CSE 3320
Operating Systems
Introduction

Jia Rao

Department of Computer Science and Engineering

http://ranger.uta.edu/~jrao

Outline

- Why study Operating Systems ?

- What to learn ?

- Course structure

- Course policy

- OS overview

Why Study Operating Systems ?

- The most complex software

  o ~ 12 million lines of code in Linux

- The most fundamental software

  o OSes are almost everywhere, e.g., supercomputer, PC, phone…

- By studying OS, you will

  o Learn how computers work

  o Gain a good understanding of OS and hardware

  o Learn about system design

    ▸ Simplicity, portability, performance, and trade-offs

What to Learn ?

- Hardware abstraction

  o processes, threads, files …

- Resource management

  o CPU scheduling, memory management, file systems …

- Coordination

  o Multiple programs and users

  o Fairness and efficiency

- Case studies: Linux

Course Structure

- Lectures
  - Tuesday and Thursday 12:30PM – 1:50 PM
  - Online lecture on Microsoft Teams

- Homework
  - 2 written assignments

- Projects
  - 4 programming assignments
  - 2 students team up

- Exams (closed books, one cheat sheet)
  - Midterm: in class, Mar. 11.
  - Final: 8:00AM – 10:30AM, May 6.

Course Policy

- Grading scale

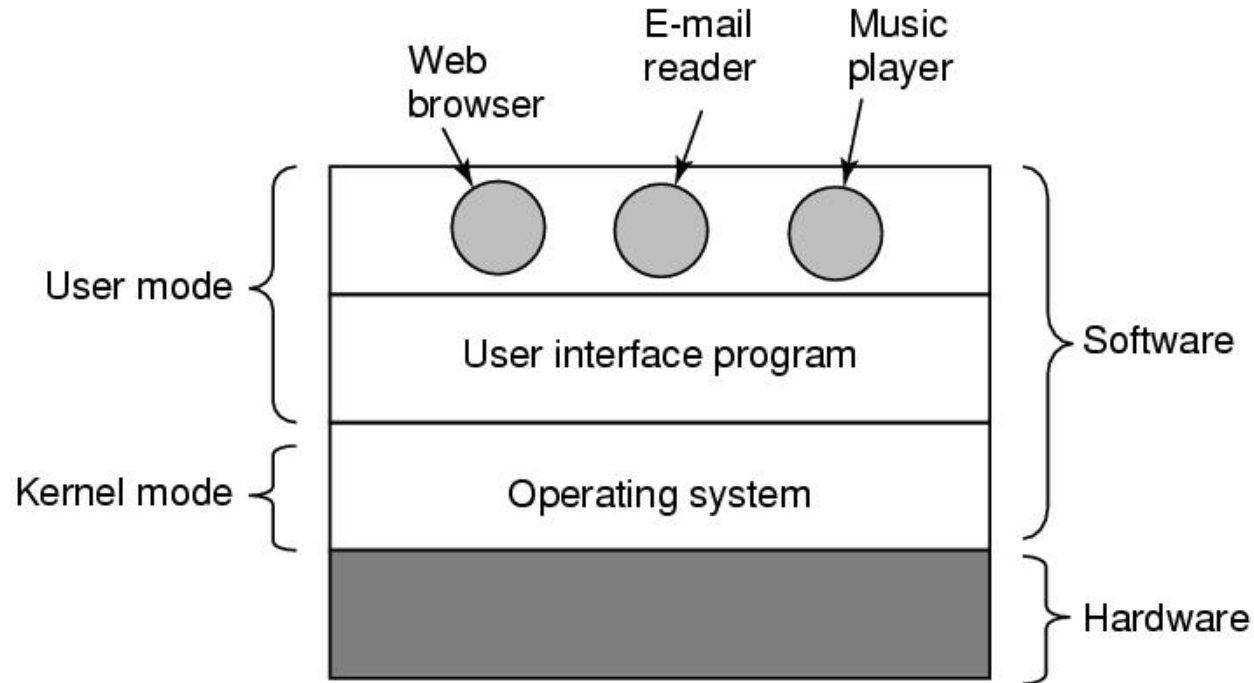| Percentage	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	E/F

Grading Policy (cont.)

- Grading percentage
  - In-class discussion: 5%
  - Homework assignments: 10%
  - Projects: 35%
  - Midterm: 18%
  - Final exam: 32%

Where to seek help ?

- Ask questions in class on Teams

- Attend office hours
  - o Dr. Jia Rao: T/Th 10:00AM – 11:00AM
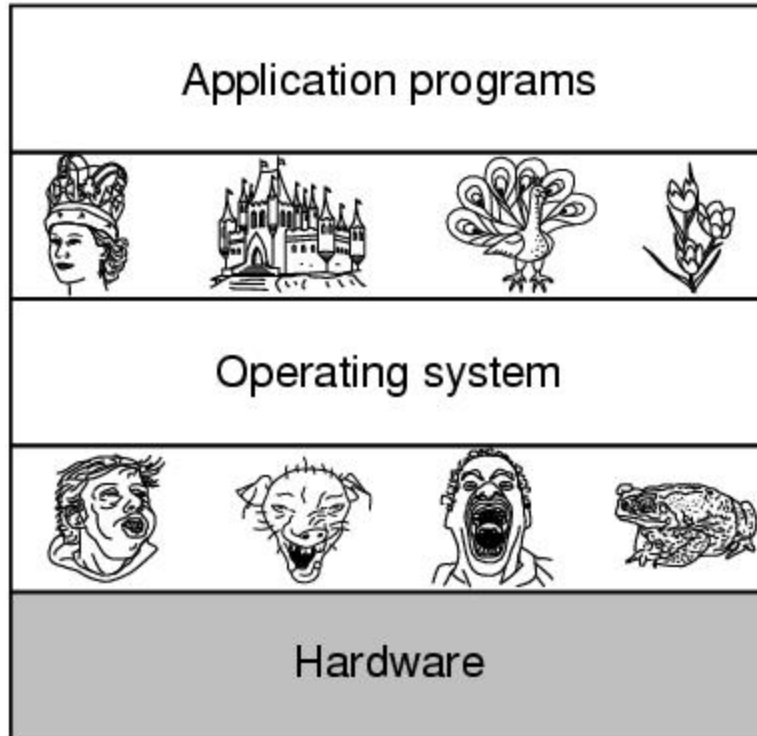
What is an Operating System ?

- A computer system consists of
  - hardware
  - system programs
  - application programs

What does an Operating System do ?

- It is an extended (or virtual) machine
  - Hides the messy details which must be performed
  - Presents user with a virtual machine, easier to use
  - Protection domain

- It is a resource manager
  - Each program gets time with the resource, e.g., CPU
  - Each program gets space on the resource, e.g., MEM

The Operating System as an Extended Machine
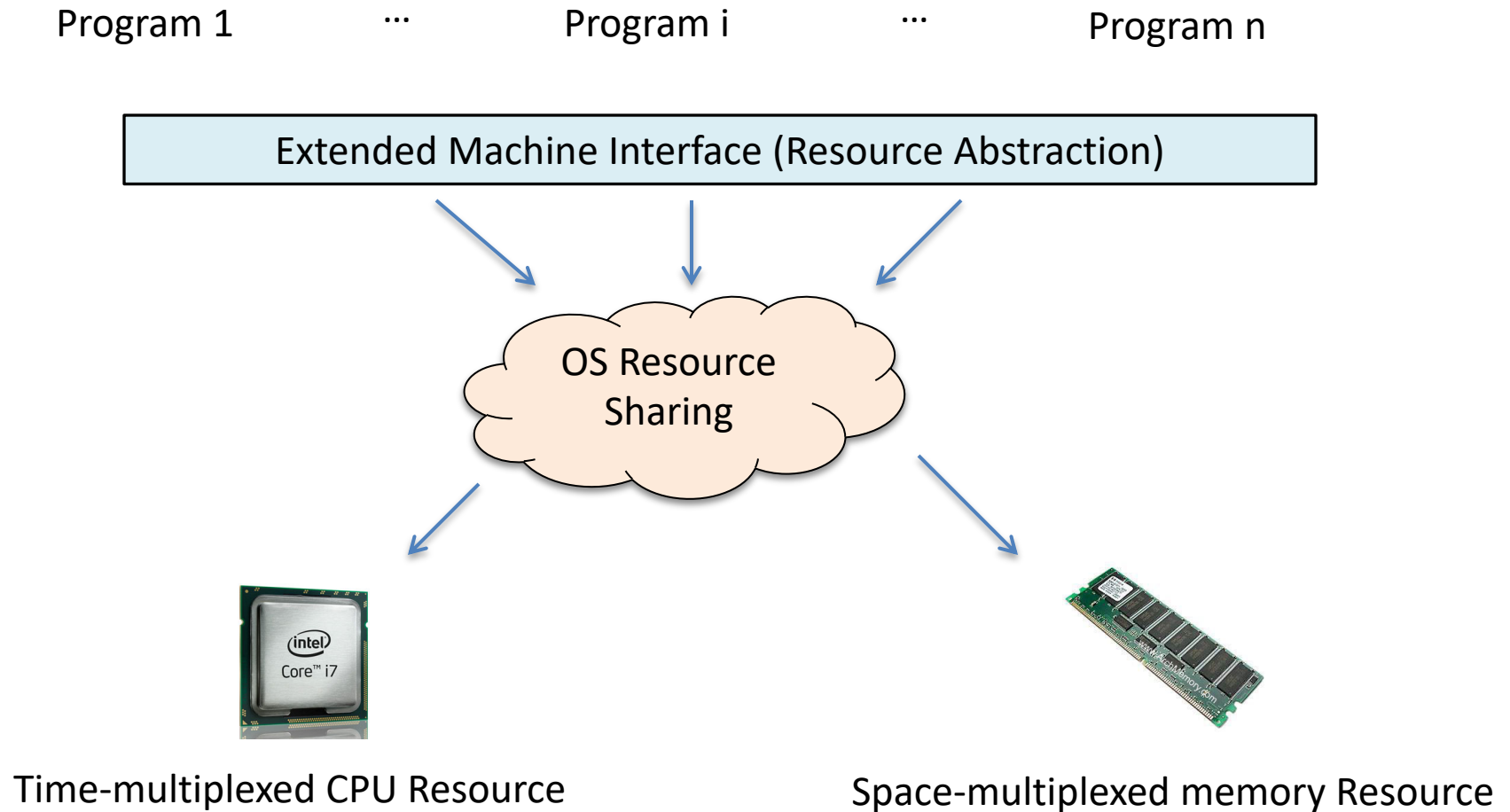


```
fprintf(fd, "%d", data);


write(fd, buffer, count) ;


file->f_op->write(file, buf,
count, pos);




        .
        .
        .


load(block, length, device);
seek(device, track);
out(device, sector);
```

The Operating System as a Resource Manager

Program 1   ···   Program i   ···   Program n

Extended Machine Interface (Resource Abstraction)

OS Resource Sharing

Time-multiplexed CPU Resource

Space-multiplexed memory Resource

Objectives of Resource Abstraction

- Resource abstraction
  - Mask complexity
  - Cover multiple devices
  - Reliability

- Resource sharing
  - Efficiency
  - Fairness
 - Protection and security

How does an OS work ?

- Dual mode operation

  o User mode (application)

  o Kernel mode (OS kernel)

- Transition between user/kernel mode

  o interrupt – HW device requests OS services

  o Trap – user program requests OS services

  o Exception – error handling

Different Types of OS

- Batch processing
  - Processes jobs one by one
- Time sharing OS
  - Processes multiple jobs in "round robin"
- Real-time OS
  - Still time-sharing, but has deadlines for certain jobs
- Distributed OS
  - Multiple computers run a single copy of OS
- Embedded OS
  - Runs on cell phones, PDAs, tailored and highly efficient
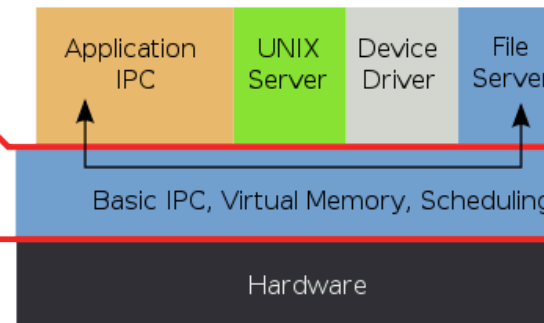
The Structure of OS

Monolithic Kernel based Operating System
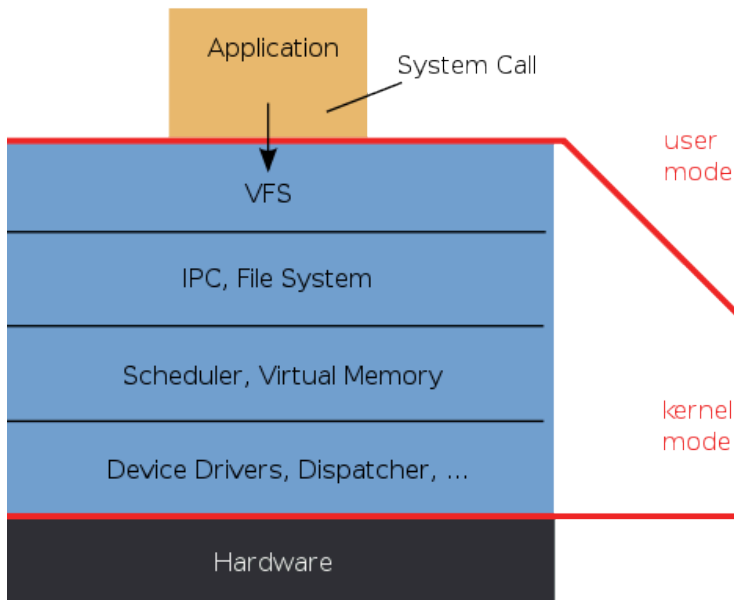
Microkernel based Operating System

Hybrid Kernel

UNIX, Linux, Windows 98

Mach

Windows NT, Mac OS

Advantage v.s. disadvantage ?

History of Operating Systems

- First generation 1945 - 1955
  - vacuum tubes, plug boards
- Second generation 1955 - 1965
  - transistors, batch systems
- Third generation 1965 – 1980
  - ICs and multiprogramming
- Fourth generation 1980 – present
  - personal computers
- Present – next 5-10 years
  - Mobile devices
  - Many-core computers

Summary

- An OS is just a special program

  - Two functionalities: resource abstraction and sharing

  - Provides services to user programs

- Three ways to request OS services

  - Interrupt, trap, and exception

- Next class

  - Overview of computer hardware

  - Organization of operating systems