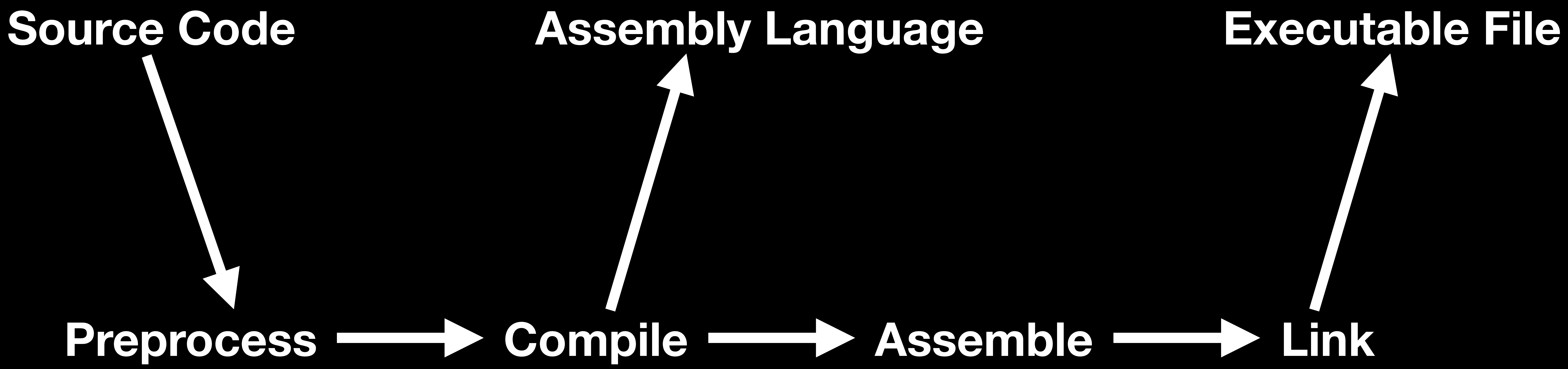
Assembly Language

AssassinQ

Introduction

- Sometimes referred to as **assembly** or **ASM**, an **assembly language** is a low-level programming language.
- Programs written in assembly languages are compiledby an assembler. Every assembler has its own assembly language, which is designed for one specific computer architecture.


Source Code
Assembly Language
Executable File
Preprocess
Compile
Assemble
Link

NASM

- Day1/intro
- nasm -f elf64 main.c -o main.o
- gcc main.o -o main.c
- The Netwide Assembler (**NASM**) is
  an assembler and disassembler for the Intel x86 architecture. It can be used to write 16-bit, 32-bit (IA-32) and 64-bit (x86-64) programs. **NASM** is considered to be one of the most popular assemblers for Linux.

Registers

- A processor **register** (CPU **register**) is one of a small set of data holding places that are part of the computer processor. A **register** may hold an instruction, a storage address, or any kind of data (such as a bit sequence or individual characters). Some instructions specify **registers** as part of the instruction.

- x86:
- EAX EBX ECX EDX
- ESI EDI
- EIP EFLAGS
- ESP EBP

Instructions

- MOV: assign value of src op to dst op (tip: size of ops must be the same)
- LEA: assign effective add of src op to dst op
- ADD/SUB(CMP)/MUL/DIV: cmp eax, 1
- AND/OR/XOR/TEST:
  - and dl, 11101100b / or dl, 00100000b
  - xor eax, eax / test eax, eax
- PUSH/POP: push op to the top of stack/pop the top of stack to op
- JMP/CALL/RET: jump to a specific location

- **Intel**:
- mov eax, 1
- mov eax, [ebx + 3]
- mov eax, [ebx + ecx * 2h]
- **AT&T**:
- movl $1, %eax
- movl 3(%ebx), %eax
- addl (%ebx, %ecx, 0x2), %eax

objdump

- objdump -M intel -d ./main

- **objdump** is a command-line program for displaying various information about object files on Unix-like operating systems. For instance, it can be used as a disassembler to view an executable in assembly form. It is part of the GNU Binutils for fine-grained control over executables and other binary data. objdump uses the BFD library to read the contents of object files.

Practice

- Day1/readasm
- Day1/overflow

GNU Debugger (gdb)

- .gdbinit: set disassembly-flavor intel
- lay src / asm / reg (Ctrl-x + a)
- **b**reak/**r**un/**c**ontinue
- **a**ttach
- **n**ext/**s**tep/**n**ext**i**/**s**tep**i**
- **p**rint/**i**nfo
- **b**ack**t**race

Practice

- Day1/add
- Day1/guess
- Day1/sum

Reference

- 11/16 社課 x86 assembly & GDB
- [107 學年第一學期社課] Reverse - Reverse 0x02