Week 3

CS110P.07

Suting Chen

- Make
- CMake

Make (>= 3.82)

Guideline

- Rule
- Phony target
- Automatic variables
- (User-defined) variable
- Function

Suting Chen

Make — Rule

What is a rule?

```
target : prerequisites
    recipe 1
    recipe 2
```

Suting Chen

Make — Rule

```
file0.txt:
    echo "foo" > file0.txt
```

```
file1.txt: file0.txt
    cat file0.txt > file1.txt
```

Suting Chen

Make — Phony target

```
.PHONY: clean all

clean:
    rm *.o
    rm *.log

all: file0.txt file1.txt
    echo "Complete!"
```

Suting Chen

Make — Automatic variables

```
$@  # File name of the target
$<  # First prerequisite
$?  # All prerequisites

$^ $| $+ $* $(*D) $(?F) ...
```

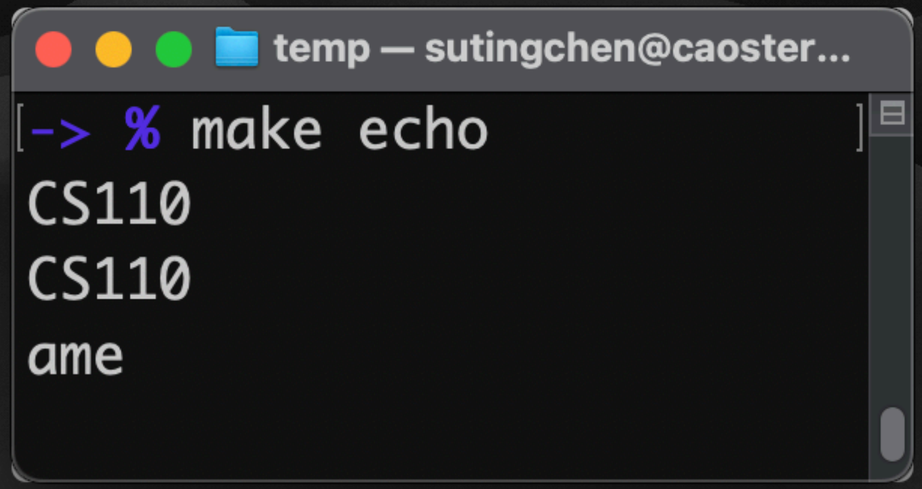
Suting Chen

Make — Variable

Equivalent to macros

```
name = CS110

echo:
        @echo $(name)
        @echo ${name}
        @echo $name
        @echo $(name0)
```



Suting Chen

Make — Variable & Function

Very complicated ...

- Declare / set variables :

```
= := ::= :::= ?= += !=
```

Suting Chen

Make — Variable & Function

Builtin functions (subst, foreach, file, shell, call, eval …)

```
truth := $(subst python,cpp,python is the best)

echo:
	@echo ${truth}
```

```
> make echo
 cpp is the best
```

Suting Chen

Make — Variable & Function

Declare functions

```
define speak
	@echo ${1} said \"${2}\"
endef

all:
	$(call speak,Flash Hu,Trivial)
```

```
> make
 Flash Hu said "Trivial"
```

Suting Chen

Make — something strange

- Tab
- wildcard
- one shell per line

Suting Chen

Make — something strange

Tab !

```
task:
    @echo "Hello" > 1.txt
```

```
> make task
 Makefile:2: *** missing separator.  Stop.
```

Make — something strange

Tab !

```
.RECIPEPREFIX = >
all:
>@echo Hello, world
> @echo Hello again, world
```

Suting Chen

Make — something strange

Use wildcard properly

```
objects := *.c

print: $(objects)
    @echo $(objects)

.PHONY: print
```

```
> make print
 *** No rule to make target '*.c', needed by 'print'.  Stop.
```

Make — something strange

Use wildcard properly

```
objects := $(wildcard *.c)

print: $(objects)
    @echo $(objects)

.PHONY: print
```

Suting Chen

Make — something strange

One shell per line

```
test:
	@export value=42
	@echo $${value}
```

Suting Chen

Make — something strange

One shell per line

```
test:
	@export value=42; \
	echo $${value}

test:
	@export value=42; echo $${value}
```

```
.ONESHELL:

test:
	@export value=42
	@echo $${value}
```

Suting Chen

Make

Conclusion

- Not related to gcc
- Smart Automation
- Turing complete!

CMake (>= 3.14)

- add_executable( )
- aux_source_directory( )
- add_subdirectory( )
- include_directories( )
- find_package( )
- target_link_libraries( )

CMake — Basic

Hello world

```
cmake_minimum_required (VERSION 3.14)
project(CS110_Lab3)
add_executable(CS110_Lab3 main.c)
```

```
mkdir build && cd build
cmake .. && make
./CS110_Lab3
----> Hello, world!
```

Suting Chen

CMake — Basic

Hello world

- CMake cannot compile

- CMake generates Makefile

```
-> % tree
.
├── CMakeLists.txt
├── build
|   ├── CS110_Lab3
|   ├── Makefile
|   └── others
└── main.c
```

Suting Chen

CMake — Basic

What if more files?


```c
#include <stdio.h>
#include "test.h"

int main() {
    hello(); // Function from `test.h'
    return 0;
}
```

Suting Chen

CMake — Basic

What if more files?

```
.
├── CMakeLists.txt
├── include
|   └── test.h
├── main.c
└── src
    └── test.c
```

```
cmake_minimum_required(VERSION 3.14)
project(CS110_Lab3 C)

include_directories(include)
aux_source_directory(src SRC_FILES)

add_executable(CS110_Lab3 main.c ${SRC_FILES})
```

Suting Chen

CMake — compile options?

Use built in variables!

```
cmake_minimum_required(VERSION 3.14)
project(CS110_Lab3 C)

set(C_FLAGS -Wall)
set(CMAKE_C_FLAGS_DEBUG "${C_FLAGS} -O0 -g")
set(CMAKE_C_FLAGS_RELEASE "${CFLAGS} -O3")
set(CMAKE_C_STANDARD 17)

add_executable(CS110_Lab3 main.c)
```

Let’s go further

Why CMake instead of make

Suting Chen

CMake — libraries

Preface
Introduction

```
accesstimer: acctime.o bentimer.o
    g++ -L/usr/local/boost/boost_1_81_0/stage/lib\
        -lboost_filesystem acctime.o bentimer.o\
        -o accesstimer
```

10. Boost.Tokenizer
11. Boost.Spirit
III. Containers
12. Boost.MultiIndex
13. Boost.Bimap
14. Boost.Array
15. Boost.Unordered
16. Boost.CircularBuffer
17. Boost.Heap
18. Boost.Intrusive
19. Boost.MultiArray
20. Boost.Container

Suting Chen

CMake — libraries

```
cmake_minimum_required(VERSION 3.14)
project(CS110_Lab3 CXX)

include_directories(include)
aux_source_directory(src SRC_FILES)

find_package(Boost REQUIRED COMPONENTS system filesystem)

add_executable(CS110_Lab3 main.cpp ${SRC_FILES})
target_link_libraries(CS110_Lab3 PRIVATE Boost::filesystem)
```

Suting Chen

CMake — libraries

“Header-only”

```
cmake_minimum_required(VERSION 3.14)
project(CS110_Lab3 CXX)
set(CMAKE_CXX_STANDARD 20)

include_directories(${Boost_INCLUDE_DIR})

add_executable(CS110_Lab3 main.cpp)
```

CMake — libraries

add_subdirectory( )

```
.
├── CMakeLists.txt
├── JUCE
|   ├── CMakeLists.txt
|   └── others
└── main.c
```

```
cmake_minimum_required(VERSION 3.14)
project(CS110_Lab3 CXX)

add_subdirectory(JUCE)

add_executable(CS110_Lab3 main.cpp)
```

Suting Chen

- Make
- CMake
- 如何在28学分5门专业课中活下来

Suting Chen

Recommended readings

1. https://learnxinyminutes.com/docs/cmake/
2. https://www.gnu.org/software/make/manual/make.html#Reference