Programming with C I

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.10.21

Header files: defining the interface

- Gets copied in by preprocessor and then compiled in the .c file

- A .h file is never in the compile command

gcc –o exe –Wall program.c

Header files: defining the interface

#include <stdio.h>
versus
#include "class.h"

▶ **Angle brackets versus quotes tells compiler where to look for the file**

.c files: the implementation

- Contain C code

- Do get compiled separately
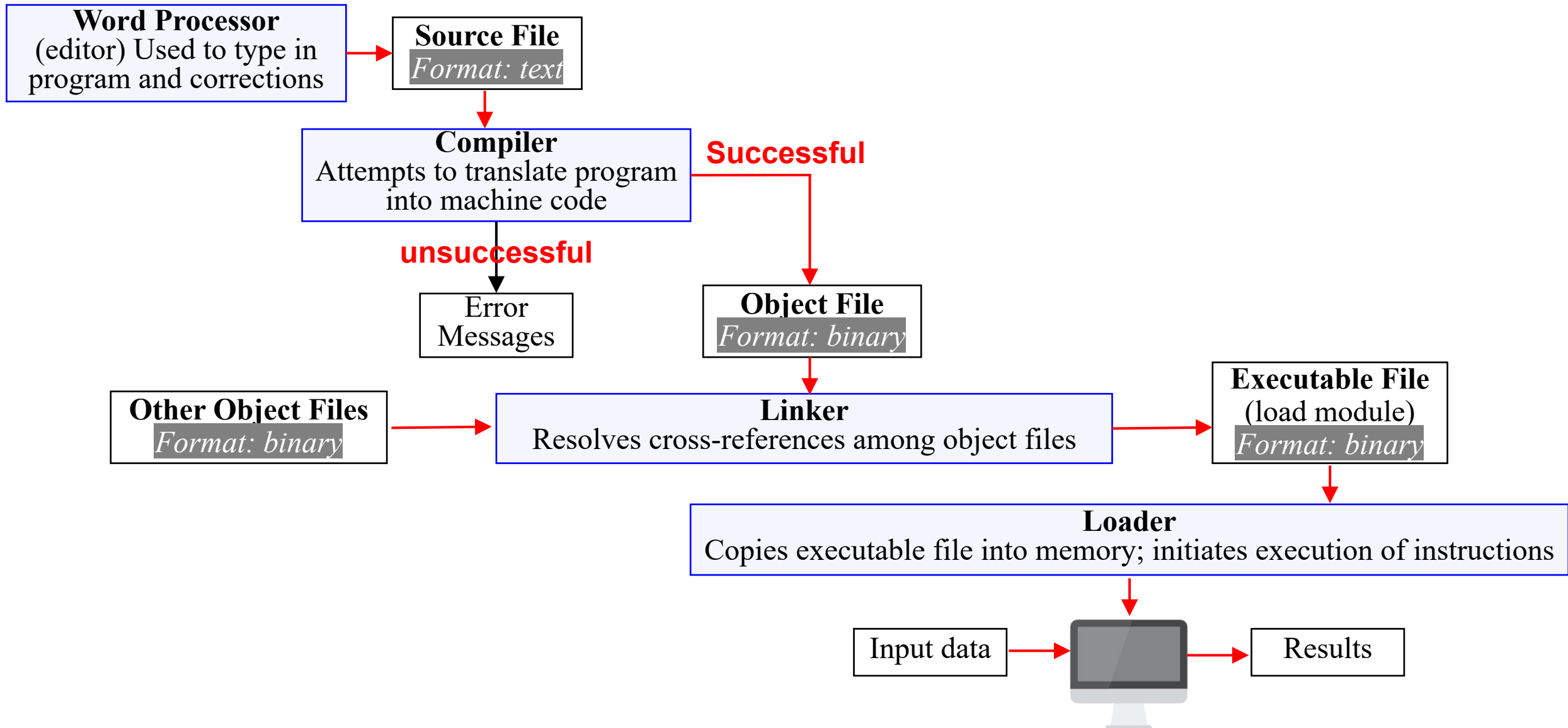
- Are linked after compilation to form the executable

gcc –o exe –Wall program.c funcs.c

Header guards

▶ **We don't want to include headers multiple times, but they may reference one another**

▶ **Solution: header guards**

```
#ifndef FILENAME_H
#define FILENAME_H
/* ... Declarations here ... */
#endif
```

Figure Entering, Translating, and Running a High-Level Language Program

Word Processor
(editor) Used to type in program and corrections

Source File
Format: text

Compiler
Attempts to translate program into machine code

Successful

unsuccessful

Error Messages

Object File
Format: binary

Other Object Files
Format: binary

Linker
Resolves cross-references among object files

Executable File
(load module)
Format: binary

Loader
Copies executable file into memory; initiates execution of instructions

Input data

Results

Object files

- use .o extension
- -c flag in gcc only compiles, does not link
- running gcc on object files will link them

gcc file1.o file2.o

warning: -o does not make object files! -c does.

Makefiles

- Once we have many .c files, which should be recompiled?
- a Makefile consists of rules (saying what depends on what) that make follows to recompile any changed code.
- first rule is default target (final executable, for us)

target: dependencies
    what to do (how to build target)

warning: tab is required.

```
exe: student.o student_prog.o
    gcc -o exe student.o student_prog.o

student_prog.o: student_prog.c student.h
    gcc -c student_prog.c -Wall

student.o: student.c student.h
    gcc -c student.c -Wall
```

- **exe depends on student.o and student_prog.o**
- **student_prog.o depends on student_prog.c and student.h**
- **student.o depends on student.c and student.h**
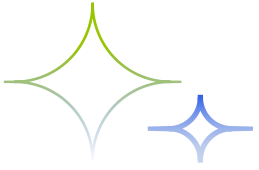
Makefiles

▶ can run with specific targets

make mytarget

▶ clean should clean up directory by removing object and executables

make clean

▶ can use variables

THE END

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.10.21