

서문

모든 독자에게

이 책에 온 것을 환영한다! 우리가 이 책을 즐거운 마음으로 집필했듯이 독자들도 이 책을 즐겁게 읽기 바란다. 이 책은 “운영체제: 아주 쉬운 세 가지 이야기”로서 제목은 현존하는 가장 훌륭한 강의 노트 중에 하나인 Richard Feynman의 물리학 강의 노트에 대한 존경의 표시로 명명되었다 [Fey96]. 이 저명한 물리학자가 이룩한 높은 수준에는 미치지 못하더라도, 이 책이 운영체제에 대한 이해, 그리고 더 넓게 시스템에 대한 이해를 구하는 데 도움이 되리라 믿는다.

아주 쉬운 세 가지 이야기란 이 책의 구성에 근간이 되는 세 가지 주요 개념, 가상화(virtualization), 병행성(concurrency), 및 영속성(persistence)을 말한다. 이 세 가지 개념을 통해 운영체제의 중요 기능에 대해 논의한다. 이러한 과정 중에서 또한 재미를 느끼길 바란다. 새로운 것을 배우는 것은 재미있지 않은가? 적어도 재미있어야 한다고 생각한다.

이 주요 개념은 여러 장에 걸쳐 설명되고, 대부분의 장에서 먼저 특정 문제를 제시하고 해결책을 제시한다. 각 장의 길이는 짧으며 가능한 한 아이디어가 처음 소개되었던 원저를 참조하려고 노력하였다. 이 책을 쓴 목적 중의 하나는 가능한 과거에서 어떻게 발전해 왔는가를 명확하게 하는 것이다. 운영체제의 과거, 현재, 그리고 미래의 모습을 더 분명하게 이해할 수 있을 것이라고 생각하기 때문이다. 이 경우 소시지가 어떻게 만들어지느냐를 아는 것이 소시지가 어디에 좋으느냐를 이해하는 것만큼 중요하다¹.

책 전반에 걸쳐 사용할 몇 가지 장치들을 이 자리를 빌려 소개한다. 첫 번째는 **핵심 질문(crux)**이다. 문제의 해결에 앞서 가장 중요한 쟁점을 명확히 하고 넘어가야 한다. 이러한 문제의 **핵심(crux of the problem)**을 본문 중에 명문화한 후에 소개하는 기법, 알고리즘 및 아이디어를 통하여 답할 수 있길 바란다.

1) 힌트: 아니면 당신이 채식주의자일 경우 멀리할 수 있다.

시스템의 동작 순서를 시간순에 따라 나타내었다. 이러한 **타임라인(timeline)**은 시스템의 동작을 이해하기 위한 핵심 요소이다. 예를 들어 페이지 폴트가 발생할 때 프로세스의 동작을 알고 있다면 가상 메모리의 동작을 제대로 이해하기 위한 올바른 과정에 있다고 할 수 있다. 저널링 파일 시스템이 블럭을 디스크에 쓸 때 무슨 일이 벌어지는지 이해한다면, 저장 장치 완전정복을 위한 첫 걸음을 내딛었다고 할 수 있다.

본문에 많은 **여담(aside)**과 **팁(tip)**을 추가하여 구성을 다채롭게 하였다. 여담을 통하여 본문과 관련된 덜 핵심적인 사항들에 대해 논의하려고 한다. 팁은 시스템에 적용될 수 있는 일반적인 교훈을 담으려고 했다. 색인에는 모든 팁과 여담뿐 아니라 모든 핵심 질문을 나열하여 쉽게 찾아볼 수 있게 했다.

책 전반에 걸쳐 몇몇 주제를 다른 관점에서 제공하기 위한 방법으로 가장 오래된 교수법 중 한 가지인 **대화법(dialogue)**을 사용하였다. 이 대화법은 주요 개념을 소개하기 위한 방식으로 사용되었고 때로는 내용을 검토하는 용도로도 사용된다. 또한, 책을 더 재미있게 만들 수 있었다. 대화법을 유용하게 혹은 재미있게 느끼느냐는 별개의 문제이다.

각 주요 절을 시작할 때 운영체제가 제공하는 **개념(abstraction)**을 먼저 제시한 후, 이후의 장에서 개념을 실현하기 위해 필요한 기법, 정책 및 다른 지원 사항들에 대해서 논의하였다. 추상화는 컴퓨터 과학 모든 분야의 근간이기 때문에 운영체제에서도 핵심적이라는 사실은 전혀 놀라운 일이 아니다.

모든 장에 걸쳐 가능한 한 **의사코드(pseudocode)**가 아니라 **실제 코드(real code)**를 사용하려고 노력했다. 사용된 거의 모든 예는 직접 입력하여 실행해 볼 수 있을 것이다. 시스템에서 실제 코드를 실행시켜 보는 것은 운영체제를 배우기 위한 가장 좋은 방법이기 때문에 할 수 있는 한 실행해 보기 바란다.

교재의 다양한 부분에서 현재 논의 중인 내용을 이해하고 있다는 것을 확인할 수 있도록 여러 **과제(homework)**가 있다. 이 중 많은 과제들이 운영체제의 여러 구성 요소에 관한 작은 **모의 실험(simulation)**이다. 과제를 다운로드하여 스스로를 실험하기 위해 실행시켜야 한다. 과제에서 제공하는 시뮬레이터는 난수 발생기를 사용하여 이론적으로 무한개의 문제 집합을 만들어 낼 수 있다. 시뮬레이터는 문제에 대한 답을 출력할 수도 있다. 따라서 운영체제를 충분히 이해할 때까지 계속 반복 실험해 볼 수 있다.

이 책의 가장 중요한 부록은 **프로젝트**로서 이를 통하여 직접 설계, 구현 및 검증할 수 있고 실제 시스템의 동작을 배울 수 있다. 앞에서 언급한 코드 예제 및 모든 프로젝트는 **C 언어**로 작성되어 있다 [KR]. C 언어는 거의 대부분의 운영체제 구현에 사용되고 있기 때문에, 당신의 도구 상자에 추가될 가치가 있는 간단하고 강력한 언어이다. 두 가지 유형의 프로젝트가 제공되며 이에 대해서는 온라인으로 제공되는 부록을 참조하기 바란다. 첫 번째 유형은 **시스템 프로그래밍** 프로젝트이다. 이 프로젝트는 C 언어와 UNIX의 사용이 처음이고 저수준의 C 언어 프로그래밍 기법을 배우려는 사람들에게 유용하다. 두 번째 유형은 실제 운영체제 커널을 기반으로 하는 MIT에서 개발한 xv6 [Cox+]으로 제공된다. 이 프로젝트는 C 언어 프로그래밍 경험이 있고 운영체제 내부에서 직접 작업하기 원하는 학생에게 유용하다. University of Wisconsin의 경우 수업을 시스템 프로그래밍, xv6 프로그래밍, 또는 혼합형으로 세 가지 다른 방식으로 운영한다.

교강사에게

이 교재를 사용하길 원하는 강사나 교수라면 자유롭게 사용하기 바란다. 교재는 무료이며 다음 웹 페이지에서 온라인으로 얻을 수 있다.

<http://www.ostep.org>

lulu.com에서 인쇄본을 구매할 수 있다. 위에 언급한 웹 페이지를 접속해 보자. 이 교재의 현재 참고 문헌 정보는 다음과 같다.

Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

Arpaci-Dusseau Books

March, 2015 (Version 0.90)

<http://www.ostep.org>

15주 강의 용으로 구성되어 있으며 대부분의 주제를 적절한 수준까지 설명할 수 있을 것이다. 10주로 압축해서 강의하려면 각 주제마다 일부 상세 설명은 포기해야 할 것이다. 가상 머신 모니터에 관한 몇몇 장은 우리가 학기 중에 몰아서 강의했던 것처럼 가상화 절의 마지막이나 학기 마지막에 하는 것이 좋다고 생각한다.

이 책의 한 가지 약간 특이한 점은 대부분의 운영체제 책들이 처음에 설명하는 병행성 주제는 CPU와 메모리 가상화 후에 나온다는 것이다. 거의 15년 동안 운영체제를 강의해 온 경험에 비추어 보면 주소 공간, 프로세스, 그리고 문맥 교환을 이해하지 못하면 병행성이나 병행성 문제를 해결해야 하는 이유를 이해하는 데 힘들어 한다. 먼저 주요 개념들을 이해하고 나면 스레드(thread) 개념과 관련 문제들을 소개하는 것이 쉬워지거나 적어도 더 쉬워진다.

가능한 한 칠판(또는 화이트보드)을 사용하여 강의한다. 개념 설명 위주의 강의에는 주요 개념과 예제를 염두에 두고 수업에 들어가 보드를 사용하여 가르친다. 강의 내용을 토대로 구체적인 문제를 해결할 수 있도록 수업자료를 제공한다. 실습 시간에는 랩톱을 빔프로젝터에 연결하여 실제 코드를 보면서 설명한다. 이런 식의 강의는 병행성에 관한 강의뿐만 아니라 수강생들이 진행하는 프로젝트에 연관된 코드를 보여야 하는 강의에 유용하다. 우리는 수업 시간에 슬라이드를 잘 사용하지 않지만 그런 유형의 강의를 선호하는 사람들을 위해 슬라이드를 만들었다.

만일 당신이 이러한 자료의 복사본을 원한다면 우리에게 이메일을 보내주기 바란다. 우리는 이미 세계의 여러 사람들과 자료를 공유해 오고 있다.

마지막 당부 하나: 만일 온라인으로 제공되는 이 교재를 활용한다면 복사본을 만들지 말고 이 자료로 링크를 걸어주기 바란다. 이를 통하여 사용량을 확인할 수 있으며 학생들이 최신의 가장 좋은 버전의 교재를 이용할 수 있게 한다 (지난 2년 동안 백만 건 이상이 다운로드 되었다!).

학생들에게

이 책으로 공부하는 학생들에게 감사의 말을 전한다. 저자들은 운영체제를 학습하는 당신에게 도움이 될 만한 자료를 제공할 수 있게 되어 영광스럽게 생각한다. 저자들은 학부 시절 몇몇 책에 대해 좋은 기억을 가지고 있으며(예를 들면, Patterson과 Hennessy [PH90]의 컴퓨터 구조에 관한 고전), 이 책이 당신에게도 그런 좋은 기억을 주기를 희망한다.

이미 알고 있을지도 모르지만 이 책은 무료이며 온라인으로 얻을 수 있다². 이렇게 하는 한 가지 주요 이유는 일반적으로 교재가 비싸다는 것이다. 우리는 이 책이 출생지나 교재를 구매할 수 있는 경제적 형편과 상관없이, 교육 받고자 하는 사람들에게 도움을 주는 무료 교재를 제공한다는 새로운 물결의 선두 주자가 되길 희망한다. 목적을 달성하지 못하더라도, 또 하나의 무료 책이 되겠지만 아무것도 하지 않는 것보다는 낫다.

이 책에서는 가능한 한 운영체제 분야를 발전시켜 온 훌륭한 논문과 사람들의 원전에 대한 소스를 알려주려고 한다. 아이디어란 허공에서 얻어지지 않는다. 아이디어는 똑똑하고 열심히 노력한(많은 Turing-award 수상자³를 포함한) 사람로부터 나온다. 우리는 그런 아이디어와 사람들을 인정해야 한다. 그런 아이디어가 항상 존재해 왔던 것처럼 책을 쓰는 대신 [Kuh62] 그들을 인정함으로써 과거의 혁명적인 발전을 더 잘 이해할 수 있기를 바란다. 게다가 이런 참고 문헌은 당신이 스스로 더 깊은 내용까지 탐구할 수 있도록 이끌 것이다. 우리 분야의 유명한 논문을 읽는 것은 지식을 습득하는 가장 좋은 방법 중의 하나이다.

2) 여담 한 가지: 우리가 여기서 “무료”라는 의미는 공개 소스를 의미하지 않으며 지적재산권에 의해 보호받지 않는다는 것을 의미하는 것이 아니다. 실제로 보호받는다. 무료가 뜻하는 바는 당신이 이 책의 장을 다운로드하여 운영체제를 공부하는 데 사용할 수 있다는 것을 의미한다. Linux 오픈소스 커널처럼, 책을 오픈소스로 못할 이유가 없다. 책 전체가 한 사람의 목소리로 일관성있게 구성되는 것이 매우 중요하다. 이 책 전체를 단일 톤으로 구성하기 위해 심혈을 기울였다. 책을 읽을 때는 설명하는 사람과 대화한다는 느낌이 들어야 한다. 바로 우리의 방식이다.

3) Turing-award는 컴퓨터 과학의 가장 영예로운 상이다. 당신이 들어보지 못했다는 것만 제외하면 노벨상과 유사하다.

감사의 글

이 절에서는 이 책을 쓸 수 있도록 도움을 준 이들에게 감사를 전한다. 중요한 사실 하나: 당신의 이름도 여기 나올 수 있다! 물론 먼저 도움을 주어야 한다. 피드백을 보내거나 이 책의 디버깅에 도움을 주기 바란다. 그리고 유명해질 수 있다. 그렇지 않더라도 몇몇 책에 이름을 올릴 수 있다.

지금까지 도움을 준 분은 다음과 같다: Abhirami Senthilkumaran*, Adam Drescher* (WUSTL), Adam Eggum, Aditya Venkataraman, Adriana Iamnitchi and class (USF), Ahmed Fikri*, Ajaykrishna Raghavan, Akiel Khan, Alex Wyler, Ali Razeen (Duke), AmirBehzad Eslami, Anand Mundada, Andrew Valencik (Saint Mary's), Angela Demke Brown (Toronto), B. Brahmananda Reddy (Minnesota), Bala Subrahmanyam-Kambala, Benita Bose, BiswajitMazumder (Clemson), Bobby Jack, Björn Lindberg, Brennan Payne, Brian Gorman, Brian Kroth, Caleb Sumner (Southern Adventist), Cara Lauritzen, Charlotte Kissinger, Chien-Chung Shen (Delaware)*, Christoph Jaeger, Cody Hanson, Dan Soendergaard (U. Aarhus), David Hanle (Grinnell), David Hartman, Deepika Muthukumar, Dheeraj Shetty (North Carolina State), Dorian Arnold (New Mexico), Dustin Metzler, Dustin Passofaro, Eduardo Stelmaszczyk, Emad Sadeghi, Emily Jacobson, Emmett Witchel (Texas), Erik Turk, Ernst Biersack (France), Finn Kuusisto*, GlenGranzow (College of Idaho), Guilherme Baptista, Hamid Reza Ghasemi, Hao Chen, Henry Abbey, Hrishikesh Amur, Huanchen Zhang*, Huseyin Sular, Hugo Diaz, Itai Hass (Toronto), Jake Gillberg, Jakob Olandt, James Perry (U. Michigan-Dearborn)*, Jan Reineke (Universität des Saarlandes), Jay Lim, Jerod Weinman (Grinnell), Jiao Dong (Rutgers), Jingxin Li, Joe Jean (NYU), Joel Kuntz (SaintMary's), Joel Sommers (Colgate), John Brady (Grinnell), Jonathan Perry (MIT), Jun He, Karl Wallinger, Kartik Singhal, Kaushik Kannan, Kevin Liu*, Lei Tian (U. Nebraska-Lincoln), Leslie Schultz, Liang Yin, Lihao Wang, Martha Ferris, Masashi Kishikawa (Sony), Matt Reichoff, Matty Williams, Meng Huang, Michael Walfish (NYU), Mike Griepentrog, Ming Chen (Stonybrook), Mohammed Alali (Delaware), Murugan Kandaswamy, Natasha Eilbert, Nathan Dipiazza, Nathan Sullivan, Neeraj Badlani (N.C. State), NelsonGomez, Nghia Huynh (Texas), Nick Weinandt, Patricio Jara, Perry Kivolowitz, Radford Smith, Riccardo Mutschlechner, Ripudaman Singh, Robert Ordóñez and class (Southern Adventist), Rohan Das (Toronto)*, Rohan Pasalkar (Minnesota), Ross Aiken, Ruslan Kiselev, Ryland Herrick, Samer Al-Kiswany, Sandeep Ummadi (Minnesota), Satish Chebrolu (NetApp), Satyanarayana Shanmugam*, Seth Pollen, Sharad Punuganti, Shreevatsa R., Sivaraman Sivaraman*, Srinivasan Thirunarayanan*, Suriyhaprakash Balaram Sankari, Sy Jin Cheah, Teri Zhao (EMC), Thomas Griebel, Tongxin Zheng, Tony Adkins, Torin Rudeen (Princeton), Tuo Wang, Varun

Vats, William Royle (Grinnell), Xiang Peng, Xu Di, Yudong Sun, Yue Zhuo (Texas A&M), Yufui Ren, Zef RosnBrick, Zuyu Zhang. * 표시가 되어 있는 분들은 단순한 제안 이상의 실제적 도움을 주신 분들로 특별한 감사를 드린다.

각 장마다 상세한 주석을 달아 주신 Joe Meehean (Lynchburg) 교수, Jerod Weinman (Grinnell) 교수와 엄청난 소책자를 위한 그의 모든 수업, 귀중하고 상세한 검토와 코멘트를 해 준 Chien-Chung Shen (Delaware) 교수, 세심한 검토와 제안을 해 주신 Adam Drescher (WUSTL), 상세한 코멘트와 팁을 알려 주신 Glen Granzow (College of Idaho) 교수, 열정적이며 자세한 제안을 해 주신 Michael Walfish (NYU) 교수에게 진심으로 감사드립니다. 모든 분들이 이 책의 내용을 향상시키는 데 큰 도움을 주셨다.

또, 수년 동안 537 강좌를 수강한 수백 명의 학생들에 감사의 말을 전한다. 특히, 이 책을 처음으로 집필하도록 격려해준 2008년 가을 학기 수강생들(읽을 교재가 없다는 사실을 싫어했던—이런 적극적인 학생들!)은 우리가 책을 계속 쓸 수 있도록 칭찬을 아끼지 않았다. 그해 강의 평가에 “맙소사! 교수님들은 책을 반드시 쓰셔야 합니다”라는 의견을 남긴 학생도 있었다.

랩 수업시간에 xv6 프로젝트에 참가했던 용감한 몇몇 학생들에게 큰 감사의 뜻을 전한다. 이 프로젝트는 이제 537 수업의 일부로 통합되었다. 2009년 봄 학기의 Justin Cherniak, Patrick Deline, Matt Czech, Tony Gregerson, Michael Griepentrog, Tyler Harter, Ryan Kroiss, Eric Radzikowski, Wesley Reardan, Rajiv Vaidyanathan, and Christopher Waclawik. 2009년 가을 학기의 Nick Bearson, Aaron Brown, Alex Bird, David Capel, Keith Gould, Tom Grim, Jeffrey Hugo, Brandon Johnson, John Kjell, Boyan Li, James Loethen, Will McCardell, Ryan Szaroletta, Simon Tso, and Ben Yule. 2010년 봄 학기의 Patrick Blesi, Aidan Dennis-Oehling, Paras Doshi, Jake Friedman, Benjamin Frisch, Evan Hanson, Pikkili Hemanth, Michael Jeung, Alex Langenfeld, Scott Rick, Mike Treffert, Garret Staus, Brennan Wall, Hans Werner, Soo-Young Yang, and Carlos Griffin (거의 대부분).

책 집필에 직접적인 도움을 준 것은 아니지만 시스템에 대해 많은 것을 가르쳐준 대학원생들에게도 크게 감사드립니다. Wisconsin 대학원 재학 시 정기미팅을 가졌지만 대학원생들이 실제적인 모든 작업을 수행하였다. 자신들의 연구내용을 자세히 설명해 주었기 때문에 매주 새로운 사실을 배울 수 있었다. 다음 명단은 같이 논문을 발표했던 이제까지의 학생들을 모두 포함한다. * 표시는 우리 지도 하에 박사 학위를 받은 학생들을 나타낸다. Abhishek Rajimwale, Andrew Krioukov, Ao Ma, Brian Forney, Chris Dragga, Deepak Ramamurthi, Florentina Popovici*, Haryadi S. Gunawi*, James Nugent, John Bent*, Jun He, Lanyue Lu, Lakshmi Bairavasundaram*, Laxman Visampalli, Leo Arulraj, Meenali Rungta, Muthian Sivathanu*, Nathan Burnett*, Nitin Agrawal*, Ram Alagappan, Sriram Subramanian*, Stephen Todd Jones*, Suli Yang, Swaminathan Sundararaman*, Swetha Krishnan, Thanh Do*, Thanumalayan S. Pillai, Timothy Denehy*, Tyler Harter, Venkat Venkataramani, Vijay Chidambaram, Vijayan Prabhakaran*, Yiying Zhang*, Yupu Zhang*, Zev Weiss.

마지막으로 몇 년 전에 (2009년 봄 학기) 이 수업을 수강하고 2009년 가을 학기에 xv6 랩 수업을 들은 후 2010년 가을부터 2012년 봄 학기까지 2년여 남짓 수업 조교로 도움을 준 Aaron Brown, 그의 지치지 않은 작업은 프로젝트의 수준, 특히 xv6 관련한 영역의 수준을 광범위하게 향상시켰다. Wisconsin에 재학 중이거나 졸업한 수많은 학생들이 수업을 듣는 데 많은 도움을 주었다. Aaron이 평소에 간결하게 말하던 것처럼 “Thx.”

맺음말

Yeats는 “교육은 양동이를 채우는 것이 아니라 불을 지펴주는 것이다”라는 유명한 말을 남겼다. 옳은 말이기도 하지만 동시에 틀린 말이기도 하다⁴. 여러분들은 양동이에 물을 채워야만 하고, 이 노트가 양동이를 채우는 데 도움이 될 것이다. Google에 면접보러 가서 세마포어의 용법에 관한 질문을 받았을 때, 세마포어가 무엇인지 알아야 답변할 수 있을 것이다. 그렇지 않나?

Yeats의 경구는 광의적 측면에서 옳은 말이다. 교육의 핵심은 여러분들이 무언가에 흥미를 느끼게 만들고, 그 주제에 대해 스스로 더 공부하도록 만드는 것이지 주어진 지식을 습득하여 수업 시간에 좋은 성적만 받으면 되도록 하는 것은 아니다. 저자의 아버지께서는(Remzi의 아버지, Vedat Arpacı) “수업에서만 배우려고 하지 마라” 라고 말씀하시곤 했다.

여러분들이 운영체제에 흥미를 갖고, 스스로 공부하고, 현장에서 진행되는 흥미로운 연구에 대해 교수들과 토론하고, 참여를 독려하기 위해 이 책을 썼다. 운영체제는 컴퓨팅 역사에 매우 중요한 영향을 준 흥미롭고 경이로운 아이디어로 가득 찬 굉장한 분야이다. 이 불꽃이 모두는 아니더라도, 많은 수의, 아니 여러분들 중 단 몇명에게라도 운영체제에 대한 열정을 심어줄 수 있기 바란다. 이 열정이 생기기 시작할 때가 바로 여러분들이 위대한 일을 할 수 있게 되는 시점이기 때문이다. 교육의 핵심: 여정을 시작하여 많은 새롭고 흥미로운 주제를 공부하고, 스스로 학습하고, 성숙하고, 가장 중요하게 여러분들에게 불을 지필 무언가를 찾는 것이다.

Andrea and Remzi

Married couple

Professors of Computer Science at the University of Wisconsin

Chief Lighters of Fires, hopefully⁵

4) 그가 정말 이 말을 했다고 한다면, 다른 유명한 인용구처럼 이 명언의 기원은 불분명하다.

5) 이 말이 우리가 방화범이었다는 사실을 인정하는 것처럼 들린다면, 아마도 여러분은 핵심을 놓치고 있는 것이다. 아마도, 만일 약간 저급하게 들린다면, 음, 실제로 그렇기 때문이다. 그러나 여러분들은 그 점에 대해서는 용서해 주기 바란다.

참고 문헌

[Cox+] **“The xv6 Operating System”**

Russ Cox, Frans Kaashoek, Robert Morris, and Nickolai Zeldovich

From: <http://pdos.csail.mit.edu/6.828/2008/index.html>

xv6는 UNIX 버전 6를 포팅하여 개발되었으며 현대의 운영체제를 이해하기 위한 아름답고 깨끗하며 단순한 방법을 제공한다.

[Fey96] **“Six Easy Pieces: Essentials Of Physics Explained By Its Most Brilliant Teacher”**

Richard P. Feynman

Basic Books, 1996

이 책은 1963년도에 출판된 파인만의 여섯 가지 물리 이야기를 재출판한 것이다. 물리학을 좋아한다면 환상적인 책이다.

[KR] **“The C Programming Language”**

Brian Kernighan and Dennis Ritchie

Prentice-Hall, April 1988

C 프로그래밍 언어를 개발한 사람들이 쓴 책으로 이 언어의 참고 문헌으로 누구나 갖고 있어야 하는 책이다.

[Kuh62] **“The Structure of Scientific Revolutions”**

Thomas S. Kuhn

University of Chicago Press, 1962

과학적 연구방법론의 핵심에 관한 유명한 책이다. 마무리 작업, 이상 현상, 위기와 혁명이다. 우리는 마무리 작업을 할 수밖에 없는 운명이다, 아아.

[PH90] **“Computer Architecture a Quantitative Approach (1st ed.)”**

David A. Patterson and John L. Hennessy

Morgan-Kaufman, 1990

두 저자가 학부에 있을 때 대학원으로 진학하도록 자극을 준 책이다. 이후에 Patterson과 같이 작업하는 기쁨을 누릴 수 있었으며 연구 경력의 기틀을 잡는 데 큰 도움을 주었다.