بسم الله الرحمن الرحیم

۱ «سیستم عامل»

جلسه ۱: مقدمه

معرفی
سیستم عامل

What is an operating system?

What is an operating system? 3

- **Operating system --"a program that controls the execution of application programs and implements an interface between the user of a computer and the computer hardware"**
  - Narrow view of a computer and OS
    - Traditional computer with applications running on it (e.g. PCs, Workstations, Servers)
  - Broad view of a computer and OS
    - Anything that needs to manage resources (e.g. router OS, embedded system, cell phone OS ...)

Two key OS functions

Two key OS functions

- **Abstract Machine**
  - Hides complex details of the underlying hardware
  - Provides common API to applications and services
  - Simplifies application writing

Two key OS functions

- **Abstract Machine**
  - Hides complex details of the underlying hardware
  - Provides common API to applications and services
  - Simplifies application writing
- **Resource Manager**
  - Controls accesses to shared resources
    - CPU, memory, disks, network, ...
  - Allows for global policies to be implemented

Why is abstraction important?

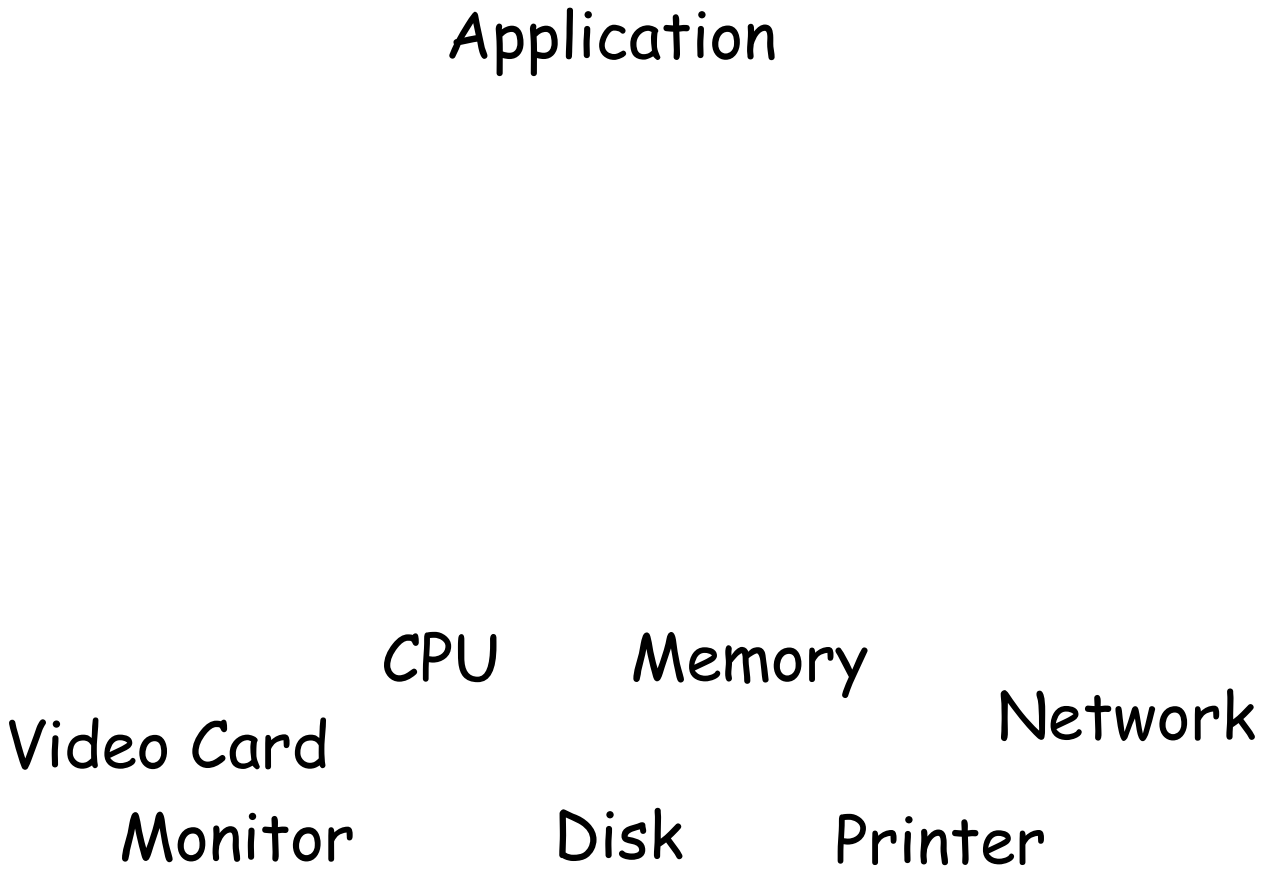
Why is abstraction important? 5

- **Without OSs and abstract interfaces, application writers must program all device access directly**
  - load device command codes into device registers
  - handle initialization, recalibration, sensing, timing etc for physical devices
  - understand physical characteristics and layout
  - control motors
  - interpret return codes … etc

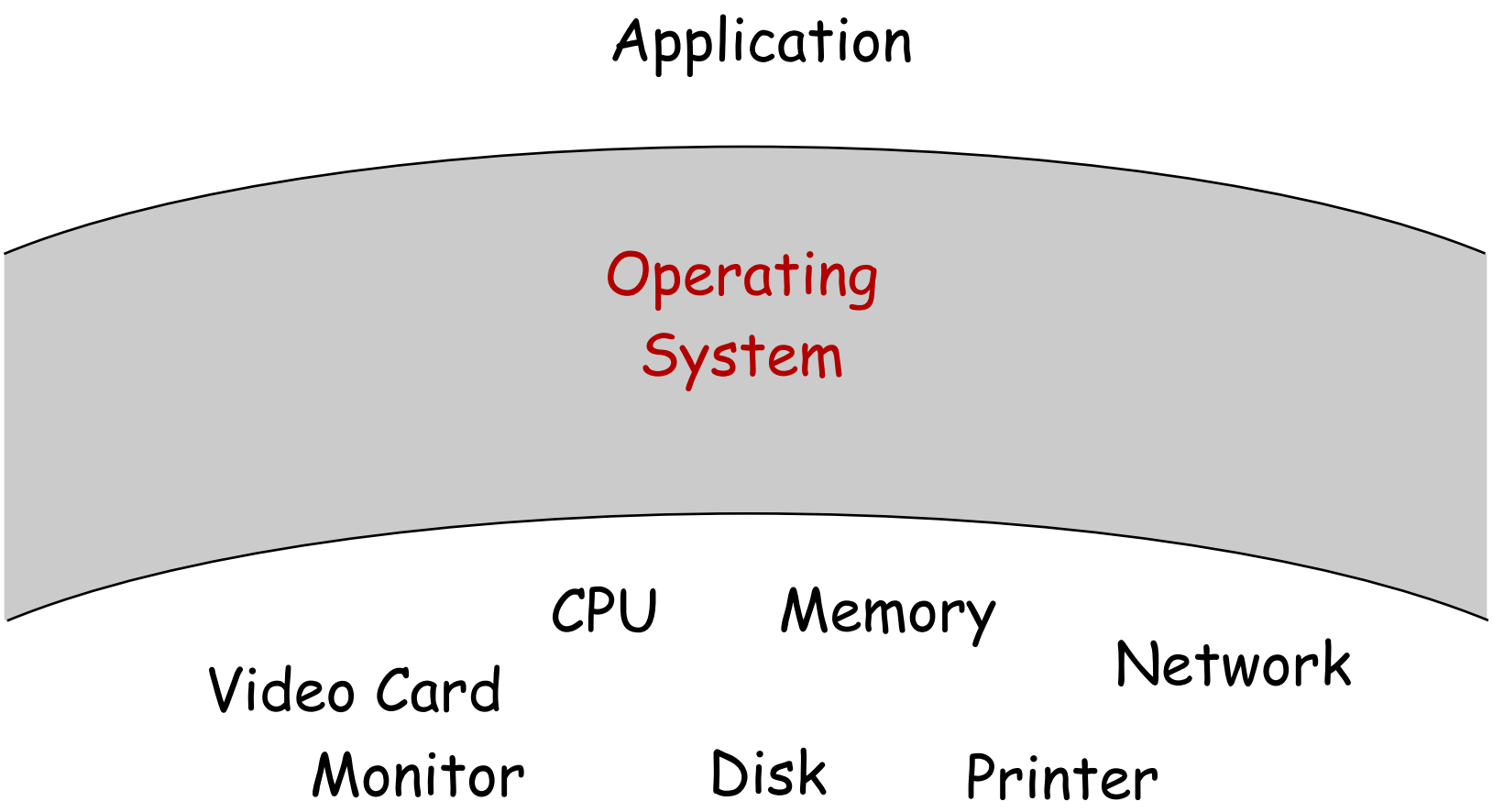
Why is abstraction important? 5

- **Without OSs and abstract interfaces, application writers must program all device access directly**
  - load device command codes into device registers
  - handle initialization, recalibration, sensing, timing etc for physical devices
  - understand physical characteristics and layout
  - control motors
  - interpret return codes … etc
- **Applications suffer severe code bloat!**
  - very complicated maintenance and upgrading
  - no portability
  - writing this code once, and sharing it, is how OS began!

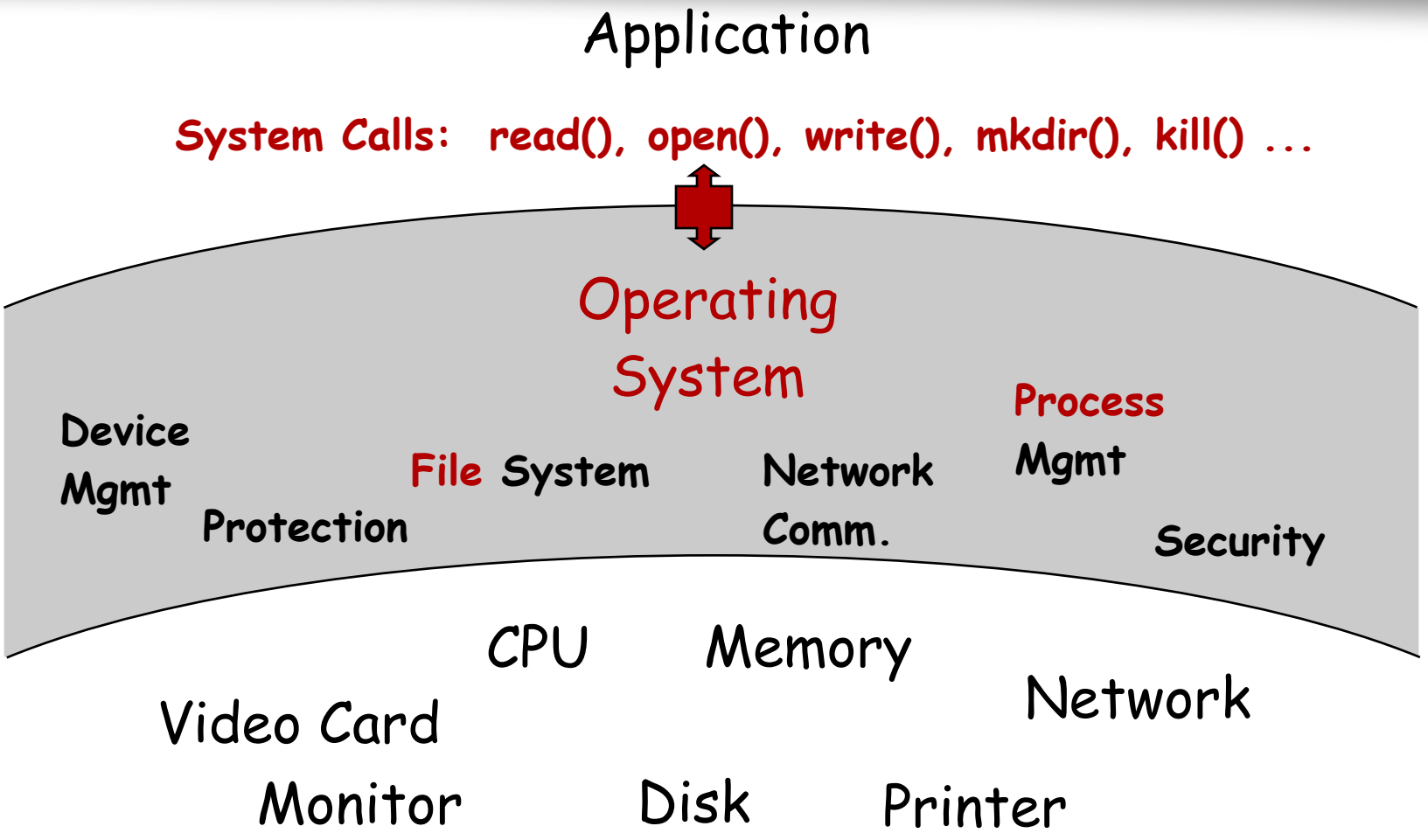
Providing abstraction via system calls



Providing abstraction via system calls



Providing abstraction via system calls



OS as a resource manager

OS as a resource manager

- **Allocating resources to applications across space and time**
  - time sharing a resource (scheduling)
  - space sharing a resource (allocation)

OS as a resource manager

- **Allocating resources to applications across space and time**
  - time sharing a resource (scheduling)
  - space sharing a resource (allocation)
- **Making efficient use of limited resources**
  - improving utilization
  - minimizing overhead
  - improving throughput/good put

OS as a resource manager

- **Allocating resources to applications across space and time**
  - time sharing a resource (scheduling)
  - space sharing a resource (allocation)
- **Making efficient use of limited resources**
  - improving utilization
  - minimizing overhead
  - improving throughput/good put
- **Protecting applications from each other**
  - enforcement of boundaries

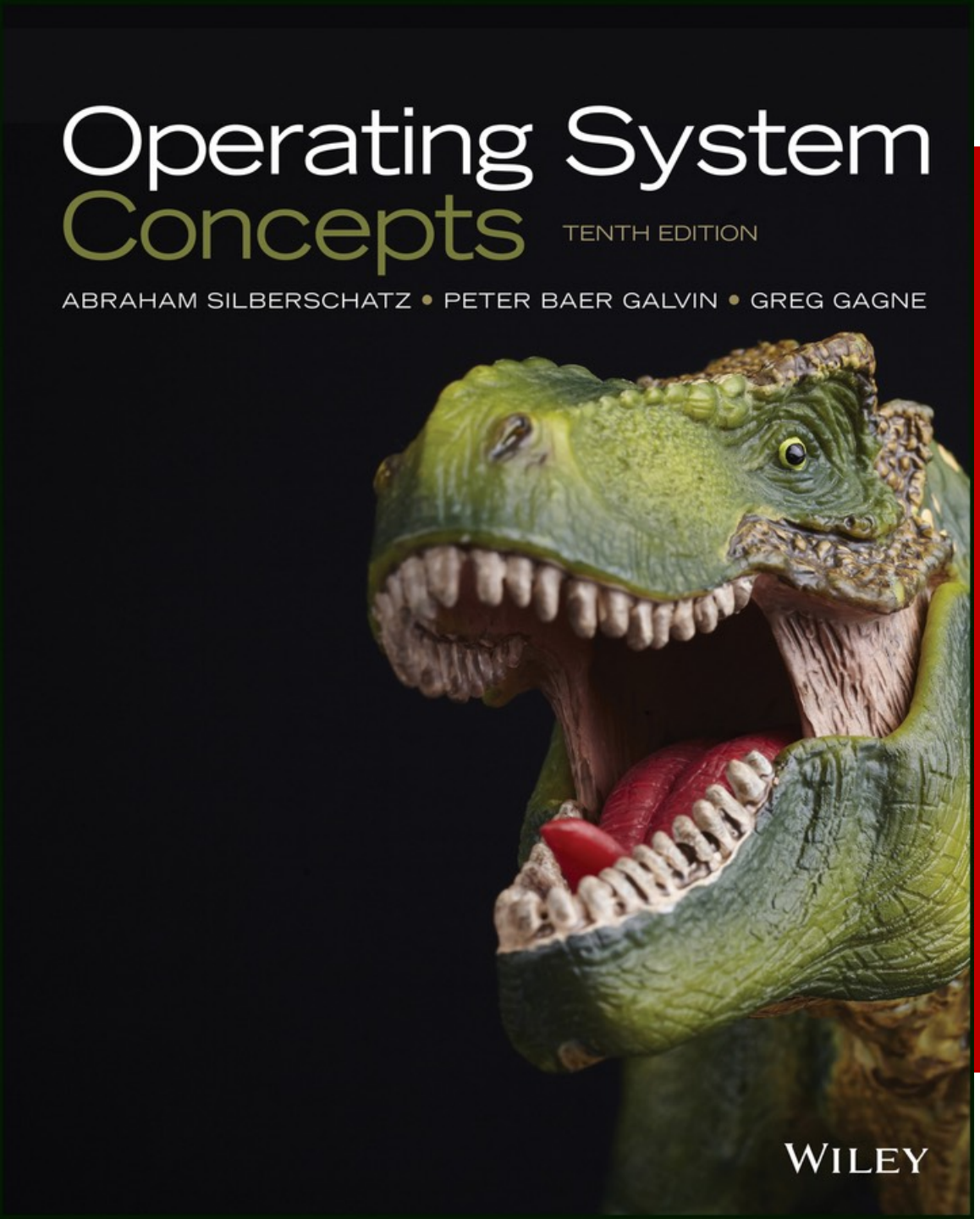
Problems an OS must solve

- Time sharing the CPU among applications
- Space sharing the memory among applications
- Space sharing the disk among users
- Time sharing access to the disk
- Time sharing access to the network

More problems an OS must solve

- **Protection**
  - of applications from each other
  - of user data from other users
  - of hardware/devices
  - of the OS itself!

- **The OS is just a program! It needs help from the hardware to accomplish these tasks!**
  - When an application is running, the OS is not running!
  - When the OS is not running, it can't do anything!


Operating System
Concepts
TENTH EDITION
ABRAHAM SILBERSCHATZ • PETER BAER GALVIN • GREG GAGNE
WILEY

کتاب درس

PART ONE ■ OVERVIEW

Chapter 1 Introduction

Chapter 2 Operating-System Structures

PART TWO ■ PROCESS MANAGEMENT

Chapter 3 Processes

Chapter 4 Threads & Concurrency

Chapter 5 CPU Scheduling

PART THREE ■ PROCESS SYNCHRONIZATION

Chapter 6 Synchronization Tools

Chapter 7 Synchronization Examples

Chapter 8 Deadlocks

PART FOUR ■ MEMORY MANAGEMENT

Chapter 9 Main Memory

Chapter 10 Virtual Memory

PART FIVE ■ STORAGE MANAGEMENT

Chapter 11 Mass-Storage Structure

Chapter 12 I/O Systems

PART SIX ■ FILE SYSTEM

Chapter 13 File-System Interface

Chapter 14 File-System Implementation

Chapter 15 File-System Internals

PART SEVEN ■ SECURITY AND PROTECTION

Chapter 16 Security

Chapter 17 Protection

PART EIGHT ■ ADVANCED TOPICS

Chapter 18 Virtual Machines

Chapter 19 Networks and Distributed Systems

PART NINE ■ CASE STUDIES

Chapter 20 The Linux System

Chapter 21 Windows 10

PART TEN ■ APPENDICES

Chapter A Influentia Operating Systems

Chapter B Windows 7

Chapter C BSD UNIX

CHAPTER 1

Introduction

An **operating system** is software that manages a computer's hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware. An amazing aspect of operating systems is how they vary in accomplishing these tasks in a wide variety of computing environments. Operating systems are everywhere, from cars and home appliances that include "Internet of Things" devices, to smart phones, personal computers, enterprise computers, and cloud computing environments.

In order to explore the role of an operating system in a modern computing environment, it is important first to understand the organization and architecture of computer hardware. This includes the CPU, memory, and I/O devices, as well as storage. A fundamental responsibility of an operating system is to allocate these resources to programs.

Because an operating system is large and complex, it must be created piece by piece. Each of these pieces should be a well-delineated portion of the system, with carefully defined inputs, outputs, and functions. In this chapter, we provide a general overview of the major components of a contemporary computer system as well as the functions provided by the operating system. Additionally, we cover several topics to help set the stage for the remainder of the text: data structures used in operating systems, computing environments, and open-source and free operating systems.

CHAPTER OBJECTIVES

- Describe the general organization of a computer system and the role of interrupts.
- Describe the components in a modern multiprocessor computer system.
- Illustrate the transition from user mode to kernel mode.
- Discuss how operating systems are used in various computing environments.
- Provide examples of free and open-source operating systems.

1.12 Summary

- An operating system is software that manages the computer hardware, as well as providing an environment for application programs to run.
- Interrupts are a key way in which hardware interacts with the operating system. A hardware device triggers an interrupt by sending a signal to the CPU to alert the CPU that some event requires attention. The interrupt is managed by the interrupt handler.
- For a computer to do its job of executing programs, the programs must be in main memory, which is the only large storage area that the processor can access directly.
- The main memory is usually a volatile storage device that loses its contents when power is turned off or lost.

Practice Exercises

1.1 What are the three main purposes of an operating system?

1.2 We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to "waste" resources? Why is such a system not really wasteful?

1.3 What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

1.4 Keeping in mind the various definitions of ***operating system,*** consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.

Programming Problems

2.24 In Section 2.3, we described a program that copies the contents of one file to a destination file. This program works by first prompting the user for the name of the source and destination files. Write this program using either the POSIX or Windows API. Be sure to include all necessary error checking, including ensuring that the source file exists.

Once you have correctly designed and tested the program, if you used a system that supports it, run the program using a utility that traces system calls. Linux systems provide the `strace` utility, and macOS systems use the `dtruss` command. (The `dtruss` command, which actually is a front end to `dtrace`, requires `admin` privileges, so it must be run using `sudo`.) These tools can be used as follows (assume that the name of the executable file is `FileCopy`:

Linux:

```
strace ./FileCopy
```

macOS:

```
sudo dtruss ./FileCopy
```

Since Windows systems do not provide such a tool, you will have to trace through the Windows version of this program using a debugger.

Programming Projects

Introduction to Linux Kernel Modules

In this project, you will learn how to create a kernel module and load it into the Linux kernel. You will then modify the kernel module so that it creates an entry in the `/proc` file system. The project can be completed using the Linux virtual machine that is available with this text. Although you may use any text editor to write these C programs, you will have to use the ***terminal*** application to compile the programs, and you will have to enter commands on the command line to manage the modules in the kernel.

As you'll discover, the advantage of developing kernel modules is that it is a relatively easy method of interacting with the kernel, thus allowing you to write programs that directly invoke kernel functions. It is important for you to keep in mind that you are indeed writing ***kernel code*** that directly interacts with the kernel. That normally means that any errors in the code could crash the system! However, since you will be using a virtual machine, any failures will at worst only require rebooting the system.

پیش‌نیاز

- شناخت سخت‌افزار
  - => اصول سیستم
- برنامه‌نویسی زیاد

نمره‌دهی

- پایان‌ترم: ۶ نمره.
- میان‌ترم: ۵ نمره.
- آزمونک‌ها: ۴ نمره (یکی از آزمونک‌ها برای هر فردی حذف می‌شود).
- پروژه: ۵ نمره.
- مجموعا پروژه‌ها را می‌توانید ۷ روز با تاخیر تحویل بدهند بدون کم شدن نمره.

پروژه درس

http://web.cecs.pdx.edu/~harry/Blitz/index.html

کلاس‌های حل تمرین

- چند کلاس حل تمرین برای پروژه‌ها
  - اولین جلسه: راه‌اندازی پروژه
- زمان جلسه‌ها در یک نظر سنجی مشخص می‌شود.

حل تمرین‌ها

- آقای توفیقی
- خانم قاسمی
- آقای رئیسی
- آقای کشیری

مباحث

CS 333
Introduction to Operating Systems

Class 1 - Introduction to OS-related Hardware and Software

Jonathan Walpole
Computer Science
Portland State University

زمان‌بندی

جلسه	تاریخ	موضوع جلسه	آزمونک	تمرین
1	۹۹/۱۱/۲۶	Course Overview and Introduction to Operating Systems Reading: Chapters 1 and 2		
2	۹۹/۱۱/۲۸	The Process Concept Reading: Chapter 3		
3	۹۹/۱۲/۰۳	Threads and Concurrency Reading: Chapter 4		حل تمرین ۱ (این هفته)
4	۹۹/۱۲/۰۵			
5	۹۹/۱۲/۱۰	Synchronization Primitives Reading: Chapter 6	آزمونک ۱	تمرین سری ۱ (یک هفته)
6	۹۹/۱۲/۱۲			
7	۹۹/۱۲/۱۷	Classic Synchronization Problems Reading: Chapter 6		تمرین سری ۲ (دو هفته)
8	۹۹/۱۲/۱۹		آزمونک ۲	حل تمرین ۲ (این هفته)
9	۹۹/۱۲/۲۴	Monitors and Message Passing Reading: Chapter 6		
10	۹۹/۱۲/۲۶			

زمان‌بندی (ادامه)

11	۰۰/۱/۱۵	Deadlock Reading: Chapter 6		تمرین سری ۳ (سه هفته)
12	۰۰/۱/۱۷			حل تمرین ۳
13	۰۰/۱/۲۲	Scheduling Reading: Chapter 5		
	۰۰/۱/۲۴	میان‌ترم		
14	۰۰/۱/۲۹			
15	۰۰/۱/۳۱	Memory Management Reading: Chapter 7	آزمونک ۳	
16	۰۰/۲/۰۵			
17	۰۰/۲/۰۷	Virtual Memory 1 Reading: Chapter 8		تمرین سری ۴ (دو هفته)
18	۰۰/۲/۱۲	Virtual Memory 2 Reading: Chapter 8	آزمونک ۴	حل تمرین ۴
19	۰۰/۲/۱۴	Virtual Memory 3 Reading: Chapter 8		
20	۰۰/۲/۱۹	Paging Algorithms Reading: Chapter 8		

زمان‌بندی (ادامه)

21	۰۰/۲/۲۱			
22	۰۰/۲/۲۶	Input/Output Reading: Chapter 12	آزمونک ۵	تمرین سری ۵ (۵و۷) (چهار هفته)
23	۰۰/۲/۲۸			
24	۰۰/۳/۰۲	Secondary Storage Management Reading: Chapter 11		
25	۰۰/۳/۰۴			
26	۰۰/۳/۰۹	File Systems 1 Reading: Chapters 9 and 10		
27	۰۰/۳/۱۱			
	۰۰/۳/۱۶	تعطیل رسمی		
28	۰۰/۳/۱۸	File Systems 2 Reading: Chapter 10		

http://foroughmand.ir/?page_id=699

سوال؟