TCSS 422: OPERATING SYSTEMS

INTRODUCTION TO CPU SCHEDULING, MULTI-LEVEL FEEDBACK QUEUE (MLFQ)

Wes J. Lloyd
School of Engineering and Technology
University of Washington - Tacoma

1

OBJECTIVES – 4/14

- Questions from 4/9
- Active Reading Quiz – Ch. 7
- Assignment 0
- Chapter 7: Scheduling Introduction
  - Scheduling metrics
  - SJF, STCF, RR schedulers
- Chapter 8: Multi-level Feedback Queue
  - MLFQ Scheduler
  - Job Starvation
  - Gaming the Scheduler
  - Examples

2

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (56 respondents):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 7.875 (↑ *from 7.52*)**

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 5.93 (↑ from 5.82)**

3

FEEDBACK FROM 4/9

- ***So the OS performs a context switch from the user process to the kernel process when it needs to perform a system trap?***
- It depends on the kind of trap:
- **System calls (planned)**- results from kernel API call. Context switches to kernel worker process to perform requested work which requires privileged access to the hardware
- **Interrupts (events)/Exceptions (errors)-**
- *****Key difference between interrupt handling and context switch*****
- Code executed by interrupt / exception handler is not a process
- Code is a kernel control path that runs at the expense of the same process that was running when the interrupt occurred.
- Invoking interrupt handlers is lighter weight than a context switch
- Less context; requires less time to set up and tear down.
- More similar to a function call

4

FEEDBACK - 2

- ***A system trap is also when the OS needs to step in to ensure the process doesn't crash correct?***

- The trap type would likely be an "exception handler" trap.

5

FEEDBACK - 3

- ***Is user mode and kernel mode a strict dichotomy? Where does super user fit into this?***

- The CPU also includes mode 1 for running code directly on the CPU from virtual machines.
  - Mode helps Virtual Machines run faster when their code can run directly on the processor without software emulation
  - Original VMs ran all code in USER MODE as they were not trusted
  - CPU extensions for virtualization have helped by allowing untrusted instructions from VMs to be trapped and replaced
- Super user is the root user in Linux that has default permission to read/write/execute all files, configure user accounts, groups, install software and more
  - Super user is essentially a built-in administrator account

6

FEEDBACK - 4

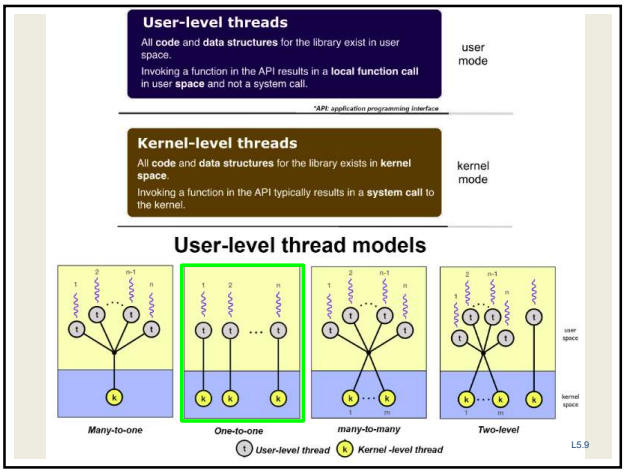
- ***I'm unsure about the difference between turnaround time versus execution time***
- **Turnaround time**: also known as the wall-clock or watch time
- This is the time that transpires from the users perspective from when an task is started until it completes
- This is the "real" value output from the Linux "time" command
- **Execution time**: also known as CPU time
- The time the task actually runs actively on the CPU.
- Add the "user" plus "sys" from the Linux "time" command
- Indicates time the CPU spent in user vs. kernel (sys) mode

7

FEEDBACK - 5

- ***We covered chapter 4, 5, and 6 last week.***
- ***I would like to read the book chapters too, but is it enough to just watch the lectures and read the powerpoints?***
- It is recommended to read ***all*** of the chapters we cover in class
- They are relatively short and go quickly
- This is especially important if the content is new to you
- Obtaining the information multiple times, in multiple ways may help with retention
- This question is a perfect segue to …

8



9

OBJECTIVES – 4/14

- **Questions from 4/9**
- **Active Reading Quiz – Ch. 7**
- **Assignment 0**
- **Chapter 7: Scheduling Introduction**
  - Scheduling metrics
  - SJF, STCF, RR schedulers
- **Chapter 8: Multi-level Feedback Queue**
  - MLFQ Scheduler
  - Job Starvation
  - Gaming the Scheduler
  - Examples

10

OBJECTIVES – 4/14

- **Questions from 4/9**
- **Active Reading Quiz – Ch. 7**
- **Assignment 0**
- **Chapter 7: Scheduling Introduction**
  - Scheduling metrics
  - SJF, STCF, RR schedulers
- **Chapter 8: Multi-level Feedback Queue**
  - MLFQ Scheduler
  - Job Starvation
  - Gaming the Scheduler
  - Examples

11

"QUIZ" 0 – C PROGRAMMING BACKGROUND SURVEY

- Original submission moved to "Ungraded Surveys"
- New Placeholder records points for Quiz 0 under: Assignments → Tutorials/Quizzes/In-class Activities
- Scores posted

12

CHAPTER 7- SCHEDULING: INTRODUCTION

13

OBJECTIVES – 4/14

- Questions from 4/9
- Active Reading Quiz – Ch. 7
- Assignment 0
- Chapter 7: Scheduling Introduction
  - Scheduling metrics
  - SJF, STCF, RR schedulers
- Chapter 8: Multi-level Feedback Queue
  - MLFQ Scheduler
  - Job Starvation
  - Gaming the Scheduler
  - Examples

14

SCHEDULING METRICS

- **Metrics**: A standard measure to quantify to what degree a system possesses some property. Metrics provide *repeatable* techniques to quantify and compare systems.
- **Measurements** are the numbers derived from the application of metrics

- Scheduling Metric #1: **Turnaround time**
- The time at which the job completes minus the time at which the job arrived in the system

$$T_{turnaround} = T_{completion} - T_{arrival}$$

- How is turnaround time different than execution time?

15

SCHEDULING METRICS - 2

- Scheduling Metric #2: **Fairness**
  - Jain's fairness index
  - Quantifies if jobs receive a fair share of system resources

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

- n processes
- x_i is time share of each process
- worst case = 1/n
- best case = 1

- Consider n=3, worst case = .333, best case=1
- With n=3 and x_1=.2, x_2=.7, x_3=.1, fairness=.62
- With n=3 and x_1=.33, x_2=.33, x_3=.33, fairness=1

16

SCHEDULING METRICS - 3

- Scheduling Metric #3: **Response Time**
- Time from when job arrives until it starts execution

$$T_{response} = T_{firstrun} - T_{arrival}$$

- STCF, SJF, FIFO
  - can perform poorly with respect to response time

What scheduling algorithm(s) can help minimize response time?

17

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
  time
  period.

Process	Burst Time
P1	12

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

Round Robin scheduling algorithm Gantt chart

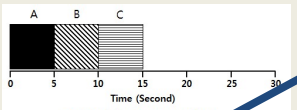
Scheduling Quantum = 5 seconds

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

18

RR EXAMPLE

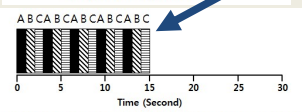
- ABC arrive at time=0, each run for 5 seconds



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)

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

19

ROUND ROBIN: TRADEOFFS

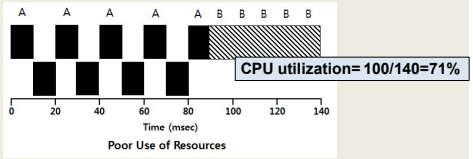
Short Time Slice	Long Time Slice
Fast Response Time	Slow Response Time
High overhead from context switching	Low overhead from context switching

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

20

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:

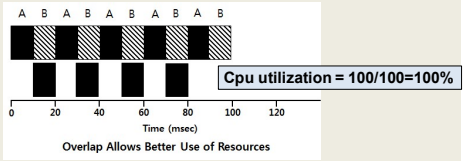


Poor Use of Resources

21

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

22

Which scheduler, thus far, best address fairness and average response time of jobs?

Respond at **PollEv.com/wesleylloyd641**

Text **WESLEYLLOYD641** to **22333** once to join, then **1, 2, 3, 4, 5...**

First In - First Out (FIFO)	1
Shortest Job First (SJF)	2
Shortest Time to Completion First (STCF)	3
Round Robin	4
None of the Above	5
All of the Above	6

Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

Total Results

23

QUESTION: SCHEDULING FAIRNESS

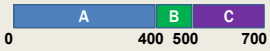
- Which scheduler, this far, best addresses fairness and average response time of jobs?
- First In – First Out (FIFO)
- Shortest Job First (SJF)
- Shortest Time to Completion First (STCF)
- Round Robin (RR)
- None of the Above
- All of the Above

24

SCHEDULING METRICS

- Consider Three jobs (A, B, C) that require: $time_A$=400ms, $time_B$=100ms, and $time_C$=200ms
- All jobs arrive at time=0 in the sequence of A B C.
- Draw a scheduling graph to help compute the **average response time (ART)** and **average turnaround time (ATT)** scheduling metrics for the FIFO scheduler.

Example:



25

When poll is active, respond at **PollEv.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

What is the Average Response Time of the FIFO scheduler?

Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

26

When poll is active, respond at **PollEv.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

What is the Average Turnaround Time of the FIFO scheduler?

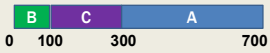
Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

27

SCHEDULING METRICS

- Consider Three jobs (A, B, C) that require: $time_A$=400ms, $time_B$=100ms, and $time_C$=200ms
- All jobs arrive at time=0 in the sequence of A B C.
- Draw a scheduling graph to help compute the **average response time (ART)** and **average turnaround time (ATT)** scheduling metrics for the SJF scheduler.

Example:



28

When poll is active, respond at **PollEv.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

What is the Average Response Time of the Shortest Job First Scheduler?

Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

29

When poll is active, respond at **PollEv.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

What is the Average Turnaround Time of the Shortest Job First Scheduler?

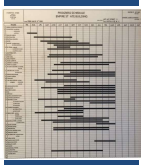
Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

30

TCSS 422 WILL RETURN AT ~2:45PM

31

CHAPTER 8 – MULTI-LEVEL FEEDBACK QUEUE (MLFQ) SCHEDULER



32

OBJECTIVES – 4/14

- Questions from 4/9
- Active Reading Quiz – Ch. 7
- Assignment 0
- Chapter 7: Scheduling Introduction
  - Scheduling metrics
  - SJF, STCF, RR schedulers
- Chapter 8: Multi-level Feedback Queue
  - MLFQ Scheduler
  - Job Starvation
  - Gaming the Scheduler
  - Examples

33

MULTI-LEVEL FEEDBACK QUEUE

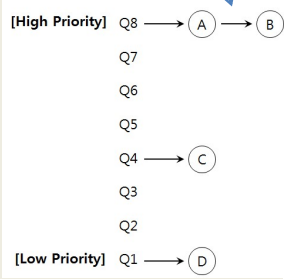
- Objectives:
  - Improve turnaround time:
    Run shorter jobs first
  - Minimize response time:
    Important for interactive jobs (UI)
- Achieve without a priori knowledge of job length

34

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



35

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

36

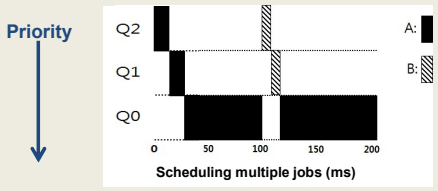
MLFQ: LONG RUNNING JOB

- Three-queue scheduler, time slice=10ms

37

MLFQ: BATCH AND INTERACTIVE JOBS

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

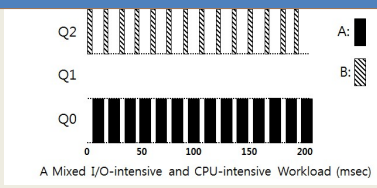


38

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

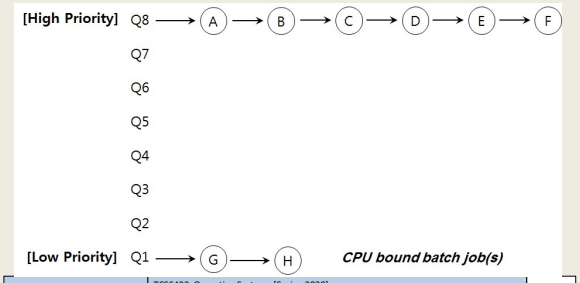
The MLFQ approach keeps interactive job(s) at the highest priority



39

MLFQ: ISSUES

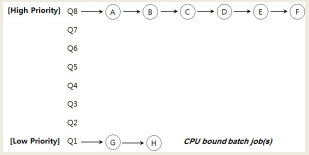
- Starvation



40

MLFQ: ISSUES - 2

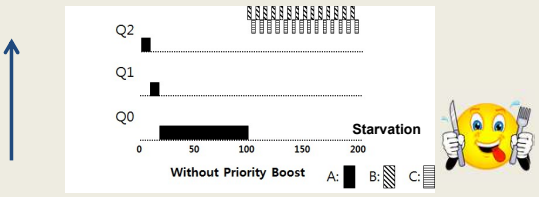
- Gaming the scheduler
  - Issue I/O operation at 99% completion of the time slice
  - Keeps job priority fixed – never lowered
- Job behavioral change
  - CPU/batch process becomes an interactive process



Priority becomes stuck

41

RESPONDING TO BEHAVIOR CHANGE

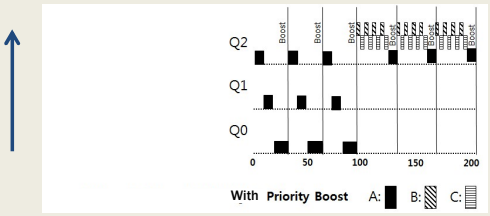


- Priority Boost
  - Reset all jobs to topmost queue after some time interval S

42

RESPONDING TO BEHAVIOR CHANGE - 2

- With priority boost
  - Prevents starvation



43

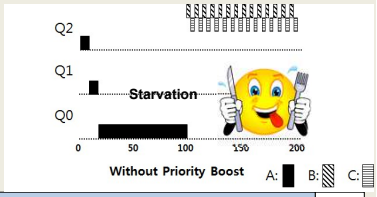
KEY TO UNDERSTANDING MLFQ – PB

- Without priority boost:
- **Rule 1:** If Priority(A) > Priority(B), A runs (B doesn't).
- **Rule 2:** If Priority(A) = Priority(B), A & B run in RR.
- **KEY:** If time quantum of a higher queue is filled, then we don't run any jobs in lower priority queues!!!

44

STARVATION EXAMPLE

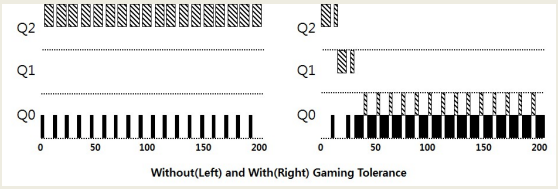
- **Consider 3 queues:**
- Q2 – HIGH PRIORITY – Time Quantum 10ms
- Q1 – MEDIUM PRIORITY – Time Quantum 20 ms
- Q0 – LOW PRIORITY – Time Quantum 40 ms
- Job A: 200ms no I/O
- Job B: 5ms then I/O
- Job C: 5ms then I/O
- Q2 fills up, starves Q1 & Q0
- A makes no progress



45

PREVENTING GAMING

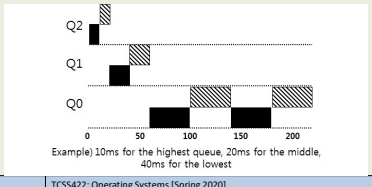
- Improved time accounting:
  - Track total job execution time in the queue
  - Each job receives a fixed time allotment
  - When allotment is exhausted, job priority is lowered



46

MLFQ: TUNING

- Consider the tradeoffs:
  - How many queues?
  - What is a good time slice?
  - How often should we "Boost" priority of jobs?
  - What about different time slices to different queues?



47

PRACTICAL EXAMPLE

- Oracle Solaris MLFQ implementation
  - 60 Queues → w/ slowly increasing time slice (high to low priority)
  - Provides sys admins with set of editable table(s)
  - Supports adjusting time slices, boost intervals, priority changes, etc.
- Advice
  - Provide OS with hints about the process
  - Nice command → Linux

48

MLFQ RULE SUMMARY

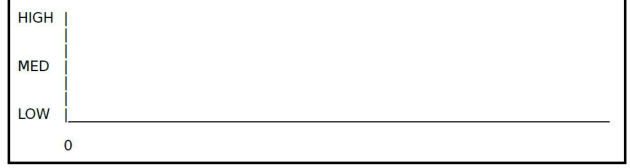
- The refined set of MLFQ rules:
- **Rule 1:** If Priority(A) > Priority(B), A runs (B doesn't).
- **Rule 2:** If Priority(A) = Priority(B), A & B run in RR.
- **Rule 3:** When a job enters the system, it is placed at the highest priority.
- **Rule 4:** Once a job uses up its time allotment at a given level (regardless of how many times it has given up the CPU), its priority is reduced(i.e., it moves down on queue).
- **Rule 5:** After some time period S, move all the jobs in the system to the topmost queue.

49

Jackson deploys a 3-level MLFQ scheduler. The time slice is 1 for high priority jobs, 2 for medium priority, and 4 for low priority. This MLFQ scheduler performs a Priority Boost every 6 timer units. When the priority boost fires, the current job is preempted, and the next scheduled job is run in round-robin order.

Job	Arrival Time	Job Length
A	T=0	4
B	T=0	16
C	T=0	8

(11 points) Show a scheduling graph for the MLFQ scheduler for the jobs above.
Draw vertical lines for key events and be sure to label the X-axis times as in the example.
Please draw clearly. An unreadable graph will loose points.



50

EXAMPLE

- Question:
- Given a system with a quantum length of 10 ms in its highest queue, how often would you have to boost jobs back to the highest priority level to guarantee that a single long-running (and potentially starving) job gets at least 5% of the CPU?
- Some combination of n short jobs runs for a total of 10 ms per cycle without relinquishing the CPU
  - E.g. 2 jobs = 5 ms ea; 3 jobs = 3.33 ms ea, 10 jobs = 1 ms ea
  - n jobs always uses full time quantum (10 ms)
  - Batch jobs starts, runs for full quantum of 10ms
  - All other jobs run and context switch totaling the quantum per cycle
  - If 10ms is 5% of the CPU, when must the priority boost be ???
  - **ANSWER** → ***Priority boost should occur every 200ms***

51

QUESTIONS

52

WILL RETURN IN A FEW MINUTES

53