Multiple Core/Computer Scheduling

Migration: Mobility of the tasks and instances

- **No migration (partitioned)** – The set of tasks is partitioned into as many disjoint subsets as there are processors available, and each such subset is associated with a unique processor. All instances generated by the tasks in a subset must execute only upon the corresponding processor.

- **Restricted migration** – Each instance must execute entirely upon a single processor. However, different instances of the same task may execute upon different processors. (The runtime context of each instance needs to be maintained upon only one processor.)

- **Full migration** – No restrictions are placed upon interprocessor migration.

Priorities of Tasks and Instances

- **Static priorities** – A priority is associated with each task, and all instances generated by a task have the same priority. (Similar to the RM algorithm.)
- **Instance-level dynamic priorities** – For every pair of instances of tasks $T_{m,i}$ and $T_{n,j}$, the priority relation between them stays the same during the course of execution. (Similar to the EDF algorithm.)
- **Unrestricted dynamic priorities** – The relative priority of two jobs may change at any time. (Similar to the LLF algorithm.)

Combinations…

	Static	**Instance Dynamic**	**Unrestricted Dynamic**
Partitioned	PS	PID	PUD
Restricted Mobility	RMS	RMID	RMUD
Full Mobility	FMS	FMID	FMUD

Examples

A: $T1 = (1,2)$, $T2 = (2,3)$, $T3 = (2,3)$; $M=2$

B: $T1 = (2,3)$, $T2 = (2,3)$, $T3 = (2,3)$; $M=2$

C: $T1 = (12,12)$, $T2 = (2,4)$, $T3 = (3,6)$; $M=2$

D: $T1 = (3,6)$, $T2 = (3,6)$, $T3 = (6,7)$; $M=2$

E: $T1 = (3,4)$, $T2 = (5,7)$, $T3 = (3,7)$; $M=2$

F: $T1 = (4,6)$, $T2 = (7,12)$, $T3 = (4,12)$, $T4 = (10,24)$; $M=2$

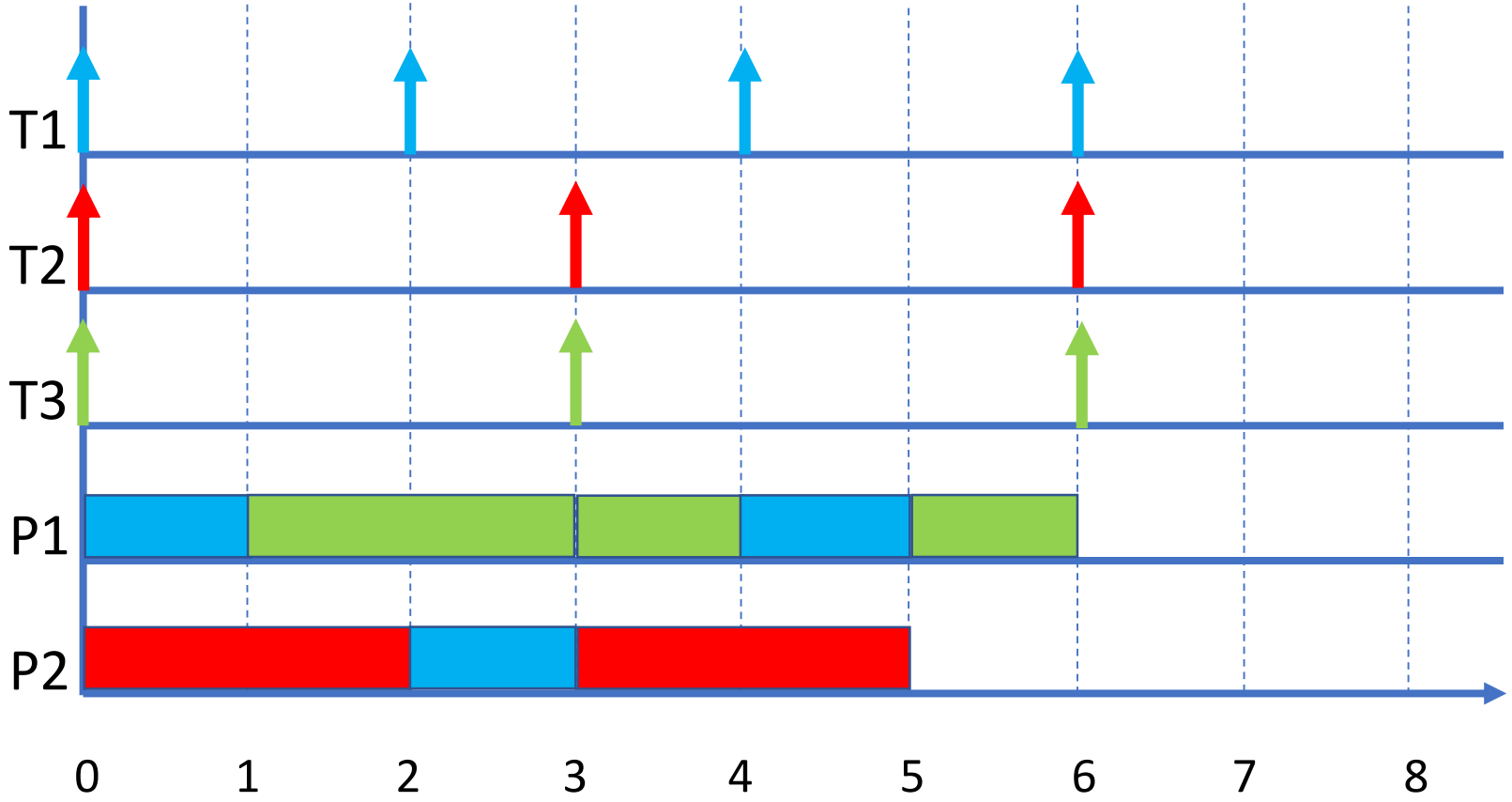
G: $T1 = (7,8)$, $T2 = (10,12)$, $T3 = (6,24)$; $M=2$

H: $T1 = (4,6)$, $T2 = (4,6)$, $T3 = (2,3)$; $M=2$

I: $T1 = (2,3)$, $T2 = (3,4)$, $T3 = (5,15)$, $T4 = (5,20)$; $M=2$

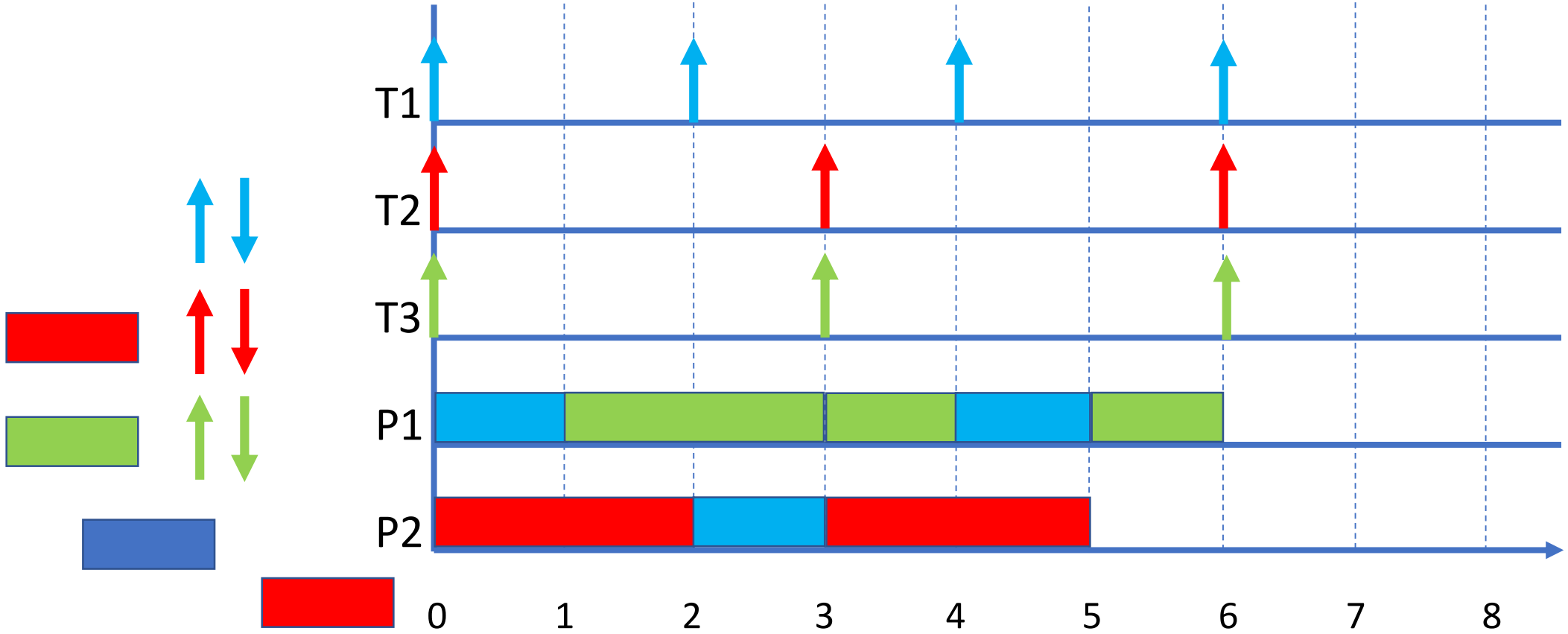
RMS

*T*1 = (1,2), *T*2 = (2,3), *T*3 = (2,3); *M*=2

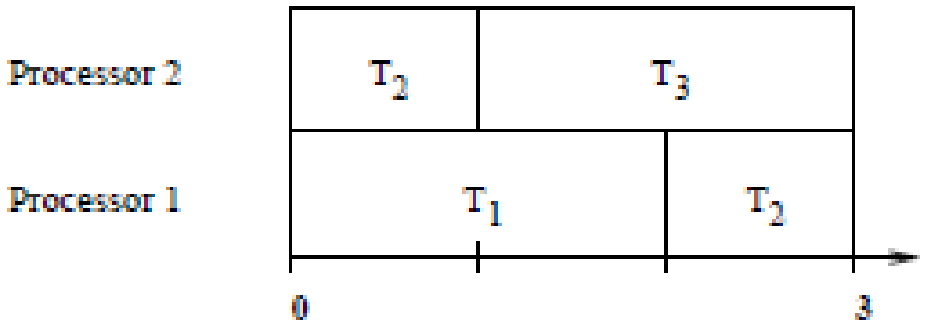


RMS

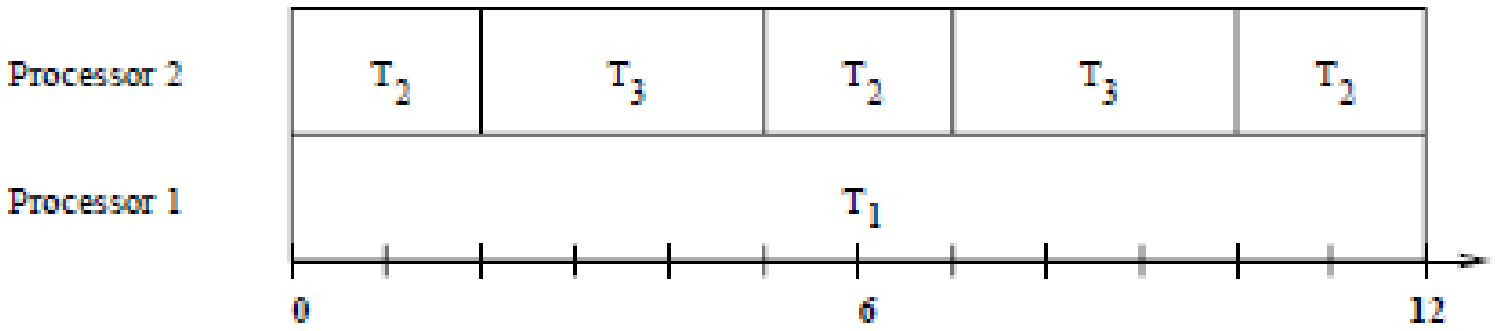
$T1 = (1,2)$, $T2 = (2,3)$, $T3 = (2,3)$; $M=2$



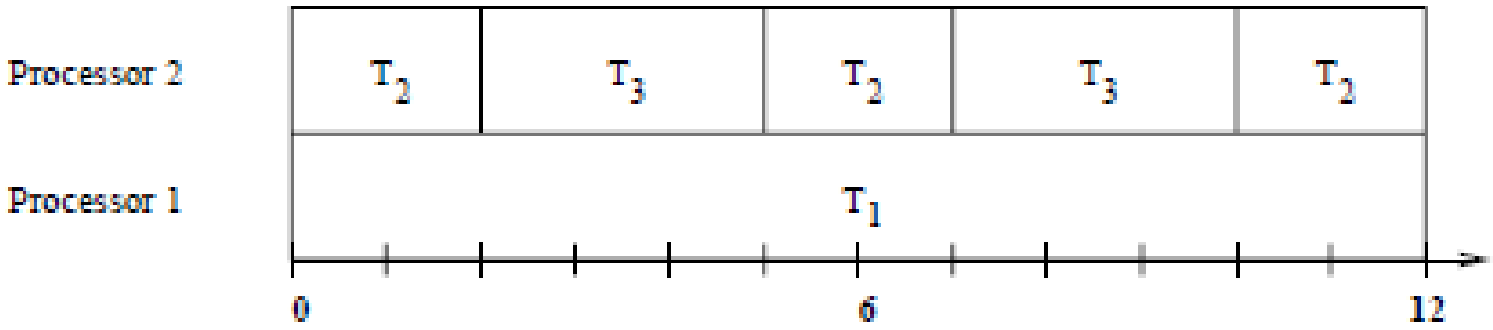
B – (3,3)

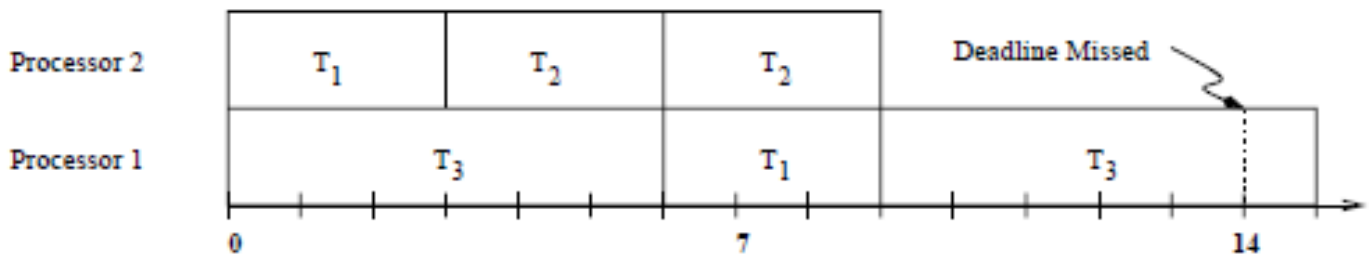


C

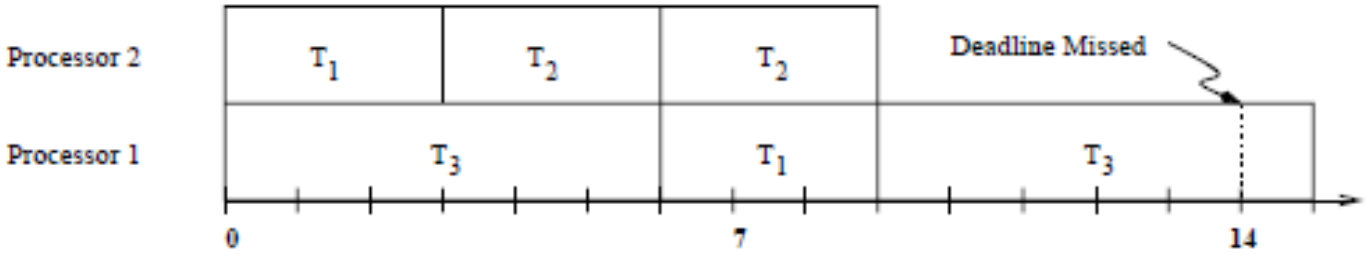


C – (2,1)

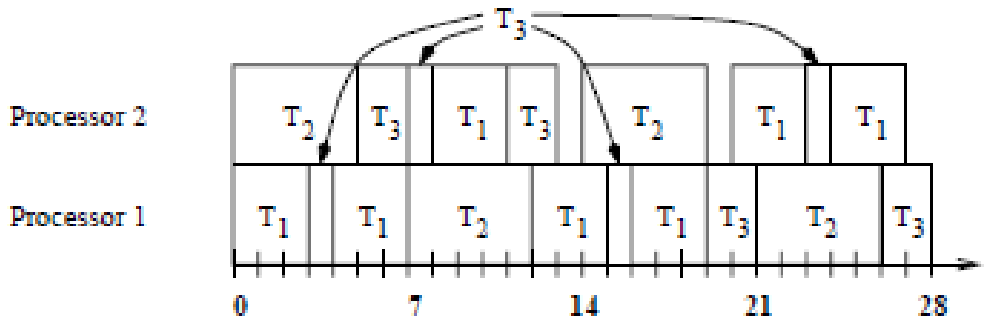



Processor 2
Processor 1
T1
T2
T2
T3
T1
T3
Deadline Missed
0
7
14

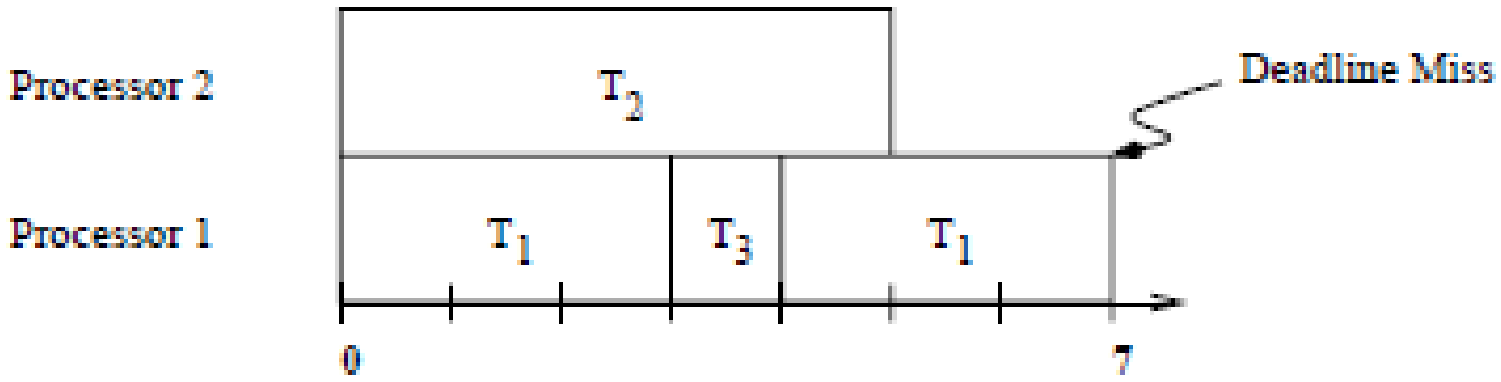
D – ()


Processor 2
Processor 1
T_1
T_2
T_2
Deadline Missed
T_3
T_1
T_3
0
7
14

E

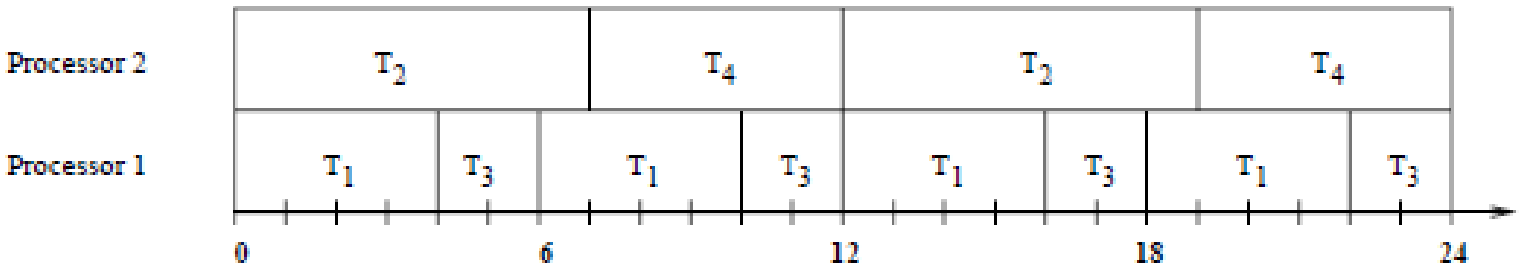

T3
Processor 2
T2
T3
T1
T3
T2
T1
T1
Processor 1
T1
T1
T2
T1
T1
T3
T2
T3
0
7
14
21
28

E –(1,2)

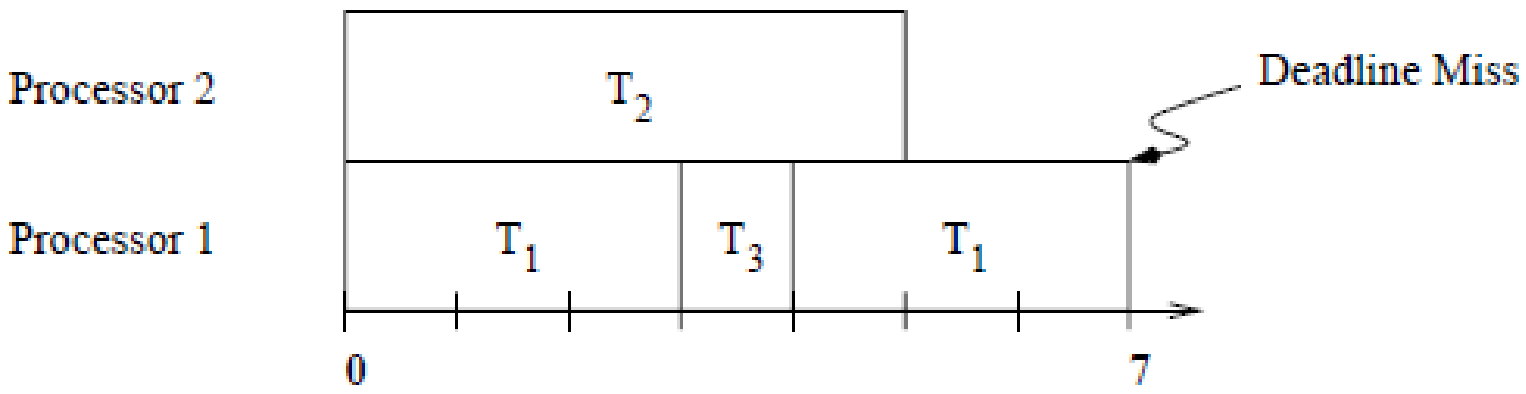


F – (1,2)

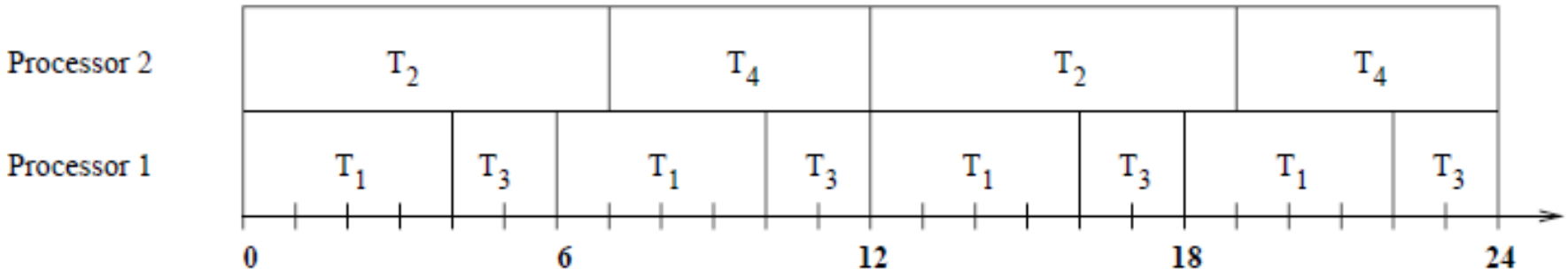
- *T*1 = (4,6), *T*2 = (7,12), *T*3 = (4,12), *T*4 = (10,24); *M*=2



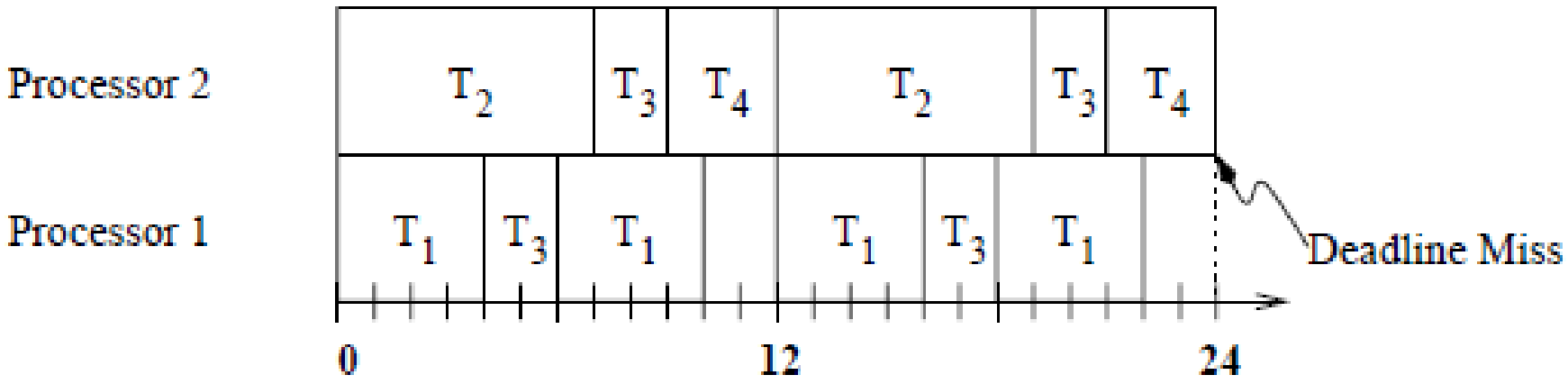
$T1 = (4,6)$, $T2 = (7,12)$, $T3 = (4,12)$, $T4 = (10,24)$; $M=2$



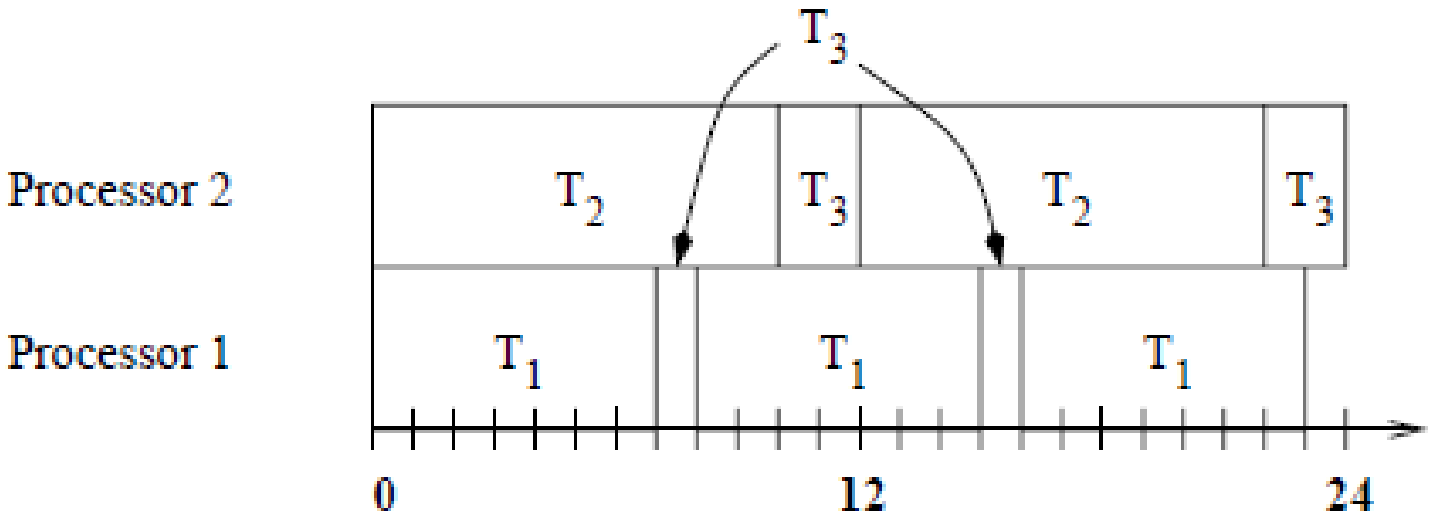
$T1 = (4,6)$, $T2 = (7,12)$, $T3 = (4,12)$, $T4 = (10,24)$; $M=2$



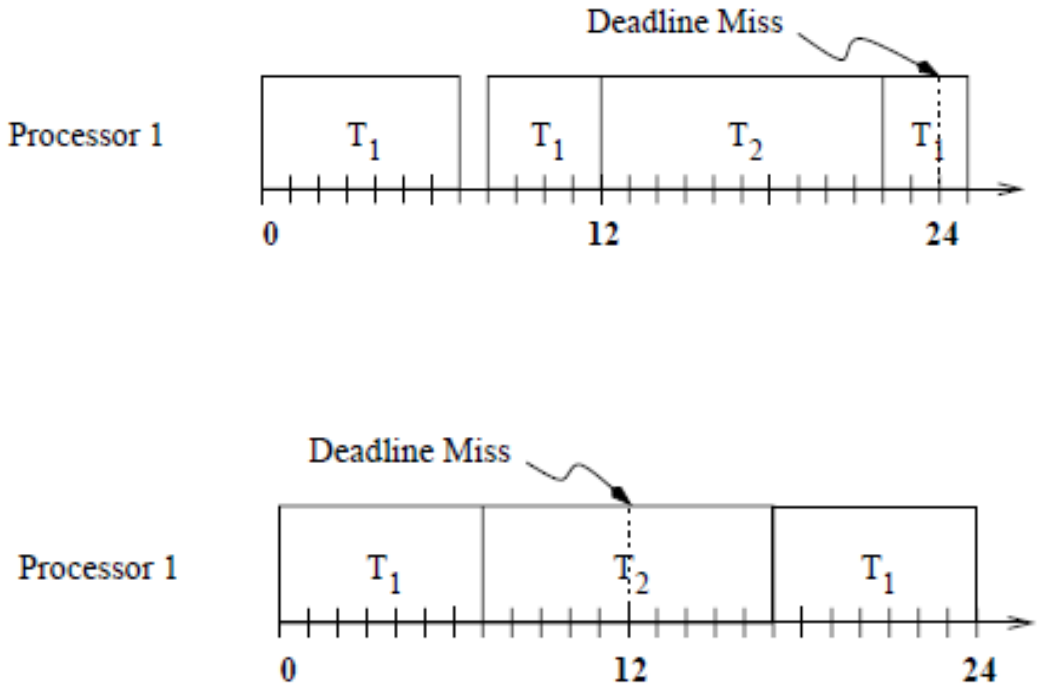
F (1,3)


Processor 2
Processor 1
T2
T3
T4
T2
T3
T4
T1
T3
T1
T1
T3
T1
Deadline Miss
0
12
24

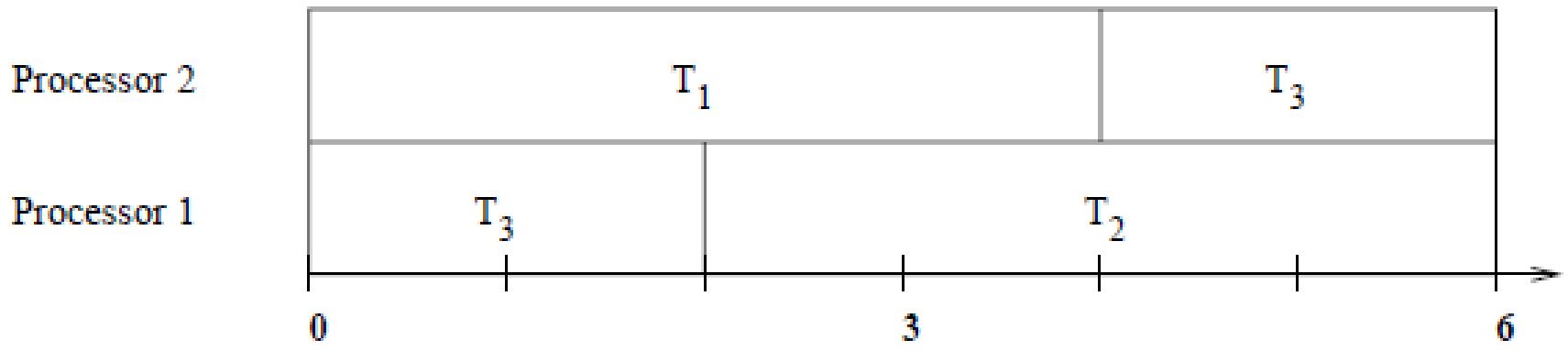
G 1,3



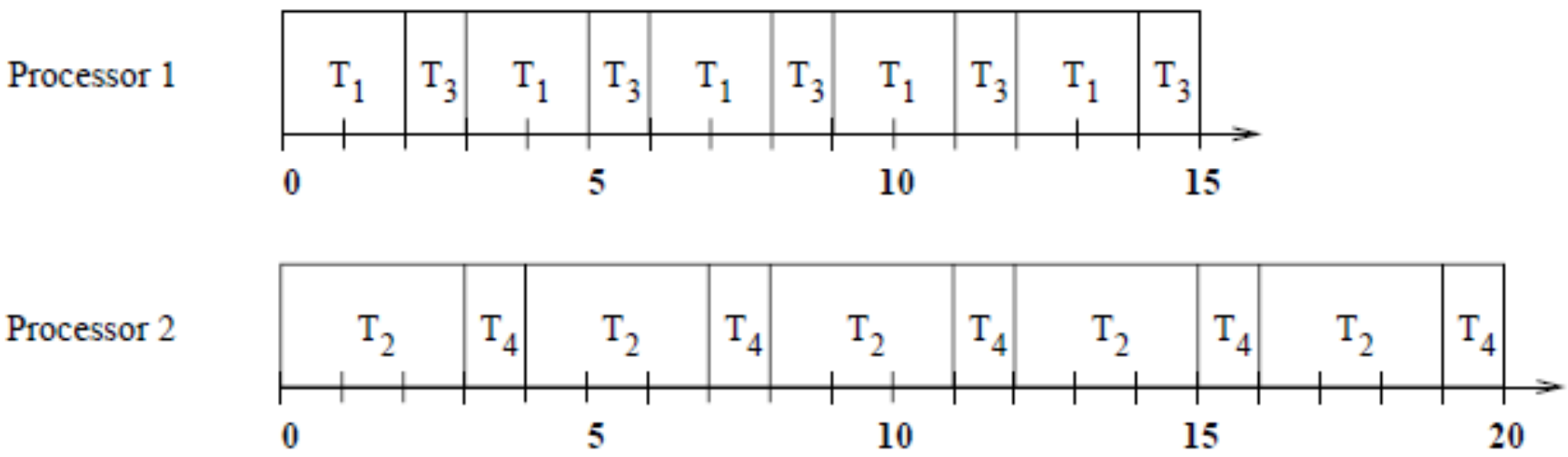
G 32


Deadline Miss
Processor 1
T1
T1
T2
T1
0
12
24
Deadline Miss
Processor 1
T1
T2
T1
0
12
24

H 32



I 11


Processor 1
T1 T3 T1 T3 T1 T3 T1 T3 T1 T3
0 5 10 15
Processor 2
T2 T4 T2 T4 T2 T4 T2 T4 T2 T4
0 5 10 15 20