ISTD 50.005: Computer System Engineering (Spring 2019)

I. Synopsis

This course is about the design and implementation of complex computing systems, from single computer systems to a large-scale network of them. At the end of the course, students will be able to:

- Understand and analyze the challenges of designing and implementing complex computing systems, particularly modern operating systems and internet systems.
- Understand and make use of basic tools and methods to overcome the challenges of creating computing systems that are easy to manage, flexible and easy to extend, have high performance, and are reliable and secure.
- Understand and analyze the basic performance of computing systems.

II. Pre-requisites

- Good standing as Term 5 student in computer or information systems and engineering.
- Ability to understand program snippets in **both** Java and C.
- Ability to write extended programs in Java and, optionally, C.

III. Workload and grading (tentative and subject to change)

- In-class activities, quizzes, or homeworks (12%). 1 A4 size double sided cheat sheet, **you can type/ print this cheat sheet. Doesn't have to be handwritten.**
- Programming assignments x 2 (10%).
- Labs x 7 (14%).
- Exam 1 (32%) and Exam 2 (32%) : 1 A4 size double sided cheat sheet, **HANDWRITTEN only**

Notes:

- Unless otherwise specified, all work is to be submitted via eDimension.
- In-class activities and written homeworks (*excluding* labs and programming assignments) won't be graded. However, you should show a real attempt in answering the questions, and you must submit your answers by the deadline to get credit. By default, in-class activities are due within class time. To cover any contingencies, students can drop two in-class activities or written homeworks in the semester.
- Regular quizzes will be conducted in class. By default, you should be prepared for a quiz every time the class meets for the lab session, and the materials tested will be the portion covered in the lectures since the previous quiz. Quizzes will be held during the lab session. To cover any contingencies, students can drop the lowest two quizzes in the semester.
- Programming assignments are to be completed in groups of two students. Unless otherwise specified, all other work is individual work.
- Unless otherwise specified, you can use either Java or C for your lab programs. You can implement programming assignment 1 in either Java or C; programming assignment 2 must be implemented in Java.
- To cover any contingencies, students will have an allowance of two late days (**total** for the semester) for submitting their labs, and another two late days (**total**) for submitting their programming assignments. Once an allowance is used up, any late submissions will not be graded or earn credit.
- **Important**. Policy of academic honesty will be strictly adhered to. In particular, everything (including texts and code) that you submit must be originally authored by you (or your group in the case of group assignments), except that you can use content from the official course textbooks (but not any solution manuals). If you copy, quote, or paraphrase any other sources (including but not limited to friends, classmates, work by previous students in the course, sample solutions given previously by instructors of the course, solution manuals of any textbooks, textbooks other than the official ones listed in Sec. IV below, and any internet or public resources), you must *explicitly* reference the sources. Borrowing liberally or unnecessarily from third-party sources will reduce your credit, but proper citation will exonerate you from academic dishonesty. We will use automated tools to check for plagiarism of writing or programs.

IV. Textbooks

Silberschatz, Galvin, Gagne. Operating Systems Concepts with Java, 8th Ed. Wiley. ISBN 978-0-470-50949-4. **(required) - Abbreviated as "SGG" in this document**
Kurose and Ross. Computer Networking – A Top-Down Approach, 7th Ed. Pearson. ISBN 978-0-273-76896-8. **(required) - Abbreviated as "KR" in this document**
Kernighan and Ritchie. The C Programming Language, 2nd Ed. Prentice Hall. ISBN 978-0131103627. (optional)

V. Meeting times and venue

- Cohort 1 Lectures: Mon 10:00 -11:30 and Wed 9:00 - 10:30 at CC13 (2.506).
- Cohort 2 Lectures: Mon 10:00 -11:30 and Wed 11:30 - 13:00 at CC13 (2.506).
- Cohort 3 Lectures: Mon 11:30 -13:00 and Wed 11:30 -13:00 at CC14 (2.507).
- Combined Labs: Thurs 10:3 - 12:30 (combined) at LT 5(2.505).

VI. Teaching staff

Staff Name and email address	Office Hours
Instructors: Prof. David Yau (david_yau@sutd.edu.sg)	Mon 1:30 to 3:30pm
Dr Natalie Agus (natalie_agus@sutd.edu.sg)	By email
TAs: Toffalini Flavio (flavio_toffalini@mymail.sutd.edu.sg)	
Juan Hernandez (juan_guarnizo@mymail.sutd.edu.sg)	
UTAs: Tan Jia Wen (jiawen_tan@mymail.sutd.edu.sg	
Isaac Ashwin (isaac_ravindran@mymail.sutd.edu.sg)	

VII. Syllabus (tentative and subject to change)

Week	Date	Topic	Lab / Project
1	28 Jan	OS introduction OS structures and shell	• Lab 1: OS shell • Quiz 1
2	4 Feb	OS structures and shell (cont'd) *(second class of week cancelled due to CNY)*	• *(No lab)*
3	11 Feb	Process and thread management	• **Programming assignment 1 release and discussion** • Quiz 2
4	18 Feb	Process synchronisation	• Lab 2: Multi-threading • Quiz 3
5	25 Feb	Deadlocks	• Lab 3: Banker's algorithm • Quiz 4
6	4 Mar	File system	• Lab 4: File operations • Quiz 5
7	11 Mar	*(term break)*	**Programming assignment 1 due**
8	18 Mar	Network basics *(second class of week cancelled due to midterm exam)*	**Midterm exam during lab 10:30am-12:30pm**
9	25 Mar	Network basics (cont'd) Network performance	• Lab 5: ping/traceroute • **Programming assignment 2 released** • Quiz 6

10	1 Apr	Network security	• Lab 6: Java JCE • Quiz 7
11	8 Apr	Network security (cont'd) Internet naming and addressing	• Lab 7: DNS • Quiz 8
12	15 Apr	Internet naming and addressing (cont'd) Client-server and the web	• Consultation for programming assignment 2 • Quiz 9
13	22 Apr	Client-server and the web (cont'd) Programming assignment 2 demos	**Programming assignment 2 due**
14	29 Apr	**Final exam** **(3 May 2019, 3-5pm)**	*(no lab)*

Learning Objectives

Week 1: SGG Ch. 1.1 - 1.6, 1.8.3, 1.13.3

1. Define what is an operating system
2. Describe basic roles of an operating system
3. Know the difference between User vs Kernel mode, and how to switch between each mode using system calls and interrupt
4. Describe computer system organization (basic architecture)
5. Describe how I/O interrupts occur and how the Kernel handles the interrupt
6. Describe and understand the basic structure of storage / memory device hierarchy
7. Understand the meaning of timesharing, context switch, scheduling, and multiprogramming - all of which is part of OS process management
8. Understand the basic difference between process and program

Week 2: SGG Ch. 2.1 - 2.6, 2.7.1 - 2.7.3, 2.9.2

1. Understand a variety of OS services and their applications
2. Describe ways on how to access OS services
3. Understand the difference between making direct system calls vs through API
4. Describe the possible ways to pass parameters using system calls
5. Describe and understand different types of system calls and their purposes
6. Understand the difference between user programs and system programs
7. Describe and understand different types of system programs and their purposes
8. Understand the basic principles behind OS design: simple, layered, microkernel using examples: macOS, JX, MSDos, and UNIX.

Week 3: SGG Ch. 3.1-3.3, 3.4.1, 3.6.1, 4.1-4.2, 4.3.1, 4.4, 4.5.1 - 4.5.2

1. Understand the concept of processes, and the difference between codes, programs and processes
2. Understand the difference between concurrency and parallelism
3. Describe and understand various process scheduling states and their transitions
4. Know the process management terms: process control block, and context switch.
5. Describe different types of process queues and I/O queues
6. Know how to create a new process, e.g: using fork() in C or ProcessBuilder in Java
7. Understand how interprocess communication (IPC) works
8. Understand the concept of threads and its difference with process
9. Describe two types of threads: user and kernels and their models

Week 4: SGG Ch. 6.1-6.4, 6.5.1, 6.5.2, 6.6.1, 6.8.1 - 6.8.2, 6.8.4

1. Understand issues of multiprogramming and concurrency: the race condition
2. Describe and understand the concept of critical section
3. Discuss how Peterson's solution solve the critical section problem
4. Understand how synchronisation hardware works in general
5. Describe and explain the concepts of mutex and condition synchronization
6. Understand other critical section solutions that do not require busy waiting: semaphores
7. Explain how semaphores can be used as a mutex and condition synchronization
8. Understand how Java implements synchronization using synchronized methods and named condition variables

Week 5: SGG Ch. 7.1-7.4, 7.5.1, 7.5.3, 7.6.2, 7.7

1. Understand the problem of deadlock
2. Describe necessary conditions for deadlock
3. Draw and explain resource allocation graphs
4. Discuss ways to handle deadlocks through prevention, detection, and recovery
5. Describe and understand deadlock avoidance algorithm: Banker's algorithm
6. Describe and understand deadlock detection algorithm
7. Understand the difference between deadlock avoidance algorithm and deadlock detection algorithm

Week 6: SGG Ch.10.1, 10.2, 10.3.2 - 10.3.7, 10.6, 11.2.1

1. Understand what is a file system and the file system name space
2. Understand file format and type
3. Describe and explain possible operations on file
4. Describe the concepts and the difference between file data and metadata (Attributes)
5. Learn the concepts of file descriptors and file system data structure, i.e: how the OS stores files in the memory
6. Understand UNIX file system data structures and mapping between table entires
7. Describe the concepts of directory structures : tree structures (acyclic and cyclic)
8. Discuss the difference between symbolic and hard links

Week 8: KR Ch. 1.1, 1.3, 1.5

1. Understand what is a network and the internet
2. Discuss challenges for the internet
3. Explain what is a protocol
4. Understand the difference and pros/cons between time domain multiplexing, frequency domain multiplexing, and packet switching
5. Understand the concept of priority inversion and emergent behavior
6. Know the internet protocol stack
7. Understand why layering is important in designing internet structure
8. Draw the internet structure

Week 9: KR Ch. 1.4

1. Describe and explain sources of packet delay through the network
2. Understand the difference between propagation delay and transmission delay
3. Describe the causes of processing delay
4. Compute total nodal delays
5. Compute effective throughput between two end systems
6. Understand how traceroute and ping works, as well as their applications

Week 10: KR Ch. 8.1 - 8.5

1. Understand what are the possible security issues in the network / internet
2. Describe desired security properties and their applications
3. Understand basic cryptography techniques
4. Explain concepts of symmetric and asymmetric keys, as well as their pros/cons
5. Explain applications of keys: confidentiality, authentication, integrity, authentication protocol and design issues, as well as secure emails
6. Explain possible attack scenarios in each of the application in point (5) above
7. Describe how public-key certification works

Week 11: KR Ch. 2.1, 2.5

1. Understand how internet naming and addressing works
2. Understand the difference between mnemonic names and IP addresses
3. Describe what is the DNS (domain name system) and its purpose
4. Understand the DNS as a distributed naming infrastructure
5. Describe and understand design principles of the DNS infrastructure and query
6. Explain different types of DNS records
7. Explain the purpose of DNS caching and how it is done as well as maintained over time
8. Understand how dnslookup and dig works, as well as their applications

Week 12: KR Ch. 2.2, 2.7

1. Understand how basic socket programming works
2. Explain how UDP and TCP works
3. Describe the main differences between UDP and TCP, as well as their applications (pros/cons)
4. Understand client-server model of network applications
5. Understand how web server works and the HTTP
6. Explain how cookies store user-server state and its function
7. Analyze how web caching impacts performance
8. Understand how wireshark packet sniffer and analyzer works and its application