Syllabus
Assignments -- Uses Blitz (facultyweb.cs.wwu.edu/~phil/classes/blitz)

Environment
  UNIX (Linux, OS X, NetBSD, FreeBSD ....)
  Should be the same since Blitz is a Virtual Machine
  Blitz uses a language called "kpl"

Reading the book ... you should do it!

What is an operating system?
  □ Hardware manager
  □ Allows "user" (application) programs to utilize the hardware
  □ Two Views:
    □ User view: "Abstract machine"
    □ System View: glue between Abstract machine and real machine
Introduction (Chapt 1 continued)

Computer systems architecture ...
- CPUs, single, multiple (SMP, parallel, distributed, clusters)
- Memory Hierarchy: registers, cache, main memory, SSD, HardDisk, ...
- I/O Devices: Disks, Tape, USB, video, ...
- Other: interrupts, timers, ...

Primary Hardware Mechanism
- Dual Mode: supervisor (system, privileged) vs user
- CPU hardware operation state
- User mode provides restriction of use of hardware
- Methods to switch between the two
Other issues and abstractions

Processes
- Abstraction for code execution
- Uses memory and Dual Mode mechanisms

Other issues
- Memory Management
- Storage Management
- Caching
- Protection and Security
- Networking and Distributed Systems
- Special Purpose systems: real time, multimedia, handheld

Open Source Operating Systems
- Linux -- Senior CS Major started it ...
- BSD -- Berkeley Computer Science Research Group
- Open Solaris -- recent open source OS, but now discontinued
- openindiana, freertos, .... more open source OSes.
Operating System Structures (Ch 2)

Services

- Program Execution
- I/O operations
- File systems -- data storage
- Communications -- (process to process, network ...)
- Error Detection
- Resource Allocation (memory, disk, cpu time, ...)
- Accounting
- Protection and Security
- User Interface?
  - Some by the OS: e.g. Windows
  - Some not by the OS: UNIX
  - GUI vs Command Line
Complete OS distributions have more than OS code

- Kernel: the actual OS itself
- Utilities: User land code to make things work
  - UNIX kernel alone is rather useless!
- Linux distribution
  - Linux Kernel (essentially same for all distros)
  - GNU utilities
  - Other programs
  - Each distro has a different set/ordering
- BSD distributions ....
  - Kernel -- unique to the xxxBSD
  - Core Utilities
  - 3rd party Utilities (e.g. NetBSD pkgsrc, FreeBSD Ports)
System Calls

The method a User land program requests services from the kernel
- Syscall API (Application Programming Interface)
- Transition from "User mode" to "kernel mode"
- Controlled entry into the kernel

Types of system calls (see p67 for examples)
- Process Control
- File Management
- Device Management
- Information maintenance (time of day, getpid(),...)
- Communications ...

Library Routines
- Often supplied as part of the Distro/OS
- Integrated into API
Design Goals
- kind of OS -- batch, real time, time sharing, mobile, embedded, parallel ...
- mechanisms and policies --
  - separation
  - policy regardless of mechanism
  - mechanism how to implement policy
- implementation
  - Choice of programming language ...

Operating System General Structures
- Simple Structure (aka the big mess)
- Layered approach (software belongs to a specific layer)
- Microkernel and "servers", message passing
- Modules (aka object oriented)
Virtual Machines

- Experiment with OSes (eg Blitz)
- VM can provided idealized machines
- Abstract VMs (java VM, ...)
- Helps with OS debugging
  - Slow down the machine
  - Simulation can make things repeatable

Real OS debugging?
- Windows -- second machine as debugger
- Kernel debuggers
- kernel dumps (blue screen or panic)
System boot ... How?
- Turn on, enter program via buttons, push "run button"
- Basic Binary Loader, read "paper tape" or "mag tape"
Todays machines -- Boot process

- Power up starts running at location 0.
- Hardware typically forces a jump to a ROM image
- ROM has "machine monitor" program
- May look for OS/Boot code on a Disk, net, ...

Blitz?

- "Power On Reset" exception
- Loads PC with the value 0
- Sets status to System mode, no memory translation, no interrupts.
- "memory" can be loaded before the "Power On Reset"
- Execution then executes the first instruction in memory ...
  - This is normally a "jump label" instruction