TCSS 422: OPERATING SYSTEMS

Scheduling: Introduction, Multi-level Feedback Queue (MLFQ)

Wes J. Lloyd
Institute of Technology
University of Washington - Tacoma

FEEDBACK FROM 1/11

- Can you go more in depth about Jain's Fairness Index? If Jain's index is .333, what does that mean?
- Consider the formula:

$$\mathcal{J}(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i^2}$$

- Now consider an example of 3 jobs.
- Job A and Job B receive 10% of the available CPU time.
- Job C receives 80% of the CPU time, for a total of 100%
- JA (.1) + JB (.1) + JC (.8) = 1.0 total time
- What is Jain's Index?
- How does this relate to .33? Is this example better or worse?

FEEDBACK - 2

- What is the difference between a maskable vs. non-maskable interrupt?
- Interrupts are produced by ***hardware***, and are different than system calls or exceptions (*software*)…

Maskable Interrupt	Non-maskable Interrupt
Can be masked (queued)	Cannot be masked (queued)
Processing can be delayed	Processing is immediate
Lower priority than non-maskable interrupts	Higher priority that maskable interrupts
Response time is high	Response time is low
Example: Peripherals (keyboard, mouse), I/O	Emergency: RAM failure, HDD failure, power failure

FEEDBACK - 3

- What does it mean by "job arrival"?
  - For scheduling questions we will often state job arrival times for the purpose of calculating scheduling metric results such as Average Response Time, Average Turnaround Time, etc.
- What is the setting "ONBOOT=no" for? Under: `/etc/sysconfig/network-scipts/ifcfg-enp0s3`
- Used to automatically start the network adapter on the virtual machine when the CentOS VM boots up
  - If disabled, network won't automatically initialize

OBJECTIVES

- Scheduling Methods: Round Robin (Ch. 7)
- Multi-level Feedback Queue (Ch. 8)

RR: ROUND ROBIN

- Run each job awhile, then switch to another distributing the CPU evenly (fairly)
- Scheduling Quantum is called a time slice
- Time
  a mu
  timer interrupt
  period.

RR is fair, but performs poorly on metrics such as turnaround time

Process	Burst Time
P1	12
P5	5

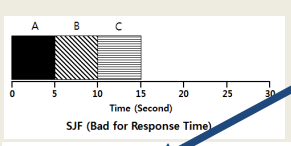
Round Robin scheduling algorithm Gantt chart

Scheduling Quantum = 5 seconds

P1	P2	P3	P4	P5	P1	P2	P4	P1
0	5	10	14	19	24	29	32	37	39

RR EXAMPLE

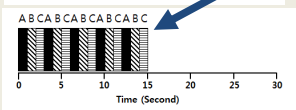
- ABC arrive at time=0, each run for 5 seconds



SJF (Bad for Response Time)

OVERHEAD not considered

$$T_{average\ response} = \frac{0+5+10}{3} = 5sec$$



RR with a time-slice of 1sec (Good for Response Time)

$$T_{average\ response} = \frac{0+1+2}{3} = 1sec$$

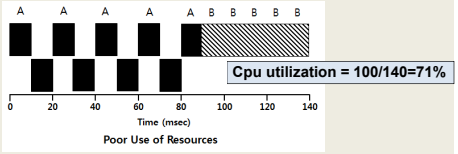
ROUND ROBIN: TRADEOFFS

Short Time Slice	Long Time Slice
Fast Response Time	Slow Response Time
High overhead from context switching	Low overhead from context switching

- Time slice impact:
  - Average turnaround time: ts(1,2,3,4,5)=14,14,13,14,10
  - Fairness: round robin is always fair, J=1

SCHEDULING WITH I/O

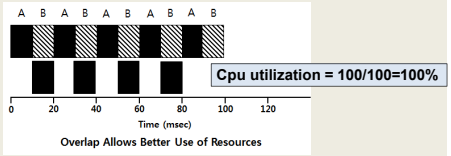
- STCF scheduler
  - A: CPU=50ms, I/O=40ms, 10ms intervals
  - B: CPU=50ms, I/O=0ms
  - Consider A as 10ms subjobs (CPU, then I/O)
- Without considering I/O:



Poor Use of Resources

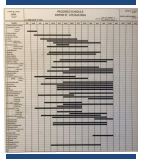
SCHEDULING WITH I/O - 2

- When a job initiates an I/O request
  - A is blocked, waits for I/O to compute, frees CPU
  - STCF scheduler assigns B to CPU
- When I/O completes → raise interrupt
  - Unblock A, STCF goes back to executing A: (10ms sub-job)



Overlap Allows Better Use of Resources

MULTI-LEVEL FEEDBACK QUEUE (MLFQ) SCHEDULER



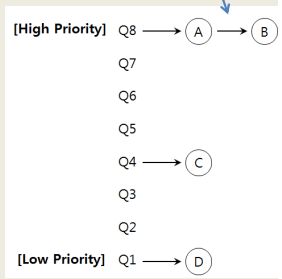
MULTI-LEVEL FEEDBACK QUEUE

- Objectives:
  - Improve turnaround time:
    Run shorter jobs first
  - Minimize response time:
    Important for interactive jobs (UI)
- Achieve without a priori knowledge of job length

MLFQ - 2

Round-Robin within a Queue

- Multiple job queues
- Adjust job priority based on observed behavior
- Interactive Jobs
  - Frequent I/O → keep priority high
  - Interactive jobs require fast response time (GUI/UI)
- Batch Jobs
  - Require long periods of CPU utilization
  - Keep priority low



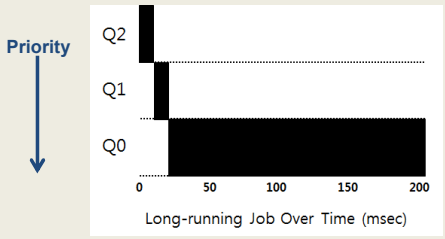
MLFQ: DETERMINING JOB PRIORITY

- New arriving jobs are placed into highest priority queue
- If a job uses its entire time slice, priority is reduced (↓)
  - Jobs appears CPU-bound ("batch" job), not interactive (GUI/UI)
- If a job relinquishes the CPU for I/O priority stays the same

MLFQ approximates SJF

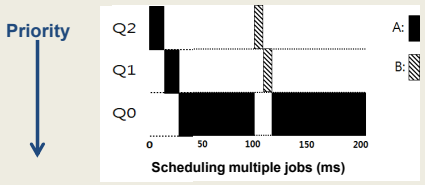
MLFQ: LONG RUNNING JOB

- Three-queue scheduler, time slice=10ms



MLFQ: BATCH AND INTERACTIVE JOBS

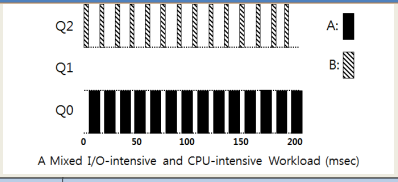
- $A_{arrival\_time}$ =0ms, A_{run_time}=200ms,
- B_{run_time} =20ms, $B_{arrival\_time}$ =100ms



MLFQ: BATCH AND INTERACTIVE - 2

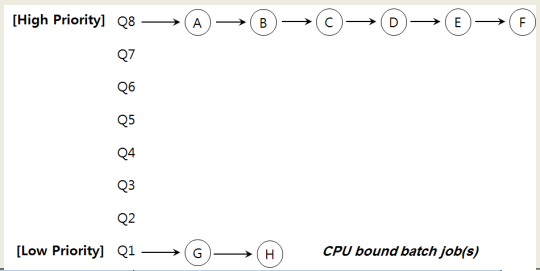
- Continuous interactive job (B) with long running batch job (A)
  - Low response time is good for B
  - A continues to make progress

The MLFQ approach keeps interactive job(s) at the highest priority



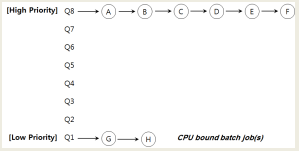
MLFQ: ISSUES

- Starvation



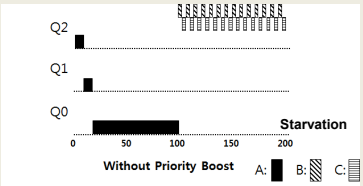
MLFQ: ISSUES - 2

- Gaming the scheduler
  - Issue I/O operation at 99% completion of the time slice
  - Keeps job priority fixed – never lowered
- Job behavioral change
  - CPU/batch process becomes an interactive process



Priority becomes stuck

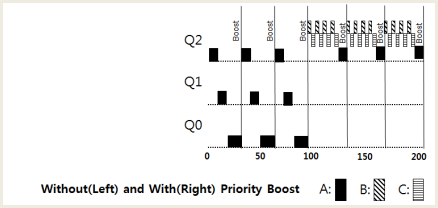
RESPONDING TO BEHAVIOR CHANGE



- Priority Boost
  - Reset all jobs to topmost queue after some time interval S

RESPONDING TO BEHAVIOR CHANGE - 2

- With priority boost
  - Prevents starvation



QUESTIONS