

Lecture Note 0: Course Introduction

March 4, 2026

Jongmoo Choi

Dept. of Software
Dankook University

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

(본 교재는 2026년도 과학기술정보통신부 및 정보통신기획평가원의 'SW중심대학사업' 지원을 받아 제작 되었습니다.)

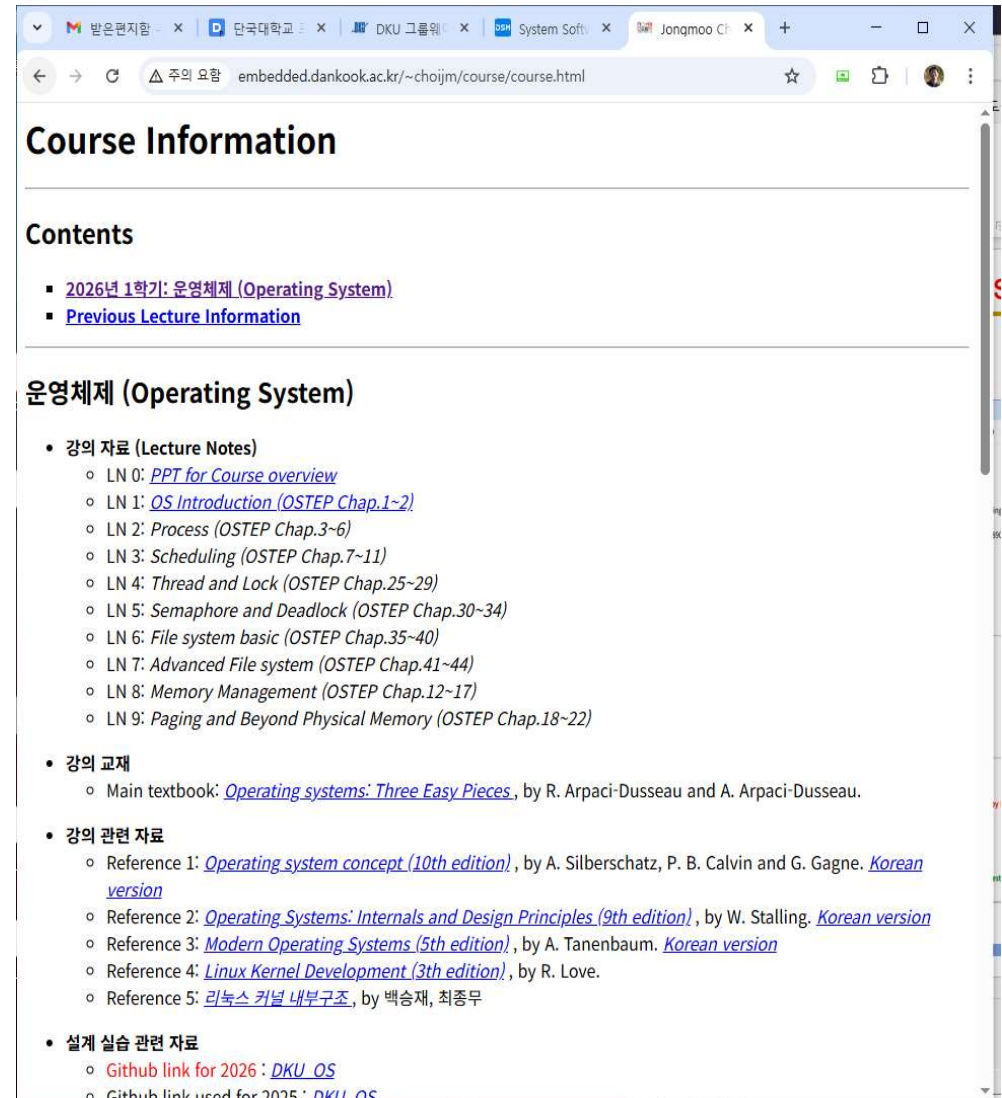
How to access lecture contents?

■ Lecture site



The screenshot shows the home page of the lecture site. At the top, the name 'Jongmoo Choi' is displayed. Below it is a profile picture of a man in a suit. To the right of the photo, his title 'Professor (Ph.D.)' and affiliation 'Dept. of Software (Dept. of Computer Science and Engineering) Dankook University' are listed, along with his address in Gyeonggi-do, Korea. Contact information including email, office location, and phone/fax numbers is provided. A link to his research homepage and Google Scholar profile is also present. The 'Contents' section lists links for Curriculum Vitae, Publication Lists, Course Information, Projects, Photographs, and Personal Interests. The 'Events' section contains several announcements regarding lab selection and paper acceptances with accompanying links.

(home page)



The screenshot shows the 'Course Information' page for the '운영체제 (Operating System)' course. The page title is 'Course Information' and the sub-section is 'Contents'. It lists two main items: '2026년 1학기: 운영체제 (Operating System)' and 'Previous Lecture Information'. The '운영체제 (Operating System)' section is expanded, showing a list of lecture notes (LN 0 to LN 9) with their respective topics and OSTEP chapter references. Below this, it lists the main textbook 'Operating systems: Three Easy Pieces' and several reference materials, including 'Operating system concept (10th edition)', 'Operating Systems: Internals and Design Principles (9th edition)', 'Modern Operating Systems (5th edition)', and 'Linux Kernel Development (3th edition)'. It also provides links for setup and lab-related materials for the years 2026 and 2025.

(lecture page)

What is Operating System?

■ Definition (from wikipedia.org)

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.

As of November 2025, Android is the most popular operating system with a 38% market share, followed by Microsoft Windows at 33%, iOS and iPadOS at 15%, macOS at 4%, and Linux at 1%. Android, iOS, and iPadOS are operating systems for mobile devices such as smartphones, while Windows, macOS, and Linux are for desktop computers.^[3] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems (special-purpose operating systems),^{[4][5]} 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.

Some operating systems require installation or may come pre-installed with purchased computers (OEM-installation), whereas others may run directly from media (i.e. live CD) or flash memory (i.e. a LiveUSB from a USB stick).

Definition and purpose

An operating system is difficult to define,^[6] but has been called "the layer of software that manages a computer's resources for its users and their applications".^[7] Operating systems include the software that is always running, called a kernel—but can include other software as well.^{[6][8]} The two other types of programs that can run on a computer are system programs—which are associated with the operating system, but may not be part of the kernel—and applications, all other software.^[8]

Operating systems

User
Application
Operating system
Hardware

Common features
Process management • Interrupts •
Memory management • File system •
Device drivers • Networking • Security •
Input/output

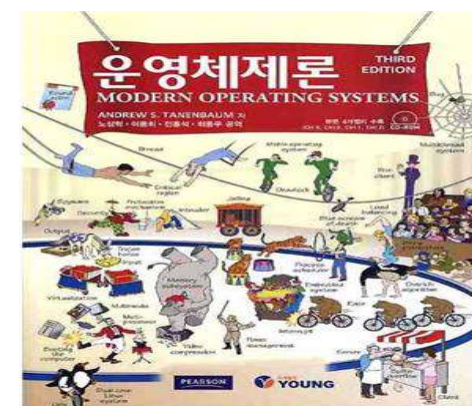
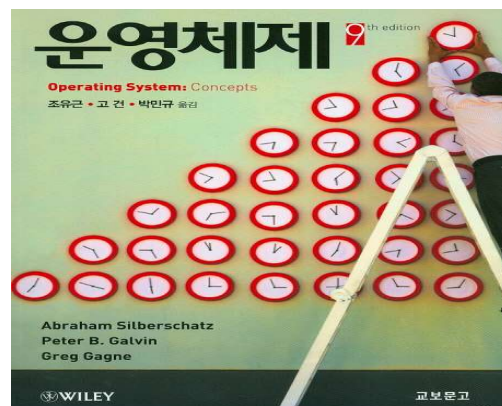
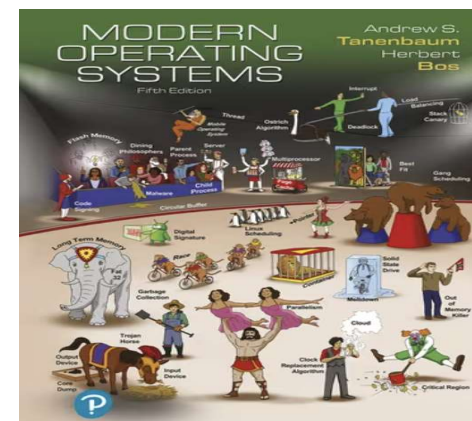
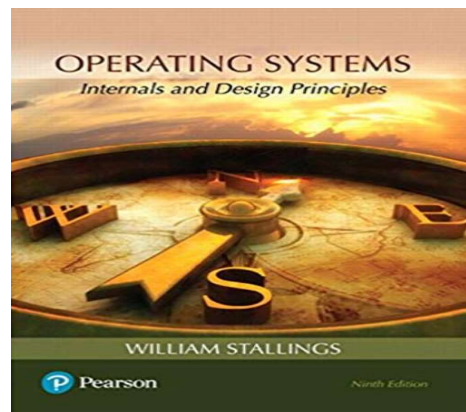
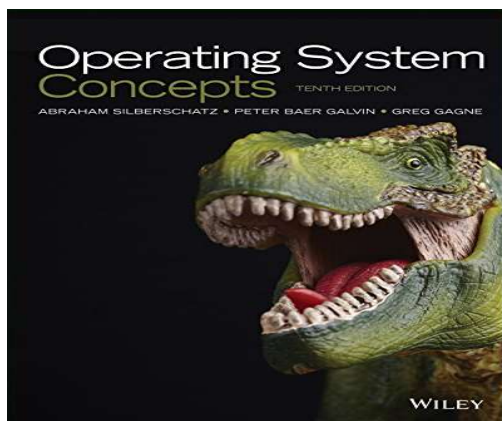
Course Objectives

- 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, by A. Silberschatz, P. Galvin and G. Gagne
 - ✓ Operating Systems: Internals and Design Principles, by W. Stalling
 - ✓ Modern Operating Systems, 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 <i>Dialogue</i>	12 <i>Dialogue</i>	25 <i>Dialogue</i>	35 <i>Dialogue</i>	52 <i>Dialogue</i>
TOC	4 <i>Processes</i>	13 <i>Address Spaces</i> <small>code</small>	26 <i>Concurrency and Threads</i> <small>code</small>	36 <i>I/O Devices</i>	53 <i>Intro Security</i>
1 <i>Dialogue</i>	5 <i>Process API</i> <small>code</small>	14 <i>Memory API</i>	27 <i>Thread API</i> <small>code</small>	37 <i>Hard Disk Drives</i>	54 <i>Authentication</i>
2 <i>Introduction</i> <small>code</small>	6 <i>Direct Execution</i>	15 <i>Address Translation</i>	28 <i>Locks</i> <small>code</small>	38 <i>Redundant Disk Arrays (RAID)</i>	55 <i>Access Control</i>
	7 <i>CPU Scheduling</i>	16 <i>Segmentation</i>	29 <i>Locked Data Structures</i>	39 <i>Files and Directories</i>	56 <i>Cryptography</i>
	8 <i>Multi-level Feedback</i>	17 <i>Free Space Management</i>	30 <i>Condition Variables</i> <small>code</small>	40 <i>File System Implementation</i>	57 <i>Distributed</i>
	9 <i>Lottery Scheduling</i> <small>code</small>	18 <i>Introduction to Paging</i>	31 <i>Semaphores</i> <small>code</small>	41 <i>Fast File System (FFS)</i>	
	10 <i>Multi-CPU Scheduling</i>	19 <i>Translation Lookaside Buffers</i>	32 <i>Concurrency Bugs</i>	42 <i>FSCK and Journaling</i>	Appendices
	11 <i>Summary</i>	20 <i>Advanced Page Tables</i>	33 <i>Event-based Concurrency</i>	43 <i>Log-structured File System (LFS)</i>	<i>Dialogue</i>
		21 <i>Swapping: Mechanisms</i>	34 <i>Summary</i>	44 <i>Flash-based SSDs</i>	<i>Virtual Machines</i>
		22 <i>Swapping: Policies</i>		45 <i>Data Integrity and Protection</i>	<i>Dialogue</i>
		23 <i>Complete VM Systems</i>		46 <i>Summary</i>	<i>Monitors</i>
		24 <i>Summary</i>		47 <i>Dialogue</i>	<i>Dialogue</i>
				48 <i>Distributed Systems</i>	<i>Lab Tutorial</i>
				49 <i>Network File System (NFS)</i>	<i>Systems Labs</i>
				50 <i>Andrew File System (AFS)</i>	<i>xv6 Labs</i>
				51 <i>Summary</i>	

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

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

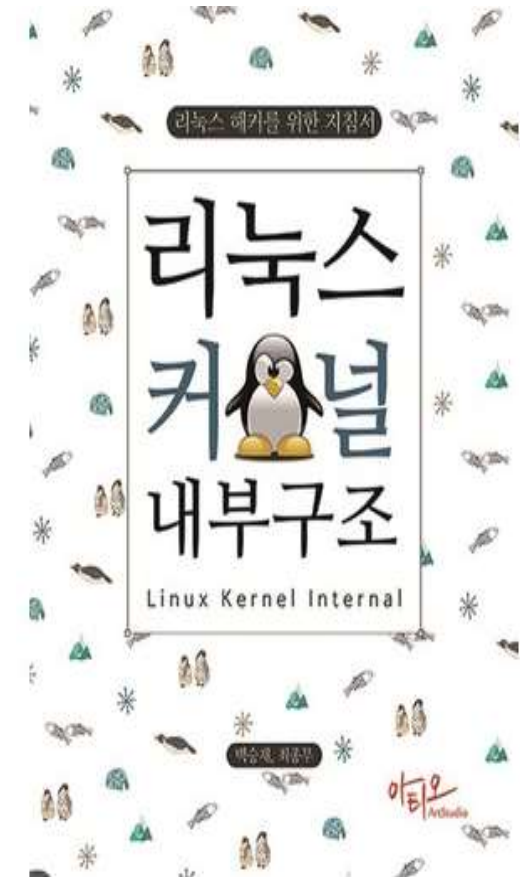
Homework (Simulation)	38
5 Interlude: Process API	41
5.1 The <code>fork()</code> System Call	41
5.2 The <code>wait()</code> System Call	44
5.3 Finally, The <code>exec()</code> System Call	44
5.4 Why? Motivating The API	46
5.5 Process Control And Users	48
5.6 Useful Tools	49
5.7 Summary	50
References	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

- ✓ Mid. Exam (30%) + Final Exam (30%) + Lab. Project/Assignment (30%) + Attendance/Discussion (10%) → can be changed later
- ✓ **Absence more than 5 times** or **Mid and Final Exam. score below 20** or **No lab. Project** → F
- ✓ Roughly, 25% 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	1 Dialogue	12 Dialogue	35 Dialogue	52 Dialogue
TOC	4 Processes	13 Address Spaces code	36 Concurrency and Threads code	53 Intro Security
1 Dialogue	5 Process API code	14 Memory API	37 Thread API code	54 Authentication
2 Introduction code	6 Direct Execution	15 Address Translation	38 Locks code	55 Redundant Disk Arrays (RAID)
	7 CPU Scheduling	16 Segmentation	39 Locked Data Structures	56 Files and Directories
	8 Multi-level Feedback	17 Free Space Management	40 Condition Variables code	57 File System Implementation
	9 Lottery Scheduling code	18 Introduction to Paging	41 Semaphores code	41 Fast File System (FFS)
	10 Multi-CPU Scheduling	19 Translation Lookaside Buffers	42 Concurrency Bugs	42 FSCK and Journaling
	11 Summary	20 Advanced Page Tables	43 Event-based Concurrency	43 Log-structured File System (LFS)
		21 Swapping: Mechanisms	44 Flash-based SSDs	44 Flash-based SSDs
		22 Swapping: Policies	45 Data Integrity and Protection	45 Data Integrity and Protection
		23 Complete VM Systems	46 Summary	46 Summary
		24 Summary	47 Dialogue	47 Dialogue
			48 Distributed Systems	48 Distributed Systems
			49 Network File System (NFS)	49 Network File System (NFS)
			50 Andrew File System (AFS)	50 Andrew File System (AFS)
			51 Summary	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

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 (hint: 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 quote¹.

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!

¹According to <http://www.barypopik.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 (Rutgers) for telling us!

사사

- 본 교재는 2026년도 과학기술정보통신부 및 정보통신기획평가원의 ‘SW중심대학사업’ 지원을 받아 제작 되었습니다.
- 본 결과물의 내용을 전재할 수 없으며, 인용(재사용)할 때에는 반드시 과학기술정보통신부와 정보통신기획평가원이 지원한 ‘SW중심대학’의 결과물이라는 출처를 밝혀야 합니다.

IITP 정보통신기획평가원
디지털인재양성단 SW인재팀

SW중심대학이 무엇인가요?

SW중심대학은
대학교육을 SW중심으로 혁신함으로써,
SW전문인력을 양성하고
학생·기업·사회의 SW경쟁력을 강화해
진정한 SW가치 확산을 실현하는 대학을 말합니다.

