

# MLFQ - Multilevel **Feedback** Queue Scheduling (preemptive)

➔ Same as MLQ but change the priority of the process based on *observations*

<b>Rule 1</b>	If $\text{Priority}(A) > \text{Priority}(B)$ , A runs
<b>Rule 2</b>	If $\text{Priority}(A) = \text{Priority}(B)$ , A & B run in round-robin fashion using the time slice (quantum length) of the given queue
<b>Rule 3</b>	When a job enters the system, it is placed at the highest priority (the topmost queue)
<b>Rule 4</b>	Once a job uses up its time allotment at a given level (regardless of how many times it has given up the CPU), its priority is reduced (i.e., it moves down one queue)
<b>Rule 5</b>	After some time period $S$ , move all the jobs in the system to the topmost queue

✓ **Good** : Turing-award winner algorithm

# Acknowledgments

Some of the course materials and projects are from

- Ryan Huang - teaching CS 318 at *John Hopkins University*
- David Mazière - teaching CS 140 at *Stanford*