CHAPTER

6

RUNNING PROGRAM ON A SYSTEM

TRUONG DAI HOC KHOA HOC TU NHIEN
KHTN
TP. HO CHI MINH

cdio 4.0

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

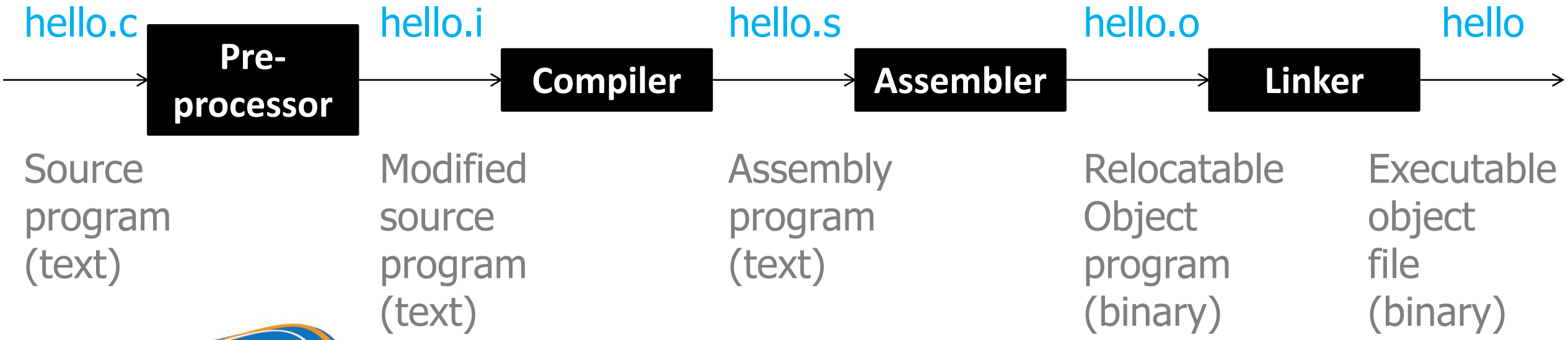
fit@hcmus

What will you learn?

- Compile drivers
- Object files
- Static Linking
- Dynamic Linking
- Loading Executable Object Files

Compiler drivers

- That invokes the language preprocessor, compiler, assembler, and linker to process a program

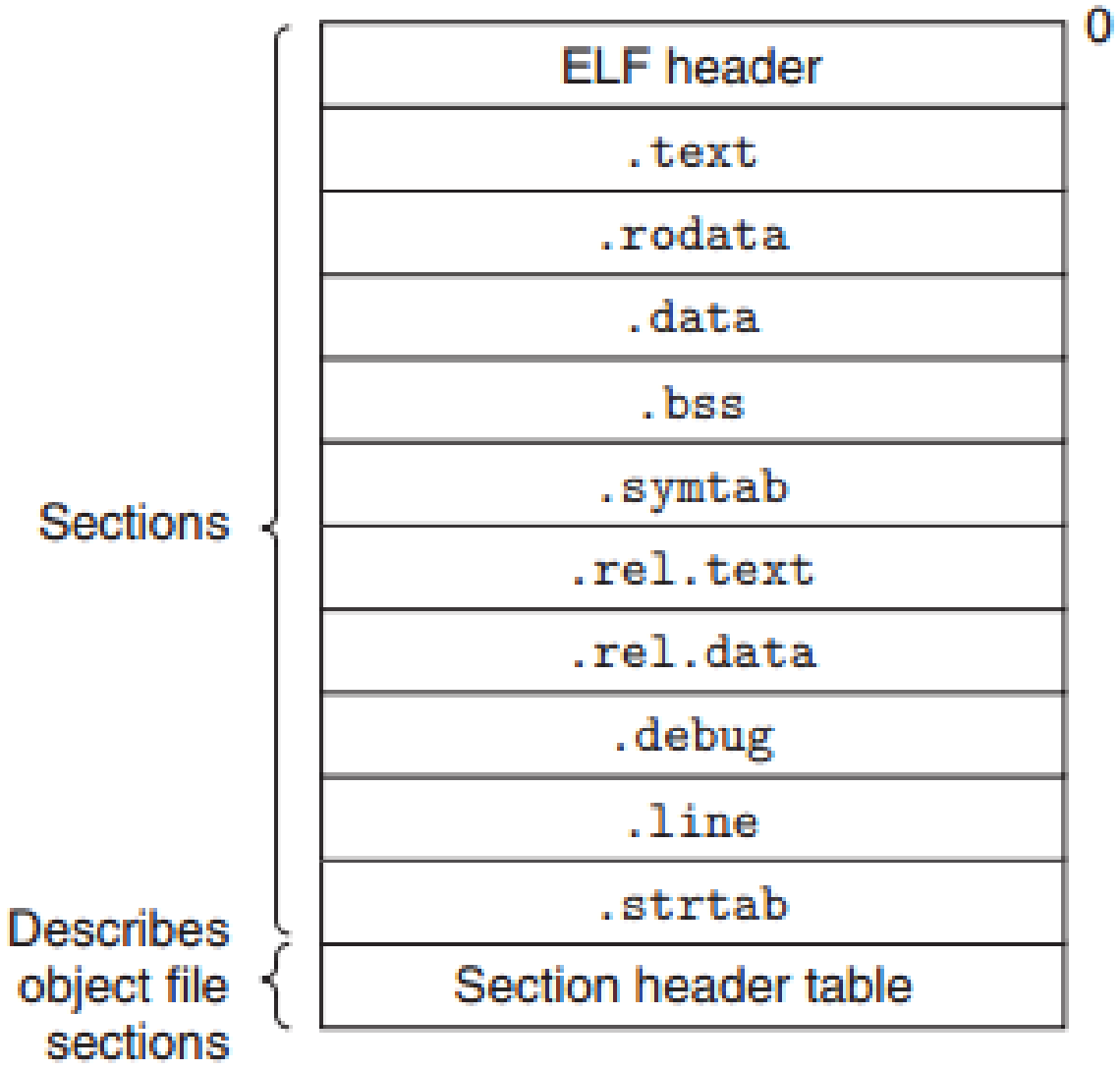


Object files

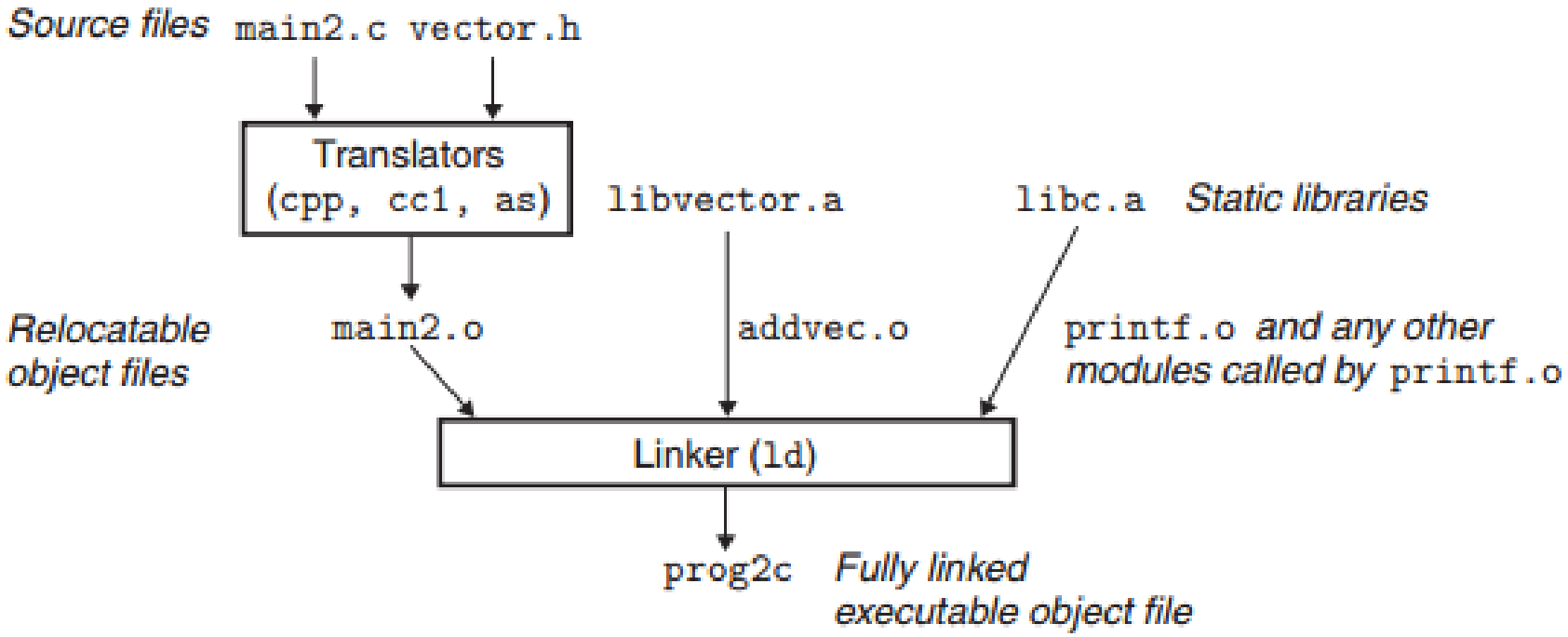
- Relocatable object file can be combined with other object files at compile time to create an executable object file, shared object file, or another relocatable object
- Executable object file holds a program that can be copied directly into memory and executed.
- Shared object file (special relocatable object file) can be loaded into memory and linked dynamically at either load-time or run-time

Object files

Typical ELF relocatable object file



Randal E. Bryant and David R. O'Hallaron, Computer Systems. A Programmer's Perspective, 2016, Fig 7.3)



Randal E. Bryant and David R. O'Hallaron, Computer Systems. A Programmer's Perspective, 2016, Fig 7.8)

Static Linking

Static linking is the result of the linker copying all library routines (a collection of relocatable object files) and command-line arguments used in the program into the executable object file

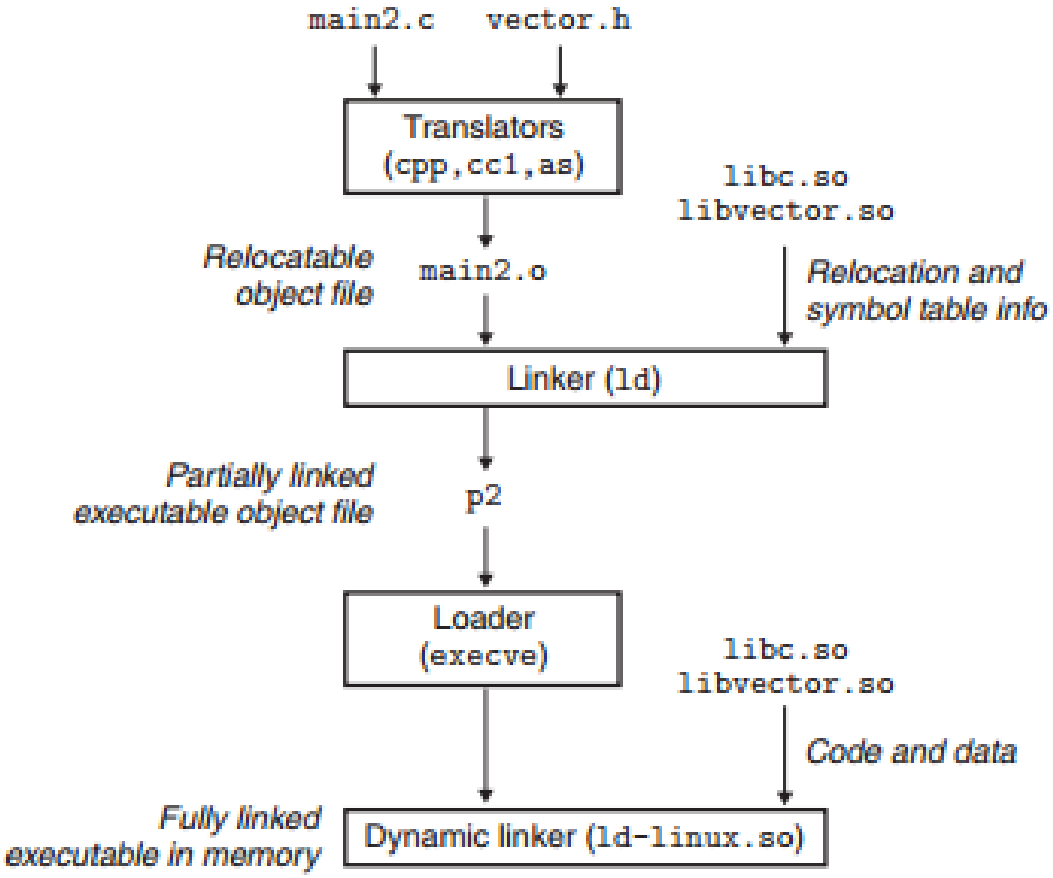
Static Linking

The linker must perform 2 main tasks:

- Symbol resolution: is to associate each symbol reference with exactly one symbol definition
- Relocation: The linker concatenates the code blocks, the data blocks, and data structures together, associates a memory location with each symbol definition, and modifies various locations within the code and data blocks.

Dynamic Linking

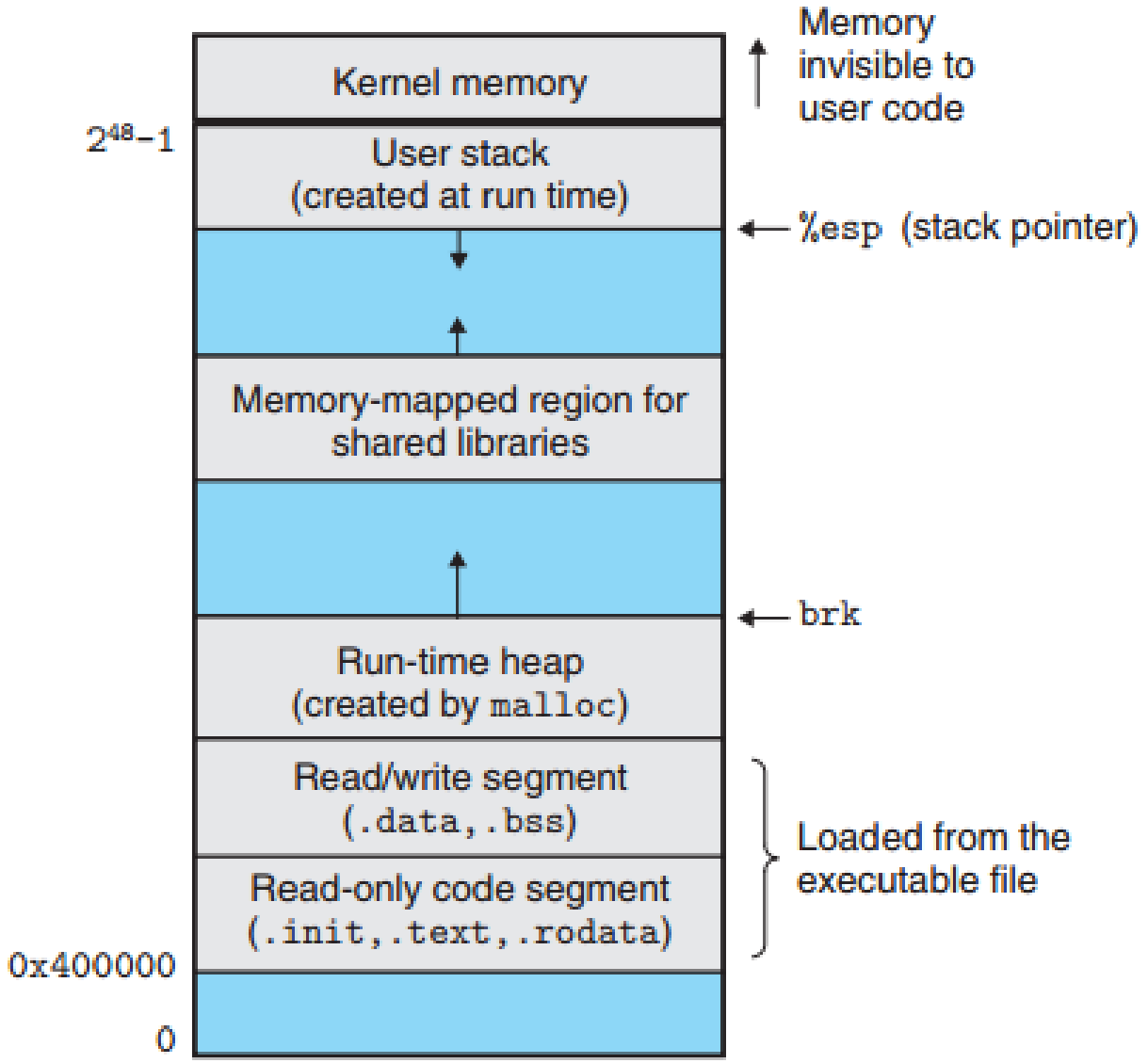
- Linkers can also produce partially linked executable object files with unresolved references to the routines and data defined in a shared library
- Dynamic linker: completes the linking task by loading the shared library and relocating the references in the program



Randal E. Bryant and David R. O'Hallaron, Computer Systems. A Programmer's Perspective, 2016, Fig 7.16)

Loading Executable Object File

The loader copies the code and data in the executable object file from disk into memory and then runs the program by jumping to its first instruction, or the program's entry point



Randal E. Bryant and David R. O'Hallaron, Computer Systems. A Programmer's Perspective, 2016, Fig 7.15)

Reference

Randal E. Bryant and David R. O'Hallaron, Computer Systems. A Programmer's Perspective, 2016, Chapter 7, p. 707