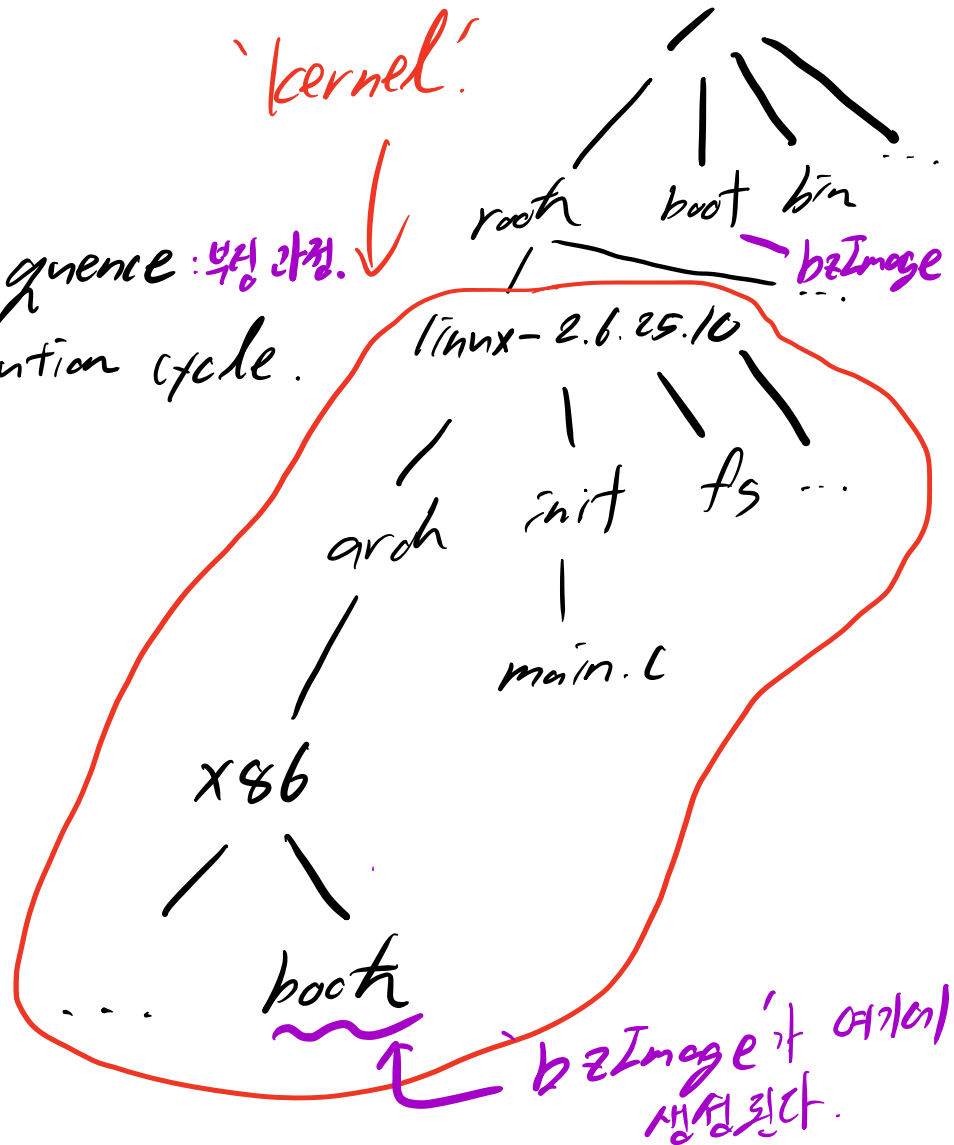- kernel Compile
  - Modify kernel
  - Compile
  - reboot
  - boot sequence :부팅 과정.
  - cpu execution cycle.

※ root : 사용자 홈 디렉토리

`kernel` ↓

root   boot bin
            ~bzImage

linux-2.6.25.10

arch   init   fs ...

main.c

x86

... boot
~~~
↑  bzImage'가 여기에
   생성된다.

. `Makefile`로 디렉토리 내 모든 C파일을
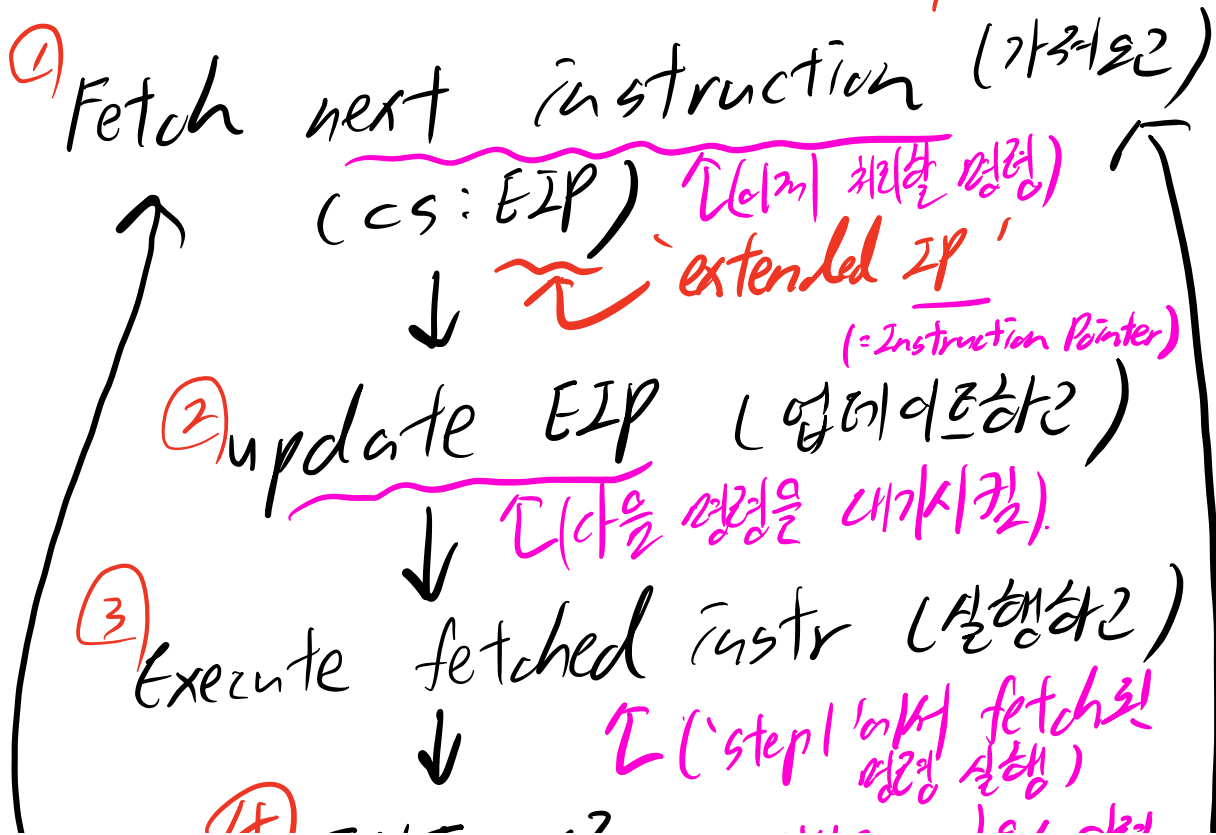  컴파일한다.
~~~
↑   ↑
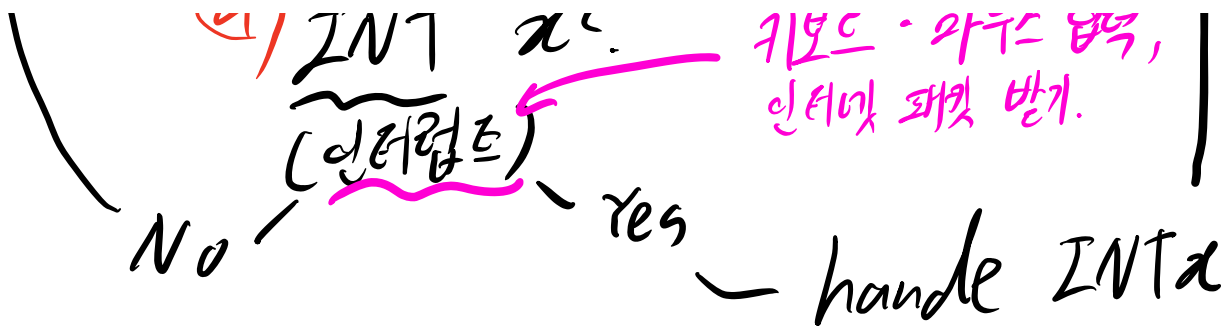   각 디렉토리 내 들어있음.
(해당 파일 내에. 디렉토리 내 모든. 내용을 컴파일하는 명령어

. # `make bzImage` ① : 현재, 하위 디렉토리 내 존재하는 모든 `makefile`을 찾고 이것을 ②두 실행한 후, 컴파일된 모든 파일들이 압축된 `bzImage` 실행파일을 생성한다.
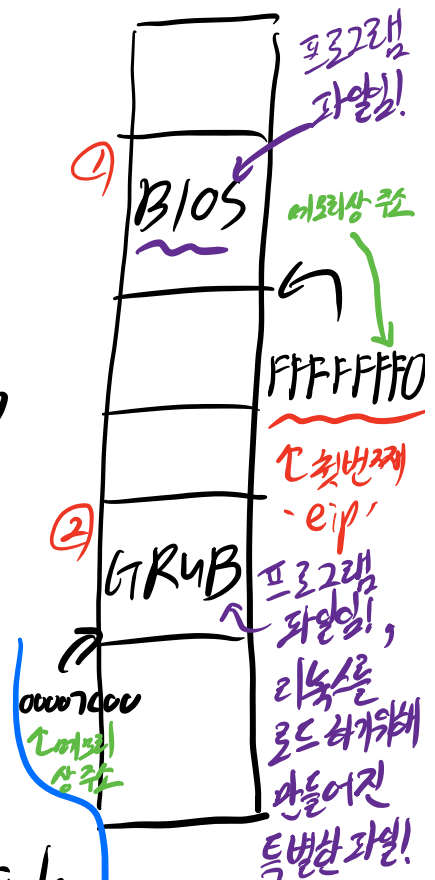
↑ `linux ~` 디렉에서 실행할 것!

Linux executable file.

── cpu는 멈추지 않는다.

✗ cpu execution cycle (cpu는 항상 이 cycle을 반복한다!)

① Fetch next instruction (가져오고)
(cs : EIP) └(이제 처리할 명령)

`extended IP`
(= Instruction Pointer)

② update EIP (업데이트하고)
└(다음 명령을 대기시킴).

③ Execute fetched instr (실행하고)
└(`step1`에서 fetch된 명령 실행)

④ ...

(예) INT x: 키보드·마우스 입력,
인터넷 패킷 받기.
(인터럽트)

No / Yes
hande INTx

X. boot sequence

power on
↓
CS=F000, IP=FFF0
↓
① BIOS
?
load 1st sector of boot disk

512byte

---

① BIOS ← 프로그램 파일!
메모리상 존재
FFFFFF0
첫번째 'eip'

② GRUB ← 프로그램 파일!, 리눅스를 로드하기위해 만들어진 특별한 파일!
00007000
메모리 상존
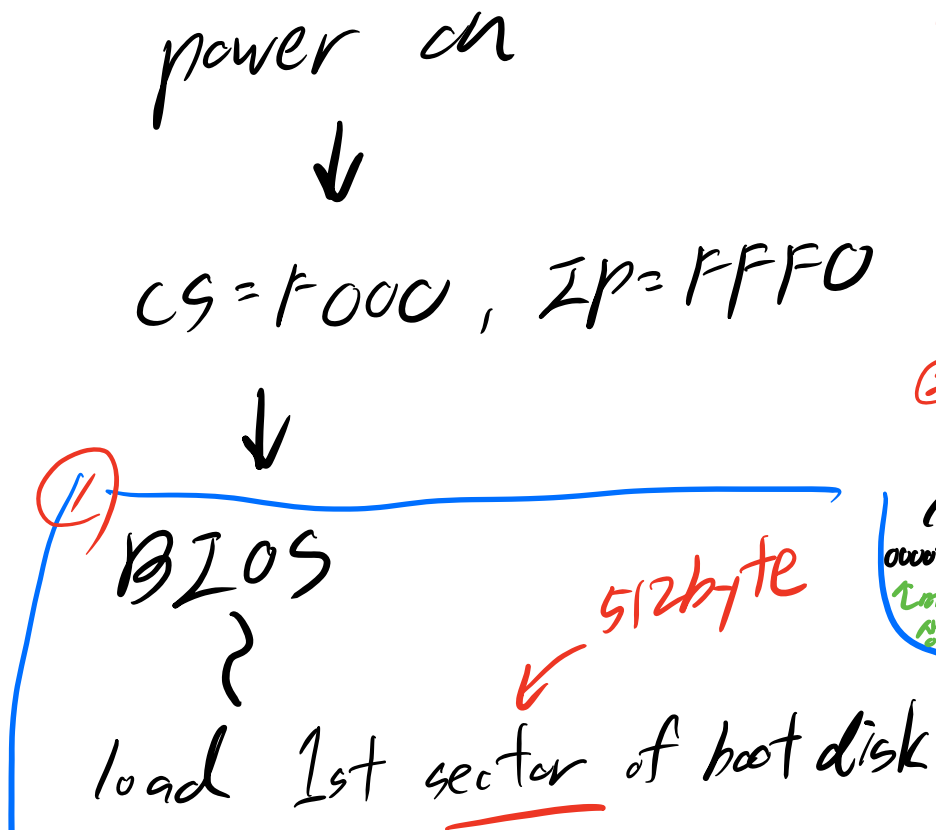
and JUMP there

② boot loader (`GRUB`)
⌐
⟹ `/boot/grub/bzImage`
⌐ 즉, GRUB이 이 파일을 실행함.

↳여기가 운영체제를 찾아서 실행시킴. 〉

③ Linux
⌐
start_kernel( ) ...

이 때, 메모리에 커널코드가 적재되고 실행된다.