Course Name : Embedded System EC5464

RA PRANJALE
LECTURER IN ELECTRONICS
GOVERNMENT POLYTECHNIC AMRAVATI

EMBEDDED SYSTEMS EC5464

CO6 :Interpret the features of REAL TIME OPERATING SYSTEM

Topic 6.1 : Operating System: General and Real time operating system.

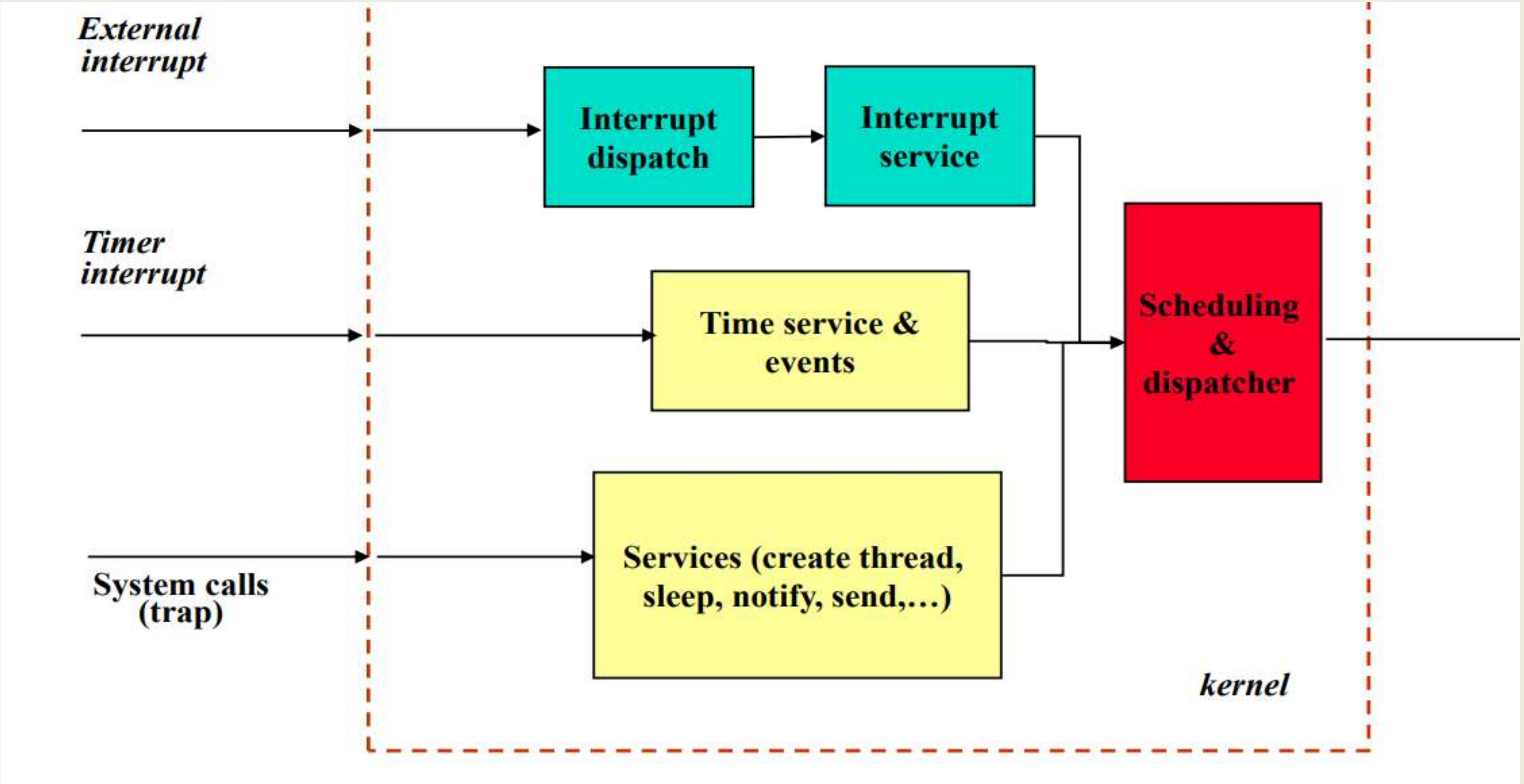
Topic 6.2: Characteristics of Real Time Operating System: Consistency, Reliability, scalability, Performance, Predictability

Topic 6.3 Functions of RTOS: Task management, Scheduling, Resource allocation and interrupt handling .

Topic 6.4 Task synchronization and Mutual Exclusion, Multitasking

Topic 6.5 : Features of RTOS: Watchdog timer, Semaphore, Deadlock. Starvation Deadlock , Multiple process

INTRODUCTION: TASK MANAGEMENT



TASK MANAGEMENT : OPERATIONS

❖Execution of quasi-parallel tasks on a processor using processes or threads by

- maintaining process states, process queuing,
- allowing for preemptive tasks (fast context switching) and quick interrupt handling

❖CPU scheduling (guaranteeing deadlines, minimizing process waiting times, fairness in granting resources such as computing power)

❖Process synchronization (critical sections, semaphores, monitors, mutual exclusion)

❖Inter-process communication (buffering)

❖Support of a real-time clock as an internal time reference

TAK MANAGEMENT : OPERATIONS

❖Task synchronization:

In classical operating systems, synchronization and mutual exclusion is performed via semaphores and monitors.

❖ In real-time OS, special semaphores and a deep integration into scheduling is necessary (priority inheritance protocols,

❖Further responsibilities:

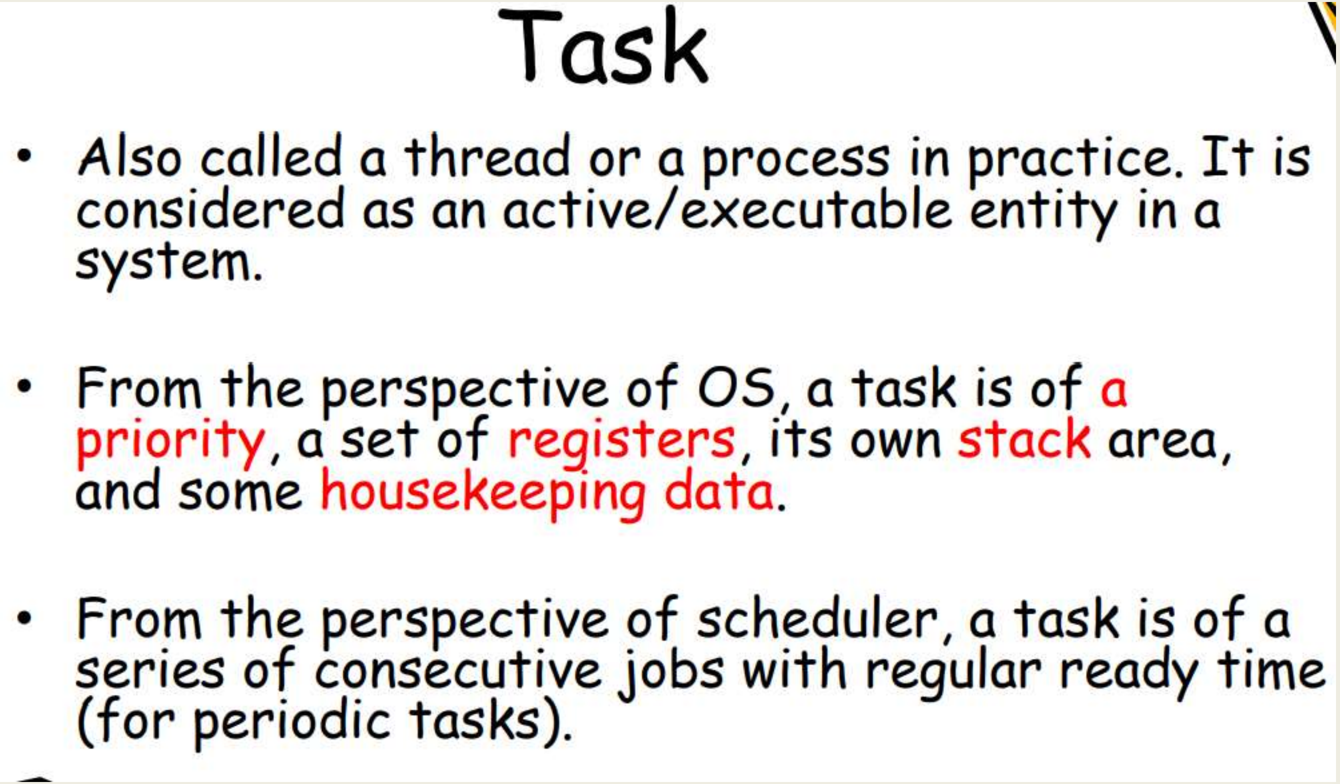
Initializations of internal data structures (tables, queues, task description blocks, semaphores, …)

PROCESS THREADS AND TASK

A process is a program in execution, whereas a thread is a path of execution within a process. Processes are generally used to execute large, ‘heavyweight’ jobs such as running different applications, while threads are used to carry out much smaller or ‘lightweight’ jobs such as auto saving a document in a program, downloading files, etc. Whenever we double-click an executable file such as Paint, for instance, the CPU starts a process and the process starts its primary thread.

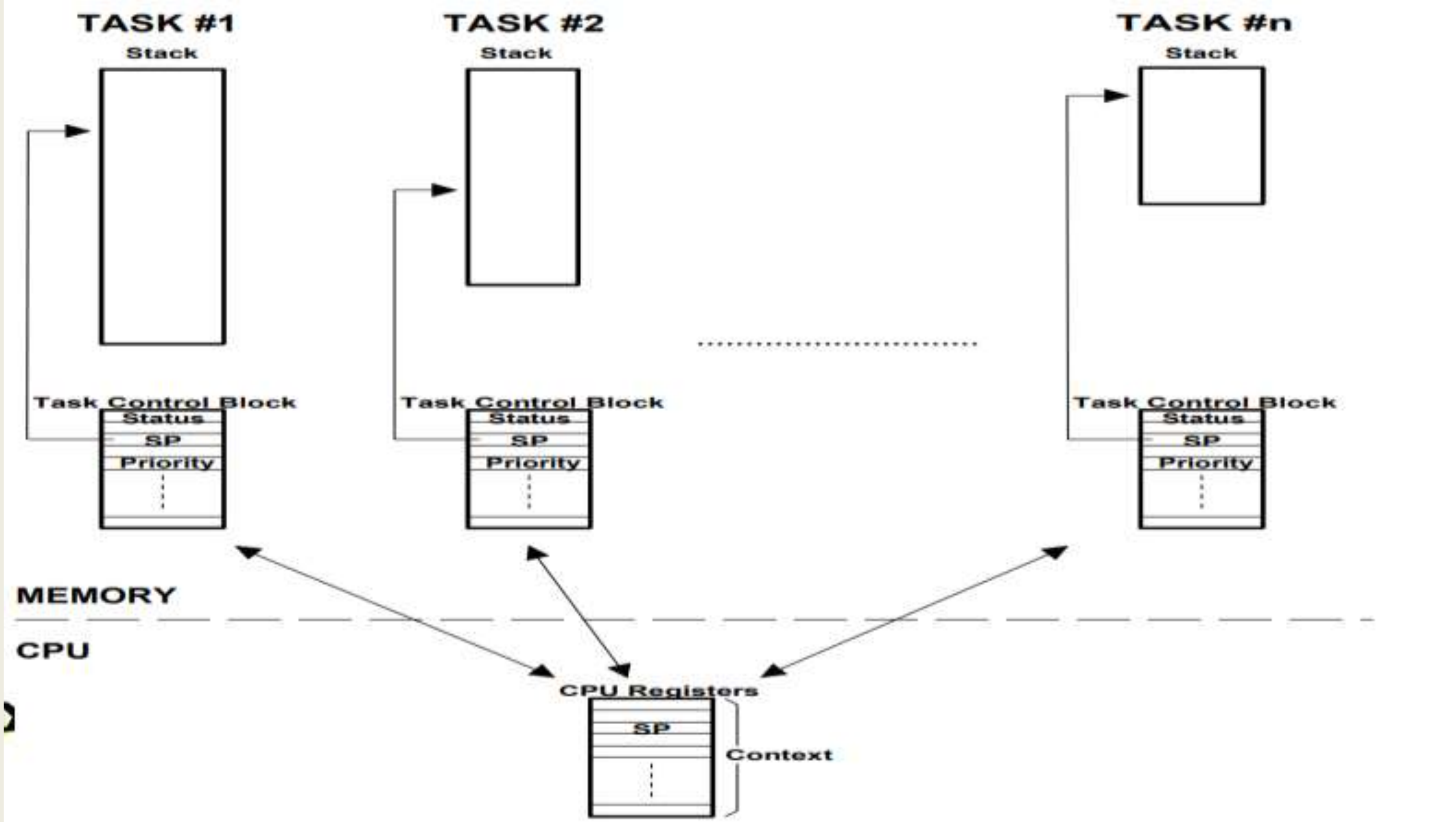
Each process runs in a separate address space in the CPU. But threads, on the other hand, may share address space with other threads within the same process. This sharing of space facilitates communication between them. Therefore, Switching between threads is much simpler and faster than switching between processes.

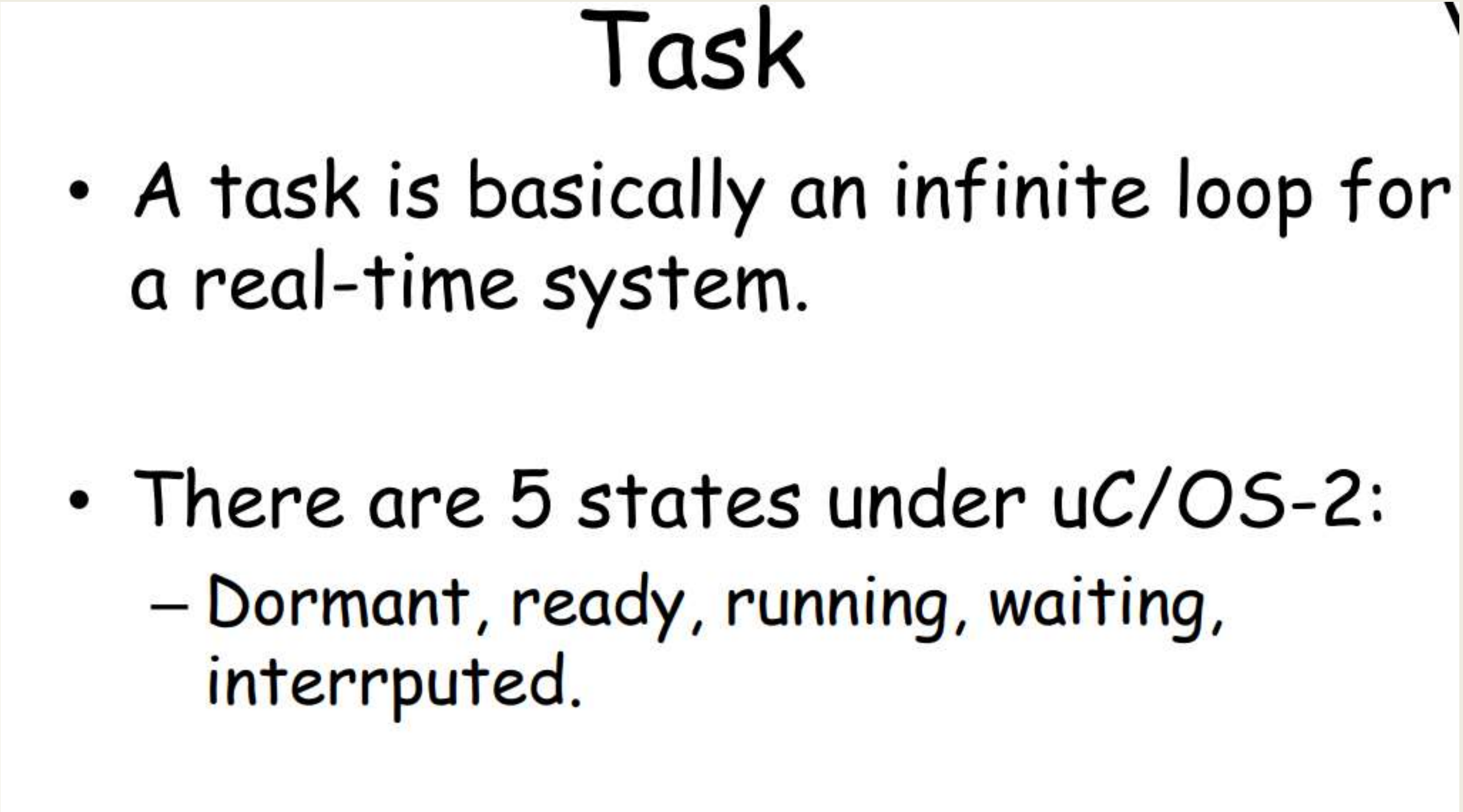
A *task* is a generic term, which, refers to an independently schedulable unit of computation, and is used typically in the context of scheduling of computation on the processor. It may refer either to a process or a thread



Task

- Also called a thread or a process in practice. It is considered as an active/executable entity in a system.
- From the perspective of OS, a task is of a priority, a set of registers, its own stack area, and some housekeeping data.
- From the perspective of scheduler, a task is of a series of consecutive jobs with regular ready time (for periodic tasks).


TASK #1
Stack
TASK #2
Stack
TASK #n
Stack
Task Control Block
Status
SP
Priority
Task Control Block
Status
SP
Priority
Task Control Block
Status
SP
Priority
MEMORY
CPU
CPU Registers
SP
Context



Task

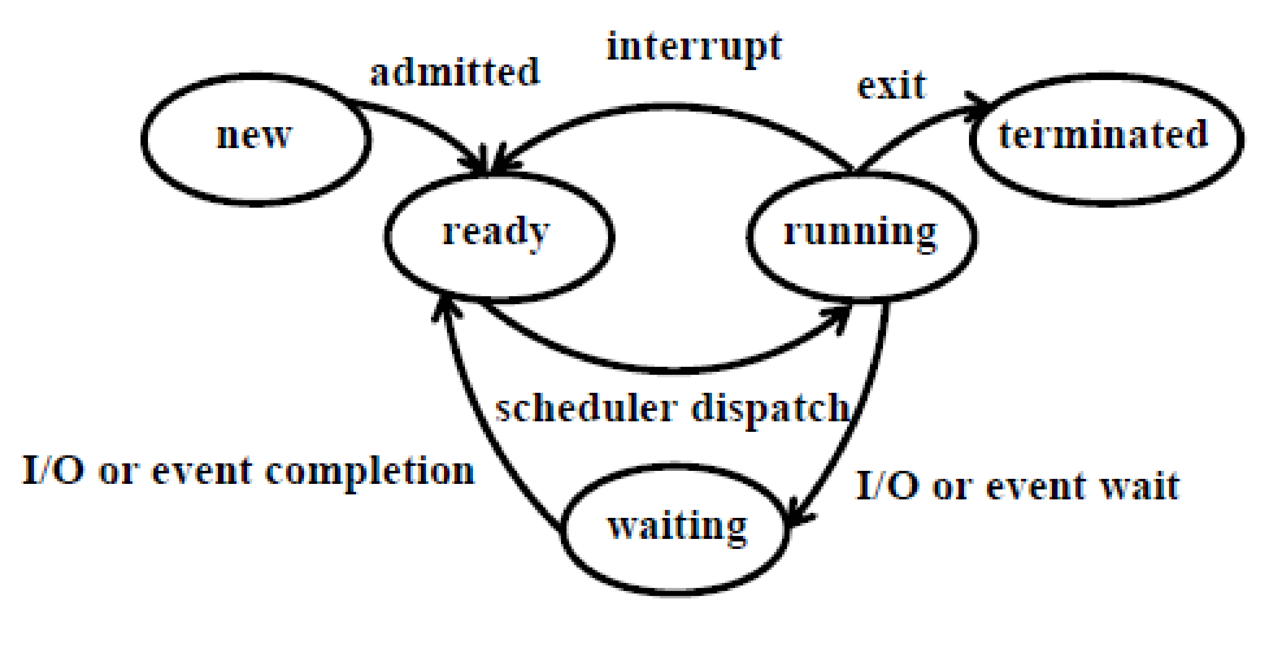
- A task is basically an infinite loop for a real-time system.

- There are 5 states under uC/OS-2:
  - Dormant, ready, running, waiting, interrputed.

TYPES OF TASK STATES

- ♦ Running: This is the task which has control of the CPU. It will normally be the task which has the highest current priority of the tasks which are ready to run.
- ♦ Ready: There may be several tasks in this state. The attributes of the task and the resources required to run it must be available for it to be placed in the 'ready' state.
- ♦ Waiting: The execution of tasks placed in this state has been suspended because the task requires some resources which is not available or because the task is waiting for some signal from the plant, e.g., input from the analog-to-digital converter, or the task is waiting for the elapse of time.
- ♦ New: The operating system is aware of the existence of this task, but the task has not been allocated a priority and a context and has not been included into the list of schedulable tasks.
- ♦ Terminated: The operating system has not as yet been made aware of the existence of this task, although it may be resident in the memory of the computer.

TASK STATES


interrupt
admitted
exit
new
terminated
ready
running
scheduler dispatch
I/O or event completion
I/O or event wait
waiting

Thank You