Lecture 5: Scheduling Policies

Mythili Vutukuru
IIT Bombay
What is a scheduling policy?

• On context switch, which process to run next, from set of ready processes?

• OS scheduler schedules the CPU requests (bursts) of processes
  – CPU burst = the CPU time used by a process in a continuous stretch
  – If a process comes back after I/O wait, it counts as a fresh CPU burst
What are we trying to optimize?

- Maximize (utilization = fraction of time CPU is used)
- Minimize average (turnaround time = time from process arrival to completion)
- Minimize average (response time = time from process arrival to first scheduling)
- Fairness: all processes must be treated equally
- Minimize overhead: run process long enough to amortize cost of context switch (~1 microsecond)
First-In-First-Out (FIFO)

• Example: three processes arrive at $t=0$ in the order A,B,C

• Problem: convoy effect

• Turnaround times tend to be high

Figure 7.1: FIFO Simple Example

Figure 7.2: Why FIFO Is Not That Great
Shortest Job First (SJF)

- Provably optimal when all processes arrive together.

- SJF is non-preemptive, so short jobs can still get stuck behind long ones.

Figure 7.3: SJF Simple Example

Figure 7.4: SJF With Late Arrivals From B and C
Shortest Time-to-Completion First (STCF)

- Also called Shortest Remaining Time First (SRTF)
- Preemptive scheduler
- Preempts running task if time left is more than that of new arrival
Round Robin (RR)

- Every process executes for a fixed quantum slice
- Slice big enough to amortize cost of context switch
- Preemptive
- Good for response time and fairness
- Bad for turnaround time

Figure 7.6: SJF Again (Bad for Response Time)

Figure 7.7: Round Robin (Good for Response Time)
Schedulers in real systems

• Real schedulers are more complex
• For example, Linux uses a Multi Level Feedback Queue (MLFQ)
  – Many queues, in order of priority
  – Process from highest priority queue scheduled first
  – Within same priority, any algorithm like RR
  – Priority of process decays with its age