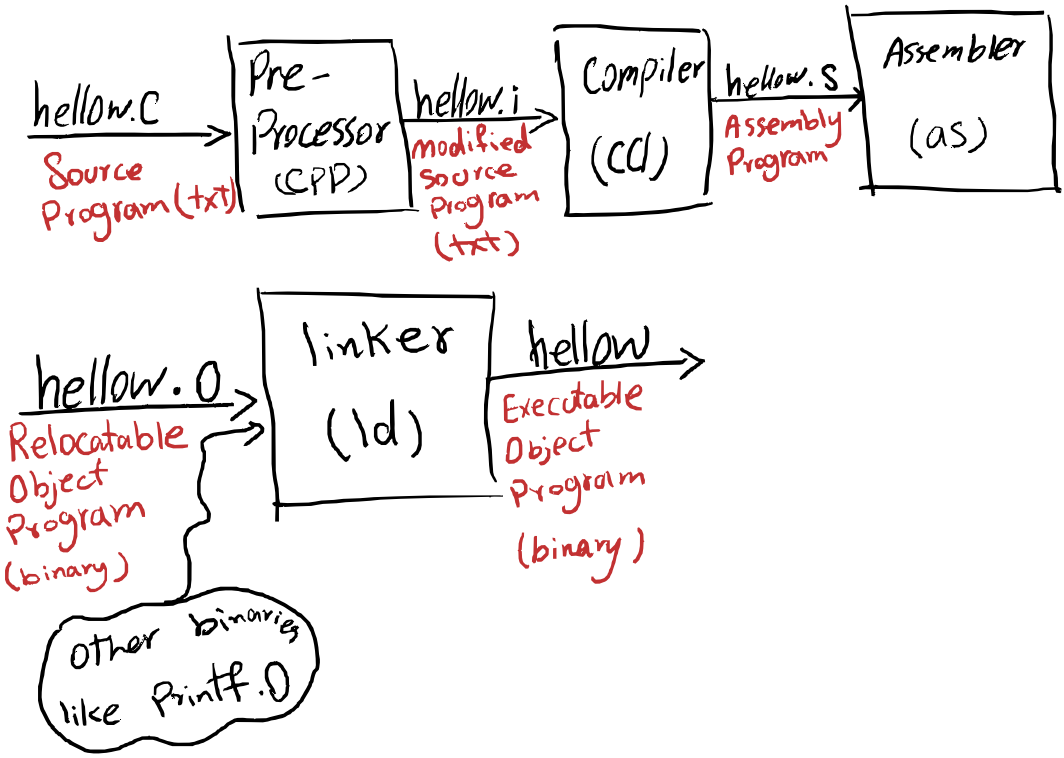
At the end → isn't it cool

How bunch of 0s 1s are interpreted

1. Computer Architecture e.g. big / little endian
2. OS e.g. \r\n vs. \n
3. Application Software

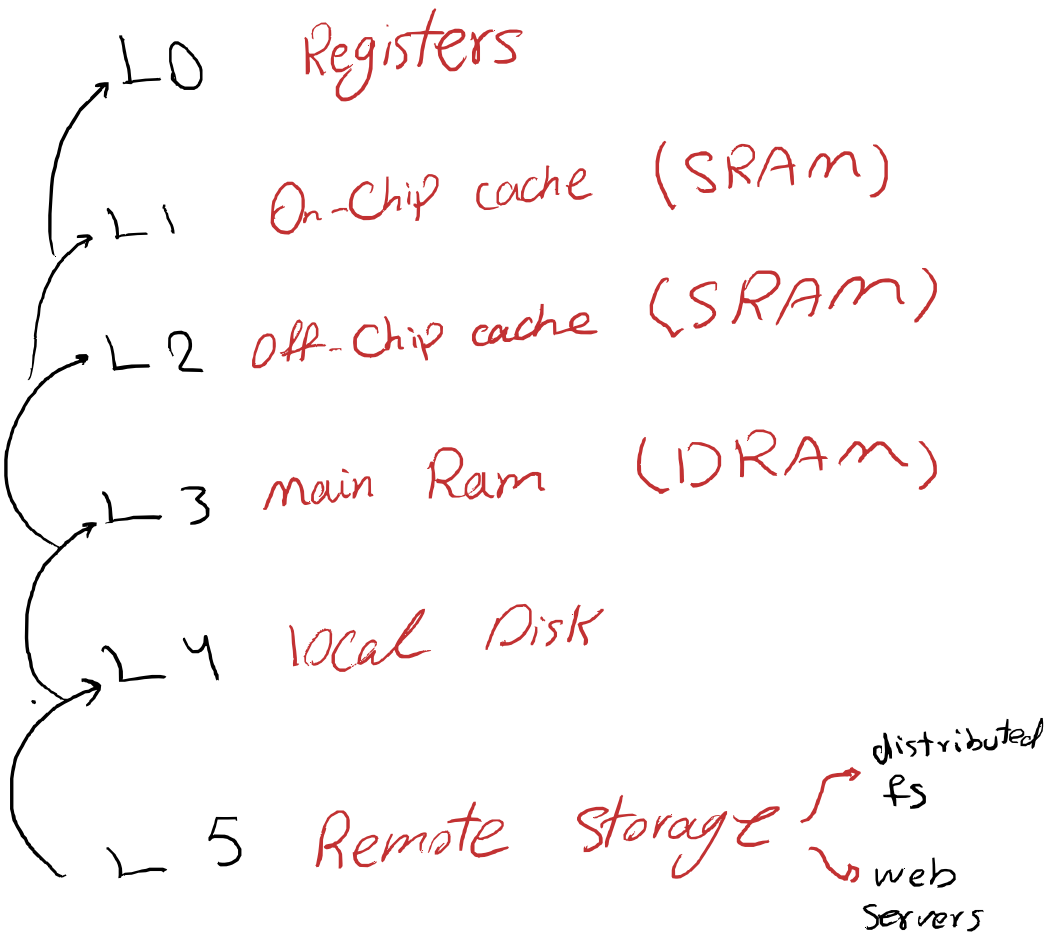
Compilation Process

C/C++



Java

Storage Hierarchy



? How we can exploit this hierarchy to improve efficiency ?

what does OS do?

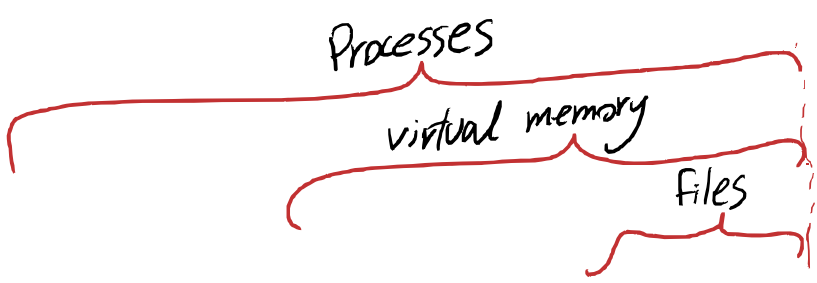
① Protect the hardware from misuse by Applications.

② Provide a uniform mechanism for manipulating various ~~complicated~~ low-level hardware.

How?

★ Same old Abstraction

"This is the way"



Processor	memory	I/O devices

Process → the program thinks its the only Process (isolation)

↳ Can have multiple executions

thread

↳ can run concurrently

files → R/W → network, disk
files → R-only → web-cam, Keyboard
files → W-only → Printer

- Uniform interface for varied I/O device.
- Abstracts away all the underlying details on how data read/written.
- Program written in one system can easily be ported.