COMP 190-088: Systems Performance Analysis

System Modeling

CPU Scheduling

Jasleen Kaur

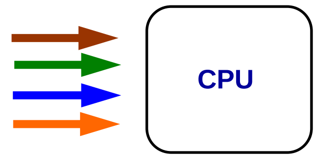
Department of Computer Science
The University of North Carolina at Chapel Hill

Spring 2005

System Modeling Overview

- Queuing Basics
- Single-server Analysis
- Multiple-server Analysis
- Network of Queues
- Scheduling Case Studies
  - **Processor Scheduling**
  - Disk Scheduling
  - Memory Management

Scheduling in Multiprogramming OSes



- Contention for resource-access in multi-user systems
  - Some users may not get proper share of resource
  - Long batch jobs may slow down small interactive jobs
- Sophisticated scheduling algorithms may be used in multi-class systems
  - Decide the order in which requests from the wait queue are served

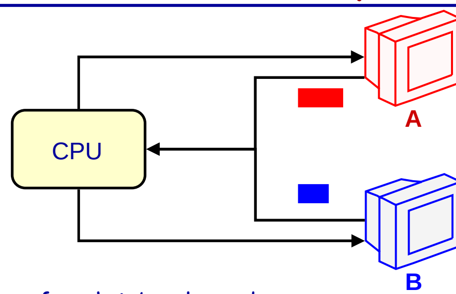
Which scheduling algorithm to use?

Metrics of Interest

- Throughput
  - Total system throughput
  - Per-class throughput
- Response time
  - Averaged for all jobs
  - Averaged on a per-class basis
- Fairness in performance received by different classes
  - Equal service to all classes ?
  - Service in proportion to service requirement ?

Metric choice depends on user requirements and perspectives

Example: A 2-user Interactive System



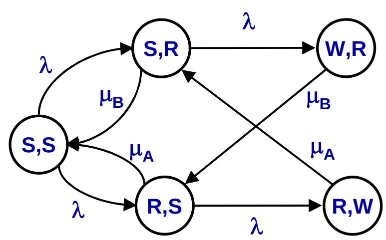
- Average service time of each job released:
  - $D_A = 1/\mu_A = 1$ s
  - $D_B = 1/\mu_A = 2/3$ s
- On job completion, user submits next job in an average of 0.25 s
  - $\lambda = 4$ jobs/s
- Context-switch overhead is assumed to be negligible

Candidate Scheduling Algorithms

- First-come-first-served (FCFS)
- Shortest-job-first (SJF) with preemptive resume
  - Jobs from user B never wait for service
- Longest-job-first (LJF) with preemptive resume
  - Jobs from user A never wait for service
- Last-come-first-served (LCFS)
  - Newly arriving job always has highest priority
- Round-robin (RR)
  - Resource-usage occurs in units of time quanta
  - At end of each quantum, waiting job from other user get served

First Come First Served (FCFS)

- Each user can be in one of 3 states:
  - **(R)**unning, **(W)**aiting, **(S)**leeping
- Represent system state as: (A's state, B's state)
- State transition diagram:



- Flow-balance equations:

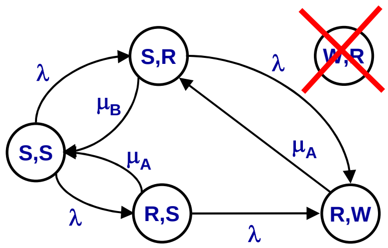
$$\mu_A P_{R,W} = \lambda P_{R,S}$$
$$(\lambda + \mu_A) P_{R,S} = \mu_B P_{W,R} + \lambda P_{S,S}$$
$$2\lambda P_{S,S} = \mu_A P_{R,S} + \mu_B P_{S,R}$$
$$(\lambda + \mu_B) P_{S,R} = \mu_A P_{R,W} + \lambda P_{S,S}$$
$$\mu_B P_{W,R} = \lambda P_{S,R}$$

FCFS: Results

- Steady-state probabilities:
  - $P_{R,W}$ = 304/667 = 0.456; $P_{R,S}$ = 76/667 = 0.114;
  - $P_{S,R}$ = 72/667 = 0.108; $P_{W,R}$ = 192/667 = 0.288
  - $P_{S,S}$ = 1/29 = 0.034;
- Observations:
  - 45.6% of time A is running, while B is waiting
  - 28.8% of the time, B is running, while A is waiting
  - 3.4% of the time, both A and B are sleeping
  - 11.4% of the time, A is running, while B is sleeping
  - 10.8% of the time, B is running, while A is sleeping
- Throughput:
  - $U_A = P_{R,W} + P_{R,S}$; $U_B = P_{W,R} + P_{S,R}$
  - $X_A = U_A/D_A$ = 0.5697 ; $X_B = U_B/D_B$ = 0.5937

Longest Job First (LJF)

- State transition diagram:
  - A never waits for service

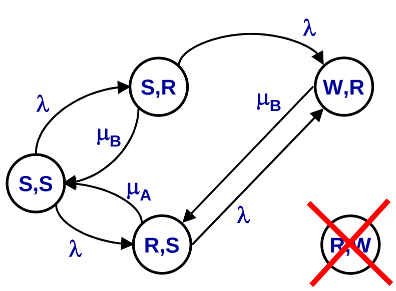


- Flow-balance equations:

$$(\lambda + \mu_A)P_{R,S} = \lambda P_{S,S}$$
$$2\lambda P_{S,S} = \mu_A P_{R,S} + \mu_B P_{S,R}$$
$$(\lambda + \mu_B)P_{S,R} = \mu_A P_{R,W} + \lambda P_{S,S}$$
$$\mu_A P_{R,W} = \lambda P_{R,S} + \lambda P_{S,R}$$

Shortest Job First (SJF)

- State transition diagram:
  - B never waits for service

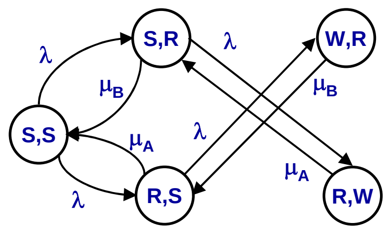


- Flow-balance equations:

$$(\lambda + \mu_B)P_{S,R} = \lambda P_{S,S}$$
$$2\lambda P_{S,S} = \mu_A P_{R,S} + \mu_B P_{S,R}$$
$$(\lambda + \mu_A)P_{R,S} = \mu_B P_{W,R} + \lambda P_{S,S}$$
$$\mu B P_{W,R} = \lambda P_{R,S} + \lambda P_{S,R}$$

Last Come First Served (LCFS)

- State transition diagram:



- Flow-balance equations:

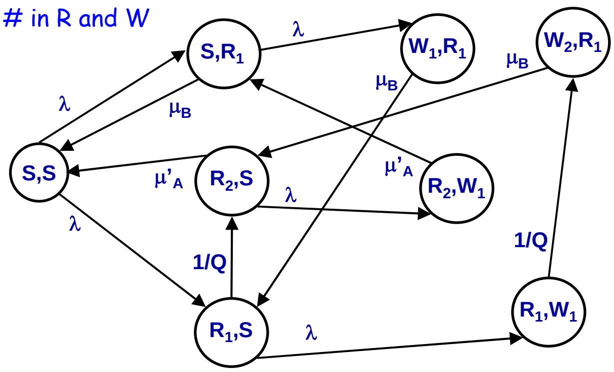
$$\mu_B P_{W,R} = \lambda P_{R,S}$$
$$(\lambda + \mu_A) P_{R,S} = \mu_B P_{W,R} + \lambda P_{S,S}$$
$$2\lambda P_{S,S} = \mu_A P_{R,S} + \mu_B P_{S,R}$$
$$(\lambda + \mu_B) P_{S,R} = \mu_A P_{R,W} + \lambda P_{S,S}$$
$$\mu_A P_{R,W} = \lambda P_{S,R}$$

Round Robin (RR) Scheduling

- Jobs receive service in units of quantum
  - How big should the quantum be?
- Consider three different values:
  - Infinity (very large quanta)
  - Zero (infinitesimally small quanta)
  - Finite value comparable to service demands of jobs
- Infinite quantum size:
  - Quantum always longer than service requirement of any job
    - No job will ever be interrupted
  - When job finishes processing, next waiting job will get service
  - Identical to FCFS!

RR: Finite Non-zero Quantum Size

- If quantum size (Q) = 5/6 seconds
  - Shorter than A's service requirement (1 s)
    - A will take 6/5 quanta to complete
  - Longer than B's requirement (2/3 s)
    - B will take 4/5 quanta to complete
- State transition diagram:
  - Include quantum # in R and W



$$\mu'_A = \frac{1}{1/\mu_A - 1/Q}$$

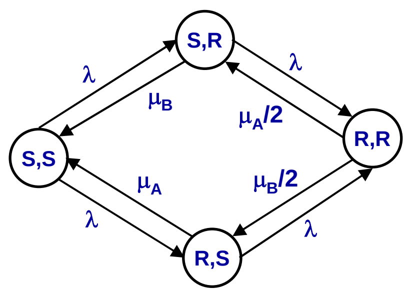
RR: Infinitesimally Small Quanta

- As quantum size decreases, distribution of service among users becomes more uniform
- In the limit as Q → 0,
  - Amount of service received by each user is exactly the same
  - Service rate for each of *n* users = 1/*n*
- Known as the **Processor Sharing (PS)** scheme
  - Abstraction only; not implementable
    - Quantum size lower bounded by system clock resolution
    - Finite context-switch overheads for infinitesimally small quanta
      - Infinite response time!
- Smaller quanta give better fairness
  - But lower throughput due to context-switch overheads

How to model a PS scheduler?

Processor Sharing (PS)

- State transition diagram:



- Flow-balance equations:

$$2\lambda P_{S,S} = \mu_A P_{R,S} + \mu_B P_{S,R}$$

$$(\lambda + \mu_A) P_{R,S} = \frac{\mu_B}{2} P_{R,R} + \lambda P_{S,S}$$

$$(\lambda + \mu_B) P_{S,R} = \frac{\mu_A}{2} P_{R,R} + \lambda P_{S,S}$$

$$\frac{\mu_A + \mu_B}{2} P_{R,R} = \lambda P_{S,R} + \lambda P_{R,S}$$

Comparison on CPU Scheduling Schemes

Policy	Throughput			Response Time			Fairness	
	A	B	Total	A	B	Total	R_A-R_B	$R_A/D_A - R_B/D_B$
FCFS	0.5697	0.5937	1.1634	1.5052	1.4343	1.4691	0.0709	0.6463
LJF	0.8000	0.2482	1.0482	1.000	3.7789	1.6580	2.7789	4.6684
SJF	0.2382	1.0909	1.3291	3.9148	0.6667	1.2548	3.2481	2.9148
LCFS	0.5057	0.6896	1.1954	1.7274	1.2001	1.4231	0.5273	0.0728
RR	0.4638	0.7327	1.1966	1.9056	1.1148	1.4214	0.7913	0.2339
PS	0.5057	0.6896	1.1954	1.7274	1.2001	1.4231	0.5273	0.0728

RR and FCFS are commonly used in today's systems

System Modeling Overview

- Queuing Basics
- Single-server Analysis
- Multiple-server Analysis
- Network of Queues
- Scheduling Case Studies
  - Processor Scheduling
  - **Disk Scheduling**
  - Memory Management