Operating System

A control program(s) managing hardware resources

A *modern* OS contains:

- **Kernel**
- **Dynamically shared libraries** – when an application needs to execute a subroutine, it calls it from a pre-compiled set, instead of adding that subroutine to its own code, thus resulting in smaller code
- **Kernel modules** – allow the current size of the kernel program to change dynamically through execution. As a program needs another element of the kernel, it is loaded into memory
- **Device driver libraries** – subroutines to control a wide variety of hardware devices
- **Graphical User Interface**

The OS is a mediator – it grants access. It can stop access, prevent access, and cause things to happen. It occasionally fails.

Four Management Roles of an OS

1. Memory management
2. Process management – time and machine state
    a. Which process is currently running on the CPU?
    b. Which processes are “alive” - are in memory?
    c. Which process should get CPU time next?
3. Device management
4. File system management

Process

An application / program becomes a process when is resident in memory and is executing.

Example Operating Systems

Personal computers – Windows, Linux, MacOS, DOS, FreeBSD Unix, OS/2
Workstation servers – Linux, Windows Server
Embedded computers