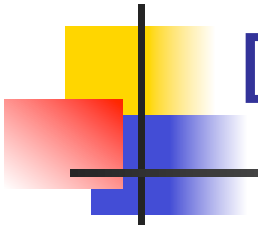
Operating Systems Design (CS 423)

Elsa L Gunter

2112 SC, UIUC

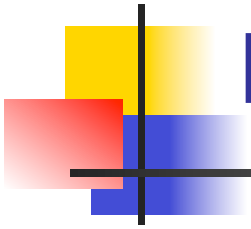
http://www.cs.illinois.edu/class/cs423/

Based on slides by Roy Campbell, Sam King, and Andrew S Tanenbaum



Deadlock

- Resources
  - Something needed by a thread
  - A thread **waits** for resources
  - E.g., locks, disk space, memory, CPU
- Deadlock
  - A circular waiting for resources leading to threads involved not being able to make progress



Deadlock

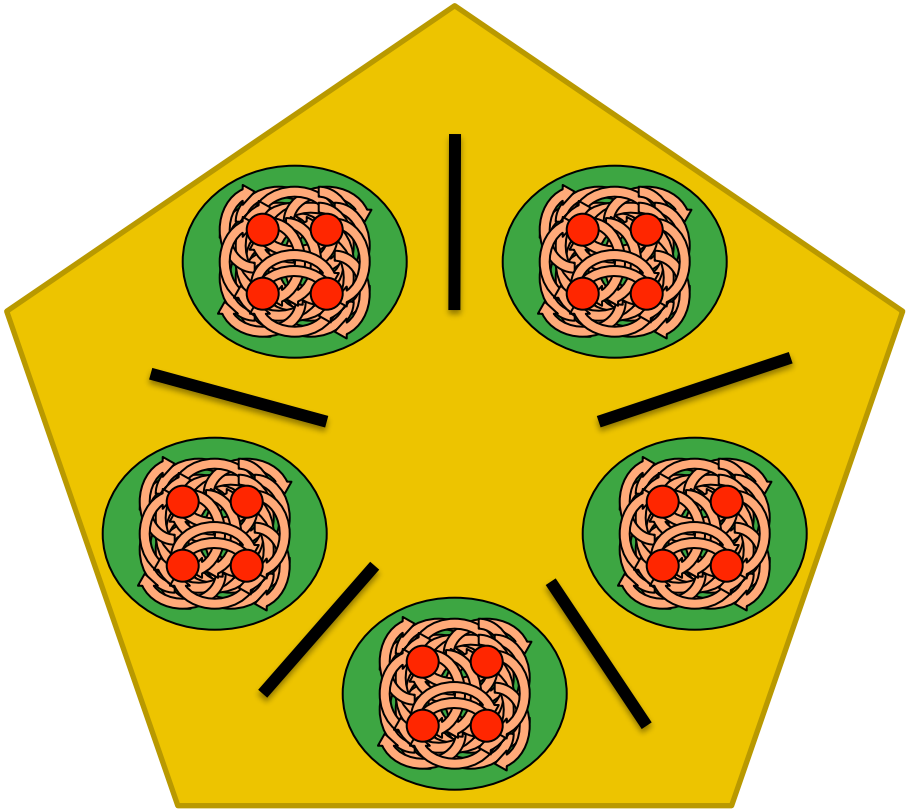
- Example:

Thread A	Thread B
`lock(x);`	`lock(y);`
`lock(y);`	`lock(x);`
…	…
`unlock(y);`	`unlock(x);`
`unlock(x);`	`unlock*(y);`

- Can deadlock occur with our code?
- Will deadlock always occur?
  - Ostrich algorithm

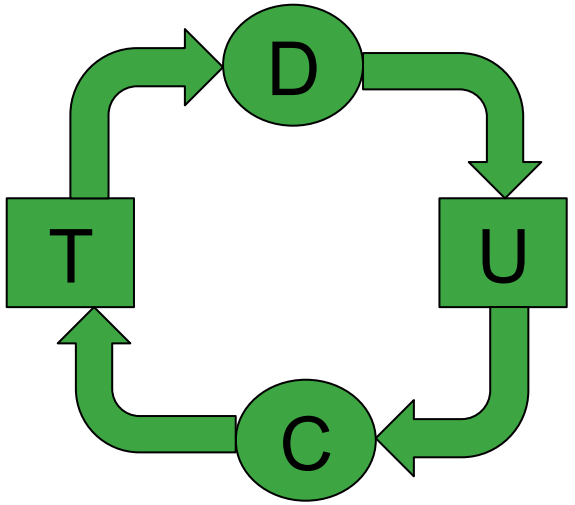
Dining Philosophers Problem

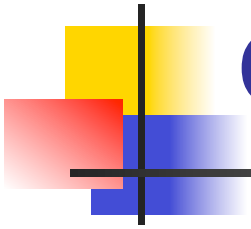
- 5 Philosophers
  - Plate for each
  - One chopstick to the left of each plate
  - Takes two chopsticks to eat
- Wait for left chopstick
- Wait for right chopstick
- Eat
- Put down chopsticks
- Deadlock?



Conditions for deadlock

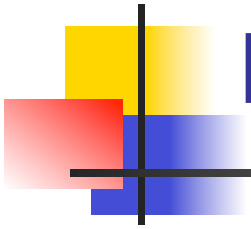
- Four conditions for deadlock
  - Limited resource
    - Only 5 chopsticks
- Hold and wait
  - Grab one chopstick at a time
- No preemption
  - Grab if you can, no guarantee
- Circular chain of requests
- Wait-for graph





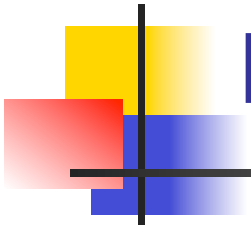
CPU scheduling

- How to choose next thread to run?
  - What are goals of CPU scheduler?
- Minimize average response time
  - Average elapsed time to do each job
- Maximize throughput of entire system
  - Rate at which jobs complete in the system
  - How do we maximize throughput?
- Fairness
  - Share CPU among threads in some "equitable" manner



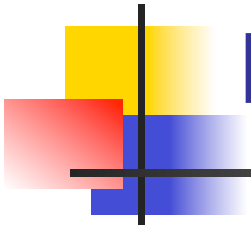
First come, first served (FCFS)

- No preemption (run until done)
  - Thread runs until it calls yield() or blocks
  - No timer interrupts
- Pros
  - + simple
- Cons
  - - short jobs get stuck
  - - bad for interactive use



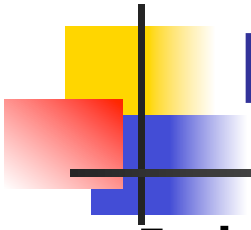
FCFS example

- Job a takes 100 seconds
- Job b takes 1 second
- Time 0: job a arrives and starts
- Time 0+: job b arrives
- Time 100: job a ends; job b starts
- Time 101: job b ends
- Average response time = (100 + 101)/2 = 100.5



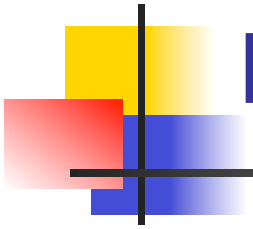
Round robin

- Goal: improve average response time for short jobs
- Solution: periodically preempt running job, let next go
- Is FCFS or round robin more “fair”?



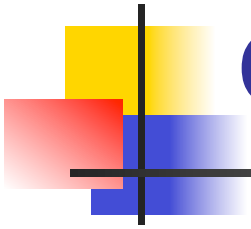
Round Robin Example

- Job a takes 100 seconds
- Job b takes 1 second
- Time slice = 1 sec
- Time 0: job a arrives and starts
- Time 0+: job b arrives
- Time 1: job b preempted; job a runs
- Time 2: job b ends; job a resumes
- Time 101: job a ends
- Average response time = (2 + 101)/2 = 51.5



Round robin

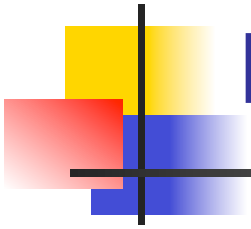
- Does round-robin always achieve lower response time than FCFS?



Choosing a time slice

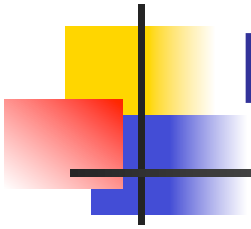
- Big time slice: degrades to FCFS
- Small time slice: each context switch too expensive

- Typical compromise 10 ms or 1ms
- If context switch takes .1ms, round robin with 10ms time slice wastes 1% of CPU time



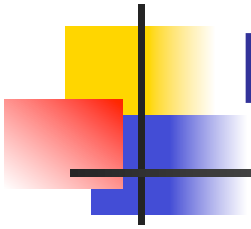
Real time scheduling

- Question is can task finish by deadline?
- Worst case analysis instead of average case analysis
- About determinism, not performance
  - Locking cache lines
- Key question: Can given tasks be scheduled and guaranteed to complete by given deadline
- Soft real time and hard real time
  - Used in multimedia systems, embedded and control systems, etc.



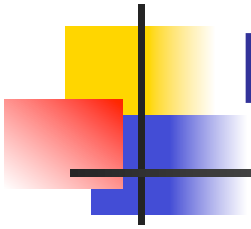
Real-time scheduling

- How do you schedule?



Earliest deadline first (EDF)

- Always run the job that has the earliest deadline
  - Dynamically adjusts the priority
- If new job arrives with earlier deadline than current job, preempt
- EDF is optimal --- we'll prove this later

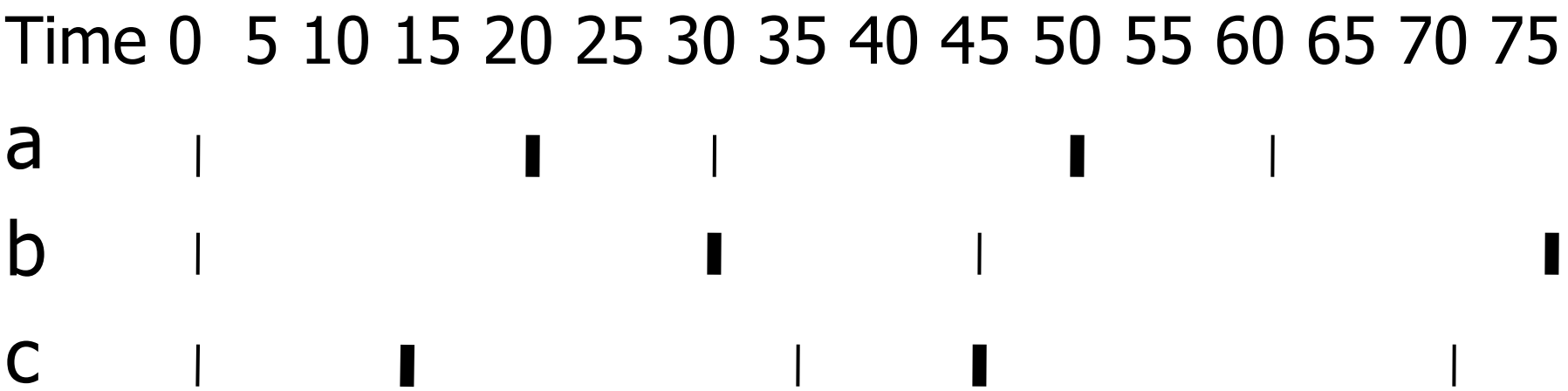


Definitions

- Tasks - unit of work
- Periodic and aperiodic tasks
- Arrival time - time when task can begin execution
- Deadline – time by which task must be done
  - Can be relative or absolute
- Execution time - amount of time task needs to finish work

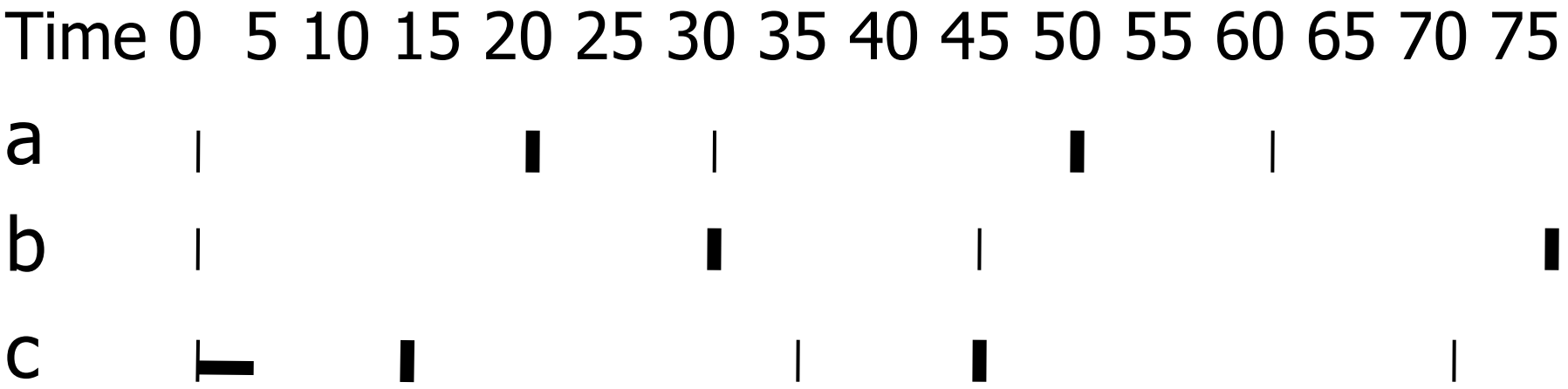
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



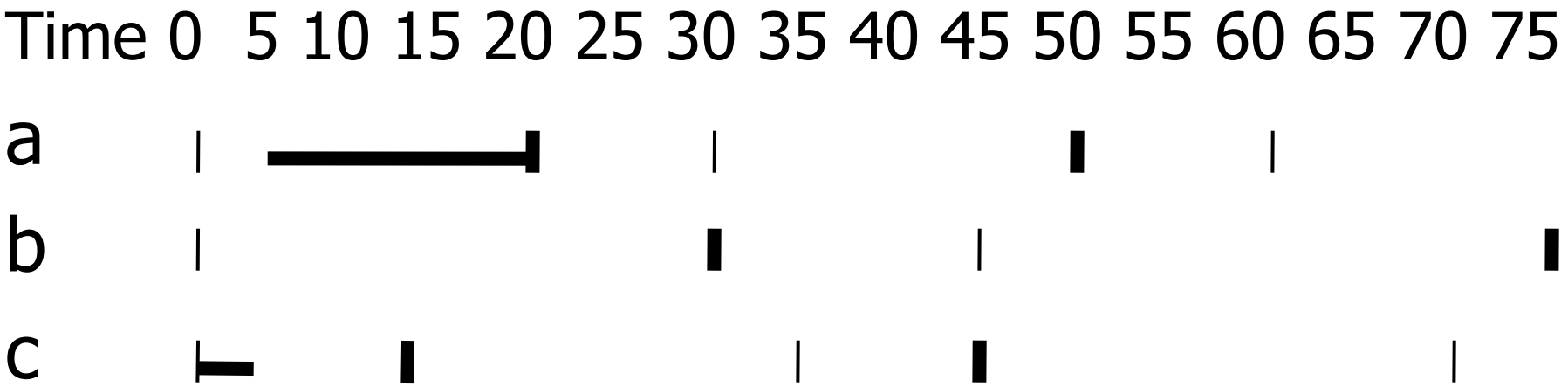
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



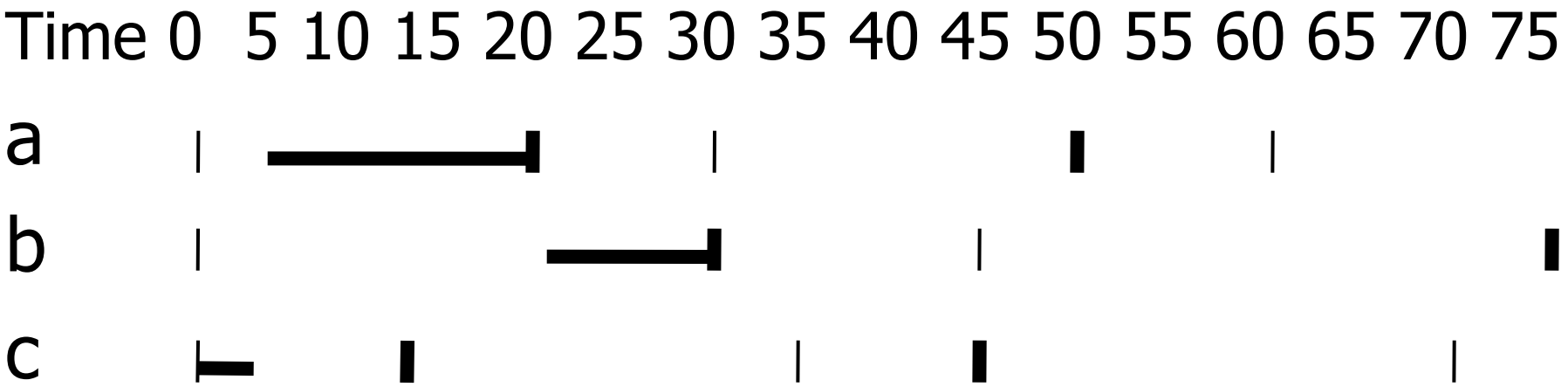
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



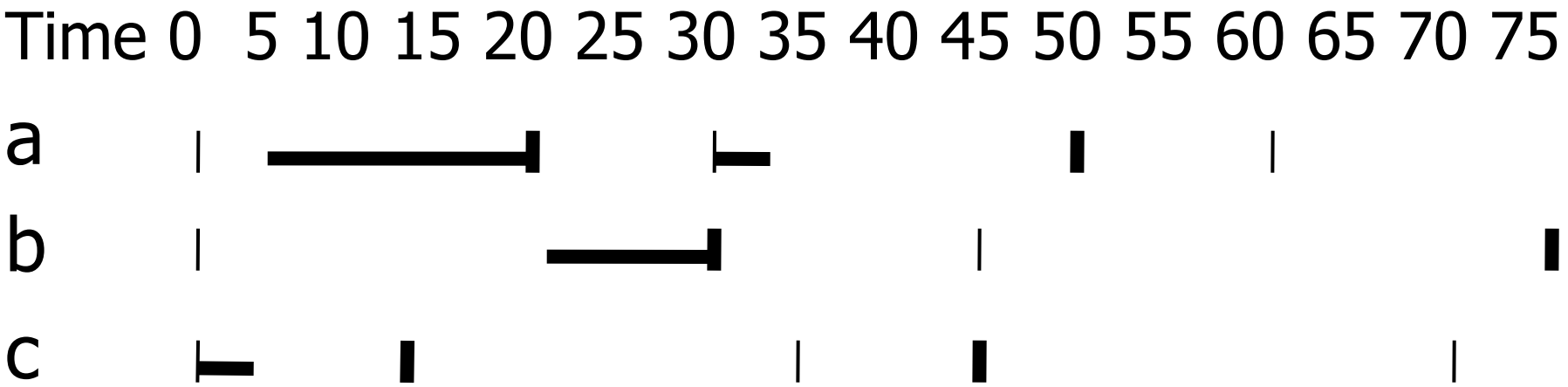
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



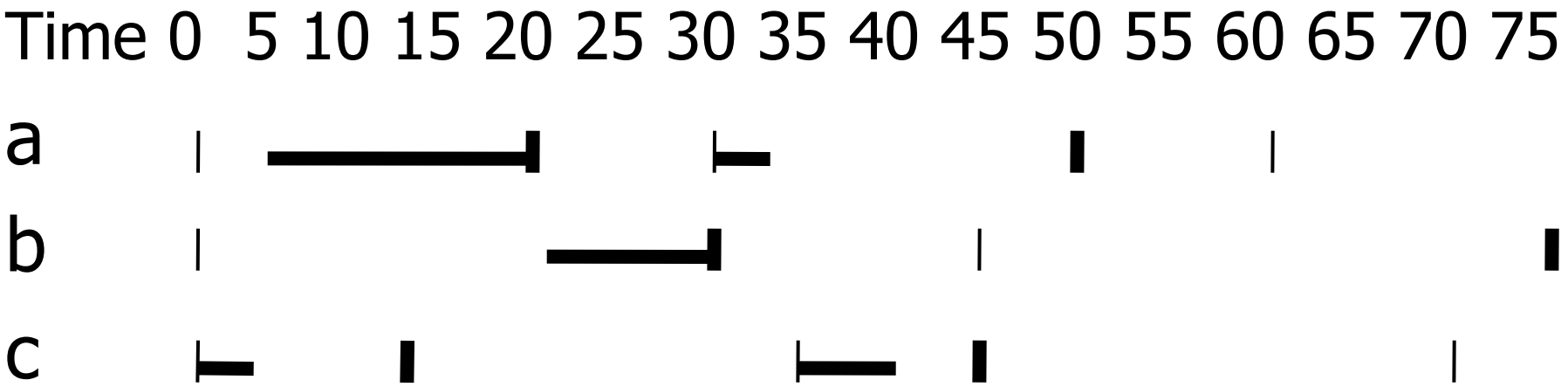
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



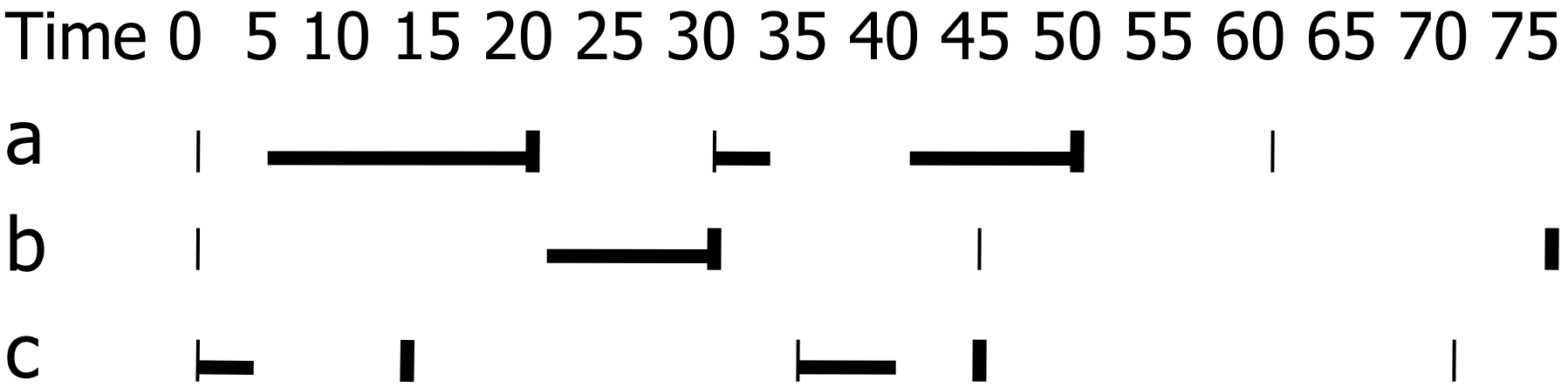
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



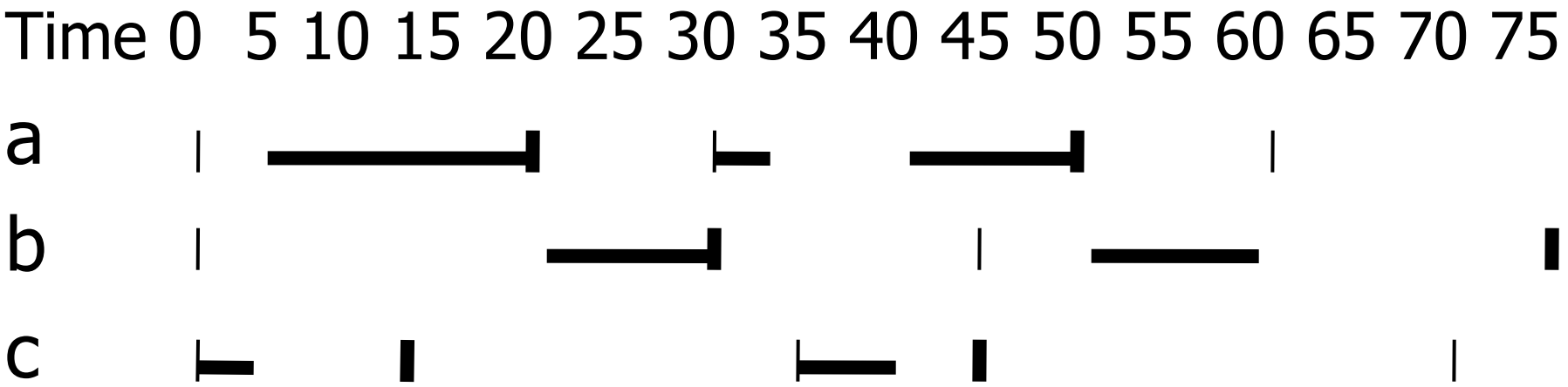
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



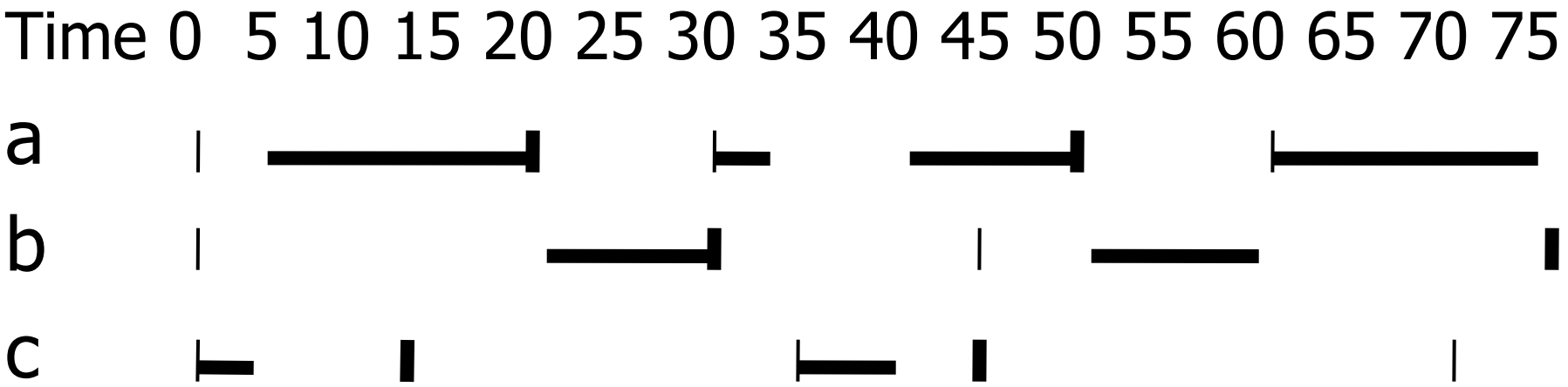
EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)



EDF example

- Job a: e(a) = 15, D(a) = 20 (period 30)
- Job b: e(b) = 10, D(b) = 30 (period 45)
- Job c: e(c) = 5, D(c) = 10 (period 35)