

CS 537 Lecture 10-A Scheduling

Michael Swift

10/23/07

© 2004-2007 Ed Lazowska, Hank Levy, Andrea and
Remzi Arpaci-Dusseau, Michael Swift

1

Solaris Schedulers

- Scheduling classes
 - Time sharing – dynamically alters priorities and timeslices – higher priority indicates lower timeslice, more responsive but less time to respond.
 - Fixed priority – priorities don't change, good for real time. Is preemptive
 - Fair share – doesn't assign priority, but shares CPU equally among processes at this level
- Preemption: will preempt lower priority thread when higher becomes able to run
- Table driven MLFQ. Priority 0 is lowest, priority 59 is highest
 - If quantum expires, priority is lowered
 - If wake up from sleep / IO, priority gets a boost
 - If waits too long without executing, gets priority boost

10/23/07

© 2004-2007 Ed Lazowska, Hank Levy, Andrea and
Remzi Arpaci-Dusseau, Michael Swift

2

Table Example

Priority	Quantum (ms)	Quantum expired Prio	Return from sleep prio	MaxWait	Wait Level
0	200	0	50	10000	5
5	200	0	50	...	10
10	160	0	51		15
15	160	5	51		20
20	120	10	52		25
25	120	15	52		30
30	80	20	53		35
35	80	25	54		40
40	40	30	55		45
45	40	35	56		50
50	40	40	58		55
55	40	45	58		59
59	20	49	59		59

10/23/07

© 2004-2007 Ed Lazowska, Hank Levy, Andrea and
Remzi Arpaci-Dusseau, Michael Swift

3

Priority Inversion

- Problem: thread 2 doesn't get to run because thread 1 holds lock, but thread 3 has higher priority and prevents thread 1 from running.
- Solution: thread 1 gets thread 2's priority when thread 2 block on the lock it holds

Thread 1: priority 10	Thread 2: priority 50	Thread 3: priority 20
acquire(lock)		
do some work	acquire(lock)	wake up
preempted	wait	execute
	BLOCKED!	execute

10/23/07

© 2004-2007 Ed Lazowska, Hank Levy, Andrea and
Remzi Arpaci-Dusseau, Michael Swift

4