What is this course about?

• Students will learn
  – How operating systems are designed and organized
  – What their purposes are and how they achieve them
  – Some examples of how they are implemented

• Students will be able to
  – Understand and solve performance problems on operating systems
  – Argue on possible improvements on operating systems
Course Info

Instructor
Georgios Varsamopoulos
BY 514
M, Tu, W, Th
3:30pm – 4:30pm

Teaching Assistant
Priyanka Bagade
BYENG 517
TBA

- Use of webpage
  http://impact.asu.edu/cse430sp11.html for material,
- Use of Blackboard as a dropbox (SafeAssign)
- Take Notes! Most testing methods are open-note.
About the instructor

- **Name:** Georgios Varsamopoulos
- **Title:** Research Assistant Professor
- **Joined ASU:** 2007 as a post-doctorate researcher
- **Research Lab**
  - Impact Lab: [http://impact.asu.edu/](http://impact.asu.edu/)
- **Research Projects** *(REU positions available)*
  - Thermal-aware and sustainable management of computing systems
  - BlueTool: Research Infrastructure for sustainable data centers
- **Interests**
  - resource allocation and management, computer networks, sustainable computing, performance optimization, cyber-physical models
Concerns

- Your Concerns
  - Amount of work
  - Difficulty of programming
  - Required background
  - Usefulness of the course

- My concerns
  - Here-for-grade or here-for-degree mentality
  - Starting homework a couple of days before it is due
  - Not using knowledge from previous experience
  - Not participating and not challenging the information given
  - Plagiarism and other forms of cheating
  - Seeing students give up and fail
Course Workload

- Quizzes roughly on a biweekly basis
- About 6 homework assignments
- Two midterms
  - Early March
  - Early April
- A project
  - Groups preferred
  - Start mid March
  - May replace final exam

Grading distribution:
- 20% homework assignments
- 10% quizzes
- 25% project/exam
- 20% each midterm
- 5% active attendance (in-class participation)

Grading Rubric:
- Ability to solve problem
- Understanding of background
- Creativity
- Clarity
My biggest concern

• Technical writing skill
  – In four words: brief, exact, complete, clear

• Research resources
  – The quality and authenticity of your sources is a defining factor of your work's quality
    • The web and wikipedia is ok as a start.
    • Did you know of Google Scholar (scholar.google.com)?
    • Have you ever used the ACM and IEEE online libraries?

• Please pay particular attention to your technical writing and your written expression skills.
What is an operating system?
What is it for? What does it do?
What is an operating system?

- It is the centerpiece software running on a computer that:
  - Manages the computer hardware
  - Manages the other software installed
  - Manages running programs
  - Abstracts the computer hardware to the installed software and provides the software with an interface to the hardware
  - Manages and organizes the use of the computer by programs
  - Provides a means to the user to manage the computer and the software (stored or running)
  - Provides a means to the programs to communicate with each other
  - Enforces access, usage and security polices among users and programs
Roles of operating system

• For the user
  - Allows to start, pause, and terminate programs
  - Allows to manage files

• For the programs (and programmers)
  - Provides an abstracted machine
    • Easier to build programs
    • Programs work on any computer of the same ISA+OS combination
      - OS's API is to the computer what ISA is to the processor
  - Allocates resources (explicitly or transparently)

• For the system
  - Controls and manages operation and coordination among hardware components
Basic terminology

- **Kernel**
  - The core, ever-running program of an operating system

- **Process**
  - A running instance of a program

- **System calls**
  - The API to the operating system

- **System program**
  - A program that is shipped with the OS and is used to perform system-management tasks

- **Application program**
  - End-user program

- **Device driver**
  - A small routine that translates the device's signals and data to what the OS is programmed to understand
The OS as an abstract machine

user 1

user 2

user 3

... user n

Compiler

Assembler

Text editor

... database system

System and application programs

Operating system

Computer hardware
The layered architecture of OS

- Kernel-space processes
- Kernel
- Drivers
- HAL
- H/W
- User-space processes
- system calls
- system calls
- User-space processes
A primer on Computer Architecture and Organization
Computer Organization

Diagram showing the organization of computer components, including:
- CPU
- Disk controller
- USB controller
- Memory
- Mouse
- Keyboard
- Printer
- Graphics adapter
- Monitor
Interrupts and traps

- A signal is raised by a h/w component
  - An interrupt register is set to the corresponding interrupt value
- The CPU stops what it was doing, *saves the context*, switches to *kernel mode* and calls the Interrupt Handler (a kernel subroutine) based on the Interrupt Descriptor/Vector Table (IDT or IVT)
- After the handling of the interrupt is done, the CPU resumes the paused process
- Trap: a “software interrupt”
  - Interrupt register is set by software and not by hardware
Memory Hierarchy

- **Volatile memory**
  - registers
  - cache
  - main memory
  - electronic disk
  - magnetic disk
  - optical disk
  - magnetic tapes

- **Non-volatile memory**

The hierarchy shows a trade-off between speed and size, with faster components having smaller capacity and larger components being slower but capable of storing more data.
# Memory Performance

<table>
<thead>
<tr>
<th>Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>registers</td>
<td>cache</td>
<td>main memory</td>
<td>disk storage</td>
</tr>
<tr>
<td>Typical size</td>
<td>&lt; 1 KB</td>
<td>&gt; 16 MB</td>
<td>&gt; 16 GB</td>
<td>&gt; 100 GB</td>
</tr>
<tr>
<td>Implementation technology</td>
<td>custom memory with multiple ports, CMOS</td>
<td>on-chip or off-chip CMOS SRAM</td>
<td>CMOS DRAM</td>
<td>magnetic disk</td>
</tr>
<tr>
<td>Access time (ns)</td>
<td>0.25 – 0.5</td>
<td>0.5 – 25</td>
<td>80 – 250</td>
<td>5,000,000</td>
</tr>
<tr>
<td>Bandwidth (MB/sec)</td>
<td>20,000 – 100,000</td>
<td>5000 – 10,000</td>
<td>1000 – 5000</td>
<td>20 – 150</td>
</tr>
<tr>
<td>Managed by</td>
<td>compiler</td>
<td>hardware</td>
<td>operating system</td>
<td>operating system</td>
</tr>
<tr>
<td>Backed by</td>
<td>cache</td>
<td>main memory</td>
<td>disk</td>
<td>CD or tape</td>
</tr>
</tbody>
</table>
Single-processor vs multi-processor vs multi-core

- **Single-processor system**
  - Has one CPU
- **Multi-processor system**
  - Has multiple CPUs
- **Multi-core processor**
  - Hardware sees one CPU
  - Kernel/OS sees multiple CPUs
- **SMP vs assymetricMP**
- **Why multi-processor and multi-core?**
Overview of OS
OS operation

- Operation of OS is based on what is called *multi-programming* and *time-sharing*
- Time-sharing
  - Method of running multiple programs on a single CPU apparently concurrently
  - Use of *time slice*
- A process that needs I/O will have to be put “on hold”, i.e. *wait* and be *blocked*, until the I/O is complete
  - In the mean time, another process is executed
- What process to run is a decision of scheduling
- OS relies on hardware timers to time its operation
Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- Process needs resources to accomplish its task
- CPU, memory, I/O, files
- Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one *program counter* specifying location of next instruction to execute
- Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
- Concurrency by multiplexing the CPUs among the processes / threads
Memory Management and Swapping

- Not all active programs (and their data) fit in the RAM at the same time.
- Memory is organized in pieces called *pages*.
- Portion of each program is loaded at a time
  - The OS allocates a number of pages.
- When part of the program or data is referenced that is not loaded in the memory
  - Process is blocked
  - The appropriate part of program or data is loaded
  - Process is resumed.
- When memory is filled, then some page is stored back to the disk (with the changes) and RAM space is freed.
Dual-mode Operation

• Two modes
  - User mode (unprivileged mode)
  - Kernel mode (privileged, system, supervisor mode)
    • Allows to run *privileged* instructions of the ISA
Storage Management

- OS provides uniform, logical view of information storage
- Abstracts physical properties to logical storage unit - file
- Each medium is controlled by device (i.e., disk drive, tape drive)
- Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
- Files usually organized into directories
- Access control on most systems to determine who can access what
- OS activities include
  - Creating and deleting files and directories
  - Primitives to manipulate files and dirs
  - Mapping files onto secondary storage
  - Backup files onto stable (non-volatile) storage media
Data and program protection

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
- Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
- User identities (**user IDs**, security IDs) include name and associated number, one per user
- User ID then associated with all files, processes of that user to determine access control
- Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
- **Privilege escalation** allows user to change to effective ID with more rights
For next time...

- Reading
  - Chapter 1
- Assignment 1
  - Due on Thursday, August 25