

CSc 360

Operating Systems

More on Scheduling

Jianping Pan

Summer 2015

P2 deliverable 1 (draft design)

- Due today 5pm through connex
 - connex remains open until midnight
 - do submit one copy before 5pm
 - the last submitted one be evaluated
 - answer questions listed on p2 spec
 - how to use pthread, mutex, condition variable?
 - how to model train, main track, signal to go?
 - include the final design in deliverable 2
 - grading [5]: (draft + final) / 2

6/4/15

CSc 360

2

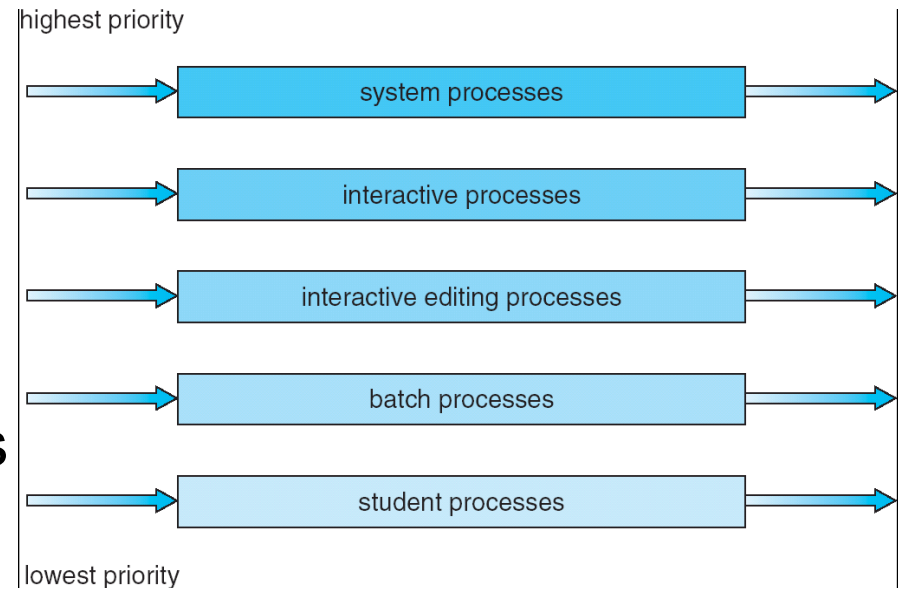
* tomorrow, tutorial instructor will go through p2 spec and your design

Review: scheduling

- Scheduling algorithms
 - FCFS
 - “convoy effect”
 - SJF
 - optimal in terms of ...
 - “starvation”
 - priority
 - RR
 - preemptive vs non-preemptive

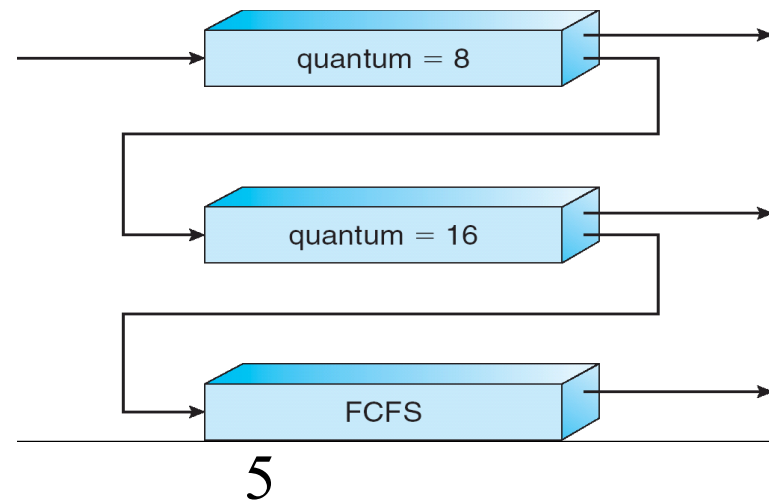
Multi-queue scheduling

- “No one fits all”
- Multi-queue approach
 - foreground queue
 - e.g., RR; better fairness
 - background queue
 - e.g., FCFS; more efficient
- Inter-queue scheduling
 - priority, time sharing (e.g., “80% rule”)



Multi-queue with feedback

- Multi-queue
 - number of queues
 - queuing algorithm for each queue
- Multi-queue with feedback
 - promote jobs
 - demote jobs
 - example

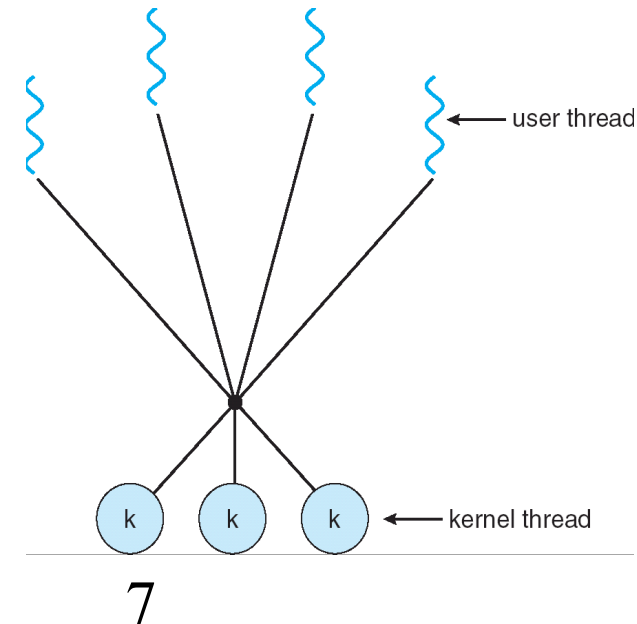


Multi-processor queuing

- Load balance between processors
 - cooperation and communication
- Asymmetric multiprocessing
 - one master scheduler
- Symmetric multiprocessing
 - cooperative schedulers
 - processor affinity: try to stick with one
 - load balancing: push or poll migration

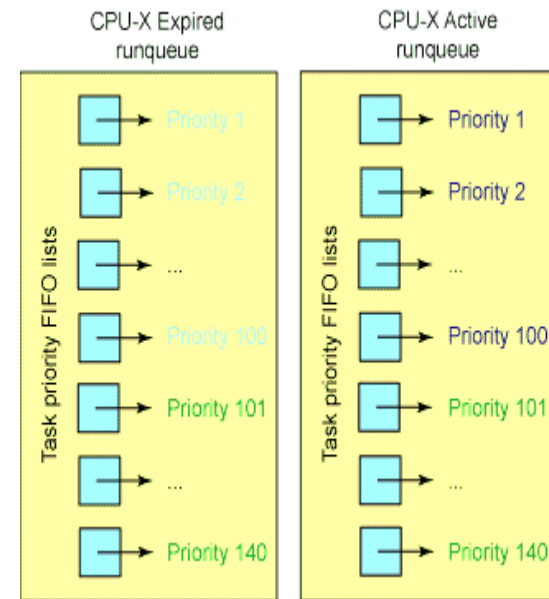
More on scheduling

- Thread scheduling
 - local: user \rightarrow kernel thread
 - e.g., within a process
 - global: kernel thread \rightarrow CPU
 - e.g., across the system
- Algorithm evaluation
 - queuing analysis
 - Little's law: $n = \lambda * W$



Linux scheduling

- Scheduler
 - $O(n)$: global run queue (Linux 2.4)
 - n : number of processes
 - $O(1)$: double linked list (Linux 2.6)
 - active/expired set
- 140 priority levels
 - realtime (0-99), user (100-139, nice/quantum sizes)
- Interactive vs CPU-bound: interactivity



6/4/15

CSc 360

8

* multi-core/CPU scheduling

Q: completely fair scheduler?

Pthread scheduling

- `pthread_attr_setschedpolicy ()`;
 - regular, non-realtime (nice-able)
 - realtime, round-robin (preemptive, privileged)
 - realtime, FCFS (non-preemptive, privileged)
- `pthread_attr_setschedparam ()`;
- `pthread_attr_setscope ()`;
 - scheduling within a process
 - scheduling for the entire system

This lecture

- More on scheduling
 - multi-queue scheduling
 - multi-processor scheduling
 - scheduling evaluation
 - Little's law, simulation
- Explore further
 - list process priority: `/usr/bin/top`
 - change priority: `/bin/nice`

Next lecture

- Process synchronization
 - read OSC7 Chapter 6 (or OSC6 Chapter 7)