# Operating Systems Lab

## Lab-4 Report

Pavan Kumar V Patil
200030041

January 30, 2023

# 1 Shortest Job First scheduler

## 1.a Scheduling schema (Non-preemptive)

### 1.a.1 Variables used

- current_time: holds the value of the current time.
- Process: a struct that holds the information of a process, such as the PID, turn-around time, waiting time, penalty ratio, two queues to store the CPU and IO burst times, arrival time, etc.
- priority_queue: two priority queues named CPU_queue and IO_queue are used to hold the current CPU-bound and IO-bound processes, respectively. Two temporary priority queues named temp_CPU_queue and temp_IO_queue are also defined.
- unordered_map: holds the information of all processes, with the PID as the key and a Process struct as the value.

### 1.a.2 Functions implemented

- build_process: takes in the PID and arrival time of a process and returns a Process struct with all its details.
- check_CPU_bound: takes in a PID and returns whether the process with the given PID is CPU-bound or not.
- check_IO_bound: takes in a PID and returns whether the process with the given PID is IO-bound or not.
- add_ready_queue: this function adds processes to the CPU_queue or IO_queue if they are ready to be processed.
- Start_functioning: the main function that implements the SJF algorithm. It first adds all ready processes to the CPU_queue or IO_queue, then selects the process with the shortest job time and decrements its time by 1 until it reaches 0. The function continues this process until there are no more processes left in both CPU_queue and IO_queue.

## 1.b Characteristics of SJF

- Optimization of Waiting Time: SJF aims to minimize the total waiting time for all processes, leading to a more efficient use of resources and faster completion times.
- Fairness: SJF can be considered fair, as all processes are given a fair chance to run based on their CPU burst time. This means that processes with shorter CPU burst times will get to run more frequently than those with longer burst times.
- Good Throughput: SJF has a good throughput, as it always selects the shortest process first and as a result, the total completion time of processes is reduced.

## 1.c Suitable test data for SJF

```
0 1 -1
1 10 2 -1
11 10 2 -1
```

```
pavan@pavan-OMEN-Laptop-15-ek0xxx:~/Documents/OS minix/os-lab/lab4$ ./SJF.exe
PID: 0, Turn around time: 0, Waiting time: 0, Penalty ratio: 0
PID: 1, Turn around time: 11, Waiting time: 0, Penalty ratio: 0.846154
PID: 2, Turn around time: 11, Waiting time: 0, Penalty ratio: 0.478261
--------------------------------------------------------------------------------

Total Through put: 0.136364
Average Turn around time: 7.33333
Average Waiting time: 0
Average Penalty ratio: 0.441472
--------------------------------------------------------------------------------
```

Figure 1: Suitable test data for SJF output

### 1.d Shortcomings test data for SJF

```
0 1000 -1
1 10 2 -1
11 10 2 -1
```

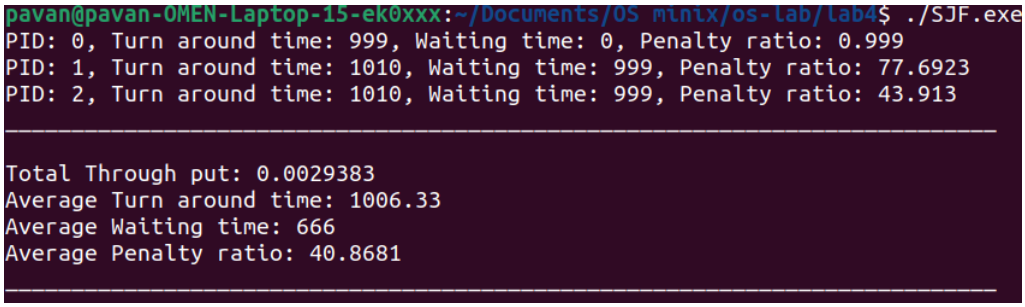


Figure 2: Shortcomings test data for SJF output, Process of PID 1 and 2 leads to Convoy Effect

# 2 Round Robin Scheduling

## 2.a Scheduling schema (Preemptive

### 2.a.1 Variables used

- current_time: holds the value of the current time.
- timeslice: represents the time slice given to each process in the CPU.
- Process: a struct that holds the information of a process, such as the PID, turn-around time, waiting time, penalty ratio, two queues to store the CPU and IO burst times, arrival time, etc.
- queue: two queues named CPU_queue and IO_queue are used to hold the current CPU-bound and IO-bound processes, respectively. Two temporary priority queues named temp_CPU_queue and temp_IO_queue are also defined.
- unordered_map: holds the information of all processes, with the PID as the key and a Process struct as the value.

### 2.a.2 Functions implemented

- build_process: takes in the PID and arrival time of a process and returns a Process struct with all its details.
- check_CPU_bound: takes in a PID and returns whether the process with the given PID is CPU-bound or not.
- check_IO_bound: takes in a PID and returns whether the process with the given PID is IO-bound or not.
- add_ready_queue: this function adds processes to the CPU_queue or IO_queue if they are ready to be processed.

- Start_functioning: the main function that implements the Round Robin algorithm, with a given timeslice. The function runs a loop to keep executing processes until there are no more processes to execute (both CPU and IO queues are empty). It calls the add_to_ready_queue function to add any newly arrived processes to the CPU or IO queue. It then checks if there are any processes in the IO queue, and if so, assigns the first process to be the current IO process and assigns its burst time. Then it checks if there are any processes in the CPU queue, and if so, decrements the burst time of the first process and updates its cpu_time accordingly. When time slice of particular process completes it puts it to back of the queue. If the burst time of a CPU process reaches 0, the process is removed from the CPU queue and its turn_around_time is updated. The loop repeats until all processes have completed execution.

### 2.b Characteristics of RR

- No Starvation: Every process is guaranteed a certain amount of time to execute, preventing starvation.
- Efficient utilization of CPU time: The CPU time is utilized effectively as each process gets a fair share of the CPU.
- Low overhead: The overhead of context switching is low as the quantum time is usually small, making it an efficient algorithm.
- Fairness: It is fair as it gives equal time slice to each and every process.

### 2.c Suitable test data for RR

```
0 5 -1
5 5 2 -1
10 10 2 -1
```

```
pavan@pavan-OMEN-Laptop-15-ek0xxx:~/Documents/OS minix/os-lab/lab4$ ./RR.exe
PID: 0, Turn around time: 4, Waiting time: 0, Penalty ratio: 0.8
PID: 1, Turn around time: 6, Waiting time: 0, Penalty ratio: 0.5
PID: 2, Turn around time: 11, Waiting time: 0, Penalty ratio: 0.5
--------------------------------------------------------------------------------

Total Through put: 0.142857
Average Turn around time: 7
Average Waiting time: 0
Average Penalty ratio: 0.6
--------------------------------------------------------------------------------
```

Figure 3: Suitable test data for RR output, timeslice is 5

### 2.d Shortcomings test data for RR

```
0 100 -1
1 100 2 -1
11 100 2 -1
```

```
pavan@pavan-OMEN-Laptop-15-ek0xxx:~/Documents/OS minix/os-lab/lab4$ ./RR.exe
PID: 0, Turn around time: 284, Waiting time: 185, Penalty ratio: 2.84
PID: 1, Turn around time: 290, Waiting time: 188, Penalty ratio: 2.81553
PID: 2, Turn around time: 290, Waiting time: 188, Penalty ratio: 2.56637
--------------------------------------------------------------------------------

Total Through put: 0.00996678
Average Turn around time: 288
Average Waiting time: 187
Average Penalty ratio: 2.74064
--------------------------------------------------------------------------------
```

Figure 4: Shortcomings test data for RR output, timeslice is 5

## 3 Analysis

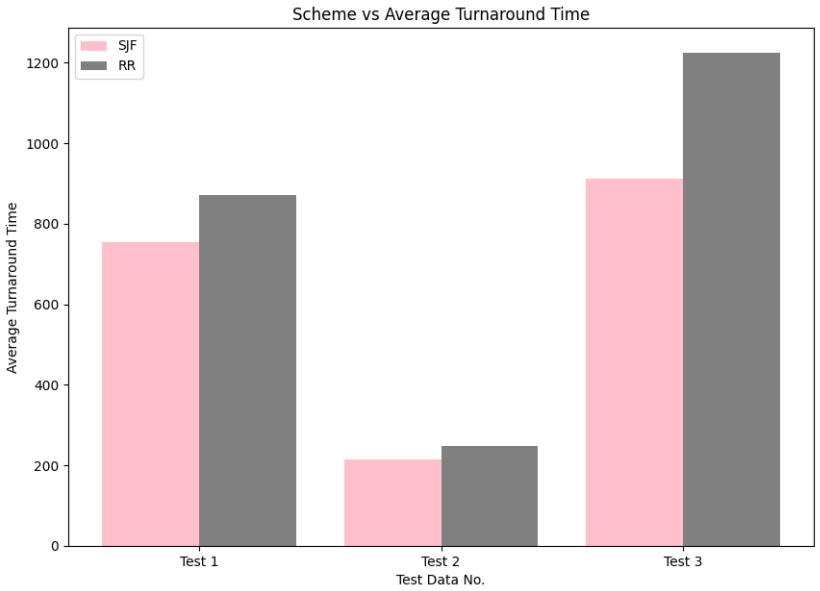


Figure 5: Scheme vs Avearage turnaround time, RR timeslice is 5


Figure 6: Scheme vs Average Waiting Time, RR timeslice is 5

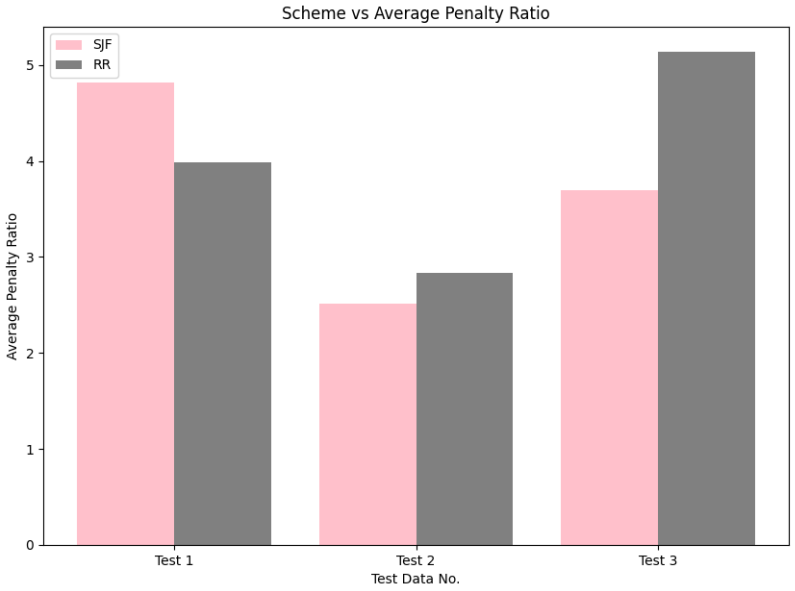


Figure 7: Scheme vs Average Penalty Ratio, RR timeslice is 5

From the above analysis we can infer that:

- Round robin scheduling has more average waiting and turnaround time compared to Shortest job first scheduler.
- But, response time of RR is better compared to SJF because it gives equal timeslice to run for each and every process therefore each process gets CPU faster.
- SJF can lead to convoy effect for some processes sequence but there will be no starvation and Convoy effect in RR.
- Average Penalty ratio of SJF is smaller than RR.
- SJF can also lead to Starvation for long processes if shorter processes keep coming in.

# 4 Output of SJF

```
g++ SJF.cpp -o SJF.exe
./SJF.exe ./test/process1.dat
PID: 0, Turn around time: 1223, Waiting time: 803, Penalty ratio:
    2.90499
PID: 1, Turn around time: 1211, Waiting time: 871, Penalty ratio:
    3.53061
PID: 2, Turn around time: 660, Waiting time: 370, Penalty ratio:
    2.2449
PID: 3, Turn around time: 549, Waiting time: 349, Penalty ratio:
    2.67805
PID: 4, Turn around time: 118, Waiting time: 106, Penalty ratio:
    6.55556
PID: 5, Turn around time: 101, Waiting time: 97, Penalty ratio:
    9.18182
PID: 6, Turn around time: 1418, Waiting time: 1214, Penalty ratio:
     6.59535
--------------------------------------------------------------------------------


Total Through put: 0.00490196
Average Turn around time: 754.286
Average Waiting time: 544.286
Average Penalty ratio: 4.81304
--------------------------------------------------------------------------------


./SJF.exe ./test/process2.dat
PID: 0, Turn around time: 4, Waiting time: 0, Penalty ratio: 0.8
PID: 1, Turn around time: 82, Waiting time: 70, Penalty ratio:
    5.85714
PID: 2, Turn around time: 480, Waiting time: 280, Penalty ratio:
    2.31884
PID: 3, Turn around time: 75, Waiting time: 63, Penalty ratio:
    2.08333
PID: 4, Turn around time: 812, Waiting time: 522, Penalty ratio:
    2.57778
PID: 5, Turn around time: 76, Waiting time: 64, Penalty ratio: 2
PID: 6, Turn around time: 934, Waiting time: 594, Penalty ratio:
    2.54496
```

```
PID: 7, Turn around time: 89, Waiting time: 77, Penalty ratio: 2.225
PID: 8, Turn around time: 228, Waiting time: 192, Penalty ratio: 3.50769
PID: 9, Turn around time: 90, Waiting time: 78, Penalty ratio: 2.14286
PID: 10, Turn around time: 103, Waiting time: 91, Penalty ratio: 2.34091
PID: 11, Turn around time: 104, Waiting time: 92, Penalty ratio: 2.26087
PID: 12, Turn around time: 117, Waiting time: 105, Penalty ratio: 2.4375
PID: 13, Turn around time: 115, Waiting time: 103, Penalty ratio: 2.16981
PID: 14, Turn around time: 130, Waiting time: 118, Penalty ratio: 2.45283
PID: 15, Turn around time: 131, Waiting time: 119, Penalty ratio: 2.38182
PID: 16, Turn around time: 145, Waiting time: 133, Penalty ratio: 2.58929
PID: 17, Turn around time: 146, Waiting time: 134, Penalty ratio: 2.51724
```

---

```
Total Through put: 0.01875
Average Turn around time: 214.5
Average Waiting time: 157.5
Average Penalty ratio: 2.51155
```

---

```
./SJF.exe ./test/process3.dat
PID: 0, Turn around time: 352, Waiting time: 125, Penalty ratio: 1.54386
PID: 1, Turn around time: 1690, Waiting time: 1397, Penalty ratio: 5.70946
PID: 2, Turn around time: 767, Waiting time: 464, Penalty ratio: 2.4822
PID: 3, Turn around time: 1349, Waiting time: 1182, Penalty ratio: 7.66477
PID: 4, Turn around time: 445, Waiting time: 277, Penalty ratio: 2.45856
PID: 5, Turn around time: 1918, Waiting time: 1537, Penalty ratio: 4.77114
PID: 6, Turn around time: 817, Waiting time: 582, Penalty ratio: 3.07143
PID: 7, Turn around time: 5, Waiting time: 0, Penalty ratio: 0.121951
PID: 8, Turn around time: 81, Waiting time: 22, Penalty ratio: 0.84375
PID: 9, Turn around time: 1435, Waiting time: 1336, Penalty ratio: 10.4745
PID: 10, Turn around time: 17, Waiting time: 9, Penalty ratio: 0.361702
PID: 11, Turn around time: 2066, Waiting time: 1678, Penalty ratio : 4.81585
```

---

```
Total Through put: 0.00569801
Average Turn around time: 911.833
Average Waiting time: 717.417
Average Penalty ratio: 3.69326
```

---

# 5 Ouput of RR

```
g++ RR.cpp -o RR.exe
./RR.exe ./test/process1.dat
PID: 0, Turn around time: 1430, Waiting time: 999, Penalty ratio:
    3.39667
PID: 1, Turn around time: 1344, Waiting time: 998, Penalty ratio:
    3.91837
PID: 2, Turn around time: 1243, Waiting time: 950, Penalty ratio:
    4.22789
PID: 3, Turn around time: 954, Waiting time: 751, Penalty ratio:
    4.65366
PID: 4, Turn around time: 83, Waiting time: 71, Penalty ratio:
    4.61111
PID: 5, Turn around time: 26, Waiting time: 21, Penalty ratio:
    2.36364
PID: 6, Turn around time: 1021, Waiting time: 814, Penalty ratio:
    4.74884
```

---

```
Total Through put: 0.00489511
Average Turn around time: 871.571
Average Waiting time: 657.714
Average Penalty ratio: 3.9886
```

---

```
./RR.exe ./test/process2.dat
PID: 0, Turn around time: 4, Waiting time: 0, Penalty ratio: 0.8
PID: 1, Turn around time: 22, Waiting time: 8, Penalty ratio:
    1.57143
PID: 2, Turn around time: 690, Waiting time: 483, Penalty ratio:
    3.33333
PID: 3, Turn around time: 102, Waiting time: 88, Penalty ratio:
    2.83333
PID: 4, Turn around time: 877, Waiting time: 577, Penalty ratio:
    2.78413
PID: 5, Turn around time: 120, Waiting time: 106, Penalty ratio:
    3.15789
PID: 6, Turn around time: 934, Waiting time: 584, Penalty ratio:
    2.54496
PID: 7, Turn around time: 132, Waiting time: 118, Penalty ratio:
    3.3
PID: 8, Turn around time: 243, Waiting time: 203, Penalty ratio:
    3.73846
PID: 9, Turn around time: 143, Waiting time: 131, Penalty ratio:
    3.40476
PID: 10, Turn around time: 144, Waiting time: 129, Penalty ratio:
    3.27273
```

```
PID: 11, Turn around time: 145, Waiting time: 130, Penalty ratio: 3.15217
PID: 12, Turn around time: 146, Waiting time: 131, Penalty ratio: 3.04167
PID: 13, Turn around time: 152, Waiting time: 137, Penalty ratio: 2.86792
PID: 14, Turn around time: 149, Waiting time: 137, Penalty ratio: 2.81132
PID: 15, Turn around time: 153, Waiting time: 138, Penalty ratio: 2.78182
PID: 16, Turn around time: 155, Waiting time: 140, Penalty ratio: 2.76786
PID: 17, Turn around time: 161, Waiting time: 149, Penalty ratio: 2.77586
_____


Total Through put: 0.01875
Average Turn around time: 248.444
Average Waiting time: 188.278
Average Penalty ratio: 2.82998
_____


./RR.exe ./test/process3.dat
PID: 0, Turn around time: 1487, Waiting time: 1249, Penalty ratio: 6.52193
PID: 1, Turn around time: 1920, Waiting time: 1624, Penalty ratio: 6.48649
PID: 2, Turn around time: 1862, Waiting time: 1533, Penalty ratio: 6.02589
PID: 3, Turn around time: 1344, Waiting time: 1176, Penalty ratio: 7.63636
PID: 4, Turn around time: 1100, Waiting time: 924, Penalty ratio: 6.07735
PID: 5, Turn around time: 2070, Waiting time: 1665, Penalty ratio: 5.14925
PID: 6, Turn around time: 1627, Waiting time: 1389, Penalty ratio: 6.11654
PID: 7, Turn around time: 95, Waiting time: 89, Penalty ratio: 2.31707
PID: 8, Turn around time: 181, Waiting time: 121, Penalty ratio: 1.88542
PID: 9, Turn around time: 905, Waiting time: 803, Penalty ratio: 6.60584
PID: 10, Turn around time: 102, Waiting time: 94, Penalty ratio: 2.17021
PID: 11, Turn around time: 2017, Waiting time: 1598, Penalty ratio : 4.70163
_____


Total Through put: 0.00574163
Average Turn around time: 1225.83
Average Waiting time: 1022.08
Average Penalty ratio: 5.14117
_____
```