CSE 30341: Operating Systems

- Instructor: Surendar Chandra (surendar@nd.edu)
  Room: 356C Fitz (631-8975)
  Office Hours: Wed 12:00-1:00, Fri 1:00-2:00
  (other times, by email appt)
  Email/iChat/AIM is the best way to reach me
- TA: Chris Boehnen
- Course Web: cse.nd.edu/courses/cse30341/www
- Mailing list: cse30341-01-sp06@listserv.nd.edu

Outline for today

- High level introduction to Operating Systems
- Course policies:
  - Course goals, organization and expectation
  - Grading policy, late policy, reevaluation policy
  - Academic honesty

Computer System Structure

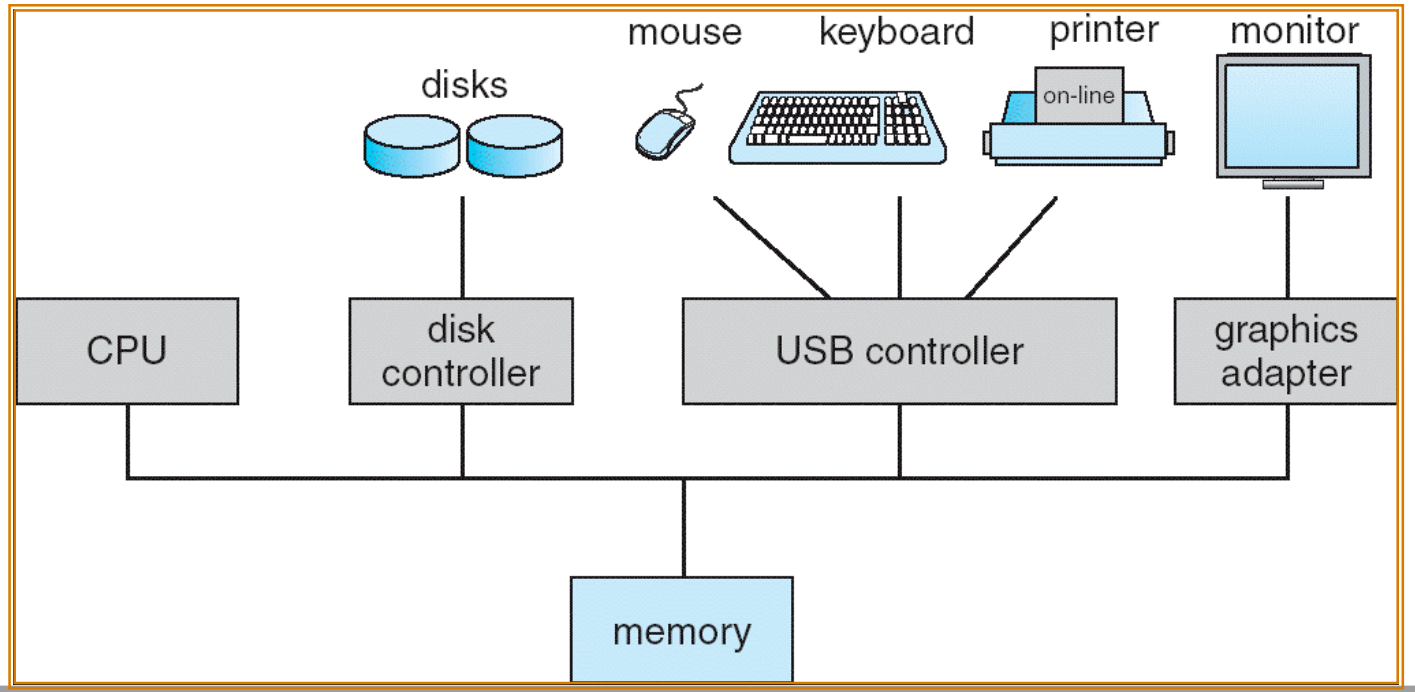
- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
  - Challenge is to manage resources for competing uses: simultaneously playing interactive games and sending a print out to an laser printer
- 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

Computer System Organization

- Computer-system operation: concurrency for utilization and performance
  - One or more CPUs and device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices while competing for memory cycles



Computer-System I/O Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer to store pending data
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an *interrupt*

Operating System Structure

- Multiprogramming needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via job scheduling
  - When it has to wait (e.g, for I/O), OS switches to another job

Operating System Structure

- Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing
  - Response time should be small (< 1 second)
  - Each user has programs executing in memory ⇨process abstraction
  - If several jobs ready to run at the same time ⇨ process management
  - If processes don't fit in memory, swapping moves them in and out to run ⇨ memory and storage management
    - Virtual memory allows execution of processes not completely in memory

Protection structure

- OS should protect the users/processes from each other as well as protect itself from users
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user

Popular operating Systems

- Desktop and server OS: Microsoft Windows, UNIX and variants (Linux, Solaris, FreeBSD, Mac OSX)
  - Main focus of this course. We will use Linux for source code examples
  - Resource utilization is important (even though servers care about different things than desktops, e.g. throughput vs interactive computing)
- Embedded and realtime systems: runs your cars, TV, washing machines, nuclear plants etc. (e.g. QNX, Vxworks)
  - Many real time applications require timing guarantees
  - There are more embedded systems than desktops - not the focus of this course

Course Goals

- Cover core technologies in depth & introduce current operating system technologies
- Goal is to cover as much breadth rather than depth
- As much hands on experience as possible
- Home work projects should help

Course Organization

We will follow the course text for the most part

I will augment the discussions with topical research topics

I encourage open discussion about the technologies

Grade distribution

- Modules:
  - Process Management
  - Process Synchronization
  - Memory management
  - Storage management (CSE 70481)
  - Protection, Security (CSE 40567), Distributed (CSE 40771) and Real time (CSE 40463)
- Exams - 40%
  - 5 module exams (5 * 6%)
  - Final (10%)
- Take home assignments – 30%
  - 5 modules (5 * 6%)
- Programming projects - 30%
  - 5 projects (5 * 6%)

Module assignments, exams and Final

- Exams:
  - Open book, open notes, in class exams
    - Module exams: 30 minutes
- Module take home assignments
  - assigned at the beginning of each module
  - designed to help you prepare for the final exams and module exams
  - You may have to perform experiments to answer some of the questions

Homework projects

- **Projects are group (ideally two) efforts**.
- Each project should be electronically turned in with a succinct report on your implementation strategy and what you learned.
- Projects should compile without any modifications. C is the preferred language. Use the Linux cluster in Cushing 208 for the projects. If you need a specific OS, you should make arrangements beforehand
- I may randomly select submissions for an one-on-one oral interview

Reevaluation policy

- Arithmetic errors, missed grading will be reevaluated promptly
- I encourage you to discuss concerns with your solution with me
- I discourage re-evaluation of partial credits (partial credits are based on the complexity of your solution and the overall class performance):
  - Football penalty policy:

    If you think you deserve a better partial grade, write down the reason why you think that you deserve a better grade and how many extra points you think you deserve. If I agree, you could get up to this many extra points. If I disagree, you will lose this much points. You can increase your odds by performing experiments to prove your answer

Late policy

- None – Projects/homework/critiques are due at 10:40 am (right before the beginning of class). **I do not accept late submissions** (not even a second)

- Please contact me regarding unforeseen emergencies

Academic Honesty

- Freedom of information rule:
  - Collaboration is acceptable (even for individual efforts such as take home assignments as long as you follow the rules of this course)
  - To assure that all collaboration is on the level, **you must always write the name(s) of your collaborators on your assignment.** Failure to adequately acknowledge your contributors is at best a lapse of professional etiquette, and at worst it is plagiarism. Plagiarism is a form of cheating.

Academic Honesty – Gilligans Island Rule

- This rule says that you are free to meet with fellow students(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you may **not take any written (electronic or otherwise) record away from the meeting**. This applies when the assignment is supposed to be an individual effort. After the meeting, engage in half hour of mind-numbing activity (like watching an episode of Gilligan's Island), before starting to work on the assignment. This will assure that **you are able to reconstruct what you learned from the meeting, by yourself, using your own brain**.