

3150 - Operating Systems

Eric Lo

0-Course Info

Why we have to learn OS?

- If you want to work for MS Windows, RedHat, Vmware, VirtualBox
- If you want to work for Nvidia, HP
 - Writing drivers
- If you want to work for Google, Facebook
 - *Interview Question: “Why the maximum file size in an USB (FAT32) is 4GB?”*
- If you want to be (or avoid) a hacker
- If you want to be a system administrator
 - 20-80 system admin vs. programmer
 - Who has a higher salary?!
- If you need to design a new system named vLLM

What's an Operating System?

- Is your tiger mom
 - Let you focus on your homework by freeing you about
 - Scheduling
 - Won't conflict with your siblings
 - Never waste your time
 - Where your books are
 - Many details behind



WEEKLY TIMETABLE			
Time\Date	Monday to Friday	Saturday	Sunday
5:00	Wake up		
6:00	Reading ancient Chinese literature for 1 hour		
Morning	At school	Piano course	Ink brush calligraphy course
Afternoon	At school	14:00-15:00 Go game training	14:10-15:00 Mathematical Olympiad training
		15:40-16:10 Taekwondo	15:40-16:10 Swimming training
19:00-19:30	China Central Television Evening News		
20:00-21:00	English course	Latin dance training	Practicing ink brush calligraphy
21:00-22:00	Finishing homework	Finishing weekend homework	Finishing 5 pages of exercises for mathematical Olympiad
23:00	Sleep		

What's an Operating System?

- Is a software
 - Let you focus on your homework by freeing you about
 - Scheduling
 - Each application gets a CPU share
 - Never waste your CPU
 - Where your files are located on which “sector” your HD
 - Many details behind



History of OS



- Before 1945
 - Specialized purpose computer
 - Program logic was **hardwired** in a computer
- 1945
 - John von Neumann proposed a “**stored-program**” **general purpose computer**
 - A **core set of instructions**
 - Instead of hardwired program
 - Programs are formed by **sequence of instructions**
 - Keep the programs in the storage
 - **Human Operators** load that program from the storage and the computer executes the instructions



Around 1950 – Batch System / Mainframe

- People submit programs
- Keep the programs in the storage
- **Operating System** loads those programs **one-by-one** from the storage and the computer executes the instructions

Early 1960s – Time Sharing

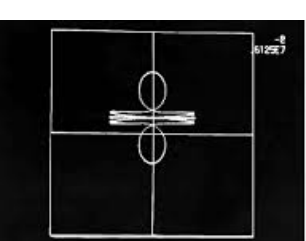
- The operating system gives each program in the queue a **time slice** in a rotational manner
 - Also introduce **priority** to jobs
- So that, to all users who have submitted jobs
 - they all have progress
 - instead of waited for 15 days and then halted by a bug



Early 1970s - UNIX



- A few newly hired engineers including Dennis Ritchie and Ken Thompson at AT&T's labs were unable to use the expensive mainframe to do their work because they **were not senior** enough
 - Only given a PDP-7 machine that was okay for playing a game called Space Travel
 - US\$72,000
 - 9KB memory
 - They developed their own OS for PDP-7
 - UNICS
 - » Uniplexed Information and Computing Service



Early 1970s - UNIX

- AT&T agreed to release UNIX as an open source operating system for **academic use**.
- Ken Thompson, one of the originators of UNIX and C, took UNIX to





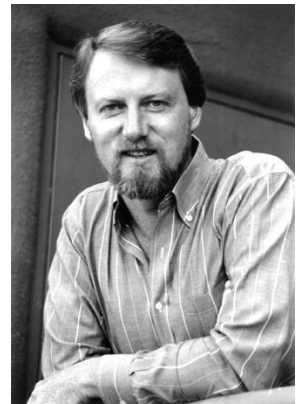
- Students loved it, improved it, leading to the world famous Berkeley Standard Distribution (BSD) form of UNIX

POSIX

- BSD UNIX quickly spread throughout the academic world (because of free for educational)
- But after AT&T settled some lawsuit, it decided to stop open-source
 - Leading to several other UNIX which is free for educational
 - E.g., FreeBSD, OpenBSD
- Afterwards, since so many UNIX variants available
 - each had some subtle differences
 - applications on top may not be portable to each different UNIX
- So, IEEE developed a standard of UNIX:
 - **POSIX**: defines a minimal system call interface that all UNIX-claim system must conform to

>1974s – Personal Computer

- With the development of LSI (Large-Scale Integration) circuit
 - Price dropped
 - Size dropped
- Intel 8080
 - First general purpose 8-bit CPU
- In early 1980s, IBM made the first IBM PC
 - Contacted Gary Kildall to make the OS
 - Gary Kildall made the probably the worst business decision in history



1980s

- Then IBM contacted Bill Gates
 - who had made a BASIC interpreter
 - bought DOS (Disk Operating System) from a Seattle Computer company
 - Grabbed the DOS developer to customize it
 - Sold it to IBM as MS-DOS

```
Starting MS-DOS...  
C:\>_
```



1980s



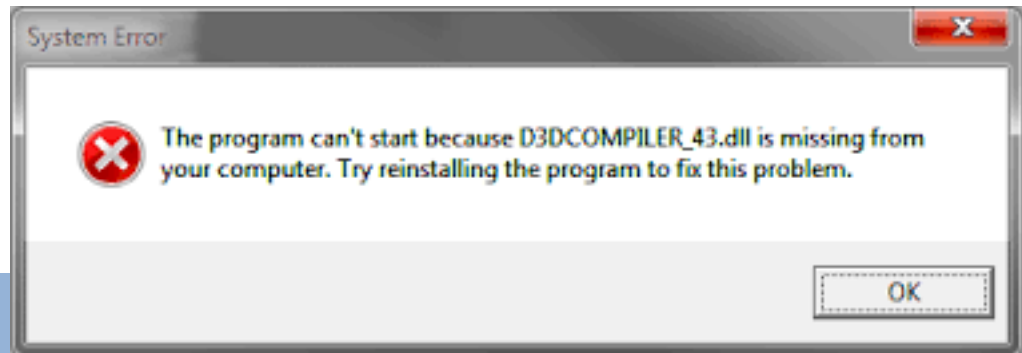
1990s - Linux

- UNIX* was not free when not for educational purpose
- In 1991, Linus Torvalds, a university student in Finland, developed Linux without using any UNIX code.
- Totally open license model
 - Free even for commercial



C Sanity Check

- What's the size of a pointer?
- Why there is .h (header file)?
- What's the size of a char?
- What is the usage of static?
- What is a function pointer?
- What is the difference between declaring and defining a variable?
- What's .DLL?



After this course, you shall know...

Why I get segmentation fault?

How can FBI (you) recover a file deleted by a terrorist (and emptied the trash) on his computer?



Piazza

- Lecture notes, assignment, source codes demo during the lectures, exercises, announcement...
 - Discussion Board
 - Healthy questions are welcome
 - E.g., “why that can cause segmentation fault?”
 - Unhealthy questions are **discouraged**
 - E.g., “**how many scores do I need in exam in order to pass?**”
 - E.g., **Questions that come from skipping (or sleeping in / daydreaming during) the lectures/lab**
 - E.g., Problems arise from you insist to test your assignments on Windows
 - Private messages through Piazza
 - No Email!
 - Private Message usage example:
 - Send us your answer of past mid-term/final for comments
 - **Private Questions that should be public get low/no priority**

Assessment (**tentative**)

- Continuous Assessment (40%)
 - **1 Warm-up Assignment A0 (1%)**
 - 3 Programming Assignments (24%; @8)
 - A1: Shell (@8)
 - A2: Multi-Thread Scheduler (@8)
 - A3: File System (@8)
 - 1 Mid-Term (15%; **20 Feb**)
 - Absence with strong justification (e.g., medical proof) will give you:
 - CA: Final = 25 : 75 instead of 40 : 60
 - Final (60%)
 - 10% Bonus
 - Bash Script (5%)
 - To encourage student participation (5%)
 - $\frac{\text{Your \# of "endorsed answers"}}{\text{\# of Piazza's "endorsed answers"}} \times 5\%$
- FYI:
Last offering, 50 endorsed answers

Lab



Lab 1 Users

```
linux> ls -l
```



Labs 2 – 3 Advanced users

```
#!/bin/bash
```



Labs 3 - 10

System Programmer

```
fork(), mmap()
```



Labs 8 – 13

Hacker, Kernel Level programmer, Google, ..

Lock-free programming, loop unrolling, Kernel I/O

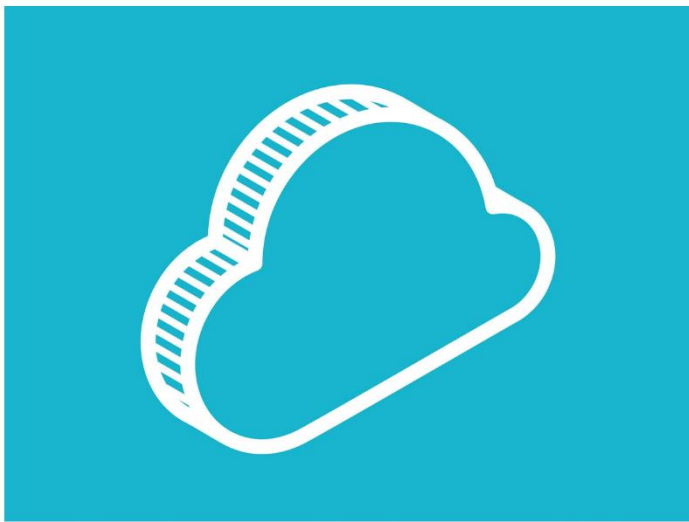
Lab

Lab		Programming Difficulty	Concept Difficulty
1	Linux		*
2	More Linux (scripting) B	*	*
3	System Programming tool (e.g., make)	*	*
4	Process (fork, wait, exec, etc.) A1	**	*
5	Pipe and Signals	**	*
6	Memory (Stack smashing attack)	*****	***
7	Threading A2	**	**
8	Synchronization I	***	*****
9	File Systems A3	*****	**
10	Advanced C for Kernel Programming	*****	*****
11	Memory Mapped I/O	****	****
12	Synchronization II	*****	*****
13	Linux Kernel Hacking (Optional)	*****	**

Why Linux?

KLINT FINLEY BUSINESS 09.18.15 4:59 PM

WHOA. MICROSOFT IS USING LINUX TO RUN ITS CLOUD




GETTY IMAGES

MICROSOFT HAS ADMITTED to something that used to be unthinkable: using Linux to run some of its own operations.

In a [blog post](#) on Thursday, Microsoft Azure networking principal architect Kamala Subramaniam explained how the company developed a new software system, dubbed Azure Cloud Switch, for running the networking gear that Microsoft's cloud service depends on.

Announcing SQL Server on Linux



SQL Server ❤️ Linux

Posted March 7, 2016 By **Scott Guthrie** - *Executive Vice President, Cloud and Enterprise Group, Microsoft*

[f](#) [in](#) 1155 [t](#)

It's been an incredible year for the data business at Microsoft and an incredible year for data across the industry. This Thursday at our [Data Driven](#) event in New York, we will kick off a wave of launch activities for SQL Server 2016 with general availability later this year. This is the most significant release of SQL Server that we have ever done, and brings with it some fantastic new capabilities. SQL Server 2016 delivers:

- Groundbreaking security encryption capabilities that enable data to always be

Programming Assignments

- All test cases are given
- Grader is given
- No unseen test cases
- What scores you see (in our course VM) is what you get
 - So, test your program on our course VM before you ask
 - So, incomplete/un-compilable code won't get scores

Programming Assignment Deadlines

You actually
get **instant**
feedback from
our test suite

Assignment	Deadline	Late Penalty
A0 (Warm-up) 1%	<u>23 Jan, 11am</u>	100%
A1 (Shell) 8%	3 weeks after release	< Apr 24, 11am 20% ≥ Apr 24, 11am 100%
A2 (Scheduler) 8%		
A3 (File System) 8%		
B (Shell Script) 5%	3 weeks after release	100%

– After Apr 24, 11am

- Absolute final deadline (*unless A3 is released late)
- TA will fetch from Gitclassroom

Reminder

- OS course is **tough**
 - And it should be
 - Programs must pass test cases to gain scores
- Suggestion: **don't take too many major when you take this course**
- Exam cover **EVERYTHING** in this course

Use of AI in the assignments

- Allowed AI tools
 - Any AI model, e.g. GPT-4o, that causes no ethical issues.
- Acknowledge your use of AI tools in `ai_declaration.txt`

Acknowledgement, References, and Disclaimer

- Source codes on the notes, mid-term, exam are mostly **pseudocode**
 - Not directly compliable
- Some notes and examples are extracted from Dr. TY Wong's previous offering
- Google Images, Stackoverflow

Reference

- Google / Youtube
- GPT!
- Past mid-term (with answers) and finals are given
 - Note: the schedule/content of each term varies, so just take those **as a reference** only

It helps a lot if you switch to use Linux as your main OS (instead of via a VM), at least during this course