

# Com S 352

# Introduction to Operating Systems

Class 0 - Syllabus

*August 21, 2017*

# Instructor

**Instructor:** Dr. Johnny Wong

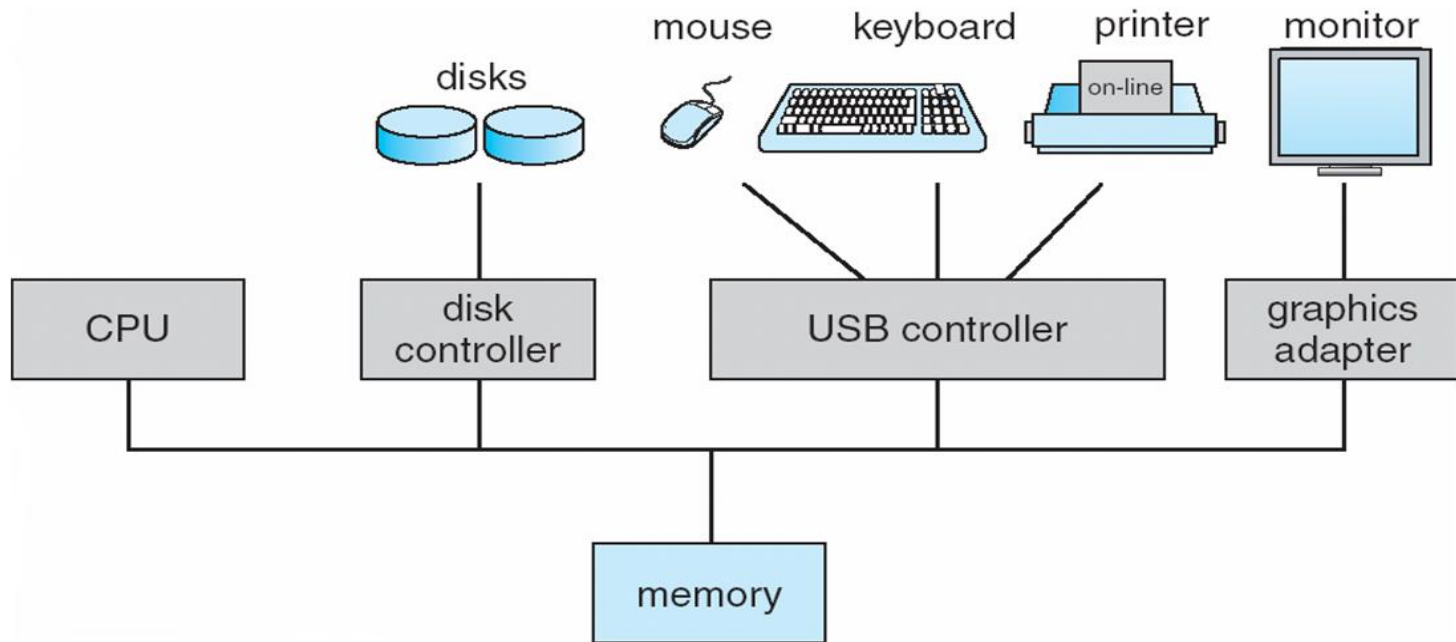
**Office:** Atanasoff 202

**Office Hours:** TT 10:00am-11:00am (or by appointment)

**Email:** wong@iastate.edu

Subject: Operating System

# Hardware Components of a Computer

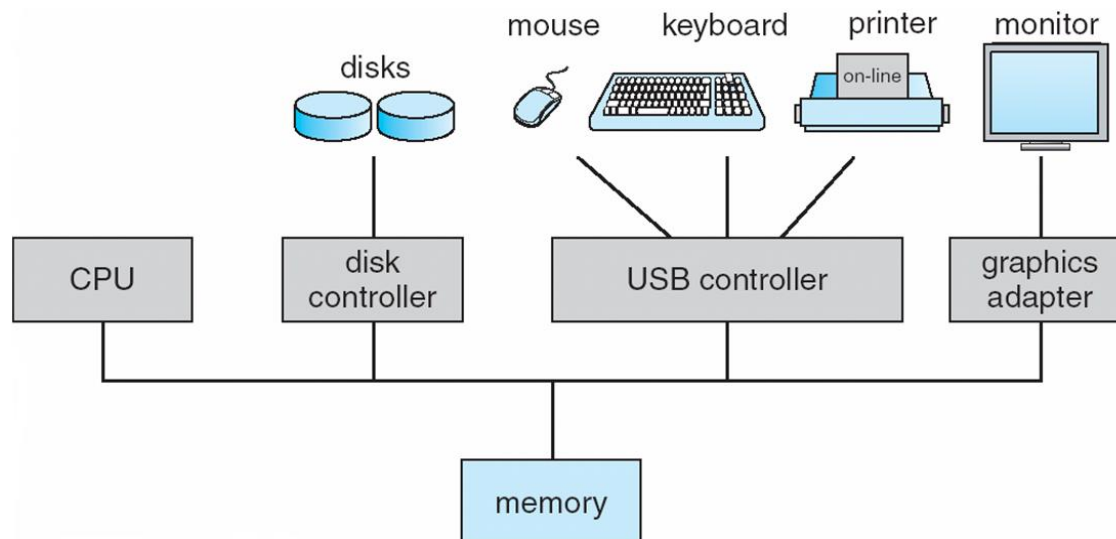


- CPU, memory, storage and other I/O devices connected by a bus
- The bus transports both data and control messages

# Computer Startup

**Bootstrap program** is loaded when a computer is powered-up or rebooted

- Typically stored in ROM or EPROM, generally known as firmware
- Initializing all aspects of system
- Loading **the most fundamental control software** (note: operating system kernel) and starts execution



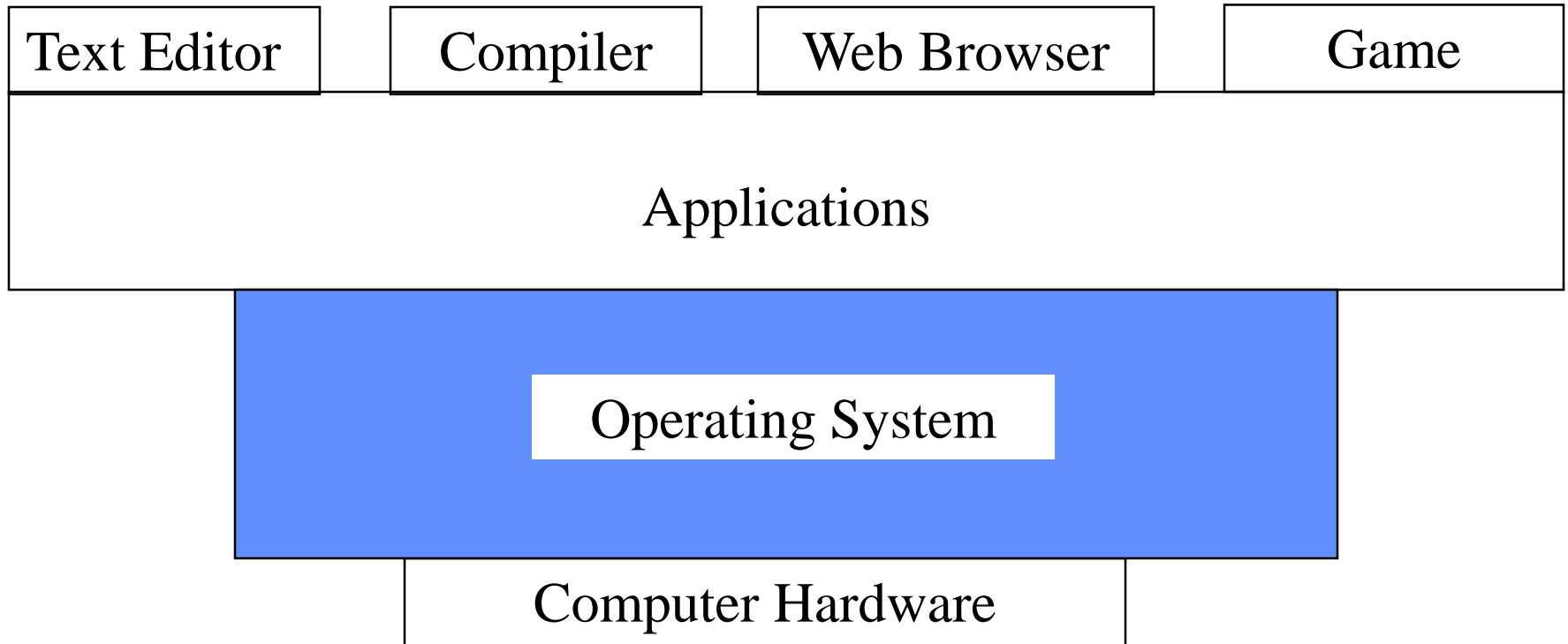
# What does an OS do?

- Manage execution resources (e.g., CPU, GPU, co-processors)
- Manage memory space (RAM)
- Manage permanent storage space (Disk, flash)
- Manage I/O devices (display, network card, sensors)
- Manage programs (code)
- Support communication between programs and between computers
- Keep the system secure and safe

# The Roles of an Operating System

- OS is a **resource allocator/manager**
  - 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





# What is this course about?

- Learn to use one or some operating system? NO!
- Learn to design an operating system? NO!
- We will learn how an operating system works
  - Some fundamental concepts
  - Some data structures
  - Some algorithms
  - Some theories

# More specifically ...

- Background: Hardware support; Key ideas and structures for OS
- Processes
- Threads
- Synchronization
- CPU Scheduling
- Deadlock

# More specifically ...

- Main Memory Management
- Virtual Memory
- File-System Interface & Implementation
- Mass-Storage Structure & I/O Systems
- Protection & Security
- Advanced topics: Distributed Operating Systems

# System Programming

- System calls, system programs
- Multi-process programs
- Inter-process communication: shared memory; message passing, socket
- Multi-thread programs: kernel-level threads; user-level thread libraries
- Develop a simple user-level thread library
- Inter-process and inter-thread synchronization

# Teaching Team

**Instructor:** Dr. Johnny Wong

**Office:** Atanasoff 202

**Office Hours:** TT 10:00am – 11:00am(or by appointment)

**Email:** wong@iastate.edu

**Teaching Assistant:** Mr. Abdullah & Eshita Zaman

**Office:** 0145 Pearson

**Office Hours:** Tue 4:00-5:00 (Abdullah), Thu 4:00-5:00 (Abdullah), Wed 10:00–11:00 and Fri 10:00-11:00 (Eshita)

**Email:** mabdulah @iastate.edu, ezaman@iastate.edu

# Meetings

## Classes (Gilman 1652, MWF 2:10-3:00pm)

- Attendance is highly encouraged. In-class quizzes will pop up “uniformly at random” to check attendance. The quizzes account for 5% of your final grade.

## Recitations (Gilman 1652, Thursday 12:10-1:00pm)

- introduce more examples for understanding OS concepts and system programming
- reviews for exams
- provide helps on homework and programming projects

# Pre-requisites & Text Book

## Prerequisites

- ComS321/CprE381, ComS 327; and Engl 250/105

## Textbook

- Silberschatz, Galvin, and Gagne, Operating System Concepts (9<sup>th</sup> Edition)
- Class notes

# Score Allocation

25% Almost Weekly assignments (no more than 10 homework assignments; the one with the lowest grade will be dropped)

25% ( $12.5\% \times 2$ ) Programming projects

30% ( $15\% \times 2$ ) Midterm exams

15% Final exam

5% Participation (solely based on the in-class attendance quizzes, ; the one with the lowest grade will be dropped)



# Weekly Assignment

- Announced on Friday; **due on the next Friday 11:59pm.**
- No “hand-in assignment” due on the 1st week or the weeks with exams.
- Discussed during recitation/help sessions.
- Announced, turned in, and graded through ISU on-line system.
- Must be submitted **in PDF or plaintext format (Other formats, such as .doc/docx, are not acceptable).** Scanned is fine but make sure your handwriting is neat and clear.

# Reading Assignment for Week #1

- Read Chapter 1 of the textbook

# Exams

- **2 midterm exams** and **1 final exam**.
- Every exam is 50-minute, closed-book/note exam.
- The *tentative* exam schedule:
  - *Friday, September 29* for Midterm 1.
  - Midterm 2 will be on *Friday, November 10*.
  - Final exam will be given according to the University Schedule.
- The final exam (non-accumulative) will cover the 3<sup>rd</sup> part lectures in depth and very general concepts from the parts covered by midterms.

# Programming Projects

## 2 programming projects

- Project 1: released in mid-September; due in late-October
- Project 2: released in mid-October; due in late-November.

## Project policies

- Projects must be turned in electronically by submitting all necessary source code (which may include test code) for correct compilation.
- Source code must include proper documentation to receive full credit.

# Programming Projects

## Project policies (Cont.)

- All projects require the use of a make file or a certain script file (accompanying with a readme file to specify how to use the script file to compile), such that the grader will be able to compile/build your executable by simply typing “make” or some simple command you specifies in your readme file.
- Source code must compile and run correctly on the department machine *pyrite*, which will be used by the TA for grading.
- Programs that do not compile successfully (warnings are acceptable) may receive zero point.

# Programming Projects

## Project policies (Cont.)

- Project assignments will provide specifications; however, implementation-specific details may not be given. It will be your responsibility to figure out the implementation approaches.
- You are responsible for thoroughly testing and debugging your code. Samples given with the project assignments are for explanation/illustration purpose only; the TA may try to break your code by subjecting it to bizarre test cases.

# Late Policy

Homework or programming projects will **not be accepted after 72 hours** later than the submission deadline.

- 10% of the full credit of the homework/project will be automatically deducted for submission later by 1 second to 1 day.
- 20% will be deducted for submission later by 1 day plus 1 second to 2 days.
- 30% will be deducted for submission later by 2 day plus 1 second to 3 days.

# Appealing Policy

Appealing for homework/programming project grades will be first handled by the TA within one week after the graded homework/project has been returned.

In case of disputes, the student should appeal to the instructor.

Appealing for exam grades will be handled similarly. The student should appeal to the TA and the instructor within one week after the graded exam is returned.



# We follow other University Policies at ISU

Academic misconduct

Disability accommodation

Dead week

Harassment and discrimination

Religious Accommodation

Please refer to the syllabus posted through on-line system for detail.