Scheduling Algorithms

- FCFS (First Comes First Served)
  - Non-Preemptive and Simplest CPU-scheduling algorithm
  - Process requesting CPU first is allocated CPU first
- Shortest-Job-First Scheduling (Non-Preemptive)
  - Assign CPU to the process with smallest next CPU burst
  - Next CPU bursts of 2 processes same, FCFS scheduling
- Shortest-Remaining-Time-First (SRTF)
  - A preemptive version of SJF algorithm
  - If the next CPU burst of the newly arrived process is shorter than what is left of the currently executing process, it will preempt the currently executing process

1

Priority Scheduling

- A priority is associated with each process
- CPU is allocated to highest priority process
  - Equal-priority processes are scheduled in FCFS order
- Priorities are generally indicated by some fixed range of numbers, such as 0 to 7 or 0 to 1,023
- External (Static)
  - non-preemptive
- Internal (Dynamic)
  - preemptive

2

Priority Scheduling

- Static Priority-non-preemptive
  - Higher number→ Higher Priority

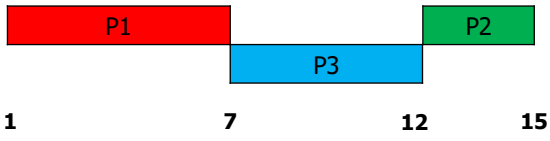
Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	2

3

Priority Scheduling

- Static Priority-non-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	2



4

Priority Scheduling

- Static Priority-non-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	2

5

Priority Scheduling

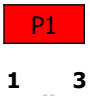
- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4

Ready	Running
P1(6)	-

➡

Ready	Running
P2(3)	P1(4)



6

Priority Scheduling

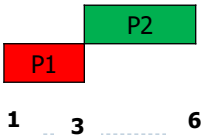
- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4

Ready	Running
P1(4)	P2(3)

➡

Ready	Running
P3(5)	-
P1(4)	



7

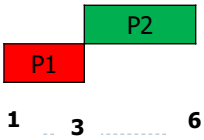
7

Priority Scheduling

- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4

Ready	Running
P1(4)	P3(5)



8

8

Priority Scheduling

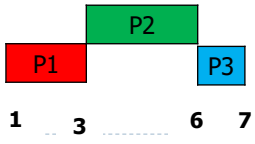
- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4

Ready	Running
P1(4)	P3(5)

➡

Ready	Running
P4(2)	P3(4)
P1(4)	



Priority Scheduling

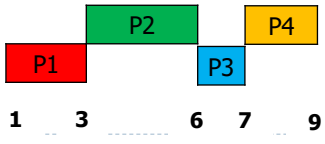
- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4

Ready	Running
P3(4)	P4(2)
P1(4)	

➡

Ready	Running
P3(4)	-
P1(4)	



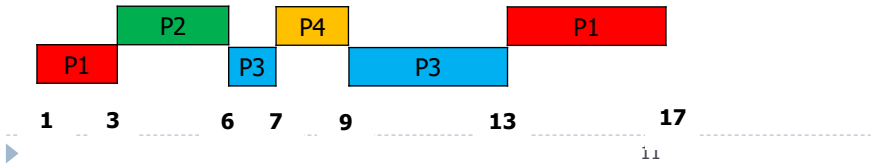
Priority Scheduling

- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Ready	Running
P1(4)	P3(4)

Ready	Running
-	P1(4)

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4



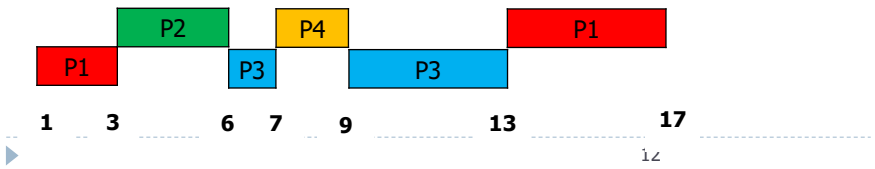
11

Priority Scheduling

- Dynamic Priority-preemptive
  - Higher number→ Higher Priority

Ready	Running
-	-

Process	Arrival	Burst	Priority
P1	1	6	1
P2	3	3	2
P3	6	5	3
P4	7	2	4



12

Round Robin Scheduling

- Designed especially for timesharing systems
- FCFS + Preemption (to context switch processes)
- Time quantum/slice defined (10 to 100 milliseconds)
- Ready queue is a circular queue
- Time slice > time required for a typical interaction

Round Robin Scheduling

Ready	Running
P1(7)	-

Ready	Running
-	P1(4)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units

P1

1 **4**

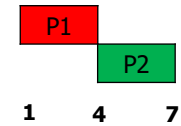
Round Robin Scheduling

Ready	Running
P2(4)	P1(4)
P3(2)	

Ready	Running
P3(2)	P2(4)
P1(4)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



15

15

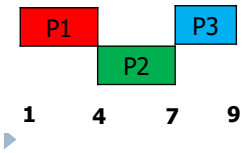
Round Robin Scheduling

Ready	Running
P3(2)	P2(1)
P1(4)	
P4(5)	

Ready	Running
P1(4)	P3(2)
P4(5)	
P2(1)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



16

16

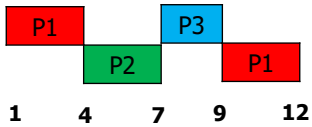
Round Robin Scheduling

Ready	Running
P4(5)	P1(4)
P2(1)	

Ready	Running
P4(5)	P1(1)
P2(1)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



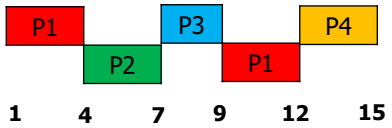
Round Robin Scheduling

Ready	Running
P2(1)	P4(5)
P1(1)	

Ready	Running
P2(1)	P4(2)
P1(1)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



Round Robin Scheduling

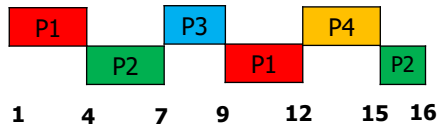
Ready	Running
P1(1)	P2(1)
P4(2)	

➡

Ready	Running
P4(2)	P1(1)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



Round Robin Scheduling

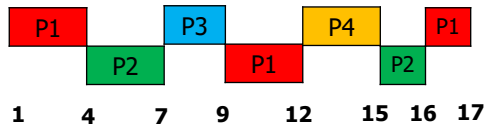
Ready	Running
P4(2)	P1(1)

➡

Ready	Running
-	P4(2)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units



Round Robin Scheduling

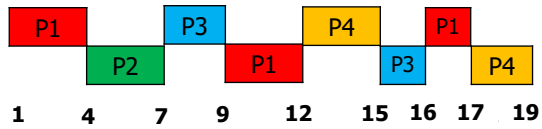
Ready	Running
-	P4(2)

➡

Ready	Running
-	-

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	2
P4	5	5

Time slot = 3 units

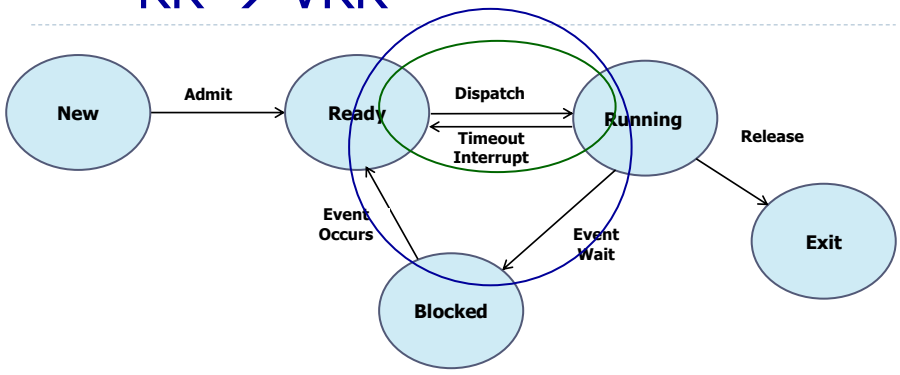


Round Robin Scheduling

- Effective in a general-purpose time-sharing system
- Unfair scheduling for I/O-bound processes
- To avoid this unfairness, refinement to Round Robin is explored

Virtual Round Robin (VRR)

RR → VRR



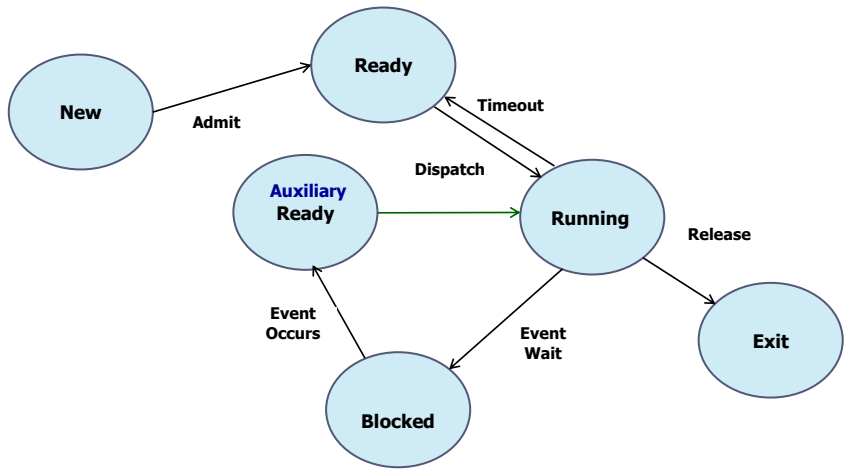
- New → Ready → Running ←→Ready
- Running → Wait (Blocked on IO)
- Compute Bound Vs IO Bound Processes
- VRR (RR + Auxiliary Ready Queue for IO bound processes

Virtual Round Robin (VRR)

Auxiliary Queue (FCFS)

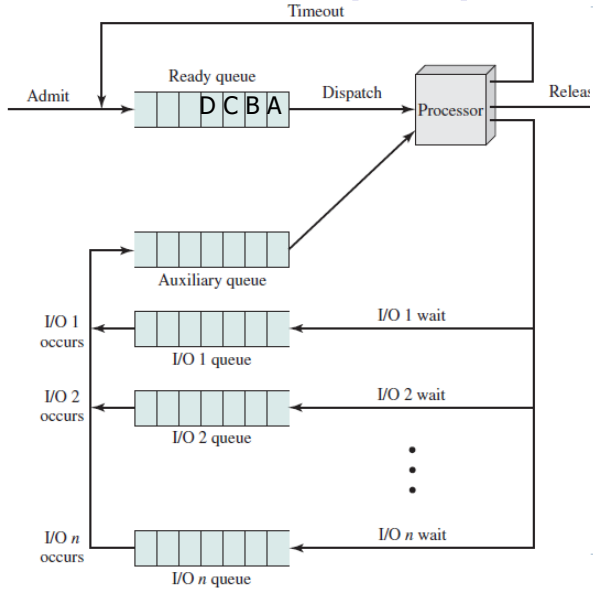
- Processes moved to Auxiliary Queue after being released from an I/O block
- When a dispatching CPU decision is to be made, processes in the Auxiliary queue get preference over main ready queue
  - A process dispatched from auxiliary queue, runs its remaining time of time slice last allocated when it initiates an I/O

Process Model of VRR



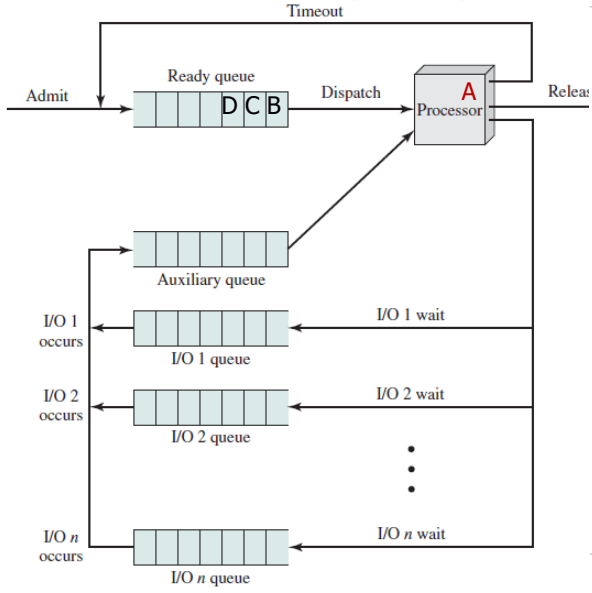
25

Virtual Round Robin (VRR)



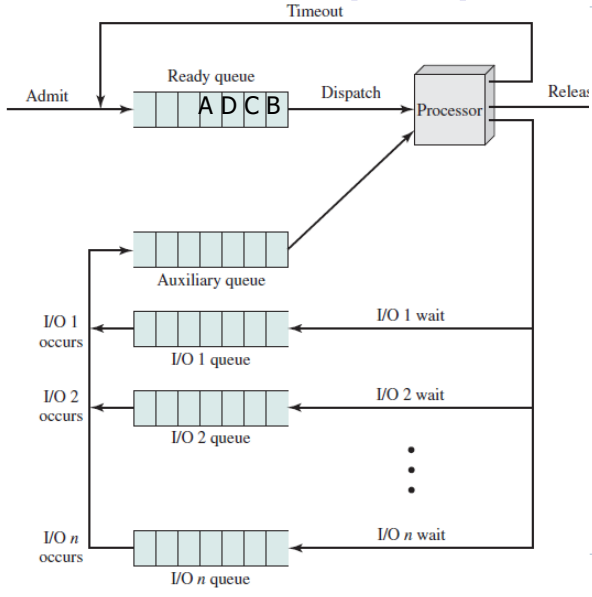
26

Virtual Round Robin (VRR)



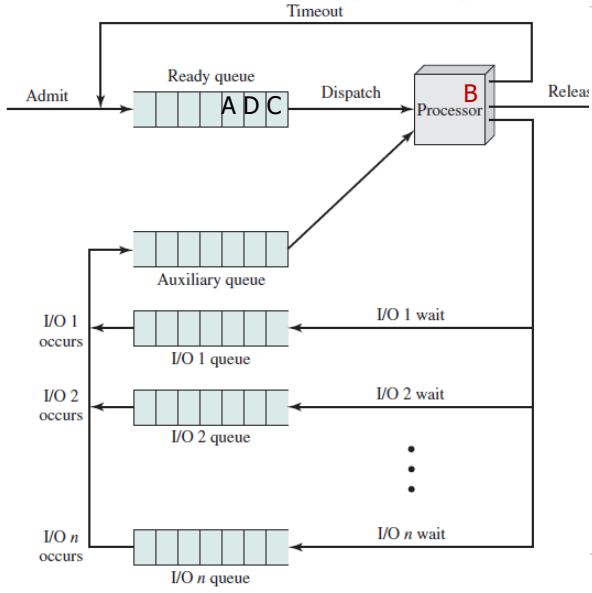
27

Virtual Round Robin (VRR)



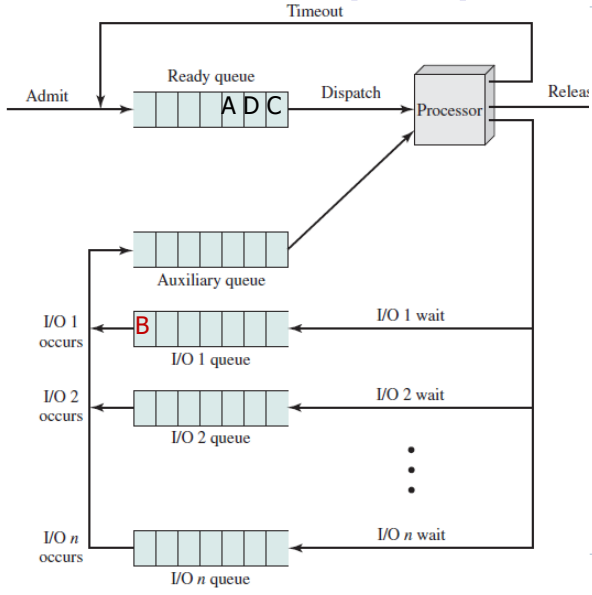
28

Virtual Round Robin (VRR)



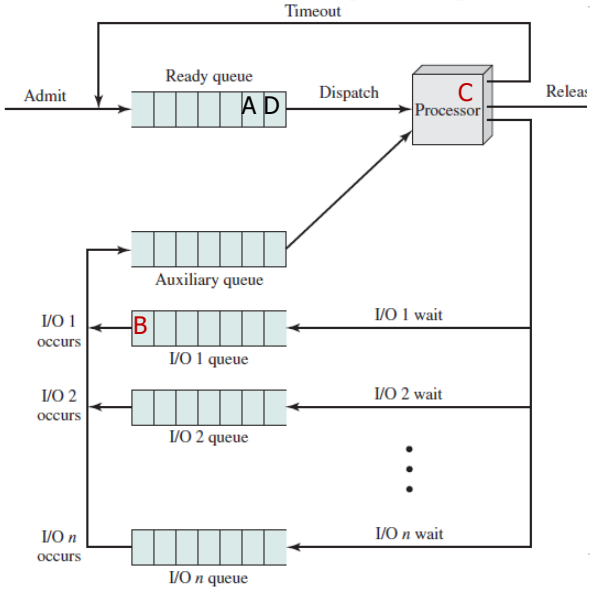
29

Virtual Round Robin (VRR)



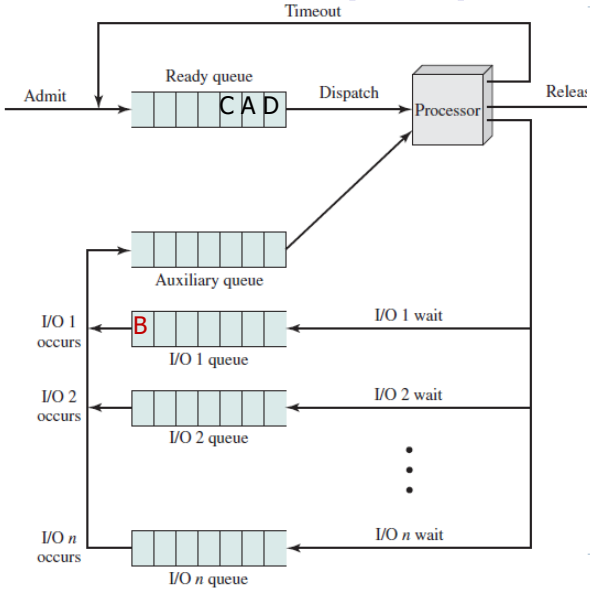
30

Virtual Round Robin (VRR)



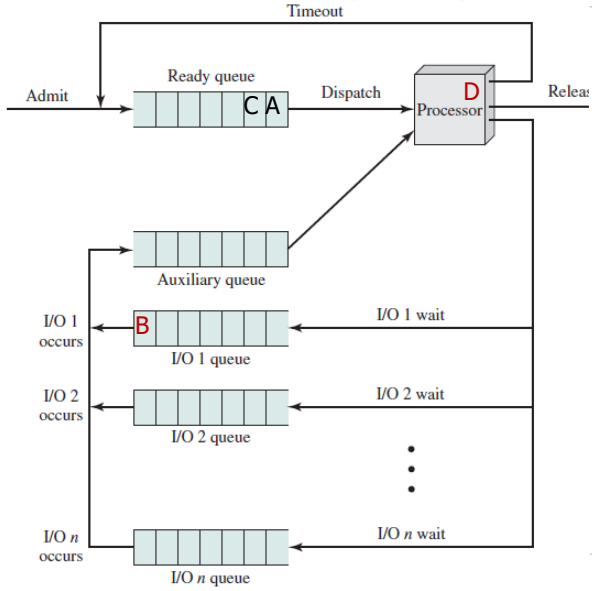
31

Virtual Round Robin (VRR)



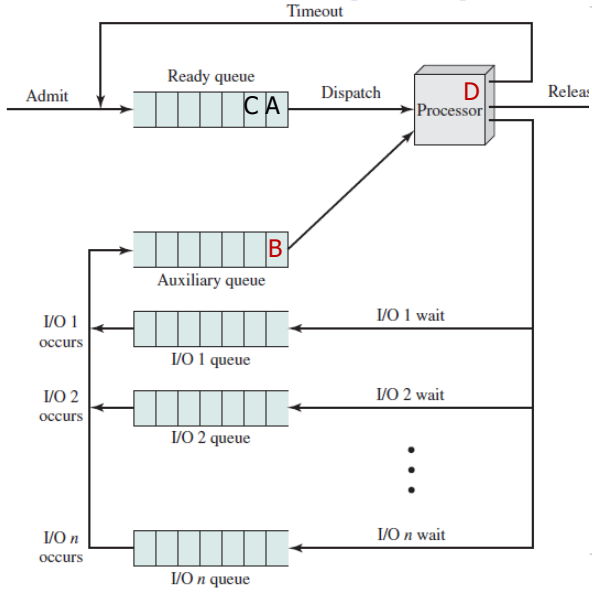
32

Virtual Round Robin (VRR)



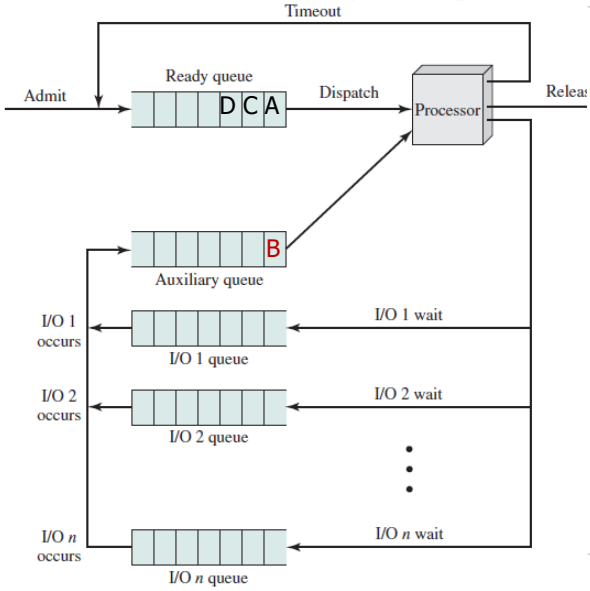
33

Virtual Round Robin (VRR)



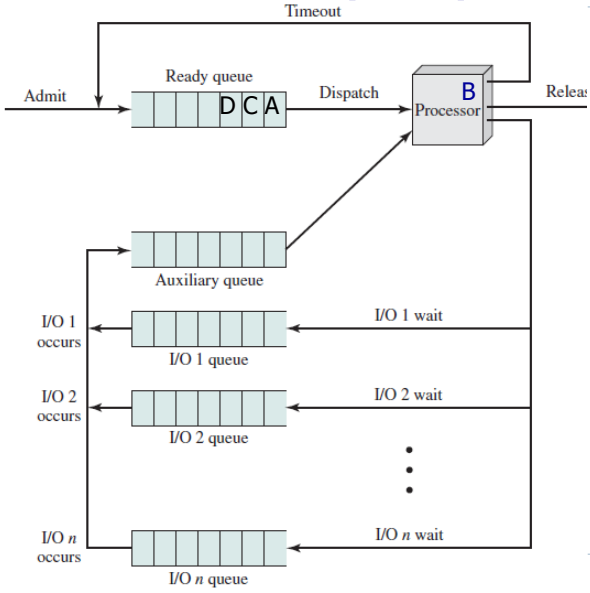
34

Virtual Round Robin (VRR)



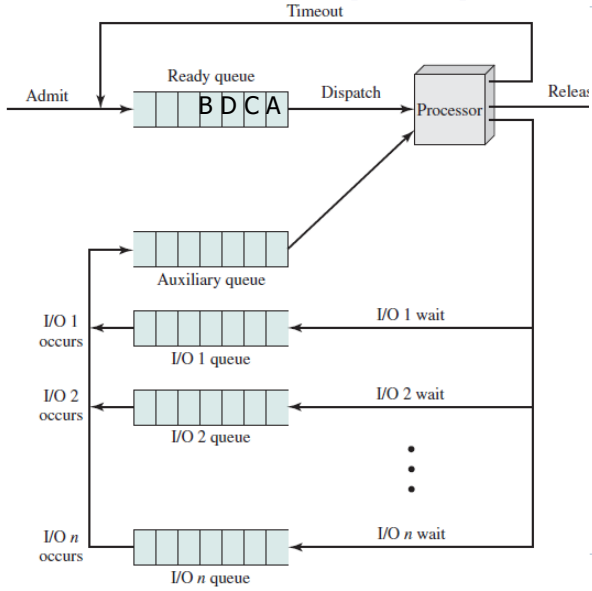
35

Virtual Round Robin (VRR)



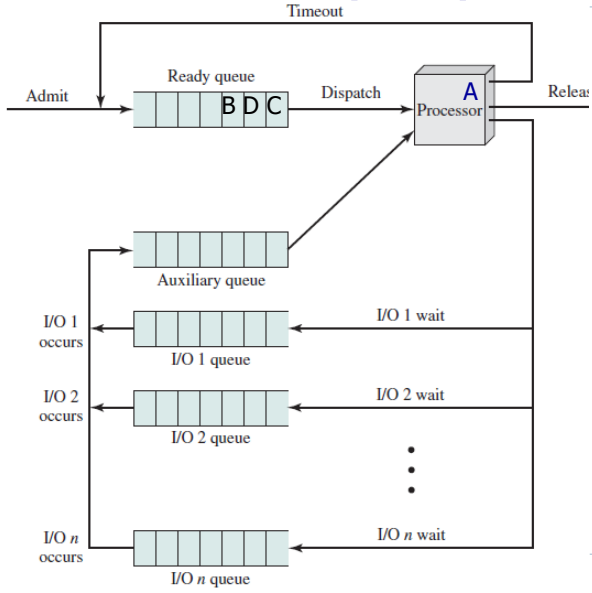
36

Virtual Round Robin (VRR)



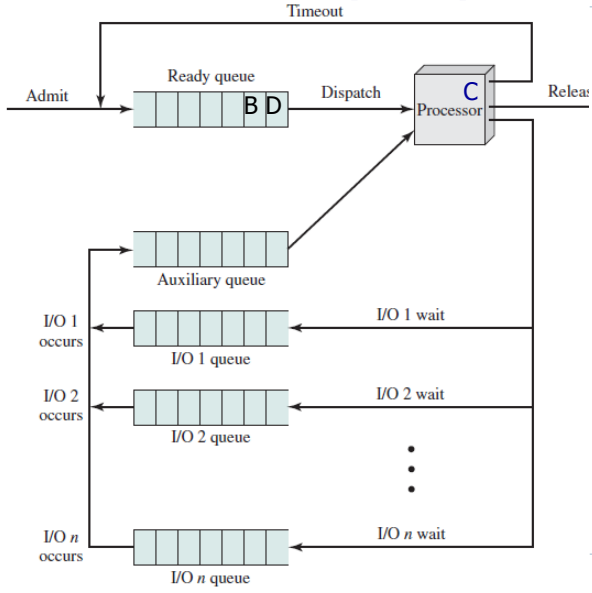
37

Virtual Round Robin (VRR)



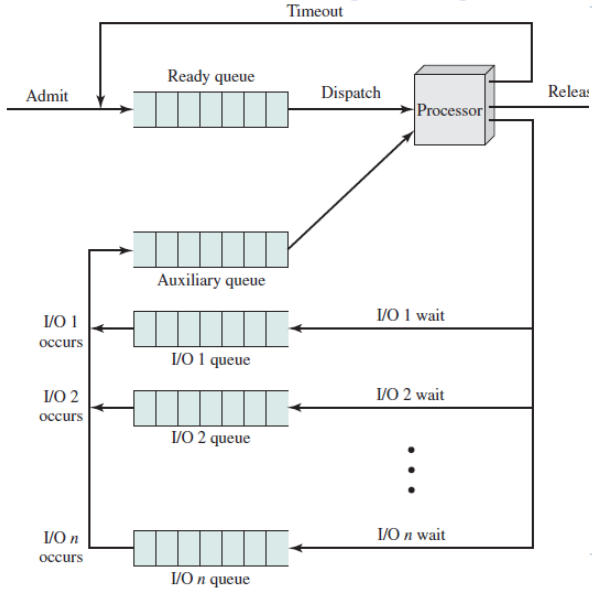
38

Virtual Round Robin (VRR)



39

Virtual Round Robin (VRR)



40

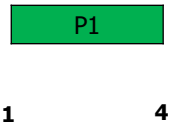
Virtual Round Robin

Ready	Running
P1(7)	-

Ready	Running
-	P1(7)
P2(4)	
P3(5)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Time slot = 3 msec
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



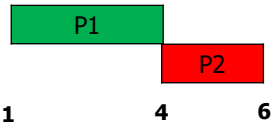
Virtual Round Robin

Ready	Running
P2(4)	-
P3(5)	
P1(4)	

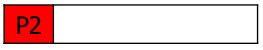
Ready	Running
P3(5)	P2(4)
P1(4)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue



Virtual Round Robin

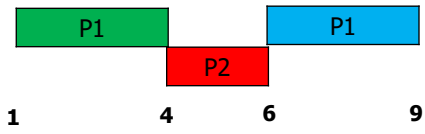
Ready	Running
P3(5)	-
P1(4)	

➡

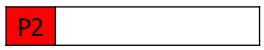
Ready	Running
P1(4)	P3(5)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	3
P3	3	5

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue



11

43

43

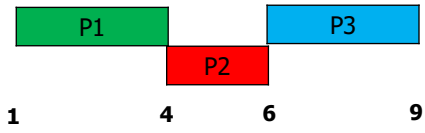
Virtual Round Robin

Ready	Running
P3(2)	P1(4)

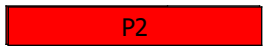
Auxiliary Ready
P2(2)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue



11

44

44

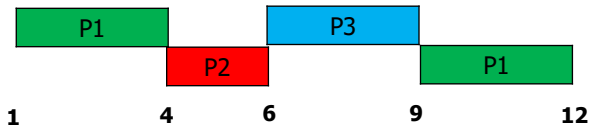
Virtual Round Robin

Ready	Running
P3(2)	P1(4)

Auxiliary Ready
P2(2)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue

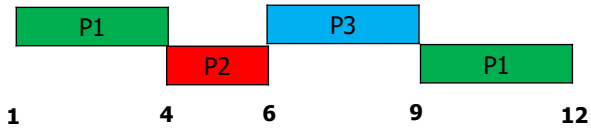
Virtual Round Robin

Ready	Running
P3(2)	-
P1(1)	

Auxiliary Ready
P2(2)

Process	Arrival Time	Burst Time
P1	1	7
P2	2	3
P3	3	5

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue

Virtual Round Robin

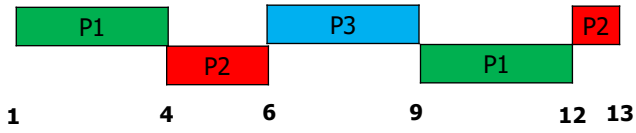
Ready	Running
P3(2)	P2(2)
P1(1)	

Ready	Running
P3(2)	-
P1(1)	
P2(1)	

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Auxiliary Ready
-

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec



IO Blocked Queue

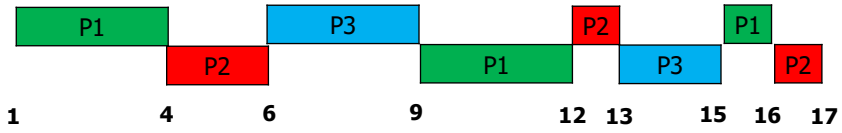
Virtual Round Robin

Ready	Running
-	-

Process	Arrival Time	Burst Time
P1	1	7
P2	2	4
P3	3	5

Auxiliary Ready
-

Time slot = 3 units
P2-IO bound
Initiates IO after 2msec CPU
IO takes 5 msec

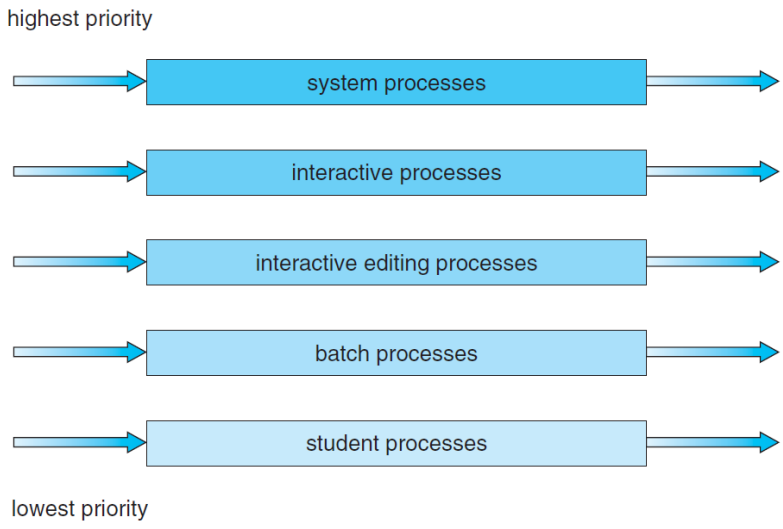


IO Blocked Queue

Multilevel Queue Scheduling

- Partitions ready queue into several separate queues
- Processes are permanently assigned to one queue
  - memory size, process priority, or process type
- Each queue has its own scheduling algorithm
  - foreground and background processes
  - foreground queue might be scheduled by RR
  - background queue is scheduled by FCFS
- Scheduling among the queues (fixed-priority preemptive scheduling)

Multilevel Queue Scheduling



Multilevel Queue Scheduling

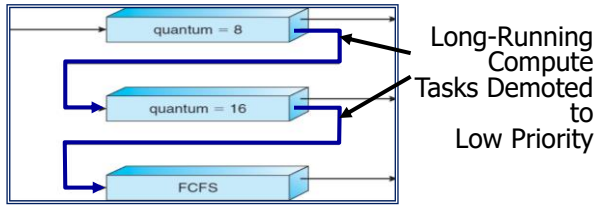
- Time-slicing among the queues
  - Each queue gets a certain portion of the CPU time
    - schedule among its various processes
  - foreground–background queue example
    - foreground queue with 80 percent of the CPU time for RR scheduling among its processes
    - background queue with 20 percent of the CPU to give to its processes on an FCFS basis

Multilevel Feedback Queue Scheduling

- Exploiting past behavior
- Multiple queues, each with different priority
- Allows a process to move between queues
- Idea is to separate processes according to the characteristics of their CPU bursts
- Each queue has its own scheduling algorithm
  - e.g. foreground – RR, background – FCFS
- Sometimes multiple RR priorities with quantum increasing exponentially (highest:4ms, next:8ms, next: 16ms, etc.)
  - A process uses too much CPU time → move to a lower-priority queue

Multilevel Feedback Queue Scheduling

- Job starts in highest priority queue
- If timeslot expires, drop one level
- If timeslot doesn't expire, push up one level (or to top)



Multilevel Feedback Queue Scheduling