



EE6094 CAD for VLSI Design



GNU Make and Makefile

Andy, Yu-Guang Chen
Assistant Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw

Slides Credit: TA金昌明/TA蔡書儀/TA施奕瑄



2024/4/11

Andy Yu-Guang Chen

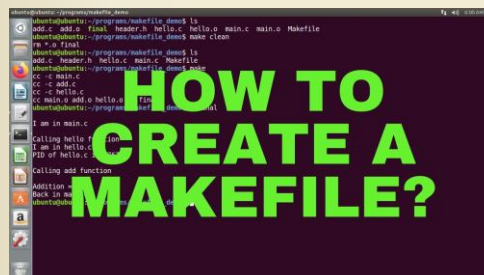
1



Outline



- ◆What is GNU make and makefile?
- ◆How to write a makefile?
- ◆How to compile and execute your code with makefile?



2024/4/11

Andy Yu-Guang Chen

2



GNU make and makefile



◆ What is GNU(GNU's not unix) Make?

- GNU is a extensive collection of free software which can be used as an operating system or can be used in parts with other operating systems.
- GNU Make is a tool which controls the generation of executables and other non-source files of a program from the program's source files.
- Make gets its knowledge of how to build your program from a file called the *makefile*, which lists each of the non-source files and how to compute it from other files.
- The make utility automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them.



2024/4/11

Andy Yu-Guang Chen

3



How to write a makefile



◆ How a makefile looks like?

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```



2024/4/11

Andy Yu-Guang Chen

4



How to write a makefile



◆ Define make-command

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```



2024/4/11

Andy Yu-Guang Chen

5



How to write a makefile



◆ Target

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
    @rm exe
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```



2024/4/11

Andy Yu-Guang Chen

6



How to write a makefile



◆Dependency

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
    @rm exe
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```



2024/4/11

Andy Yu-Guang Chen

7



How to compile your code with makefile



◆Make all

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
    @rm exe
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```

※Note: $\$(...)$'s are parameters that can defined by user



2024/4/11

Andy Yu-Guang Chen

8



How to compile your code with makefile



◆ Make run \$(input) \$(output)

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
    @rm exe
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```

※Note: \$(...)'s are parameters that can defined by user



2024/4/11

Andy Yu-Guang Chen

9



How to compile your code with makefile



◆ Make clean

```
#PA1 cpp compiler
.PHONY: all run clean
all: pa1_main.o pa1_function.o
    @g++ -std=c++11 pa1_main.o pa1_function.o -o exe
run:
    @./exe $(input) $(output)
clean:
    @rm *.o
    @rm exe
pa1_main.o: pa1_main.cpp
    @g++ -std=c++11 -c pa1_main.cpp
pa1_function.o: pa1_function.h pa1_function.cpp
    @g++ -std=c++11 -c pa1_function.h pa1_function.cpp
```

※Note: \$(...)'s are parameters that can defined by user



2024/4/11

Andy Yu-Guang Chen

10



GNU make and makefile



- ◆ An example of using make to compile and execute your program

```
[ta110521028@eda359_forclass ~/PA1]$ ls
c17.v  Makefile  PA1_main.cpp
→ [ta110521028@eda359_forclass ~/PA1]$ make all
[ta110521028@eda359_forclass ~/PA1]$ ls
c17.v  exe  Makefile  PA1_main.cpp  PA1_main.o
→ [ta110521028@eda359_forclass ~/PA1]$ make run input=c17.v output=c17.bench
----- Verilog Parser -----
Input file: c17.v
Output file: c17.bench
[ta110521028@eda359_forclass ~/PA1]$ ls
c17.bench  c17.v  exe  Makefile  PA1_main.cpp  PA1_main.o
→ [ta110521028@eda359_forclass ~/PA1]$ make clean
[ta110521028@eda359_forclass ~/PA1]$ ls
c17.bench  c17.v  Makefile  PA1_main.cpp
```



2024/4/11

Andy Yu-Guang Chen

11



PA2 requirement



- ◆ Your makefile should at least contain these 3 commands, which is (1) **make all**, (2) **make run**, and (3) **make clean**.
- ◆ **make all**: This command will automatically compile your source codes and generate the corresponding objects and executable file.
- ◆ **make run**: This command will execute your executable file and run your program.
- ◆ **make clean**: This command will automatically remove all the objects and executable file generated by **make all**.



2024/4/11

Andy Yu-Guang Chen

12



Demo session

◆ In demo session, TA will test your makefile in the following commands.

- make all
- make run input=case output=case.out
- make clean

```
[109521018@eda359_forclass ~/PA3]$ make all
Creating object directory
Compiling: PA3_109521018.cpp -> build/PA3_109521018.o
Compiling: linkedlist.cpp -> build/linkedList.o
Generating executable file: build/PA3_109521018.o build/linkedList.o -> go
[109521018@eda359_forclass ~/PA3]$ make run INPUT=case0 OUTPUT=case0.out
./go case0 case0.out
[109521018@eda359_forclass ~/PA3]$ make clean
Removing objects
Removing executable file
[109521018@eda359_forclass ~/PA3]$
```



2024/4/11

Andy Yu-Guang Chen

13



Reference

- ◆ <https://mropengate.blogspot.com/2018/01/makefile.html>
- ◆ <https://michaelchen.tech/gnu-make/>
- ◆ <https://blog.jaycetye.com/2015/01/linux-gcc-makefile/>
- ◆ <https://www.cc.ntut.edu.tw/~yccheng/oop2005f/GNUMakeManual.pdf>
- ◆ <https://www.youtube.com/watch?v=O5mG8H36V44>



2024/4/11

Andy Yu-Guang Chen

14