**2. Simple Burst Demonstrations**
**Example of FCFS superior turn around time:**

First Come First Serve will have superior turn around time when the jobs come shortest first

Processes:
p1 1 2
p2 2 3
p3 3 7
FCFS results:
1:p1 3:p2 6:p3 13:END
Average Turnaround Time: 5.333333333333333
Average Response Time: 1.3333333333333333
Timeslice? 2
RR results:
1:p1 3:p2 5:p3 7:p2 8:p3 10:p3 12:p3 13:END
Average Turnaround Time: 6.0
Average Response Time: 1.0

**Example of RR superior turn around time:**

Round Robin will have superior turn around time when the time the first processes take a long time to run

Processes:
p1 1 7
p2 2 3
p3 3 2
FCFS results:
1:p1 8:p2 11:p3 13:END
Average Turnaround Time: 8.666666666666666
Average Response Time: 4.666666666666667
Timeslice? 2
RR results:
1:p1 3:p2 5:p3 7:p1 9:p2 10:p1 12:p1 13:END
Average Turnaround Time: 8.0
Average Response Time: 1.0

**Example of SJF superior turn around time:**

SJF results from the first set of processes above:
1:p1 3:p2 6:p3 13:END
Average Turnaround Time: 5.333333333333333
Average Response Time: 1.3333333333333333
SJF results from the second set of processes above:
1:p1 2:p2 5:p3 7:p1 13:END
Average Turnaround Time: 6.333333333333333
Average Response Time: 0.6666666666666666

**3. Complex Burst Demonstrations**
**SJF is a pre-emptive algorithm:**

Here you can see p2 interrupts p1 as soon as it arrives.

How many context switches? 20
Processes:
p1 0 6
p2 2 3
SJF results:
CPU: 0:p1 2:p2 5:p1 9:END
IO: 0:END
Average Turnaround Time: 6.0
Average Response Time: 0.0

**Starvation Example:**

How many context switches? 20
Processes:
p1 0 3 3 3
p2 2 1 1 -1
p3 3 1 1 -1
SJF results:
CPU: 0:p1 3:p2 4:p3 5:p2 6:p3 7:p2 8:p3 9:p2 10:p3 11:p2 12:p3 13:p2 14:p3 15:p2 16:p3 17:p2 18:p3 19:p2 20:p3 21:p2 22:END
IO: 0:IDLE 3:p1 3:IDLE 4:p2 4:IDLE 5:p3 5:IDLE 6:p2 6:IDLE 7:p3 7:IDLE 8:p2 8:IDLE 9:p3 9:IDLE 10:p2 10:IDLE 11:p3 11:IDLE 12:p2 12:IDLE 13:p3 13:IDLE 14:p2 14:IDLE 15:p3 15:IDLE 16:p2 16:IDLE 17:p3 17:IDLE 18:p2 18:IDLE 19:p3 19:IDLE 20:p2 20:IDLE 21:p3 21:IDLE 22:p2 22:END
No finished processes

**FCFS and RR don't starve:**

FCFS results:
CPU: 0:p1 3:p2 4:p3 5:p2 6:p3 7:p1 10:p2 11:p3 12:p2 13:p3 14:p2 15:p3 16:p2 17:p3 18:p2 19:p3 20:p2 21:p3 22:p2 23:p3 24:END
IO: 0:IDLE 3:p1 3:p2 4:p3 5:p2 6:p3 7:p2 8:p3 9:p2 10:p3 11:p2 12:p3 13:p2 14:p3 15:p2 16:p3 17:p2 18:p3 19:p2 20:p3 21:END
Average Turnaround Time: 10.0
Average Response Time: 0.0
Timeslice? 2
RR results:
CPU: 0:p1 2:p2 3:p1 4:p3 5:p2 6:p3 7:p1 9:p2 10:p3 11:p1 12:p2 13:p3 14:p2 15:p3 16:p2 17:p3 18:p2 19:p3 20:p2 21:p3 22:END
IO: 0:IDLE 3:p2 1:IDLE 3:p1 4:p3 5:p2 6:p3 7:p2 8:p3 9:p2 10:p3 11:p2 12:p3 13:p2 14:p3 15:p2 16:p3 17:p2 18:p3 19:END
Average Turnaround Time: 12.0
Average Response Time: 0.0

**4. Approximating SJF with exponential average**
**Various alpha and initial taus:**

How many context switches? 25
Processes:
p1 0 3 3 3
p2 2 3 4 -1
p3 3 1 1 -1

p4 9 2 1 4 5 2

Actual burst weight? 0.8
Initial prediction? 1
Approximation results:
    Average Turnaround Time: 16.0
    Average Response Time: 1.0

Actual burst weight? 0.2
Initial prediction? 1
Approximation results:
    Average Turnaround Time: 18.5
    Average Response Time: 1.0

Actual burst weight? 0.8
Initial prediction? 3
Approximation results:
    Average Turnaround Time: 15.5
    Average Response Time: 1.5

Actual burst weight? 0.2
Initial prediction? 3
Approximation results:
    Average Turnaround Time: 18.5
    Average Response Time: 1.0

Actual burst weight? 0.8
Initial prediction? 7
Approximation results:
    Average Turnaround Time: 18.5
    Average Response Time: 3.0

Actual burst weight? 0.2
Initial prediction? 7
Approximation results:
    Not all finite processes completed

Initial prediction seemed to affect the response time the most to the point where if you put enough weight on a high prediction, it took more context switches for the finite processes to complete.

The turn around time was always better when I put more weight on the previous actual burst than the prediction but that may be because the dataset I chose had relatively close burst sizes.

**Prediction accuracy:**

How many context switches? 15
Processes:
p1 0 3 3 5
p2 2 4 4 1 2 -1
p3 9 2 1 4 5 2

Actual burst weight? 0.8
Initial prediction? 5

Approximation results:

| | | | | |
|---|---|---|---|---|
| p1<br>Actual burst: 3<br>Approximation: 5 | p2<br>Actual burst: 4<br>Approximation: 5 | p3<br>Actual burst: 2<br>Approximation: 5 | p1<br>Actual burst: 5<br>Approximation: 3 | p2<br>Actual burst: 1<br>Approximation: 4 |
| p1<br>Actual burst: None<br>Approximation: 2 | p2<br>Actual burst: 4<br>Approximation: 2 | p3<br>Actual burst: 4<br>Approximation: 3 | p2<br>Actual burst: 1<br>Approximation: 4 | p3<br>Actual burst: 2<br>Approximation: 4 |
| p2<br>Actual burst: 4<br>Approximation: 2 | p2<br>Actual burst: 1<br>Approximation: 4 | p3<br>Actual burst: None<br>Approximation: 4 | p2<br>Actual burst: 4<br>Approximation: 2 | p2<br>Actual burst: 1<br>Approximation: 4 |

I don't really know how to measure the accuracy of the approximation but in general for processes with steady CPU bursts I think it would be much better than one's with unsteady CPU bursts. I think with less steady CPU burst values it would help to weight the prediction over the actual.

**5. Managing Starvation with a Lottery**
**No more starvation!**

How many context switches? 15

Processes:
p1 0 3 3 3
p2 2 1 1 -1
p3 3 1 1 -1

Initial burst weight? 0.5
Initial prediction? 2
Lottery results:
0:p1 2:p2 3:p3 4:p1 5:p3 6:p2 7:p3 8:p1 9:p1 11:p2 12:p3 13:p2 14:p3 15:p2 16:p3 17:END
0:IDLE 3:p2 1:IDLE 4:p3 2:IDLE 5:p1 5:IDLE 6:p3 6:IDLE 7:p2 7:IDLE 8:p3 8:IDLE 12:p2 9:IDLE 13:p3 10:IDLE 14:p2 11:IDLE 15:p3 12:IDLE 16:p2 13:IDLE 17:p3 14:END
Average Turnaround Time: 11.0
Average Response Time: 0.0

**Lottery out performs FCFS and RR:**

How many context switches? 10
Processes:
p1 0 6 3 3
p2 2 5 1 4

FCFS results:
    0:p1  6:p2  11:p2  15:p1  18:END
    0:IDLE  6:p1  3:IDLE  7:p2  4:END
    Average Turnaround Time: 15.5
    Average Response Time: 2.0

Timeslice? 2
RR results:
    0:p1  2:p2  4:p1  6:p2  8:p1  10:p2  11:p2  13:p1  15:p2  17:p1  18:END
    0:IDLE  6:p1  3:IDLE  7:p2  4:END
    Average Turnaround Time: 16.5
    Average Response Time: 0.0

Initial burst weight? 0.5
Initial prediction? 4
Lottery results:
    0:p1  2:p1  6:p2  9:p2  11:p1  12:p1  14:p2  18:END
    0:IDLE  6:p1  3:IDLE  11:p2  4:END
    Average Turnaround Time: 15.0
    Average Response Time: 2.0

**Drawbacks?**

When programming this I struggled some with the best way to assign tickets. It's not necessarily a cheap process. Furthermore, I had trouble coming up examples where lottery frequently out performs, many time it came in even.

**6. Multi-level feedback queues**

How many context switches? 15
Processes:
p1   1   7
p2   2   3
p3   3   2
SJF results:
    CPU: 1:p1  2:p2  5:p3  7:p1  13:END
    IO: 0:END
    Average Turnaround Time: 6.333333333333333
    Average Response Time: 0.6666666666666666
Timeslice? 2
Multi-level feedback queue results:
    CPU: 1:p1  3:p2  5:p3  7:p1  9:p2  10:p1  12:p1  13:END
    IO: 0:END
    Average Turnaround Time: 8.0
    Average Response Time: 1.0

Processes:
p1   0   3 3 3
p2   2   1 1 -1
p3   3   1 1 -1
SJF results:

CPU: 0:p1  3:p2  4:p3  5:p2  6:p3  7:p2  8:p3  9:p2  10:p3  11:p2  12:p3  13:p2  14:p3  15:p2  16:p3  17:END
IO: 0:IDLE  3:p1  3:IDLE  4:p2  4:IDLE  5:p3  5:IDLE  6:p2  6:IDLE  7:p3  7:IDLE  8:p2  8:IDLE  9:p3  9:IDLE  10:p2  10:IDLE  11:p3  11:IDLE  12:p2  12:IDLE  13:p3  13:IDLE  14:p2  14:IDLE  15:p3  15:IDLE  16:p2  16:IDLE  17:p3  17:END
No finished processes
Timeslice? 2
Multi-level feedback queue results:
CPU: 0:p1  2:p2  3:p3  4:p2  5:p3  6:p2  7:p1  8:p3  9:p2  10:p1  12:p3  13:p2  14:p1  15:p3  16:p2  17:END
IO: 0:IDLE  3:p2  1:IDLE  4:p3  2:IDLE  5:p2  3:IDLE  6:p3  4:IDLE  7:p2  5:p1  8:p3  9:p2  10:p3  11:p2  12:p3  13:p2  14:END
Average Turnaround Time: 15.0
Average Response Time: 0.0

Processes:
p1   0   6 3 3
p2   2   5 1 4
SJF results:
CPU: 0:p1  6:p2  11:p1  14:p2  18:END
IO: 0:IDLE  6:p1  3:IDLE  11:p2  4:END
Average Turnaround Time: 15.0
Average Response Time: 2.0
Timeslice? 2
Multi-level feedback queue results:
CPU: 0:p1  2:p2  4:p1  6:p2  8:p1  10:p2  11:p2  13:p1  15:p2  17:p1  18:END
IO: 0:IDLE  6:p1  3:IDLE  7:p2  4:END
Average Turnaround Time: 16.5
Average Response Time: 0.0

Processes:
p1   0   6 3 3
p2   2   5 1 4
SJF results:
CPU: 0:p1  6:p2  11:p1  14:p2  18:END
IO: 0:IDLE  6:p1  3:IDLE  11:p2  4:END
Average Turnaround Time: 15.0
Average Response Time: 2.0
Timeslice? 5
Multi-level feedback queue results:
CPU: 0:p1  5:p2  10:p1  11:p2  15:p1  18:END
IO: 0:IDLE  7:p2  1:IDLE  6:p1  4:END
Average Turnaround Time: 15.5
Average Response Time: 1.5

Well, for starters, with MLFQueue you don't have starvation. It does have a slower turn around time but it seems like you may be able to adjust your turn around times and possibly your reset duration (which I didn't mess with for these) to make it more comparable.

As for Response Time MLFQueue typically performed better, I think this can be adjusted by the time slice you chose for your round robins too though.