

## Lecture 2: Embedded Systems Definitions

|                        |  |
|------------------------|--|
| Asynchronous -         | Two tasks executing independently of each other which share memory/resources   |
| Cache -                | Very high speed buffer memory device which computes what RAM memory is about to be requested by CPU and places it in its memory                                    |
| Concurrency -          | Operating system's management of inter process synchronization, computer resource sharing, and communication   |
| Conduction band -      | Lowest energy level in an atom in which electrons do not completely fill   |
| Context switch -       | Act of kernel to switch from execution of one task to resume another task  |
| Crash -                | Computer system has non-recoverable response to further input  |
| Critical section -     | Sequence of computer instructions which cannot be shared at the same time by different tasks and must be executed indivisibly                                      |
| Deadlock -             | Situation where no task can run because each is held up by another's resource it needs (livelock is deadlock where at least one task can run but with no progress) |
| Device driver -        | Software application that runs or controls a piece of hardware   |
| Embedded system -      | Real-time computer system which interfaces with more hardware components via external interrupts from the hardware   |
| Fault tolerance -      | Capability of computer system to continue functioning in presence of errors  |
| Interrupt -            | Signal input to real-time system to give execution priority to its associated task   |
| Interrupt controller - | External device for handling multiple interrupts at different priority levels which interfaces with CPU's single interrupt input line                              |
| Interrupt handler -    | Software routine associated with internal or external interrupt  |
| ISR -                  | Interrupt Service Routine which is an interrupt handler servicing its interrupt  |
| Kernel -               | Lowest layer of computer operating system which controls process synchronization, concurrency, scheduling, time services, and interrupt handling                   |
| Mutual exclusion -     | Situation in which two actions or tasks cannot be executed at the same time  |
| Nonvolatile-           | Memory which stores information independent of power on or off   |
| Pipe -                 | Distinct communications channel between two synchronous tasks with two possible operations: write or read  |
| Priority inversion     | Act of lower priority task blocking a higher priority task from executing  |
| Processor -            | Computer or server   |
| Process -              | Distinct executable programming construct and acts as an active object or task of a real-time system   |
| Proxy -                | Task or message which allows for Asynchronous communication between two separate tasks   |
| Race condition -       | Situation which arises when different tasks compete for the same computer resource in order to get it first  |
| Real-time system -     | Computer system which interfaces with external events  |
| Recovery -             | Computer system's act of resuming from some previous state (backward recovery) or from some future pre-programmed state (forward recovery)                         |
| Semaphore -            | Non-negative integer variable used for protection of critical sections and mutual exclusion with simply two possible operations: wait and signal                   |
| Socket -               | Virtual (software) I/O device that serves as a network node point  |
| Spinlock -             | Task spins continuously waiting for resource to become available to lock   |
| Stack -                | List of computer instructions and values stored in random access memory  |
| Starvation -           | Situation in which a task never gets required CPU priority to execute  |
| Synchronous -          | Act of two tasks executing in step according to a shared message send and response   |
| Task -                 | Sequence of computing instructions that execute a single system function   |
| Task Control Block -   | Data structure containing task management and user input information; ie task ID, task pointer, type, priority, state, period, deadline                            |
| Thread -               | Program construct within a process which shares its memory   |
| Valence Band -         | Highest energy level in an atom in which electrons completely fill   |
| Watchdog -             | Timer process that interrupts the system error handler when it is not reset by some predefined time by another system process                                      |
| Zombie -               | Terminated task that still exists in kernel's task control block memory area   |