Scheduling

Lecture 10
Scheduling Criteria

• Fairness - processes should receive an equal share of the processor if at all possible. (Modulo priority assignments)

• Efficiency - percentage of time CPU is busy doing useful work.

• Response time - performance for interactive tasks.

• Waiting time - how long a process waits in the "ready queue" to be allowed to run (note that this is different from waiting for I/O to complete - the process is not "ready" in that case)

• Turnaround time - submission to completion time for batch jobs.

• Throughput - number of jobs completed per hour. (Also a batch measure.)
Algorithms

- Round robin
- Priority
- Shortest Job First
- Deadline
- Multilevel Queue
Multilevel Feedback

• Move processes between queues if behavior changes

• What kind of processes should have highest priority?

• What if they never get to run?
Swapping

• Swapping manages the "degree of multiprogramming," i.e. the number of processes that are currently active in the system.

• If some processes are swapped out, a second level of scheduling is necessary. (Medium-term scheduling.)
Multiprocessor Scheduling

Diagram showing a system with multiple CPUs and local memories, connected to a system memory.
Multiprocessor Scheduling

- Non-Uniform Memory Access (NUMA) – access to local memory is faster than access to other memory

- Processor affinity – processes can be assigned to a processor or set of processors
Multiprocessor Scheduling

- Assigning processes to processors
  - static vs dynamic
  - centralized vs distributed
- Multiprogramming on individual processors
  - depends on how many processors
- Dispatching processes
  - Do we need complexity of uniprocessor scheduling algorithms?
Efficiency

• Where is the ready queue?
• Is there just one?
  – If yes, asymmetric multiprocessing
    • May not take best advantage of affinity
  – If no, symmetric multiprocessing
• If multiple queues, what if one queue empties while others have waiting jobs?
Load Balancing

• Check periodically for imbalance
  – Recent utilization
  – Queue length
• Move process to idle processor
  – push vs. pull migration
• Balancing vs. affinity
  – Gang scheduling