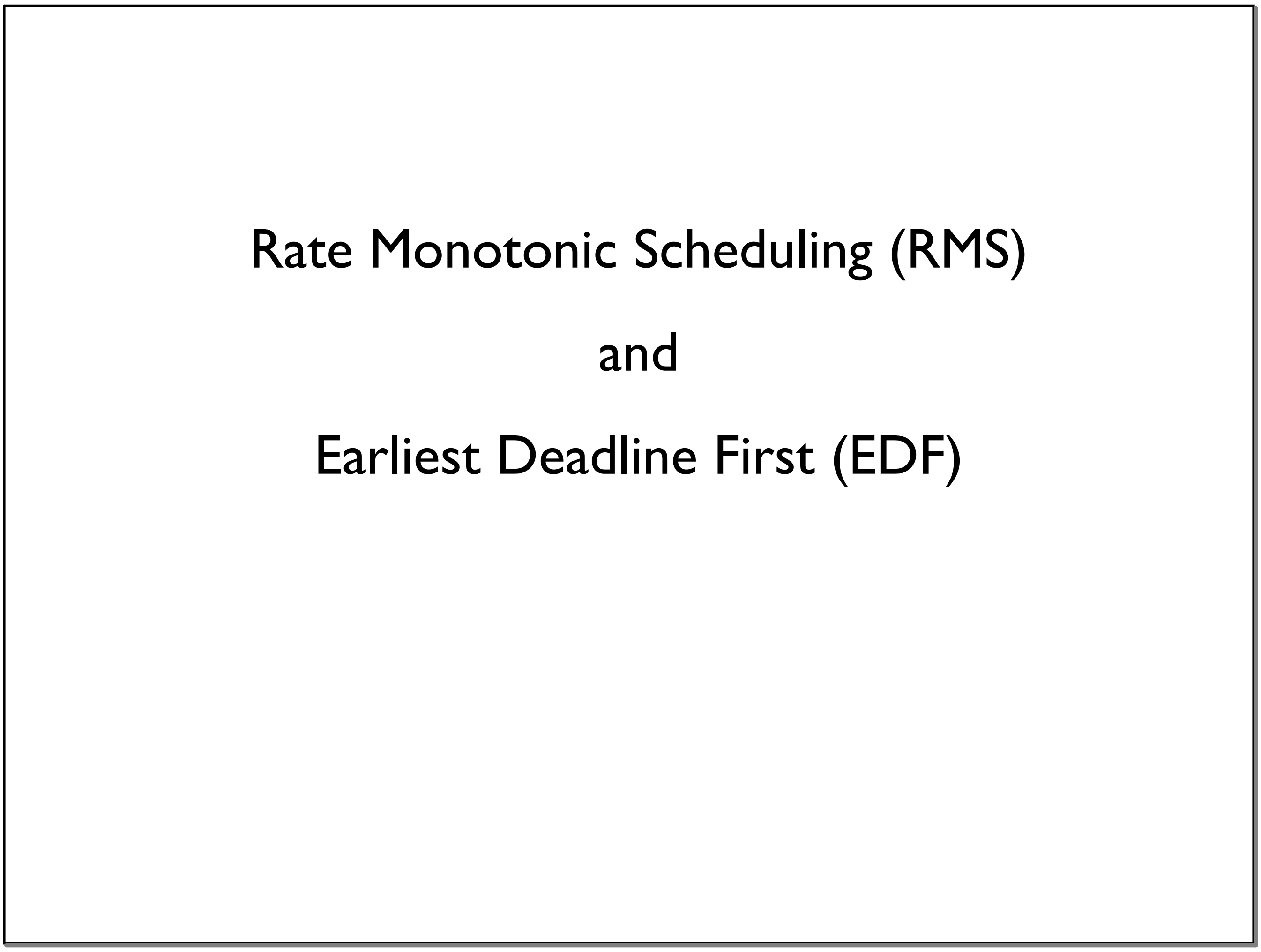
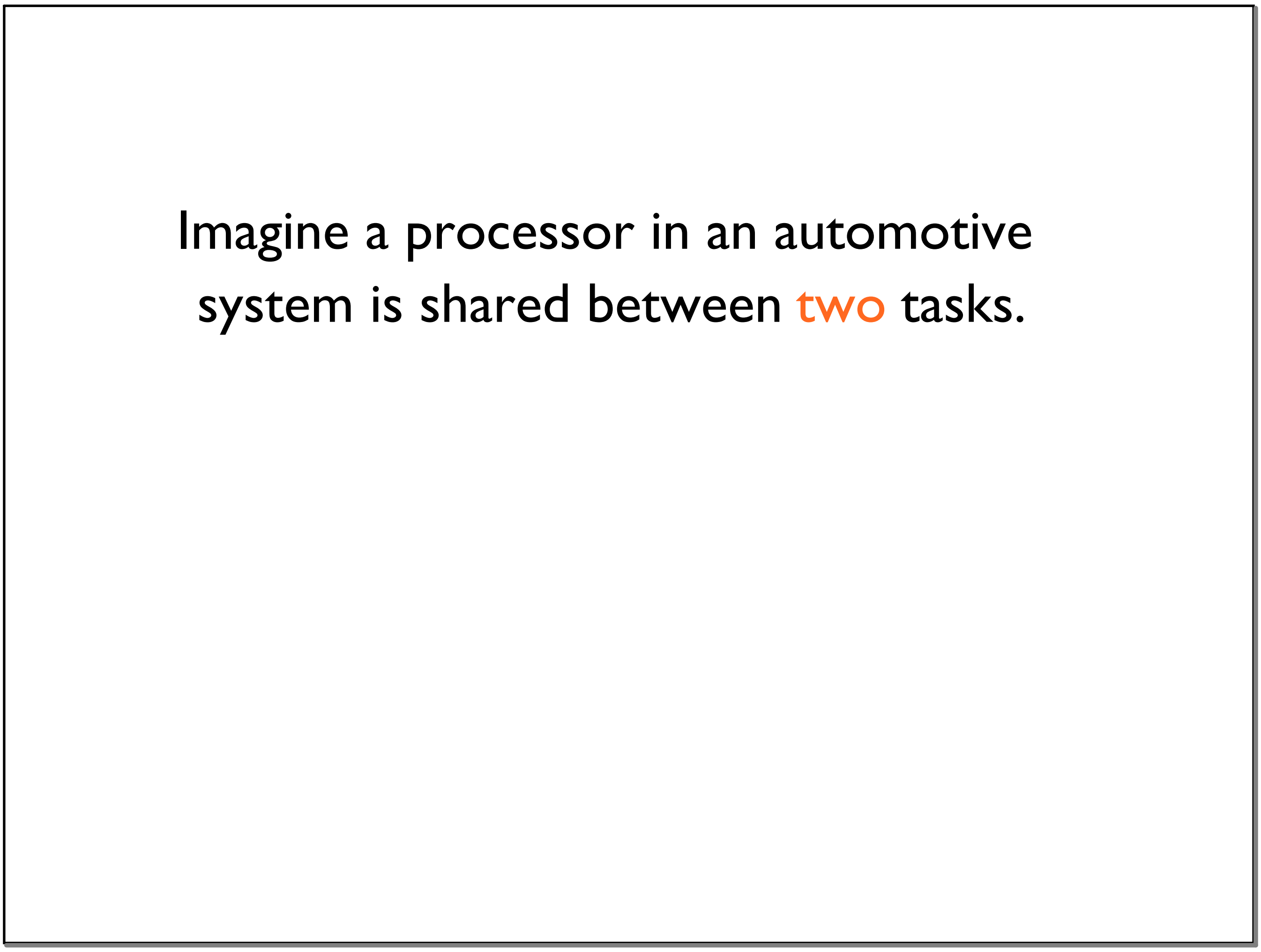


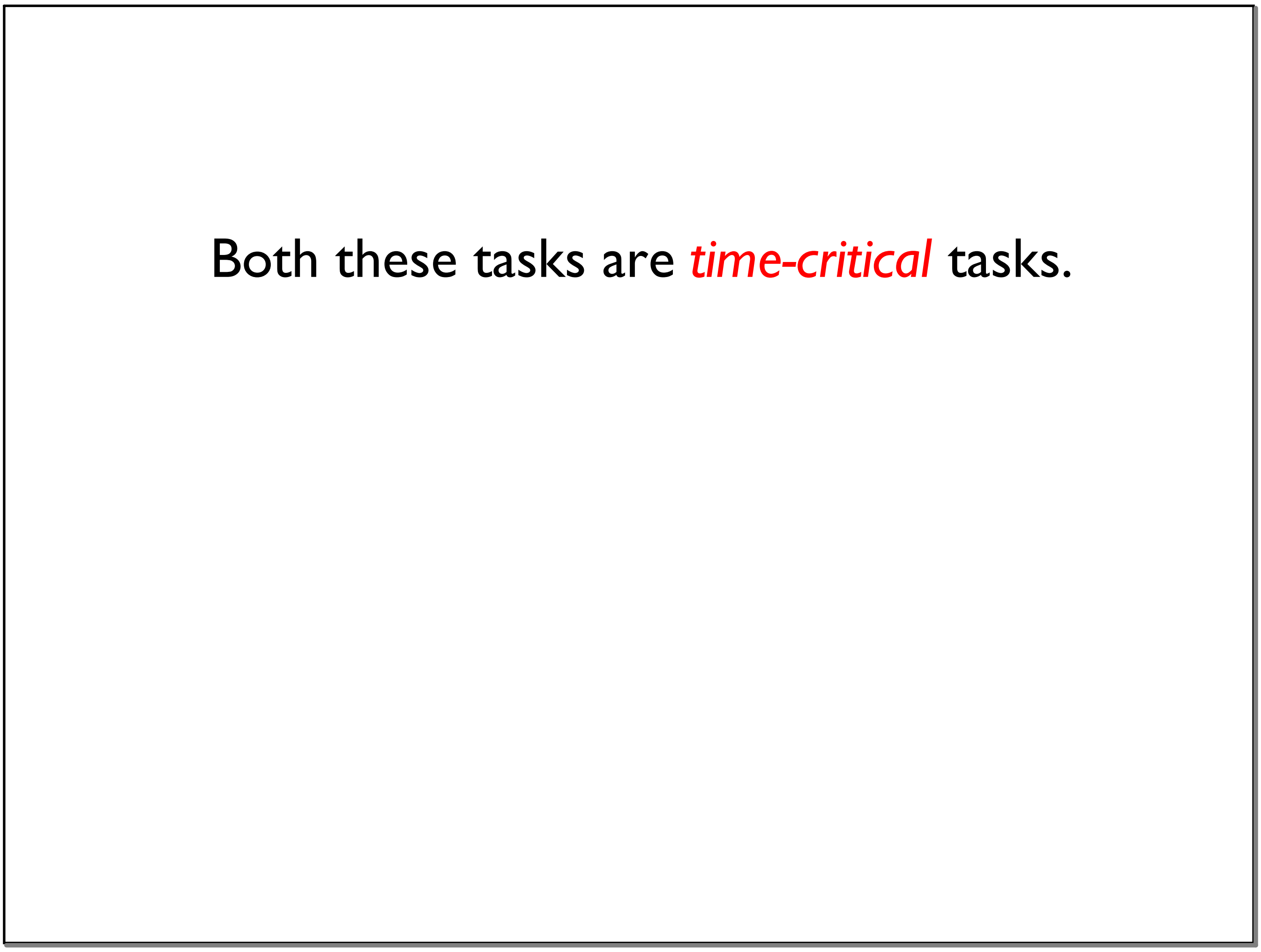
Rate Monotonic Scheduling (RMS)

and

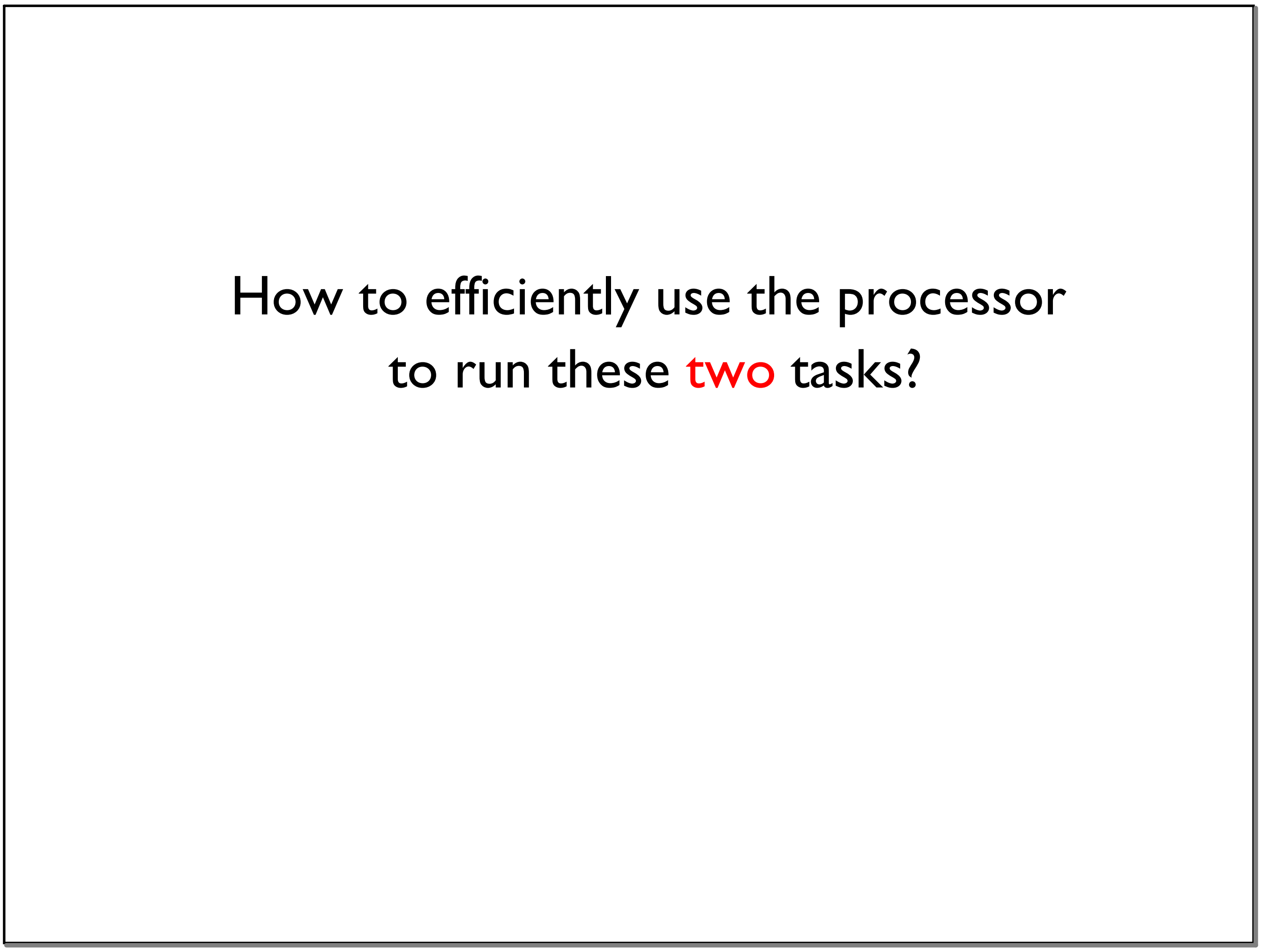
Earliest Deadline First (EDF)



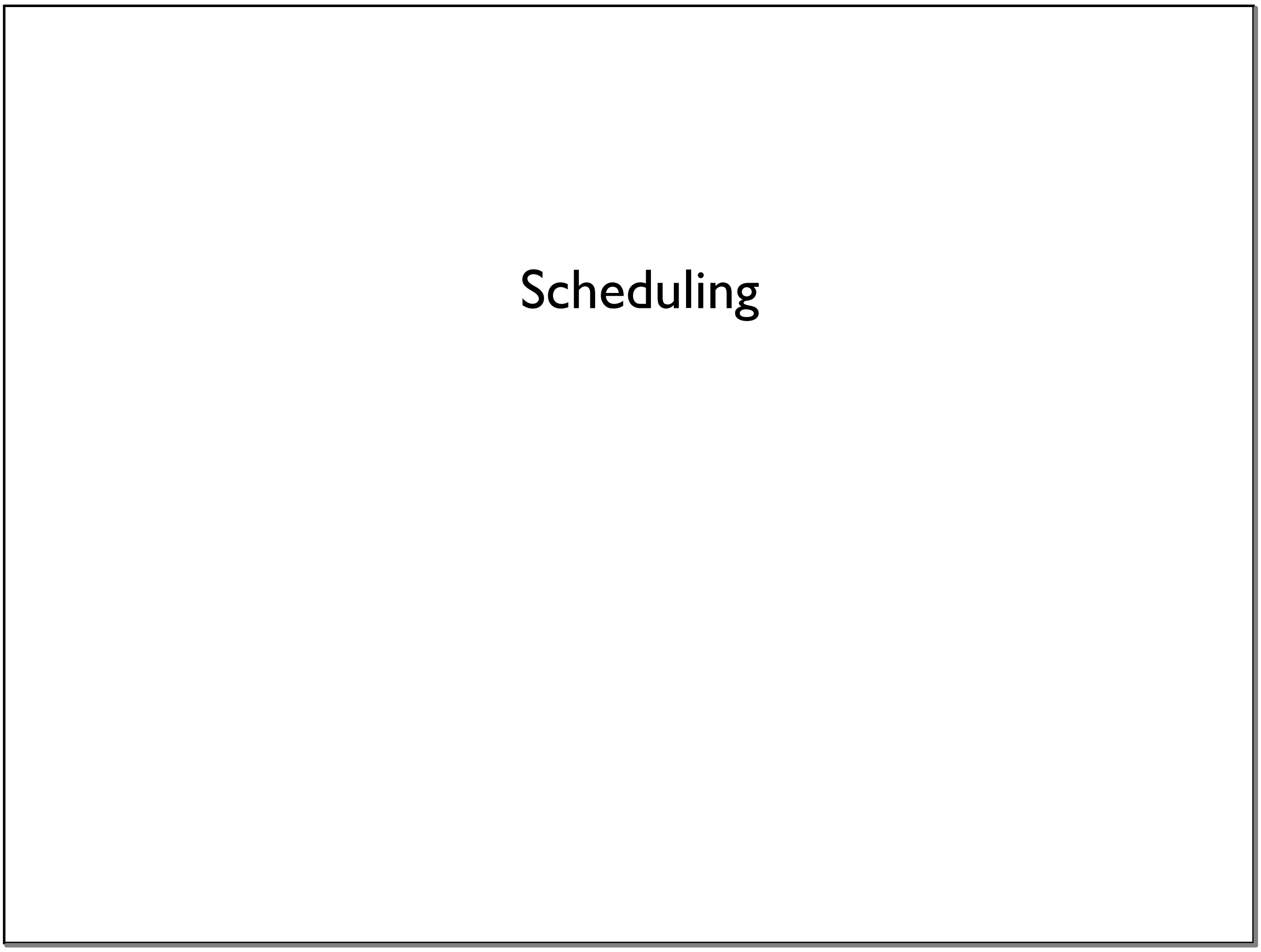
Imagine a processor in an automotive system is shared between two tasks.



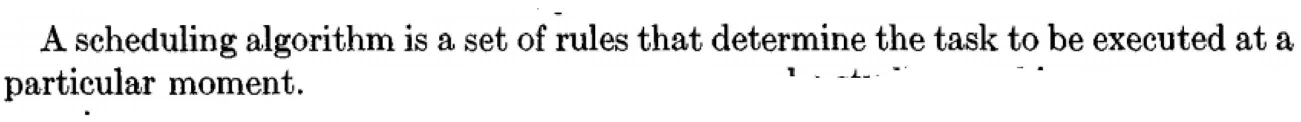
Both these tasks are *time-critical* tasks.



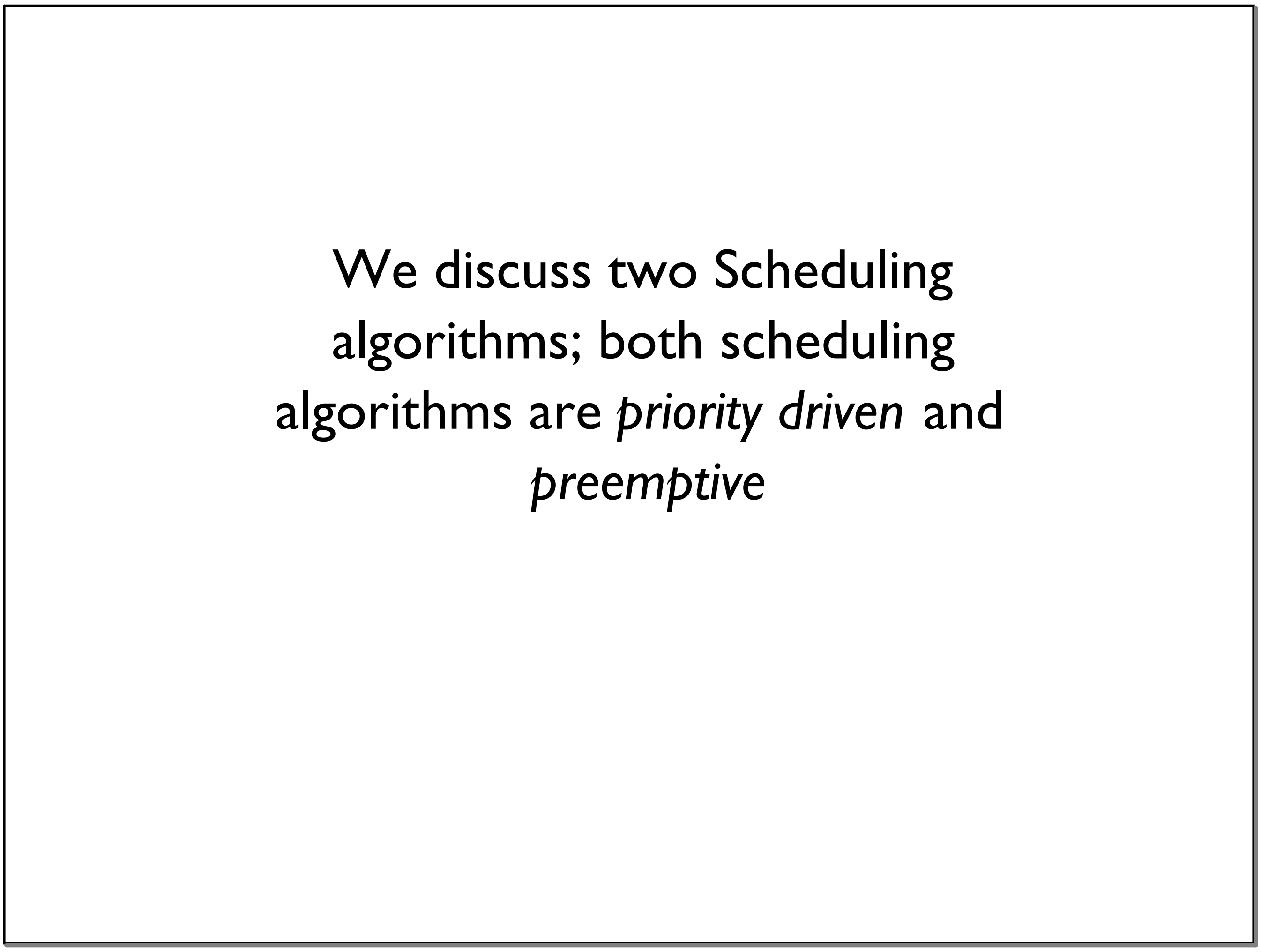
How to efficiently use the processor
to run these two tasks?



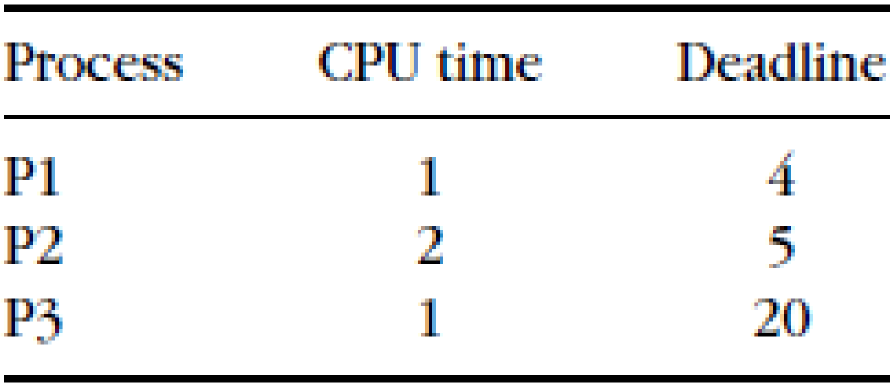
Scheduling



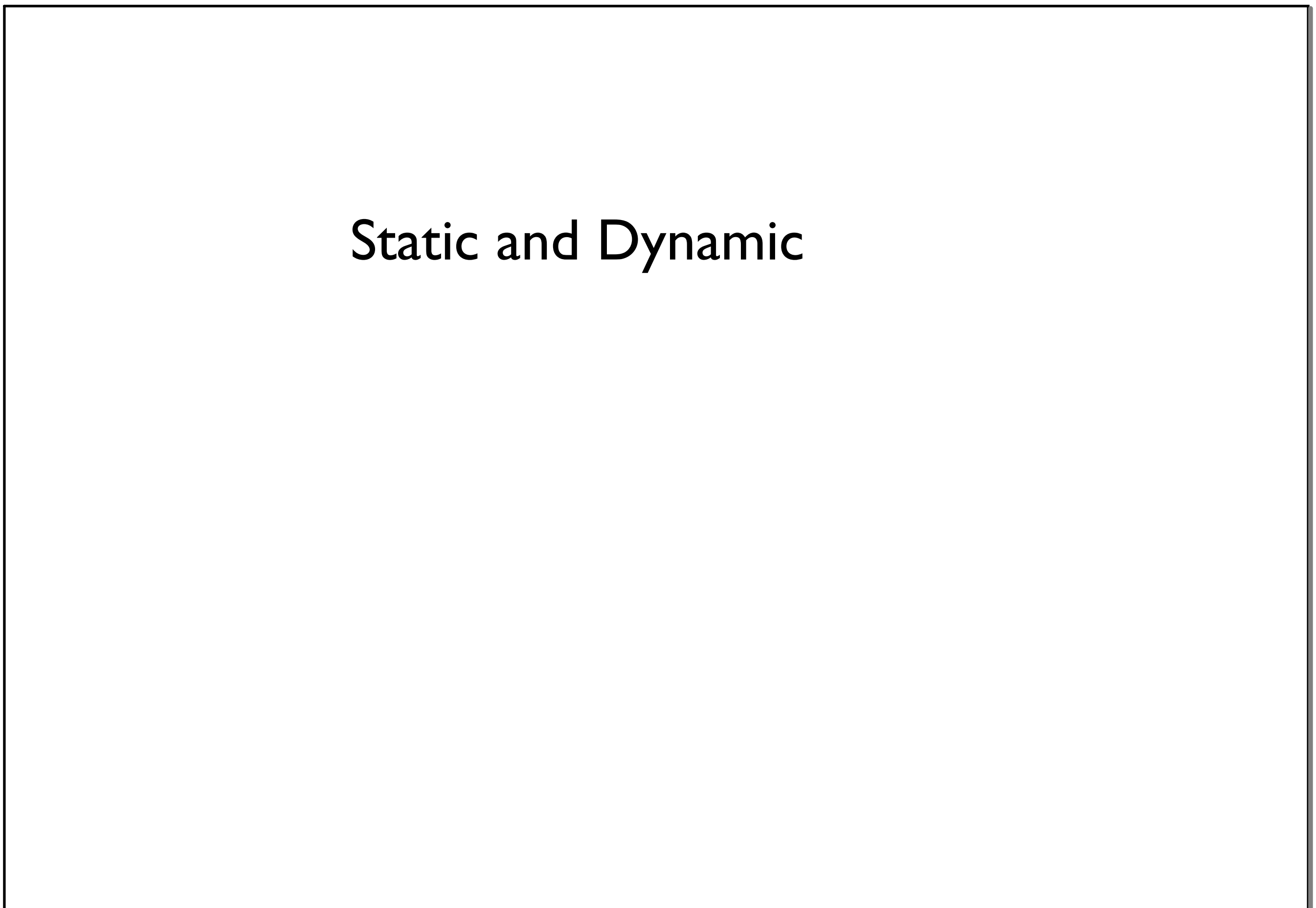
A scheduling algorithm is a set of rules that determine the task to be executed at a particular moment.



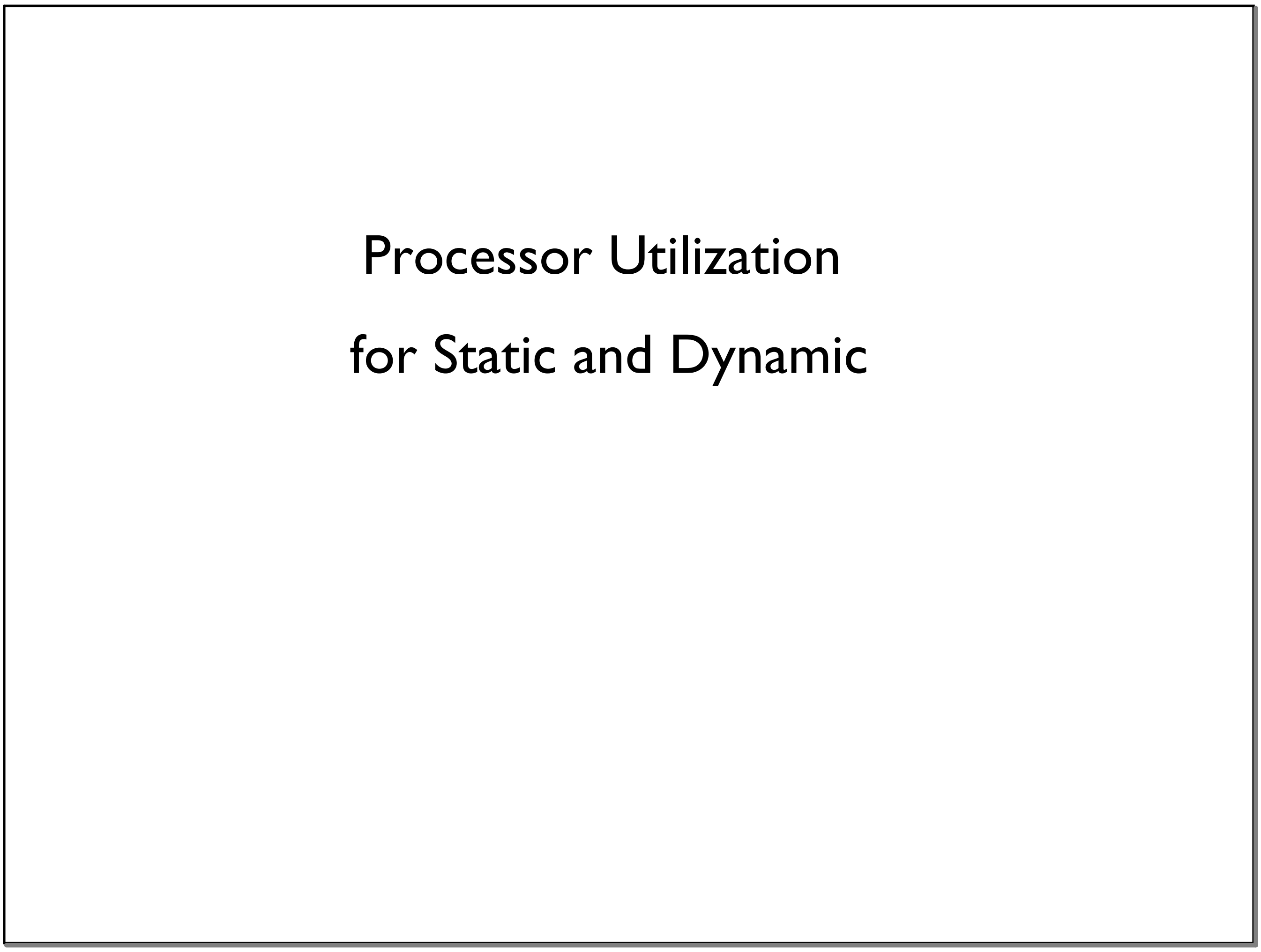
We discuss two Scheduling algorithms; both scheduling algorithms are *priority driven* and *preemptive*



Process	CPU time	Deadline
P1	1	4
P2	2	5
P3	1	20



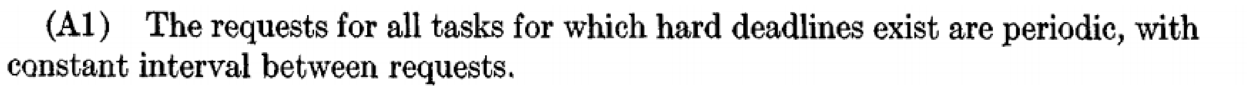
Static and Dynamic



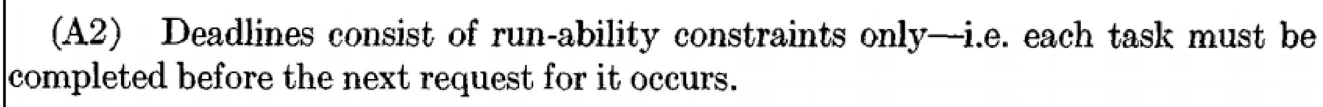
Processor Utilization

for Static and Dynamic

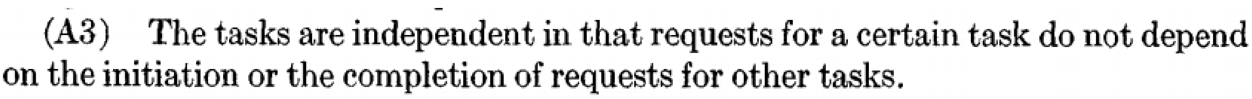
Static Priority Assumptions



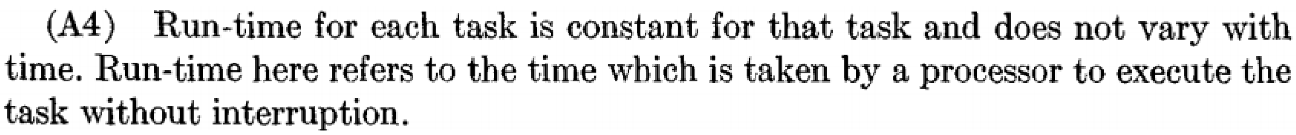
(A1) The requests for all tasks for which hard deadlines exist are periodic, with constant interval between requests.



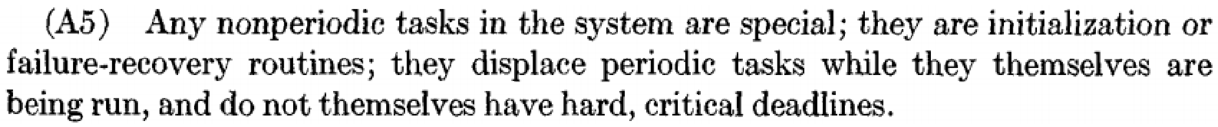
(A2) Deadlines consist of run-ability constraints only—i.e. each task must be completed before the next request for it occurs.



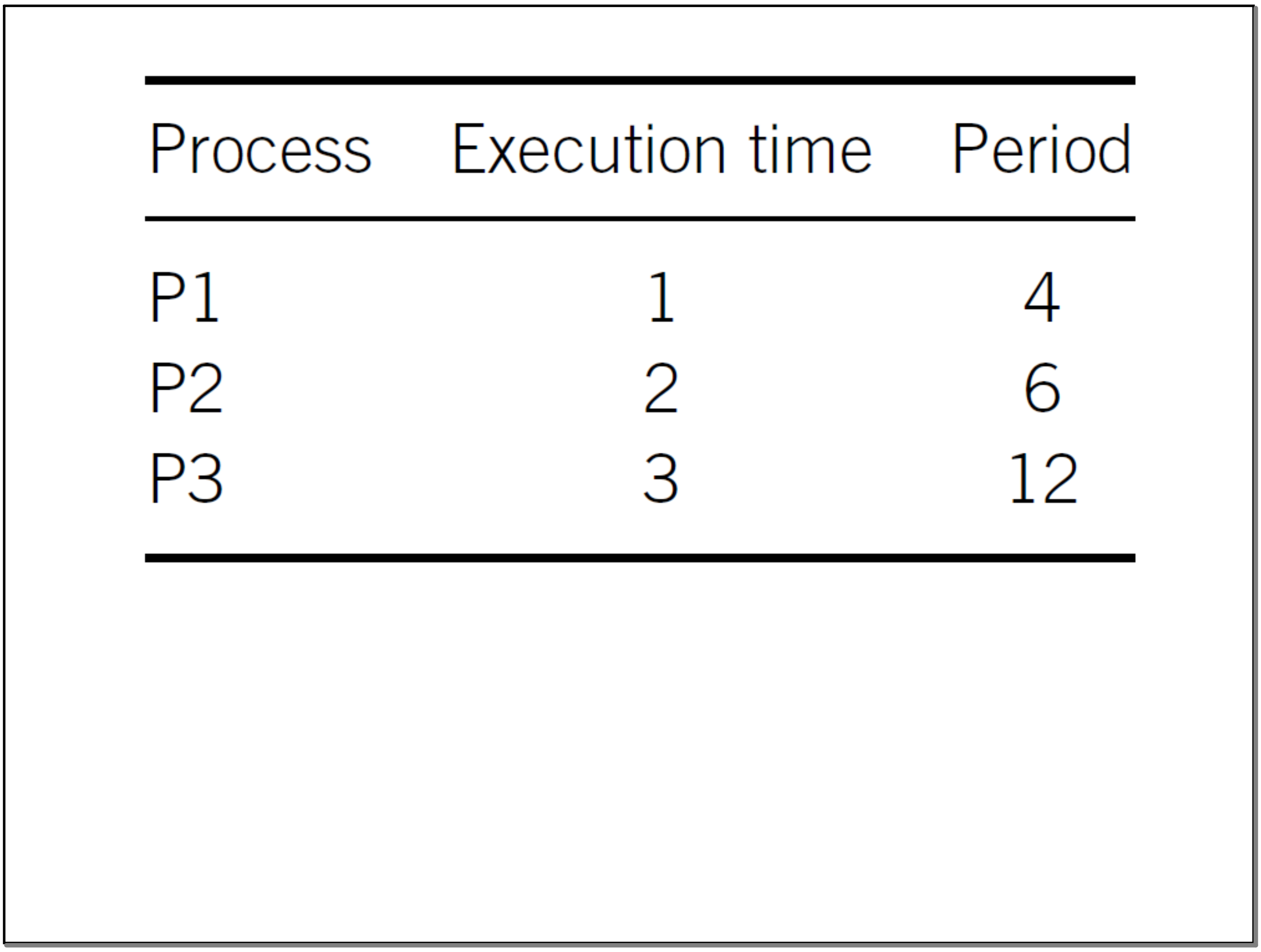
(A3) The tasks are independent in that requests for a certain task do not depend on the initiation or the completion of requests for other tasks.



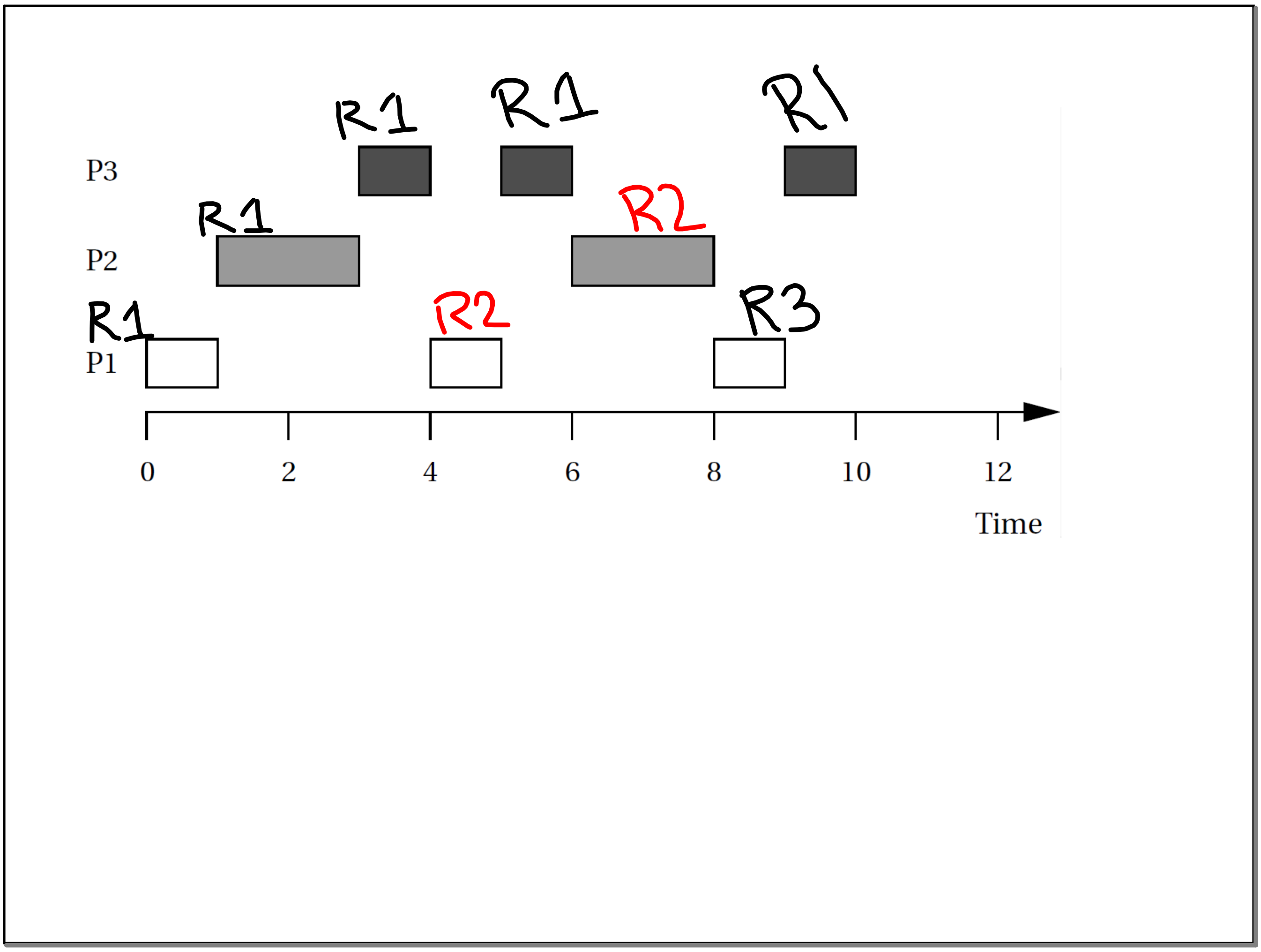
(A4) Run-time for each task is constant for that task and does not vary with time. Run-time here refers to the time which is taken by a processor to execute the task without interruption.



(A5) Any nonperiodic tasks in the system are special; they are initialization or failure-recovery routines; they displace periodic tasks while they themselves are being run, and do not themselves have hard, critical deadlines.

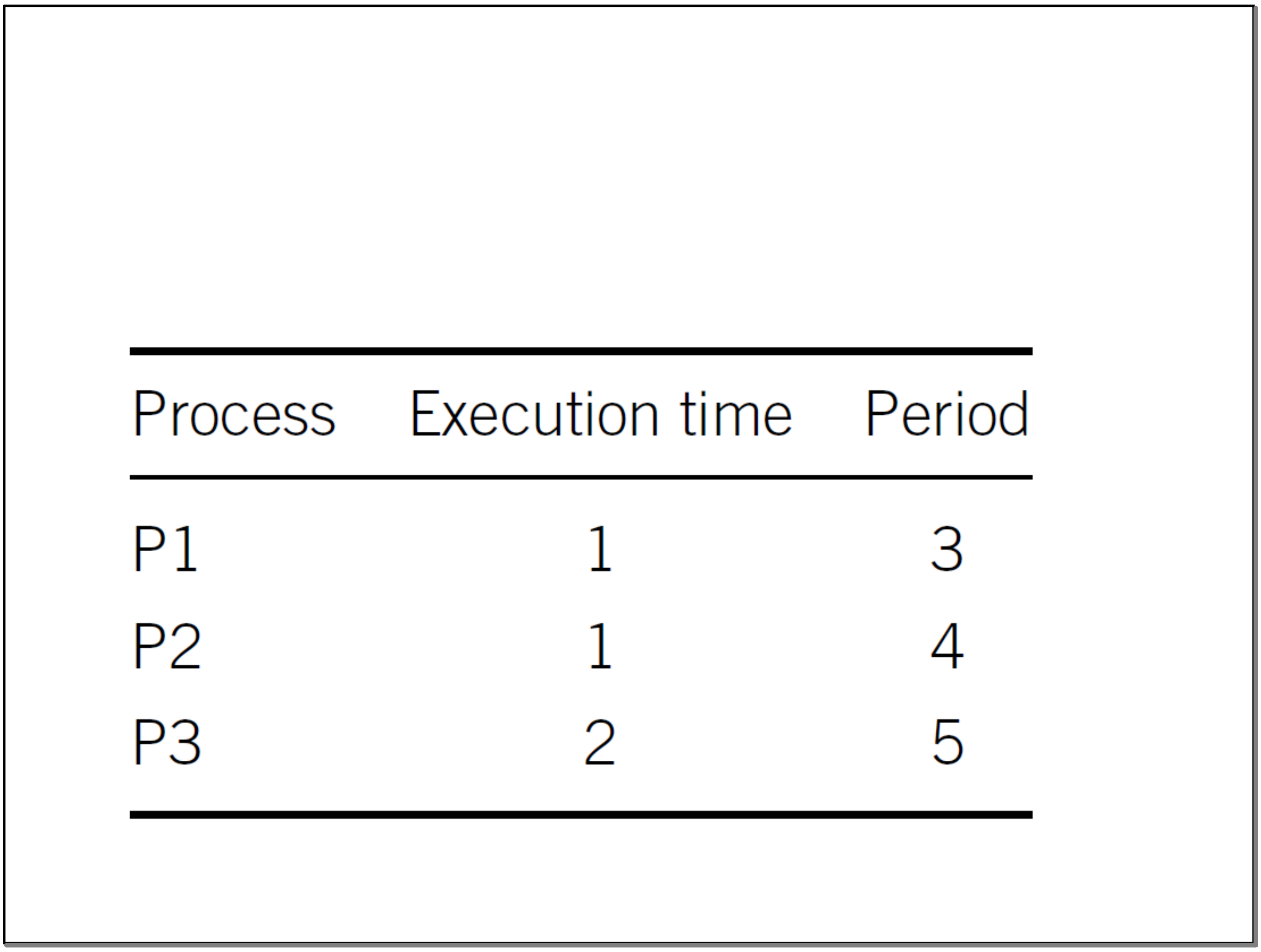


Process	Execution time	Period
P1	1	4
P2	2	6
P3	3	12

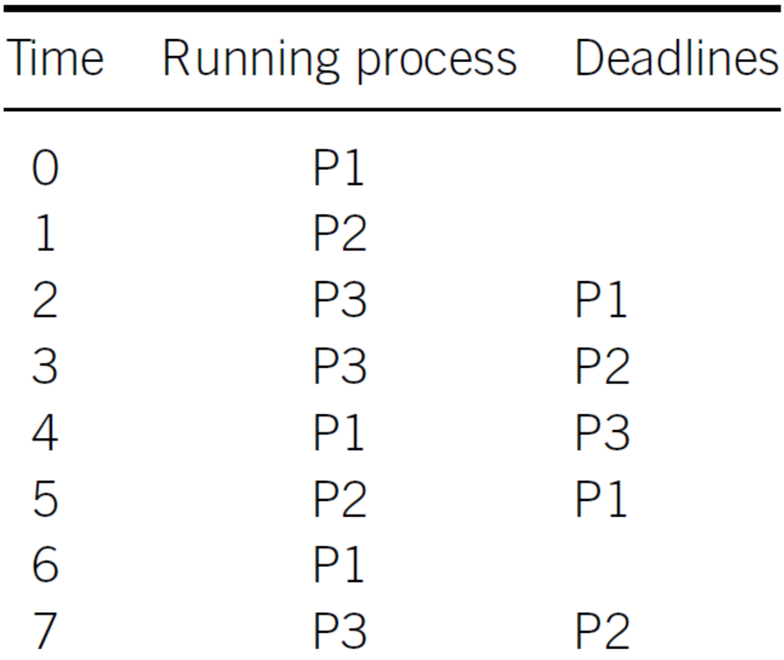


Sep 21-05:57

Earliest Deadline First Scheduling



Process	Execution time	Period
P1	1	3
P2	1	4
P3	2	5



Time	Running process	Deadlines
0	P1	
1	P2	
2	P3	P1
3	P3	P2
4	P1	P3
5	P2	P1
6	P1	
7	P3	P2

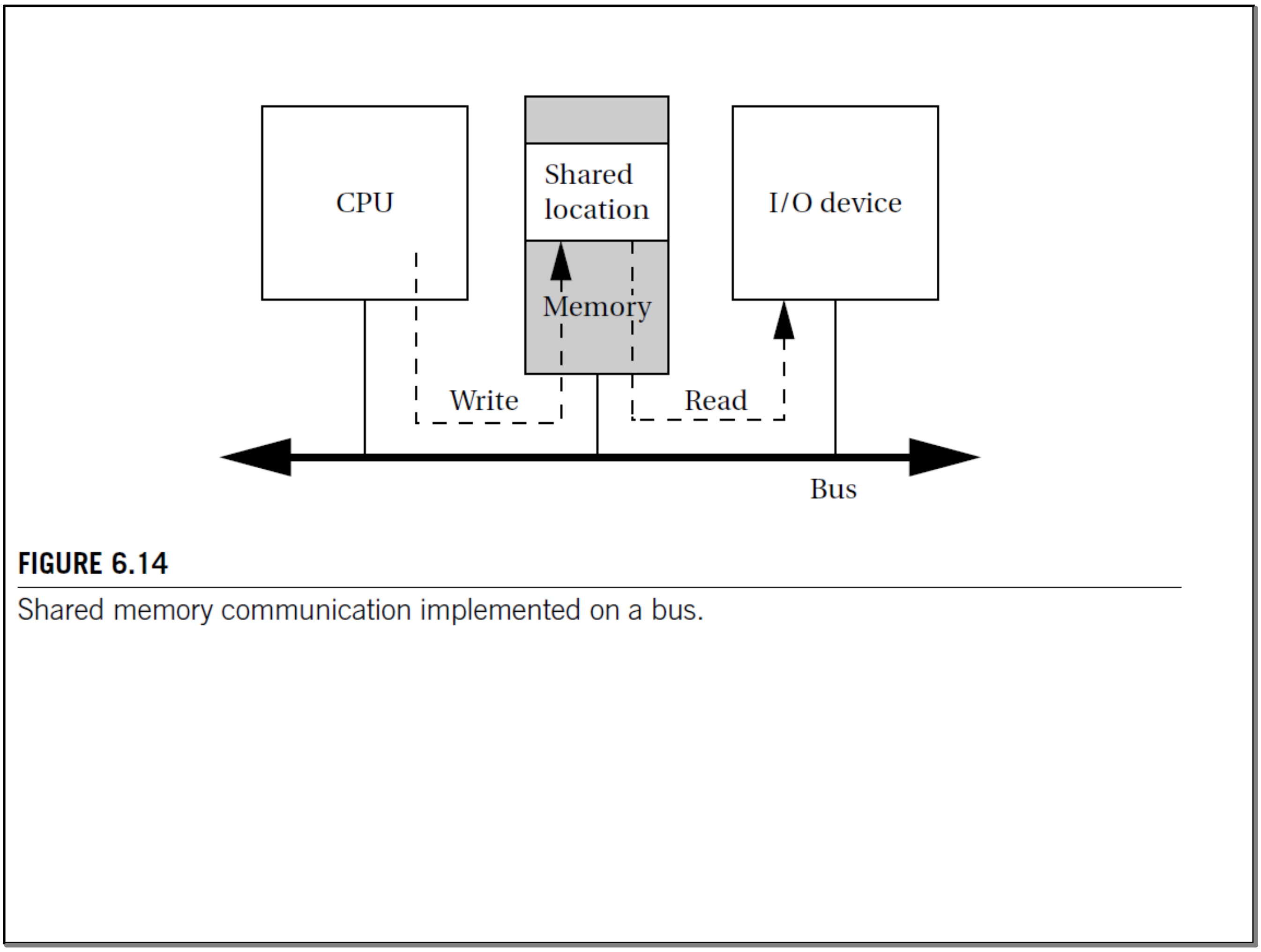
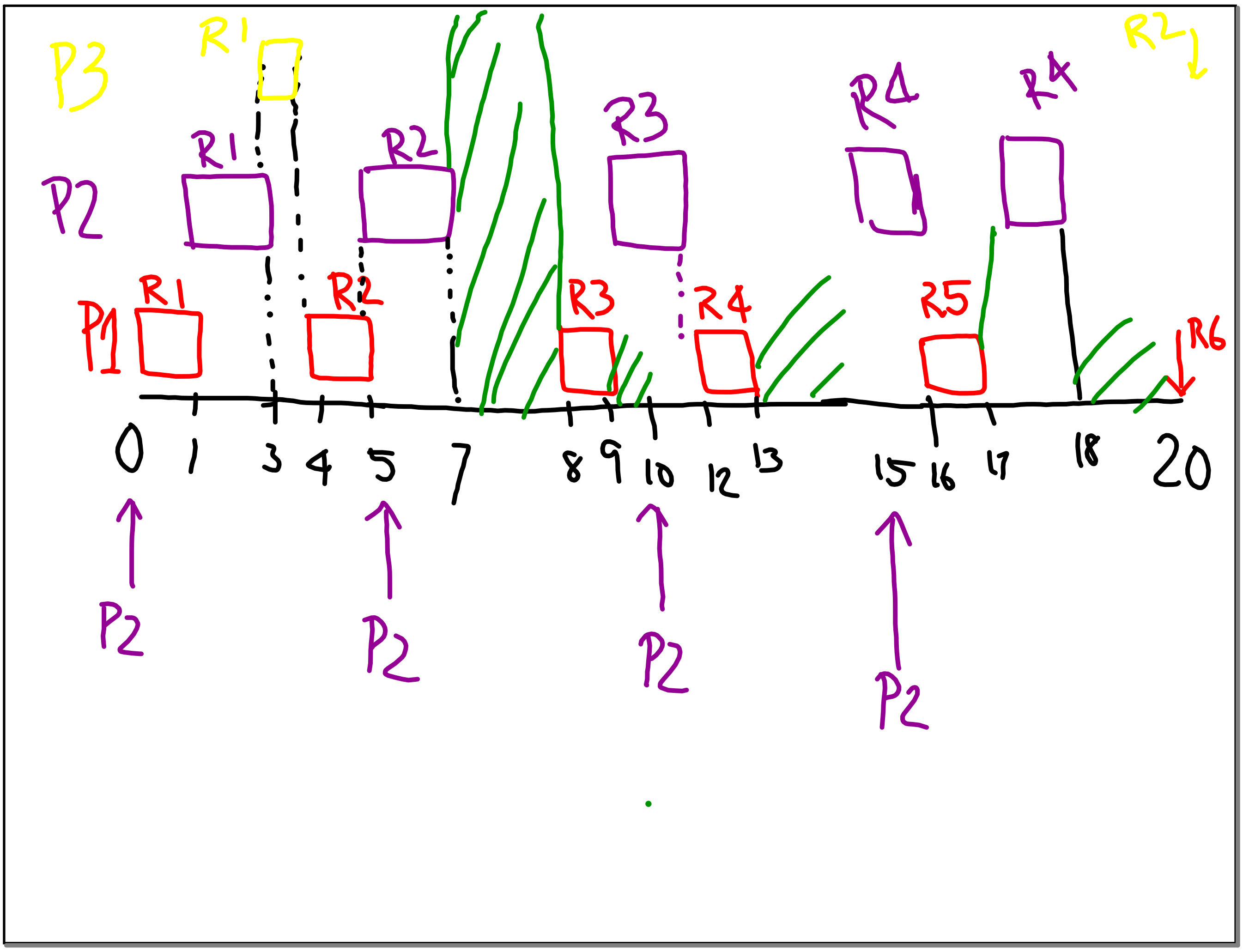
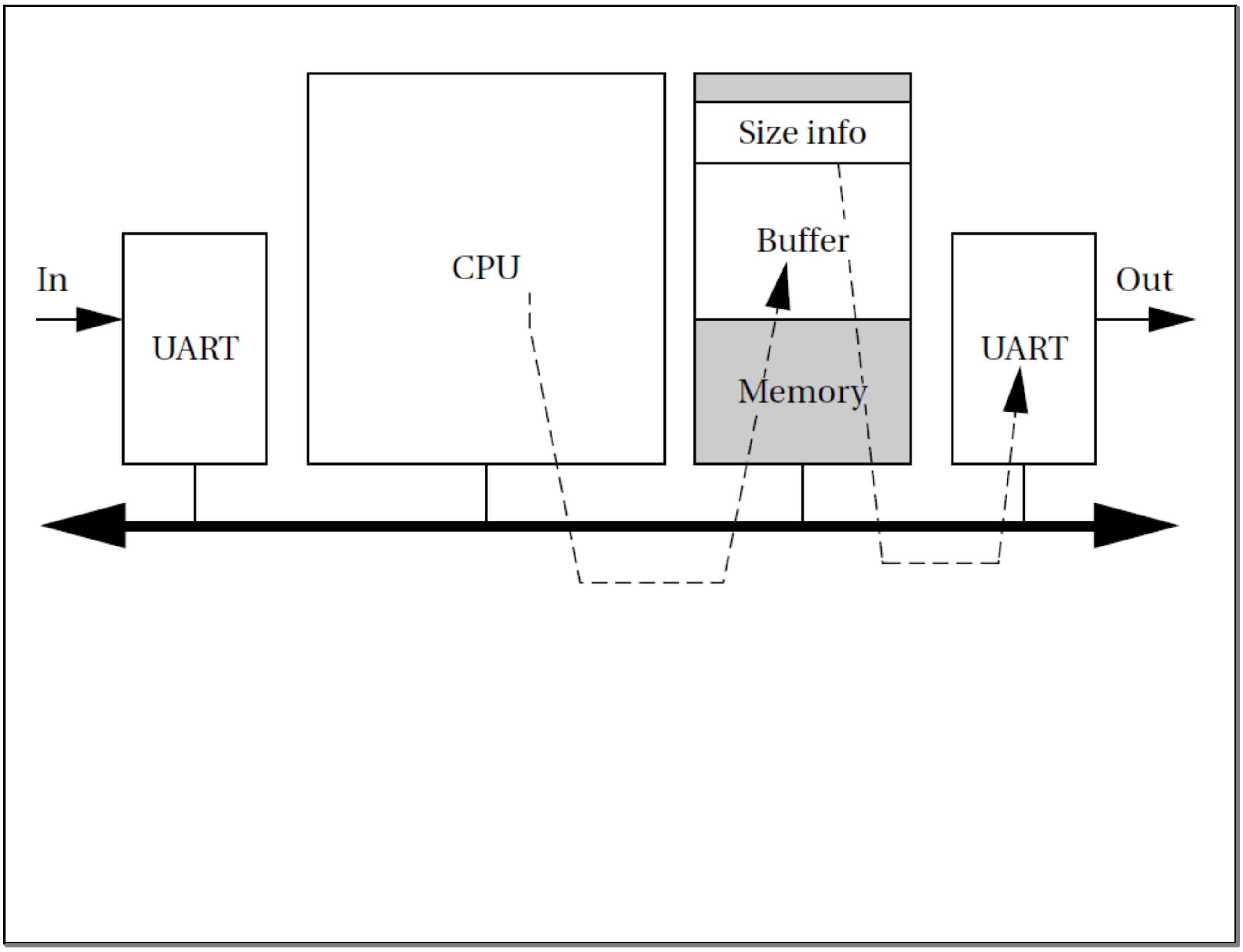


FIGURE 6.14

Shared memory communication implemented on a bus.


P3
R1
R2
P2
R1
R2
R3
R4
R4
P1
R1
R2
R3
R4
R5
R6
0
1
3
4
5
7
8
9
10
12
13
15
16
17
18
20
P2
P2
P2
P2

Sep 26-09:37



Sep 21-06:05

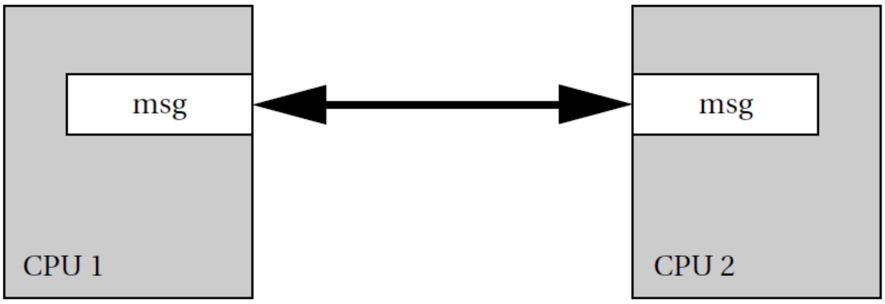


FIGURE 6.15

Message passing communication.