Lecture 4

Process Scheduling II

Operating Systems

Priority scheduling

- Not all jobs equal
  - So: rank them.
- Each process has a priority
  - Run highest priority ready job in system
- Priorities can be static or dynamic or both
- Among the Processes of equal priority
  - Round robin
  - FCFS

Priority scheduling

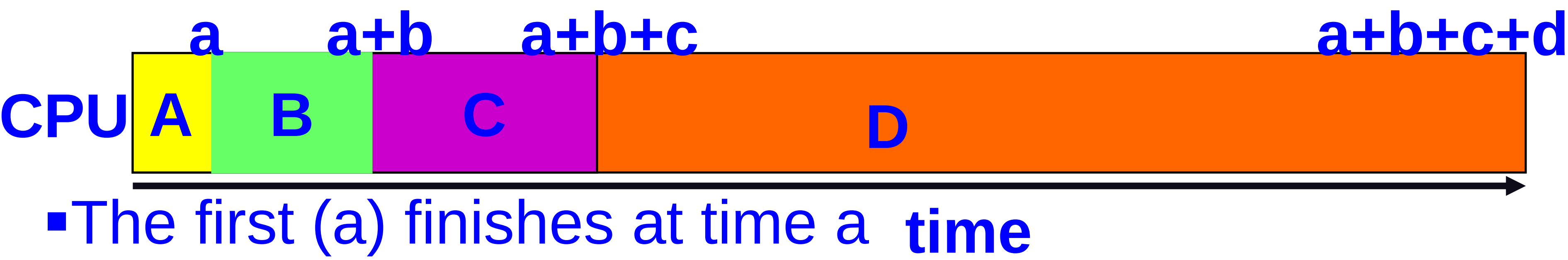
- Priority scheduling can be Preemptive or Non-Preemptive
- When a process arrives and enters the Ready Queue
- Its priority is compared with the currently Running Process
- If Higher
  - Preemptive Scheduling
    - Run the New Thread
  - Non-Preemptive Scheduling
    - Continue running the Current Thread

Priority scheduling

- High priority always runs over low priority.
- Starvation
  - A low Priority process may indefinitely wait for the CPU
- Solution: Aging
  - Gradually increase the Priority of processes that wait in the system for a long time.
- Which type of processes should be given Higher Priority:
  - I/O Bound???
  - CPU Bound???
- In order to keep I/O busy increase priority for jobs that often block on I/O

Shortest Job First (SJF)

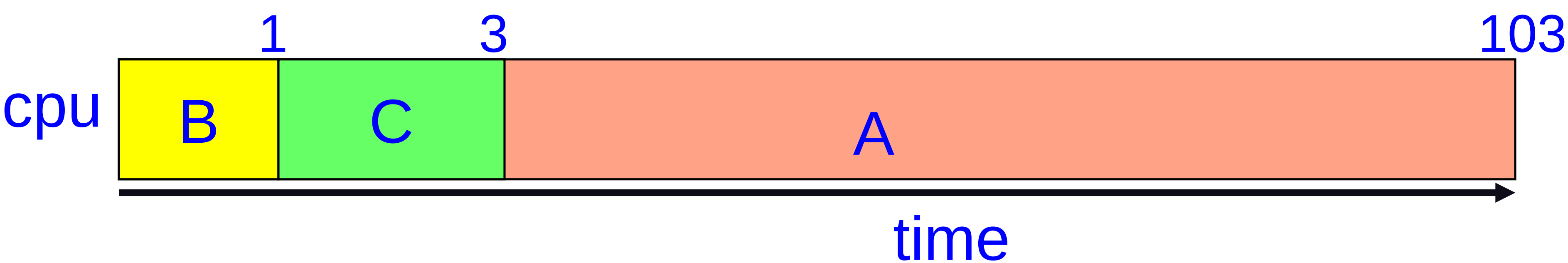
- Consider 4 jobs, a, b, c, d, run in lexical order



- The first (a) finishes at time a
- The second (b) finishes at time a+b
- The third (c) finishes at time a+b+c
- The fourth (d) finishes at time a+b+c+d
- Therefore average completion = (a + (a + b) + (a +b+c) + (a + b + c + d))/4 = (4a+3b+2c+d)/4
- Minimizing this requires a <= b <= c <= d.
- or Shortest Job First

Shortest Job First (SJF)

- Run whatever job has smallest next CPU burst
- Can be pre-emptive or non-pre-emptive
- Example: same jobs (given jobs A, B, C)

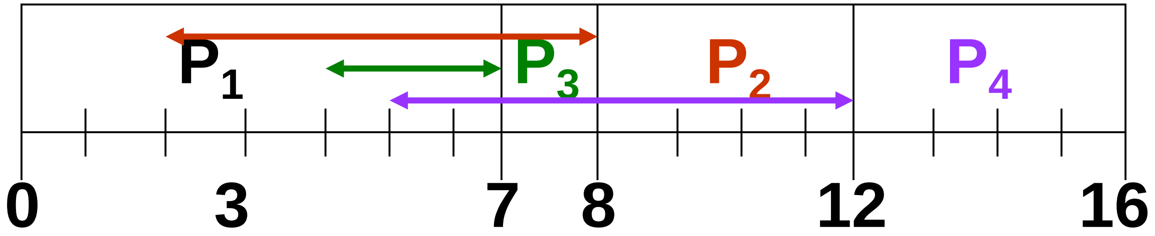


Average completion = (1+3+103) / 3 = ~35

Example of Non-Preemptive SJF

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)

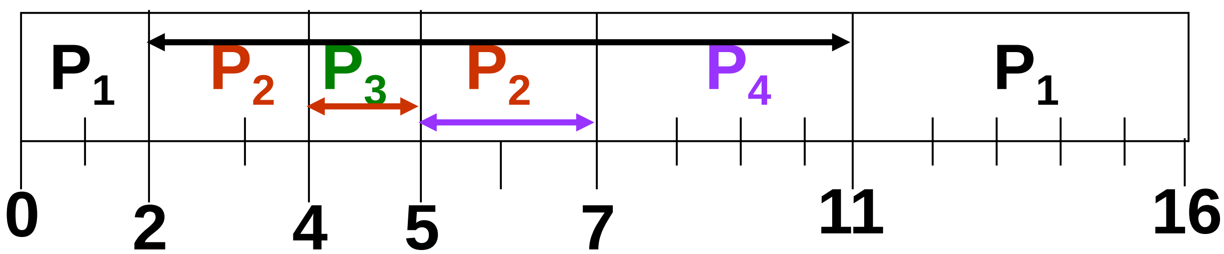


- Average waiting time
  - = (0 + 6 + 3 + 7)/4 = 4

Example of Preemptive SJF

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

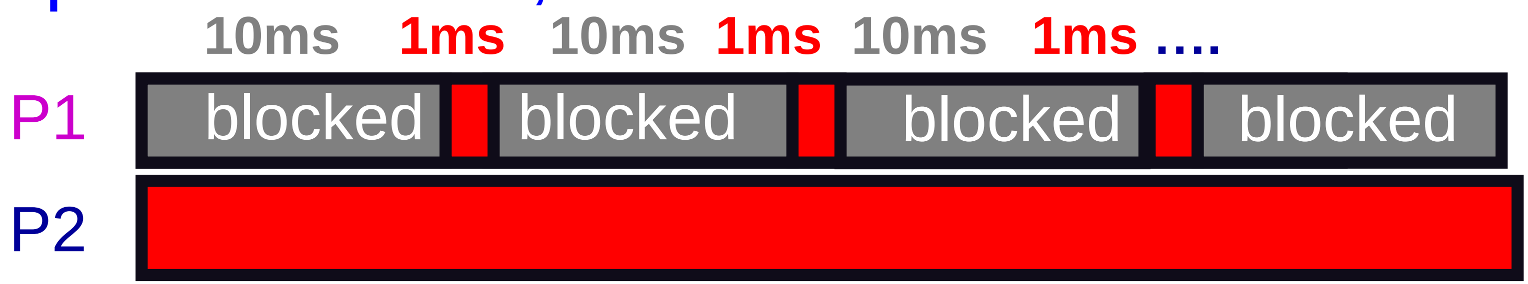
- SJF (preemptive)



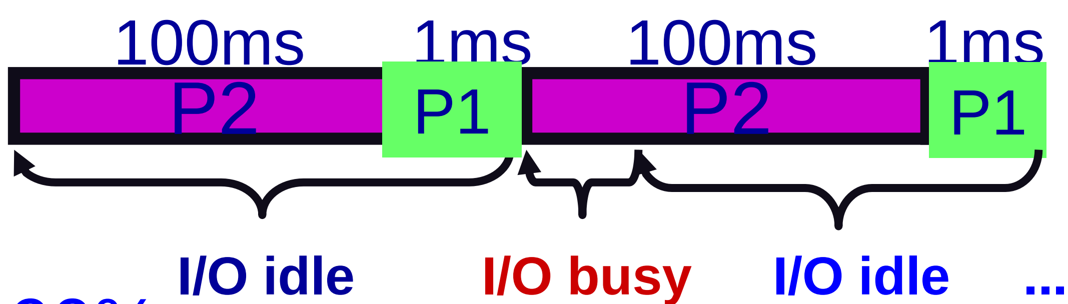
- Average waiting time
  - = (9 + 1 + 0 + 2)/4 = 3

SJF vs. RR

- Two processes P1, P2



- RR with 100ms time slice:



- I/O idle ~90%

- SJF Offers better I/O utilization

Shortest Job First

- The most important issue in SJF
  - Accuracy in estimation of Job length

Multilevel Queue Scheduling

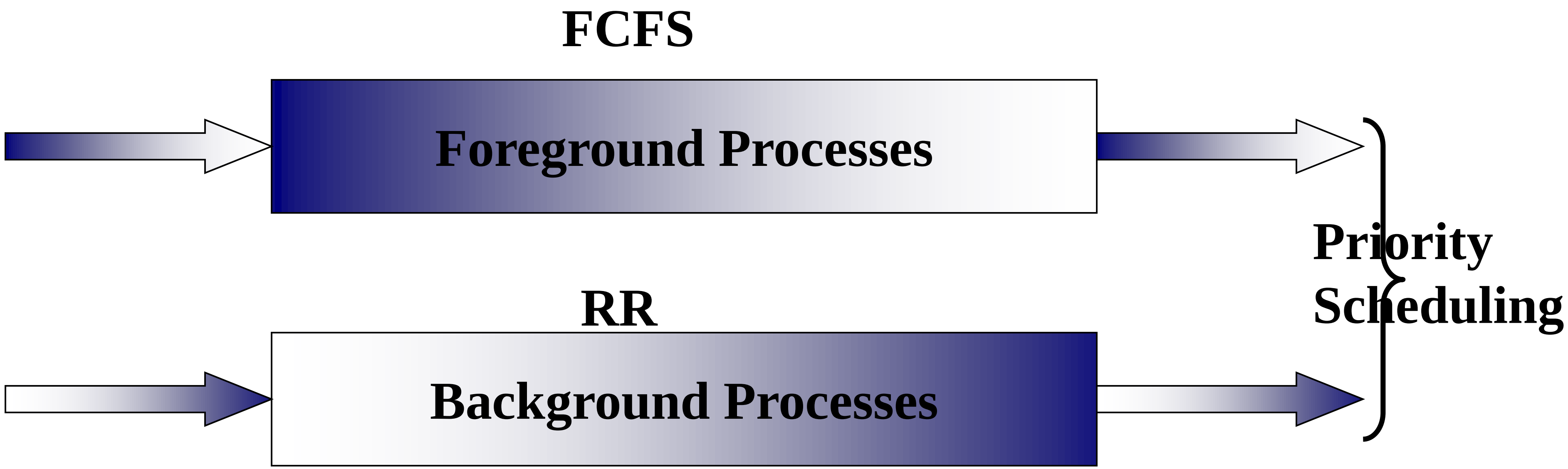
- Sometimes processes are classified into groups
- One classification can be:
  - Foreground (or Interactive) processes
  - Background (or batch) processes
- Different response time requirement
- => Different scheduling requirements
- Foreground processes usually have higher priorities

Multilevel Queue Scheduling (MQS)

- Partition the Ready queue into a number of queues
- Processes are permanently assigned to one of the queues
- Each queue may have its own scheduling algorithm
- In addition, there must be scheduling between the queues

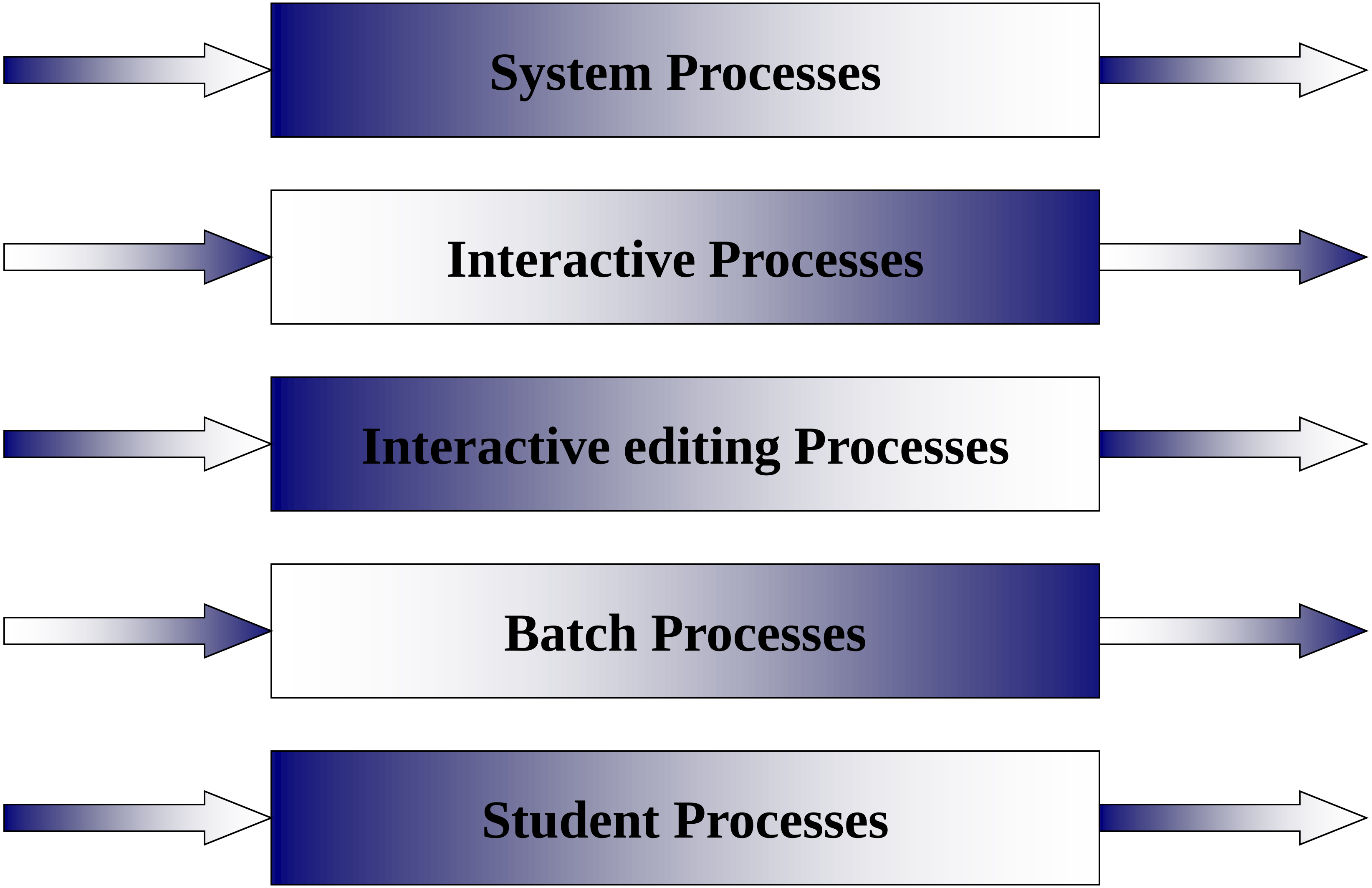
Multilevel Queue Scheduling

- Example:



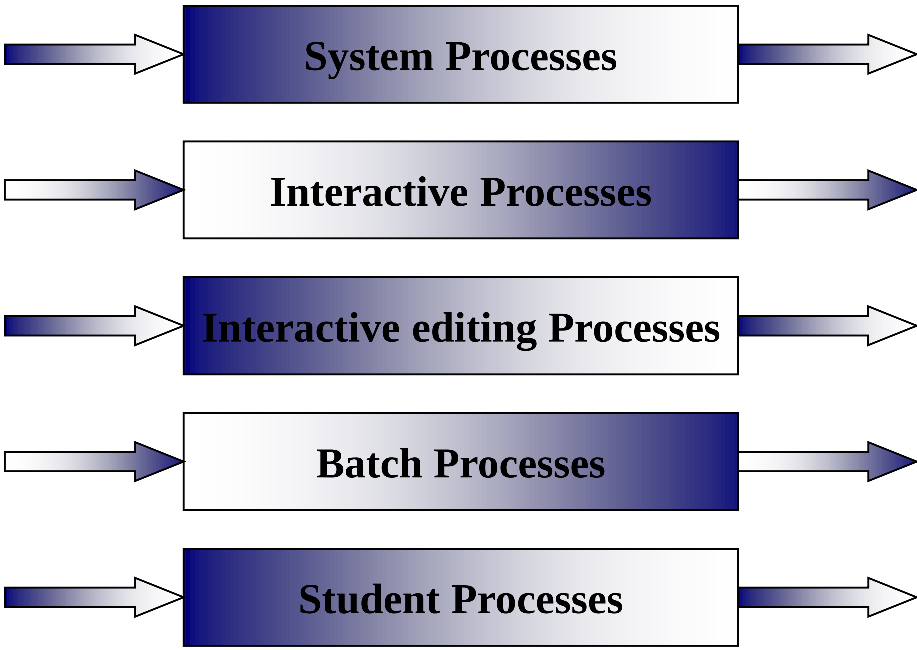
Multilevel Queue Scheduling

- Example:



Multilevel Queue Scheduling

- Each queue may have absolute priority over the other queue
- Alternatively, Time slice between the queues
  - Time slots can be equal
  - Or
    - 80% time for Foreground processes
    - 20% time for Background processes



Multilevel Feedback Queue Scheduling

- In Multilevel Queue a process is permanently assigned to a queue
- The queue to which a process should belong is decided statically
- Multilevel Feedback Queue Scheduling:
  - A Process may move between the Queues
  - Aging can be implemented this way.

Multilevel Feedback Queue Scheduling

- Multilevel-feedback-queue scheduler defined by:
  - Number of queues
  - Scheduling algorithms for each queue
  - Method used to select when upgrade process
  - Method used to select when demote process
  - Method used to determine which queue a process will enter when that process needs service

Multilevel Feedback Queue Scheduling

- Example
  - If a process used too much CPU time, then move it to a lower-priority queue
  - If a process waits too long in a lower priority queue, then move it to a higher priority queue

Multilevel Feedback Queue Scheduling

- Example: Three queues:
  - Q_0 – RR time quantum 8 milliseconds
  - Q_1 – RR time quantum 16 milliseconds
  - Q_2 – FCFS
- Scheduling
  - A new job enters queue Q_0 served by RR.
  - Then job receives 8 milliseconds.
  - If not finished in 8 milliseconds, moved to Q_1.
  - At Q_1 job served by RR.
  - Then receives 16 milliseconds.
  - If not complete, preempted and moved to Q_2.