

# Linux For Embedded Systems

# Cairo University Computer Eng. Dept. CMP445-Embedded Systems



**Ahmed ElArabawy** 





Lecture 3:

**OSs for Embedded Platforms** 



# Operating Systems .....





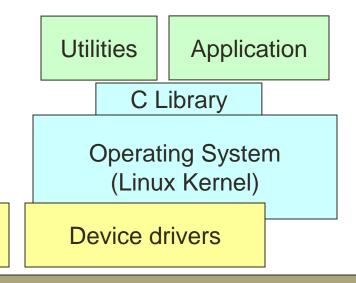
## Operating Systems .....



- What the end user sees is just the Operating System frond end user interface (Desktop)
- Behind the scene there is the KERNEL
- The Kernel runs a lot of functionality in the background and performs a lot of tasks for the user and the developer, and abstracts a lot of HW details







(bootloader)

Initialization

Hardware device (processor, memory, storage, peripherals)



#### **FUNCTIONS OF OPERATING SYSTEM**







- The kernel supports multiple users using it
- Each user has his own resources and privileges
- There is the super-user (root/admin) that have full control on the system
- User privileges decides his rights on his owned resources and other users resources
- The kernel manages user privileges and protects each user resources from unauthorized access by other users
- The kernel manages also the system environment for each user







- The kernel enables the system to run multiple applications (processes) at the same time
- Each Application thinks that it owns the whole system
- Each process may even have multiple threads that run simultaneously
- The kernel has a SCHEDULER that provides time slots to each process/thread in a pre-determined order to enable multi-tasking
- The order of time slot assignment and the time slot size is based on the SCHEDULING ALGORITHM

# Process Management



#### Process Scheduling Algorithm





- There are multiple scheduling techniques:
  - Round robin
  - Priority Based
- Processes may block on events such as hardware job completion, other process will take the processor cycles

# Process Management Preemption

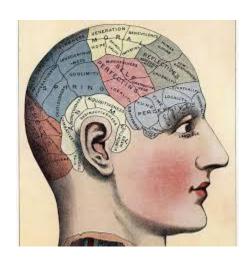




- The scheduler can be:
  - Pre-emptive
    - A higher priority task can interrupt the low priority task before it completes its job
  - Non-preemptive
    - Once a process takes control, it has to finish its job before it releases control







- When the processor is running multiple processes, each process will need to have its own memory
- The kernel works with the Processor to provide two functionalities:
  - Memory Protection (MPU)
    - Protects each process memory from being corrupted by the other processes
  - Memory Management (MMU)
    - Each process will think that it owns all the system memory



## File-System Management



- The kernel performs jobs related to handling of storage devices
- It gives the applications the feel of files and directories and hide the hardware details behind all of that







- The kernel handles power related functionality such as:
  - Managing sleep modes of the processor
  - Adjusting the processor clock rate based on its load



# I/O and Interrupt Handling



- Processor is connected to several I/O devices
- Those devices may use Interrupts to get the processor/kernel attention
- Kernel handles the interrupts received from Input/Output devices



#### Timers and Time Related Tasks



- The Processor is interrupted per time tick
- Kernel uses this interrupt to keep track of time
- Tick duration defines the time resolution for the kernel/system
- Kernel is responsible to keep track of time:
  - Handle timers
  - Handle time of day clock
  - Handle deferred jobs

## Networking and Communication





- The kernel is responsible to enable communication via:
  - Wired / wireless networking
  - USB/SDIO/Serial/...
  - Other communication means

