What is an Operating System? (Review)

- 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)

History of Operating Systems

- Phase 0 — hardware is a very expensive experiment; no operating systems exist
  1. One user at console
     - One function at a time (computation, I/O, user think/response)
     - Program loaded via card deck
       - Libraries of device drivers (for I/O)
     - User debugs at console
- Phase 1 — hardware is expensive, humans are cheap
  2. Simple batch processing: load program, run, print results, dump, repeat
     - User gives program (cards or tape) to the operator, who schedules the jobs
     - *Resident monitor* automatically loads, runs, dumps user jobs
     - Requires memory management (relocation) and protection
     - More efficient use of hardware, but debugging is more difficult (from dumps)

History of Operating Systems (cont.)

- Phase 1 — hardware is expensive, humans are cheap
  3. Overlapped CPU & I/O operations
     - First: buffer slow I/O onto fast tape drives connected to CPU, replicate I/O devices
     - Later: *spool* data to disk
  4. Multiprogrammed batch systems
     - Multiple jobs are on the disk,waiting to run
     - *Multiprogramming* — run **<u>several</u>** programs at the “same” time
       - Pick some jobs to run (*scheduling*), and put them in memory (*memory management*)
       - Run one job; when it waits on something (tape to be mounted, key to be pressed), switch to another job in memory
     - First big failures:
       - MULTICS announced in 1963, not released until 1969
       - IBM’s OS/360 released with 1000 known bugs
     - OS design should be a science, not an art

History of Operating Systems (cont.)

- Phase 2 — hardware is less expensive than before, humans are expensive
  5. Interactive *timesharing*
     - Lots of cheap terminals, one computer
       - All users interact with system at once
       - Debugging is much easier
     - Disks are cheap, so put programs and data online
       - 1 punch card = 100 bytes
       - 1MB = 10K cards
       - OS/360 was several feet of cards
     - New problems:
       - Need *preemptive scheduling* to maintain adequate *response time*
       - Need to avoid *thrashing* (swapping programs in and out of memory too often)
       - Need to provide adequate security measures
     - Success: UNIX developed at Bell Labs so a couple of computer nerds (Thompson, Ritchie) could play Star Trek on an unused PDP-7 minicomputer

History of Operating Systems (cont.)

- Phase 3 — hardware is very cheap, humans are expensive
  6. Personal computing
     - CPUs are cheap enough to put one in each terminal, yet powerful enough to be useful
       - Computers for the masses!
     - Return to simplicity; make OS simpler by getting rid of support for multiprogramming, concurrency, and protection

---

- Modern operating systems are:
  - Enormous
    - Small OS = 100K lines of code
    - Big OS = 10M lines
  - Complex (100-1000 person year of work)
  - Poorly understood (outlives its creators, too large for one person to comprehend)

History Lessons

- None of these operating systems were particularly bad; each depended on tradeoffs made at that point in time
  - Technology changes drive OS changes
- Since 1953, there has been about 9 orders of magnitude of change in almost every computer system component
  - Unprecedented! In past 200 years, gone from horseback (10 mph) to Concorde (1000 mph), only 2 orders of magnitude
- Changes in “typical” academic computer:

	1981	1996
MIPS	1	400
price / MIPS	$100,000	$50
memory	128 KByte	64 MByte
disk	10 MByte	4 GByte
network	9600 bit/sec	155 Mb/s
address bits	16	64

Modern OS Functionality (Review)

- 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

More Recent Developments

- Parallel operating systems
  - Shared memory, shared clock
  - Large number of tightly-coupled processors
  - Appearance of single operating system
- Distributed operating systems
  - No shared memory, no shared clock
  - Small number of loosely-coupled processors
  - Appearance of single operating system is ideal goal, but not realized in practice
  - May try to simulate a shared memory
- Real-time operating systems
  - Meet hard / soft real-time constraints on processing of data