به نام خدا

تمرین امتیازی

فرزان رحمانی ۹۹۵۲۱۲۷۱

۱- FCFS

```
farzanrahmani_99521271@DES
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  Priority.c  RR.c  SJF.c
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ sudo cc FCFS.c -o FCFS.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  FCFS.out  Priority.c  RR.c  SJF.c
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ./FCFS.out
what is number of processes?
3
what is service time of process with pid = 1
4
what is service time of process with pid = 2
2
what is service time of process with pid = 3
1
process with pid:0 wait time:0, service time:4, total time: 4
process with pid:1 wait time:4, service time:2, total time: 6
process with pid:2 wait time:6, service time:1, total time: 7
avg wtime=3.333333
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$
```

```c
#include <stdio.h>
struct process
{
    int pid, st, wt, tt;
};

int main()
{
    int processes_num;
    printf("what is number of processes? \n");
    scanf("%d", &processes_num);
```

```
    struct process processes[processes_num];

    for (int i = 0; i < processes_num; i++)
    {
        int service_time;
        printf("what is service time of process with pid = %d\n", i + 1);
        scanf("%d", &service_time);
        processes[i].st = service_time;
        processes[i].pid = i + 1;
    }

    int total_waiting = 0;
    for (int i = 0; i < processes_num; i++)
    {
        if (i == 0)
            processes[i].wt = 0;
        else
            processes[i].wt = processes[i - 1].tt;
        total_waiting += processes[i].wt;
        processes[i].tt = processes[i].wt + processes[i].st;
        printf("process with pid:%d wait time:%d, service time:%d, total time:
%d\n", i, processes[i].wt, processes[i].st, processes[i].tt);
    }
    double avg_wtime = (double) total_waiting / (double) processes_num;
    printf("avg wtime=%f\n", avg_wtime);

    return 0;
}
```

SJF -۲

```
farzanrahmani_99521271@DES
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  FCFS.out  Priority.c  RR.c  SJF.c
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ sudo cc SJF.c -o SJF.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  FCFS.out  Priority.c  RR.c  SJF.c  SJF.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ./SJF.out
what is number of processes?
4
what is service time of process with pid = 1
5
what is service time of process with pid = 2
7
what is service time of process with pid = 3
2
what is service time of process with pid = 4
4
process with pid:0 wait time:0, service time:2, total time: 2
process with pid:1 wait time:2, service time:4, total time: 6
process with pid:2 wait time:6, service time:5, total time: 11
process with pid:3 wait time:11, service time:7, total time: 18
avg wtime=4.750000
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$
```

```c
#include <stdio.h>
struct process
{
    int pid, st, wt, tt;
};

int main()
{
    int processes_num;
    printf("what is number of processes? \n");
    scanf("%d", &processes_num);

    struct process processes[processes_num];

    for (int i = 0; i < processes_num; i++)
    {
        int service_time;
        printf("what is service time of process with pid = %d\n", i + 1);
```

```
        scanf("%d", &service_time);
        processes[i].st = service_time;
        processes[i].pid = i + 1;
    }

    int i, j;
    struct process swap_temp;
    // sorting
    for (i = 0; i < processes_num - 1; i++)
    {
        for (j = 0; j < processes_num - i - 1; j++)
        {
            if (processes[j].st > processes[j + 1].st)
            {
                swap_temp = processes[j];
                processes[j] = processes[j + 1];
                processes[j + 1] = swap_temp;
            }
        }
    }

    int total_waiting = 0;
    for (int i = 0; i < processes_num; i++)
    {
        if (i == 0)
            processes[i].wt = 0;
        else
            processes[i].wt = processes[i - 1].tt;
        total_waiting += processes[i].wt;
        processes[i].tt = processes[i].wt + processes[i].st;
        printf("process with pid:%d wait time:%d, service time:%d, total time:
%d\n", i, processes[i].wt, processes[i].st, processes[i].tt);
    }
    double avg_wtime = (double) total_waiting / (double) processes_num;
    printf("avg wtime=%f\n", avg_wtime);

    return 0;
}
```

Priority -۳

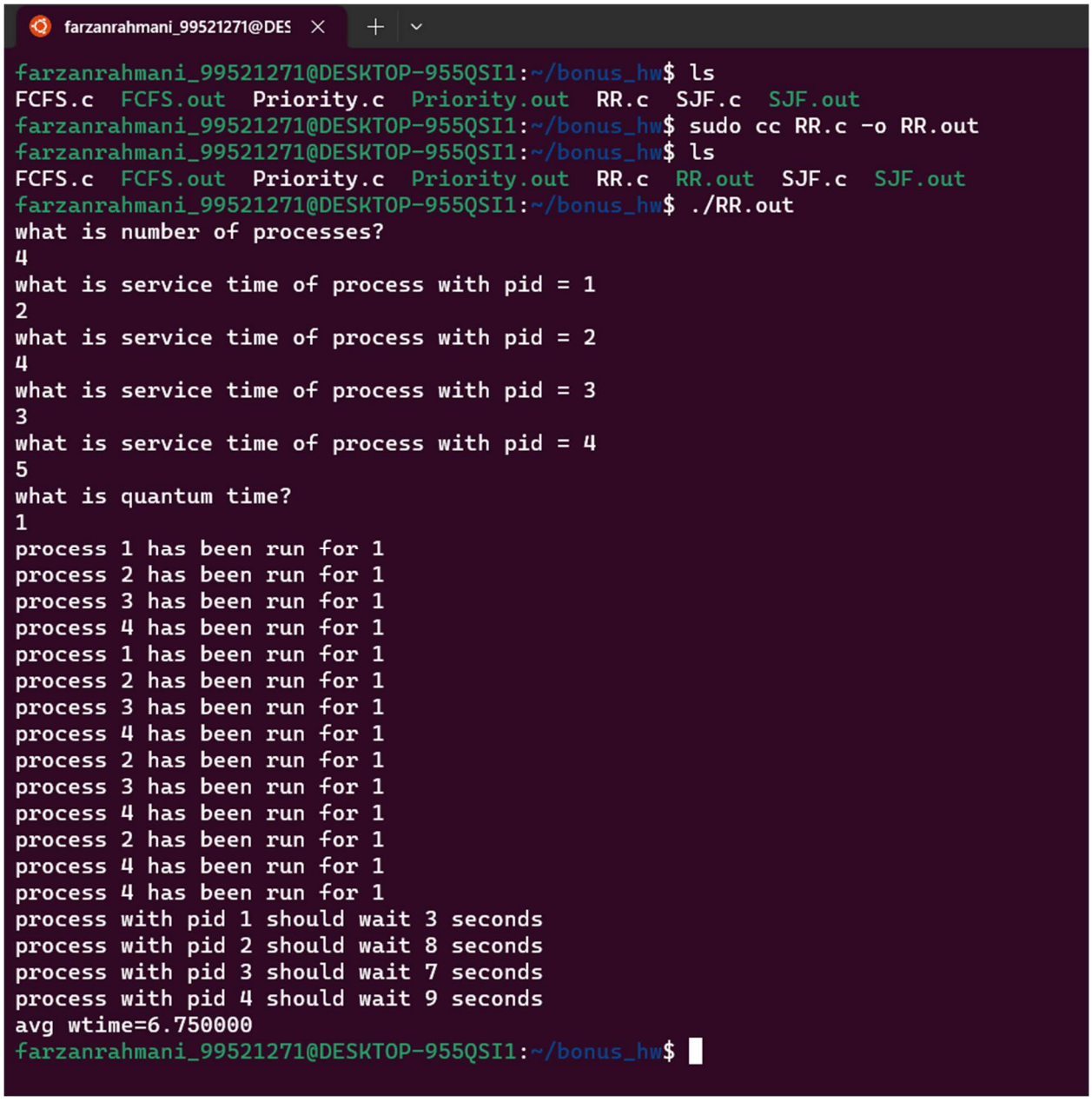
```
farzanrahmani_99521271@DES
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  FCFS.out  Priority.c  RR.c  SJF.c  SJF.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ sudo cc Priority.c -o Priority.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ls
FCFS.c  FCFS.out  Priority.c  Priority.out  RR.c  SJF.c  SJF.out
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$ ./Priority.out
what is number of processes?
4
what is service time of process with pid:1
3
what is priority of process with pid 1
3
what is service time of process with pid:2
2
what is priority of process with pid 2
2
what is service time of process with pid:3
8
what is priority of process with pid 3
1
what is service time of process with pid:4
7
what is priority of process with pid 4
4
process with pid:0 wait time:0, service time:7, total time: 7, priority time:4
process with pid:1 wait time:7, service time:3, total time: 10, priority time:3
process with pid:2 wait time:10, service time:2, total time: 12, priority time:2
process with pid:3 wait time:12, service time:8, total time: 20, priority time:1
avg wtime=7.250000
farzanrahmani_99521271@DESKTOP-955QSI1:~/bonus_hw$
```

```c
#include <stdio.h>
struct process
{
    int pid, st, wt, tt, priority;
};

int main()
{
    int processes_num;
    printf("what is number of processes? \n");
    scanf("%d", &processes_num);

    struct process processes[processes_num];

    for (int i = 0; i < processes_num; i++)
    {
        // assign service time
        int service_time;
```

```
        printf("what is service time of process with pid:%d\n", i + 1);
        scanf("%d", &service_time);
        processes[i].st = service_time;

        // assign priority
        int priority;
        printf("what is priority of process with pid %d\n", i + 1);
        scanf("%d", &priority);
        processes[i].priority = priority;

        // assign id
        processes[i].pid = i + 1;
    }

    int i, j;
    struct process swap_temp;
    // sorting
    for (i = 0; i < processes_num - 1; i++)
    {
        for (j = 0; j < processes_num - i - 1; j++)
        {
            if (processes[j].priority < processes[j + 1].priority)
            {
                swap_temp = processes[j];
                processes[j] = processes[j + 1];
                processes[j + 1] = swap_temp;
            }
        }
    }

    int total_waiting = 0;
    for (int i = 0; i < processes_num; i++)
    {
        if (i == 0)
            processes[i].wt = 0;
        else
            processes[i].wt = processes[i - 1].tt;
        total_waiting += processes[i].wt;
        processes[i].tt = processes[i].wt + processes[i].st;
        printf("process with pid:%d wait time:%d, service time:%d, total time:
%d, priority time:%d\n", i, processes[i].wt, processes[i].st, processes[i].tt,
processes[i].priority);
    }
    double avg_wtime = (double) total_waiting / (double) processes_num;
    printf("avg wtime=%f\n", avg_wtime);
```

```
    return 0;
}
```

Round Robin -۴



```c
#include <stdio.h>
struct process
{
    int pid, st, wt, tt, finished;
};

int main()
{
    int quantum_time = 0;
    int processes_num = 0;
    printf("what is number of processes? \n");
    scanf("%d", &processes_num);

    struct process processes[processes_num];

    for (int i = 0; i < processes_num; i++)
    {
        int service_time;
        printf("what is service time of process with pid = %d\n", i + 1);
        scanf("%d", &service_time);
        processes[i].st = service_time;
        processes[i].pid = i + 1;
        processes[i].tt += service_time;
        processes[i].wt = 0;
    }

    printf("what is quantum time? \n");
    scanf("%d", &quantum_time);

    int total_waiting = 0;
    int finished_processes = 0;
    int i = 0;
    while (finished_processes < processes_num)
    {
        if (processes[i].st > 0)
        {
            if (processes[i].st > quantum_time)
            {
                for (int j = 0; j < processes_num; j++)
                {
                    if (processes[j].finished != 1 && j != i)
                        processes[j].wt += quantum_time;
                }

                processes[i].st -= quantum_time;
```

```
                printf("process %d has been run for %d\n", processes[i].pid,
quantum_time);
            }
            else
            {
                for (int j = 0; j < processes_num; j++)
                {
                    if (j != i && processes[j].finished != 1)
                        processes[j].wt += processes[i].st;
                }

                processes[i].st = 0;
                printf("process %d has been run for %d\n", processes[i].pid,
quantum_time);
                processes[i].finished = 1;
                finished_processes++;
            }
        }
        i = (i + 1) % processes_num;
    }

    for (int i = 0; i < processes_num; i++)
    {
        printf("process with pid %d should wait %d seconds\n", i + 1,
processes[i].wt);
        total_waiting += processes[i].wt;
    }

    double avg_wtime = (double) total_waiting / (double) processes_num;
    printf("avg wtime=%f\n", avg_wtime);

    return 0;
}
```

پایان