



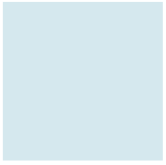
Introduction to Linux Systems

GNU Make

Chia-Heng Tu

Dept. of Computer Science and Information
Engineering

National Cheng Kung University
Fall 2022



Lab

- A Simple C Project Structure



New folder for this lab

- 安裝相關套件

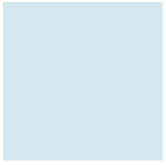
```
$ sudo apt install gcc tree make
```

- 建立作業資料夾

```
$ cd <Your local git repo>
```

```
$ mkdir lab_5_<student_ID>
```

```
$ cd lab_5_<student_ID>
```



Add .gitignore to prevent tracking object files

```
# Ignore the build and lib dirs
build
lib/*

# Ignore any executables
bin/*

# Ignore temporary files
*.swp
```

```
.gitignore
├── include/
│   └── strcpy.h
├── Makefile
├── src/
│   ├── strcpy.c
│   └── main.c
```



Split function into different file (Header)

```
#pragma once
```

```
char *strcpy(char *dest, const char
*src);
```


or

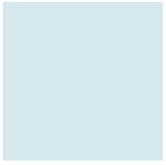
```
#ifndef STRCPY_H
#define STRCPY_H
```

```
char *strcpy(char *dest, const char
*src);
```

```
#endif /* STRCPY_H */
```

```
├── .gitignore
├── include/
│   └── strcpy.h
├── Makefile
├── src/
│   ├── strcpy.c
│   └── main.c
```

Note:  include guard
https://en.wikipedia.org/wiki/Pragma_once
https://en.wikipedia.org/wiki/Include_guard



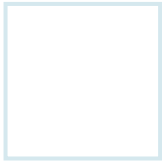
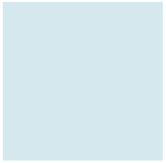
Split function into different file (Source)

```
#include "../include/strcpy.h"

char *strcpy(char *dest, const char
*src)
{
    // FIXME
}
```

不能使用其他 library 提供的
function

```
.gitignore
include/
└─ strcpy.h
Makefile
src/
└─ strcpy.c
└─ main.c
```

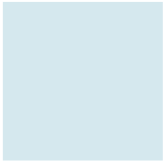


Program entry point

```
#include <stdio.h>
#include <stdlib.h>
#include "../include/strcpy.h"
int main() {
    const char *src = "f12345678";
    char *dest = malloc(10);
    dest = sstrcpy(dest, src);
    printf("%s\n", dest);
    return 0;
}
```

改成你的學號

```
├── .gitignore
├── include/
│   └── strcpy.h
├── Makefile
├── src/
│   ├── strcpy.c
│   └── main.c
```



Makefile

```
CC := gcc
SRCDIR := src
BUILDDIR := build
BINDIR := bin
INCDIR := include
TARGET := $(BINDIR)/runner

SRCEXT := c
SOURCES := $(shell find $(SRCDIR) -type f -name *.$(SRCEXT))
HEADERS := $(shell find $(INCDIR) -type f -name *.h)
OBJECTS := $(patsubst $(SRCDIR)/%, $(BUILDDIR)/%, $(SOURCES:.$(SRCEXT)=.o))
CFLAGS := -O2 -Wall

all: $(TARGET)

$(TARGET): $(OBJECTS)
    mkdir -p $(BINDIR)
    @echo "> Linking..."
    $(CC) $^ -o $(TARGET)

$(BUILDDIR)/%.o: $(SRCDIR)/%.$(SRCEXT)
    mkdir -p $(BUILDDIR)
    @echo "> Compiling..."
    $(CC) $(CFLAGS) -c -o $@ $<

clean:
    @echo "> Cleaning...";
    $(RM) -rf $(BUILDDIR) $(TARGET)
```

```
.gitignore
include/
└─ strcpy.h
Makefile
src/
└─ strcpy.c
   main.c
```

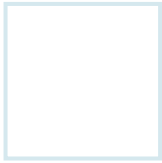
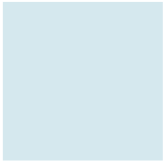



Variable and alias in Makefile

- SOURCES := \$(shell find \$(SRCDIR) -type f -name *.\$(SRCEXT))
 - src/strcpy.c src/main.c
- HEADERS := \$(shell find \$(INCDIR) -type f -name *.h)
 - include/strcpy.h
- OBJECTS := \$(patsubst \$(SRCDIR)/%, \$(BUILDDIR)/%, \$(SOURCES:.\$(SRCEXT)=.o))
 - \$(SOURCES:.\$(SRCEXT)=.o) = src/strcpy.o src/main.o
 - build/strcpy.o build/main.o
- \$(CC) \$(CFLAGS) -c -o \$@ \$<
 - gcc -O2 -Wall -c -o build/strcpy.o src/strcpy.c
 - gcc -O2 -Wall -c -o build/main.o src/main.c
- main.o: main.c a.o
 - \$(CC) \$(CFLAGS) -c -o \$@ \$<

```

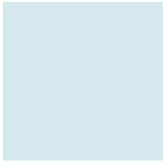
├── .gitignore
├── include/
│   └── strcpy.h
├── Makefile
└── src/
    ├── strcpy.c
    └── main.c
  
```



Current file architecture

\$ tree .

```
.
├── include/
│   └── strcpy.h
├── Makefile
└── src/
    ├── strcpy.c
    └── main.c
```



Use Makefile and execute program

\$ make

mkdir -p build

> Compiling...

gcc -O2 -Wall -I include -c -o build/strcpy.o src/strcpy.c

mkdir -p build

> Compiling...

gcc -O2 -Wall -I include -c -o build/main.o src/main.c

mkdir -p bin

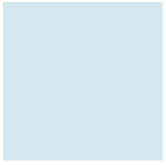
> Linking...

gcc build/strcpy.o build/main.o -o bin/runner

這邊輸出可能會有 warning，如果
程式正常執行可忽略

\$./bin/runner

f12345678



File architecture after using make

\$ tree .

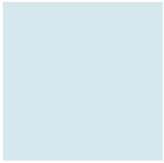
```
.
├── bin/
│   └── runner
├── build/
│   ├── strcpy.o
│   └── main.o
├── include/
│   └── strcpy.h
├── Makefile
└── src/
    ├── strcpy.c
    └── main.c
```



Demo

- 作業上傳：

將 lab_5_<student_ID> 資料夾上傳至 moodle (請同學使用 Linux 環境撰寫作業)
- 評分方式：
 - 80 %
 - 執行 `$ make && $./bin/runner` 正確輸出學號
 - 20 %
 - 其他測資驗證 `sstrcpy` function (測資不公開)



QUESTIONS