



* git 원격저장소를 local로 복제하기

local (c:\users\bitcamp\git\)



clone



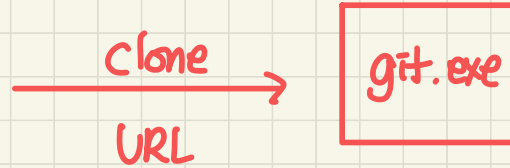
https://github.com



← 공유
폴더

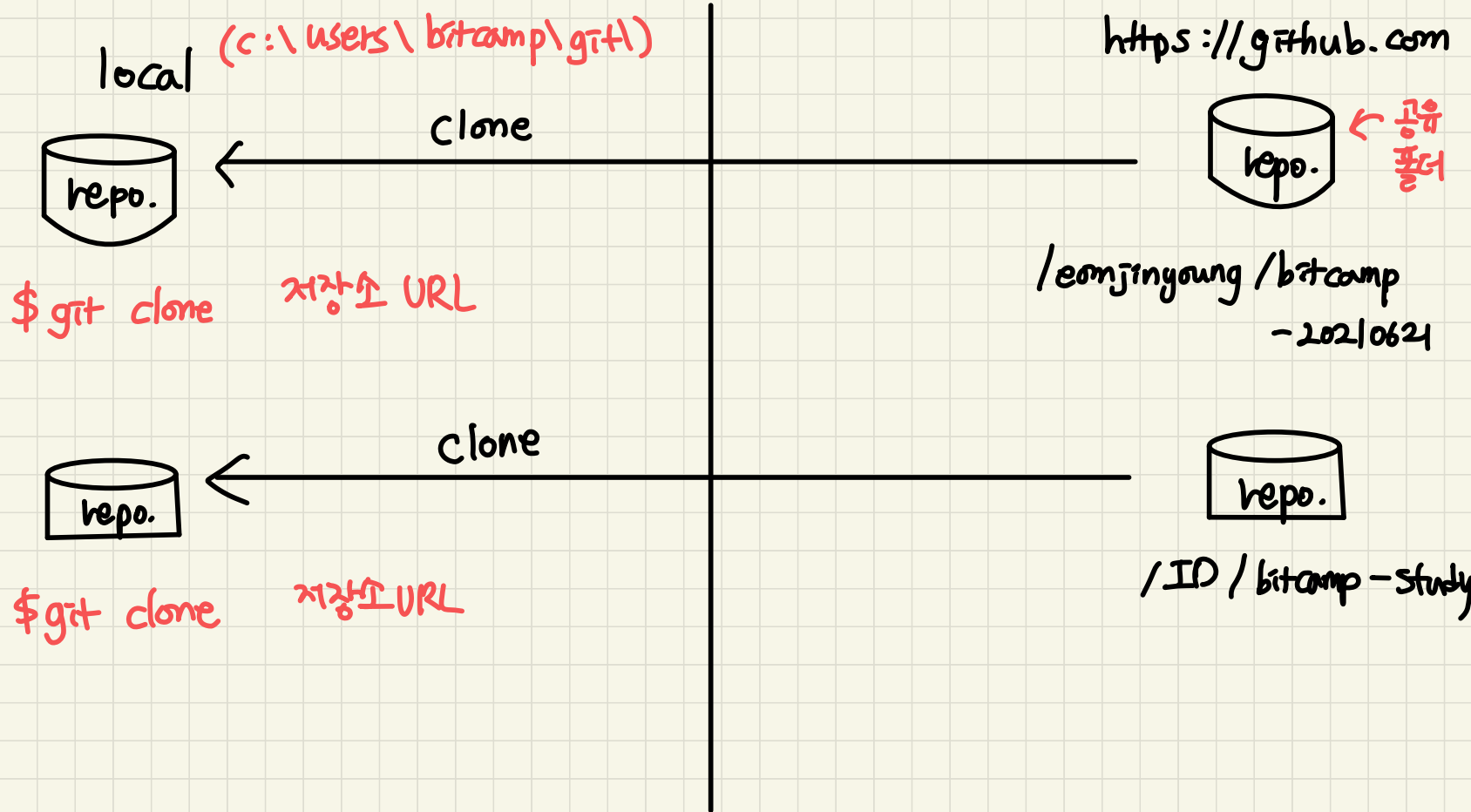
/eomjinyoung/bitcamp
-20210621

\$ git clone https://github.com/
eomjinyoung/bitcamp-20210621
juricho-jay/bitcamp-study
<저장소 URL>



bin 저장소, binary 약자

* git 원격저장소를 local로 복제하기



* git 로컬 저장소에 파일 백업하기

[working Directory] 추가
변경
삭제 된

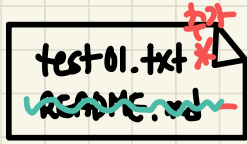
README.md
test01.txt

좋은 커밋 메시지만? 기자의 규칙?!
→ depends on each firm

① 백업 명단 작성



② 백업 수행



bitcamp-study \$ git add .

\$ git commit -m "백업 내용 + 이유"

message



변경된 파일을 등록

→ 추가/삭제/수정

* git 로컬 저장소에 백업된 파일을 서버 저장소에 업로드

[local repo.]



upload

[server repo.]



bitcamp-study

bitcamp-study \$ git push

_____ : username

_____ : token

* 백업 100번하고 push 1번해도

100번 백업 기록은 모두 기록됨 (버전 관리 시스템)

* 서버 저장소에서 변경된 파일 가져오기

[local 저장소]



`$ git pull`



[server 저장소]



`/eamjinyoung/bitcamp-20210621`

* 수업 도구 준비 (계속)

③ VS Code (편집기) 설치

↳ 소스 파일 편집

- 명령어를 작성한 파일
- 실행 파일을 만들 때 사용

↳ HTML / CSS / JavaScript 파일 편집

↳ 기타 설정 파일 편집

↳ Microsoft에서 다운로드
Pref → Settings

* font-size
↓
{ Ajgk 높이를 의미

④ C/C++ 컴파일러 설치

↳ Window gcc 설치

⇒ MinGW

⇒ MinGW-w64 posix seh

→ C:\tools에 압축풀기

tools/mingw64/bin

↳ gcc 컴파일러가 설치된
폴더의 경로를 OS 환경 변수

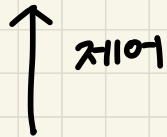
PATH에 등록

↓

C:\tools\mingw64\bin\gcc --version
gcc 컴파일러를 실행할 때
경로를 적을 필요가 없다.

↳ 폴더 등록

컴퓨터



제어

S/W ⇒ OS

A → OS

B → OS'

C → OS''

compiler 필요...

PAST



사용자 = 개발자

PRESENT

개발자 V

명령서
번역

A ← DOS

A ← DOS

A ← DOS

실행파일
복사
복사

실행



UNIX → 프로그래밍 할 줄 아는

(라눅스, 맥)

사람을 위한 것이었기 때문...

compiler 내장

* 수첩 도구 설치 (계속)

⑤ node.js 설치

↳ javascript로 작성한 명령어를 실행

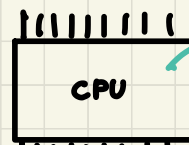
↳ nodejs.org에서 다운로드

mac OS → `$ brew install node`

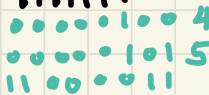
* 프로그래밍 - 프로그램 만들고 실행하기 <1
00000010 / <

1 → 4
2 → 5
3 → + (연산)

4?
+
5



메모리 필요

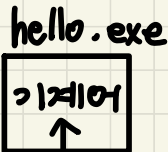


① 컴파일 방식

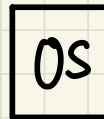
hello.c



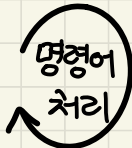
번역



loading



전달



CPU가 인식할 수 있는 = CPU instruction set

명령어

Compile

실행

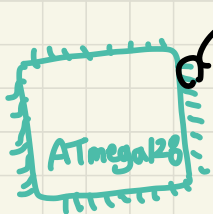
ADD → C6 → 11000110
Instruction set

정해놓은 규칙

Intel CPU → AMD?
Instruction set 호환?!

* 아두이노

명령어 Instruction



전기가 왔다갔다~

+, -로 처리

10101011 8 bit CPU

101011110001001 16 bit CPU

x2

32 bit CPU

x2

64 bit CPU

AVR Instruction Set

생각법 매뉴얼

어떤 규칙인지!

0 - Low Voltage

1 - High Voltage

엄연히 말하면
가계용은 아님

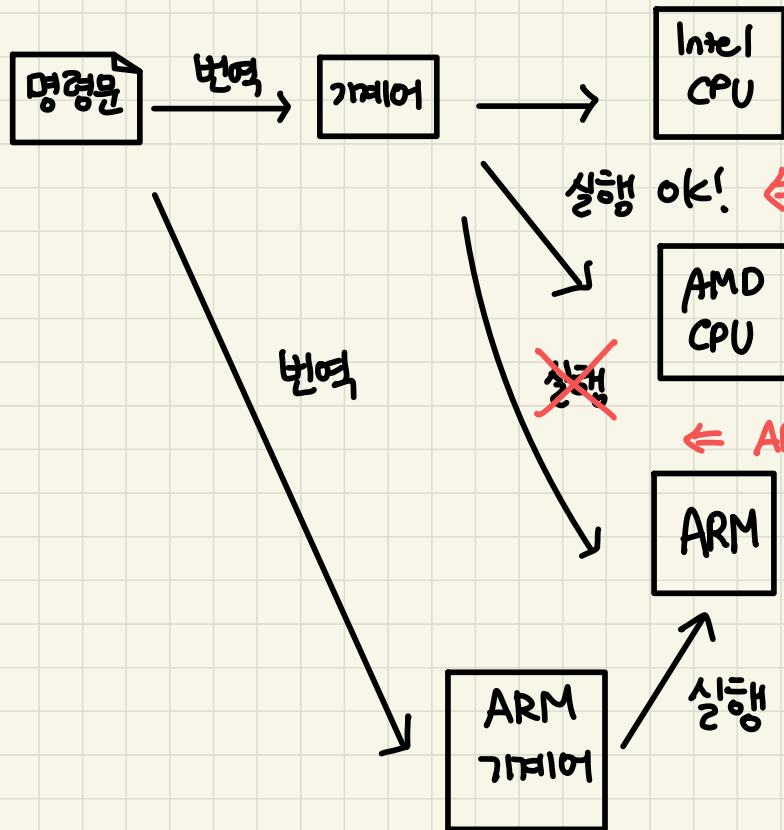
Assembly 언어!



Assembly compiler
필요

* 기계어와 CPU

CPU가 다른 편에 흐르는 규칙이 다름
명령문이 다름



AMD CPU Intel CPU 명령 규칙이 따라
동작하도록 설계되었다.

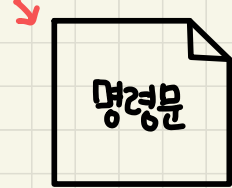
"Intel CPU compatible"
(호환)

ARM CPU는
Intel CPU와 규칙이 다르다.

번역을 해주는 소프트웨어
→ Compiler

* 기계어 만들기

Source 파일



Compiler \Rightarrow 예) gcc (windows용 \rightarrow mingw64)

\Downarrow
번역

(compile)



\rightarrow 특정 CPU에 맞는!

Object 파일

(목적) - Intel CPU에 들어갈지!

\$ gcc 소스파일명

아 \$ gcc -o 실행파일명 소스파일명

\$ gcc -o hello hello.c

\hookrightarrow hello.exe

명령문을 작성할 때 사용할 언어

"Programming" + "language"

\hookrightarrow 예) Java, C, C++

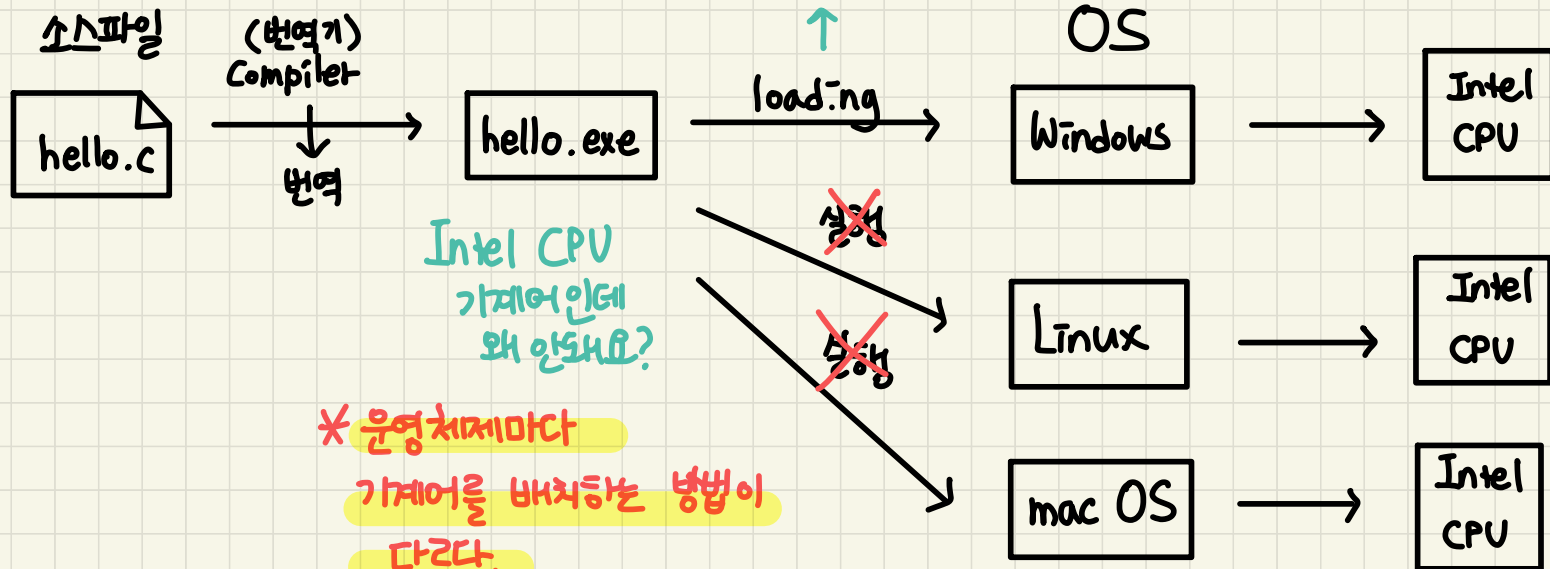
* \$ gcc -o 실행파일명 소스파일명

\$ hello.exe 실행

\$ hello 실행

\hookrightarrow Hello, World!

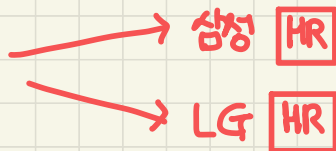
* 기계어와 OS (운영체제)



* 운영체제마다
가제어를 배치하는 방법이
다르다.

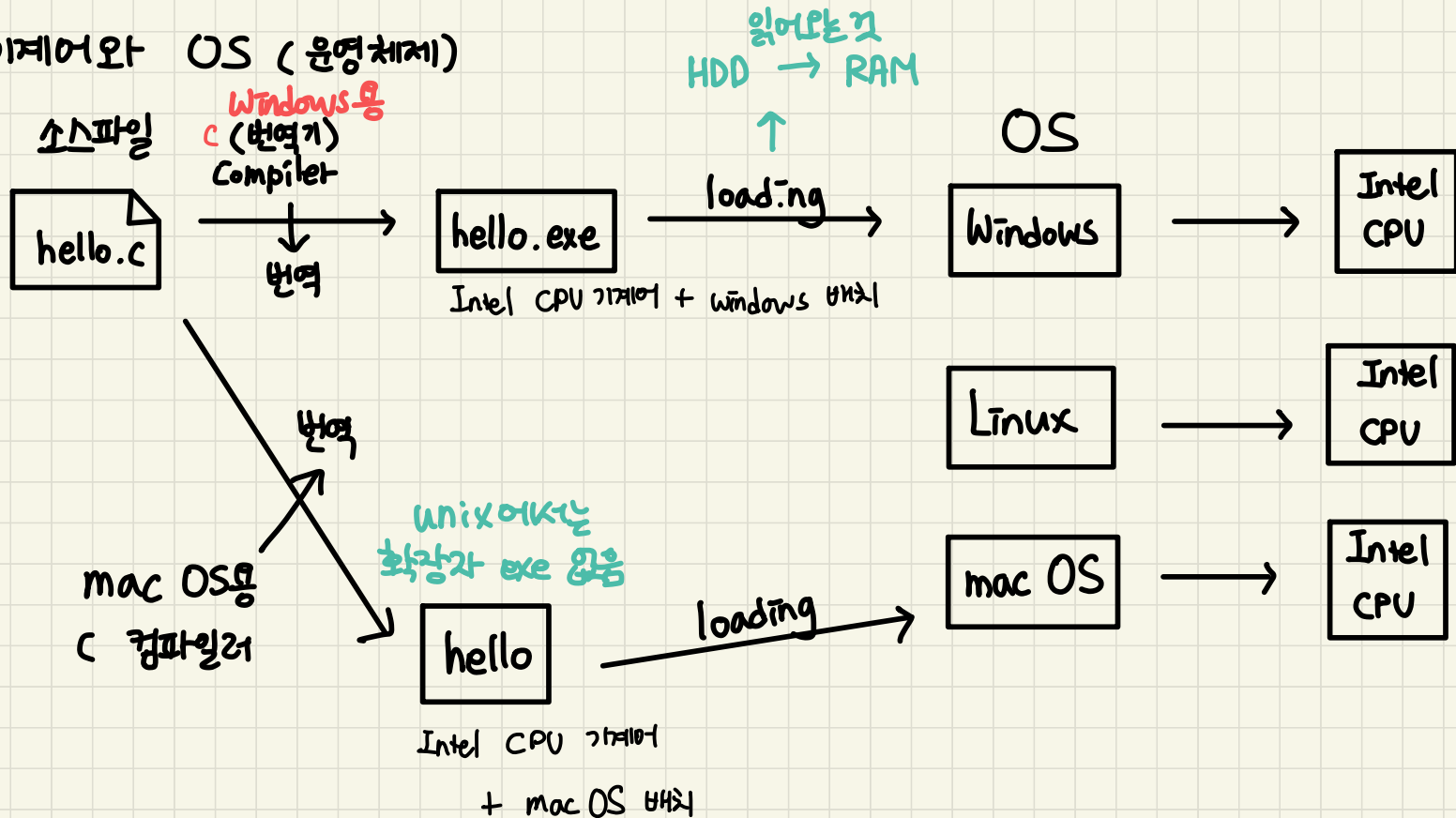


이력서



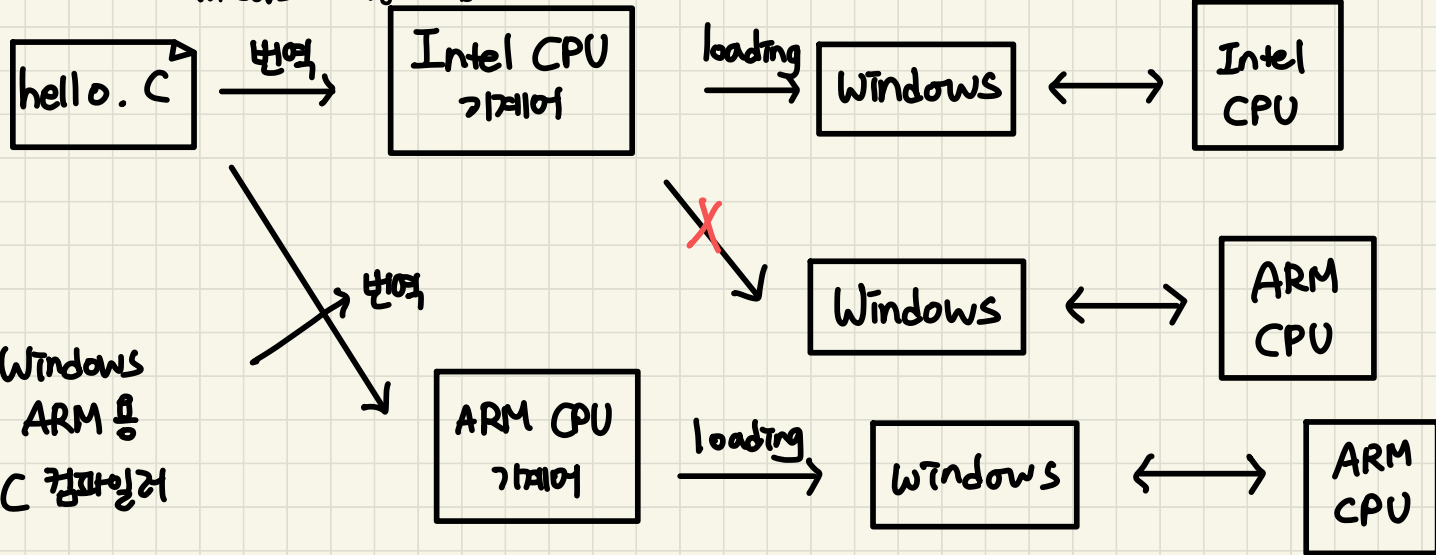
이력서
양식이 달라!

* 기계어와 OS (운영체제)



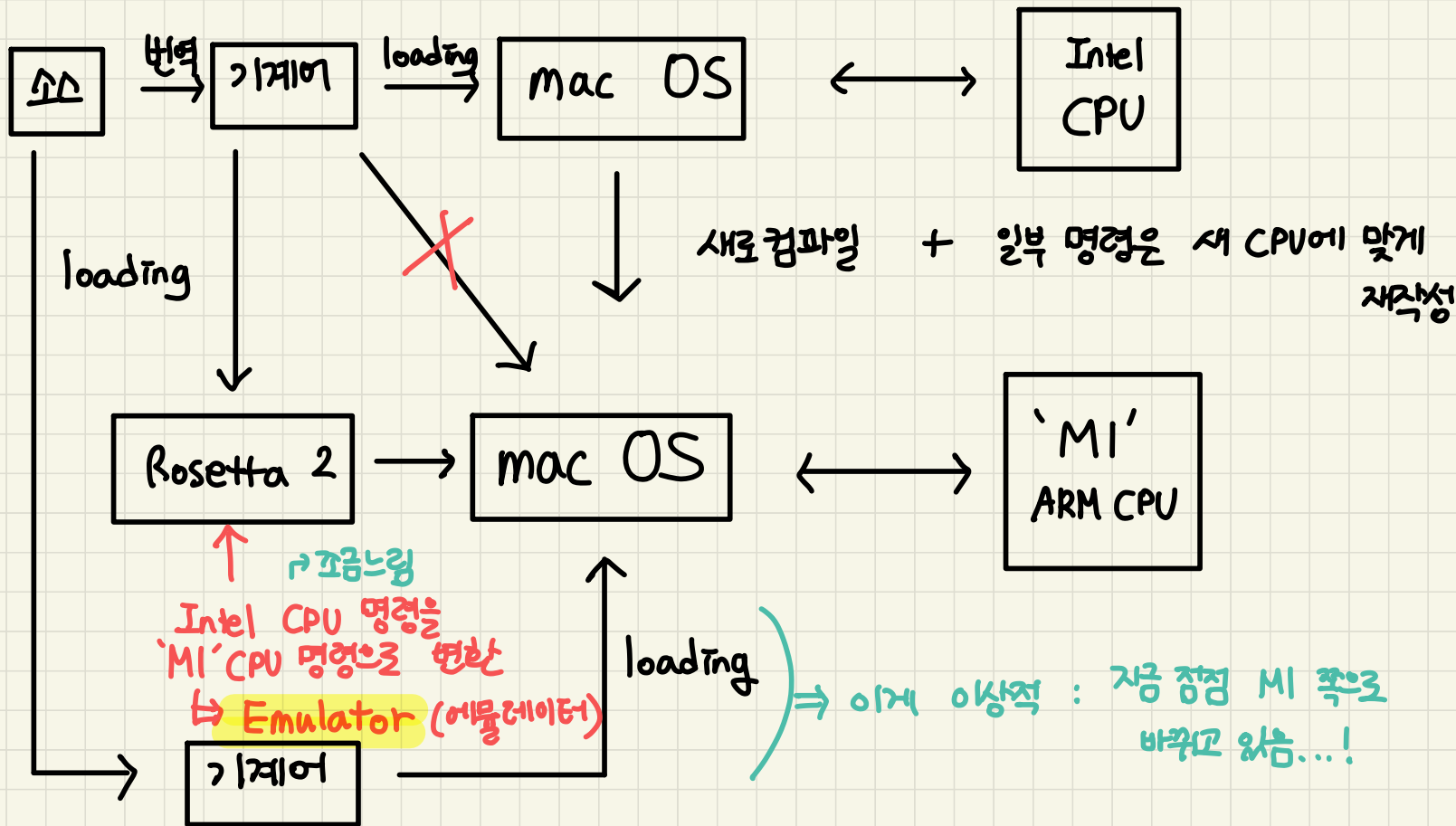
* 가제어와 OS

Windows Intel용 C 컴파일러



* mac OS의 상황

Emulator!!! CPU 명령 변환해주는 것!



② 인터프리터 방식 ex) JavaScript, PHP, Python 등

↳ 소스 파일의 명령을 도움이 프로그램을 통해 바로 실행

동역기 = interpreter

\$node 소스파일명
\$node hello.js
excel

* 기계어가
아니라서
도움이
프로그램이
필요함

