What is an Operating System?

- An *operating system* (OS) is the interface between the user and the hardware
  - It implements a virtual machine that is easier to program than bare hardware
- An OS provides standard **services** (an interface) which are implemented on the hardware, including:
  - Processes, CPU scheduling, memory management, file system, networking
- The OS **coordinates** multiple applications and users (multiple processes) in a fair and efficient manner

↪The goal in OS development is to make the machine **convenient** to use (a software engineering problem) and **efficient** (a system and engineering problem)

Why Study Operating Systems?

- Abstraction — how do you give the users the illusion of infinite resources (CPU time, memory, file space)?
- System design —tradeoffs between:
  - performance and convenience of these abstractions
  - performance and simplicity of OS
  - functionality in hardware or software
- Primary intersection point — OS is the point where hardware, software, programming languages, data structures, and algorithms all come together
- Curiosity — “look under the hood”
- “Operating systems are among the most complex pieces of software yet developed”, William Stallings, 1994

Modern OS Functionality

- Concurrency
  - Multiple processes active at once
  - Processes can communicate
  - Processes may require mutually-exclusive access to some resource
  - CPU scheduling, resource management
- Memory management — allocate memory to processes, move processes between disk and memory
- File system — allocate space for storage of programs and data on disk
- Networks and distributed computing — allow computers to work together
- Security & protection

What is an Operating System?

- A **magician** — provides each user with the illusion of a dedicated machine with infinite memory and CPU time
- A **government** — allocates resources efficiently and fairly, protects users from each other, provides safe and secure communication
- A **parent** — always there when you need it, never breaks, always succeeds
- A **fast food restaurant** — provides a service everyone needs, always works the same everywhere (standardization)
- A **complex system** — but keep it as simple as possible so that it will work