



Ceng 111 – Fall 2021

Week 5a

Digital Computation

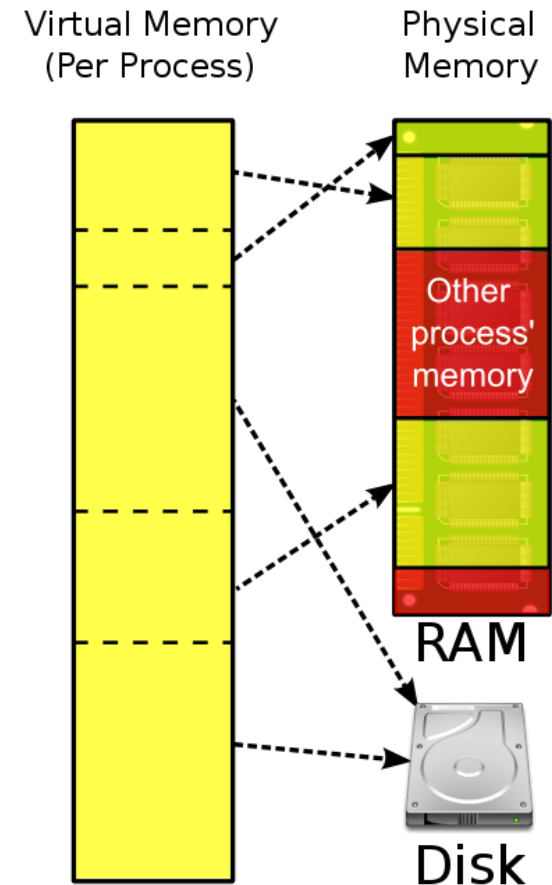
Credit: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.



Virtual Memory

- Paging -

- A **page**: at least 4KB of consecutive virtual memory addresses.
- An application program is stored in a set of pages.
- A **page table** maps the **logical/virtual addresses** of the pages with the **physical addresses**.

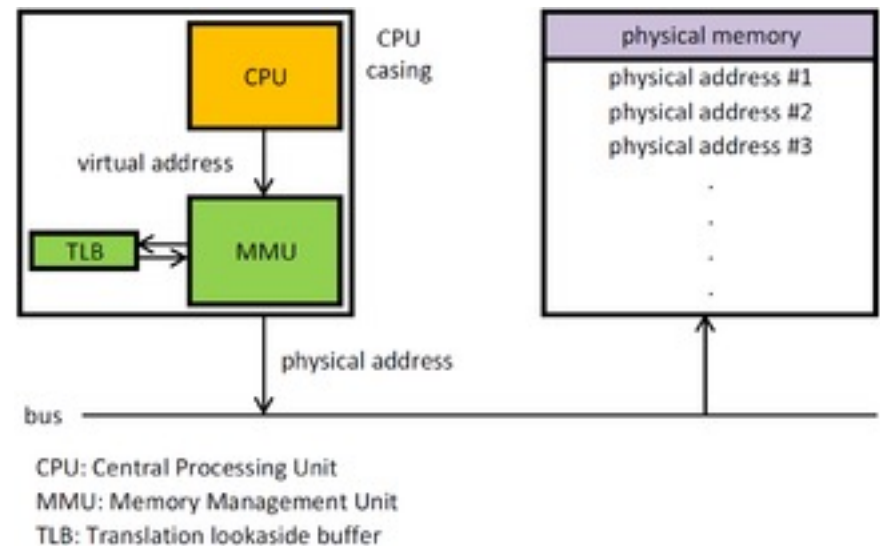


http://en.wikipedia.org/wiki/Virtual_memory



Memory Management Unit (MMU)

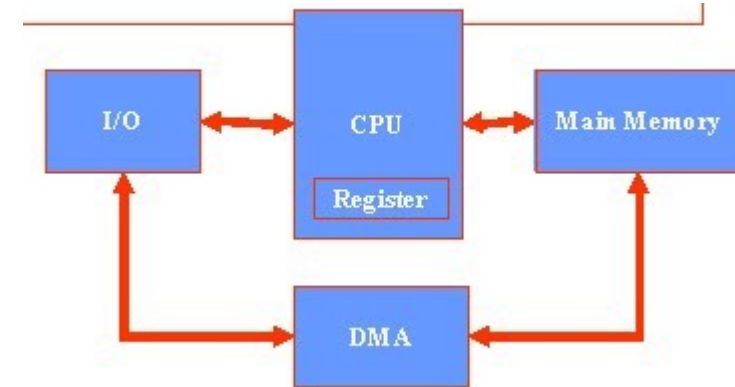
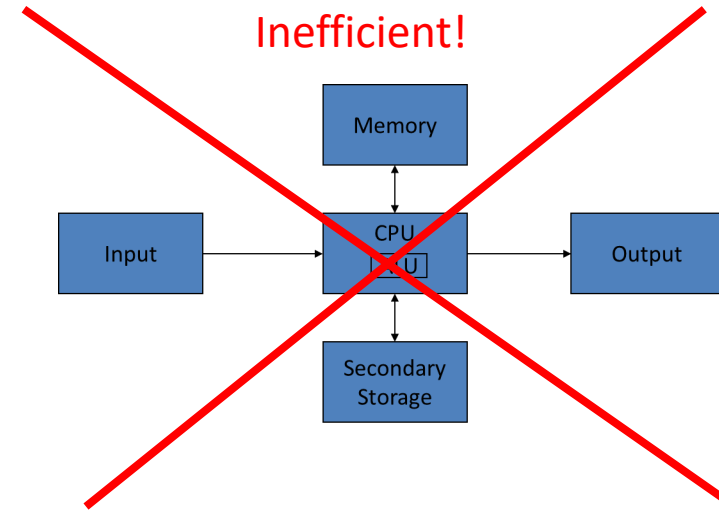
- Stores a lookup table in memory or a small device called Translation Lookaside Buffer (TLB)
- Can keep track of whether a page is on RAM, in use, last-used, updated, ..
- Non-contiguous physical addresses can be mapped to contiguous virtual addresses



https://en.wikipedia.org/wiki/Memory_management_unit

Direct Memory Access (DMA)

- All memory accesses go over the CPU.
 - CPU gets the virtual address,
 - It looks up the physical address,
 - It writes the data to the physical device (memory, in most cases)
- This slows down the CPU for simple and consecutive memory accesses.
- “Modern” computers have a channel for DMA.
- The CPU just initiates the DMA command, and a separate unit (DMA controller) handles the copy of data between a device and a memory.





Previously on CENG111!

I/O

■ Memory-mapped I/O

- Treats I/O devices as if they are memory
- “Load” and “Store” instructions are used for accessing an I/O device
- Single physical address space for memory and I/O devices
- Devices: Graphics card, hard drive

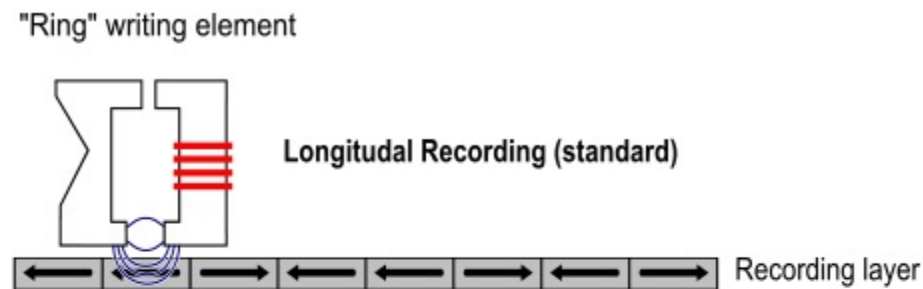
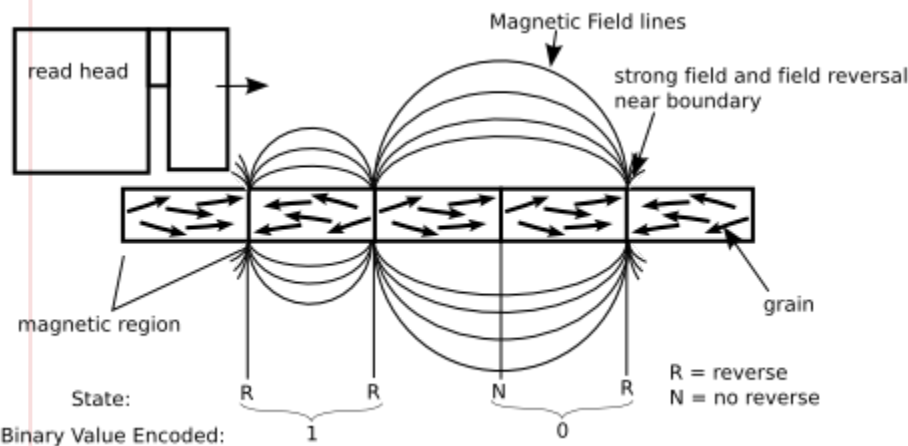
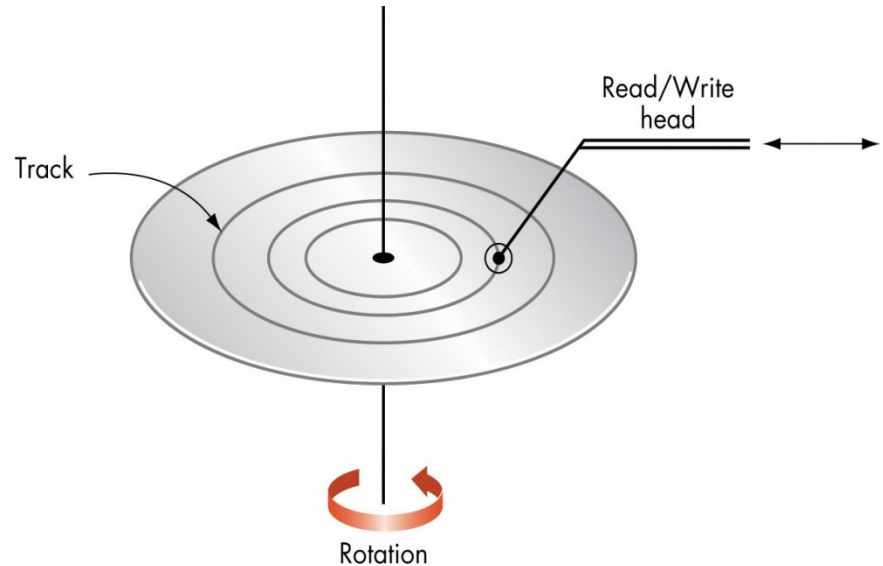
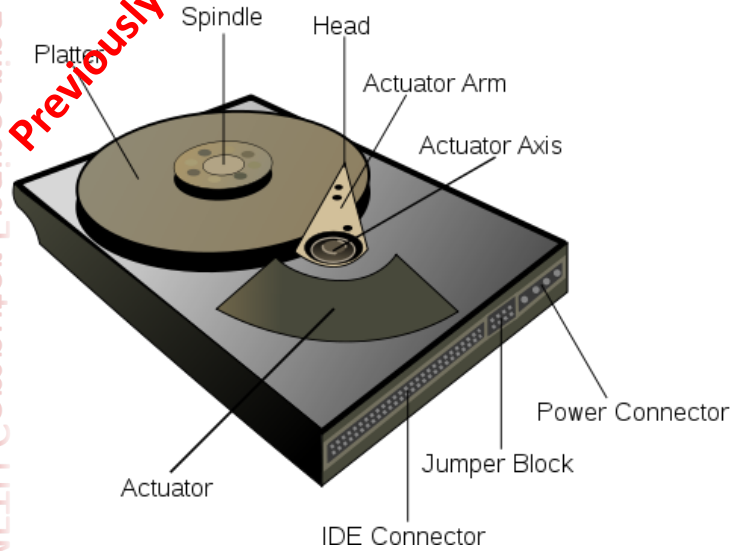
■ Port-mapped I/O (or isolated I/O)

- Special instructions for performing I/O
- Separate bus/pin for I/O devices
- Devices: Keyboard, mouse, ...



Rotational Media as External Storage

Previously on CENG111!





Transfer of Information

The Concept of Ports

- What is a port, literally?
- What is a computer port, then?
 - SATA, e-SATA, USB, Firewire, Serial, Parallel (e.g., PS/2) ports.





Interrupts

- Polling vs. interrupts
- Handling interrupts
- Interrupt table
- Different types of interrupts
- Timer interrupt
- Masking interrupts
- Interrupts & multi-tasking



Today

- Booting a computer
- Operating System



Administrative Notes

- Midterm date:
 - 22 December, Wednesday, 18:00



What happens after that power button

BOOTING YOUR COMPUTER

For more details, see e.g.: <https://neosmart.net/wiki/mbr-boot-process/>



Booting the Computer

- BIOS (Basic Input-Output System)
 - BIOS is on a Read-Only Memory (ROM)
 - When starting, CPU addresses the address space of the ROM and starts executing the BIOS.
 - BIOS first initiates **self-check of hardware** (Power-on self test – POST)
 - BIOS initiates the memory and loads the rest of the BIOS into the RAM (because ROM is slow)
 - Then, it goes over non-volatile storage devices to find something bootable.
- Boot Device & MBR
 - BIOS loads MBR from the boot device
 - MBR is not OS specific
 - MBR checks for a partition to boot
 - Loads OS (starts OS kernel)

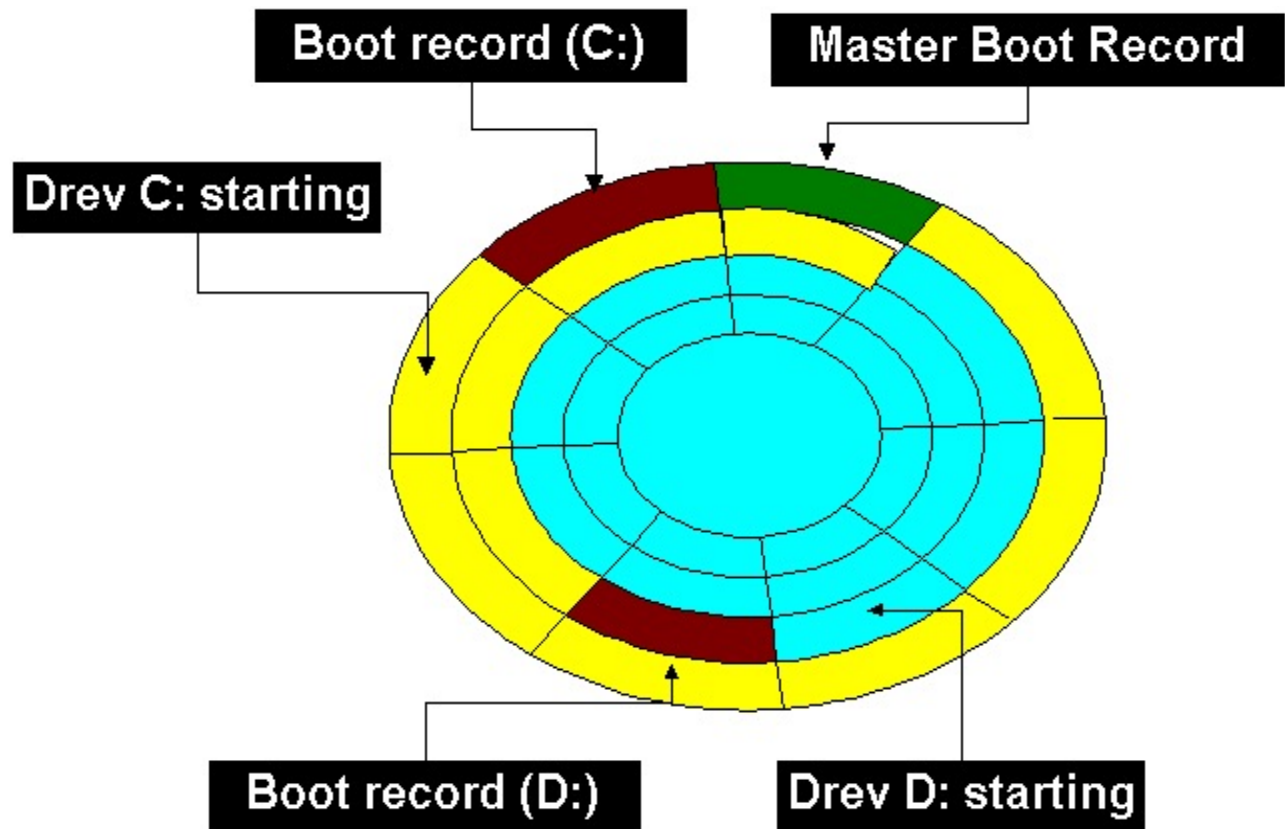


Master Boot Record

- First 512-bytes of a partitioned data storage device; e.g., a disk.
- Responsibilities:
 - Holding the partition table
 - Helps booting of operating systems
 - Stores the ID of the disk (not used)
- A part of these 512-bytes is a code that searches the “**primary**” partitions and finds the one that is “**active**”. The “**volume boot record**” of the active partition boots the OS.



Master Boot Record





Master Boot Record

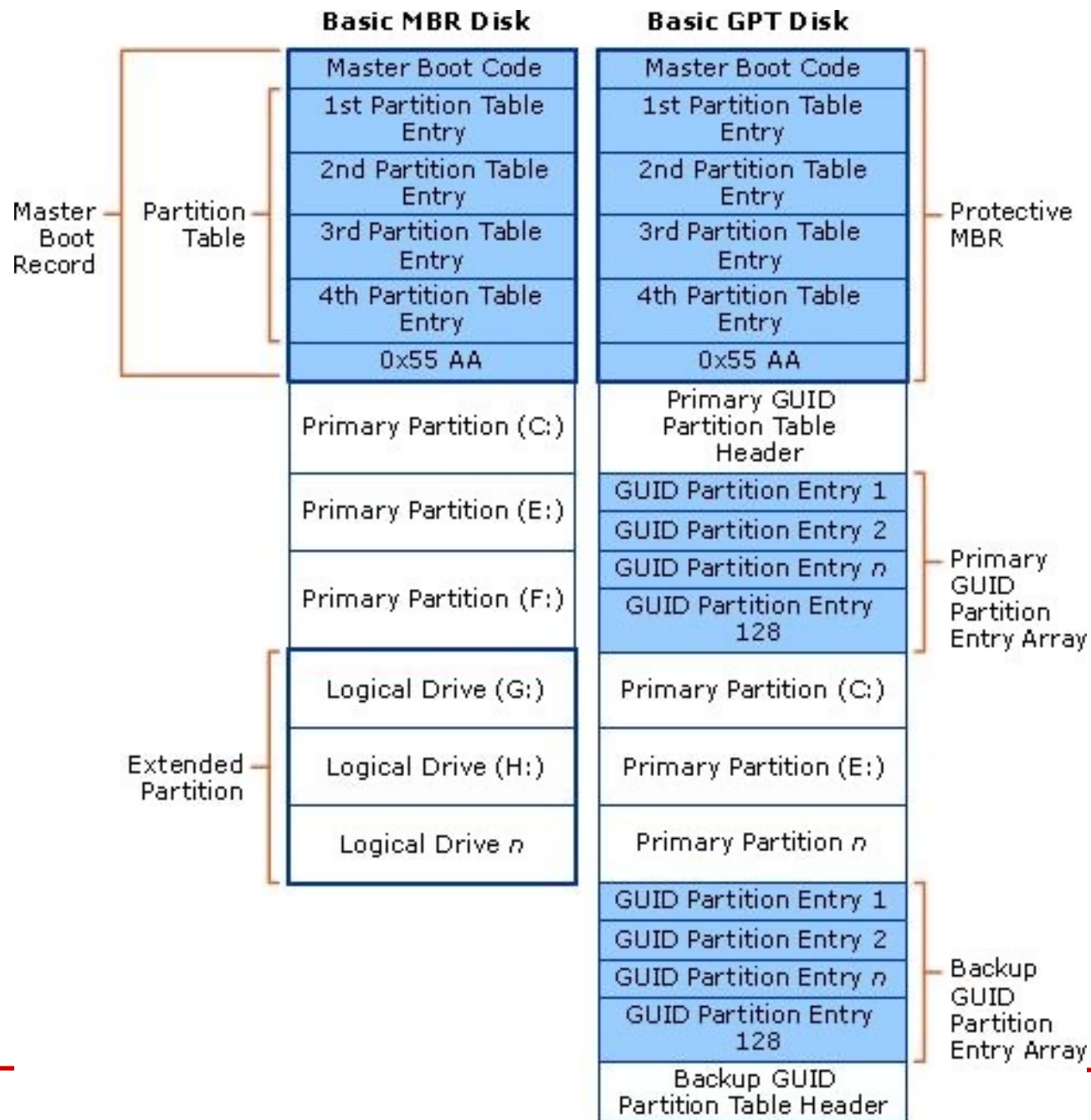
Structure of a Master Boot Record

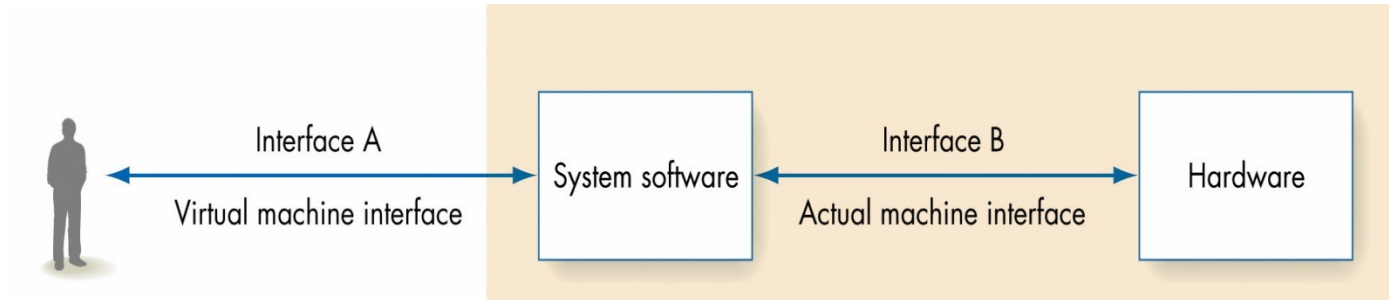
Address			Description		Size in bytes
Hex	Oct	Dec			
0000	0000	0	code area		440 (max. 446)
01B8	0670	440	disk signature (optional)		4
01BC	0674	444	Usually nulls; 0x0000		2
01BE	0676	446	Table of primary partitions (Four 16-byte entries, IBM partition table scheme)		64
01FE	0776	510	55h	MBR signature; 0xAA55 [1]	2
01FF	0777	511	AAh		
MBR, total size: 446 + 64 + 2 =					512



UEFI & GPT

- Unified Extensible Firmware Interface
 - Extends BIOS
 - managed by a group of chipset, hardware, system, firmware, and operating system vendors called the UEFI Forum
 - Faster hardware check compared to BIOS
 - Allows developers to add applications, enabling UEFI to be a lightweight OS
 - Provides secure boot, disables unauthorized applications from booting
 - Provides GUI
- GPT (GUID Partition Table) – required by UEFI
 - Extends MBR
 - GUID: Globally (Universally) unique identifier. Easily creatable almost unique identifier for partitions
 - Supports bigger disks (> 2TB), more partitions (up to 128)
 - It has backup at the end, which can help recover partitions





OPERATING SYSTEM



Introduction

- Von Neumann computer
 - “Naked machine”
 - Hardware without any helpful user-oriented features
 - Extremely difficult for a human to work with
- An **interface** between the user and the hardware is needed to make a Von Neumann computer usable



Introduction (continued)

- Tasks of the interface
 - **Hide details** of the underlying hardware from the user
 - Present information in a way that does not require in-depth knowledge of the internal structure of the system



Introduction (continued)

- Tasks of the interface (continued)
 - **Allow easy user access** to the available resources
 - **Prevent accidental or intentional damage** to hardware, programs, and data



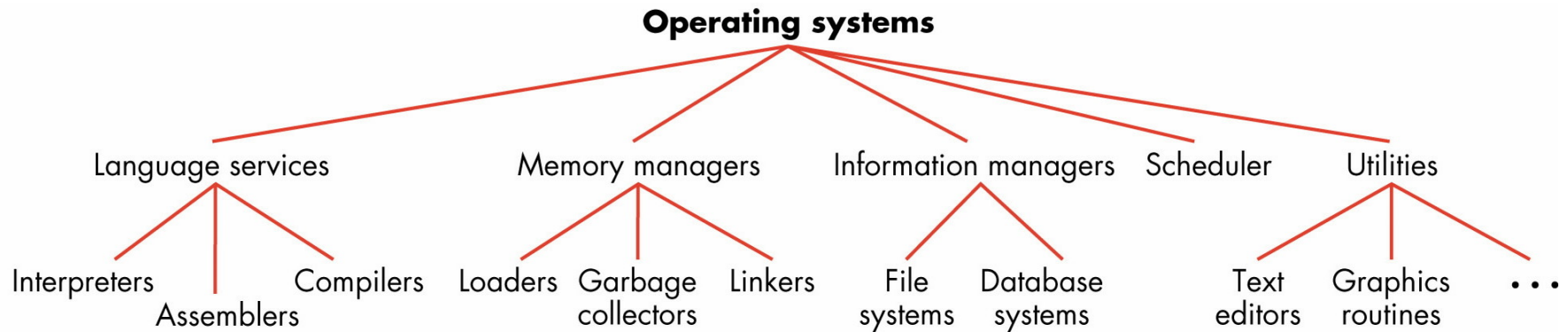
System Software: The Virtual Machine

- System software
 - Acts as an intermediary between users and hardware
 - Creates a virtual environment for the user that hides the actual computer architecture
- Virtual machine (or virtual environment)
 - Set of services and resources created by the system software and seen by the user



Types of System Software

- System software is a collection of many different programs
- Operating system
 - Controls the overall operation of the computer
 - Communicates with the user
 - Determines what the user wants
 - Activates system programs, application packages, or user programs to carry out user requests



Responsibilities of an OS



Types of System Software (continued)

- User interface
 - Graphical user interface (GUI) provides graphical control of the capabilities and services of the computer
- Language services
 - Assemblers, compilers, and interpreters
 - Allow you to write programs in a high-level, user-oriented language, and then execute them



Types of System Software (continued)

- Memory managers
 - Allocate and retrieve memory space
- Information managers
 - Handle the organization, storage, and retrieval of information on mass storage devices
- I/O systems
 - Allow the use of different types of input and output devices



Types of System Software (continued)

■ Scheduler

- Keeps a list of programs ready to run and selects the one that will execute next

■ Utilities

- Collections of library routines that provide services either to user or other system routines