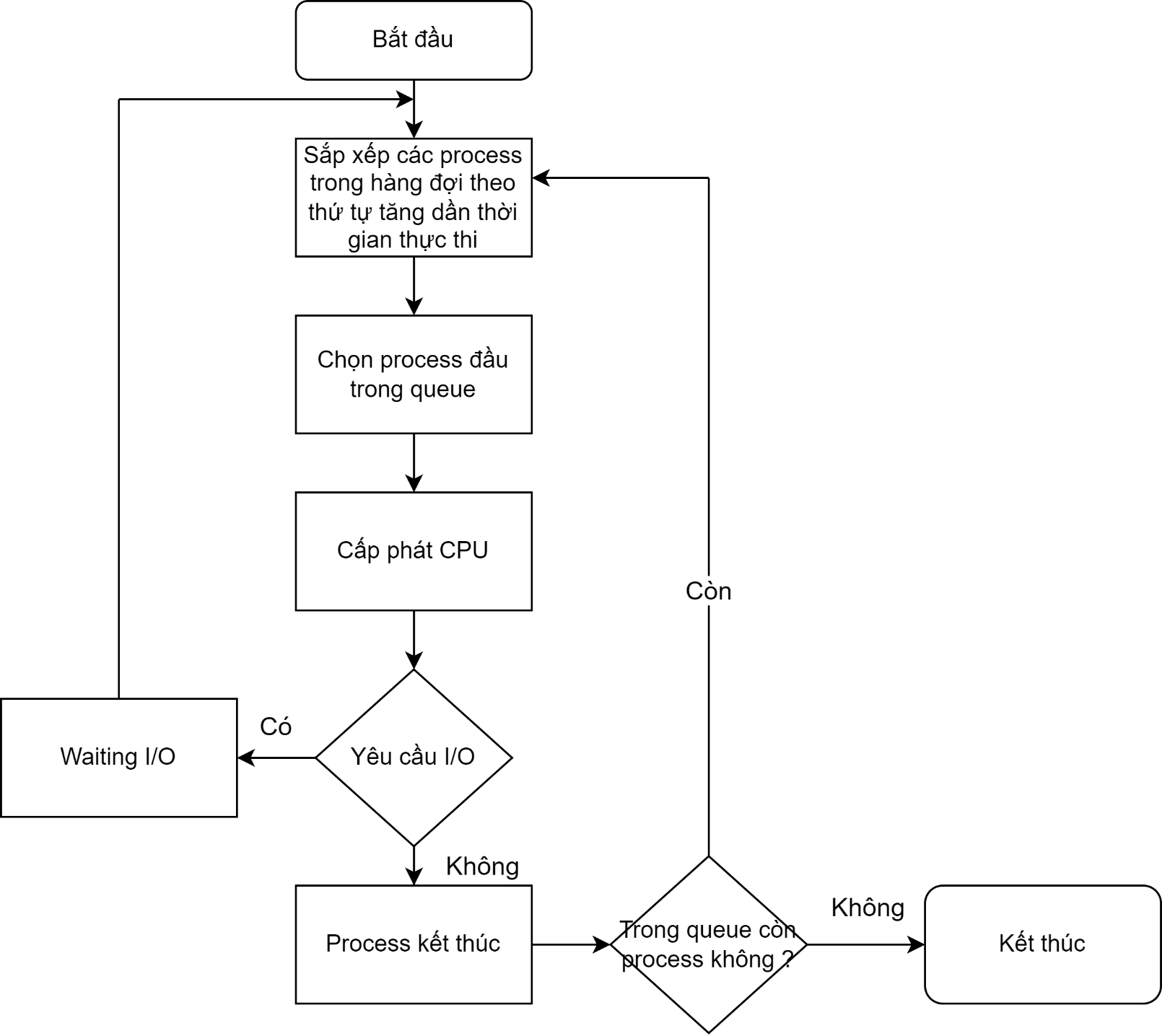
Description automatically generated

|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian thực hiện |
| P1 | 0 | 12 |
| P2 | 2 | 7 |
| P3 | 5 | 8 |
| P4 | 9 | 3 |
| P5 | 12 | 6 |

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

0 4 8 12 16 19 22 26 30 34 36

* SJF (Shortest Job First)



|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian thực hiện |
| P1 | 0 | 12 |
| P2 | 2 | 7 |
| P3 | 5 | 8 |
| P4 | 9 | 3 |
| P5 | 12 | 6 |

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

0 12 15 21 28 36

* SRT (Shortest Remain Time)

A screenshot of a phone

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian thực hiện |
| P1 | 0 | 12 |
| P2 | 2 | 7 |
| P3 | 5 | 8 |
| P4 | 9 | 3 |
| P5 | 12 | 6 |

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

0 2 9 12 18 26 36

**2. Giải thích các thuật ngữ sau**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ** | **Mô tả** |
| 1 | Arrival Time | Thời gian (Thời điểm) một tiến trình được nạp vào hệ thống. |
| 2 | Burst Time | Tổng thời gian một tiến trình thực hiện. |
| 3 | Quantum Time | Thời gian định mức trong thuật toán Ronud Robin. |
| 4 | Response Time | *Thời gian đáp ứng:*Khoảng thời gian process nhận yêu cầu lần đầu tiên khi yêu cầu đầu tiên được đáp ứng. |
| 5 | Waiting Time | *Thời gian chờ:*Tổng thời gian một process đợi trong ready queue. |
| 6 | Turnaround Time | *Thời gian hoàn thành:*Khoảng thời gian từ lúc nạp đến lúc kết thúc của một process. |
| 7 | Average Waiting Time | **Thời gian chờ trung bình:** Tổng thời gian chờ của các process chia cho số process. |
| 8 | Average Turnaround Time | *Thời gian hoàn thành trung bình:*Tổng thời gian hoàn thành của các process chia cho số process. |

**\*Code FCFS:**

#include<stdio.h>

void main(){

    int pn[10];

    int arr[10], bur[10], star[10], finish[10], tat[10], wt[10], i, n;

    int totwt=0, tottat=0;

    printf("Enter the number of processes:");

    scanf("%d",&n);

    for(i=0;i<n;i++) {

        printf("Enter the Process Name, Arrival Time & Burst Time:");

        scanf("%d%d%d",&pn[i],&arr[i],&bur[i]);

    }

    for(i=0;i<n;i++) {

        if(i==0) {

        star[i]=arr[i];

        wt[i]=star[i]-arr[i];

        finish[i]=star[i]+bur[i];

        tat[i]=finish[i]-arr[i];

        }

        else{

            star[i]=finish[i-1];

            wt[i]=star[i]-arr[i];

            finish[i]=star[i]+bur[i];

            tat[i]=finish[i]-arr[i];

        }

    }

    printf("\nPName Arrtime Burtime Start TAT Waitingtime Finish");

    for(i=0;i<n;i++) {

        printf("\n%d\t%6d\t\t%6d\t%6d\t%6d\t%6d\t%6d",pn[i],arr[i],bur[i],star[i],tat[i], wt[i],finish[i]);

        totwt+=wt[i];

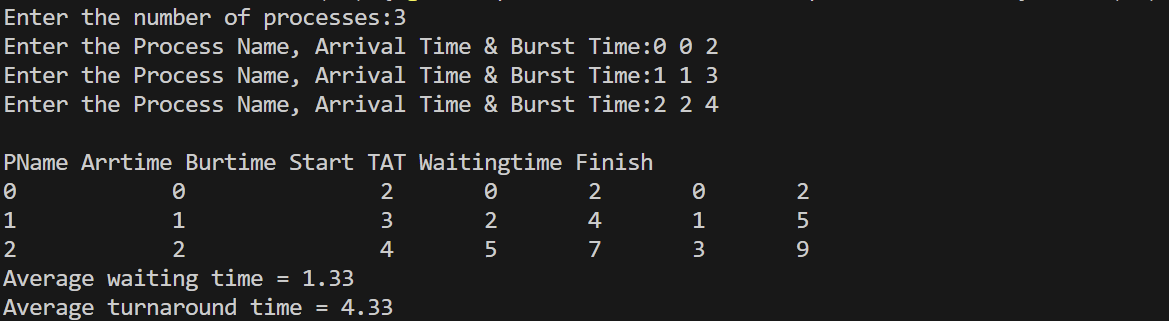
        tottat+=tat[i];

    }

    printf("\nAverage waiting time = %.2f", totwt \* 1.0 / n);

    printf("\nAverage turnaround time = %.2f", tottat \* 1.0 / n);

}



Kết quả chạy giống với kết quả trong đề bài.

**Bài tập**

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

#include<stdio.h>

void main(){

    int pn[10];

    int arrival\_time[10], burst\_time[10], sum\_burst\_time = 0, time, smallest, i, n;

    int total\_waiting\_time = 0, total\_turnaround\_time = 0;

    printf("Enter the number of processes:");

    scanf("%d", &n);

    for(i = 0; i < n; i++){

        printf("Enter the Process Name, Arrival Time & Burst Time:");

        scanf("%d%d%d", &pn[i], &arrival\_time[i], &burst\_time[i]);

        sum\_burst\_time += burst\_time[i];

    }

    burst\_time[9] = 9999;

    printf("\nPName     Arrival time     Burst time     Waiting time     Turnaround time\n");

    for (time = 0; time < sum\_burst\_time;){

        smallest = 9;

        for (i = 0; i < n; i++) {

            if (arrival\_time[i] <= time && burst\_time[i] > 0 && burst\_time[i] < burst\_time[smallest])

                smallest = i;

            }

        printf("P[%d]\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t\n", smallest + 1, arrival\_time[smallest], burst\_time[smallest], time - arrival\_time[smallest], time + burst\_time[smallest] - arrival\_time[smallest]);

        total\_turnaround\_time += time + burst\_time[smallest] - arrival\_time[smallest];

        total\_waiting\_time += time - arrival\_time[smallest];

        time += burst\_time[smallest];

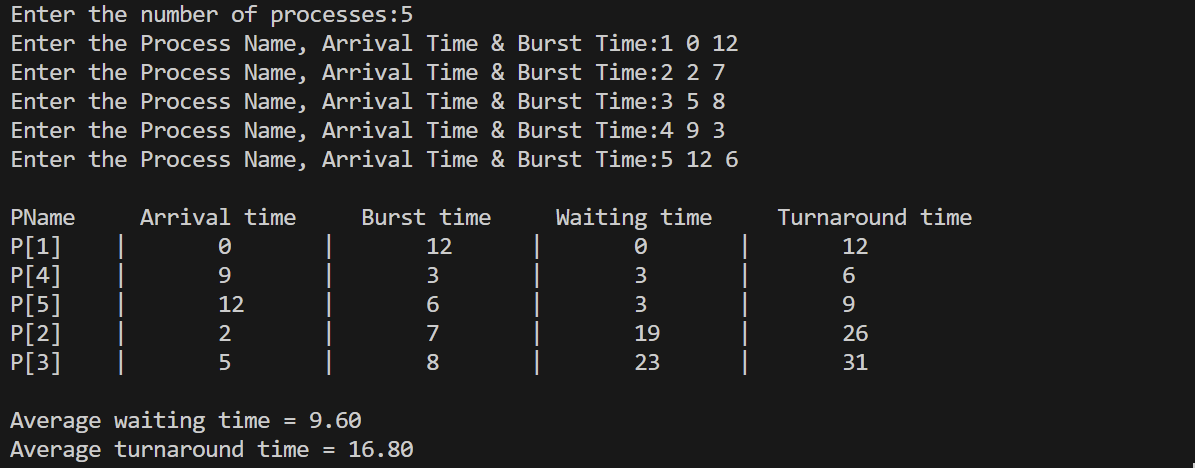
        burst\_time[smallest] = 0;

    }

    printf("\nAverage waiting time = %.2f", total\_waiting\_time \* 1.0 / n);

    printf("\nAverage turnaround time = %.2f", total\_turnaround\_time \* 1.0 / n);

}



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

#include<stdio.h>

void main(){

    int pn[10];

    int arrival\_time[10], burst\_time[10], remain\_time[10], endTime, smallest, i, n;

    int total\_waiting\_time = 0, total\_turnaround\_time = 0, time, remain = 0;

    printf("Enter the number of processes:");

    scanf("%d", &n);

    for(i = 0; i < n; i++){

        printf("Enter the Process Name, Arrival Time & Burst Time:");

        scanf("%d%d%d", &pn[i], &arrival\_time[i], &burst\_time[i]);

        remain\_time[i] = burst\_time[i];

    }

    remain\_time[9] = 9999;

    printf("\nPName     Arrival time     Burst time     Waiting time     Turnaround time\n");

    for(time=0;remain!=n;time++) {

        smallest=9;

        for(i=0;i<n;i++){

            if(arrival\_time[i]<=time && remain\_time[i]<remain\_time[smallest] && remain\_time[i]>0){

                smallest=i;

            }

        }

        remain\_time[smallest]--;

        if(remain\_time[smallest]==0){

            remain++;

            endTime=time+1;

            printf("\nP[%d]\t|\t%d\t|\t%d\t|\t%d\t|\t%d", smallest + 1, arrival\_time[smallest], burst\_time[smallest],endTime - burst\_time[smallest] - arrival\_time[smallest], endTime - arrival\_time[smallest]);

            total\_waiting\_time += endTime - burst\_time[smallest]-arrival\_time[smallest];

            total\_turnaround\_time += endTime-arrival\_time[smallest];

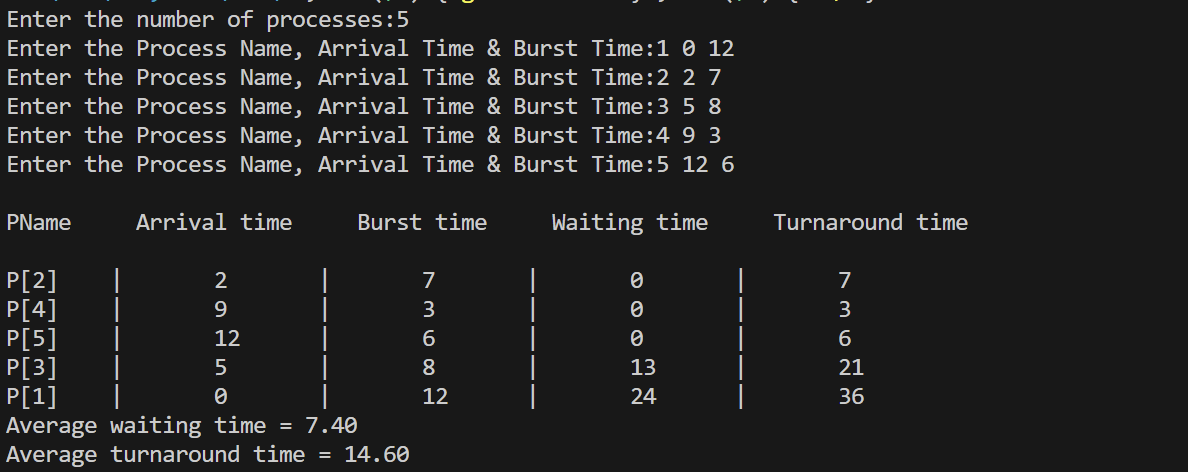
        }

    }

    printf("\nAverage waiting time = %.2f", total\_waiting\_time \* 1.0 / n);

    printf("\nAverage turnaround time = %.2f", total\_turnaround\_time \* 1.0 / n);

}



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

#include <stdio.h>

void main() {

    int i, n, total = 0, x, counter = 0, time\_quantum;

    int waiting\_time = 0, turn\_around\_time = 0, pn[10], arrival\_time[10], burst\_time[10], temp[10];

    float total\_waiting\_time = 0, total\_turnaround\_time = 0;

    printf("\nEnter numbers of processes: "); scanf("%d", &n); x = n;

    for(i = 0; i < n; i++) {

        printf("\nEnter Process Name, Arrival Time, Burst time: ");

        scanf("%d%d%d", &pn[i], &arrival\_time[i], &burst\_time[i]);

        temp[i] = burst\_time[i];

    }

    printf("\nEnter time quantum: ");

    scanf("%d", &time\_quantum);

    printf("\nProcess name\tArrival Time\tBurst Time\t Turnaround TIme\t Waiting TIme\n");

    for(total = 0, i = 0; x !=0;) {

        if(temp[i] <= time\_quantum && temp[i] > 0) {

            total = total + temp[i];

            temp[i] = 0;

            counter = 1;

        }

        else if(temp[i] > 0) {

            temp[i] = temp[i] - time\_quantum;

            total = total + time\_quantum;

        }

        if(temp[i] == 0 && counter == 1) {

            x--;

            printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", pn[i], burst\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]); waiting\_time +=  total - arrival\_time[i] - burst\_time[i];

            turn\_around\_time += total - arrival\_time[i];

            counter = 0;

        }

        if (i == n - 1)

            i = 0;

        else if ( arrival\_time[i+1] <= total )

            i++;

        else i=0;

    }

    total\_waiting\_time = waiting\_time \* 1.0/n;

    total\_turnaround\_time = turn\_around\_time \* 1.0/n;

    printf("\nAverage waiting time: %f", total\_waiting\_time); printf("\nAverage turnaround time: %f\n", total\_turnaround\_time);

}

