

# Introduction to Information Systems

– Understanding the digital world

**7** Operating System and Software Systems

Liang Zhao

ILA, Doshisha University

12001102, Fall, 2023



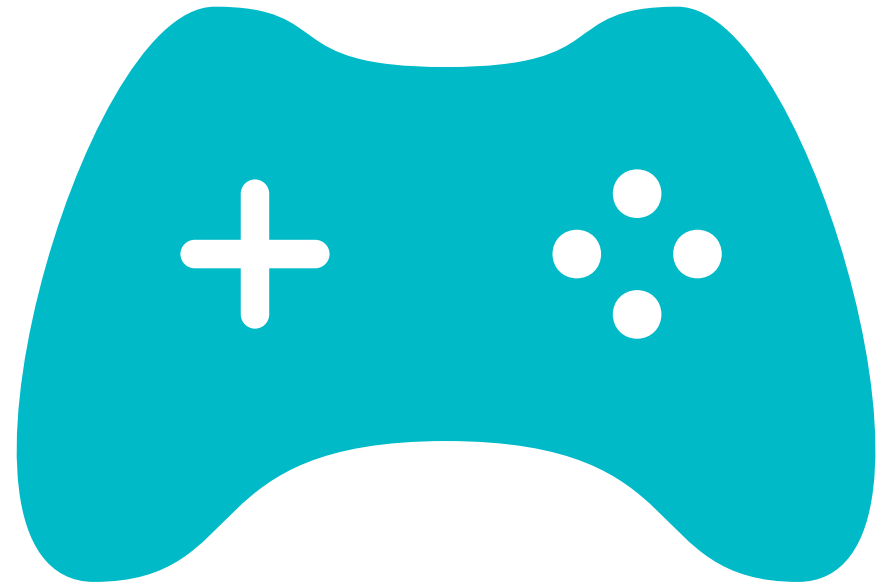


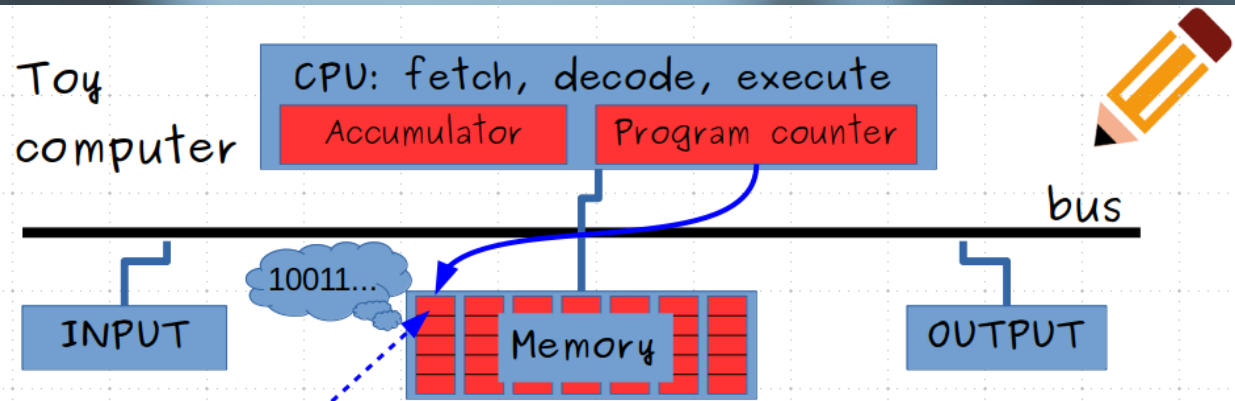
# Today's schedule

- Review of the Scratch game (5')
- BIOS (10')
- Operating System & Software (55')
- Mini test (15')
- Information and homework (5')

# Review of the game

- Reviewers: three

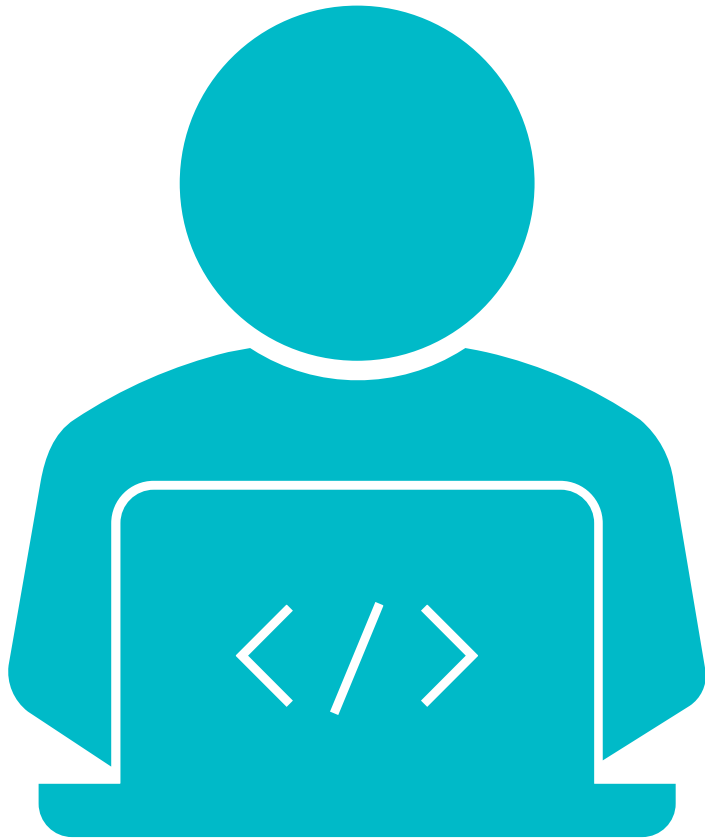




label	instruction	description
	get	get a number from keyboard into accumulator
L	print	print contents of accumulator
	load Val	load accumulator with Val (Val unchanged)
	store M	store contents of accumulator into memory location called M
	add Val	add Val to contents of accumulator (Val unchanged)
	sub Val	subtract Val from contents of accumulator (Val unchanged)
	goto L	go to instruction labeled L
	ifpos L	go to instruction labeled L if accumulator is $\geq$ zero
	ifzero L	go to instruction labeled L if accumulator is zero
	stop	stop running
M	Num	before program runs, set this memory location (called M) to Num

# BIOS

- **Basic Input/Output System (BIOS):** firmware used to perform hardware initialization during the booting process (power-on startup), and to provide runtime services for operating systems and programs. (<https://en.wikipedia.org/wiki/BIOS>)
- In short, the firmware that connects hardware and software.
- <https://www.youtube.com/watch?v=D1R2ttrvbdI> (6')



# OS (Operating System)

With OS, we don't need to write millions of lines of code by ourselves. We only need to know how to use the OS.

Crash course -> Computer Science #18 (14')

<https://www.youtube.com/watch?v=26QPDBe-NB8>

# What does an OS do?

---

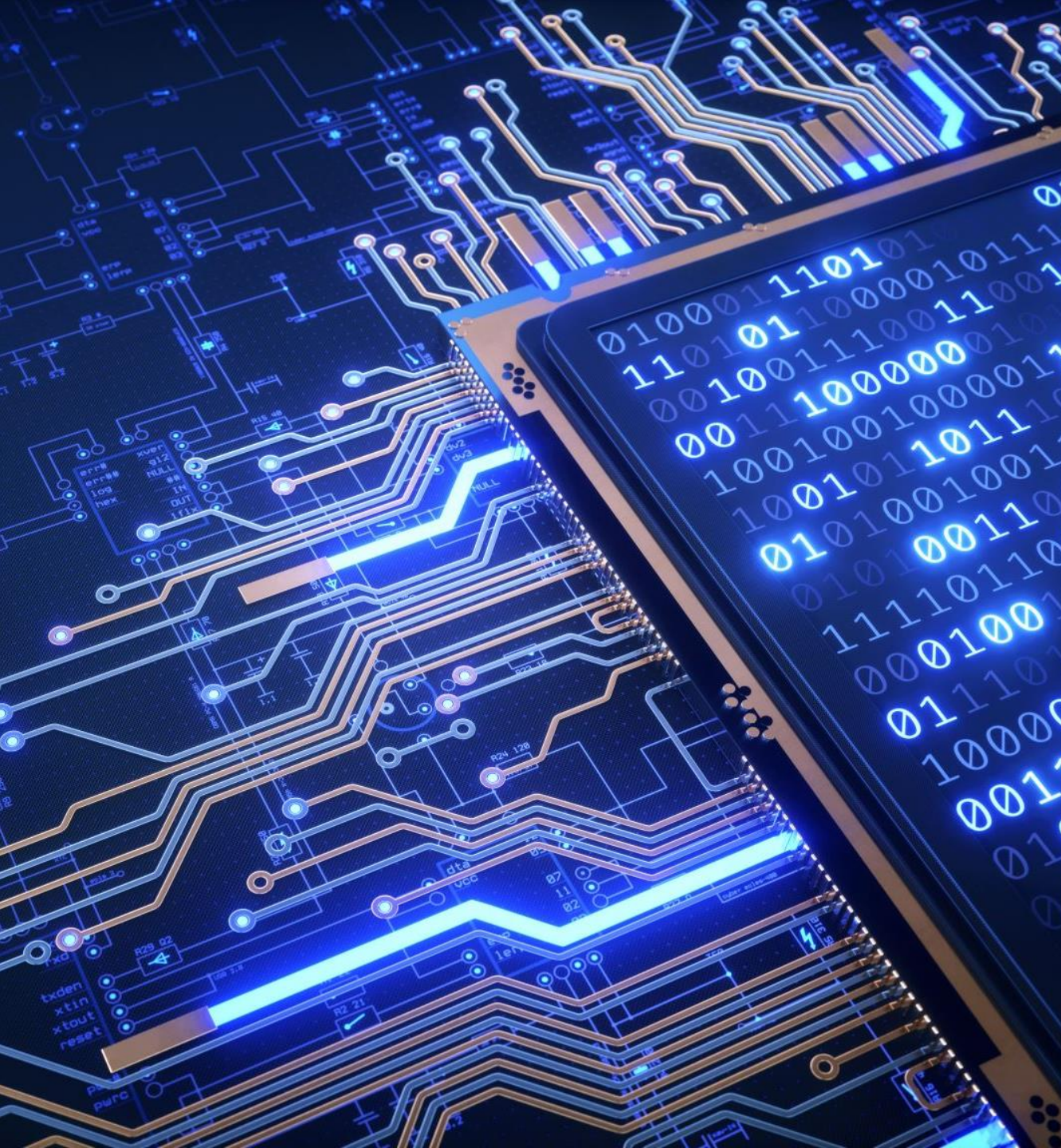
**CPU (task  
management)**

**Disk (HDD, SSD,  
etc) and file**

**RAM (memory)**

**Devices (monitor,  
keyboard, mice,  
printer, etc)**





# **CPU (task management)**

**Single-task (DOS, etc)**

**Multi-task (Unix/Linux, Windows, Mac OS, etc)**

**See the previous video (Crash course -> Computer Science #18)**





# Memory (RAM) management

Loads programs and data into memory.  
Swaps them to disk when memory is low.  
Protects the programs from interfering.





# Disk and file management

- File System (FAT, FAT32, NTFS, ext4, APFS, etc)
- Directory (folder): special container file
- Executable files (Word, Photoshop, etc)
- Documents (txt, doc, jpg, mp3, html, etc)
- System files (lib, sys, etc)
- Extension distinguishes types: doc/docx -> Word document, jpg -> JPEG file (editable by Photoshop etc), exe -> executable file, ...

Crash Course -> Computer Science #20 (12')

<https://www.youtube.com/watch?v=KN8YgJnShPM>

# Path: location of a file



**Absolution** path: e.g., `C:¥Users¥liang¥Desktop¥book.docx` (Windows, starting with the **drive**) and `/home/liang/Desktop/book.tex` (UNIX/Linux/Mac, starting with the root `"/"`)



**Relative** path: path that is related to the **working directory** (W.D., directory where we are working with). E.g., if W.D. is `C:¥Users¥liang`, then `Desktop¥book.docx` means `C:¥Users¥liang¥Desktop¥book.docx`.



URI extends this notation with protocol and server name: e.g., (where **protocol** = https, **server name** = `ila.doshisha.ac.jp`) `https://ila.doshisha.ac.jp/en/index.html`



# Shell

User <-> OS interface software

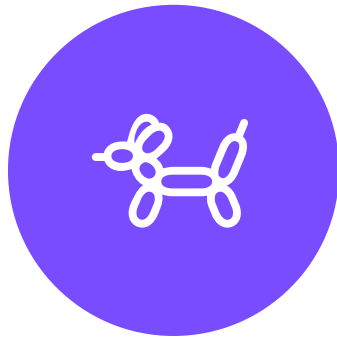
Crash course -> Computer Science 22 (11')

<https://www.youtube.com/watch?v=4RPtJ9UyHS0>



# Homework

---



## WATCH VIDEOS

Watch the movies mentioned so far if you have not  
(You are not expected to understand everything)



## READ CHAPTERS 1-6 (IF YOU HAVE NOT)

# Appendix: Advanced topics

---



**Device driver:** program for a special hardware. Ex: printer drivers provide detailed control (two-sided printing, etc).



**System call:** function provided by the OS to apps. Ex: input, drawing on the display (DirectX, OpenGL), network function, etc.



**Memory management:** Crash Course -> Computer Science #19  
<https://www.youtube.com/watch?v=TQCr9RV7twk>



**Other Uses:** Linux, FreeBSD, Android, etc