CS3841 – Scheduling

Scheduling Considerations:

- The OS needs to schedule processes to run. The OS has several options each with considerations
  - Pre-emptive – Do processes run to completion without being stopped?
  - Round Robin – How long should the time slice be?
  - Multi-processor – Which processor should processes be run on? What about cache efficiency? What about cache coherency?
  - Priority – Is one process more important than another?
  - Fairness – I/O Intensive processes vs CPU intensive processes

Fairness

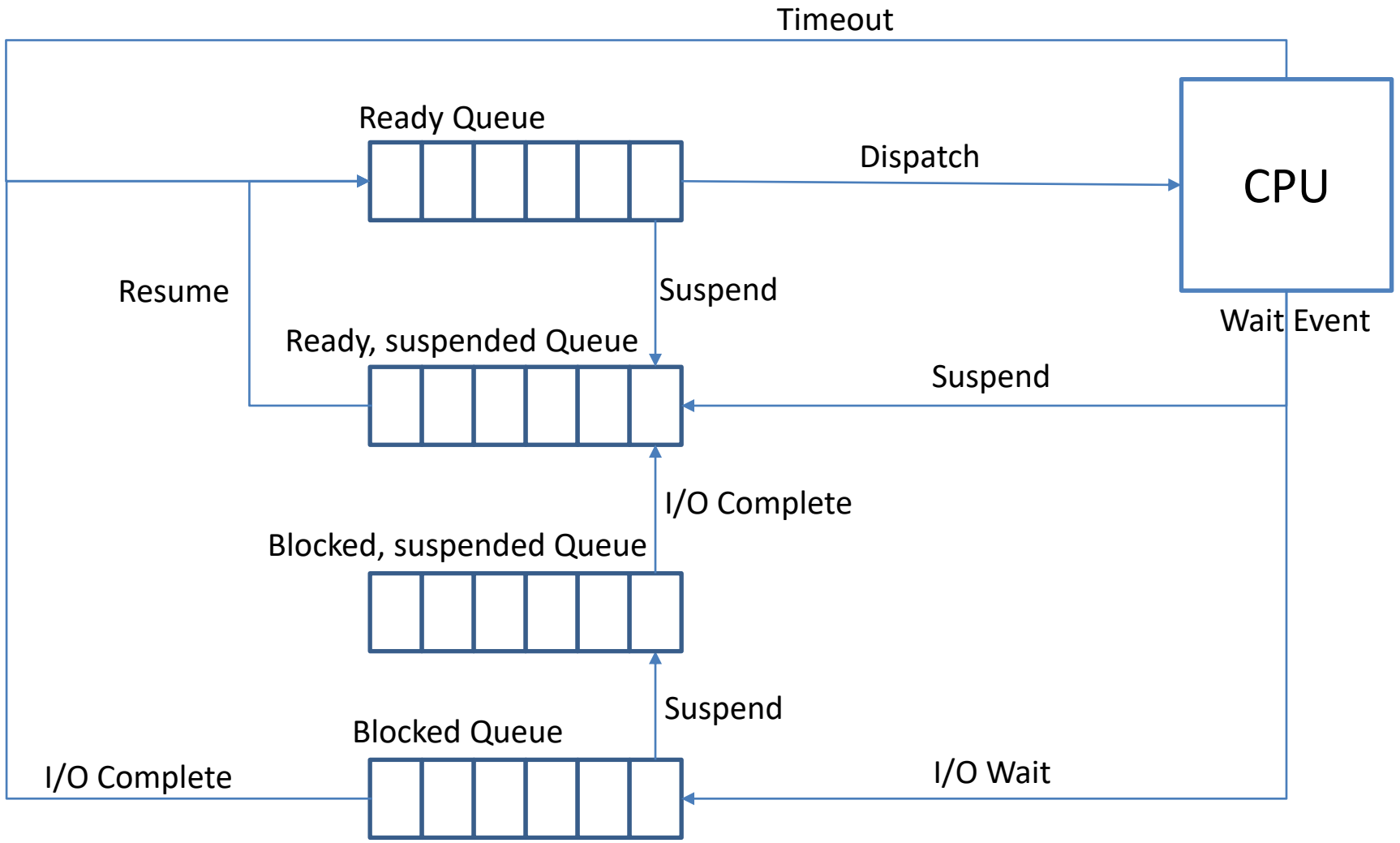
- Want to schedule processes so that all make forward progress in a fair way.
- Don't want to 'starve' a process of CPU access

- I/O intensive – Processes that take more time waiting for input/output than using the CPU for computation – e.g. a shell

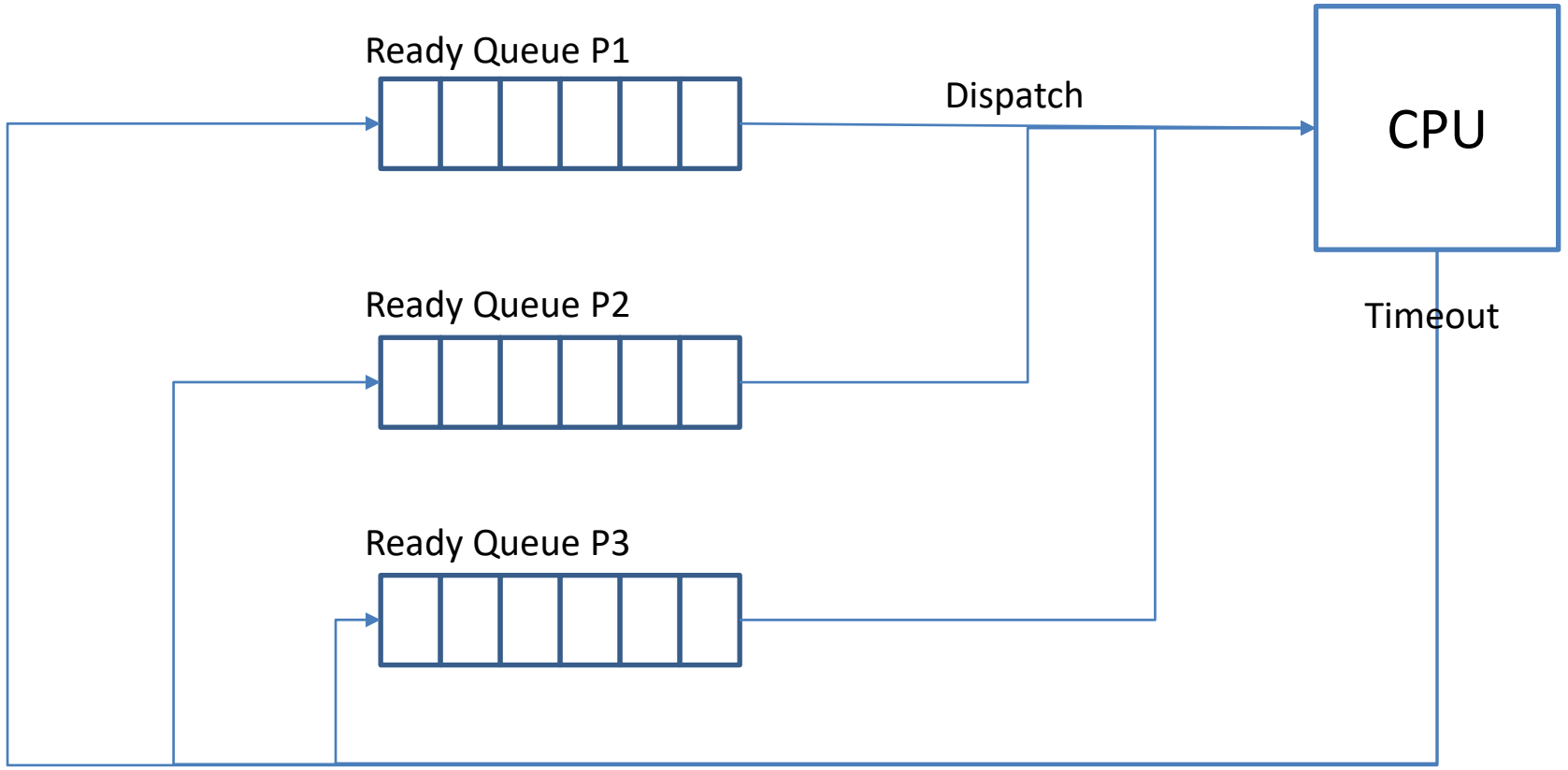
- CPU intensive – Processes that need lots of CPU cycles and don't spend much time (if any) waiting for I/O – e.g. matrix math

- What's fair scheduling when there are a mixture of I/O intensive and CPU intensive processes?

Queuing and Scheduling



Priority Scheduling



Scheduling Options

- No preemption – OS lets processes run to completion without interruption
  - First come/first served (FCFS)
    - Processes are executed in the order in which they are submitted
  - Shorted Job Next (SJN)
    - Of all processes in the ready queue, pick the one that needs the least time
- Preemption – OS interrupts processes running for some reason
  - Shorted Remaining Time (SRT)
    - At any point in time swap to the process in the ready queue that has the least amount of time remaining
  - Round Robin
    - Processes are given a time quantum
    - Removed from CPU after quantum has expired

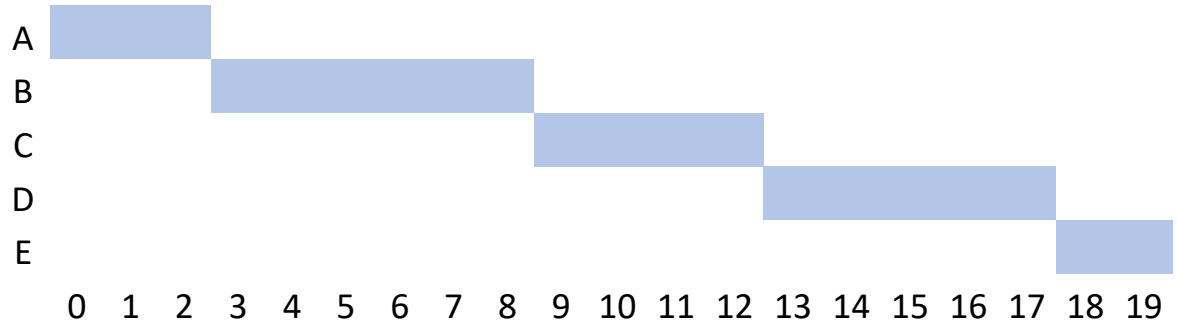
Evaluation Criteria

- Fairness
  - Turnaround time – Elapsed time from the time of submission to the time of completion
  - Wait time – The amount of time spent in the ***ready*** queue
    - NOTE: time spent in the suspended or blocked queue is not recorded
  - Response Ratio – Turnaround time / Service time
    - How many times longer did it take for the process to complete than what was required?
- CPU Utilization – Amount of time the CPU spends executing ‘useful’ work
- Throughput – Number of processes completed per unit time
- Deadline – Did process complete on or before it needed to

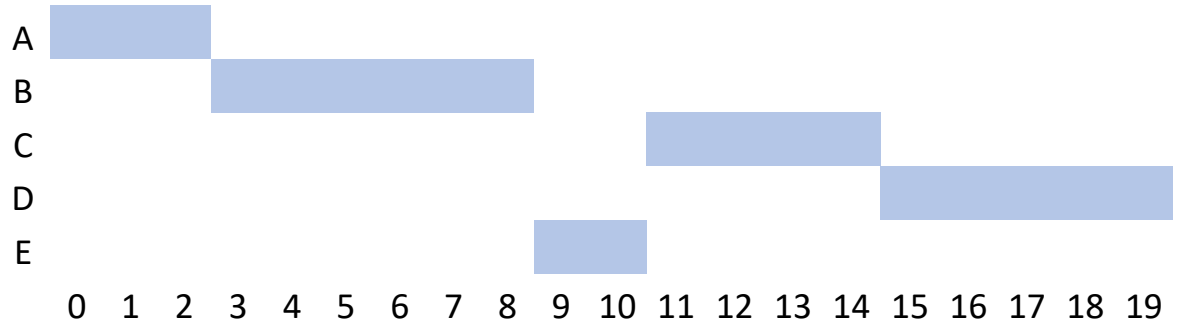
Example – No Preemption

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

FCFS



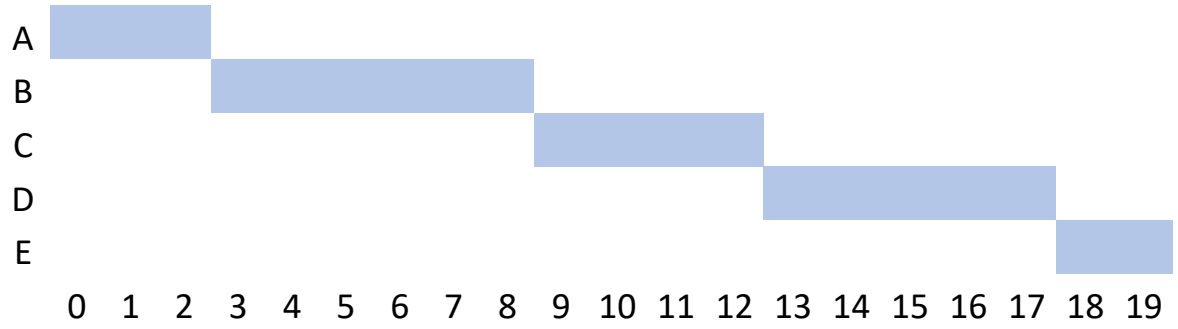
SJN



Example – No Preemption

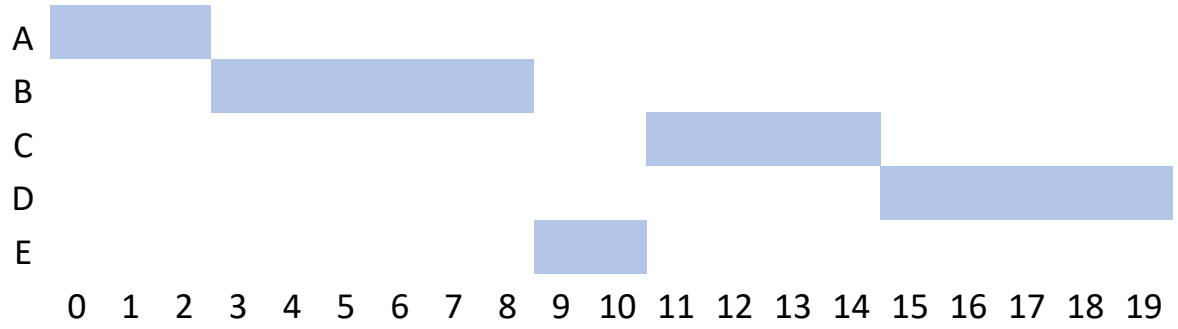
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

FCFS



FCFS	A	B	C	D	E	Avg
Finish Time	3	9	13	18	20	
Turnaround Time	3	7	9	12	12	8.6
Response Ratio	1	1.17	2.25	2.4	6	2.56

SJN

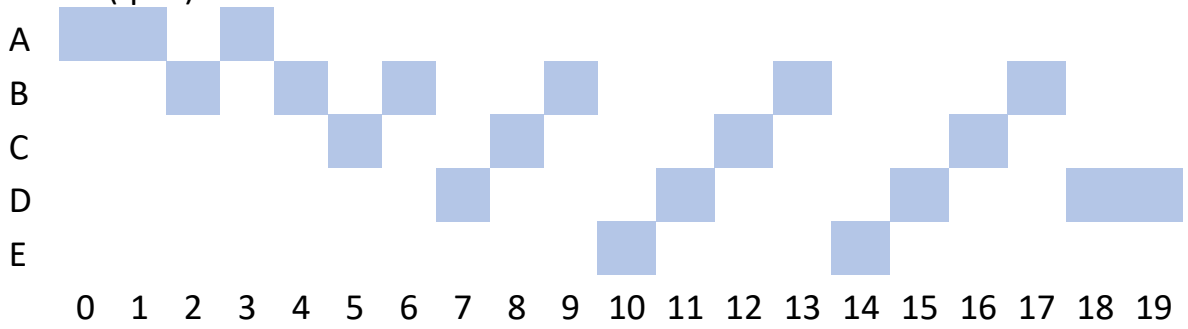


SJN	A	B	C	D	E	Avg
Finish Time	3	9	15	20	11	
Turnaround Time	3	7	11	14	3	7.6
Response Ratio	1	1.17	2.75	2.8	1.5	1.84

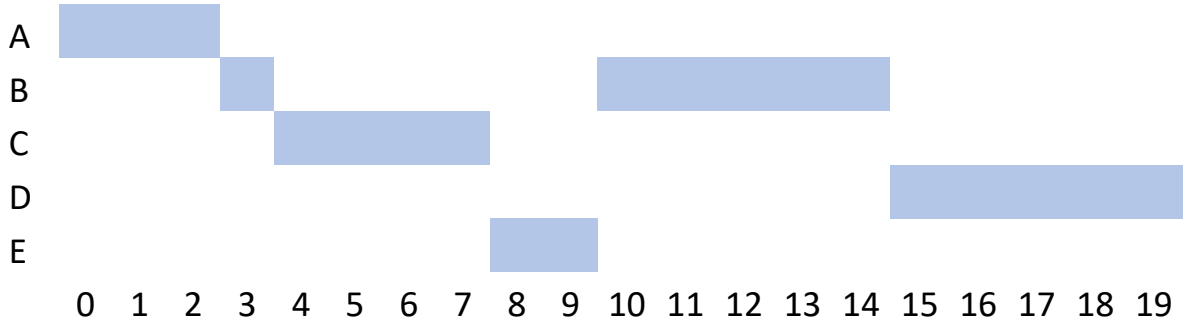
Example – Preemption

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Round Robin (q =1)



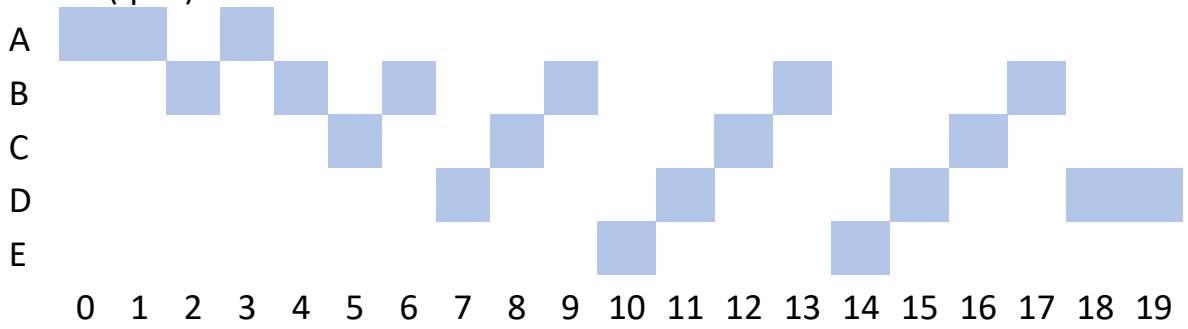
SRT



Example – Preemption

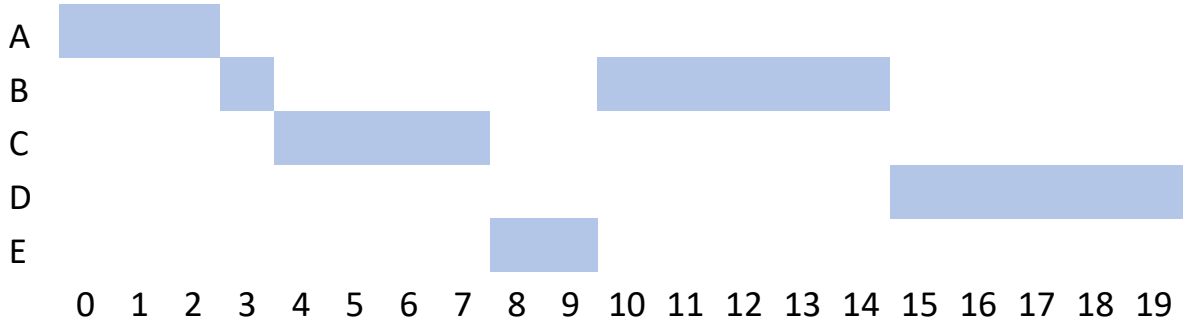
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Round Robin (q =1)



Round Robin	A	B	C	D	E	Avg
Finish Time	4	18	17	20	15	
Turnaround Time	4	16	13	14	7	10.8
Response Ratio	1	2.67	3.25	2.8	3.5	2.71

SRT



SRT	A	B	C	D	E	Avg
Finish Time	3	15	8	20	10	
Turnaround Time	3	13	4	14	2	7.2
Response Ratio	1	2.17	1	2.8	1	1.59

Preemption vs Non-Preemption

- FCFS and SJN
  - Attempts to minimize turn around time
  - Works well for CPU intensive processes
- Round Robin
  - Gives appearance of multiprogramming
  - Large time quantum
    -> low overhead
    -> poor response for multiprogramming
  - Low time quantum
    -> high overhead
    -> better response for multiprogramming
  - Optimal? Enough for user interaction
- SRT
  - Provably optimal if service time is known
- SJN and SRT
  - Often not possible to know service time