Introduction to Scientific Computing I

Lecture 3

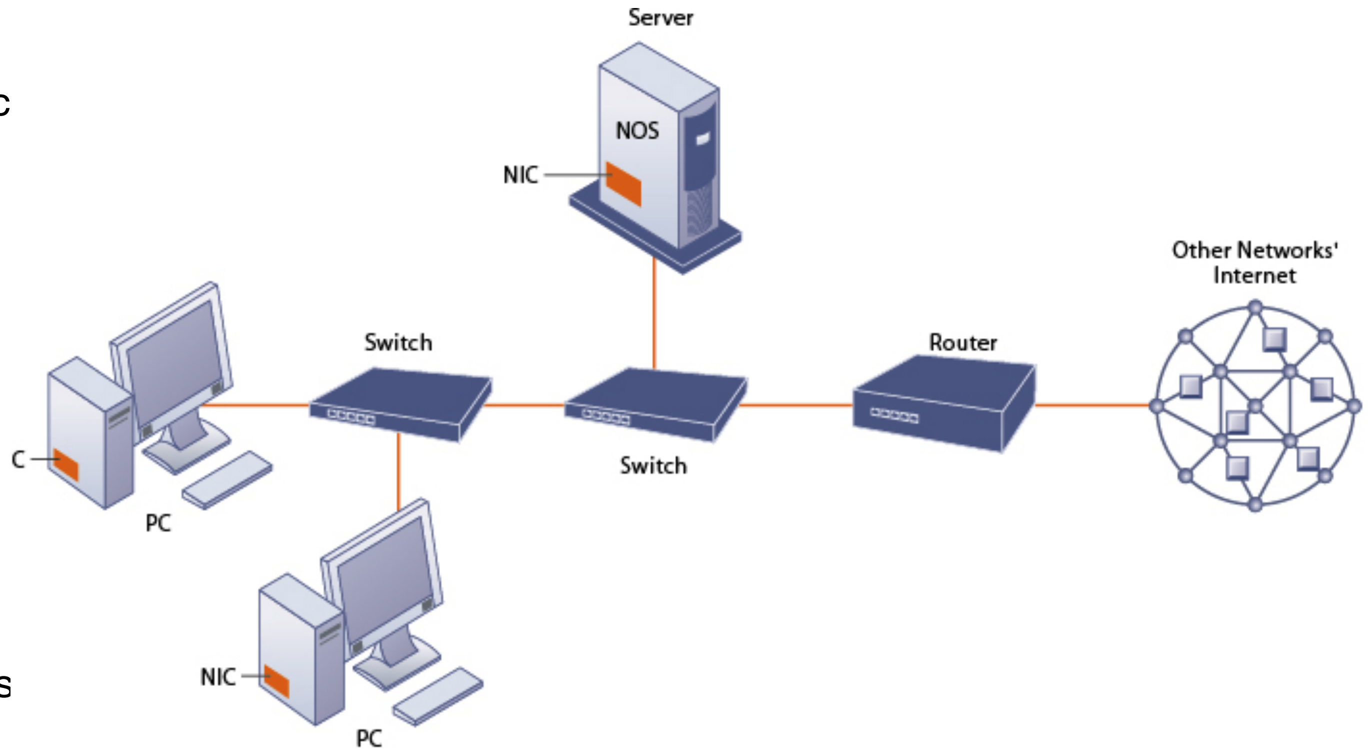
Amir Farbin

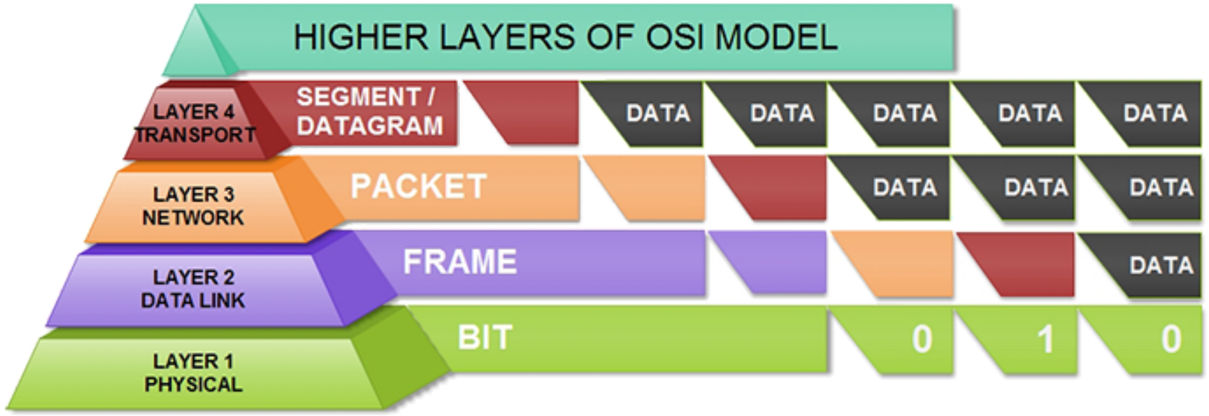
Announcements

- Lab 1 reminder
- Clinic reminder
- Laptops

Network

- Local Area Network (LAN) / Wide Area Networks (WAN)
- Signals sent via coaxial cable, twister pair, fiber optics, …
- Components: adapter, switch (connects computers), router (connec networks), wireless…
- Layers
    - Ethernet: (physical)
        - Rates now approaching 100 Gb
        - Every component has a unique address (48-bit MAC address)
        - Data broken into frames, with source/destination address and error checking data.
    - Network: for example Internet Protocol (IP)
        - Packets sent via IP address
        - Addresses kept in Domain Name System (DNS): Match name → address.
    - Transport:
        - How data is exchanged, broken up, transmitted, routed, …
        - Transmission Control Protocol (TCP): Services listen / communicate on ports.



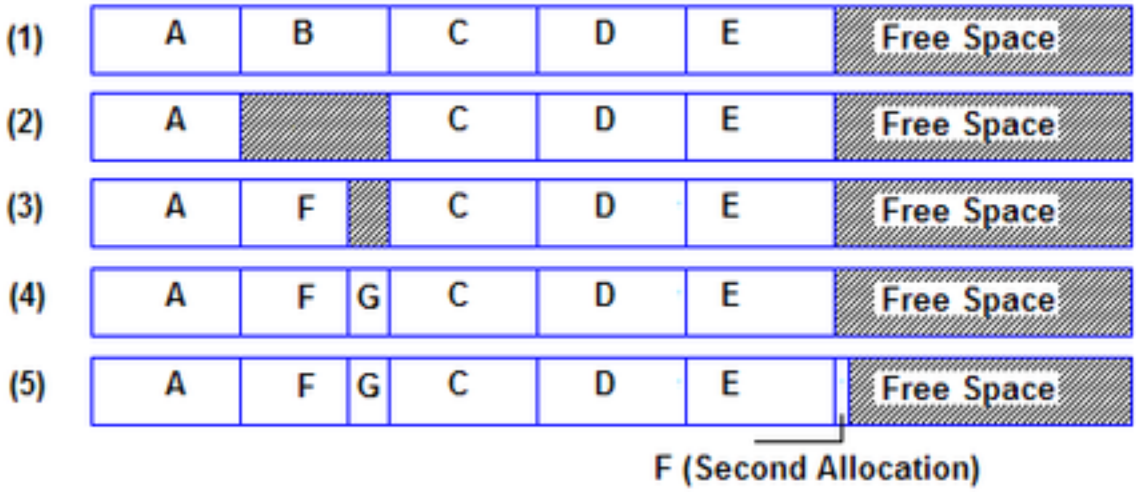


Storage

- Storage devices provide an interface to read / write data into specific locations of some in non-violative media.
    - On traditional ***Hard Drive***, the data is written magnetically on a spinning metal disk.
        - The disk is divided up into sectors, the minimum storage unit.
        - Each sector has an address, which corresponds its physical location on the disk.
    - ***Solid State Drives*** store data on silicon… originally presented same interface as HDs, but new interfaces such as non-volatile memory express (NVMe) are designed for SSDs.
    - In Unix these are referred to as: /dev/hda, /dev/hdb, … /dev/sda, /dev/sdb
- ***Partition***: The disk sectors are partitioned into groups of sectors, each where a different file system can be created.
    - Partition table: keeps track of the locations of the partitions.
    - /dev/hda1, /dev/sdb2
    - Partitions can be further divided into logical partitions.

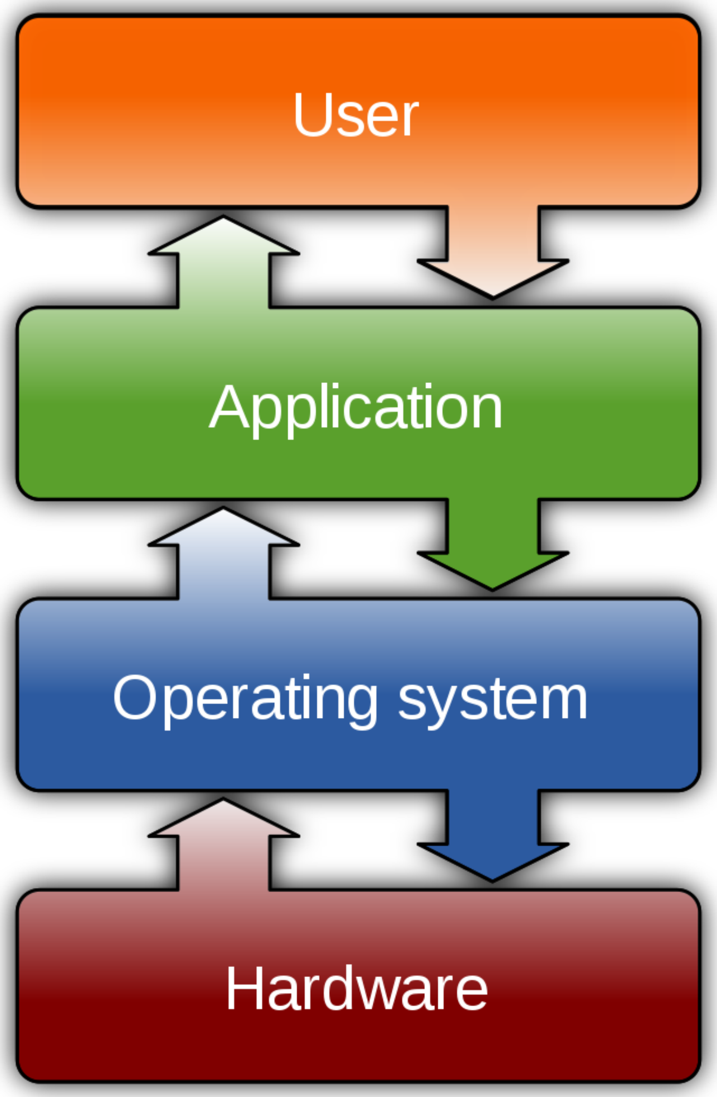
File system

- File system is a scheme for controlling how data is stored and retrieved from a partition.
  - File system organizes the sectors into blocks.
  - Organization:
    - Data is grouped into files.
      - Files have meta-data: e.g. when created, access permissions, …
    - Files are organized into directories.
  - FS holds a map Files names → Blocks.
    - Different file systems have different restrictions file names (e.g. allowed characters or case sensitivity).
    - General convention, “FOO.BAR”, FOO is filename, BAR is the extension which helps indicate the format of the file contents.
  - FS enforce Access Control or Permissions… who can read what file.
  - Different File system types:
    - Windows: FAT (FAT16, FAT32), exFAT NTFS, …
    - Mac: HFS, HFS+, APFS
    - Unix: ext2, ext3, ext4, ZFS, …
  - SWAP: In some Operating Systems, the system can use storage as RAM when out of memory.



Software- Firmware

- There are several layers of software that interact with the computer at different levels.
  - During system boot, system starts with the Firmware, and hands control to next level, which then hands control on to then next level, and so on …
- ***Firmware***: Associated with the motherboard. Read Only Memory (ROM).
  - When you start a computer, it needs some instructions on what to do.
  - Test that everything is OK… e.g. memory.
  - Load user configuration.
  - Allow users to change configuration via menu system.
  - Start and configure peripherals, storage, …
  - Load the boot loader from storage into memory and hand-off
  - In the x86 world:
    - Basic Input/Output System (BIOS): Old
    - Unified Extensible Firmware Interface (UEFI): New



Boot Loader …

- Small program that loads the operating system kernel.
- Loaded by the firmware.
- Usually sits on the first sectors of the storage boot drive.
- May give the user the option of loading different operating systems sitting on different partitions of the storage.
- Must mount the partition, find the kernel, load it into memory, pass control to the kernel with potentially some configuration options (e.g. the root partition).

Operating System

- Software system that manages the computer hardware and software and provides common services for programs.
    - Sharing of resource between programs: processor, memory, storage, …
    - Intermediary between programs and hardware.
    - Provides Application Program Interface (API) / Software Development Kit (SDK) for building programs and interfacing them with OS.
- Examples: Windows, MacOS, Linux, iOS, Android, …
- Modern OSs are Multi-tasking: allow multiple programs to simultaneously run.
    - Each program ~ a process.
    - Pre-emptive multitasking: the OS gives slice of CPU time to each process.

Unix

- Multitasking, Multi-user, OS originally developed in 1970 by AT&T Bell Labs to run on mainframes with many connected terminals.
    - Written in C programming language.
- Many modern operating systems, including MacOS and Linux, implement Unix standards.
- “Unix Philosophy”
    - Plain text data storage.
    - Hierarchical file system.
    - Devices and inter-process communication via files
    - Main program that runs is the kernel.
    - Primary user interface is a *command-line* interpreter, called a shell.
    - Modular:
        - lots of small programs serve as tools
        - strung together via the *command-line* interpreter
        - passing information between each other via pipes

Graphical User Interface

- Though most of you associate an OS with its GUI, it is generally a layer on top of the core OS.
  - Establishes graphical metaphors aimed at simplifying user interaction with the OS.
    - Mouse cursor, windows, menus, buttons, …
  - Provides API for applications to build GUIs for themselves.
- For Unix, the X Windows system implements a client-server model:
  - The server runs the application that makes the API GUI calls.
  - These calls are transmitted to the network to a client.
  - The client runs the graphical environment, collects input, and draws the graphical elements.