**Lottery Scheduler**

1. @seed1

Random 495435 -> Winning ticket 30 (of 109) -> Run 0

  Jobs:  (\* job:0 timeleft:1 tix:84 )   (  job:1 timeleft:7 tix:25 )

--> JOB 0 DONE at time 1

Random 449491 -> Winning ticket 16 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:7 tix:25 )

Random 651593 -> Winning ticket 18 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:6 tix:25 )

Random 788724 -> Winning ticket 24 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:5 tix:25 )

Random 93859 -> Winning ticket 9 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:4 tix:25 )

Random 28347 -> Winning ticket 22 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:3 tix:25 )

Random 835765 -> Winning ticket 15 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:2 tix:25 )

Random 432767 -> Winning ticket 17 (of 25) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:1 tix:25 )

@seed2  
Random 669731 -> Winning ticket 61 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:9 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 308136 -> Winning ticket 21 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:8 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 605944 -> Winning ticket 68 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:7 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 606802 -> Winning ticket 91 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:6 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 581204 -> Winning ticket 44 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:5 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 158383 -> Winning ticket 67 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:4 tix:94 )   (  job:1 timeleft:8 tix:73 )

Random 430670 -> Winning ticket 144 (of 167) -> Run 1

  Jobs:  (  job:0 timeleft:3 tix:94 )   (\* job:1 timeleft:8 tix:73 )

Random 393532 -> Winning ticket 80 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:3 tix:94 )   (  job:1 timeleft:7 tix:73 )

Random 723012 -> Winning ticket 69 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:2 tix:94 )   (  job:1 timeleft:7 tix:73 )

Random 994820 -> Winning ticket 1 (of 167) -> Run 0

  Jobs:  (\* job:0 timeleft:1 tix:94 )   (  job:1 timeleft:7 tix:73 )

--> JOB 0 DONE at time 10

Random 949396 -> Winning ticket 31 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:7 tix:73 )

Random 544177 -> Winning ticket 35 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:6 tix:73 )

Random 444854 -> Winning ticket 65 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:5 tix:73 )

Random 268241 -> Winning ticket 39 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:4 tix:73 )

Random 35924 -> Winning ticket 8 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:3 tix:73 )

Random 27444 -> Winning ticket 69 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:2 tix:73 )

Random 464894 -> Winning ticket 30 (of 73) -> Run 1

  Jobs:  (  job:0 timeleft:0 tix:--- )   (\* job:1 timeleft:1 tix:73 )

@seed 3  
Random 625720 -> Winning ticket 88 (of 114) -> Run 1

  Jobs:  (  job:0 timeleft:2 tix:54 )   (\* job:1 timeleft:3 tix:60 )

Random 65528 -> Winning ticket 92 (of 114) -> Run 1

  Jobs:  (  job:0 timeleft:2 tix:54 )   (\* job:1 timeleft:2 tix:60 )

Random 13168 -> Winning ticket 58 (of 114) -> Run 1

  Jobs:  (  job:0 timeleft:2 tix:54 )   (\* job:1 timeleft:1 tix:60 )

--> JOB 1 DONE at time 3

Random 837469 -> Winning ticket 37 (of 54) -> Run 0

  Jobs:  (\* job:0 timeleft:2 tix:54 )   (  job:1 timeleft:0 tix:--- )

Random 259354 -> Winning ticket 46 (of 54) -> Run 0

  Jobs:  (\* job:0 timeleft:1 tix:54 )   (  job:1 timeleft:0 tix:--- )

1. When the number of tickets is so unbalanced in each job, then no jobs get done. Job 0 will never run before job1 completes. It completely destroys the behavior of the lottery scheduler.
2. Job 0 gets done after 200 rotations. This is very unfair because Job 0 takes up the entire length of Job 1. If we run it with seed 3, I noticed that Job 1 will get done at time 200.
3. It does not matter how large the quantum size gets, if the tickets are not the same and the length of each job is not different, then regardless how how we change the quantum size, the lottery scheduler will still be unfair.
4. With a stride scheduler, the graph would look like an exponential curve that keep going upward.

Multi-Level Paging

1. You will need 2 registers to locate a two-level paging, and then you will need 4 registers to locate a three-level paging.
2. With seed 0, we will need 2 memory reference. With seed 1, we will need 4 memory references. With seed 2, we will need 6 memory references.