Operating Systems cs4/53201

Instructor	Dr. Mikhail Nesterenko office: MCB 356 email: mikhail@cs.kent.edu phone: 672-9109
office hours:	TU 5:30-6:45pm or by appointment
webpage:	http://www.cs.kent.edu/~mikhail/classes/os/
textbook:	Operating System Concepts by A. Silberschatz and P. B. Galvin

Lecture 1 Introduction, OS History

- notion of an operating system
- OS history
  - no OS
  - batch system
  - multiprogramming
  - multitasking
- features of modern Oses
- new OS research and development

What is operating system?

Operating system is a program that acts as an intermediary between a user of computer and computer hardware

- Hardware - central processing unit (CPU), memory, input/output (I/O) devices - provides basic computer resources
- Software
  - OS
  - applications programs - compilers, database systems, games, business programs

OS makes the hardware more convenient to program and use and ensures that it is used efficiently. It does so by:

- providing standard services to the application programs (file access, memory management, networking, etc.)
- coordination of multiple applications and users

An OS is like:

- 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

OS history

- The history of OS follows the decrease of hardware cost and increase in speed and sophistication:
- From 1953 the speed of computer systems increased 9 orders of magnitude (cf. the speed of means of public transportation throughout modern history increased only 2 orders of magnitude: horseback - 10 mph, supersonic passenger jet - 1000 mph)
- 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

OS history (cont.)

Phase 0 - no OS

One user at console

- One function at a time (computation, I/O, user think/response)
- Program loaded via card deck
- User debugs at console

Phase 1 - Batch Systems

- No direct interaction between user (programmer) and machine: user submits her program (job) to another person - *operator*, the jobs are *batched* to speed up processing and run by the operator, the result is returned to the user
- the processor (CPU) is idle most of the time since the peripheral devices are slow
- maybe we can *spool* the jobs on disk to speed up the loading process?

Successes/Failures

- MULTICS announced in 1963, not released until 1969
- IBM's OS/360 released with 1000 known bugs
- UNIX developed at Bell Labs so a couple of computer nerds (Thompson, Ritchie) could play Star Trek on an unused PDP-7 minicomputer

OS history (cont.)

Phase 2 - Multiprogramming

- *Multiprogramming* - several programs are executed in parallel
  - 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, card read, key pressed), switch to another job in memory

- Phase 3 Time sharing (Multitasking)
  - multiprogramming is done so fast that the user(s) can access the system simultaneously
  - 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 protection

Features of Modern OSes

- 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 communicate and work together
- Security & protection

Problems with modern OSes

- 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)
- have to support a variety of software created at different times and with different demands (legacy code/modern software)

New Research and Development

- 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