CS 537
Lecture 6
Scheduling

Michael Swift

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

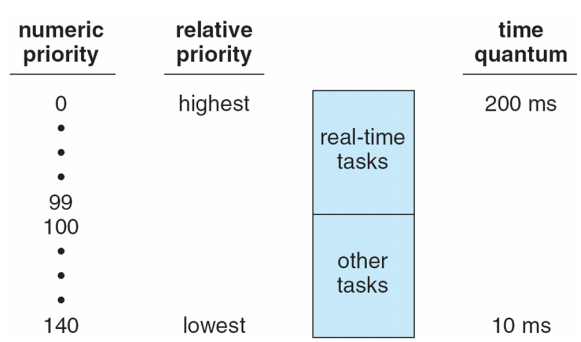
Table Example

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

Linux Scheduling

- Constant order *O*(1) scheduling time
- Two priority ranges: time-sharing and real-time
- **Real-time** range from 0 to 99 and **nice** value from 100 to 140
- Two arrays of runqueues
  - One for ready, not-yet-run tasks
  - One for ready, already-ran-and-expired tasks
- Run all highest priority tasks from active queue
  - move to expired on timeout
- Then swap active/expired for that priority
- Repeat
- **Dynamic Priority** adds +/- 5 based on how much a process sleeps; more sleeping = higher priority

Priorities and Time-slice length



List of Tasks Indexed According to Priorities