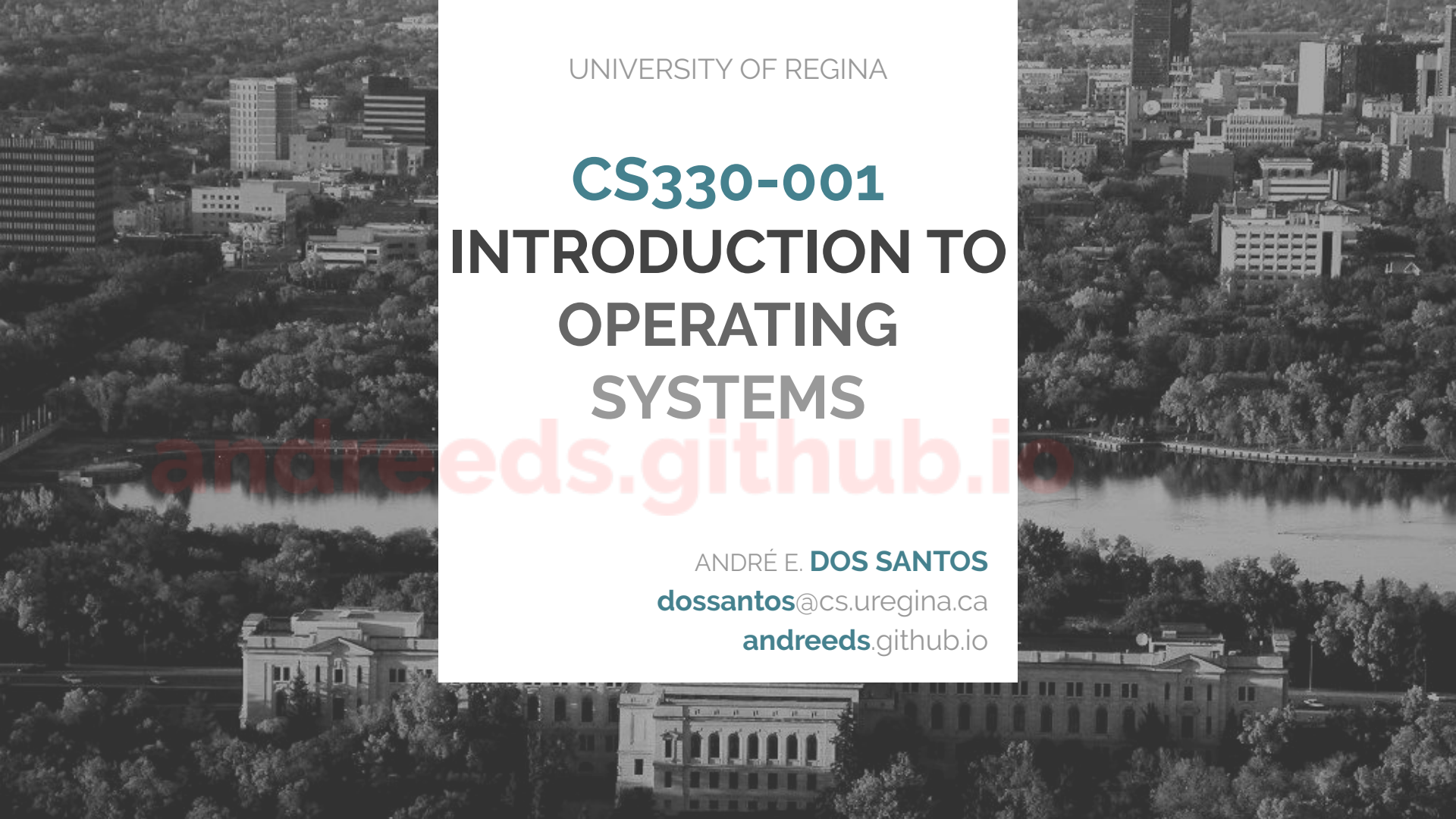
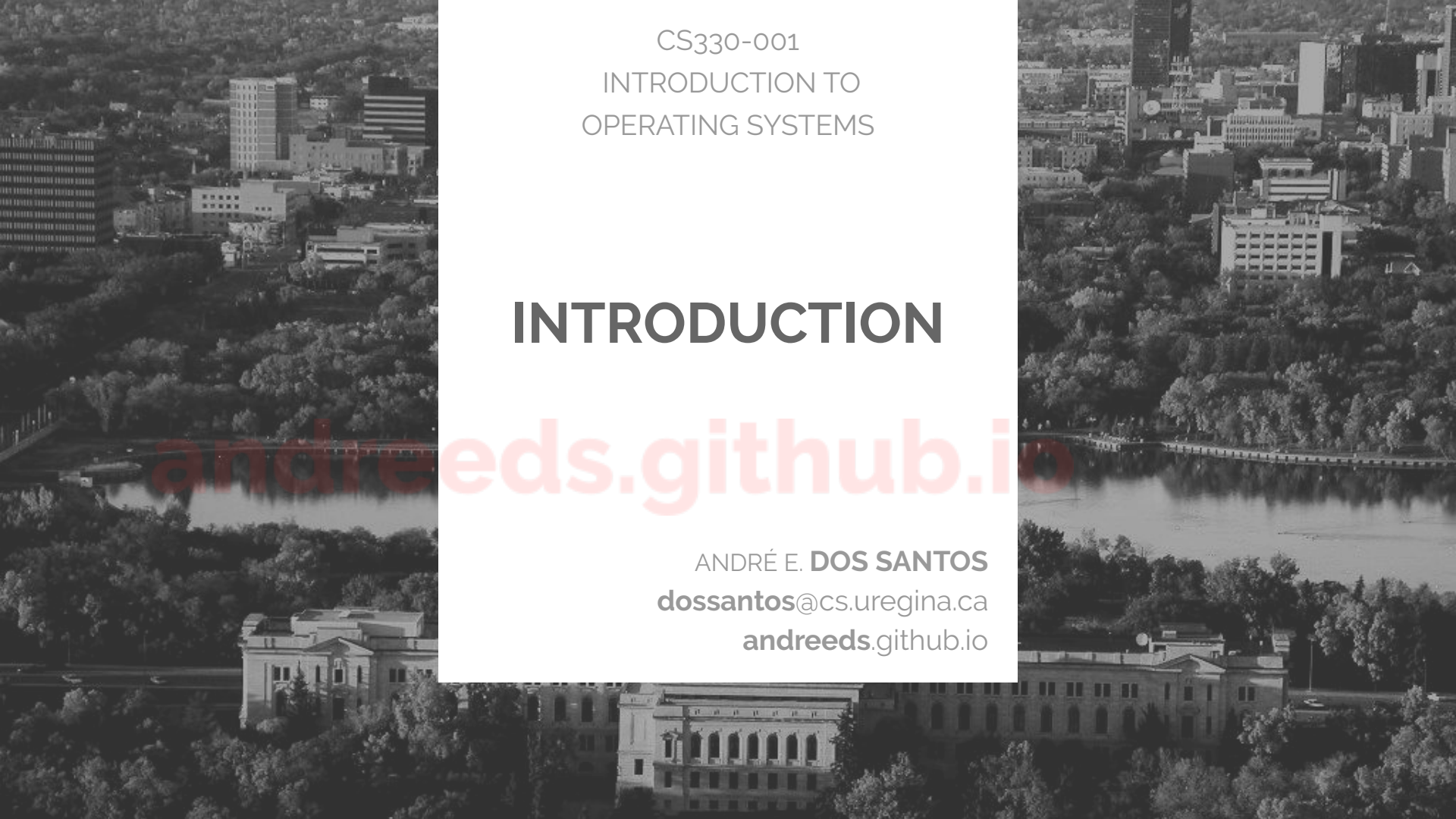


UNIVERSITY OF REGINA

CS330-001
INTRODUCTION TO OPERATING SYSTEMS

ANDRÉ E. **DOS SANTOS**
dossantos@cs.uregina.ca
andreeds.github.io



CS330-001
INTRODUCTION TO
OPERATING SYSTEMS

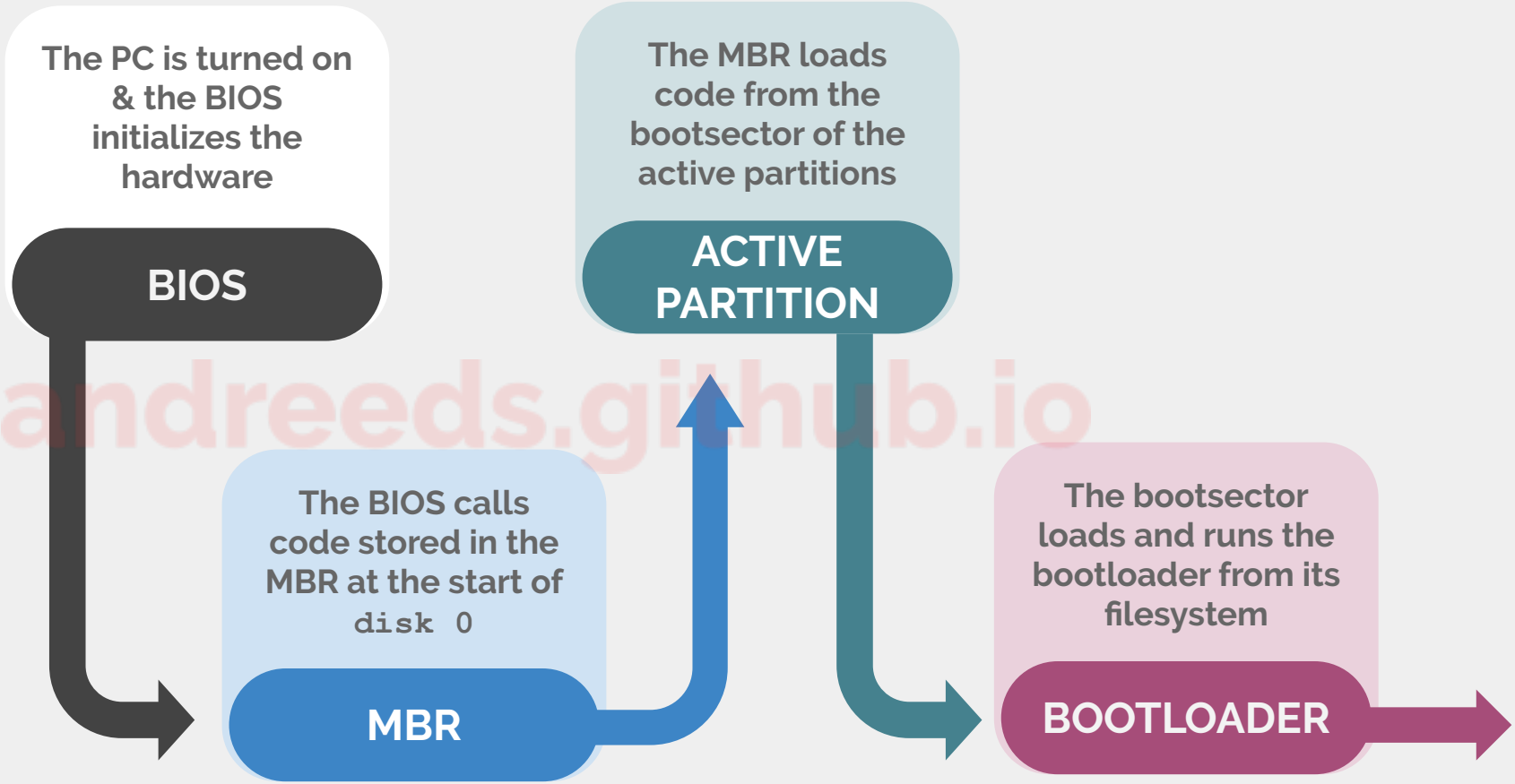
INTRODUCTION

ANDRÉ E. **DOS SANTOS**
dossantos@cs.uregina.ca
andreeds.github.io

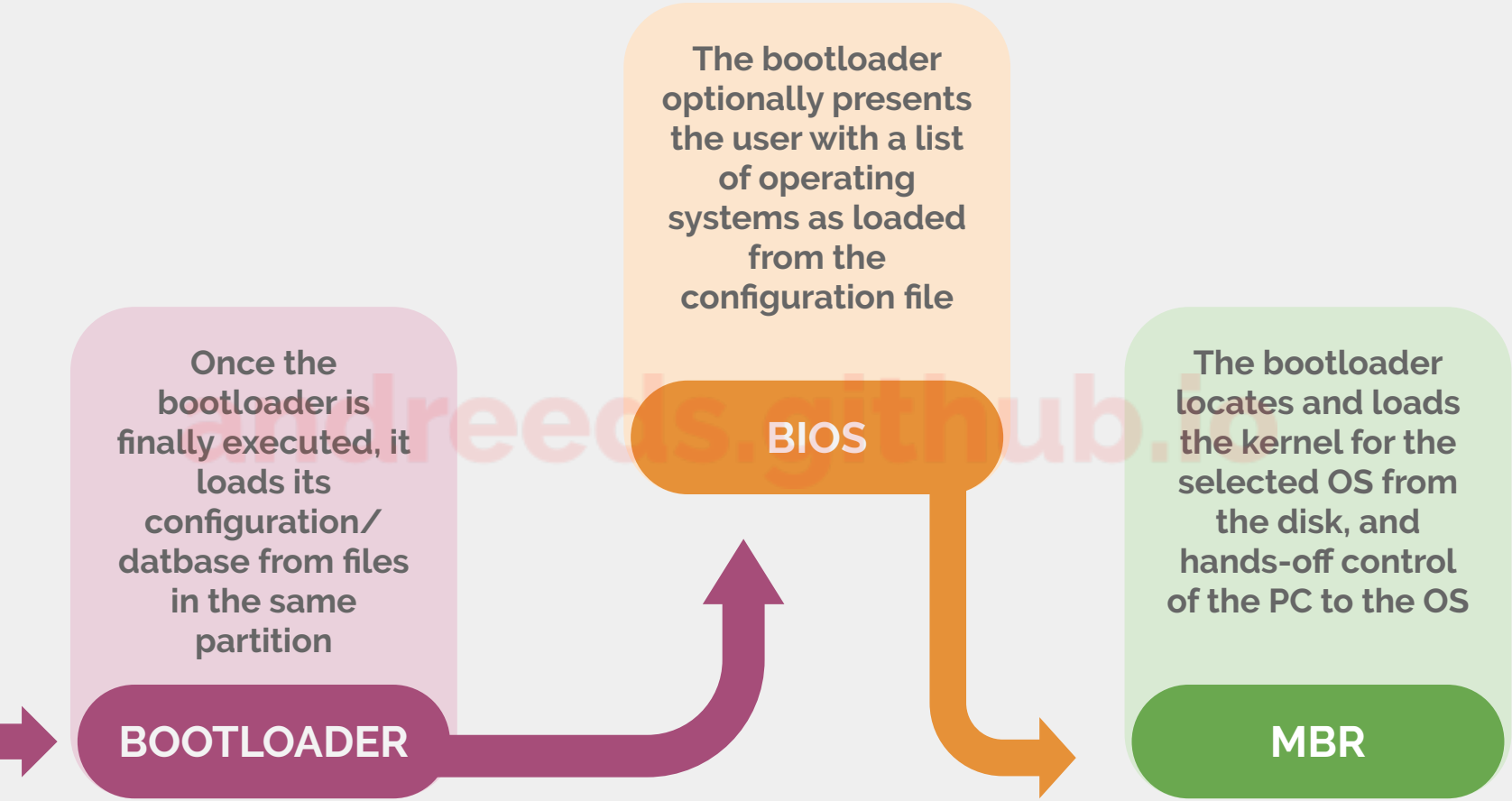
WHAT HAPPENS WHEN A COMPUTER IS POWERED UP?

https://www.youtube.com/watch?v=C_xxra5HMz0

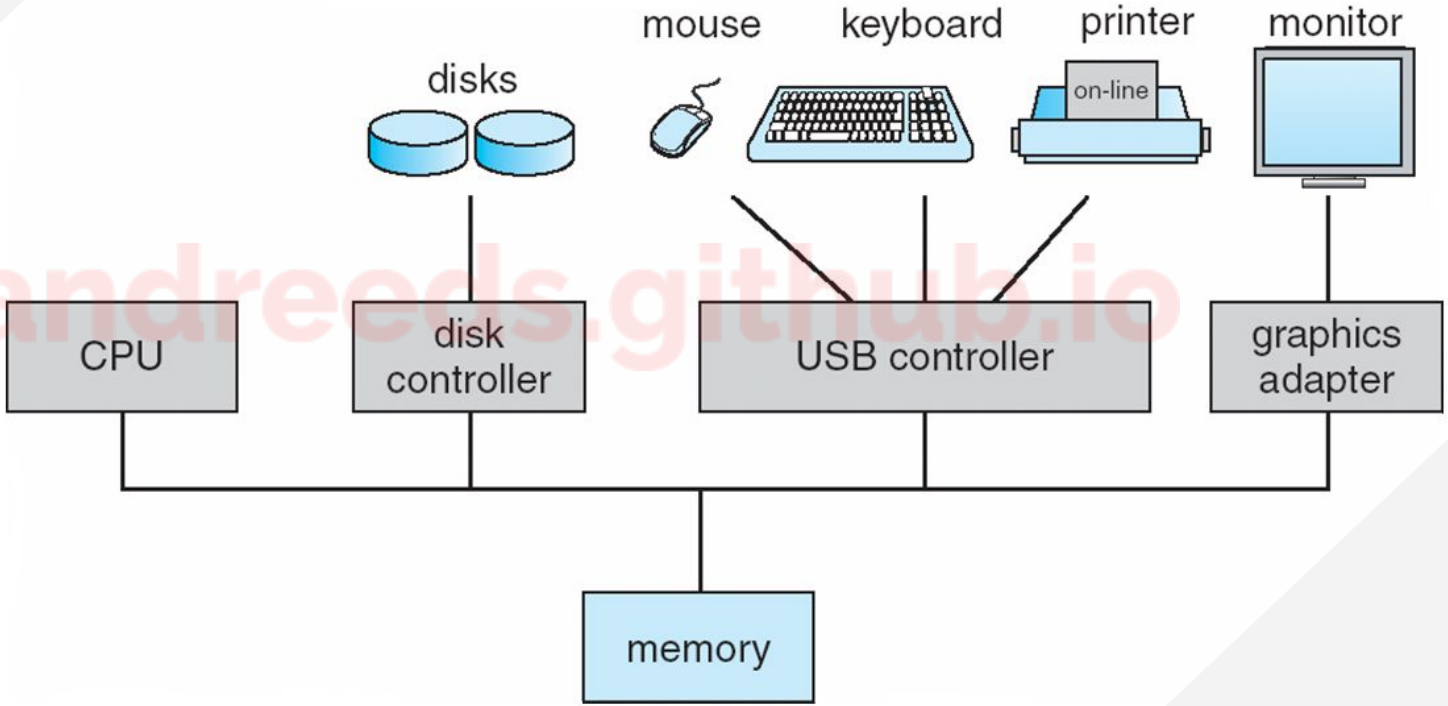
WHAT HAPPENS WHEN A COMPUTER IS POWERED UP?



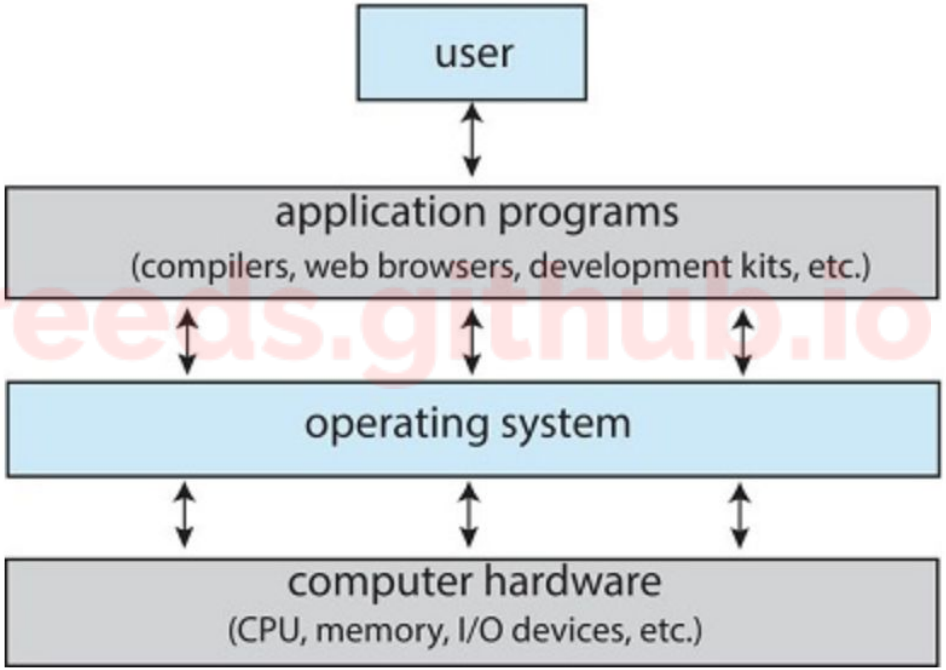
WHAT HAPPENS WHEN A COMPUTER IS POWERED UP?



COMPUTER SYSTEM ORGANIZATION



COMPUTER SYSTEM STRUCTURE



COMPUTER SYSTEM STRUCTURE

WHAT IS AN OPERATING SYSTEM?

- A program that acts as an intermediary between a user of a computer and the computer hardware

Purpose 1

- Provide **an environment** to:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

WHAT IS AN OPERATING SYSTEM?

Purpose 2

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

Purpose 3

- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

WHAT OSs DO

- Depends on the point of view
  - Convenient vs Efficient
  - **Users** want convenience, **ease of use** and **good performance**
    - Don't care about r**esource utilization**

OS DEFINITION

"Everything a vendor ships when you order an operating system"

OR

"The one program running at all times on the computer" is the kernel.

Everything **else** is either

- a **system program** (ships with the operating system) , or
- an **application** program.

Mobile operating systems often include not only a core kernel but also **middleware**—a set of software frameworks that provide additional services to application developers.

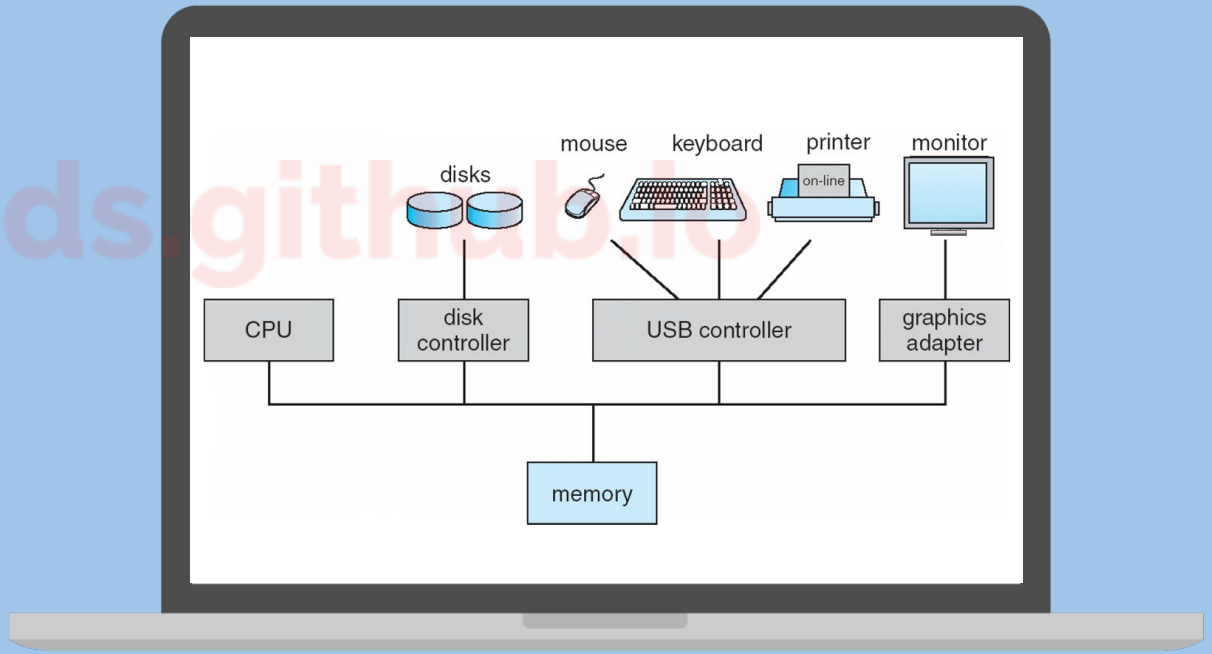
WHY STUDY OSs?

AND...?

all code runs on top of an operating system, knowledge of how operating systems work is crucial to proper, efficient, effective, and secure programming

KEY ASPECTS OF OS

Interrupts
Storage structure
I/O structure.



KEY ASPECTS OF OS

Interrupts

Storage structure

I/O structure

- Hardware interrupt
  - e.g. services requests of I/O devices
- Software interrupt
  - e.g. signals, invalid memory access, division by zero, system calls, etc
  - **trap**
- Procedures: generic handler or interrupt vector
  - MS-DOS,UNIX

KEY ASPECTS OF OS

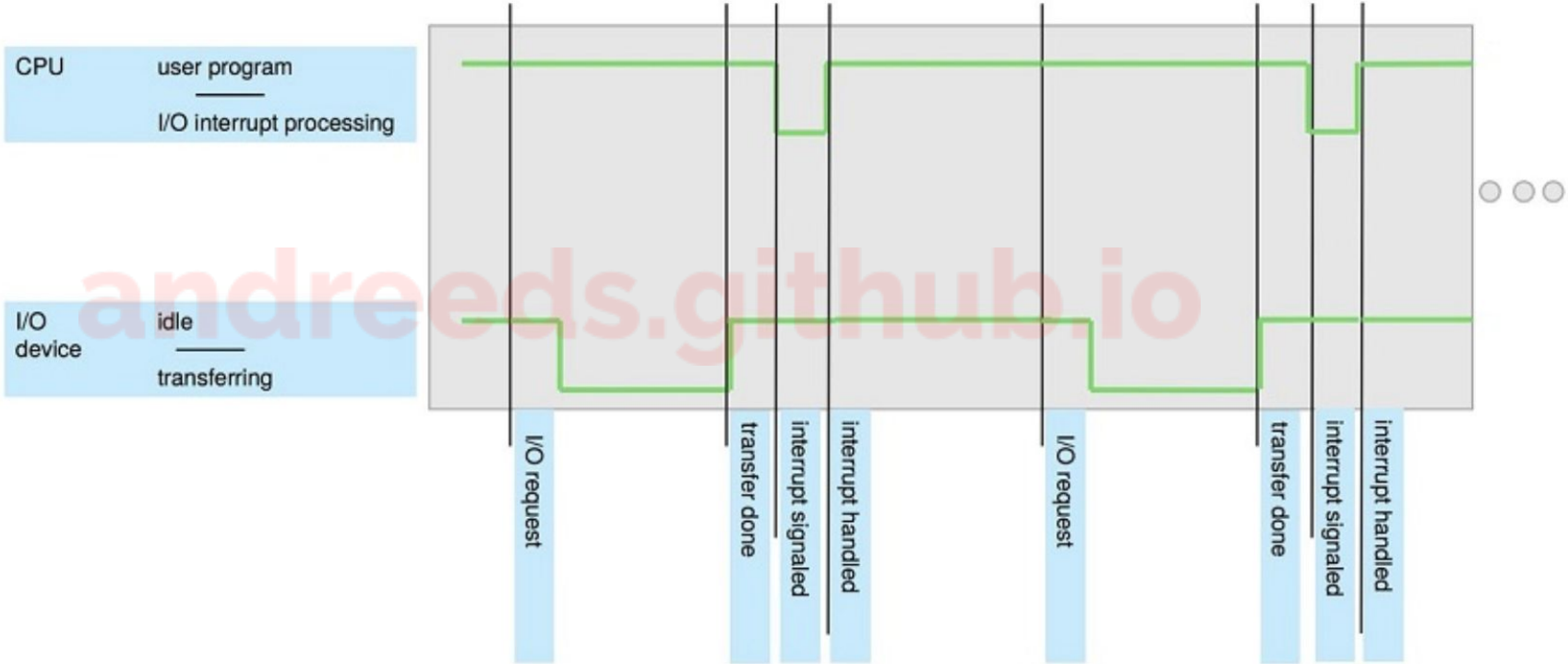
Interrupts

Storage structure

I/O structure

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**

INTERRUPT TIMELINE

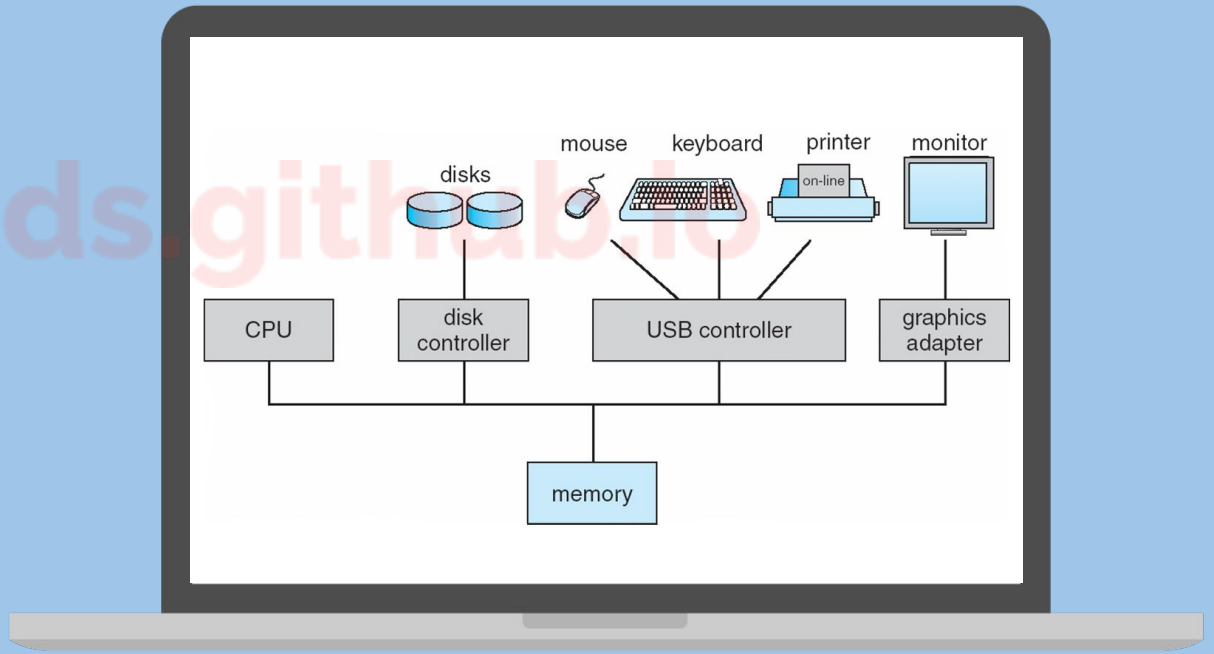


KEY ASPECTS OF OS

Interrupts

Storage structure

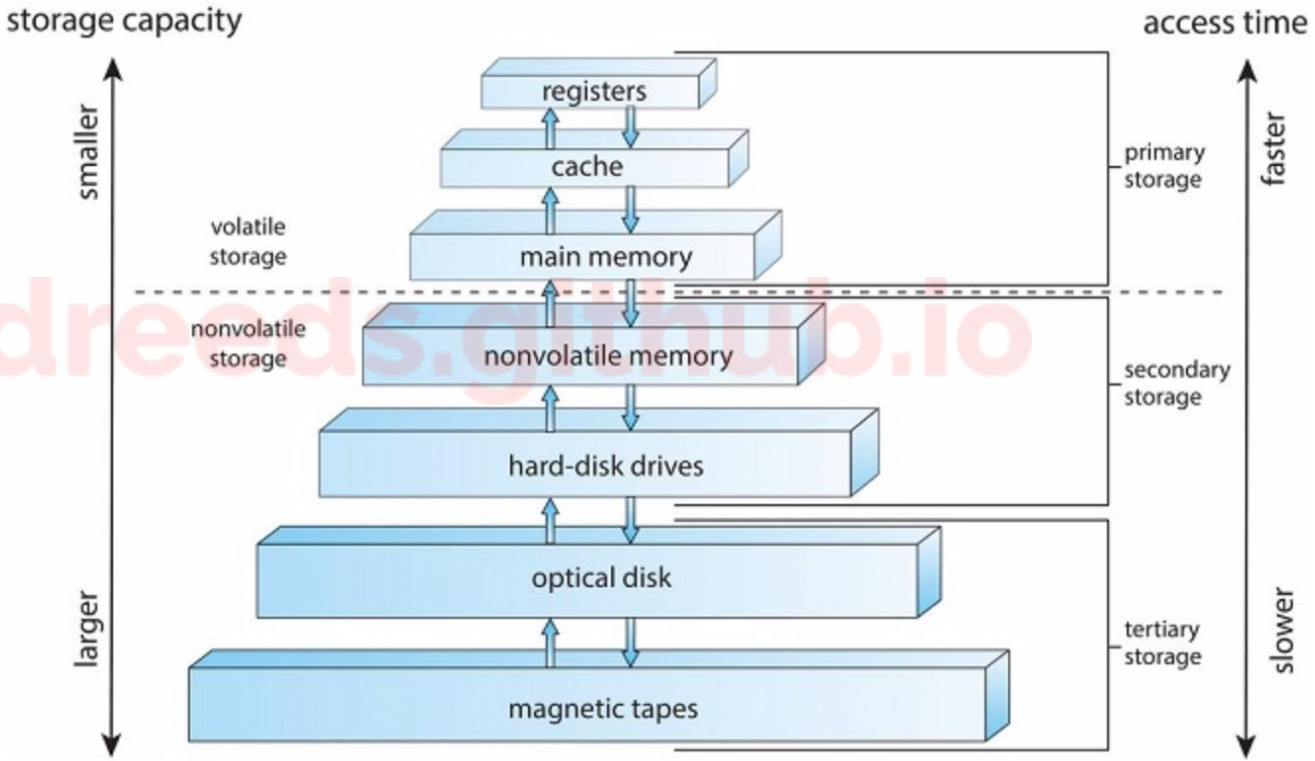
I/O structure.



KEY ASPECTS OF OS

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

STORAGE-DEVICE HIERARCHY

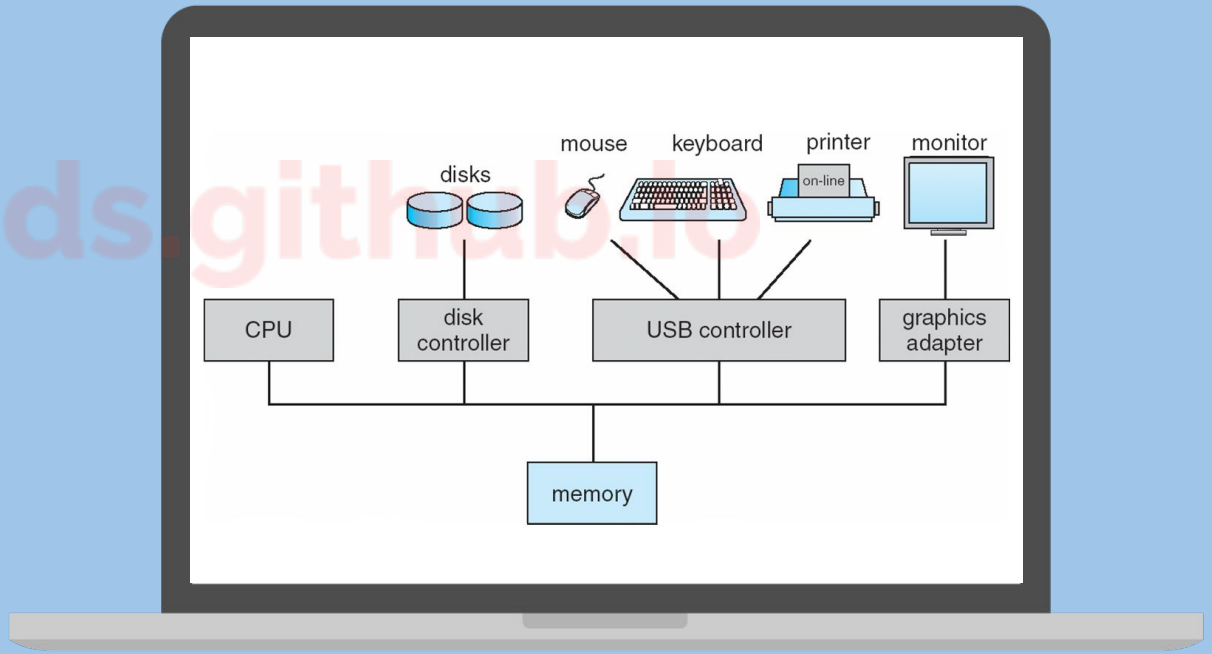


KEY ASPECTS OF OS

Interrupts

Storage structure

I/O structure.



KEY ASPECTS OF OS

Interrupts

Storage structure

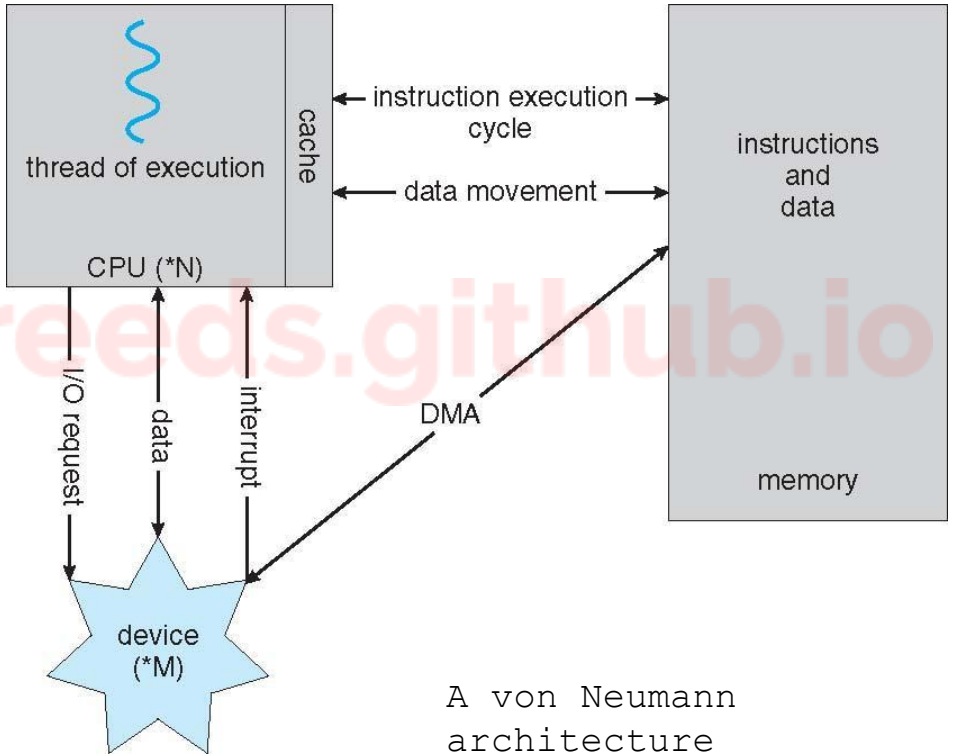
I/O structure

- Device controllers are responsible of moving data between the peripheral devices and their local buffer storages

Direct Memory Access (DMA) Structure

- Goal: Release CPU from handling excessive interrupts
- Procedure
  - Execute the device driver to set up the registers of the DMA controller
  - DMA moves blocks of data between the memory and its own buffer
  - Transfer from its buffers to its device
  - Interrupt the CPU when the job is done

HOW A MODERN COMPUTER WORKS



A von Neumann architecture

COMPUTER-SYSTEM ARCHITECTURE

- Single-Processor Systems
- Multiprocessor System
- Clustered Systems

Modern computer architectures are multiprocessor systems in which each CPU contains several computing cores

OS OPERATIONS

Definition

To best utilize the CPU, modern operating systems employ **multiprogramming**, which allows several jobs to be in memory at the same time, thus ensuring that the **CPU always has a job to execute**

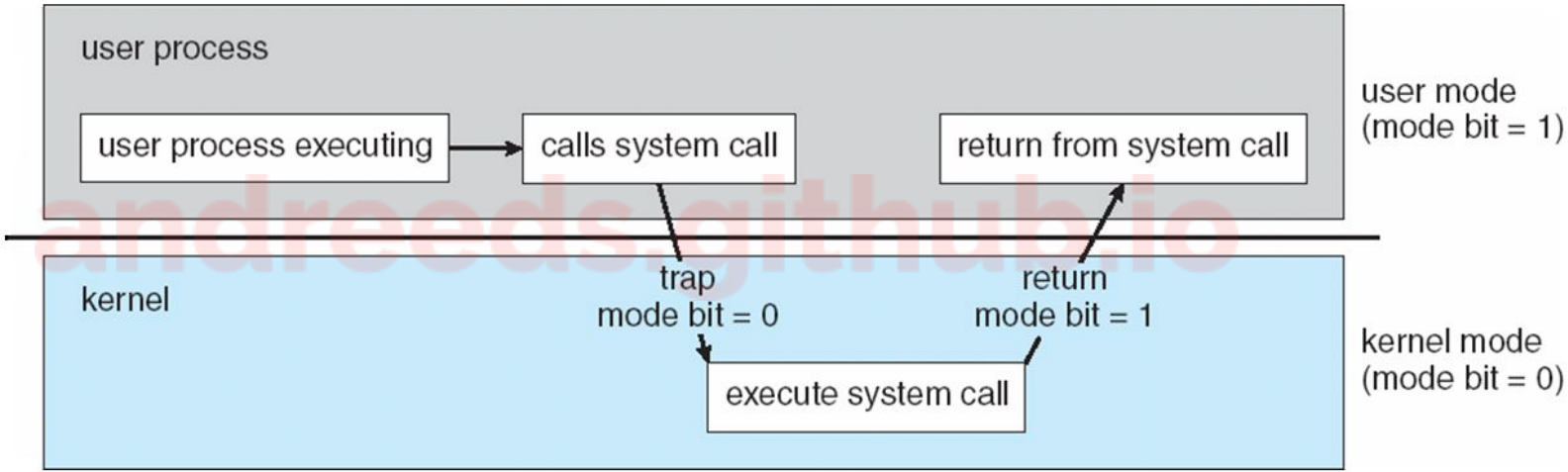
Definition

Multitasking is an extension of multiprogramming wherein **CPU scheduling** algorithms rapidly switch between **processes**, providing users with a fast response time

OS OPERATIONS

- **Dual-mode** operation allows OS to protect itself and other system components
  - To prevent user programs from interfering with the proper operation of the system
  - The system hardware has two modes:
    - **user mode** and **kernel mode**
  - **Various instructions are privileged and can be executed only in kernel mode**
    - e.g. *instruction to switch to kernel mode, I/O control, timer management, and interrupt management*

TRANSITION FROM USER TO KERNEL MODE



RESOURCE MANAGEMENT

- An operating system is a resource manager
- The system's CPU, memory space, file-storage space, and I/O devices are among the resources that the operating system must manage

PROCESS MANAGEMENT

- A process is a program in execution. It is a unit of work within the system
  - **Program is a passive entity, process is an active entity**
- Process management includes creating and deleting processes and providing mechanisms for processes to communicate and synchronize with each other.
  - Process needs resources to accomplish its task
    - CPU, memory, I/O, files
    - Initialization data
- Process termination requires reclaim of any reusable resources

PROCESS MANAGEMENT

Process Management Activities

- Creating and deleting both user and system processes
- **Suspending and resuming processes**
- Providing mechanisms for **process synchronization**
- Providing mechanisms for **process communication**
- Providing mechanisms for **deadlock handling**

MEMORY MANAGEMENT

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory
- Memory management determines what is in memory and when
- Optimizing CPU utilization and computer response to users

MEMORY MANAGEMENT

Memory Management Activities

- Keeping track of which parts of memory are currently being used and by whom
- Deciding which processes (or parts thereof) and data to move into and out of memory
- Allocating and deallocating memory space as needed

In summary, an OS manages memory by keeping track of what parts of memory are being used and by whom. It is also responsible for dynamically allocating and freeing memory space

FILE-SYSTEM MANAGEMENT

OS provides uniform, logical view of information storage

- Abstracts physical properties to logical storage unit - **file**
- Files usually organized into directories
- Access control on most systems to determine who can access what

FILE-SYSTEM MANAGEMENT

File-System Activities

- Creating and deleting files and directories
- Primitives to manipulate files and directories
- Mapping files onto secondary storage
- Backup files onto stable (non-volatile) storage media

MASS-STORAGE MANAGEMENT

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

MASS-STORAGE MANAGEMENT

Mass-Storage Activities

- Free-space management
- Storage allocation
- Disk scheduling

CACHE MANAGEMENT

Two important design issues for cache memory are

- size
- replacement policy

CACHE MANAGEMENT

Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

I/O SYSTEM MANAGEMENT

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering, caching, spooling
  - General device-driver interface
  - Drivers for specific hardware devices

PROTECTION AND SECURITY

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

PROTECTION AND SECURITY

- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

KERNEL DATA STRUCTURES

- Data structures that are used in an operating system include lists, stacks, queues, trees, and maps.

OPEN-SOURCE OSs

- Free and open-source operating systems are available in source-code format. Free software is licensed to allow no-cost use, redistribution, and modification. GNU/Linux, FreeBSD, and Solaris are examples of popular open-source systems.

REVIEW QUESTIONS
INTRODUCTION

- What are the three main purposes of an operating system?
- Two important design issues for cache memory are ____.
- A(n) _____ is the unit of work in a system.
- How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?
- Give two reasons why caches are useful. What problems do they solve? What problems do they cause?
- Which of the following instructions should be privileged?
    a. Set value of timer.
    b. Read the clock.
    c. Clear memory.
    d. Issue a trap instruction.
    e. Turn off interrupts.
    f. Modify entries in device-status table.
    g. Switch from user to kernel mode.
    h. Access I/O device.