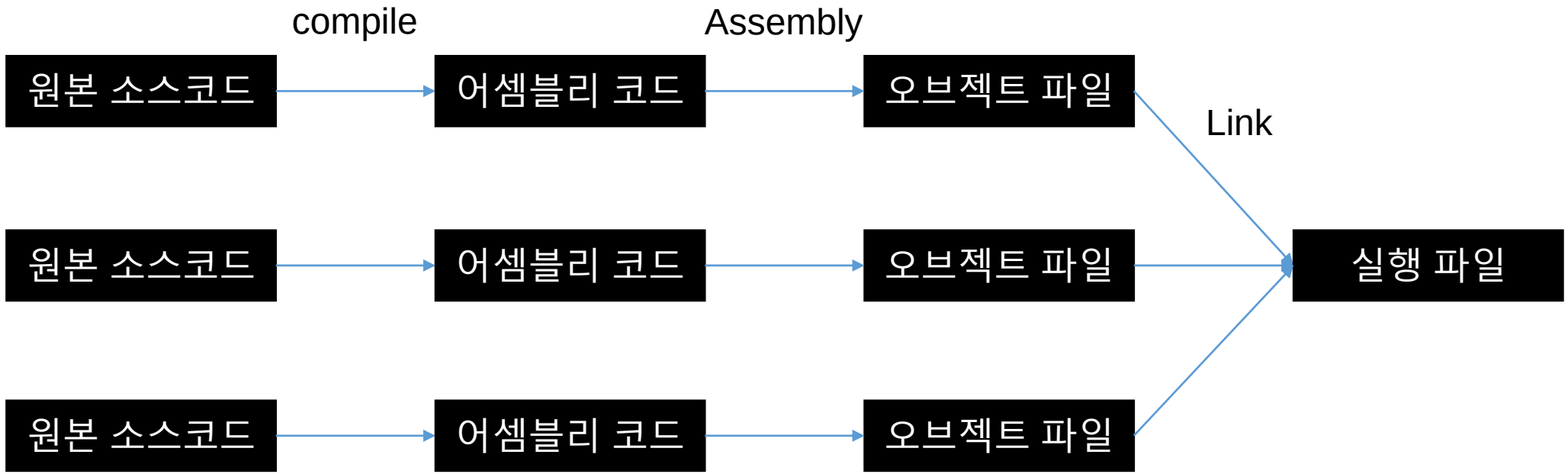
어셈블리 실습

https://youtu.be/k9tiR86f4is

컴파일러와 어셈블러


compile
Assembly
원본 소스코드
어셈블리 코드
오브젝트 파일
Link
원본 소스코드
어셈블리 코드
오브젝트 파일
실행 파일
원본 소스코드
어셈블리 코드
오브젝트 파일

프로그램 소개

- NASM
  - intel x86 아키텍처용 어셈블러.
  - 대중적인 리눅스용 어셈블러.
- ld
  - 리눅스용 링커
- gdb
  - GNU 디버거로 프로그램이 실행되는 동안 실행되는 내부에서 어떤 일이 일어나는지 보여주는 프로그램

실습 (PUSH / POP)

- 오브젝트파일 생성
  - nasm –felf32 pupo.asm
- 실행 파일 생성
  - ld –entry main pupo.o
- GDB 확인
  - gdb –q a.out

```
root@stud:/home/example/pupo# cat pupo.asm
global main

section .text
main:
        push    10h
        pop     eax
```

```
root@stud:/home/example/pupo# ls
a.out  pupo.asm  pupo.o
```

실습 (PUSH / POP)

```
root@stud:/home/example/pupo# gdb -q a.out
Reading symbols from a.out...(no debugging symbols found)...done.
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
   0x08048060 <+0>:     push   0x10
   0x08048062 <+2>:     pop    eax
End of assembler dump.
```

실습 (ADD)

```
root@stud:/home/example/add# cat add.asm
global main

section .text
main:
        mov     eax, 1
        mov     ebx, 3
        add     eax, ebx

        mov     eax, 1
        mov     ebx, 0
        int     80h
```

Exit(0)
종료코드

실습 (ADD)

```
(gdb) disas main
Dump of assembler code for function main:
   0x08048060 <+0>:     mov    eax,0x1
   0x08048065 <+5>:     mov    ebx,0x3
   0x0804806a <+10>:    add    eax,ebx
   0x0804806c <+12>:    mov    eax,0x1
   0x08048071 <+17>:    mov    ebx,0x0
   0x08048076 <+22>:    int    0x80
End of assembler dump.
```

실습 (LEA / MOV)

```
root@stud:/home/example/leamov# cat leamov.asm
global main

section .text
main:
        mov     eax, 1
        mov     ebx, 4
        mov     ecx, 7
        lea     eax, [eax+ecx]
        lea     ebx, [ebx*4]
        mov     eax, [eax+ecx]


        mov     eax, 1
        mov     ebx, 0
        int     80h
```

오류 발생
LEA와 다르게 연산 값이 주소로 인식됨.

실습 (LEA / MOV)

```
(gdb) disas main
Dump of assembler code for function main:
   0x08048060 <+0>:     mov    eax,0x1
   0x08048065 <+5>:     mov    ebx,0x4
   0x0804806a <+10>:    mov    ecx,0x7
   0x0804806f <+15>:    lea    eax,[eax+ecx*1]
   0x08048072 <+18>:    lea    ebx,[ebx*4+0x0]
   0x08048079 <+25>:    mov    eax,DWORD PTR [eax+ecx*1]
   0x0804807c <+28>:    mov    eax,0x1
   0x08048081 <+33>:    mov    ebx,0x0
   0x08048086 <+38>:    int    0x80
End of assembler dump.
```

연산값 15는 주소로 인식되어 0x00000015의 주소의 저장된 값을 eax에 넣음

```
(gdb) x/x 0x00000015
0x15:   Cannot access memory at address 0x15
```

해당 주소에는 접근 할 수 없어 Segentation fault 가 발생함

Q & A