CS323 Operating Systems
CPU Scheduling

Yuanyuan Zhou
Lecture 5
1/31/2003

Content of this lecture

- Administrative announcements
- Why Scheduling?
- Scheduling Levels
- Basic Scheduling Algorithm (FCFS)
- Summary

Administrative

- Test quiz due today
- Quiz1 will start next Monday, due Friday 5pm
- MP1(thread scheduling) starts now

Process Review

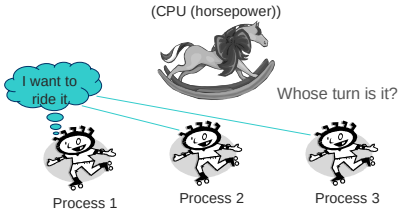
- So What Is A Process?
  - It's one executing instance of a "program"
  - It's separate from other instances
  - It can start ("launch") other processes
  - It can be launched by them
- What's in a process?
  - Code (text), data, stack, heap
  - Process control block
    - Process state, priority, accounting
    - Program counter, register variables, stack pointers, etc
    - Open files and devices

Threads: Lightweight Processes

- A thread is a single execution
  - Stack
  - Program counters
  - Registers
- All threads in a process share resources
  - Address space
  - Text, data, heap
  - Open files
- Implementations of threads
  - User-level, kernel-level, Hybrid, Service activation, pop-up

Scheduling

- Deciding which process/thread should occupy the resource (CPU, disk, etc)

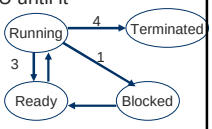


When to schedule?

- A new process starts
- The running process exits
- The running process is blocked
- I/O interrupt (some processes will be ready)
- Clock interrupt (every 10 milliseconds)

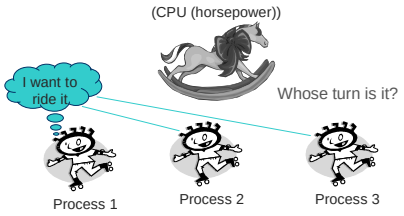
Preemptive vs. Non-preemptive

- **Non-preemptive scheduling:**
  - The running process keeps the CPU until it voluntarily gives up the CPU
    - process exits
    - switches to blocked state
    - 1and 4 only (no 3)
- **Preemptive scheduling**:
  - The running process can be interrupted and must release the CPU (can be forced to give up CPU)



What are the scheduling objectives?

- Group discussion (2 minutes)



Scheduling Objectives

- Fair (nobody cries)
- Priority (lady first)
- Efficiency (make best use of equipment)
- Encourage good behavior (good boy/girl)
- Support heavy loads (degrade gracefully)
- Adapt to different environments (interactive, real-time, multi-media)

Performance Criteria

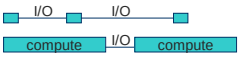
- Fairness
- Efficiency: keep resources as busy as possible
- Throughput: # of processes that completes in unit time
- Turnaround Time (also called elapse time)
  - amount of time to execute a particular process from the time its entered
- Waiting Time
  - amount of time process has been waiting in ready queue
- Response Time
  - amount of time from when a request was first submitted until first response is produced.
  - predictability and variance
- Policy Enforcement:
  - seeing that stated policy is carried out
- Proportionality:
  - meet users' expectation
- Meeting Deadlines: avoid losing data

Different Systems, Different Focuses

- For all
  - Fairness, policy enforcement, resource balance
- Batch Systems
  - Max throughput, min turnaround time, max CPU utilization
- Interactive Systems
  - Min Response time, best proportionality
- Real-Time Systems
  - predictability, meeting deadlines

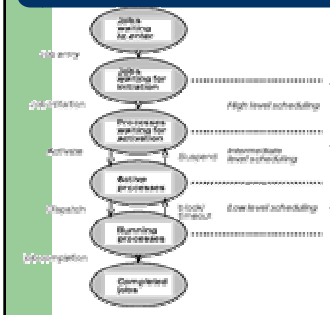
Program Behaviors Considered in Scheduling

- Is it I/O bound? Example?
- Is it CPU bound? Example?



- Batch or interactive environment
- Urgency
- Priority
- Frequency of page faults
- Frequency of preemption
- How much execution time it has already received
- How much execution time it needs to complete

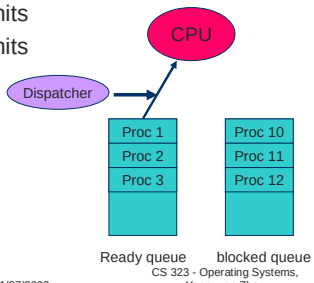
Scheduling Level



- Which jobs should be allowed to compete for the CPU and other resources
- Which jobs to temporarily suspend or resume to smooth fluctuations in system load.
- Assigning a cpu to a ready process.

CPU Scheduler

- Proc 1: 14 time units
- Proc2: 8 time units
- Proc3: 8 time units



- Dispatcher
- Preemptive vs. non-preemptive

Dispatcher

- Gives the control of the CPU to the process, scheduled by the short-term scheduler.
- Functions:
  - switching context
  - switching to user mode
  - jumping to the proper location in the user program.
- ***Dispatch Latency*** :time to stop process and start another one.
  - Pure overhead
  - Needs to be fast

Single Processor Scheduling Algorithms

- Batch systems
  - First Come First Serve (FCFS)
  - Short Job First
- Interactive Systems
  - Round Robin
  - Priority Scheduling
  - Multi Queue & Multi-level Feedback
  - Shortest process time
  - Guaranteed Scheduling
  - Lottery Scheduling
  - Fair Sharing Scheduling

First Come First Serve (FCFS)

- Process that requests the CPU FIRST is allocated the CPU FIRST.
- Also called FIFO
- Non-preemptive
- Used in Batch Systems
- Real life analogy: Fast food restaurant
- Implementation: FIFO queues
  - A new process enters the tail of the queue
  - The schedule selects from the head of the queue.
- Performance Metric: **Average Waiting Time**.
- Given Parameters:
  - Burst Time (in ms), Arrival Time and Order

FCFS Example

Process	Duration	Order	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

The final schedule:

P1 waiting time: 0
P2 waiting time: 24
P3 waiting time: 27

The average waiting time:
(0+24+27)/3 = 17

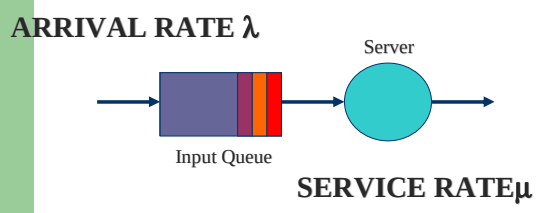
Problems with FCFS

- Non-preemptive
- Not optimal AWT
- Cannot utilize resources in parallel:
  - Assume 1 process CPU bounded and many I/O bounded processes
  - result: Convoy effect, low CPU and I/O Device utilization
  - Why?

Why Convoy Effects?

- Consider n-1 jobs in system that are I/O bound and 1 job that is CPU bound.
- I/O bound jobs pass quickly through the ready queue and suspend themselves waiting for I/O.
- CPU bound job arrives at head of queue and executes until complete.
- I/O bound jobs rejoin ready queue and wait for CPU bound job to complete.
- I/O devices idle until CPU bound job completes.
- When CPU bound job complete, other processes rush to wait on I/O again.
- CPU becomes idle.

Queuing Theory



Queueing Theory

- Steady state
- Poisson arrival with λ constant arrival rate (customers per unit time) each arrival is independent.
- $P(\tau \le t) = 1 - e^{-\lambda t}$

Analysis of Queueing Behavior

- Probability n customers arrive in time interval t is:
  $e^{-\lambda t} (\lambda t)^{n} / n!$
- Assume random service times (also poisson): μ constant service rate (customers per unit time)

Useful Facts from Queuing Theory

Little's Theorem:

- W_q = mean time a customer spends in the queue
- λ = arrival rate
- L_q = number of customers in queue
- W = mean time a customer spends in the system
- L = number of customers in the system
- $L_q = \lambda W_q$
- $L = \lambda W$

Example

- Example
  - Arrival 2 jobs/sec
  - Service 3 jobs/sec
  - Utilization 66.66%
  - Time in system 1 sec
  - Time in queue .6666 sec
  - Length of queue 1.3333

Analysis of FCFS

- Server utilization: $\rho = \frac{\lambda}{\mu}$
- Time in System: $W = \frac{1}{\mu - \lambda}$
- Time in Queue: $W_q = \frac{\rho}{\mu - \lambda}$
- Number in Queue (Little): $L_q = \frac{\rho^2}{1-\rho}$
- Example
  - Arrival 2 jobs/sec.
  - Service 3 jobs/sec.
  - Utilization 66.66%
  - Time in system 1 sec.
  - Time in queue .6666 sec.
  - Length of queue 1.3333.
- Time spent in system depends on size of other jobs.
- Variance large.

Summary

- What is scheduling
- Scheduling objectives
- CPU Scheduling
- FCFS
- Queuing theory
- Next lecture: other scheduling algorithms
  - Short Job First
  - Round Robin
  - Priority
  - Multi-Queue
  - Multi-level Feedback

Reminder

- Quiz next week
- MP1 starts next week