KOÇ UNIVERSITY

COMP304
Operating Systems (OS)

Introduction

Didem Unat

Lecture 1

Motivation

- Operating Systems: Major field of Computer Science and Engineering
  - The MOST important and challenging course

- Around 20% of questions in GRE Computer Science subject test are from the OS concepts

- Forms a good knowledge base for other subject areas
- Provides a complete understanding of software/hardware infrastructure

Prerequisite and Elements

- Course prerequisite:
  - COMP 303 Computer Architecture
  - COMP 132 Advanced Programming
- Good knowledge in
  - C programming
  - Data structures
  - Algorithms

What is an Operating System?

- A program that acts as an intermediary (supervisor) between a user of a computer and the computer resources

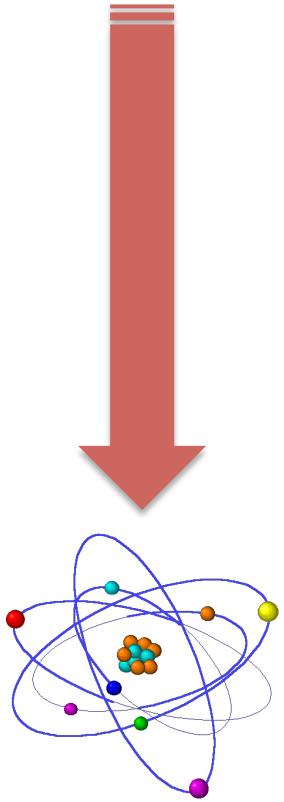
- Duties of an OS
  1) Provide resource abstraction
  2) Manage and coordinate resources
  3) Provide *security and protection*
  4) Provide *fairness* among users (or programs)

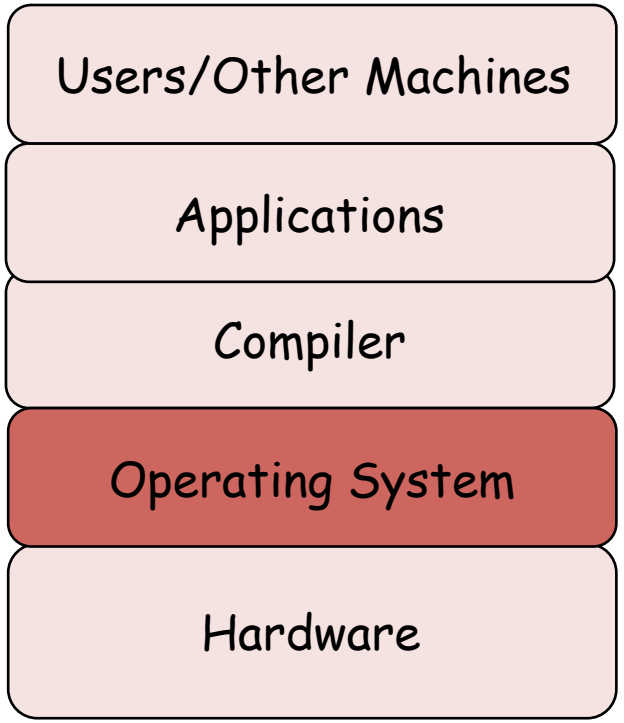
Computer Startup

- **Bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM, generally known as **firmware**
  - Initializes all aspects of a system
  - Loads operating system **kernel** into main memory and starts execution
    - The first system process is 'init' in Linux
  - When the system is fully booted, it waits for some event to occur
- Kernel
  - The ``one" program running at all times (the core of OS)
    - Everything else is an application program
- Process
  - An executing program (active program)

(1) OS creates resource abstractions

```
foo(int x) {…}
```

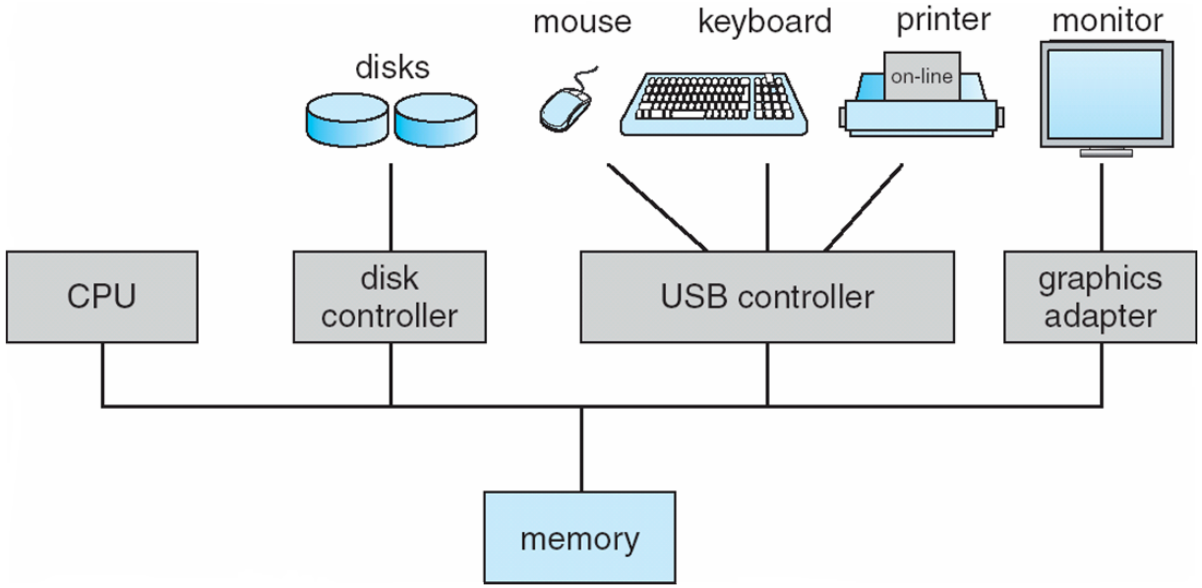




There are other layers in software stack such as runtime, libraries etc. Operating System and Compilers are essentials.

(2) OS manages resources

- OS is a **resource allocator**
  - Manages all resources for processes
  - Decides between conflicting requests for efficient and fair resource use



(3) OS provides protection and security

- OS is a **control program**
  - Controls execution of programs to prevent errors and improper/malicious use of the computer
  - Dual mode and Multimode OS
    - User mode and Kernel mode

User 1: I would like to open a file

User 2: I would like to edit a password file

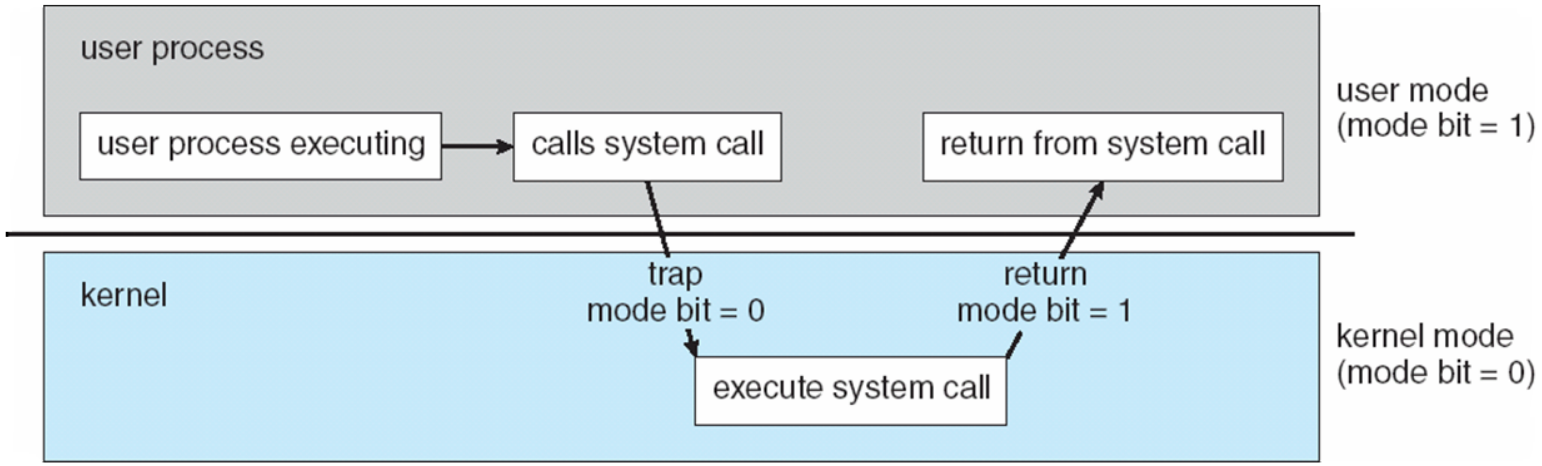
OS: Let me check if you have permissions to do so, wait a sec

OS: I'll open it for you!

OS: No, you don't have authorization for that file

(3) OS provides protection and security

- **System Call**
  - How a program requests a service from an OS
  - Results in a transition from user to kernel mode
  - Return from call resets it to user mode
- Software error or request creates exception or trap

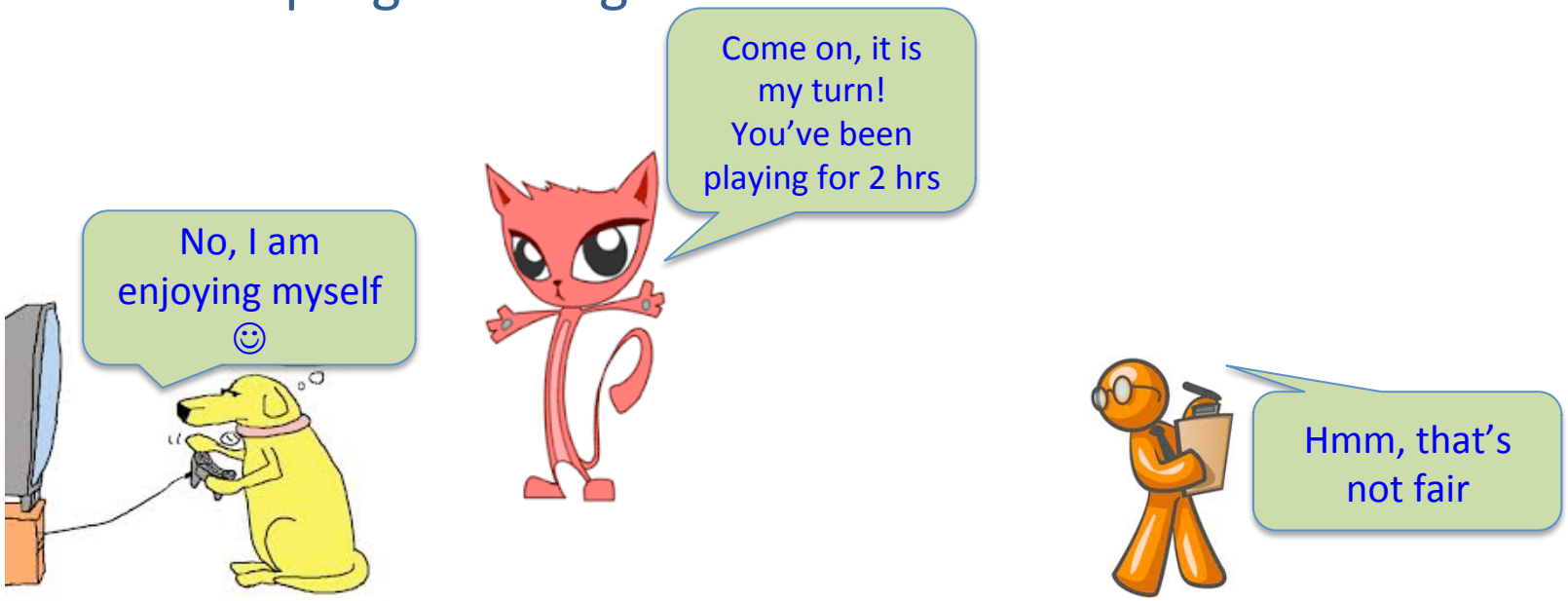


Interrupts

- An operating system is **interrupt driven**
  - It sits and waits for an event to occur
- Device or hardware interrupts
  - I/O device is done or
  - Hardware throws an exception (e.g overflow)
- Software interrupts
  - A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request (system call)
- OS has an **interrupt vector**, which contains the addresses of all the service routines for interrupt handling

(4) OS provides *fair* execution

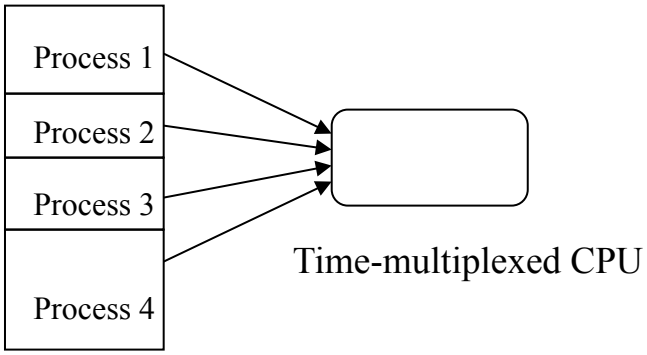
- OS provides fair execution and resource sharing between users and programs
  - via multiprogramming



How Multiprogramming Works

- **Multiprogramming** needed for efficiency
  - Single user or program cannot keep CPU and I/O devices busy at all times
  - Organize processes so that CPU always has one process to execute
  - A subset of total jobs is kept in main memory

- One job selected and run via **CPU scheduling**
  - When it has to wait (for I/O for example), OS switches to another job



Brief History of OS

- No operating system 1940s
  - Computers are exotic
  - Program in machine language
  - Programs manually loaded
  - No concurrency: no multiple jobs, no multiple users
- 1950s
  - First compiler is developed
  - OS uses batch scheduling
    - No human-computer interaction
    - Still used in servers, clusters and data centers today

Brief History of OS

- 1960s
  - Multics – one of the most important real OS
    - Hierarchical file system (directory structure)
    - Access control list and protection
- 1970s
  - Computers became affordable
  - UNIX is born at Bell Labs by Ken Thompson and Dennis Ritchie
    - Written in C, allow people to experiment

Brief History of OS

- 1980s
  - MS-DOS
    - IBM needed software for their personal computers
    - Approached Bill Gates (Microsoft) and he created MS-DOS
  - BSD Unix
    - University of California developed BSD Unix
    - Became open source later
  - Mach
    - Carnegie Mellon Univ. developed Mach to replace Unix
    - Apple chose BSD/Mach as the foundation for MacOS X
- 1983
  - Richard Stallman started the GNU project
    - Advocates free, open-source UNIX compatible operating system
    - GNU General Public License (GPL) is now a common license under which free software is released

Brief History of OS

- 1990s
  - Linux
    - Developed by a student (Linus Torvalds) in Finland
    - Unix-based
    - Several distributions: SUSE, Fedora, Ubuntu, Redhat
    - Open-source operating system under GNU General Public License
  - Windows 95 and MacOS X became mature and complex
- 2000s
  - Mobile devices: Android (based on Linux)
  - Trend is to have a smaller OS (network storage)
  - Virtualization has become common (Vmware Player, VirtualBox etc.)

Course Basics

- Website
  - Blackboard : https://ku.blackboard.com/
  - All course materials will be posted
- *Main Book*
  - *Operating System and Concepts (9th edition)*
    - *By Silberschatz, Galvin and Gagne*
- *Additional Book*
  - *Linux Kernel Development (3rd Eddition)*
    - *By Robert Love*
    - *http://it-ebooks.info/book/819/*

Linux Operating System

- In assignments and projects, we will be using Linux environment. You have two options:

BACK UP YOUR DATA

1) Install a Linux OS environment (recommended)

   Installation package (latest distributions of Ubuntu or Fedora)

   Install as dual boot on your own computer

2) Install Linux Virtual Machine on your computer

   http://people.westminstercollege.edu/faculty/ggagne/osc/vm/index.html

Have it ready by next Thursday for the PS, consult your TAs or peers if you have problems

Linux Tutorial

- Learning Unix commands
  - http://home.ku.edu.tr/~oozkasap/comp304/unixtut/index.html
  - Study the first 4 sections by next PS hour
    - Experiment with these basic commands
    - First PS will go over the commands
  - First assignment will require you to have a running Linux environment and basic Unix command knowledge

Grading

- Grading
  - %10 Written/Coding Assignments (3 of them)
  - %30 Projects (3 of them)
  - %23 Midterm
  - %35 Final
  - %02 Participation, and Random Attendance

Course Policy

- NO CELL phones
- NO laptops or other similar devices
- NO talking
- NO late arrival or early leave to/from the lecture

- **You must submit your own work in all assignments, projects and exams. Academic dishonesty includes using other people's words or ideas without acknowledgement, cheating on exams, projects, and homework.**
- **In case any of the academic dishonesties are disclosed, disciplinary action will follow.**

Office Hours

PS hours on Thursdays at 17.30

- Didem Unat's Office Hours at ENG 129C
- Office hours are at 4-6 pm everyday
  - Fareed on Monday
  - Doğa on Tuesday
  - Ilyas on Wed
  - Didem on Thursday (4-5.15 pm)

Doga

Fareed

Ilyas

Reading

- From text book
  - Read 1.1, 1.4-1.10 (OS Structure – Kernel Data Structures)
  - Read 1.12 (Open-Source OS)
  - Read 1.2-1.3 if you want to refresh your Computer Architecture knowledge
- Install the Linux Distribution or Virtual Machine by next Thursday
- Subscribe to Blackboard Discussion Forum
- Unix Tutorial for Beginners (first 4 sections)
  - http://home.ku.edu.tr/~oozkasap/comp304/unixtut/index.html

Acknowledgments

- These slides are adapted from
  - Öznur Özkasap (Koç University)
  - Operating System and Concepts (9th edition) Wiley