CSC139 Operating System Principles

Fall 2020, Part 1-1

Instructor: Dr. Yuan Cheng

Session Plan

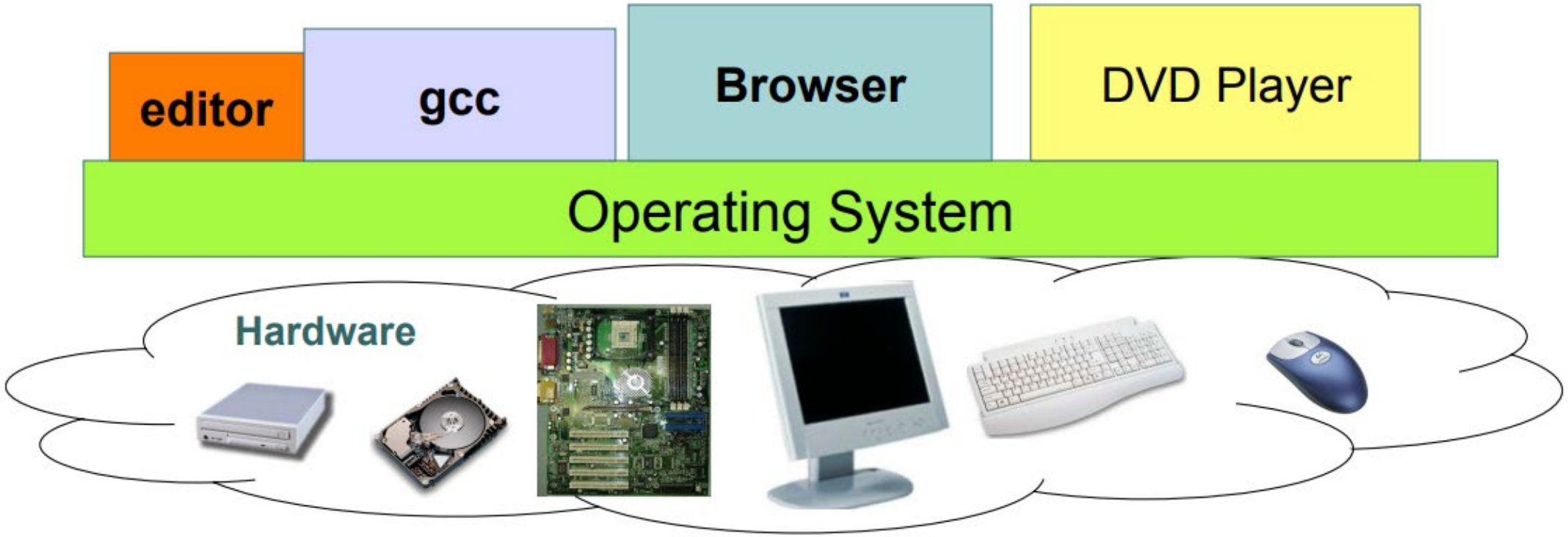
- Introduction to Operating Systems
  - What is an OS? What does an OS do?
  - Evolution of OS
  - Why study OS?
  - OS functionalities
  - OS components

On August 25, 1991

- 21-yo Linus Torvalds announced that he was working on a new OS, which was ‘just a hobby, won’t be big and professional’.

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki


Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones.  This has been brewing
since april, and is starting to get ready.  I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want.  Any suggestions
are welcome, but I won't promise I'll implement them :-)

               Linus (torvalds@kruuna.helsinki.fi)

PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

What is an Operating System?



What is an Operating System?

- A program that acts as an *intermediary* between the user and the hardware
  - Provides a virtual execution environment on top of hardware that is more convenient than the raw hardware interface

Consider reading from disks

- Different types of disks, with very different structures
  - Floppy, various kinds of hard drives, Flash, IDE, …
- Different hardware mechanisms to read, different layouts of data on disk, different mechanics

- Floppy disk has ~20 commands to interact with it
- Read/write have 13 parameters; controller returns 23 codes
- Motor may be on or off, don’t read when motor off, etc.
- And this is only one disk type

- Rather, a simple abstraction: data are in files, you read from and write to files using simple interfaces
- OS manages all the rest

What Do Operating Systems Do?

- Provides abstractions to user-level software above
  - User programs can deal with simpler, high-level concepts
  - Hide complex and unreliable hardware, and the variety of hardware
  - Provide illusions like “sole application running” or “infinite memory”

- Implement the abstractions: manage resources
  - Manage application interaction with hardware resources
  - Allow multiple applications and multiple users to share resources effectively without hurting one another
  - Protect application software from crashing a system

Operating System Definition

- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer

- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Operating System Definition (cont.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is a good approximation
  - But varies wildly
- “The one program running at all times on the computer” is the kernel.
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.

Question: Operating System Components

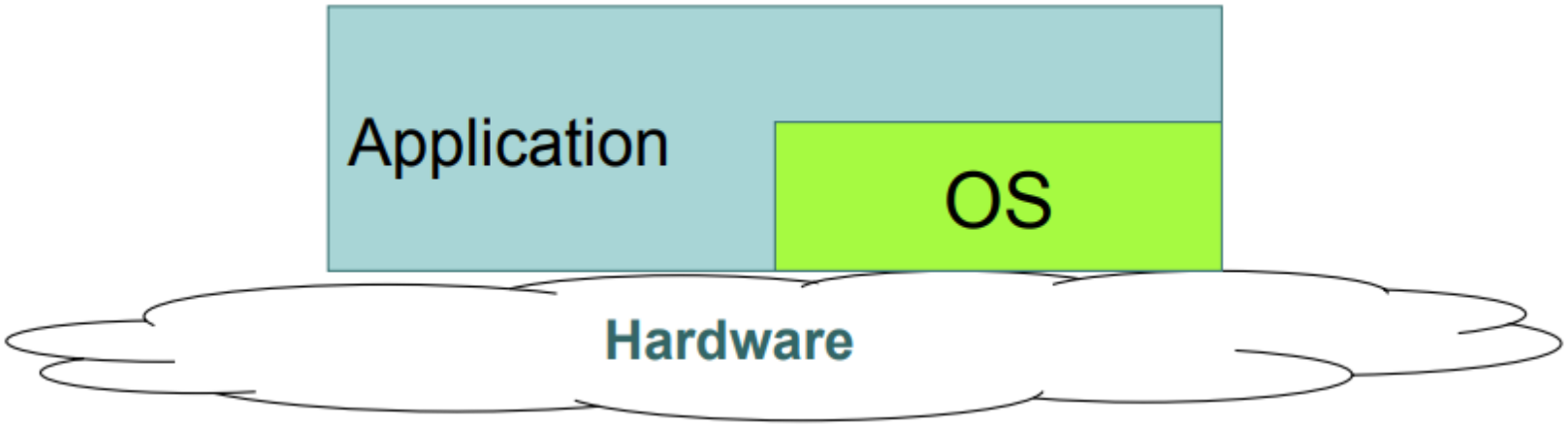
- Which of the following are likely components of an operating system? Check all that apply.
  - A. File editor
  - B. File system
  - C. Device driver
  - D. Cache memory
  - E. Web browser
  - F. Scheduler

Evolution of Operating Systems

- Mainframe Systems
  - Batch Systems (mid 1950s – mid 1960s)
  - Multi-programmed Systems (1960s)
  - Time-sharing Systems (1970s)
- Personal Computer Operating Systems (1980s)
- Modern Variants
  - Parallel Systems
  - Distributed Systems
  - Real-time and Embedded Systems
  - Ubiquitous Systems

Phase 1: Hardware Expensive, Human Cheap

- User at console, OS as subroutine library
- Batch monitor (no protection): load, run, print
- A lot of the (expensive) hardware sits idle a lot. Developments
  - Interrupts; Overlap I/O and CPU
  - Direct Memory Access (DMA)
  - Memory protection: keep bugs to individual programs
  - Multics: designed in 1963 and run in 1969; multiprogramming
- Assumption:
  - No bad people
  - No bad programs.
  - Minimum interactions

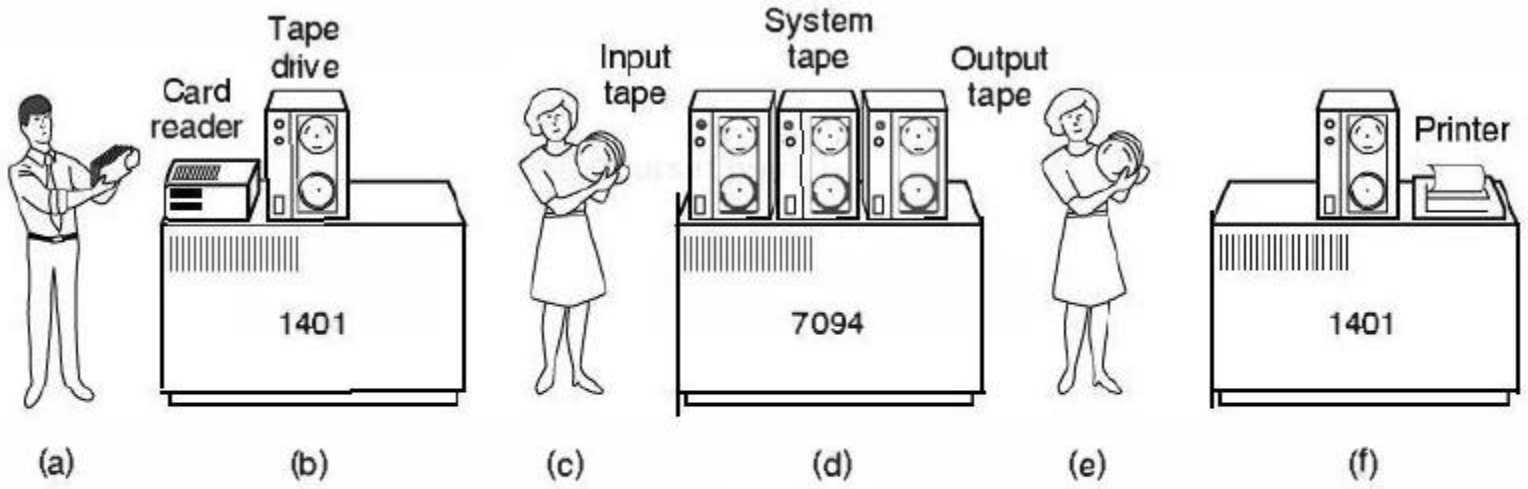


Mainframe Systems

- First computers used to tackle many commercial and scientific applications
- Evolved from simple batch systems, multi-programmed systems, to time-sharing system

Batch Systems

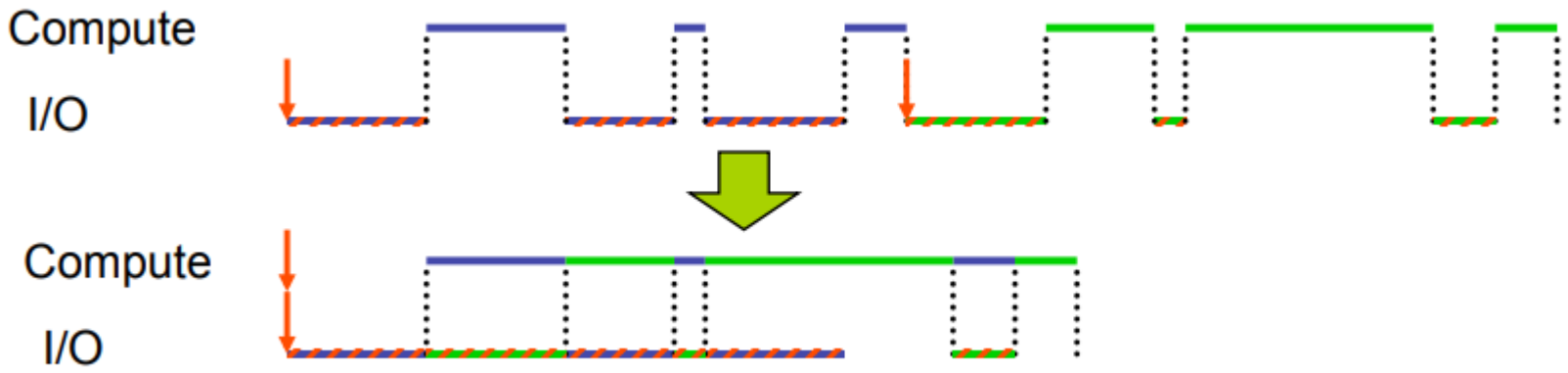
- To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group.

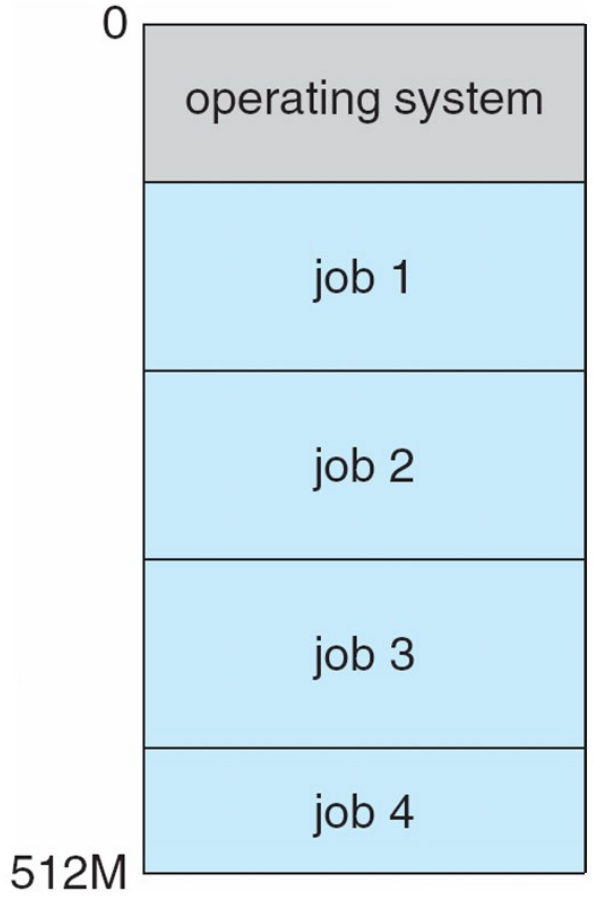


An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

Multi-programming Systems

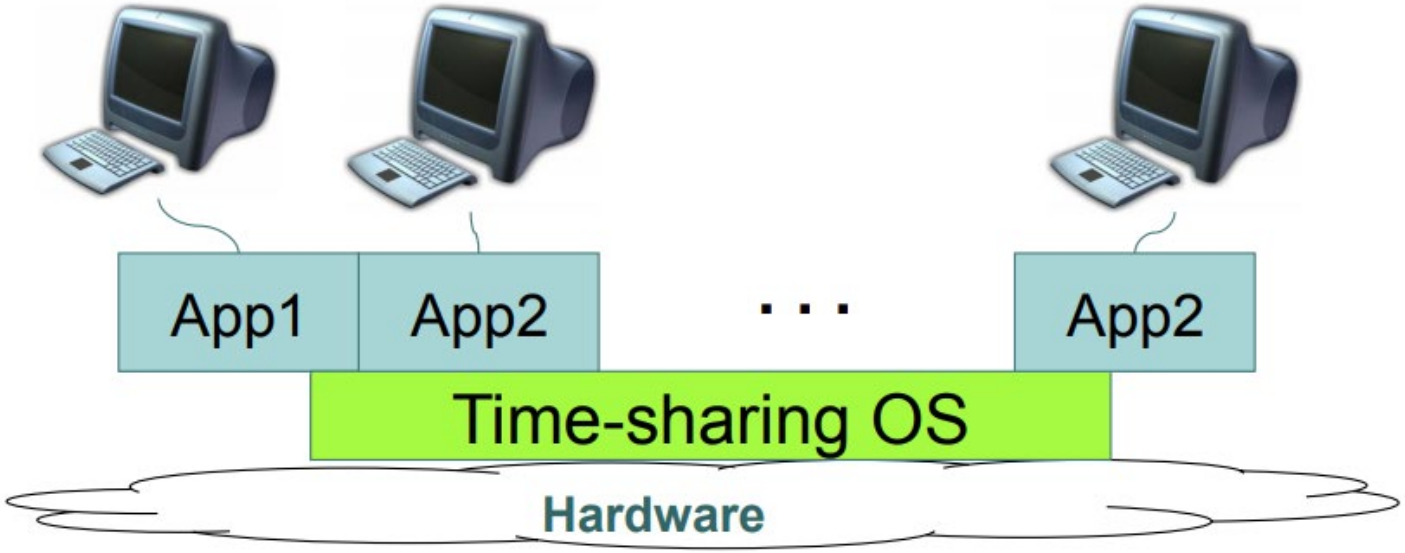
- The OS keeps several jobs in memory simultaneously.
- The CPU is switched to another job when I/O takes place





Phase 2: Hardware Cheap, Human Expensive

- Use cheap terminals to share a computer
- Time-sharing OS
- Unix enters the mainstream as hardware got cheaper
- Problems: thrashing as the number of users increases



Time-sharing Systems

- Extension of multiprogramming systems to allow on-line interaction with users
- User feels as if she has the entire machine
- Based on time-slicing: divides CPU time equally among active users
- Optimizes for *response time* at the cost of *throughput*

Phase 3: HW Cheaper, Human More Expensive

- Personal Computer
  - Altos OS, Ethernet, Bitmap display, laser printer
  - Pop-menu window interface, email, spreadsheet, FTP, Telnet
- PC Operating System
  - Memory protection
  - Multiprogramming
  - Networking

PC Operating Systems

- Earliest ones in the 1980s
- computer system originally dedicated to a single user
- Objective: User convenience and responsiveness
  - Individuals have sole use of computers
  - A single user may not need advanced features of mainframe OS (maximize utilization, protection)

Parallel Systems

- Parallel systems growing in use and importance
  - Also known as multiprocessor systems, tightly-coupled systems
  - Processors share memory and a clock; communication usually takes place through the shared memory
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types:
    1. Asymmetric Multiprocessing – each processor is assigned a specific task.
    2. Symmetric Multiprocessing – each processor performs all tasks

Parallel Systems (cont.)

- Symmetric multiprocessing (SMP)
  - All processors are peers
  - Kernel routines can execute on different CPUs, in parallel
- Asymmetric multiprocessing (AMP)
  - Master/slave structure
  - The kernel runs on a particular processor
  - Other CPUs can execute user programs and OS utilities

The “Master/Slave” Controversy

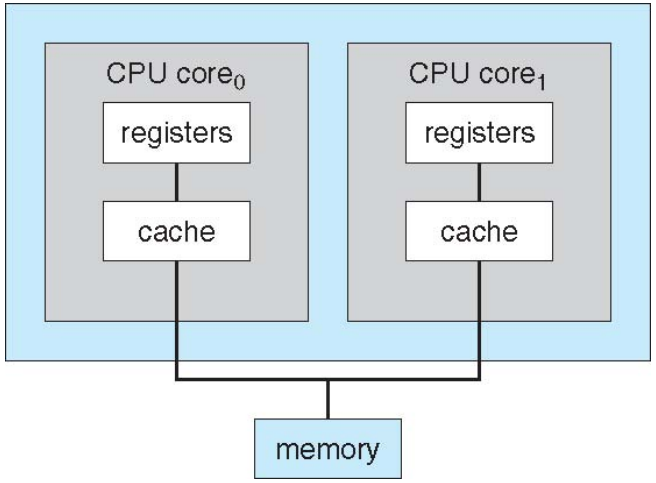
- The term is written into the code of many software products.
- Concerns about the use of master/slave terminology has been simmering for years.
  - Los Angeles County, 2003
- Python moved to eliminate the language in 2018
  - Some developers supported continuing to use it
  - Others wanted to jettison “slave” but debated the acceptability of “master.”

Polls

- Q: Do you think terms like master/slave are appropriate?
- Q: Do you think terms like blacklists/whitelists are appropriate?
- Q: What alternative names will you propose?

Parallel Systems (cont.)

- Multi-core architectures
  - Include multiple computing cores on a single chip
  - Need to exploit parallelism at run-time



Distributed Systems

- Distribute the computation among several physical processors
- Loosely coupled clustered system – each processor has its own local memory; processors communicate with one another through various communications lines
- Advantages of distributed systems
  - Resource and Load Sharing
  - Scalability

Real-time and Embedded Systems

- A real-time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data.
- An embedded system is a component of a more complex system – Control of a nuclear plant
  - Missile guidance
  - Control of home and car appliances
- Real-time systems
  - Real-time means “predictable” not “fast” – have well-defined time constraints
  - May be either hard or soft real-time
    - Hard real-time: OS guarantees that applications will meet their deadlines
    - Soft real-time: OS provides prioritization, on a best-effort basis

Ubiquitous Systems

- PDAs, personal computers, cellular phones, sensors
- Challenges:
  - Small memory size
  - Slow processor
  - Different display and I/O
  - Battery concerns
  - Scale
  - Security
  - Naming

Exit Slips

- Take 1-2 minutes to reflect on this lecture
- On a sheet of paper write:
  - One thing you learned in this lecture
  - One thing you didn’t understand

Next session

- We will continue to discuss:
  - Introduction to Operating Systems
- Reading assignment: (skim through before class and continue reading over the weekend)
  - SGG: Ch. 1, Ch. 2
    - You may skip 1.11, 2.7, 2.9, and 2.10

Acknowledgment

- The slides are partially based on the ones from
  - The book site of *Operating System Concepts (Tenth Edition)*: http://os-book.com/