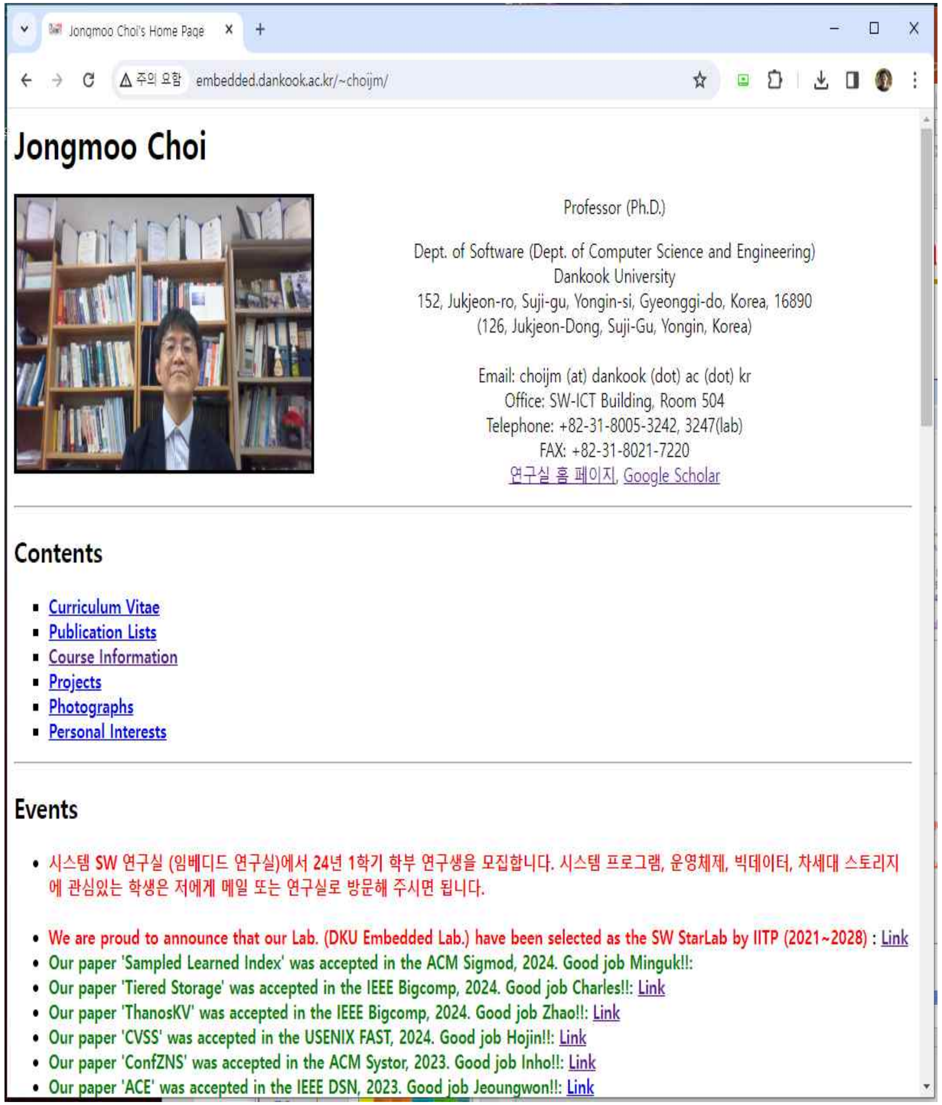
Lecture Note 0: Course Introduction

March 4, 2024
Jongmoo Choi

Dept. of Software
Dankook University
http://embedded.dankook.ac.kr/~choijm

How to access lecture contents?

- Lecture site



Jongmoo Choi

Professor (Ph.D.)

Dept. of Software (Dept. of Computer Science and Engineering)
Dankook University
152, Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, Korea, 16890
(126, Jukjeon-Dong, Suji-Gu, Yongin, Korea)

Email: choijm (at) dankook (dot) ac (dot) kr
Office: SW-ICT Building, Room 504
Telephone: +82-31-8005-3242, 3247(lab)
FAX: +82-31-8021-7220
연구실 홈 페이지, Google Scholar

Contents

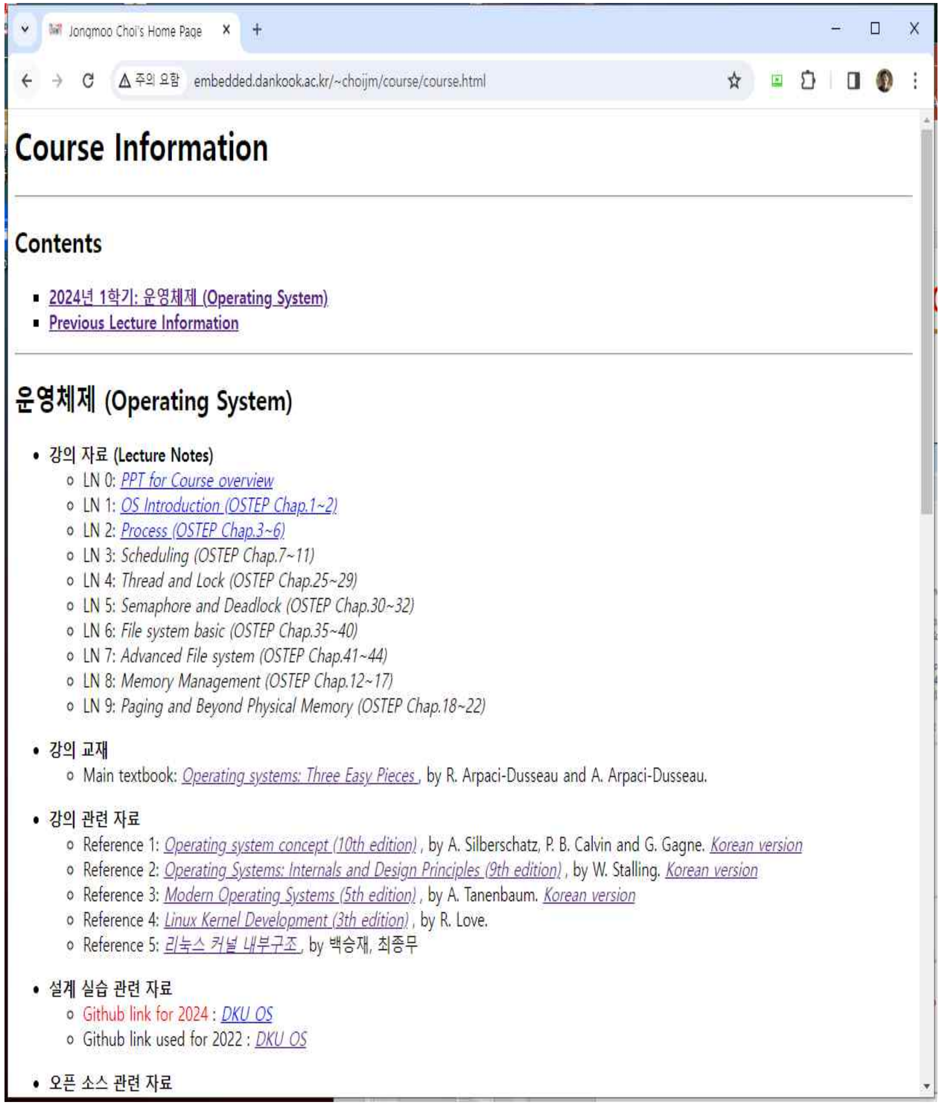
- Curriculum Vitae
- Publication Lists
- Course Information
- Projects
- Photographs
- Personal Interests

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 Jeoungwon!!: Link

(home page)



Course Information

Contents

- 2024년 1학기: 운영체제 (Operating System)
- 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

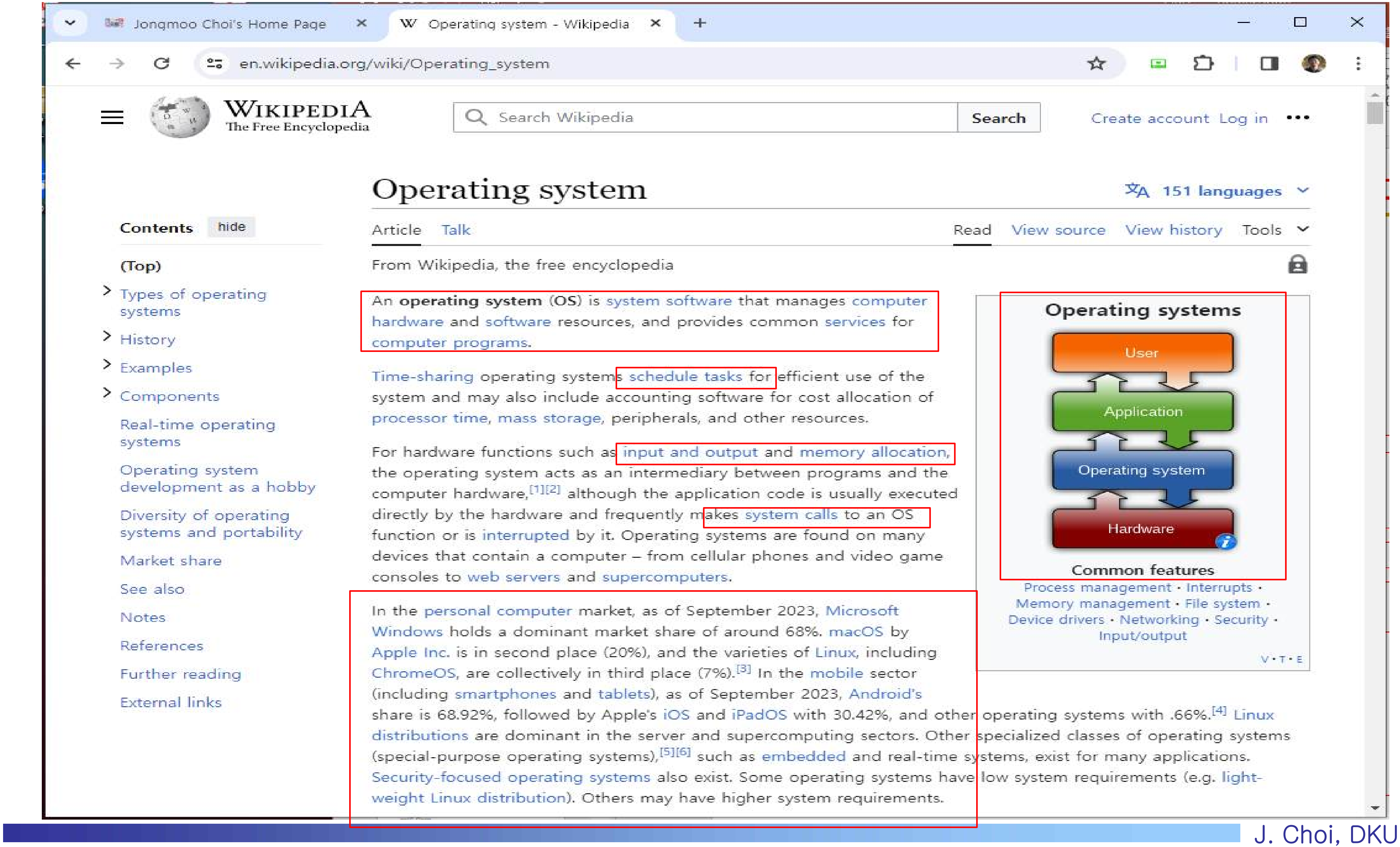
- 설계 실습 관련 자료
  - Github link for 2024 : *DKU OS*
  - Github link used for 2022 : *DKU OS*

- 오픈 소스 관련 자료

(lecture page)

What is Operating System?

- Definition (from wikipedia.org)



Operating system

Contents hide

(Top)
Types of operating systems
History
Examples
Components
Real-time operating systems
Operating system development as a hobby
Diversity of operating systems and portability
Market share
See also
Notes
References
Further reading
External links

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.

Common features

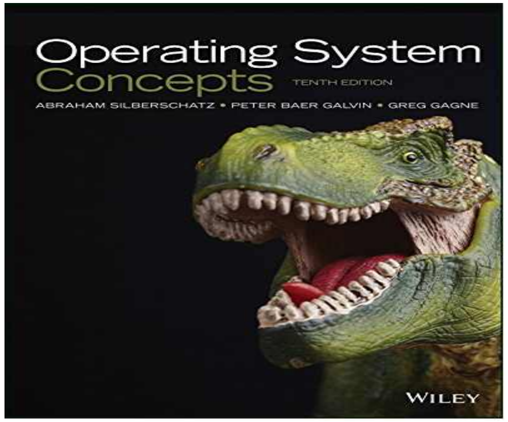
Process management · Interrupts · Memory management · File system · Device drivers · Networking · Security · Input/output

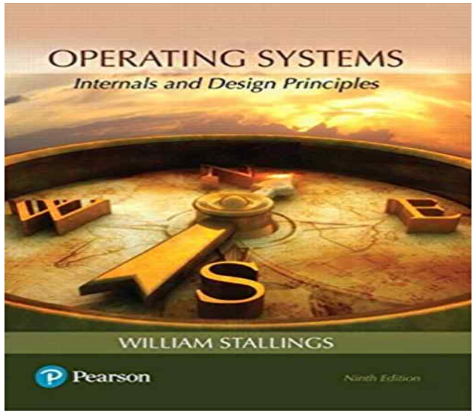
Course Objectives

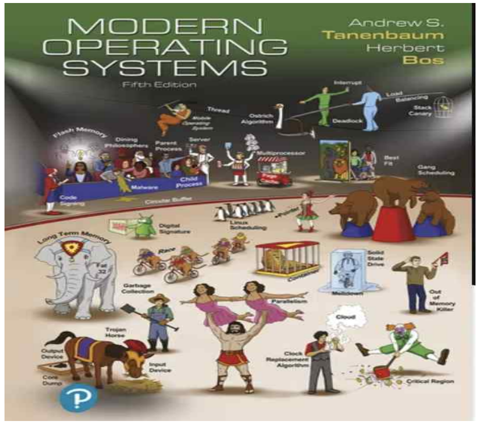
- Understand the definition, role and goal of OS
  - ✓ Resource manager, computing environments, …
- Know the existing operating systems
  - ✓ UNIX, Windows, Apple OS X, Linux, Android, iOS, WebOS, Mach, …
- Learn the internal structure of OS
  - ✓ Process, Virtual memory, File system, Driver, Protocol, 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

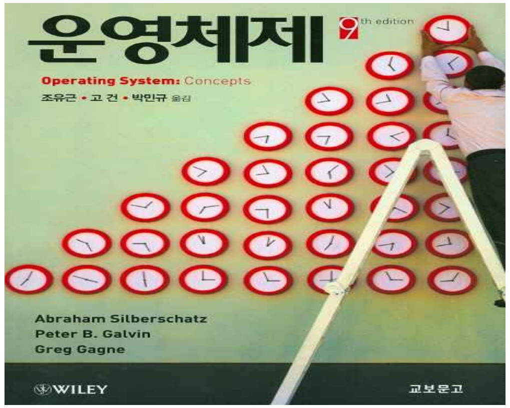
Traditional Textbook

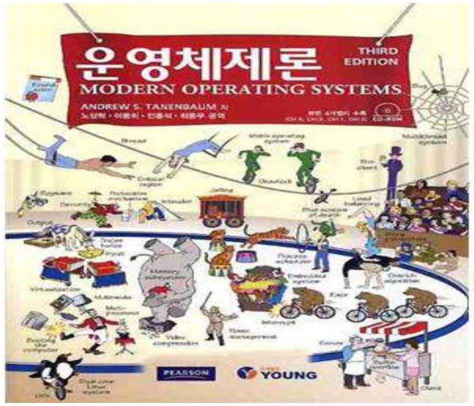
- Three representative textbooks for operating system course
  - ✓ Operating Systems Concepts (10th edition), by A. Silberschatz, P. Galvin and G. Gagne
  - ✓ Operating Systems: Internals and Design Principles (9th edition), by W. Stalling
  - ✓ Modern Operating Systems (5th edition), by A. Tanenbaum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


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



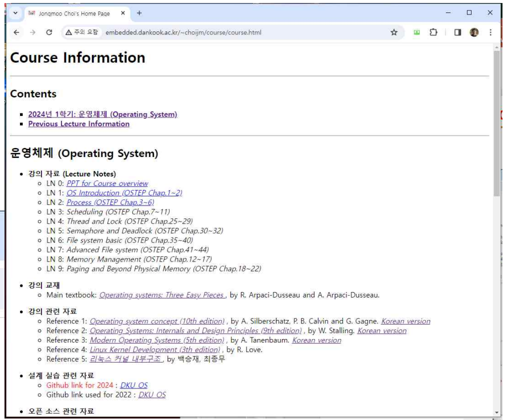






Textbook in this course

- Remzi's OSTEP (OS Three Easy Pieces)
  - ✓ http://pages.cs.wisc.edu/~remzi/OSTEP/



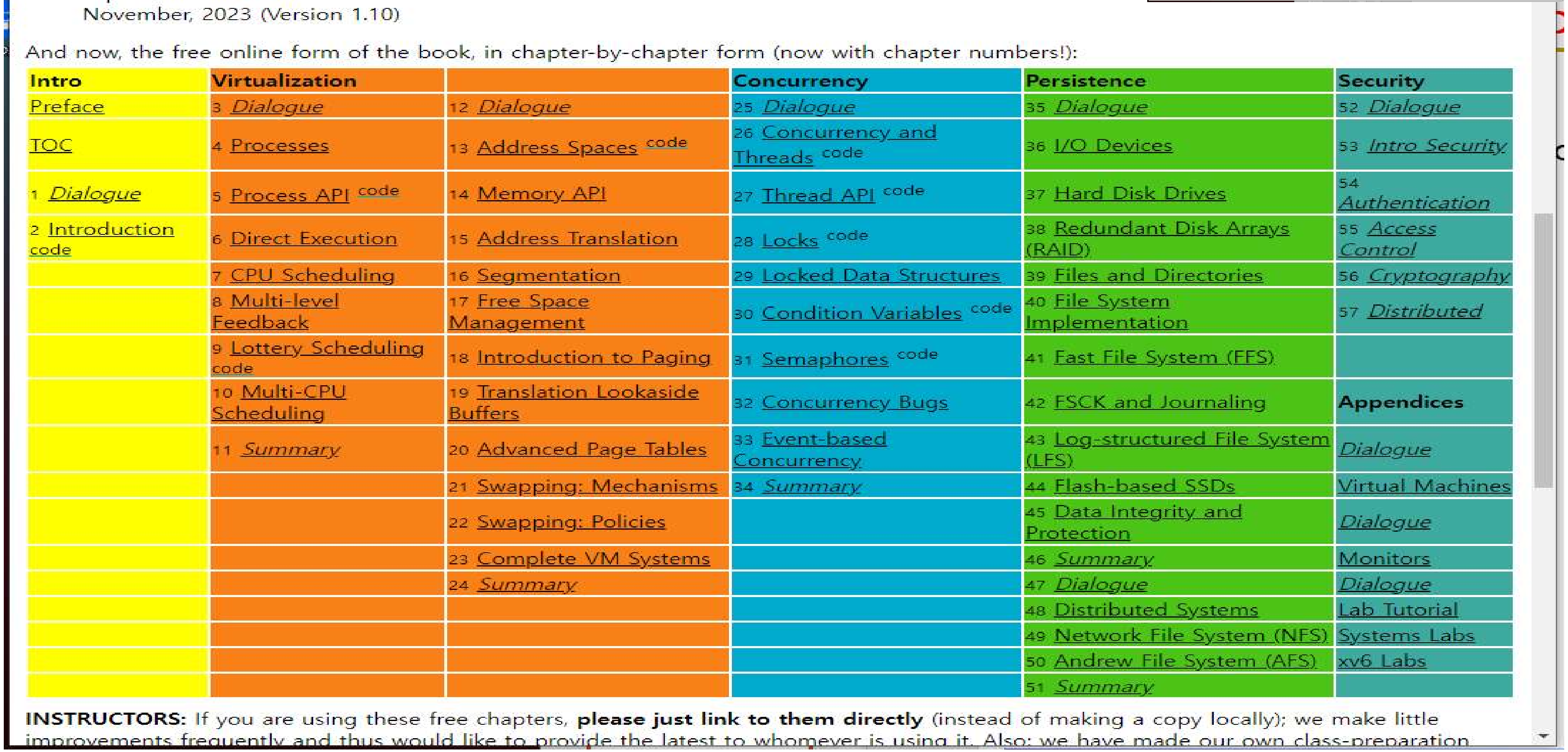
Course Information

Contents

- 2024년 1학기: 운영체제 (Operating System)
- 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


Operating Systems: Three Eas
Operating system - Wikipedia
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)

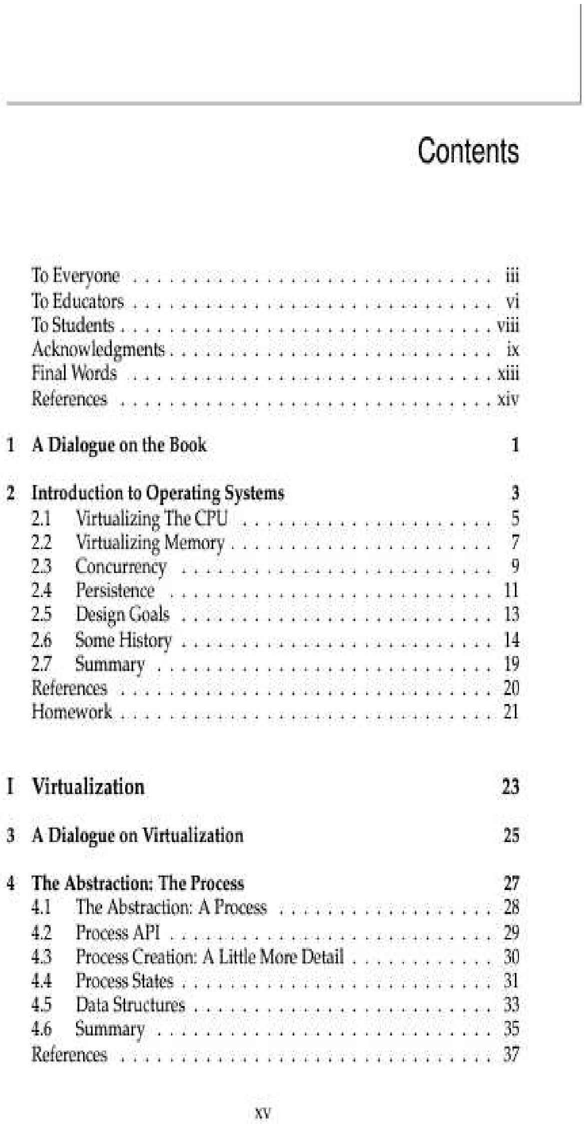
And now, the free online form of the book, in chapter-by-chapter form (now with chapter numbers!):

Intro	Virtualization		Concurrency	Persistence	Security
Preface	3 *Dialogue*	12 *Dialogue*	25 *Dialogue*	35 *Dialogue*	52 *Dialogue*
TOC	4 Processes	13 Address Spaces code	26 Concurrency and Threads code	36 I/O Devices	53 *Intro Security*
1 *Dialogue*	5 Process API code	14 Memory API	27 Thread API code	37 Hard Disk Drives	54 *Authentication*
2 Introduction code	6 Direct Execution	15 Address Translation	28 Locks code	38 Redundant Disk Arrays (RAID)	55 *Access Control*
	7 CPU Scheduling	16 Segmentation	29 Locked Data Structures	39 Files and Directories	56 *Cryptography*
	8 Multi-level Feedback	17 Free Space Management	30 Condition Variables code	40 File System Implementation	57 *Distributed*
	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	**Appendices**
	11 *Summary*	20 Advanced Page Tables	33 Event-based Concurrency	43 Log-structured File System (LFS)	*Dialogue*
		21 Swapping: Mechanisms	34 *Summary*	44 Flash-based SSDs	Virtual Machines
		22 Swapping: Policies		45 Data Integrity and Protection	*Dialogue*
		23 Complete VM Systems		46 *Summary*	Monitors
		24 *Summary*		47 *Dialogue*	*Dialogue*
				48 Distributed Systems	Lab Tutorial
				49 Network File System (NFS)	Systems Labs
				50 Andrew File System (AFS)	xv6 Labs
				51 *Summary*	

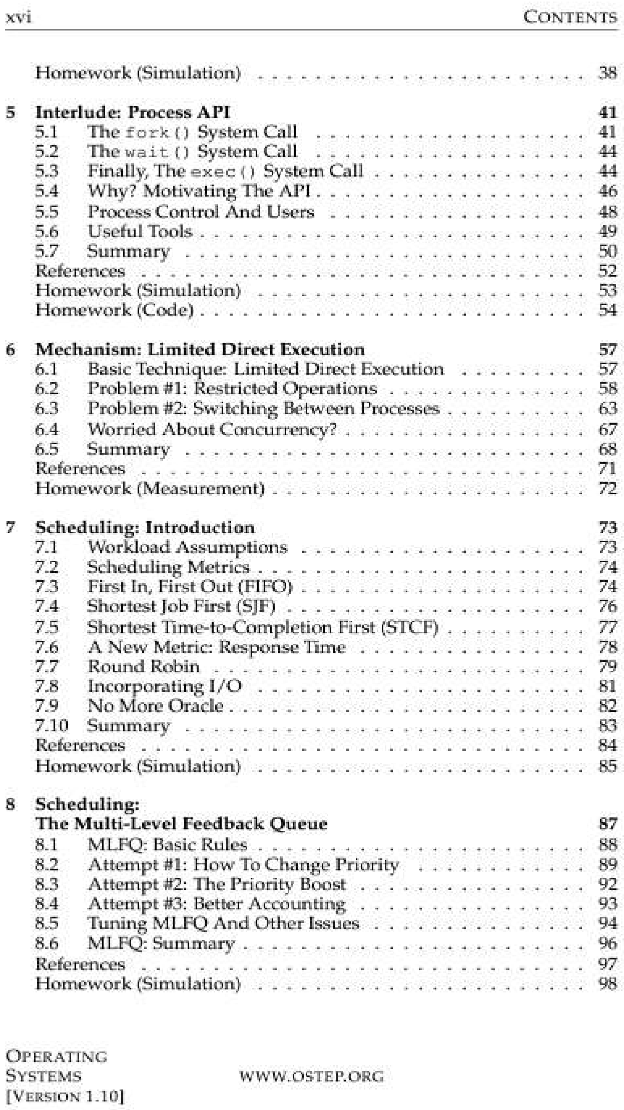
INSTRUCTORS: If you are using these free chapters, **please just link to them directly** (instead of making a copy locally); we make little improvements frequently and thus would like to provide the latest to whomever is using it. Also: we have made our own class-preparation

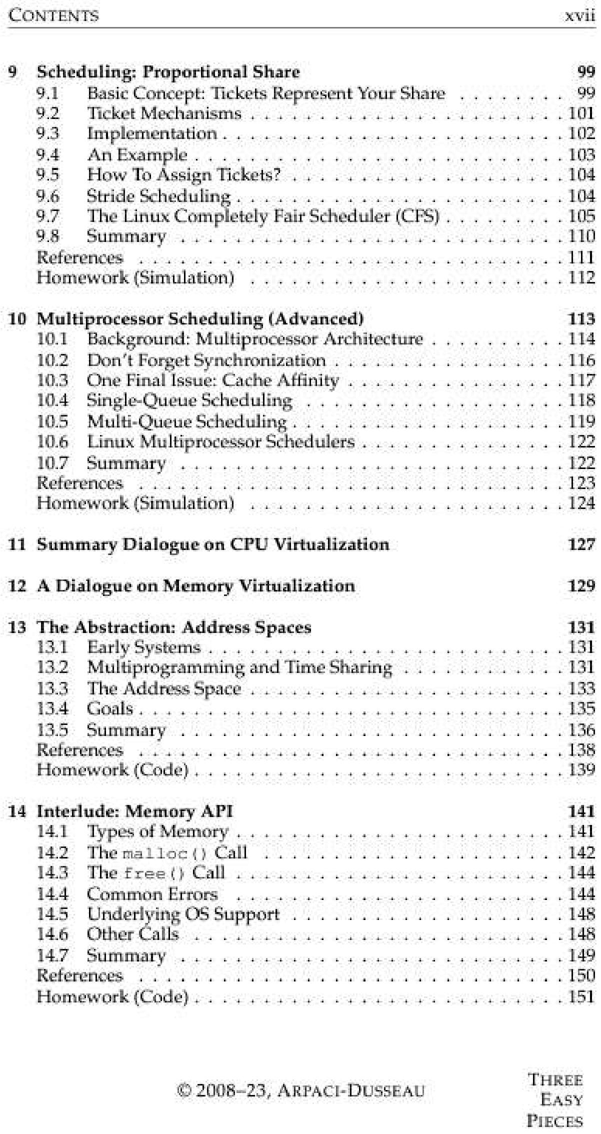
Textbook in this course

- TOC (Table of Contents) of OSTEP



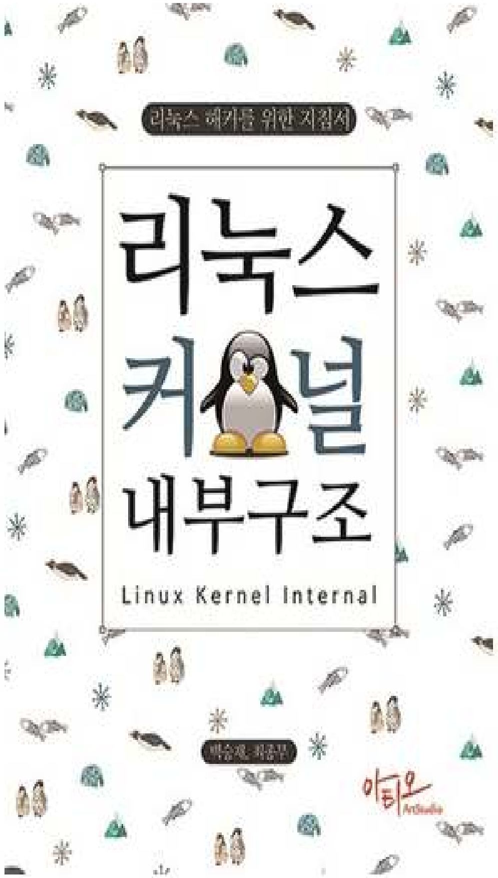
Contents





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.

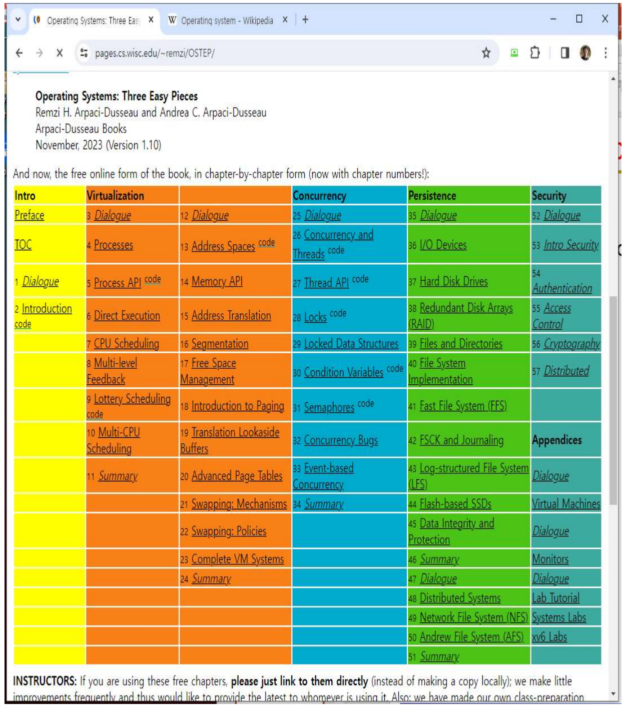
Discussion

☛ 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 Dialog on the Book", of the OSTEP. Explain what he said.

(Source: Google Image)

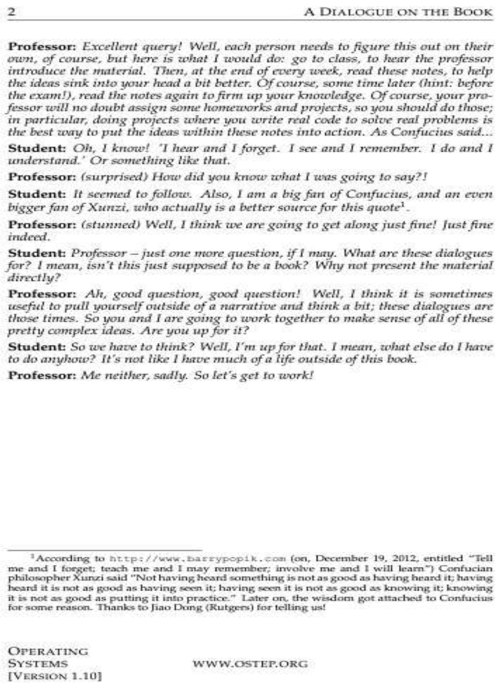

Operating Systems: Three Eas
Operating system - Wikipedia
pages.cs.wisc.edu/~remzi/OSTEP/

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*	52 *Dialogue*
TOC	4 Processes	13 Address Spaces code	26 Concurrency and Threads code	36 I/O Devices	53 *Intro Security*
1 *Dialogue*	5 Process API code	14 Memory API	27 Thread API code	37 Hard Disk Drives	54 *Authentication*
2 Introduction code	6 Direct Execution	15 Address Translation	28 Locks code	38 Redundant Disk Arrays (RAID)	55 *Access Control*
	7 CPU Scheduling	16 Segmentation	29 Locked Data Structures	39 Files and Directories	56 *Cryptography*
	8 Multi-level Feedback	17 Free Space Management	30 Condition Variables code	40 File System Implementation	57 *Distributed*
	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	**Appendices**
	11 *Summary*	20 Advanced Page Tables	33 Event-based Concurrency	43 Log-structured File System (LFS)	*Dialogue*
		21 Swapping: Mechanisms	34 *Summary*	44 Flash-based SSDs	Virtual Machines
		22 Swapping: Policies		45 Data Integrity and Protection	*Dialogue*
		23 Complete VM Systems		46 *Summary*	Monitors
		24 *Summary*		47 *Dialogue*	*Dialogue*
				48 Distributed Systems	Lab Tutorial
				49 Network File System (NFS)	Systems Labs
				50 Andrew File System (AFS)	xv6 Labs
				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 quote*[1]*.*

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!*

[1] According to `http://www.barrypopik.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!

OPERATING SYSTEMS [VERSION 1.10] WWW.OSTEP.ORG