Lecture Notes 6

CPU Scheduling

- I/O Blocking Cycle
  - CPU Burst – Burst of processing on the CPU
  - I/O Burst – Burst of I/O requests that are interleaved with CPU Bursts
- CPU Scheduler (Short-Term Scheduler)
- Scheduling
  - Types of Scheduling
    - Preemptive – Scheduling can occur at any time
    - Non-preemptive (Cooperative) – Scheduling occurs at specific call points
  - State Changes
    - Running to Wait (I/O Request, or wait) (Both)
    - Running to Ready (Interrupt or signal) (Preemptive Only)
    - Waiting to Ready (I/O Completion typically via Interrupt) (Preemptive Only)
    - Process/Task/Thread Terminates (Explicit call) (Both)
- Dispatcher – Actually gives control to scheduled process
  - Switching Context – Changing the stack, CPU registers and pointer to page table
  - Switching to User Mode – Changing the CPU protection mode to user
  - Jumping to Proper Location – Program counter is set to previous saved instruction
  - Dispatch Latency – Time to stop a process and switch to another
- Scheduling Criteria
  - CPU Utilization – How busy is the CPU?
  - Throughput – How much work is completed per unit time?
  - Turnaround Time – How much time did the process take to complete?
  - Waiting Time – How much time does the process spending waiting?
  - Response Time – How much time does it take for a process to respond when ready?
- Scheduling Algorithms
  - First-Come, First-Served (FCFS) – Handled with FIFO, first process starts first and executes to completion (causes Convoy Effect, many small get blocked behind)
  - Shortest Job First (SJF) – The shortest expect CPU burst runs next (Exponential Average used to predict: $\tau_{n+1}=\alpha t_n+(1-\alpha)\tau_n$)
  - Shortest Remaining Time First – Preemptive SJF
  - Priority Scheduling
    - Indefinite Blocking (Starvation) – Low priority tasks can be starved if high priority doesn't block
    - Aging – Gradual increase in priority the longer a process waits
  - Round Robin (RR) – Like FCFS with preemption
    - Processor Sharing – Share the processor each gets a time quantum
    - Time Quantum (Time Slice) – A period of time a task gets the processor, typically 10 – 100ms

- Multilevel Queue Scheduling – Multiple queues are used for scheduling
  - Foreground (Interactive) – Want low response time
  - Background (Batch) – Might want higher throughput or other metric
- Multilevel Feedback-Queue Scheduling – Allows tasks to move between queues
- Thread Scheduling – May be User space (LWT) or Kernel space (HWT)
  - Process-Contention Scope (PCS) – LWT are competing for time to run
  - System-Contention Scope (SCS) – HWT are competing for CPU time
- Multiple-Processor Scheduling
  - Load Sharing – All CPUs share the load of running tasks
  - Asymmetric Multiprocessing – Specific CPU(s) handle system (or special) tasks
  - Symmetric Multiprocessing (SMP) – All CPUs do self scheduling
  - Processor Affinity (Caching Solution) – Task is likely to keep running on same CPU
    - Soft Affinity (Attempt to Keep Processor)
    - Hard Affinity (Process Specifies No Processor Switching)
  - Load Balancing – Attempt to keep workload balanced on all CPUs
    - Push Migration – Task checks and pushes tasks to balance load
    - Pull Migration – Idle CPU pulls waiting task from another CPU
  - Multicore Processors – Multiple processing cores on single chip
    - Memory Stall – Task stalls waiting for information from memory
    - Coarse-Grained Multithreading – Thread runs until memory stall
    - Fine-Grained Multithreading – Thread instructions are interleaved
  - Symmetric Multithreading (Hyperthreading)
- Real-Time CPU Scheduling
  - Soft Real-Time – No guarantee of when critical real-time task is scheduled
  - Hard Real-Time – Strict requirements of scheduling and deadlines
  - Event Latency – Time between event and when it is serviced
  - Interrupt Latency – Time from interrupt to begin of ISR
  - Dispatch Latency – Time to stop a process and switch to another
  - Admission-Control – Admits new tasks or not into the system
  - Rate Monotonic Scheduling – Periodic tasks have static priority
  - Earliest-Deadline-First (EDF) – Earliest deadline will be scheduled first
  - Proportional Share – T shares are split up amongst the N tasks
- Algorithm Evaluation
  - Deterministic Modeling – Analysis of predetermined workload
    - Analytic Evaluation
  - Queuing Models
    - Network Analysis – Use of queuing theory from network to analyze performance on arrival/wait times etc.
    - Little's Formula ($n = \lambda W$)
  - Simulations – Use of simulation on traces to determine the performance