

Lecture Note 0: Course Introduction

March 4, 2024

Jongmoo Choi

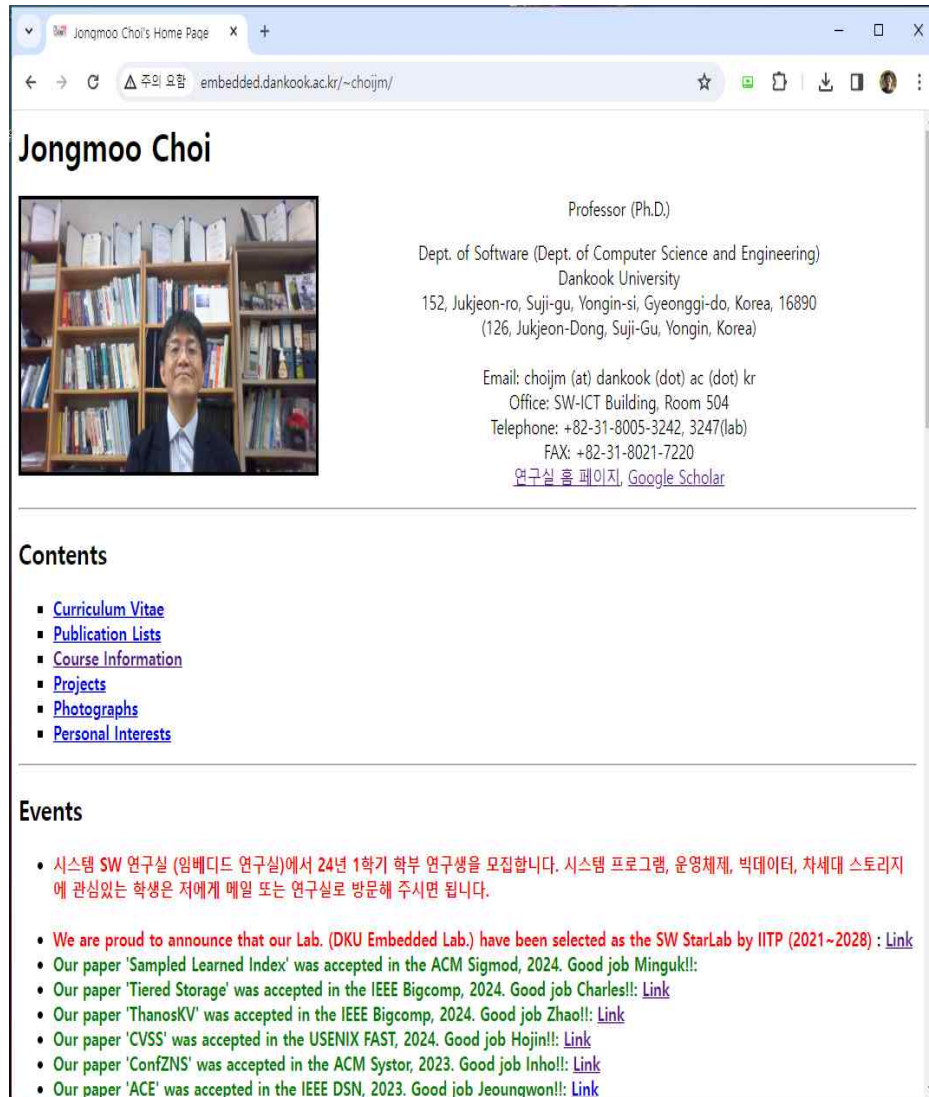
Dept. of Software
Dankook University

<http://embedded.dankook.ac.kr/~choijm>

(Copyright © 2024 by Jongmoo Choi, All Rights Reserved. Distribution requires permission.)

How to access lecture contents?

■ Lecture site



The screenshot shows the home page of Jongmoo Choi. The browser address bar indicates the URL is `embedded.dankook.ac.kr/~choijm/`. The page features a header with the name "Jongmoo Choi" and a profile picture. To the right of the photo, his title "Professor (Ph.D.)" and department "Dept. of Software (Dept. of Computer Science and Engineering) Dankook University" are listed, along with his address: "152, Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, Korea, 16890 (126, Jukjeon-Dong, Suji-Gu, Yongin, Korea)". Contact information includes an email address `choijm (at) dankook (dot) ac (dot) kr`, office location "SW-ICT Building, Room 504", telephone number "+82-31-8005-3242, 3247(lab)", and fax number "+82-31-8021-7220". There are also links for "연구실 홈페이지" and "Google Scholar".

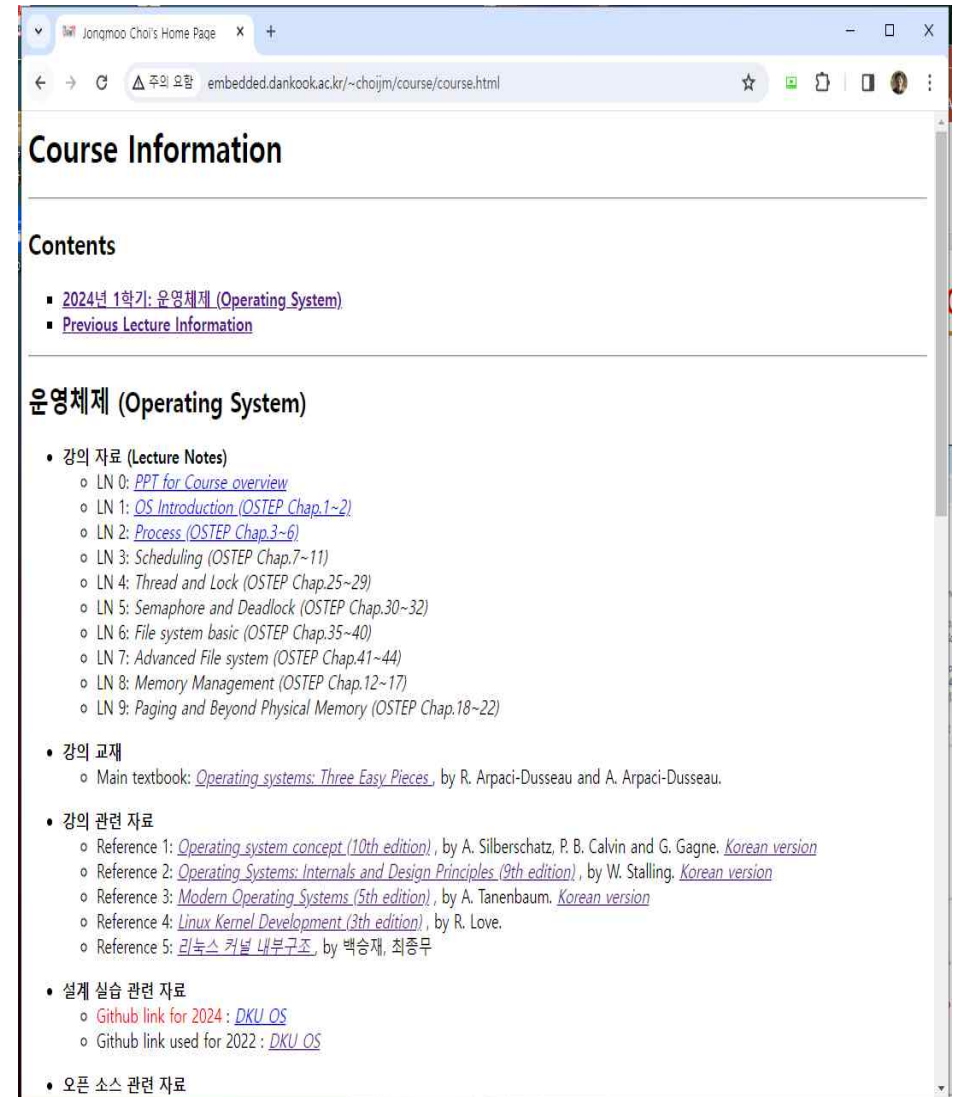
Contents

- Curriculum Vitae
- Publication Lists
- Course Information
- Projects
- Photographs
- Personal Interests

Events

- 시스템 SW 연구실 (임베디드 연구실)에서 24년 1학기 학부 연구생을 모집합니다. 시스템 프로그램, 운영체제, 빅데이터, 차세대 스토리지에 관심있는 학생은 저에게 메일 또는 연구실로 방문해 주시면 됩니다.
- We are proud to announce that our Lab. (DKU Embedded Lab.) have been selected as the SW StarLab by IITP (2021~2028) : [Link](#)
- Our paper 'Sampled Learned Index' was accepted in the ACM Sigmod, 2024. Good job Minguk!!
- Our paper 'Tiered Storage' was accepted in the IEEE Bigcomp, 2024. Good job Charles!!: [Link](#)
- Our paper 'ThanosKV' was accepted in the IEEE Bigcomp, 2024. Good job Zhao!!: [Link](#)
- Our paper 'CVSS' was accepted in the USENIX FAST, 2024. Good job Hojin!!: [Link](#)
- Our paper 'ConfZNS' was accepted in the ACM Systor, 2023. Good job Inho!!: [Link](#)
- Our paper 'ACE' was accepted in the IEEE DSN, 2023. Good job Jeungwon!!: [Link](#)

(home page)



The screenshot shows the "Course Information" page for the Operating System course. The browser address bar indicates the URL is `embedded.dankook.ac.kr/~choijm/course/course.html`. The page has a header "Course Information" and a "Contents" section with two main items: "2024년 1학기: 운영체제 (Operating System)" and "Previous Lecture Information".

운영체제 (Operating System)

- 강의 자료 (Lecture Notes)
 - LN 0: [PPT for Course overview](#)
 - LN 1: [OS Introduction \(OSTEP Chap.1~2\)](#)
 - LN 2: [Process \(OSTEP Chap.3~6\)](#)
 - LN 3: [Scheduling \(OSTEP Chap.7~11\)](#)
 - LN 4: [Thread and Lock \(OSTEP Chap.25~29\)](#)
 - LN 5: [Semaphore and Deadlock \(OSTEP Chap.30~32\)](#)
 - LN 6: [File system basic \(OSTEP Chap.35~40\)](#)
 - LN 7: [Advanced File system \(OSTEP Chap.41~44\)](#)
 - LN 8: [Memory Management \(OSTEP Chap.12~17\)](#)
 - LN 9: [Paging and Beyond Physical Memory \(OSTEP Chap.18~22\)](#)
- 강의 교재
 - Main textbook: [Operating systems: Three Easy Pieces](#), by R. Arpaci-Dusseau and A. Arpaci-Dusseau.
- 강의 관련 자료
 - Reference 1: [Operating system concept \(10th edition\)](#), by A. Silberschatz, P. B. Calvin and G. Gagne. [Korean version](#)
 - Reference 2: [Operating Systems: Internals and Design Principles \(9th edition\)](#), by W. Stalling. [Korean version](#)
 - Reference 3: [Modern Operating Systems \(5th edition\)](#), by A. Tanenbaum. [Korean version](#)
 - Reference 4: [Linux Kernel Development \(3th edition\)](#), by R. Love.
 - Reference 5: [리눅스 커널 내부구조](#), by 백승재, 최종무
- 설계 실습 관련 자료
 - Github link for 2024: [DKU OS](#)
 - Github link used for 2022: [DKU OS](#)
- 오픈 소스 관련 자료

(lecture page)

What is Operating System?

■ Definition (from wikipedia.org)

The screenshot shows the Wikipedia article for 'Operating system'. The main text defines an operating system (OS) as system software that manages computer hardware and software resources, and provides common services for computer programs. It also describes how time-sharing systems schedule tasks for efficient use of the system and how the OS acts as an intermediary between programs and hardware, making system calls and handling interrupts. A diagram on the right illustrates the layers: User, Application, Operating system, and Hardware, with arrows indicating bidirectional communication. Below the diagram, common features are listed: Process management, Interrupts, Memory management, File system, Device drivers, Networking, Security, and Input/output. The article also includes market share data for September 2023, showing Microsoft Windows at 68%, macOS at 20%, and Linux at .66%.

en.wikipedia.org/wiki/Operating_system

Operating system

From Wikipedia, the free encyclopedia

An **operating system (OS)** is system software that manages computer hardware and software resources, and provides common services for computer programs.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, peripherals, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,^{[1][2]} although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

In the personal computer market, as of September 2023, Microsoft Windows holds a dominant market share of around 68%. macOS by Apple Inc. is in second place (20%), and the varieties of Linux, including ChromeOS, are collectively in third place (7%).^[3] In the mobile sector (including smartphones and tablets), as of September 2023, Android's share is 68.92%, followed by Apple's iOS and iPadOS with 30.42%, and other operating systems with .66%.^[4] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems (special-purpose operating systems),^{[5][6]} such as embedded and real-time systems, exist for many applications. Security-focused operating systems also exist. Some operating systems have low system requirements (e.g. light-weight Linux distribution). Others may have higher system requirements.

Operating systems

Common features

- Process management
- Interrupts
- Memory management
- File system
- Device drivers
- Networking
- Security
- Input/output

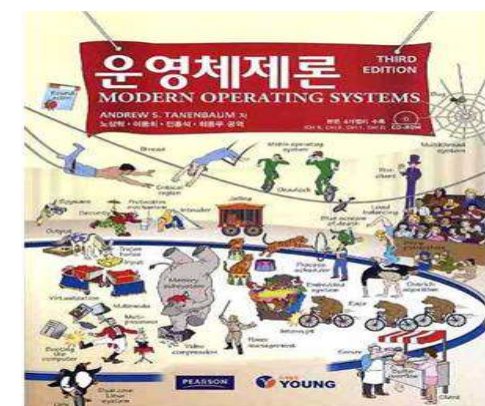
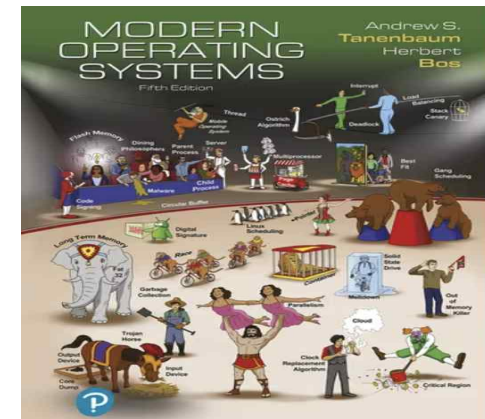
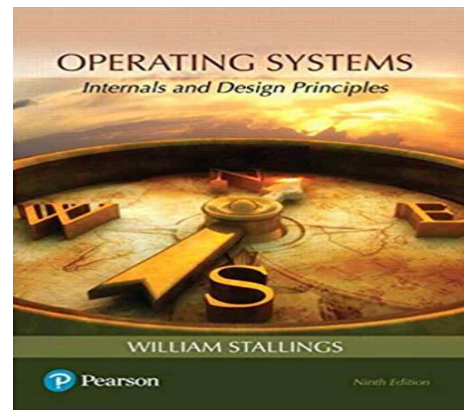
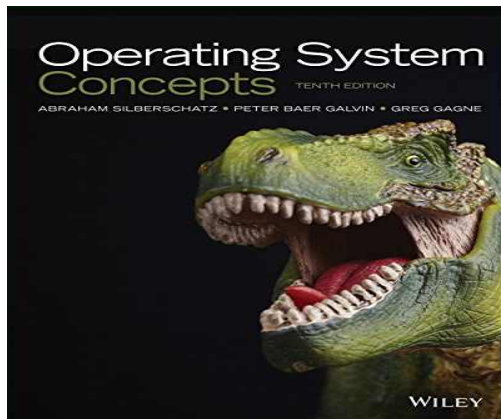
Course Objectives

- Understand the **definition**, role and goal of OS
 - ✓ Resource manager, computing environments, ...
- Know the existing operating systems
 - ✓ UNIX, Windows, Apple OS X, Linux, Android, iOS, WebOS, Mach, ...
- Learn the **internal structure** of OS
 - ✓ Process, Virtual memory, File system, Driver, Protocol, Interrupt, ...
- Comprehend the **policies** and **mechanisms** used by OS
 - ✓ CPU scheduling, Demand paging, LRU, inode, System call, ...
- Grasp the idea of abstraction
 - ✓ Information Hiding, Illusion, Interface, Layered architecture, ...
- **Demonstrate** what we have learned
 - ✓ Lab. project



Traditional Textbook

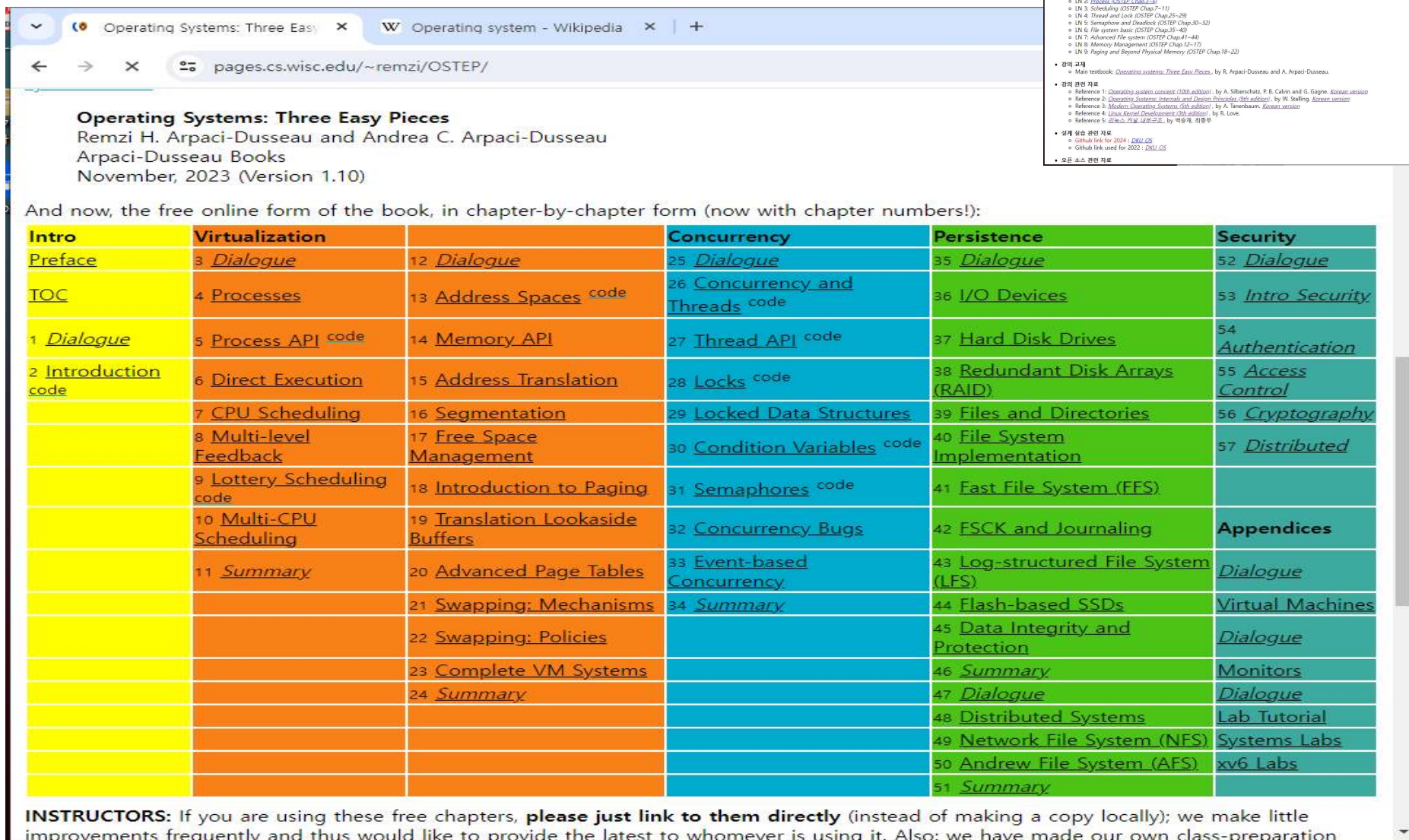
- Three representative textbooks for operating system course
 - ✓ Operating Systems Concepts (10th edition), by A. Silberschatz, P. Galvin and G. Gagne
 - ✓ Operating Systems: Internals and Design Principles (9th edition), by W. Stalling
 - ✓ Modern Operating Systems (5th edition), by A. Tanenbaum and H. Bos



Textbook in this course

■ Remzi's OSTEP (OS Three Easy Pieces)

✓ <http://pages.cs.wisc.edu/~remzi/OSTEP/>

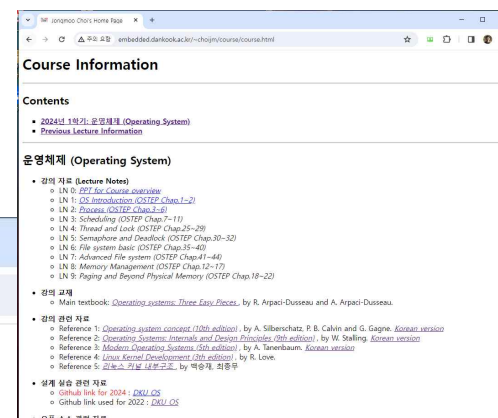


Operating Systems: Three Easy Pieces
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
Arpaci-Dusseau Books
November, 2023 (Version 1.10)

And now, the free online form of the book, in chapter-by-chapter form (now with chapter numbers!):

Intro	Virtualization	Concurrency	Persistence	Security
Preface	3 Dialogue	12 Dialogue	25 Dialogue	35 Dialogue
TOC	4 Processes	13 Address Spaces <small>code</small>	26 Concurrency and Threads <small>code</small>	36 I/O Devices
1 Dialogue	5 Process API <small>code</small>	14 Memory API	27 Thread API <small>code</small>	37 Hard Disk Drives
2 Introduction <small>code</small>	6 Direct Execution	15 Address Translation	28 Locks <small>code</small>	38 Redundant Disk Arrays (RAID)
	7 CPU Scheduling	16 Segmentation	29 Locked Data Structures	39 Files and Directories
	8 Multi-level Feedback	17 Free Space Management	30 Condition Variables <small>code</small>	40 File System Implementation
	9 Lottery Scheduling <small>code</small>	18 Introduction to Paging	31 Semaphores <small>code</small>	41 Fast File System (FFS)
	10 Multi-CPU Scheduling	19 Translation Lookaside Buffers	32 Concurrency Bugs	42 FSCK and Journaling
	11 Summary	20 Advanced Page Tables	33 Event-based Concurrency	43 Log-structured File System (LFS)
		21 Swapping: Mechanisms	34 Summary	44 Flash-based SSDs
		22 Swapping: Policies		45 Data Integrity and Protection
		23 Complete VM Systems		46 Summary
		24 Summary		47 Dialogue
				48 Distributed Systems
				49 Network File System (NFS)
				50 Andrew File System (AFS)
				51 Summary

INSTRUCTORS: If you are using these free chapters, please just link to them directly (instead of making a copy locally); we make little improvements frequently and thus would like to provide the latest to whomever is using it. Also: we have made our own class-preparation



Course Information

Contents

- 2024년 1학기: 운영체제 (Operating System)
- Previous Lecture Information

운영체제 (Operating System)

- 강의 자료 (Lecture Notes)
 - LN 0: [OSTEP Course Overview](#)
 - LN 1: [OS Introduction \(OSTEP Chap.1-2\)](#)
 - LN 2: [Processes \(OSTEP Chap.3-6\)](#)
 - LN 3: [Scheduling \(OSTEP Chap.7-11\)](#)
 - LN 4: [Thread and Lock \(OSTEP Chap.25-29\)](#)
 - LN 5: [Semaphore and Deadlock \(OSTEP Chap.30-32\)](#)
 - LN 6: [File system base \(OSTEP Chap.35-40\)](#)
 - LN 7: [Advanced File system \(OSTEP Chap.41-44\)](#)
 - LN 8: [Memory Management \(OSTEP Chap.15-17\)](#)
 - LN 9: [Paging and Beyond Physical Memory \(OSTEP Chap.18-22\)](#)
- 강의 과제
 - Main textbook: [Operating systems: Three Easy Pieces](#), by R. Arpaci-Dusseau and A. Arpaci-Dusseau.
- 강의 참고 자료
 - Reference 1: [Operating system concept \(10th edition\)](#), by A. Silberschatz, P. B. Galvin and G. Gagne. [Arpaci version](#)
 - Reference 2: [Operating Systems: Internals and Design Principles \(9th edition\)](#), by W. Stallings. [Arpaci version](#)
 - Reference 3: [Modern Operating Systems \(5th edition\)](#), by A. Tanenbaum. [Arpaci version](#)
 - Reference 4: [Linux Kernel Development \(2nd edition\)](#), by R. Love.
 - Reference 5: [리눅스 커널 1.0부터 2.6까지](#), by 박승기, 최중우
- 세계 상용 관련 자료
 - Github link for 2024: [DKU_OS](#)
 - Github link used for 2022: [CSU_OS](#)
- 요즘 스스 관련 자료

Textbook in this course

■ TOC (Table of Contents) of OSTEP

Contents

To Everyone	iii
To Educators	vi
To Students	viii
Acknowledgments	ix
Final Words	xiii
References	xiv
1 A Dialogue on the Book	1
2 Introduction to Operating Systems	3
2.1 Virtualizing The CPU	5
2.2 Virtualizing Memory	7
2.3 Concurrency	9
2.4 Persistence	11
2.5 Design Goals	13
2.6 Some History	14
2.7 Summary	19
References	20
Homework	21
I Virtualization	23
3 A Dialogue on Virtualization	25
4 The Abstraction: The Process	27
4.1 The Abstraction: A Process	28
4.2 Process API	29
4.3 Process Creation: A Little More Detail	30
4.4 Process States	31
4.5 Data Structures	33
4.6 Summary	35
References	37

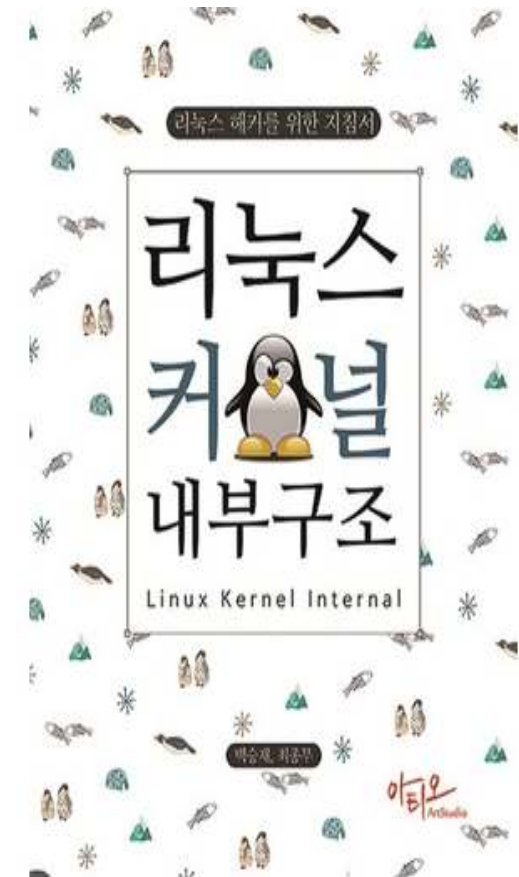
Homework (Simulation)	38
5 Interlude: Process API	41
5.1 The <code>fork()</code> System Call	41
5.2 The <code>wait()</code> System Call	44
5.3 Finally, The <code>exec()</code> System Call	44
5.4 Why? Motivating The API	46
5.5 Process Control And Users	48
5.6 Useful Tools	49
5.7 Summary	50
References	52
Homework (Simulation)	53
Homework (Code)	54
6 Mechanism: Limited Direct Execution	57
6.1 Basic Technique: Limited Direct Execution	57
6.2 Problem #1: Restricted Operations	58
6.3 Problem #2: Switching Between Processes	63
6.4 Worried About Concurrency?	67
6.5 Summary	68
References	71
Homework (Measurement)	72
7 Scheduling: Introduction	73
7.1 Workload Assumptions	73
7.2 Scheduling Metrics	74
7.3 First In, First Out (FIFO)	74
7.4 Shortest Job First (SJF)	76
7.5 Shortest Time-to-Completion First (STCF)	77
7.6 A New Metric: Response Time	78
7.7 Round Robin	79
7.8 Incorporating I/O	81
7.9 No More Oracle	82
7.10 Summary	83
References	84
Homework (Simulation)	85
8 Scheduling: The Multi-Level Feedback Queue	87
8.1 MLFQ: Basic Rules	88
8.2 Attempt #1: How To Change Priority	89
8.3 Attempt #2: The Priority Boost	92
8.4 Attempt #3: Better Accounting	93
8.5 Tuning MLFQ And Other Issues	94
8.6 MLFQ: Summary	96
References	97
Homework (Simulation)	98

9 Scheduling: Proportional Share	99
9.1 Basic Concept: Tickets Represent Your Share	99
9.2 Ticket Mechanisms	101
9.3 Implementation	102
9.4 An Example	103
9.5 How To Assign Tickets?	104
9.6 Stride Scheduling	104
9.7 The Linux Completely Fair Scheduler (CFS)	105
9.8 Summary	110
References	111
Homework (Simulation)	112
10 Multiprocessor Scheduling (Advanced)	113
10.1 Background: Multiprocessor Architecture	114
10.2 Don't Forget Synchronization	116
10.3 One Final Issue: Cache Affinity	117
10.4 Single-Queue Scheduling	118
10.5 Multi-Queue Scheduling	119
10.6 Linux Multiprocessor Schedulers	122
10.7 Summary	122
References	123
Homework (Simulation)	124
11 Summary Dialogue on CPU Virtualization	127
12 A Dialogue on Memory Virtualization	129
13 The Abstraction: Address Spaces	131
13.1 Early Systems	131
13.2 Multiprogramming and Time Sharing	131
13.3 The Address Space	133
13.4 Goals	135
13.5 Summary	136
References	138
Homework (Code)	139
14 Interlude: Memory API	141
14.1 Types of Memory	141
14.2 The <code>malloc()</code> Call	142
14.3 The <code>free()</code> Call	144
14.4 Common Errors	144
14.5 Underlying OS Support	148
14.6 Other Calls	148
14.7 Summary	149
References	150
Homework (Code)	151

Reference

■ Linux Kernel Internals (리눅스 커널 내부 구조)

- ✓ 1장. 리눅스 소개
- ✓ 2장. 리눅스 커널 구조
- ✓ 3장. 태스크 관리
- ✓ 4장. 메모리 관리
- ✓ 5장. 파일시스템과 가상 파일시스템
- ✓ 6장. 인터럽트와 트랩 그리고 시스템 호출
- ✓ 7장. 리눅스 모듈 프로그래밍
- ✓ 8장. 디바이스 드라이버
- ✓ 9장. 네트워킹
- ✓ 10장. 운영체제 관련 실습
- ✓ 부록1. 리눅스와 가상화 그리고 XEN
- ✓ 부록2. MTD와 YAFFS



Teaching Method

■ Mainly Lecturing

- ✓ Discussion (Q&A) during the course is quite important

■ Homework

- ✓ Reading assignment
 - 1 or 2 times
- ✓ Lab. Project (Programming or Analysis)
 - Lab1: scheduling
 - Lab2: concurrency
 - Lab3: file system
 - Lab4: virtual memory



■ Grading

- ✓ Exam(50%) + Lab. Project/Assignment (40%) + Attendance/Discussion (10%) → can be changed later
- ✓ **Absence more than 5 times** or **Mid or Final Exam. score below 20** or **No lab. Project** → F
- ✓ Roughly, 20% students are expected to get the A grade.

Discussion



• Any questions? Feel free to ask at our class or send an email to me: choijm@dankook.ac.kr



Quiz for this Lecture

Quiz

- ✓ 1. What are the differences between Operating System (e.g. MS Windows or Linux) and Application (e.g. MS Word or Chrome)? Explain the difference using the word “mode”.
- ✓ 2. What are three pieces of Operating System?
- ✓ 3. There is a Confucian philosopher, Xunzi, in Chapter 1, “A Dialogue on the Book”, of the OSTEP. Explain what he said.



(Source: Google Image)

Operating Systems: Three Easy Pieces
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
Arpaci-Dusseau Books
November 2023 (Version 1.10)

And now, the free online form of the book, in chapter-by-chapter form (now with chapter numbers!):

Intro	Virtualization	Concurrency	Persistence	Security
Preface	8 Dialogue	12 Dialogue	38 Dialogue	52 Dialogue
TOC	4 Processes	13 Address Spaces code	28 Concurrency and Threads code	38 I/O Devices
1 Dialogue	5 Process API code	14 Memory API	29 Thread API code	37 Hard Disk Drives
2 Introduction code	6 Direct Execution	15 Address Translation	29 Locks code	38 Redundant Disk Arrays (RAID)
	7 CPU Scheduling	16 Segmentation	30 Locked Data Structures	38 Files and Directories
	8 Multi-level Feedback	17 Free Space Management	30 Condition Variables code	38 File System Implementation
	9 Lottery Scheduling code	18 Introduction to Paging	31 Semaphores code	41 Fast File System (FFS)
	10 Multi-CPU Scheduling	19 Translation Lookaside Buffers	32 Concurrency Bugs	42 FSCK and Journaling
	11 Summary	20 Advanced Page Tables	33 Event-based Concurrency	43 Log-structured File System (LFS)
		21 Swapping: Mechanisms	34 Summary	44 Flash-based SSDs
		22 Swapping: Policies		45 Data Integrity and Protection
		23 Complete VM Systems		46 Summary
		24 Summary		47 Dialogue
				48 Distributed Systems
				49 Network File System (NFS)
				50 Andrew File System (AFS)
				51 Summary
				52 Dialogue
				53 Access Control
				56 Cryptography
				57 Distributed
				Appendices
				Monitors
				Dialogue
				Lab Tutorial
				Systems Labs
				xxv Labs

INSTRUCTORS: If you are using these free chapters, please just link to them directly (instead of making a copy locally); we make little improvements frequently and thus would like to provide the latest to whomever is using it. Also, we have made our own class preparation.

2 A DIALOGUE ON THE BOOK

Professor: Excellent query! Well, each person needs to figure this out on their own, of course, but here is what I would do: go to class, to hear the professor introduce the material. Then, at the end of every week, read these notes, to help the ideas sink into your head a bit better. Of course, some time later (that, before the exam!), read the notes again to firm up your knowledge. Of course, your professor will no doubt assign some homeworks and projects, so you should do those; in particular, doing projects where you write real code to solve real problems is the best way to put the ideas within these notes into action. As Confucius said...

Student: Oh, I know! I hear and I forget. I see and I remember. I do and I understand. Or something like that.

Professor: (surprised) How did you know what I was going to say?!

Student: It seemed to follow. Also, I am a big fan of Confucius, and an even bigger fan of Xunzi, who actually is a better source for this quote³.

Professor: (stunned) Well, I think we are going to get along just fine! Just fine indeed.

Student: Professor – just one more question, if I may. What are these dialogues for? I mean, isn't this just supposed to be a book? Why not present the material directly?

Professor: Ah, good question, good question! Well, I think it is sometimes useful to pull yourself outside of a narrative and think a bit; these dialogues are those times. So you and I are going to work together to make sense of all of these pretty complex ideas. Are you up for it?

Student: So we have to think? Well, I'm up for that. I mean, what else do I have to do anyhow? It's not like I have much of a life outside of this book.

Professor: Me neither, sadly. So let's get to work!

³According to <http://www.bareillypopik.com> (on, December 19, 2012, entitled "Tell me and I forget; teach me and I may remember; involve me and I will learn.") Confucian philosopher Xunzi said "Not having heard something is not as good as having heard it; having heard it is not as good as having seen it; having seen it is not as good as knowing it; knowing it is not as good as putting it into practice." Later on, the wisdom got attached to Confucius for some reason. Thanks to Jiao Dong (Kajigas) for telling us!

OPERATING SYSTEMS [VERSION 1.10] WWW.OSTEP.ORG