INTER-PROCESS COMMUNICATION AND SYNCHRONISATION:

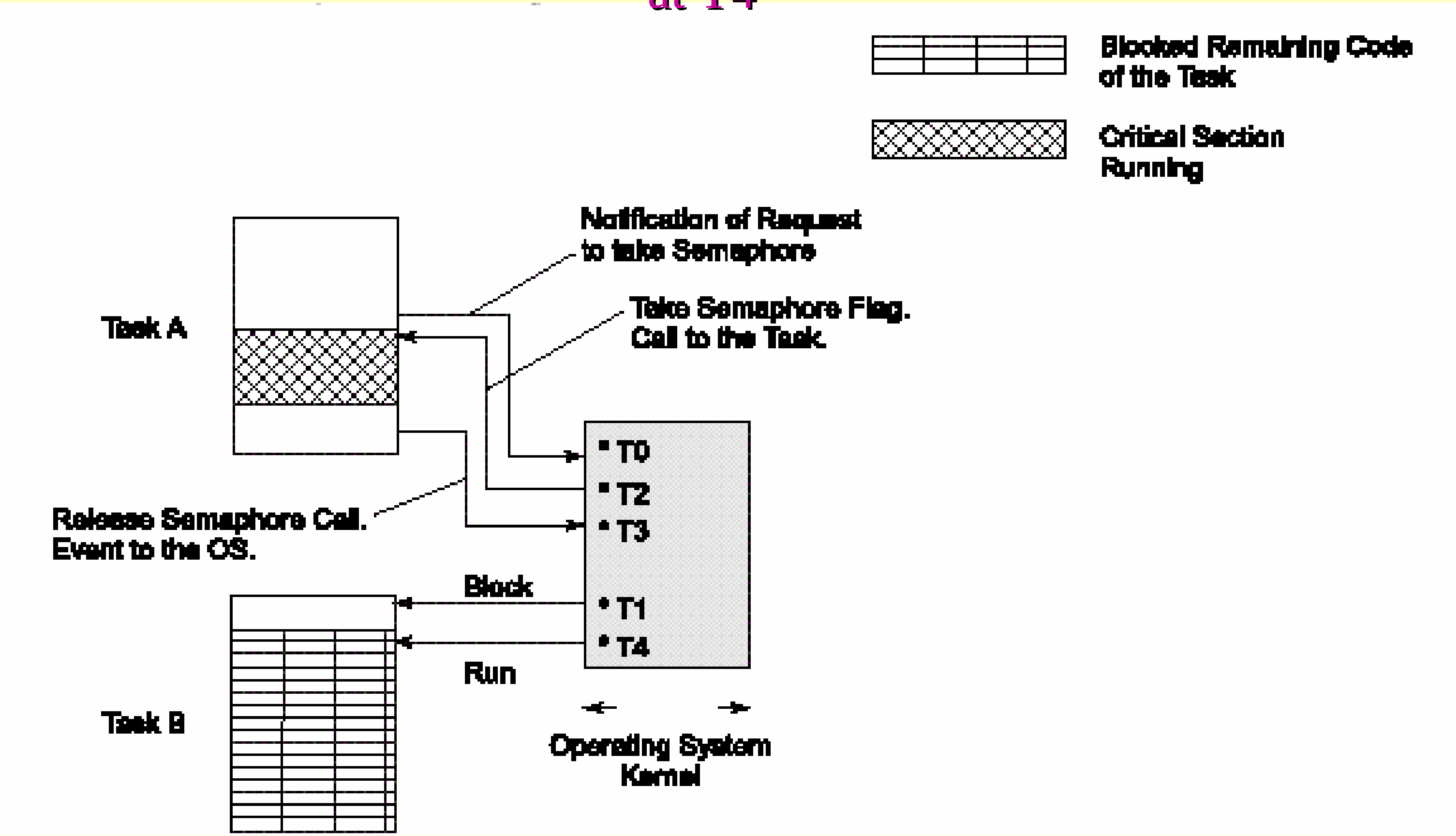
Lesson-14: Mutex, Lock and Spin Lock functions

1. Mutex Semaphore

Mutex Semaphore

- Process using a Mutex *blocks* a critical section in a task for taking the mutex and unblocks on releasing the mutex.
- The mutex wait for lock can be specified a timeout .

Task A taking semaphore at T2, releasing at T3, B then runs at T4



Blocked Remaining Code of the Task
Critical Section Running
Notification of Request to take Semaphore
Take Semaphore Flag. Call to the Task.
Task A
Release Semaphore Call. Event to the OS.
T0
T2
T3
T1
T4
Block
Run
Task B
Operating System Kernel

2. Lock function

Lock ( )

- Process using lock ( ) enters a critical section *locks* the resources to a critical section till at the end of the section unlock ( ) is used.
- A wait loop creates on executing lock ( ) and wait is over when other critical section executes unlock ( )
- lock ( ) and unlock ( ) involves little overhead (number f operations) than the OSSemPend ( ) and OSSemPost ( )

3. Lock function disadvantages

Disadvantage

- *A resource of high priority should not lock the other processes by blocking an already running task in the following situation*. Suppose a task is running and a little time is left for its completion.

4. Spin Lock function

Spinlock ( )

- Suppose a task is running and a little time is left for its completion.
- The running time left for it is less compared to the time that would be taken in blocking it and context switching.
- There is an innovative concept of spin locking in certain schedulers.
- A ***spin lock*** is a powerful tool in the situation described above.

Spinlock ( )…

- The scheduler locking processor for a task waits to cause the blocking of the running task first for a time-interval t, then for (t - δt), then (t - 2δt) and so on.
- When this time interval spin downs to 0, the task that requested the lock of the processor now unlocks the running task and blocks it from further running. The request is now granted.

Spinlock ( )…

- A spin lock does not let a running task be blocked instantly
- First successively tries decreasing the trial periods before finally blocking a task

Summary

We learnt

- OS provides the IPC functions Create, Post, Pend, Accept and Query for using mutex semaphores. The time out and error handling function can be provided with Pend function as arguments.
- OS provides the IPC functions for creating and using lock ( ) and unlock ( ) for the resource for a process and to lock and unlock the resources for the other processes.

We learnt

- OS provides the IPC functions for creating and using spinlock ( ) and unlock ( ) for the resource for a process and to lock and unlock the resources for the other processes for successively decreasing trial periods for before doing unlocking.

End of Lesson-14 on Mutex, Lock and Spin Lock functions