Syllabus

Operating Systems

School of Data & Computer Science
Sun Yat-sen University

Lecture Notes: os_sysu@163.com
Instructor: Guoyang Cai
email: isscgy@mail.sysu.edu.cn

A. Description

- The Importance of the course of Operating Systems
  - *Some of you* may design and build operating systems or components of them.
    - Perhaps more now than ever.
  - *Many of you* will create systems that utilize the core concepts in operating systems.
    - Whether you build software or hardware.
    - The concepts and design patterns appear at many levels.
  - *All of you* will build applications that utilize operating systems
    - How to correctly writing your applications.
    - The better you understand design and implementation of operating systems, the better use you'll make of them.

A. Description

- The Importance of the course of Operating Systems
  - It is a fundamental computer science and engineering course with full of challenges.
    - Operating systems are Large and complex systems that have a high economic impact and result in interesting problems of management.
    - It combines concepts from many other areas of Computer Science.
      - Architecture
      - Languages
      - Data Structures
      - Algorithms
      - etc.

A. Description

- Prerequisites
  - Principles of Computer Organization
  - Data Structures
  - Proficiency in C
  - X86 assembly language
  - *Linux programming environment
    - POSIX

A. Description

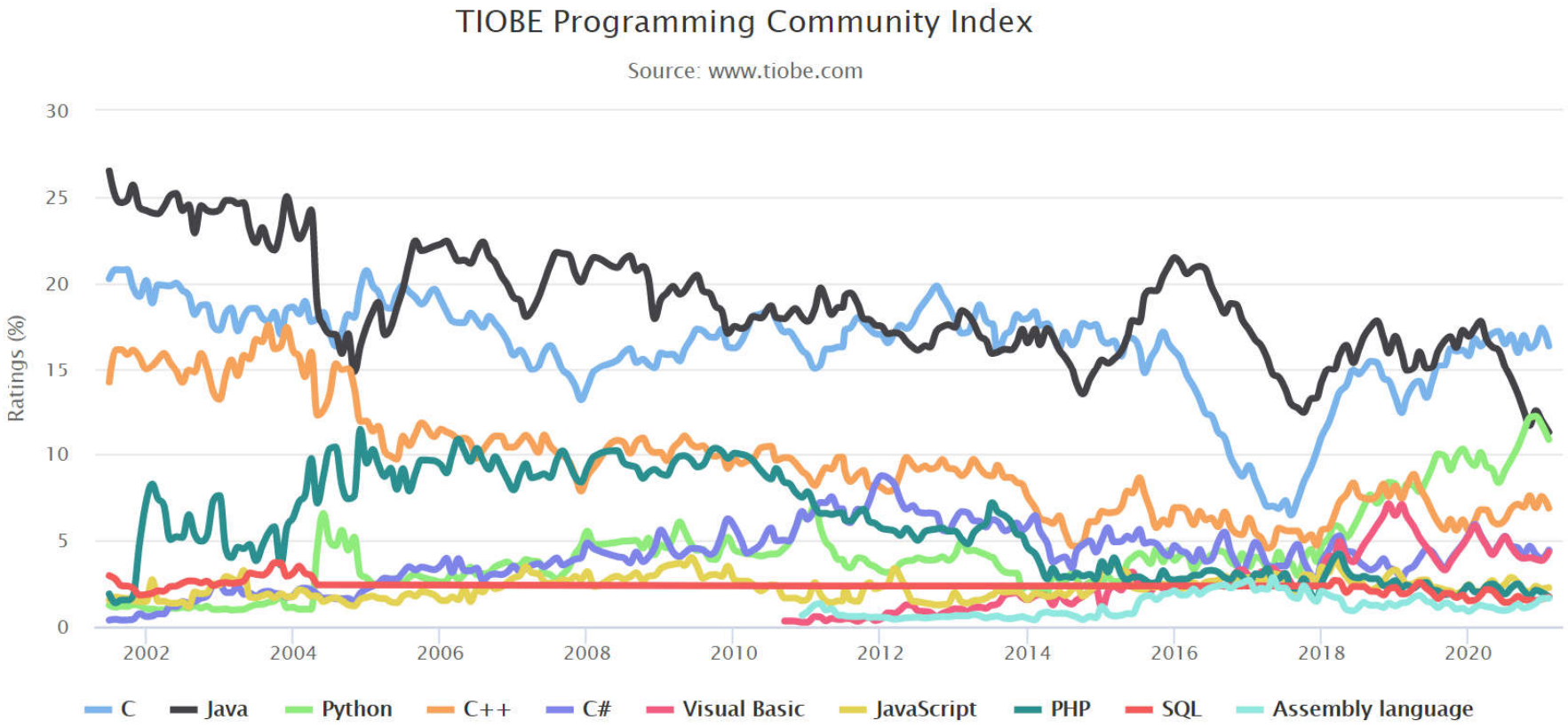
- The TIOBE Programming Community Index
  - indicator of the popularity of programming languages
  - ratings based on the number of skilled engineers world-wide, courses and third party vendors

Feb 2021	Feb 2020	Change	Programming Language	Ratings	Change
1	2	↑	C	16.34%	-0.43%
2	1	↓	Java	11.29%	-6.07%
3	3		Python	10.86%	+1.52%
4	4		C++	6.88%	+0.71%
5	5		C#	4.44%	-1.48%
6	6		Visual Basic	4.33%	-1.53%
7	7		JavaScript	2.27%	+0.21%
8	8		PHP	1.75%	-0.27%
9	9		SQL	1.72%	+0.20%
10	12	↑	Assembly language	1.65%	+0.54%

TIOBE Top 10 in Feb 2021

A. Description

The TIOBE Programming Community Index



A. Description

- The TIOBE Programming Community Index
  - The Programming Language Hall of Fame.
    - The programming language that has the highest rise in ratings in a year.

Year	Winner
2020	Python
2019	C
2018	Python
2017	C
2016	Go

Programming Language Hall of Fame for recent 5 years

A. Description

- Why Linux?
  - Linux, original author of *Linus Torvalds*, contributed by thousands of programmers all over the world, has snowballed into something so big that companies like IBM and Sun stopped making their own operating systems and they sell their servers with Linux on them instead.
    - When Google wanted to create a new operating system for phones, they used Linux to create Android.
  - Linux succeeded because of community. Those contributors and that sharing of source code were the keys to success.
    - Linux was an “Open Source” operating system. The system and its source code are free, but not in the public domain.
    - They are licensed under the GNU Public License (GPL), which is a license that says you don’t have to pay but you must share.
    - Read more about the GPL at the Free Software Foundation.

A. Description

- Why Linux?
  - Free and Open Source Software (FOSS) is a movement in the technology industry over the past twenty years. The collaborative approach to building software is a way to harness the power of every programmer in the world on a single project.
    - Some of the most successful examples we have seen are Linux, the Apache webserver, MySQL, PHP, WordPress, and OpenOffice.org.
  - The collaborative nature of these projects has the effect of "keeping programmers honest."
    - It has been widely speculated that some famous products from big companies have "back doors" in their systems that allow to access private user data. Surely these claims are denied and it is not really possible to independently verify.
    - Open Source technologies do not have back doors because everyone can look inside and check.

A. Description

- Why Linux?
  - When vulnerabilities are found in open source software, they are widely publicized and fixed quickly. They can be fixed quickly because anyone who has the skills can do it.
  - Some advantages of Linux:
    - Stability
    - Affordability
    - Speed
    - Cost
    - Performance
    - Reliability

B. Course Organization

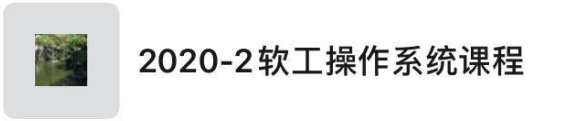
- Lecture Time
  - **Tuesday** 14:20-16:00, C302 (W1-9, W11-19)
  - **Thursday** 14:20-16:00, C302 (W1-9, W11-19)
- Lab Time
  - **Thursday** 10:00-11:40, 南实验楼 D402 (W1-9, W11-19)

- Instructor: 蔡国扬, email: *isscgy@mail.sysu.edu.cn*
- TA: 郭孔靖, *guokj@mail2.sysu.edu.cn*,
  陈皓炜, *chenhw39@mail2.sysu.edu.cn*

- Classroom Rules of Conduct
  - Attendance
    - The University requires attendance in all classes.
  - Cell phone using
  - Laptop using
  - Food & drink
  - Being Late for class

B. Course Organization

- The Wechat group (2020-2软工操作系统课程)
  - Member’s naming rule: studentno_studentname
  - Members violating the naming regulation will be kick out anytime.
  - No gossip, no giving liked, …



该二维码7天内(2月28日前)有效，重新进入将更新

C. Course Objectives

- General Goals
  - To provide the necessary general knowledge of the fundamental concepts underlying the design and implementation of operating systems, including the concepts of
    - System structures
    - Processes
    - Threads
    - Mutual Exclusion
    - Synchronization
    - Deadlock
    - Processor Management
    - Real/Virtual Memory Management
    - Mass Storage Systems
    - File systems
    - Input/output Subsystems
    - Protection and Security issues.

C. Course Objectives

- Specific Goals
  - Ability to describe an O/S as an abstract machine; to describe privileged & user states and how they are accessed and controlled
  - Ability to use and explain the semantics and operation of semaphores, condition variables, monitors and critical sections
  - Ability to implement programs using processes and threads
  - Ability to explain the consequences of various page replacement algorithms
  - Ability to explain the hardware architecture requirements supporting any O/S

D. Course Topics

- Introduction
  - What Operating Systems Do
  - Evolution of Operating Systems
  - Computer System Organization & Architectures
  - Functional Views of Operating System
  - Operating System Structures
- Processes
  - Fundamental Concepts of Process
  - Process Scheduling
  - InterProcess Communication
- Threads
  - Multicore Programming
  - Multithreading Models
  - Threads Libraries
- Process Synchronization
  - Cooperating Processes
  - Process Synchronization

D. Course Topics

- CPU Scheduling
  - Scheduling Criteria
  - Simple Scheduling Algorithms
  - Advanced Scheduling Algorithms
  - Algorithms Evaluation
- Main Memory Management
  - Introduction
  - Memory Partitioning
  - Memory Segmentation & Paging
- Virtual Memory Management
  - Demand Paging
  - Demand Segmentation
  - Virtual Memory Policies
- File systems
  - Mass Storage Systems
  - File System Interface
  - File System Implementation

D. Course Topics

- I/O Systems
  - I/O Hardware
  - Application I/O Interface
  - Kernel I/O Subsystem
- Security
  - The Security Problem
  - Threats
  - Cryptography
  - Authentication
  - Security Defenses
- Protection
  - Principles of Protection
  - Access Matrix
  - Access Control

E. Course Schedule (Tentative)

- Assigns: 40 rewriting algorithms, 7 designs other than some assembly works. (2 uCore projects scored only for lab course)

1	00. Syllabus 01. Introduction to Operating Systems (Project1 released.)
2	02. Computer System Organization & Architecture 03. Functional Views of Operating Systems Assign: Assembly works of Blum's chap04, chap05
3	04. Structures of Operating Systems (1) 05. Structures of Operating Systems (2) Assign: Assembly works of Blum's chap06, chap07
4	06. Introduction to Process (1) 07. Introduction to Process (2) Assign: Assembly works of Blum's chap08, chap10
5	08. InterProcess Communication (1) Shared Memory 09. InterProcess Communication (2) Message Passing Assign: Alg.6-1~6-6

E. Course Schedule (Tentative)

- Assigns: 40 rewriting algorithms, 7 designs other than some assembly works. (2 uCore projects scored only for lab course)

6	10. InterProcess Communication (3) Pipe 11. Threads (1) Assign: Alg.8-1~8-5
7	12. Threads (2) 13. Threads (3) Assign: Design a concurrent heap with mem-sharing
8	14. Cooperating Processes(1) 15. Cooperating Processes(2) Assign1: Alg.9-1~9-2 Assign2: Design a concurrent heap with mem-sharing and message-passing
9	16. Process Synchronization (1) 17. Process Synchronization (2) Monitor Assign: Alg.10-1~10-4
10	Midterm Test

E. Course Schedule (Tentative)

- Assigns: 40 rewriting algorithms, 7 designs other than some assembly works. (2 uCore projects scored only for lab course)

11	18. Process Synchronization (3) Deadlock 19. Process Synchronization (4) Examples Assign: Alg.12-1~12-8
12	20. CPU Scheduling (1) 21. CPU Scheduling (2) Advanced and Multi-processor Scheduling Assign: Alg.13-1~13-5 (Project2 released.)
13	22. CPU Scheduling (3) Real-time Scheduling 23. Introduction to Memory Management Assign: Design a thread pool with pthread API
14	24. Memory Segmentation & Paging 25. Virtual Memory & Demand Paging (1) Assign: Rewrite Alg. 8-3 with Peterson Algorithm.
15	26. Virtual Memory & Demand Paging (2) 27. Virtual Memory & Demand Paging (3) Virtual Memory Policies Assign: Alg.19-1~19-9

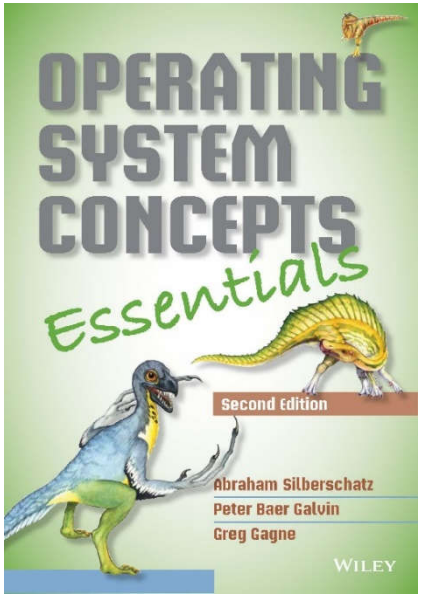
E. Course Schedule (Tentative)

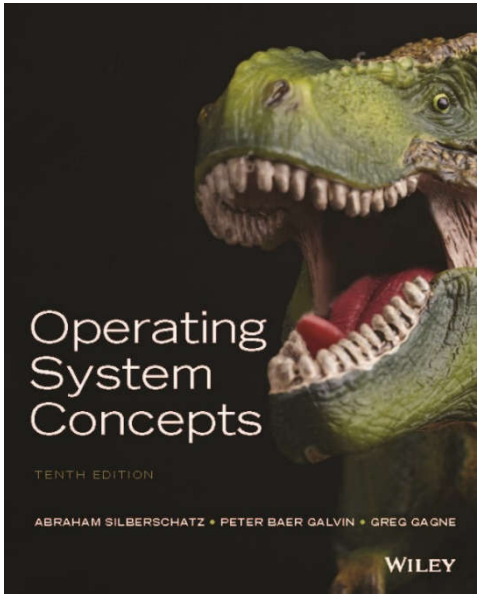
- Assigns: 40 rewriting algorithms, 7 designs other than some assembly works. (2 uCore projects scored only for lab course)

16	28. Mass Storage Systems 29. File System Interface Assign: Design a CPU scheduling simulator with FCFS, SJF, PS, RR, MPQ and MFQ policies
17	30. File System Implementation 31. I/O Systems Assign: Design a Demand paging simulator with FIFO, LRU and Second chance policies
18	32. Security 33. Protection Assign: Design a DISK scheduling simulator with FCFS, SSTF, SCAN, C-SCAN, LOOK and C-LOOK policies
19	Reviews
20	Final Exam

F. Text Books and Required Supplies

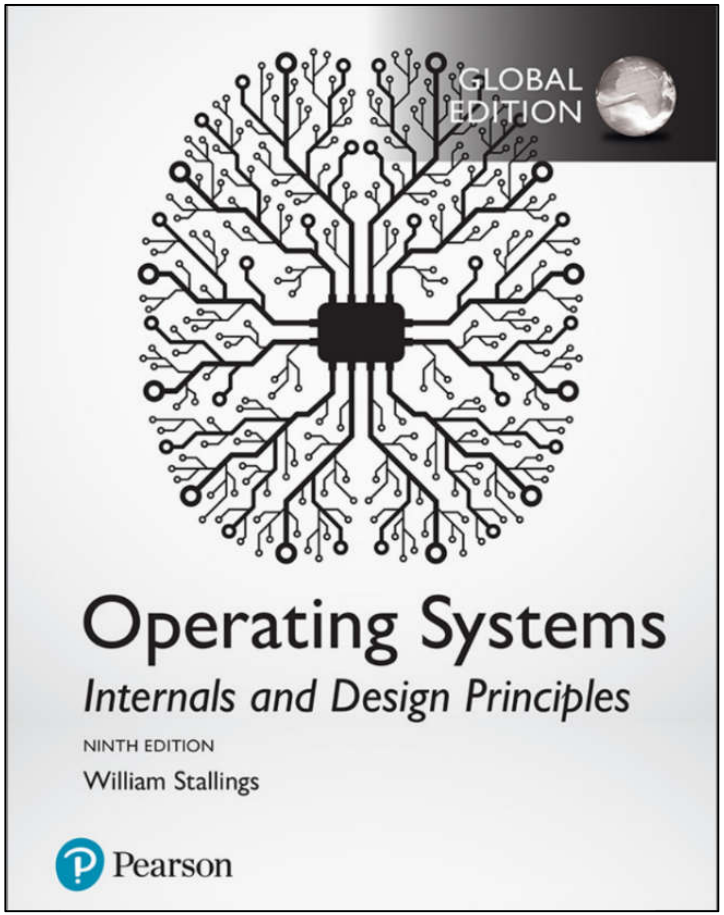
- Lecture Notes
  - Check os_sysu@163.com, os_sysu@sina.com / se2019_sysu
- Textbooks
  - Silberschatz, Galvin, and Gagne, *Operating System Concepts*, 10th Edition, Wiley 2018.
  - Silberschatz, Galvin, and Gagne, *Operating System Concepts essential*, 2th Edition, Wiley 2014.
  - *Students are asked to read through one of the textbooks.*





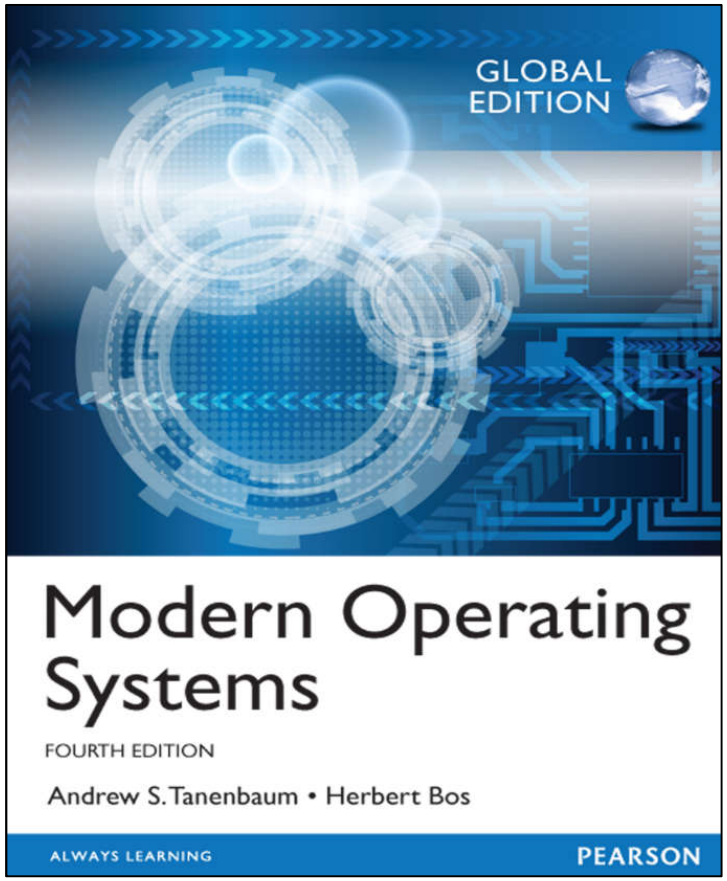
F. Text Books and Required Supplies

- References
  - William Stallings, *Operating Systems Internals and Design Principles,* 9th Edition, Pearson 2018.



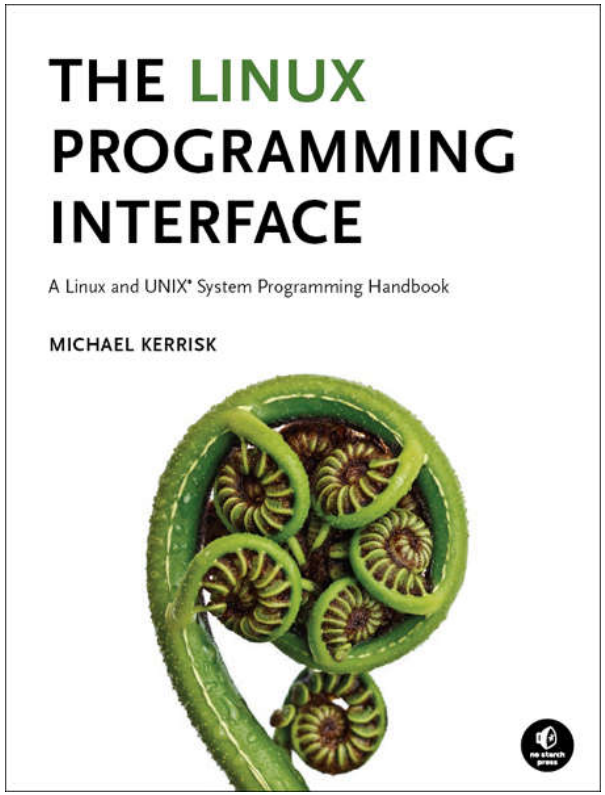
F. Text Books and Required Supplies

- References
  - Andrew S. Tanenbaum, *Modern Operating Systems,* 4th Global Edition, Pearson 2015.



F. Text Books and Required Supplies

- References
  - Michael KerrisK, *The Linux Programming Interface,* No Starch Press 2011.
    https://man7.org/



F. Text Books and Required Supplies

- References
  - 张尧学，计算机操作系统教程 (第4版) 习题解答与实验指导

F. Text Books and Required Supplies

- References
  - 清华大学计算机系2020春季操作系统课程主页

    http://os.cs.tsinghua.edu.cn/oscourse/OS2020spring
    - 课堂和课后练习
      - 操作系统习题集、课堂练习和讨论题
    - 课堂&实验日程安排
      - 第1讲 – 第22讲 课件阅读

F. Text Books and Required Supplies

- Lab Guide
  - Architecture: Intel X86-32/64
  - Operating System: *Ubuntu* 1804/2004
  - Compiler: gcc (GNU C Compiler)
  - QEMU/VB/VM/…
  - Markdown Editor (remarkable, atom, …)
  - Reference web:
    - *https://www.kernel.org/doc/man-pages/*
    - *http://man7.org/linux/man-pages/index.html*
    - *http://www.gnu.org/software/software.en.html*
    - *https://gcc.gnu.org/*
    - *http://www.gnu.org/software/libc/libc.html*
    - *https://sourceware.org/*
    - *https://sourceware.org/gdb/*
    - *https://sourceware.org/binutils/docs/*
    - *https://sourceware.org/binutils/docs/as/*
    - *https://sourceware.org/binutils/docs/ld/*

G. Grading Plan

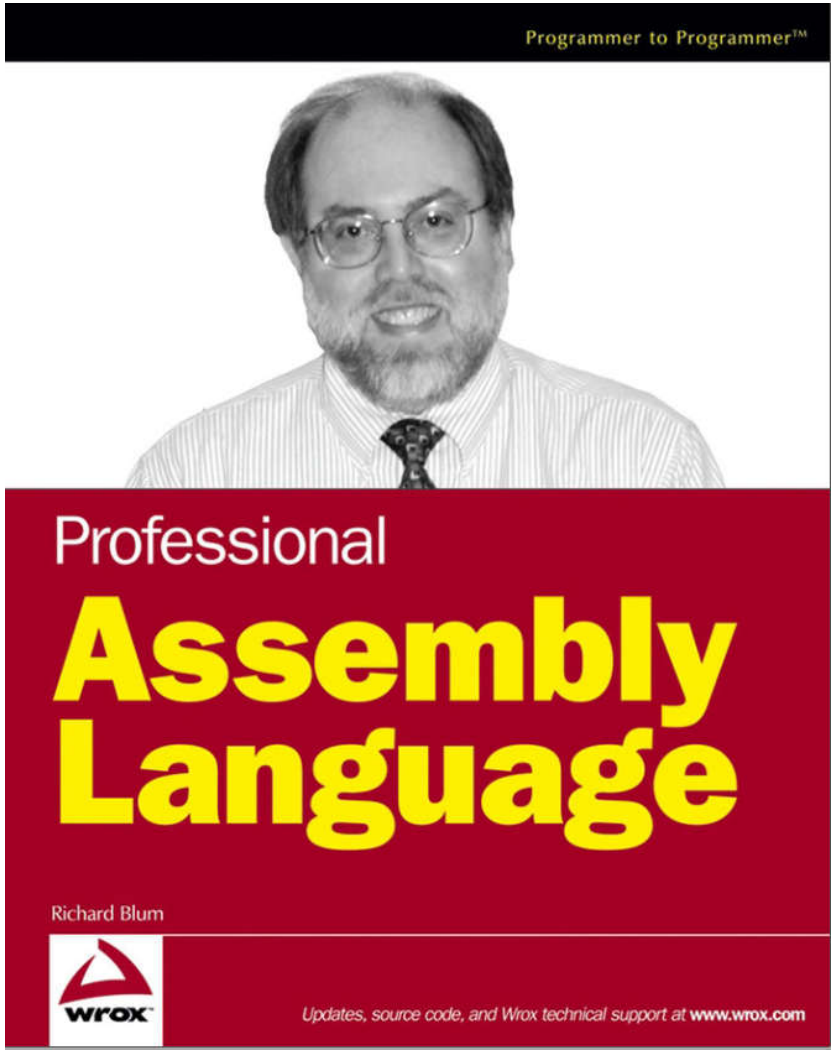
- Theory Course
  - Attendance 10%.
  - Sixteen weekly lab works 30%.
  - Midterm Exam 20%.
  - Final Exam 40%.
- Lab Course
  - Attendance.
    - No lab-time can be missed.
    - 5 points lost in total score for every absence.
  - Sixteen weekly lab works 60%.
    - Submitting your lab works on Matrix on time.
  - Two Lab Projects 20%.
    - emailing to: *os_assignment_sysu@163.com*
  - Midterm Exam 10%.
  - Final Exam 10%.

H. Academic Offences

- Academic Offences include, but are not limited to:
  - Infringing unreasonably on the work of other members
    - E.g., disrupting classes
  - Cheating
  - Plagiarism
  - Misrepresentations
- **Uphold academic honesty in completing your assignments, projects, and exams**
  - You can discuss the problems with your classmates, but all work handed in should be original, written by you in your own words and submit to TA on time.

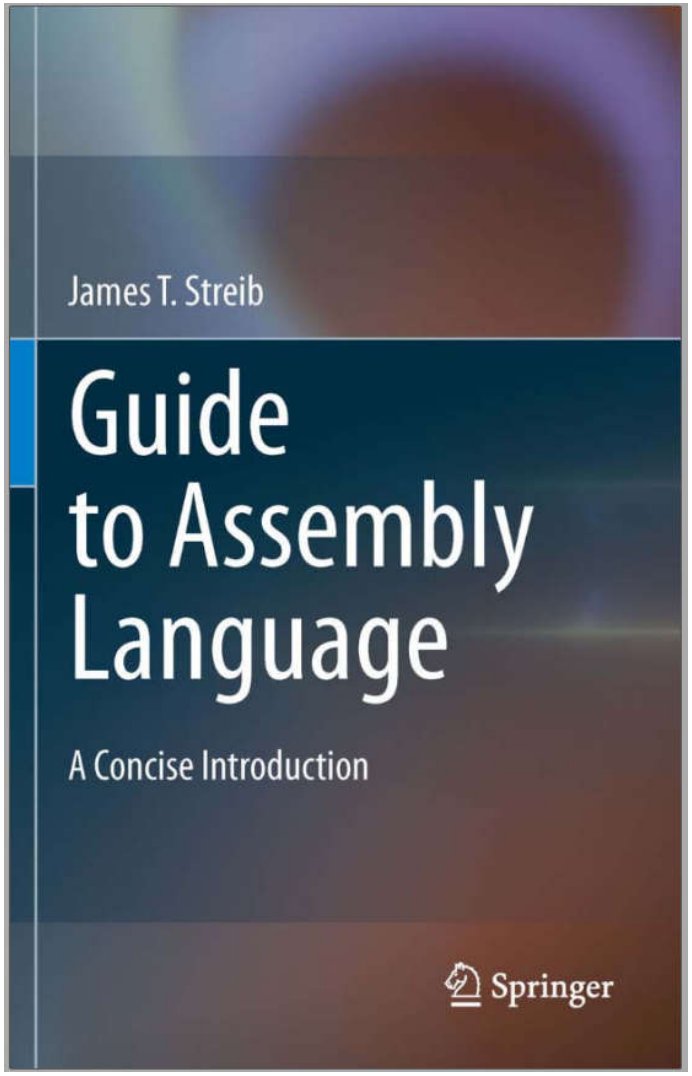
References

- Richard Blum, *Professional Assembly Language*, Wiley 2005.



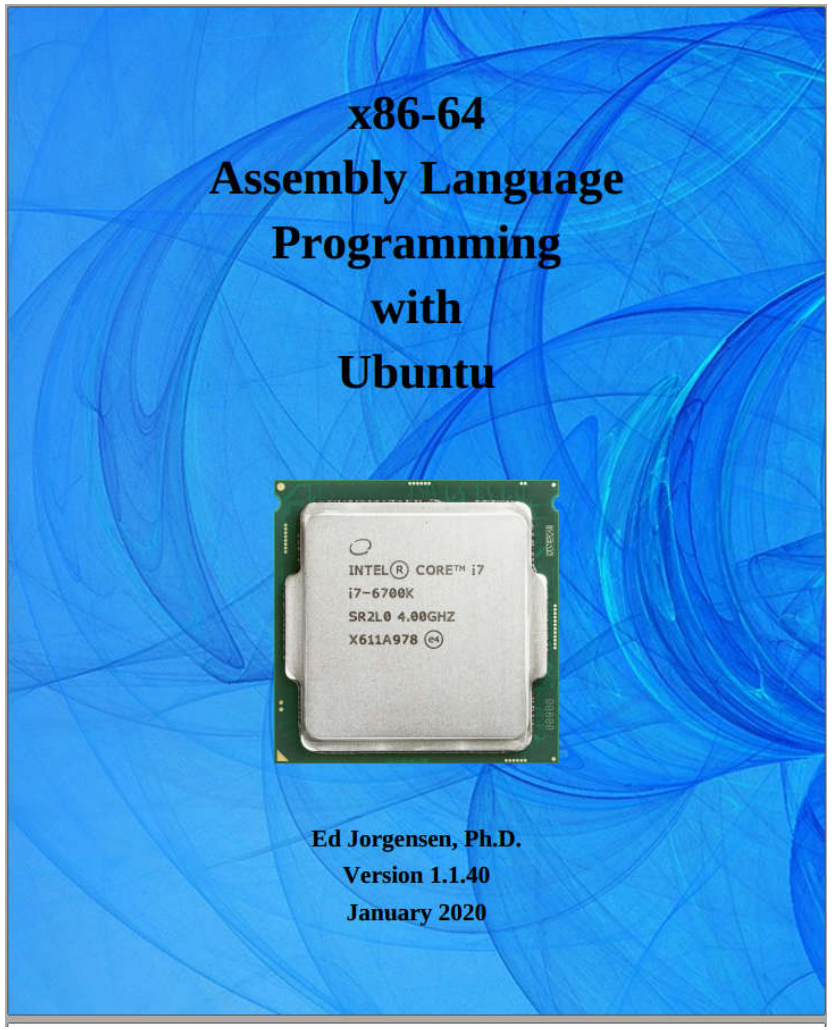
References

- James T. Streib, *Guide to Assembly Language*, Springer 2011.



References

- Ed Jorgensen, *x86-64 Assembly Language Programming with Ubuntu*, FDL 2020 (last version)



Lab Environments

- Architecture: Intel X86-32/64
- Operating System: *Ubuntu* 1804/2004
- Assembler:
  - `as` (GNU Assembler) in AT&T mode and `ld` linker
  - `gcc`

  https://sourceware.org/binutils/docs/as/

Course Schedule

- Lab Time: week2
  - *Blum*'s book: Sample programs in Chapter 04, 05 (Moving Data)
- Lab Time: week3
  - *Blum*'s book: Sample programs in Chapter 06, 07 (Controlling Flow and Using Numbers)
- Lab Time: week4
  - *Blum*'s book: Sample programs in Chapter 08, 10 (Basic Math Functions and Using Strings)
- Note*: Blum*' sample programs are based on x86-32. Modification may be needed in source or in compiling if you have a 64-bit system. E.g.,
  - Instruction `push`, `pop` in x86-64 assembler
  - Multiple `_start` definition / `-m32` option in `gcc` compiling

uCore Lab Project

References

- uCore Lab Project from CST, Tsinghua.
  - 清华大学计算机系2020春季操作系统课程主页
    http://os.cs.tsinghua.edu.cn/oscourse/OS2020spring
    - 教学实验
      - uCore_OS相关内容

    https://chyyuu.gitbooks.io/ucore_os_docs/content/
    https://github.com/LearningOS/ucore_os_docs
    https://github.com/LearningOS/ucore_os_lab

Lab Environments

- Architecture: Intel X86-32/64 (Real or Virtual machine)
- Operating System: *Ubuntu* 1804/2004
- Assembler: gas (GNU Assembler) in AT&T mode
- Compiler: gcc

Course Schedule

- Lab Project1: uCore Lab 1 系统软件启动过程 (Week 01 ~ Week 08)
- Lab Project2: uCore Lab 2 物理内存管理 (Week 12 ~ Week 18)