



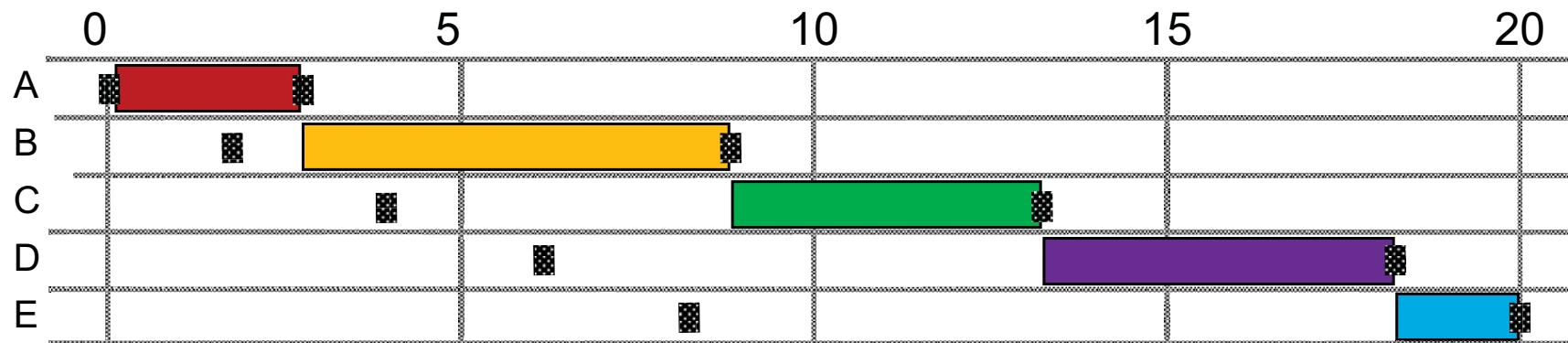
# First-Come-First-Served (FCFS)

- Simplest scheduling policy
- Also known as first-in-first-out (FIFO) or a strict queuing scheme
- When the current process ceases to execute, the process that has been in the Ready queue the longest is selected
- Non-preemptive policy
- It may be implemented by using the ready queue as a FIFO queue.
- Performs much better for long processes than short ones
- Tends to favor processor-bound processes over I/O-bound processes



# FCFS - Example

FCFS on this set of jobs:

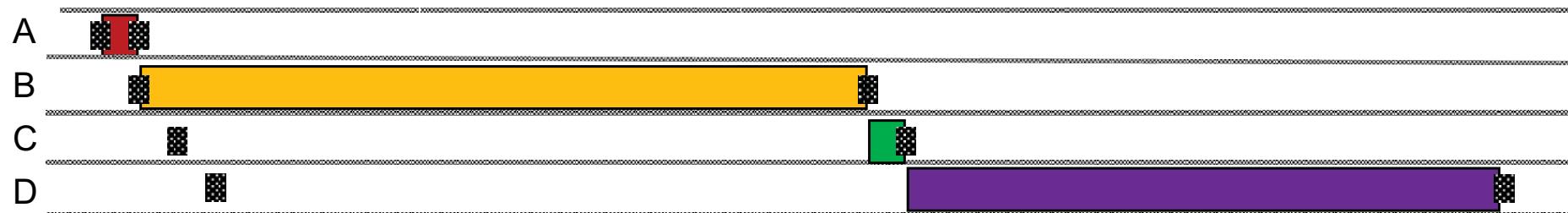


Process	A	B	C	D	E	
Arrival Time	0	2	4	6	8	
Service Time ( $T_s$ )	3	6	4	5	2	Mean
<b>FCFS</b>						
Finish Time	3	9	13	18	20	
Turnaround Time ( $T_r$ )	3	7	9	12	12	8.60
$T_r/T_s$	1.00	1.17	2.25	2.40	6.00	2.56

# FCFS Performance

- FCFS is simple but performs badly if the job mix contains jobs of widely different characteristics, for example:

process	A	B	C	D
arrival time	0	1	2	3
service time	1	100	1	100
start time	0	1	101	102
finish time	1	101	102	202
turnaround time $T_r$	1	100	100	199
$T_r/T_s$	1.0	1.0	100	2.0



- Here the long jobs (B, and D) get a reasonable turnaround time, but the short job (C) has an unreasonable turnaround time.

# FCFS Performance

38

- FCFS also has the **disadvantage** that a CPU bound job can monopolise the processor, leaving I/O devices idle.
  - Suppose there is one processor bound job and many I/O bound jobs
    - When CPU bound job runs all I/O bound jobs wait and I/O devices are idle
    - When the CPU bound job finishes, the I/O bound jobs quickly become blocked waiting and the CPU becomes idle.
    - May result in inefficient use of both the processor and the I/O devices
    - **Convoy effect**
- FCFS is not an attractive scheduling policy on its own
- FCFS is best used in **combination** with another policy