CSE 5343/7343 Fall 2006
PROCESS SCHEDULING

Professor Margaret H. Dunham
Department of Computer Science and Engineering
Southern Methodist University
Dallas, Texas 75275
(214) 768-3087
fax: (214) 768-3085
email: mhd@seas.smu.edu
www: http://www.seas.smu.edu/∼mhd

October 10, 2006

PROCESS SCHEDULING

- Which process should get the CPU next?
- Long Term
  - switch between new and ready states
  - MP level
  - Pick new Process (FCFS, Priority)
  - Create PCB (Allocate memory, read in program)
  - Primarily batch
  - Executes infrequently
- Intermediate Level
  - Temporarily suspending and activating processes
- Short Term
  - Which processor gets which process?
  - Dispatcher - Gives control of CPU to selected process
  - Executes frequently

PREEMPTIVE/NONPREEMPTIVE SCHEDULING

- Nonpreemptive
  - Process keeps CPU until it decides to give it up
  - Give up at I/O or SVC or termination (When it chooses)
  - Fair treatment
  - Response times more predictable
  - Simple
- Preemptive
  - CPU can be taken away
  - High priority processes require rapid attention
  - Overhead high

MEASURING CPU SCHEDULING PERFORMANCE

- Response Time - Time to first response
- Turnaround Time - Total elapsed time including I/O
- Normalized Turnaround - Turnaround divided by run time
- Throughput - Number of processes per unit time
- Waiting Time - Time in ready queue
- For multiples processes you need the average value
- CPU Utilization

SCHEDULING CONSIDERATIONS

- Availability of limited resources
- Program needed resources
- Priority
- User specified requirements (deadline)
- I/O or CPU Bound (balance resource usage)
- Variance of response times
- Minimize average response time
- Minimize maximum response time
- Maximize throughput
- Avoid starvation

SCHEDULING ALGORITHMS

- FCFS (nonpreemptive)
- SJF (nonpreemptive)
- SPN (nonpreemptive)
- SRT (preemptive)
- HRRN (nonpreemptive)
- Priority (preemptive and nonpreemptive)
- Round-Robin (preemptive)
- Multilevel Queue
- Multilevel Feedback Queue
- Example (No I/O, Context Switch time is 0, Suppose quantum = 0.5 where applicable):

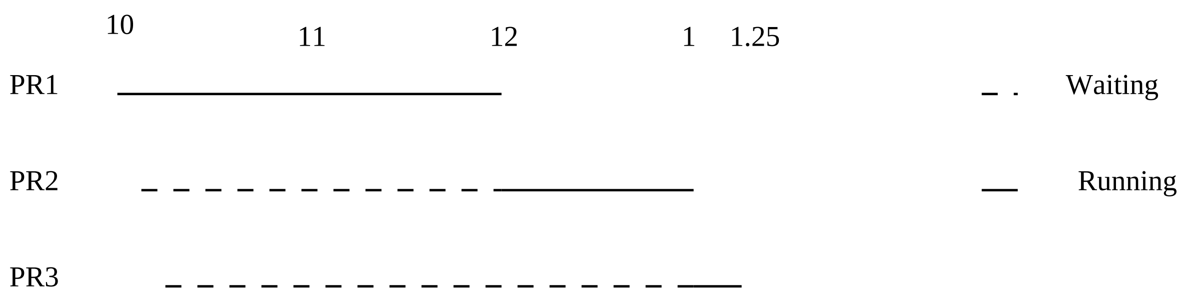
Process	Arrival	Runtime
1	10.00	2.0
2	10.10	1.0
3	10.25	.25

TIME SLICE

- Program gives up CPU at predefined service time
- Avoids Convoy effect
- How Choose?
  - Large - approaches FCFS
  - Small - frequent context switches (May cause thrashing)
  - Maximum Wait Time = (N-1)*quantum
  - Typical values 10-100 milliseconds

FCFS (FIFO)

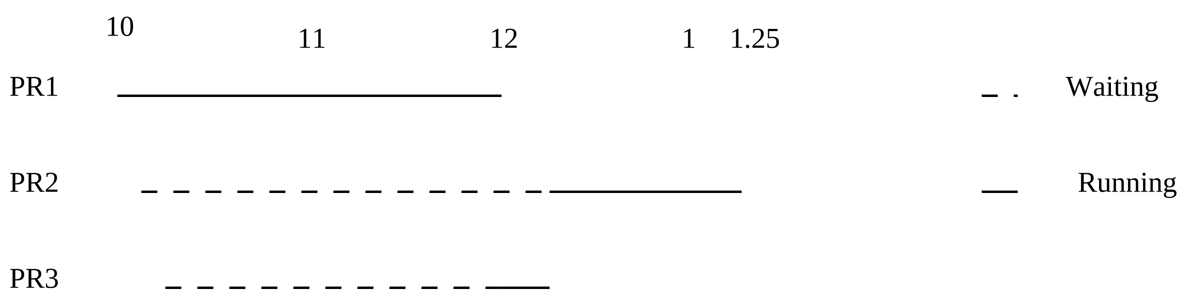
- Simplest - Nonpreemptive
- FIFO Queue - Ordered according to time process arrives
- Convoy Effect
- Poor response time; Large variance
- Example:



Process	Turnaround	Normalized Turnaround
1	2	1
2	2.9	2.9
3	3	12
Average	2.63	5.3

SHORTEST JOB FIRST (SJF)

- Schedule process with shortest CPU burst (total) time
- Nonpreemptive
- Queue ordered according to CPU burst time
- Minimum average waiting time



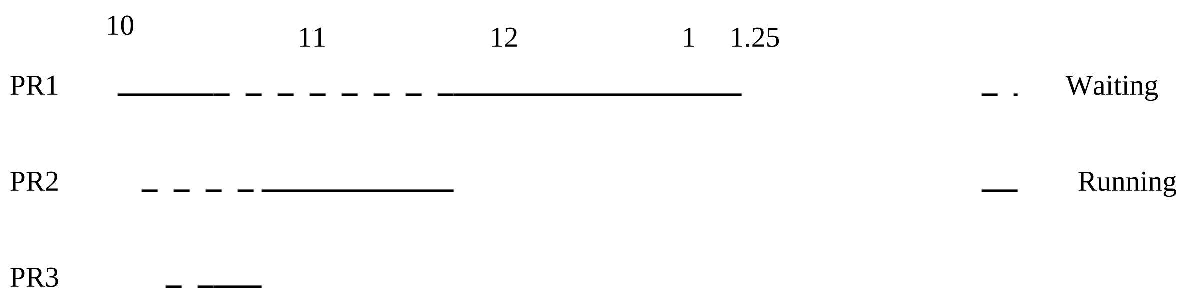
Process	Turnaround	Normalized Turnaround
1	2	1
2	3.15	3.15
3	2	8
Average	2.38	4.05

SHORTEST PROCESS NEXT (SPN)

- Schedules based on prediction of next run time
- Nonpreemptive
- How do you know CPU time? (Burst time)
  - Programmer estimate
  - Historical
  - Similar processes
  - Average of past bursts
- Suppose bursts are $T_1, T_2, ..., T_n$. What is T_{n+1}?
  - $S_{n+1} = T_n$
  - $S_{n+1} = \Sigma_{i=1}^{n} T_i/n$
  - $S_{n+1} = \alpha T_n + (1-\alpha)S_n$ where $0 \leq \alpha \leq 1$
    - Exponential Smoothing
    - Older times have the least weight

SHORTEST REMAINING TIME (SRT)

- Preemptive version of SJF,SPN
- Schedules process with shortest burst time
- Queue ordered according to burst time
- Example (Burst time assumed to be time to completion. Process allowed to continue if shortest time.):



Process	Turnaround	Normalized Turnaround
1	3.25	1.625
2	1.65	1.65
3	.5	2
Average	1.8	1.758

HIGHEST RESPONSE RATIO NEXT (HRRN)

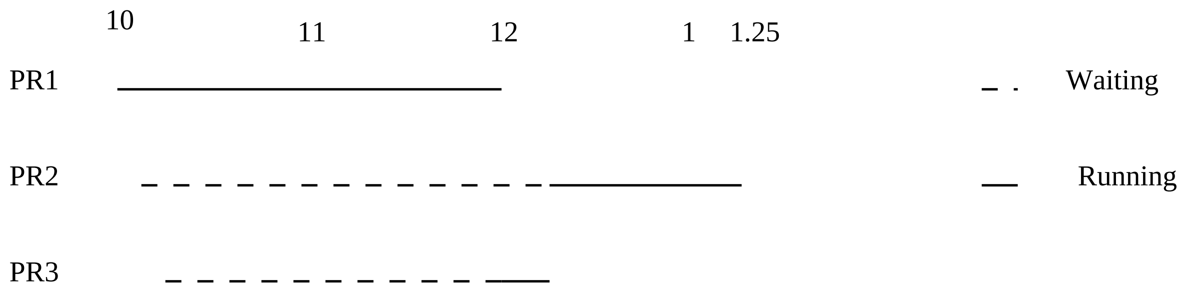
- Schedules process with largest Ratio $R = (w + s)/s$
- Nonpreemptive
- Queue ordered ???
- Extreme overhead

PRIORITY

- Schedule process with highest priority (lowest number)
- Priorities 0 (best) .... n(worst)
- Preemptive and nonpreemptive versions
- Queue ordered according to priorities of processes
- Internal(system)/External(user)
- Static/Dynamic
- Aging - Longer process in system, higher priority. Avoids starvation.

PRIORITY EXAMPLE

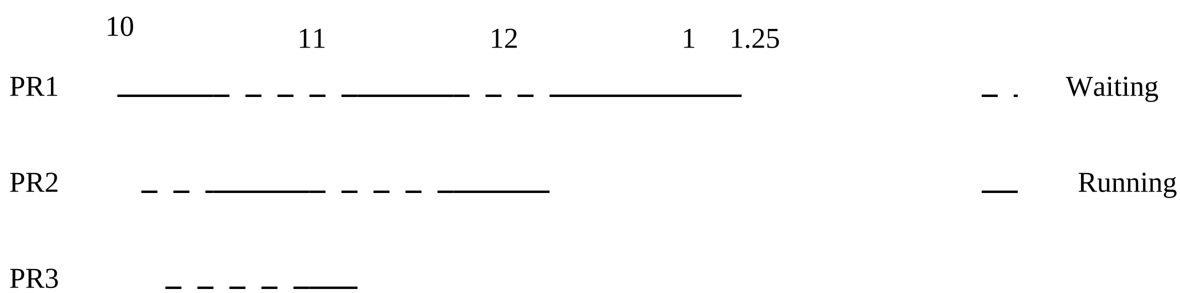
- Assume priorities are 1,3,2 and nonpreemption:



Process	Turnaround	Normalized Turnaround
1	2	1
2	3.15	3.15
3	2	8
Average	2.38	4.05

ROUND ROBIN (RR)

- Preemptive version of FCFS
- Queue ordered according to time process arrives
- Each process allocated a time slice (quantum)
- Popular in time-sharing systems
- Example:



Process	Turnaround	Normalized Turnaround
1	3.25	1.625
2	2.15	2.15
3	1	4
Average	2.133	2.592

IMPACT OF VARYING TIME SLICES

- This table seems to indicate you want quantum as small as possible. Is this really true?

	RR(∞)	RR(1)	RR(.5)	RR(.25)
CPU Util	100	100	100	100
Throughput	.92	.92	.92	.92
Ave Turn	2.63	2.38	2.133	1.967
Ave Wait	1.55	1.3	1.05	.883
Ave Norm	5.3	3.84	2.592	1.925

MULTILEVEL QUEUE

- Multiple queues for scheduling
- Priority among queues
- Multilevel Feedback Queue
  - Multilevel queues
  - Processes move among the queues