SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT312 – EMBEDDED SYSTEM DESIGN

III YEAR/ VI SEMESTER

UNIT 4 : EMBEDDED OPERATING SYSTEM AND MODELING

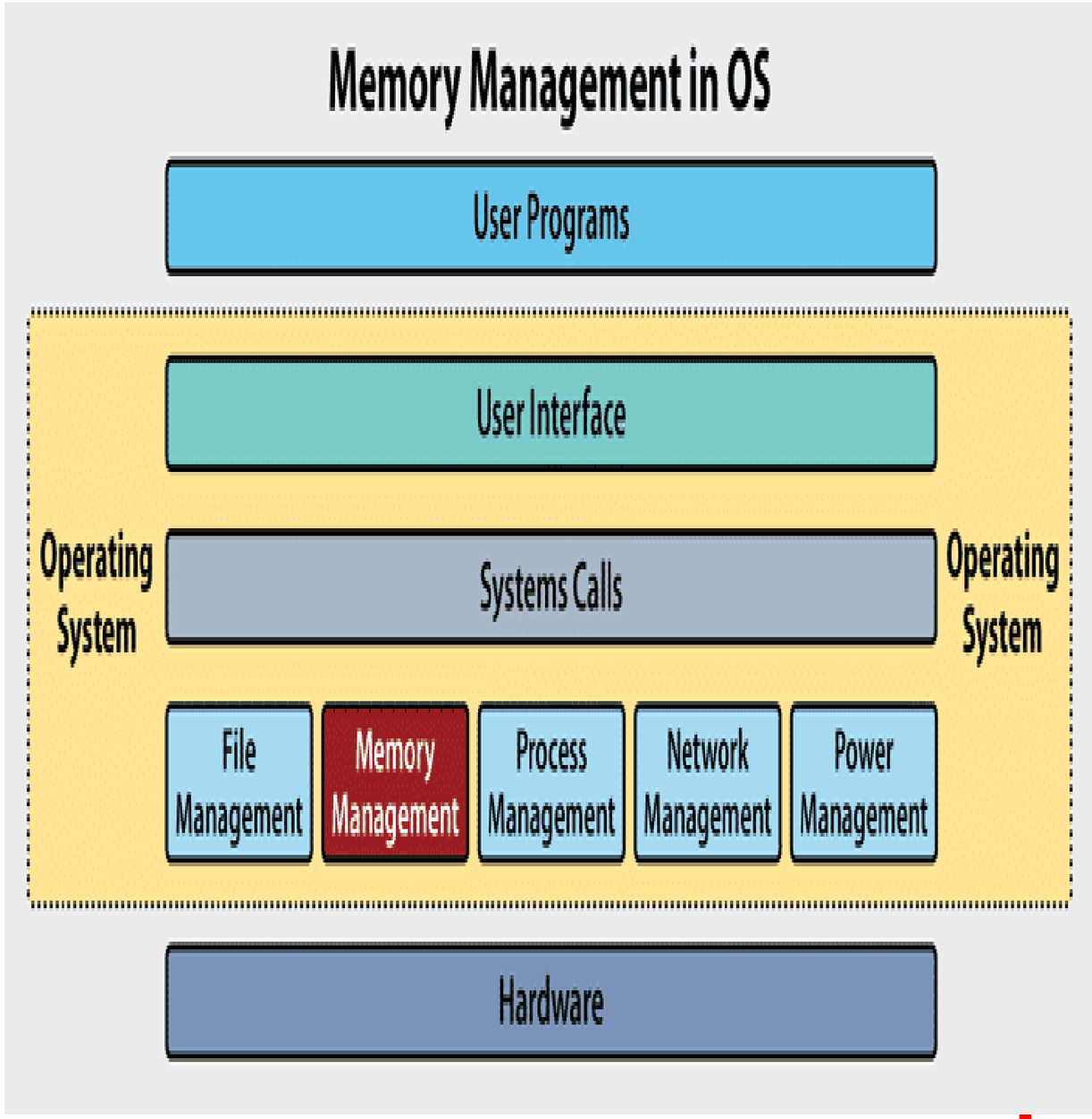
TOPIC 4.3- Memory Management

MEMORY MANAGEMENT

- Memory management is all about making sure that there is as much available memory space as possible for new programs, data and processes to execute

- As memory is used by multiple parts of a modern system, memory allocation and memory management can take on different forms like - Operating systems, Programming Languages, Applications and storage memory management.

- To be effective, a computer's memory management function must sit between the hardware and the operating system



Why do We Need Memory Management?

- Memory management is an essential element of all modern computing systems
- With the continued used of virtualization and the need to optimize resource utilization, memory is constantly being allocated, removed, segmented, used and re-used
- With memory management techniques, memory management errors, that can lead to system and application instability and failures can be mitigated

How Does Memory Management Work?

- Memory management is all about allocation and optimization of finite physical resources. Memory is not uniform – for example a 2GB RAM DIMM is not used as one large chunk of space.

- Rather memory allocation techniques are used for segmentation of RAM into usable blocks of memory cache.

- Memory management strategies within an operating system or application typically involve an understanding of what physical address space is available in RAM and performing memory allocation to properly place, move and remove processes from memory address space

Types of Memory Addresses

- Static and dynamic memory allocation in an operating system is linked to different types of memory addresses

- Fundamentally, there are two core types of memory addresses

Physical addresses – The physical address is the memory location within system RAM and is identified as a set of digits.

Logical addresses – referred to as virtual memory, a logical address is what operating systems and applications access to execute code, as an abstraction of physical address space

Memory Allocation: Static vs. Dynamic Loading

Applications and data can be loaded into memory in a number of different ways, with the two core approaches being static and dynamic loading.

- **Static Loading** – Code is loaded into memory, before it is executed. Used in structured programming languages including C

- **Dynamic Loading** – Code is loaded into memory as needed. Used in object oriented programming languages, such as Java

Memory Fragmentation

When memory is allocated in a system, not all of the available is always consumed in a linear manner, which can lead to fragmentation.

There are two core types of memory fragmentation, internal and external

Internal Fragmentation – Memory is allocated to a process or application and isn’t used, leaving un-allocated or fragmented memory

External Fragmentation – As memory is allocated and then deallocated, there can be small spaces of memory leftover, leaving memory holes or “fragments” that aren’t suitable for other processes.

Paging

- Within logical address space, virtual memory is divided up using *paging*, meaning it's divided into fixed units of memory, referred to as *pages*

- Pages can have different sizes, depending on the underlying system architecture and operating system

- The process of page table management can be intricate and complex.

Segmentation

- Memory segmentation within the primary memory of a system is a complicated process that references specific bits within a memory unit
- Each segment within system memory gets its own address, in an effort to improve optimization and memory allocation
- Segment registers are the primary mechanism by which modern systems handle memory segmentation
- Paging and segmentation are similar but do have a few distinct differences.

Swapping

- Swapping is the process by which addition memory is claimed by an operating system from a storage device
- “swap space,” that is storage space where memory process will be stored and run as physical and virtual memory space is exhausted, released and reclaimed
- The usage of swap space with traditional storage is a sub-optimal way of expanding available memory as it incurs the overhead of transferring to and from physical RAM.
- Additionally, traditional storage devices run with slower interface speeds that RAM.

Advantages and Disadvantages

Advantages

Maximizes the availability of memory to programs

Enables re-use and reclamation of memory that is not actively in use

Can help to extend available physical memory with swapping

Disadvantages

Can lead to fragmentation of memory resources

Adds complexity to system operations

Introduces potential performance latency

THANK YOU