This Week

- Homework
  Chap 5, problems 3, 7 and 9
  Chap 8, problems 1 and 6.
- No tutorial

CPU Scheduling

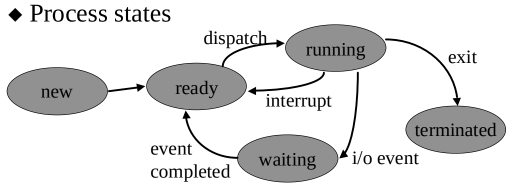
Comp 305, Lecture 5

Today's Lecture

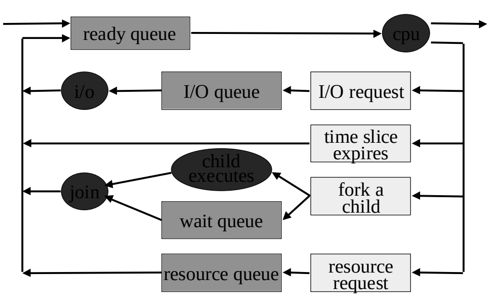
- CPU Scheduling
  - Review Process States
  - Requirements
  - Algorithms
- Memory Mangement

What is a Process?

- Program in execution
- Process states



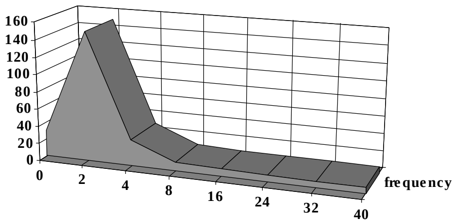
Process Queues



Scheduling Opportunities

- Running process yields the CPU
  - Process enters a wait state
  - Process terminates
- An interrupt occurs
  - Current process is still ready
  - Process switches from wait state to ready
- Preemptive v. non-preemptive scheduling

CPU Burst Duration



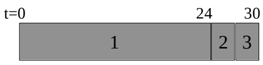
Scheduling Goals

- CPU utilization
- Throughput
- Turnaround time
- Waiting time
- Response time

First Come First Served

- Simple
- Highly variable

Job	Burst
1	24
2	3
3	3

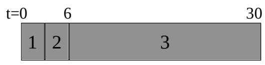


- Average response =27

First Come First Served

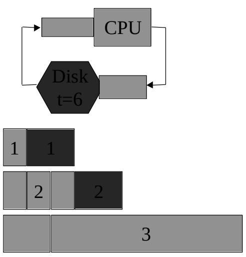
- Simple
- Highly variable

Job	Burst
1	3
2	3
3	24

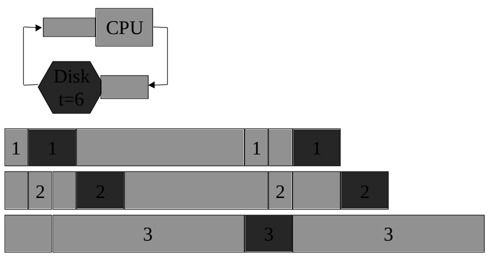


- Average response =13

A System View of FCFS



A System View of FCFS



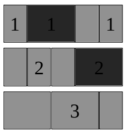
Shortest Job Next

- Always select job with shortest processing time
- Minimal average waiting times
- But we don't know the next CPU burst
- Theory behind several real algorithms
- For example, decaying average:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n$$

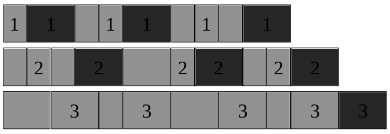
Round Robin

- Each job limited to one quantum
- Quantum chosen to exceed most bursts
- System view with Q=6

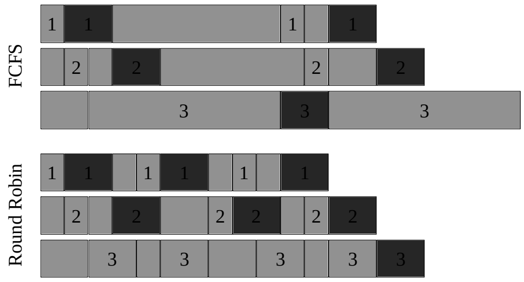


Round Robin

- Each job limited to one quantum
- Quantum chosen to exceed most bursts
- System view with Q=6

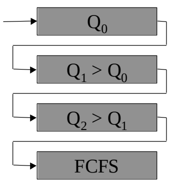


Comparison: FCFS v RR



Multi-Level Feedback

- Priority sinks when quantum exceeded
- Greater discrimination agains longer jobs.
- Better response for shorter jobs.



Unix CPU Scheduling

- Kernel mode priorities
  - Fixed by activity

0 - while swapping
10 - waiting for file ctrl
20 - waiting on disk i/o
25 - PZERO, baseline
30 - waiting on resources
35 - waiting on locks
40 - waiting for an event
50 - PUSER, base user

Unix CPU Scheduling

- Kernel mode priorities
  - Fixed by activity
- User mode priorities
  - *pusrpri* < 128
  - *pnice* reduces priority
  - *pcpu* reflects cpu use

pusrpri =
PUSER + *pcpu*/4 +
2**pnice*

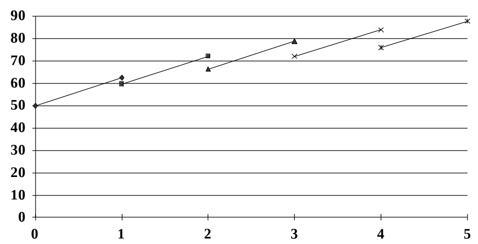
Unix CPU Scheduling

- Kernel mode priorities
  - Fixed by activity
- User mode priorities
  - *pusrpri* < 128
  - *pnice* reduces priority
  - *pcpu* reflects cpu use
  - *load* is weighted average run queue length

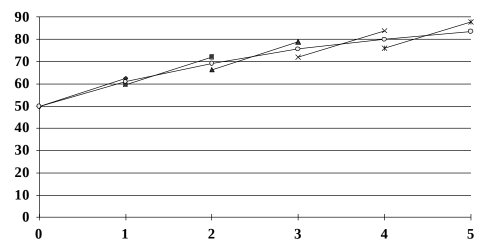
pusrpri =
PUSER + *pcpu*/4 +
2**pnice*

pcpu = *pcpu* *
(2**load*) / (2**load*+1)
+ *pnice*

Unix Decaying Priority



Unix Decaying Priority



Unix Decaying Priority