CPU Scheduling

Organized By: Vinay Arora
Assistant Professor
CSED, TU

Disclaimer

This is NOT A COPYRIGHT MATERIAL

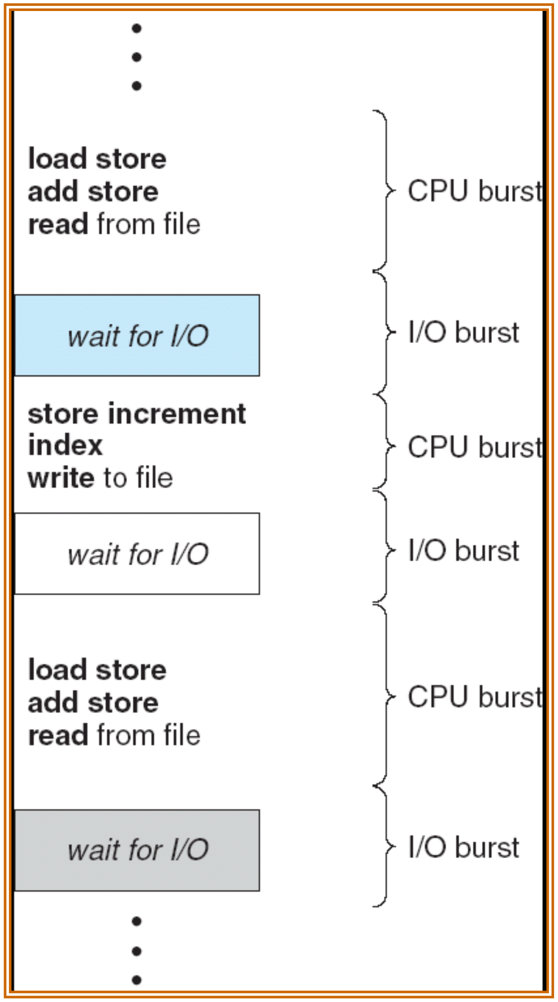
<u>Content has been taken mainly from the following books</u>:

Operating Systems Concepts By Silberschatz & Galvin,
Operating Systems: Internals and Design Principles By William Stallings
www.os-book.com
www.cs.jhu.edu/~yairamir/cs418/os2/sld001.htm
www.personal.kent.edu/~rmuhamma/OpSystems/os.html
http://msdn.microsoft.com/en-us/library/ms685096(VS.85).aspx
http://www.computer.howsttuffworks.com/operating-system6.htm
http://williamstallings.com/OS/Animations.html
Etc…

CPU Scheduling

- CPU Scheduling – Basis of Multiprogrammed Operating Systems.
- Multiprogramming is to have some process running at all times, to maximize CPU Utilization.
- Process Execution consists of a cycle of CPU execution and I/O wait.
- All the processes in the ready queue are lined up waiting for a chance to run on the CPU.
- The Records in the QUEUE are PCB of the Processes.

CPU – I/O Burst


load store
add store
read from file
wait for I/O
store increment
index
write to file
wait for I/O
load store
add store
read from file
wait for I/O
CPU burst
I/O burst
CPU burst
I/O burst
CPU burst
I/O burst

Dispatcher

- Module that gives control of the CPU to the Process selected by the Short Term Scheduler.

- Functions Involved are:

  - Switching Context.
  - Switching to User Mode.
  - Jumping to proper location in the user program to restart that program.

Dispatch Latency – Time it takes for the Dispatcher to Stop one process and Start another Running.

Scheduling Criteria

- CPU Utilization – Keep the CPU as busy as possible
- Throughput – Number of Processes that complete their execution per time unit
- Turnaround Time – Amount of time to Execute a Particular Process
- Waiting Time – Amount of Time a Process has been waiting in the Ready Queue
- Response Time – Amount of time it takes from when a request was submitted until the first response is produced, **not** output (For Time-sharing Environment)

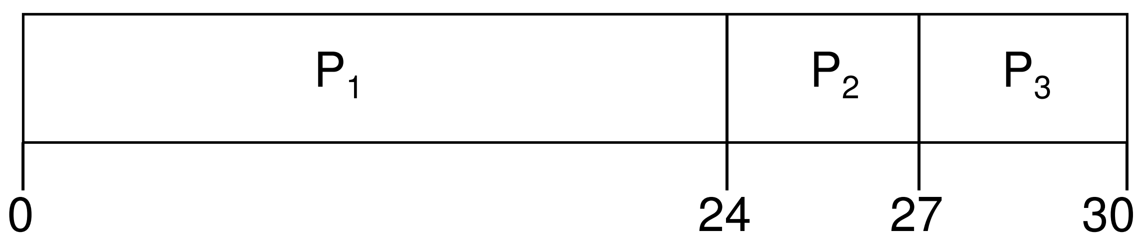
Optimization Criteria

- Max CPU Utilization
- Max Throughput
- Min Turnaround Time
- Min Waiting Time
- Min Response Time

FCFS

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The Gantt Chart for the schedule is:



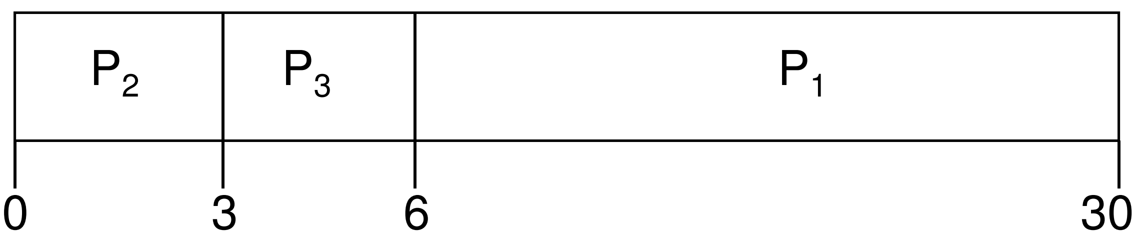
- Waiting time for $P_1 = 0$; $P_2 = 24$; $P_3 = 27$
- Average waiting time: $(0 + 24 + 27)/3 = 17$

FCFS

Suppose that the processes arrive in the order

P_2 , P_3 , P_1

- The Gantt chart for the schedule is:



- Waiting time for $P_1 = 6$; $P_2 = 0$; $P_3 = 3$
- Average waiting time: $(6 + 0 + 3)/3 = 3$
- Much better than previous case

- Convoy Effect short process behind long process

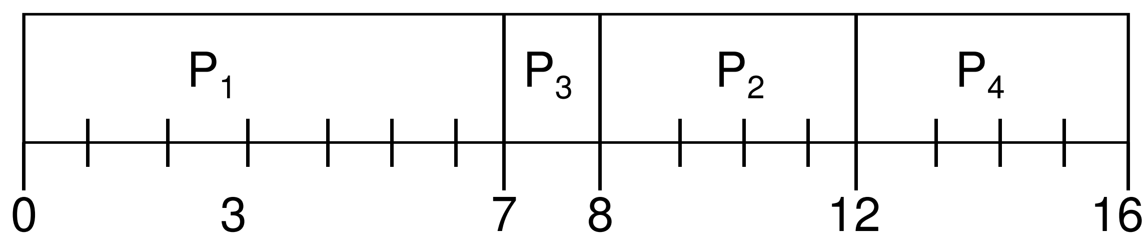
SJF

- Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time
- Two Schemes:
  - Non Preemptive – Once CPU given to the process it cannot be preempted until completes its CPU burst
  - Preemptive – If a New Process arrives with CPU burst length less than remaining time of current executing process, preempt. This scheme is know as the Shortest-Remaining-Time-First (SRTF)
- SJF is Optimal – Gives minimum average waiting time for a given set of processes

SJF (Non Preemptive)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (Non-Preemptive)

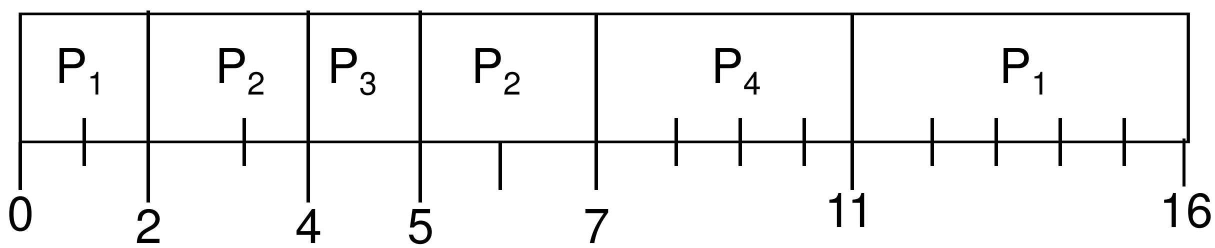


- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

SJF (Preemptive)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (Preemptive)



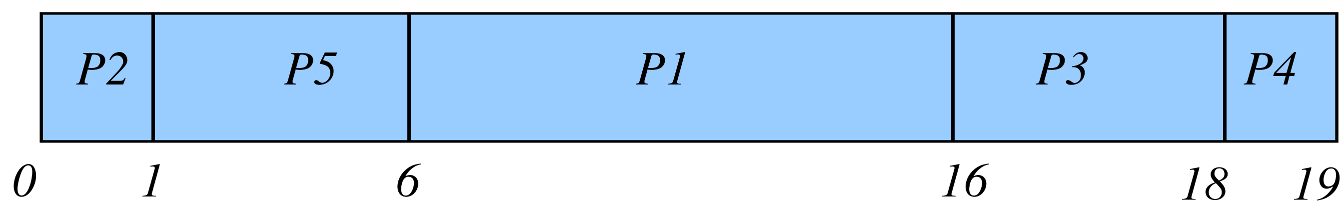
- Average waiting time = (9 + 1 + 0 +2)/4 = 3

Priority Scheduling

- A Priority Number (Integer) is associated with each Process.
- The CPU is allocated to the Process with the Highest Priority (Smallest Integer ≡ highest priority)
  - Preemptive
  - Non Preemptive
- SJF is a Priority Scheduling where PRIORITY IS THE PREDICTED NEXT CPU BURST TIME
- Problem in Priority scheduling is STARVATION – Low Priority processes may never execute
- Solution for above mentioned Problem is AGING – as time progresses increase the priority of the process

Priority Scheduling

Process	Priority	Burst Time
P_1	3	10
P_2	1	1
P_3	4	2
P_4	5	1
P_5	2	5



Round Robin

- Each Process gets a Small Unit of CPU time (Time Quantum).
- After this time has elapsed, the PROCESS IS PREEMPTED and added to the end of the Ready Queue.
- If there are n Processes in the Ready Queue and the time quantum is q, then each process gets at most q time units at once.
- *PERFORMANCE*

  q Large $\Rightarrow$ FIFO

  q Small $\Rightarrow$ q must be large with respect to Context Switch, Otherwise overhead is too high

Round Robin

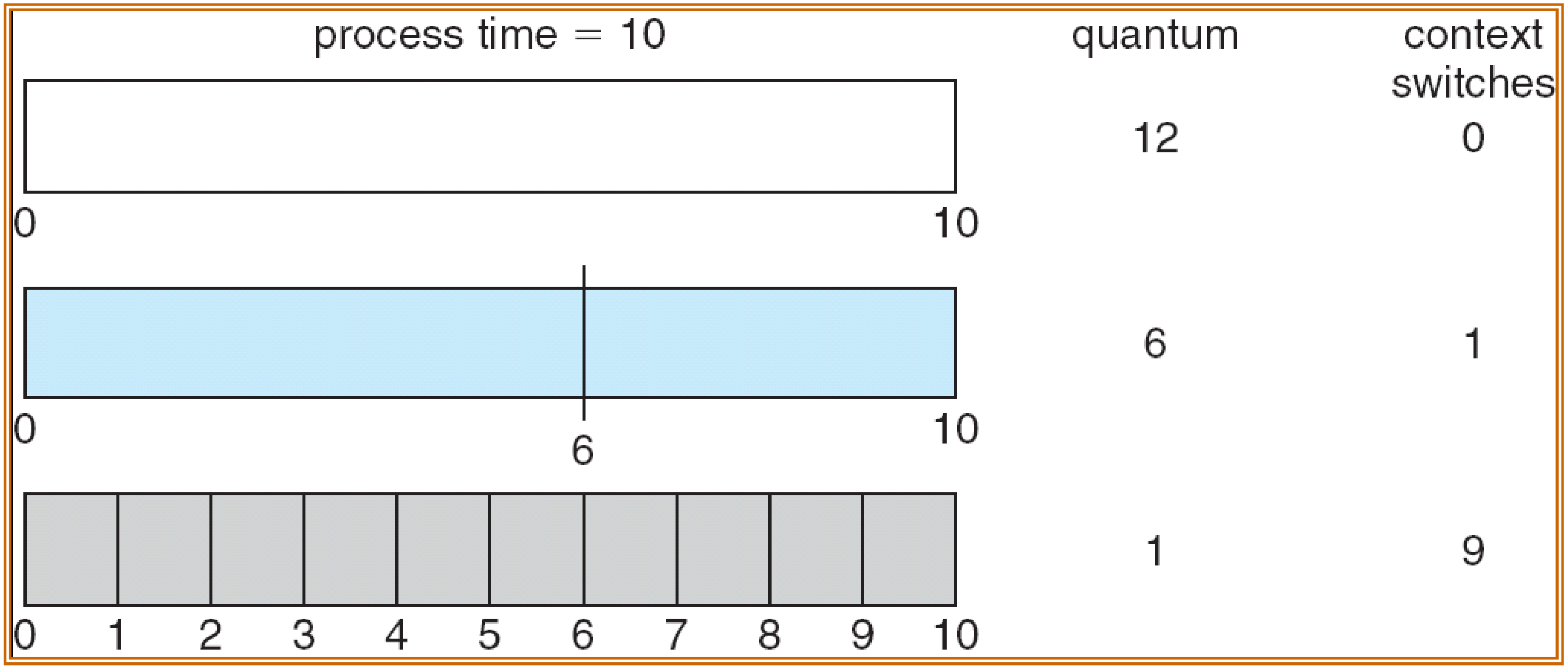
Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

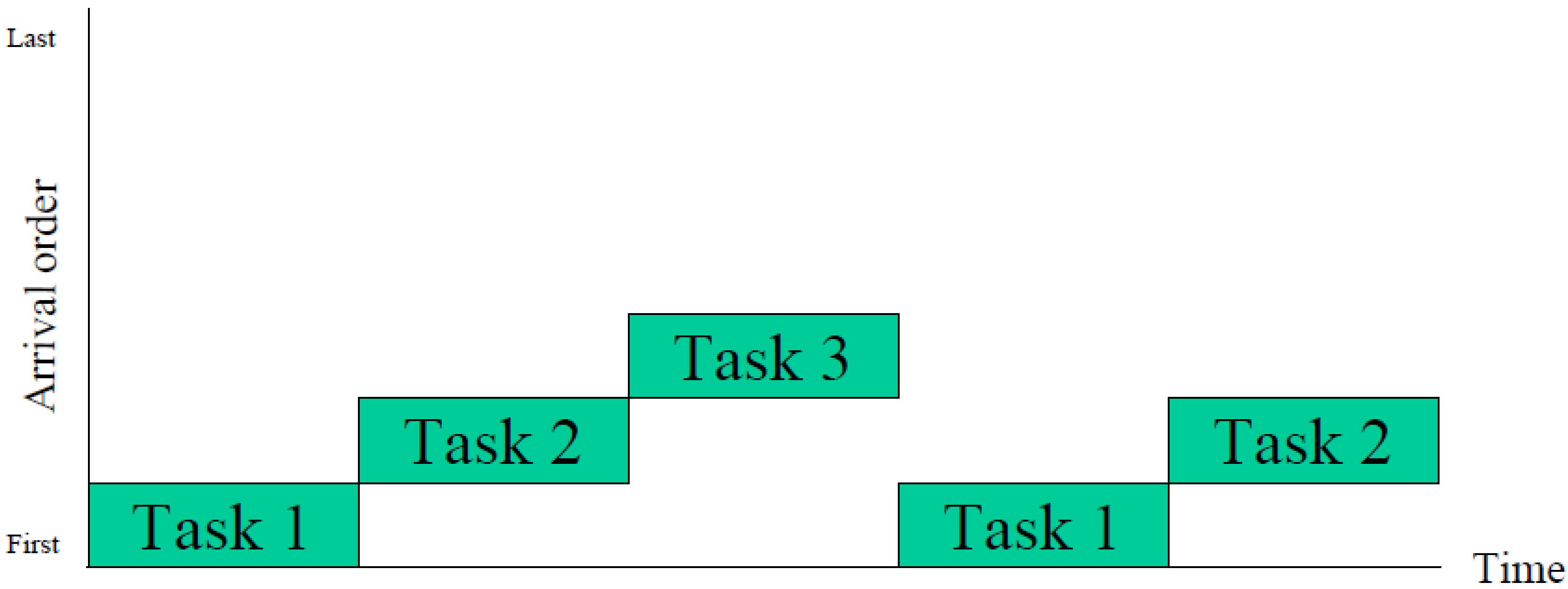
- The Gantt Chart is:

P_1	P_2	P_3	P_4	P_1	P_3	P_4	P_1	P_3	P_3

0 20 37 57 77 97 117 121 134 154 162

Time Quantum and Context Switch



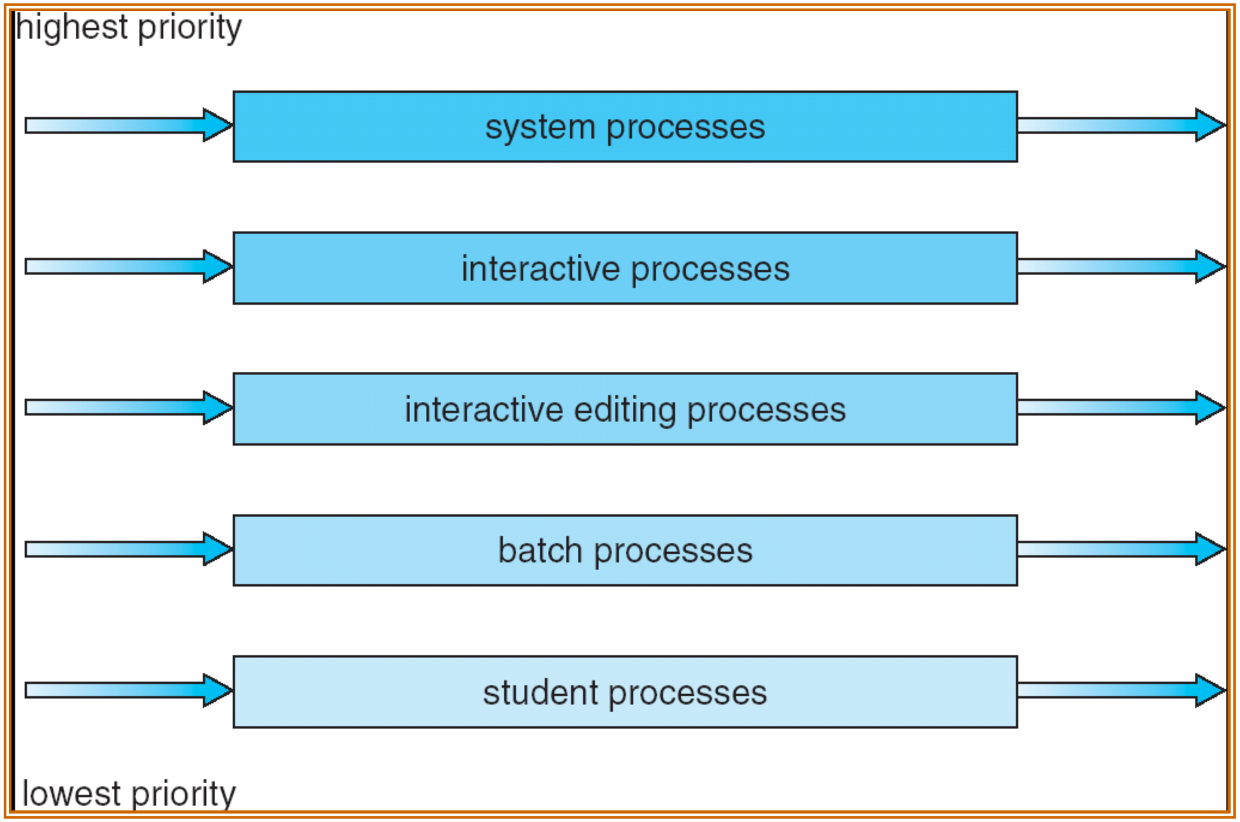

Last
Arrival order
First
Task 1
Task 2
Task 3
Task 1
Task 2
Time

MultiTasking

Multilevel Queue

- Ready Queue is partitioned into separate Queues:
  - Foreground
  - Background
- Each QUEUE has its own Scheduling Algorithm
  - Foreground – RR
  - Background – FCFS
- Scheduling must be done between the queues
  - Fixed Priority Scheduling - (i.e., Serve all from foreground then from background). Possibility of Starvation.
  - Time Slice – Each Queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR
  - 20% to background in FCFS

Multilevel Queue Scheduling



Algorithm Evaluation

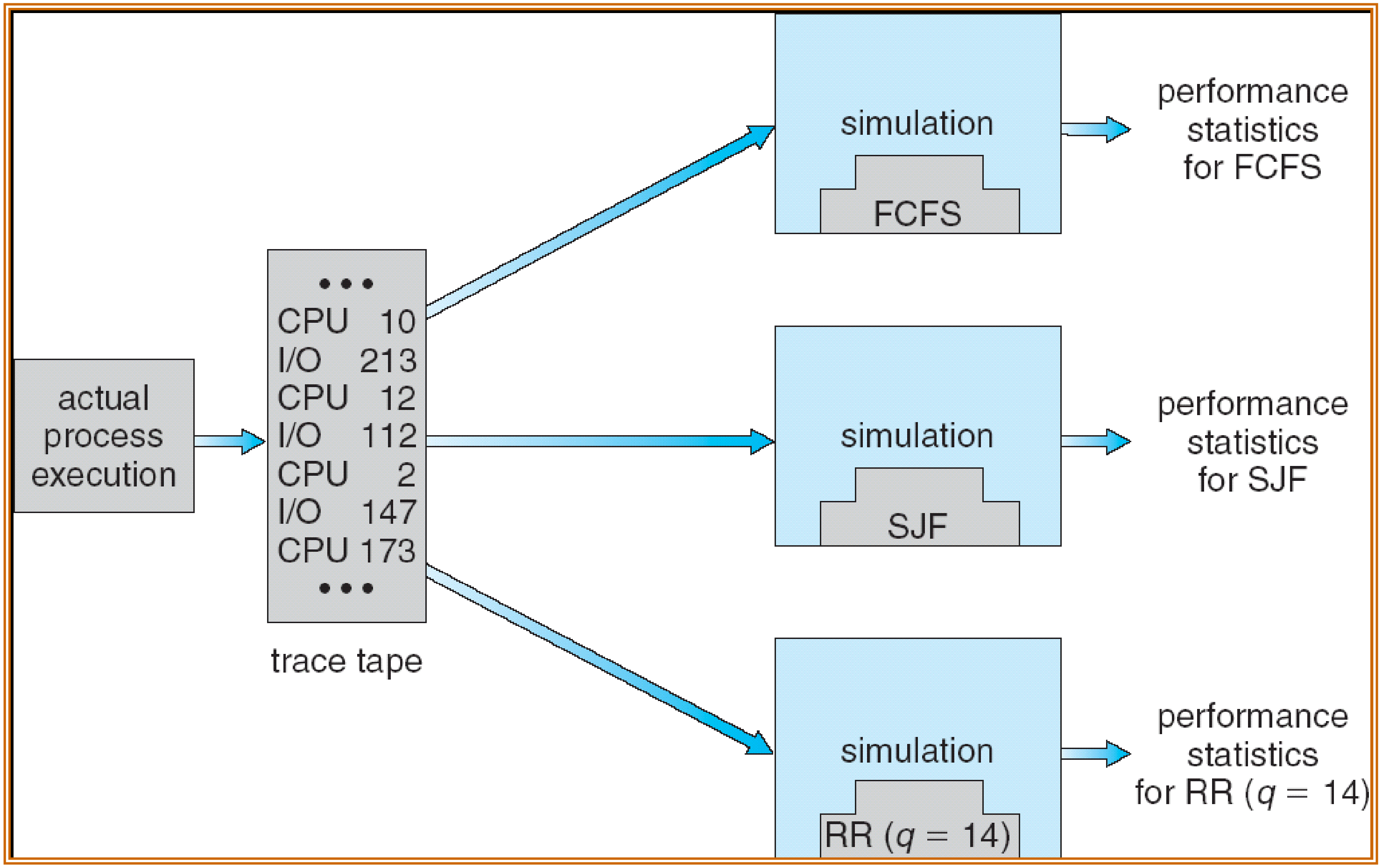
- Deterministic Modeling:-

  Takes a particular Predetermined workload

  Defines the Performance of each Algorithm for that workload

- Queuing Models

- Implementation


actual process execution
• • •
CPU 10
I/O 213
CPU 12
I/O 112
CPU 2
I/O 147
CPU 173
• • •
trace tape
simulation
FCFS
performance statistics for FCFS
simulation
SJF
performance statistics for SJF
simulation
RR (q = 14)
performance statistics for RR (q = 14)

Thnx…