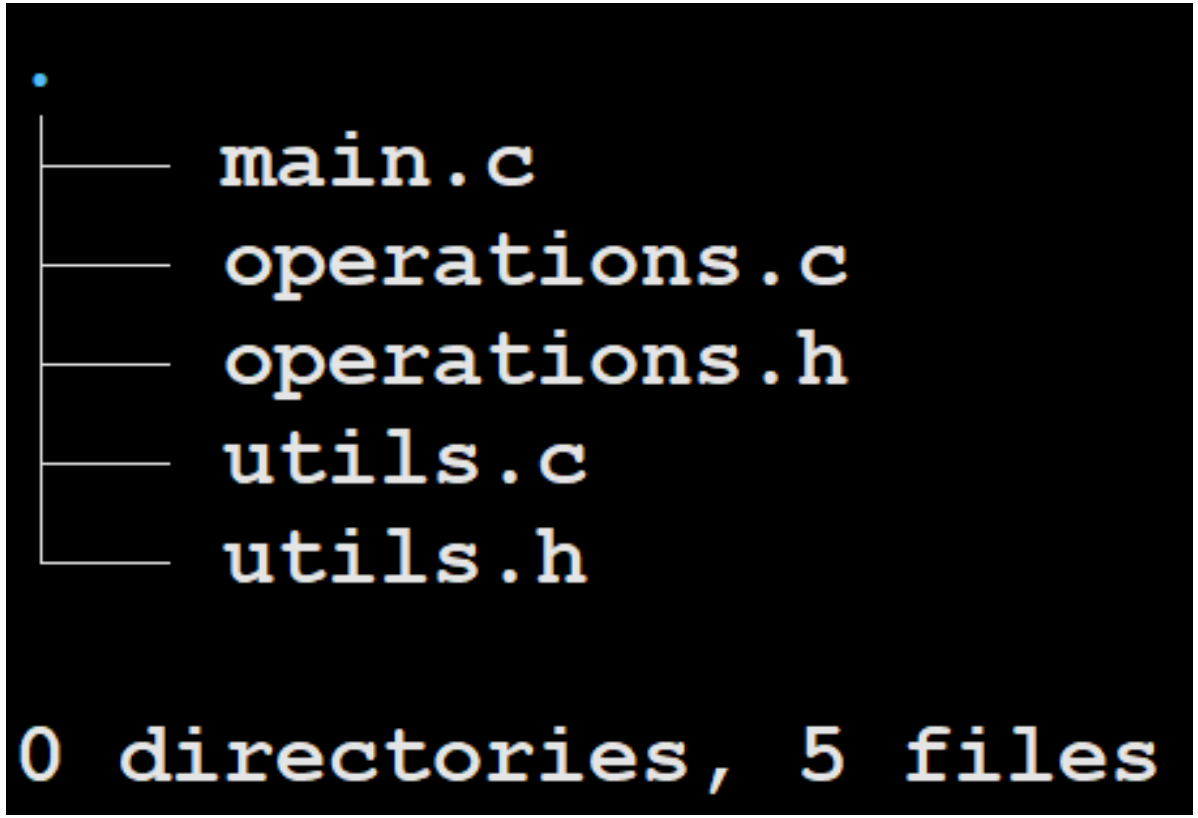
程序设计
Programming

Lecture 14: Makefile简介

GNU Make：自动化编译和程序构建

- 在Makefile中编写大型项目的编译规则
  - ✓目标：前提1 前提2 前提3 …
    gcc/g++ command
- 使用make命令来执行自动编译和程序构建
  - ✓Make 目标（可省略）
- gcc和make参考Tutorial
  - ✓https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html

gcc的使用


.
main.c
operations.c
operations.h
utils.c
utils.h
0 directories, 5 files

gcc的使用

```
.
├── main.c
├── operations.c
├── operations.h
├── utils.c
└── utils.h

0 directories, 5 files
```

```
gcc -o complexSum main.c operations.c utils.c
```

gcc的使用

```
.
├── main.c
├── operations.c
├── operations.h
├── utils.c
└── utils.h

0 directories, 5 files
```

```
.
├── complexSum
├── main.c
├── operations.c
├── operations.h
├── utils.c
└── utils.h
```

```
gcc -o complexSum main.c operations.c utils.c
```

gcc的使用

```
.
├── main.c
├── operations.c
├── operations.h
├── utils.c
└── utils.h

0 directories, 5 files
```

缺点

- 每次编译都要编写完整的gcc命令
- 每次编译都要编译所有源文件

```
gcc -o complexSum main.c operations.c utils.c
```

Makefile

```
IDIR=include
ODIR=obj
SDIR=src

CC=gcc

_OBJS=main.o utils.o operations.o
OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
_MAINDEPS=utils.h operations.h
MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
CFLAGS=-c -o
DEPFLAGS=-I$(IDIR)
BINARY=complexSum

all: $(OBJS)
	$(CC) -o $(BINARY) $(OBJS) $(DEPFLAGS)

$(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

$(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

clean:
	rm -f $(OBJS)
	rm $(BINARY)
```

Makefile基本语法：一系列规则

```
target: prerequisites
    command
    command
    command
    …
```

每个规则包含：

- ✓ target: make目标，一般每个rule只包含一个目标
- ✓ prerequisites: make当前目标需要的先决文件（依赖文件）
- ✓ command: 一系列编译命令或者其他shell命令（前面有tab）

一个简单的Makefile

```
all: main.c utils.c operations.c
	gcc -o complexSum main.c utils.c operations.c
```

一个简单的Makefile

```
all: main.c utils.c operations.c
	gcc -o complexSum main.c utils.c operations.c
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -o complexSum main.c utils.c operations.c
luxuesong@10-24-0-196:~/complexNumberOperation$ make all
gcc -o complexSum main.c utils.c operations.c
luxuesong@10-24-0-196:~/complexNumberOperation$
```

一个简单的Makefile

```
all: main.c utils.c operations.c
	gcc -o complexSum main.c utils.c operations.c
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -o complexSum main.c utils.c operations.c
luxuesong@10-24-0-196:~/complexNumberOperation$ make all
gcc -o complexSum main.c utils.c operations.c
luxuesong@10-24-0-196:~/complexNumberOperation$
```

缺点：每次编译仍然要编译所有源文件

Makefile V2

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

Makefile V2

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
cc     -c -o main.o main.c
cc     -c -o utils.o utils.c
cc     -c -o operations.o operations.c
gcc -o complexSum main.o utils.o operations.o
luxuesong@10-24-0-196:~/complexNumberOperation$
```

Makefile V2

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
cc     -c -o main.o main.c
cc     -c -o utils.o utils.c
cc     -c -o operations.o operations.c
gcc -o complexSum main.o utils.o operations.o
luxuesong@10-24-0-196:~/complexNumberOperation$
```

系统自动编译相关object（目标）文件，最后gcc将目标文件进行链接生成可执行文件

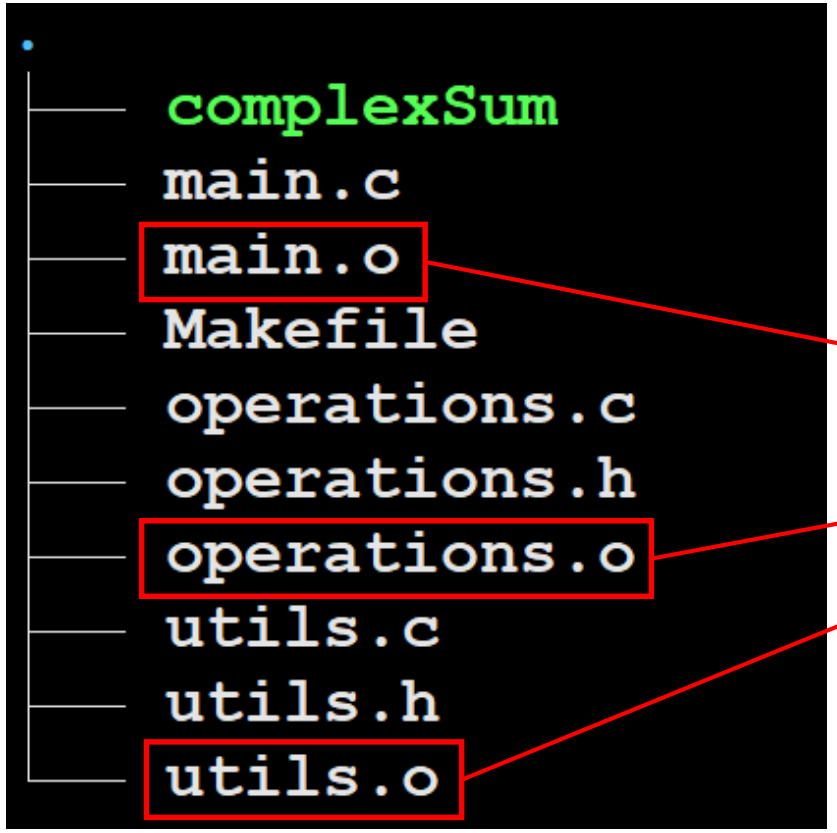
Makefile V2：当前文件结构

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

```
.
├── complexSum
├── main.c
├── main.o
├── Makefile
├── operations.c
├── operations.h
├── operations.o
├── utils.c
├── utils.h
└── utils.o
```

Makefile V2：当前文件结构

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```



目标文件

Makefile V2：改动源文件

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

假如改动utils.c

```
luxuesong@10-24-0-196:~/complexNumberOperation$ vim utils.c
luxuesong@10-24-0-196:~/complexNumberOperation$ make
cc     -c -o utils.o utils.c
gcc -o complexSum main.o utils.o operations.o
luxuesong@10-24-0-196:~/complexNumberOperation$
```

只重新编译了utils.c

Makefile V2：改动源文件

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o
```

假如改动utils.c

```
luxuesong@10-24-0-196:~/complexNumberOperation$ vim utils.c
luxuesong@10-24-0-196:~/complexNumberOperation$ make
cc     -c -o utils.o utils.c
gcc -o complexSum main.o utils.o operations.o
luxuesong@10-24-0-196:~/complexNumberOperation$
```

缺点：假如更改了.h文件，相关.c文件不会重新编译

只重新编译了utils.c

Makefile V3：关联*.c和*.h

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

%.o: %.c %.h
	gcc -c -o $@ $<
```

Makefile V3：关联*.c和*.h

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

%.o: %.c %.h
	gcc -c -o $@ $<
```

假如改动了utils.h，utils.c会被重新编译

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -c -o utils.o utils.c
gcc -o complexSum main.o utils.o operations.o
```

Makefile V3：关联*.c和*.h

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

%.o: %.c %.h
	gcc -c -o $@ $<
```

假如改动了utils.h，utils.c会被重新编译

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -c -o utils.o utils.c
gcc -o complexSum main.o utils.o operations.o
```

但是main.c没有重新编译!

Makefile V3：关联*.c和*.h

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

main.o: main.c utils.h operations.h
	gcc -c -o $@ $<

%.o: %.c %.h
	gcc -c -o $@ $<
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -c -o main.o main.c
gcc -c -o utils.o utils.c
gcc -o complexSum main.o utils.o operations.o
```

Makefile V3：关联*.c和*.h

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

main.o: main.c utils.h operations.h
	gcc -c -o $@ $<

%.o: %.c %.h
	gcc -c -o $@ $<
```

```
luxuesong@10-24-0-196:~/complexNumberOperation$ make
gcc -c -o main.o main.c
gcc -c -o utils.o utils.c
```

缺点：Makefile不美观，啰嗦

Makefile V4：使用宏（macro）

```
CC=gcc
OBJS=main.o utils.o operations.o
MAINDEPS=utils.h operations.h
CFLAGS=-c -o

all: $(OBJS)
	$(CC) -o complexSum $(OBJS)

main.o: main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $<

%.o: %.c %.h
	$(CC) $(CFLAGS) $@ $<
```

Makefile V4：使用宏（macro）

```
CC=gcc
OBJS=main.o utils.o operations.o
MAINDEPS=utils.h operations.h
CFLAGS=-c -o

all: $(OBJS)
	$(CC) -o complexSum $(OBJS)

main.o: main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $<

%.o: %.c %.h
	$(CC) $(CFLAGS) $@ $<
```

宏

规则

Makefile V4：使用宏（macro）

```
CC=gcc
OBJS=main.o utils.o operations.o
MAINDEPS=utils.h operations.h
CFLAGS=-c -o

all: $(OBJS)
	$(CC) -o complexSum $(OBJS)

main.o: main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $<

%.o: %.c %.h
	$(CC) $(CFLAGS) $@ $<
```

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

main.o: main.c utils.h operations.h
	gcc -c -o $@ $<

%.o: %.c %.h
	gcc -c -o $@ $<
```

Makefile V4：使用宏（macro）

```
CC=gcc
OBJS=main.o utils.o operations.o
MAINDEPS=utils.h operations.h
CFLAGS=-c -o

all: $(OBJS)
	$(CC) -o complexSum $(OBJS)

main.o: main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $<

%.o: %.c %.h
	$(CC) $(CFLAGS) $@ $<
```

```
.
├── complexSum
├── main.c
├── main.o
├── Makefile
├── operations.c
├── operations.h
├── operations.o
├── utils.c
├── utils.h
└── utils.o
```

```
all: main.o utils.o operations.o
	gcc -o complexSum main.o utils.o operations.o

main.o: main.c utils.h operations.h
	gcc -c -o $@ $<
```

缺点：所有文件都在同一目录下，没有层次

Makefile V5：将各类文件放到相应文件夹

```
.
├── include
│   ├── operations.h
│   └── utils.h
├── Makefile
├── obj
└── src
    ├── main.c
    ├── operations.c
    └── utils.c

3 directories, 6 files
```

Makefile V5：将各类文件放到相应文件夹

```
.
├── include
│   ├── operations.h
│   └── utils.h
├── Makefile
├── obj
└── src
    ├── main.c
    ├── operations.c
    └── utils.c

3 directories, 6 files
```

```
IDIR=include
ODIR=obj
SDIR=src

CC=gcc

_OBJS=main.o utils.o operations.o
OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
_MAINDEPS=utils.h operations.h
MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
CFLAGS=-c -o
DEPFLAGS=-I$(IDIR)

all: $(OBJS)
	$(CC) -o complexSum $(OBJS) $(DEPFLAGS)

$(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

$(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)
```

Makefile V5：将各类文件放到相应文件夹

make过后

```
.
├── complexSum
├── include
│   ├── operations.h
│   └── utils.h
├── Makefile
├── obj
│   ├── main.o
│   ├── operations.o
│   └── utils.o
└── src
    ├── main.c
    ├── operations.c
    └── utils.c

3 directories, 10 files
```

```
IDIR=include
ODIR=obj
SDIR=src

CC=gcc

_OBJS=main.o utils.o operations.o
OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
_MAINDEPS=utils.h operations.h
MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
CFLAGS=-c -o
DEPFLAGS=-I$(IDIR)

all: $(OBJS)
	$(CC) -o complexSum $(OBJS) $(DEPFLAGS)

$(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

$(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)
```

Makefile V5：增加clean目标

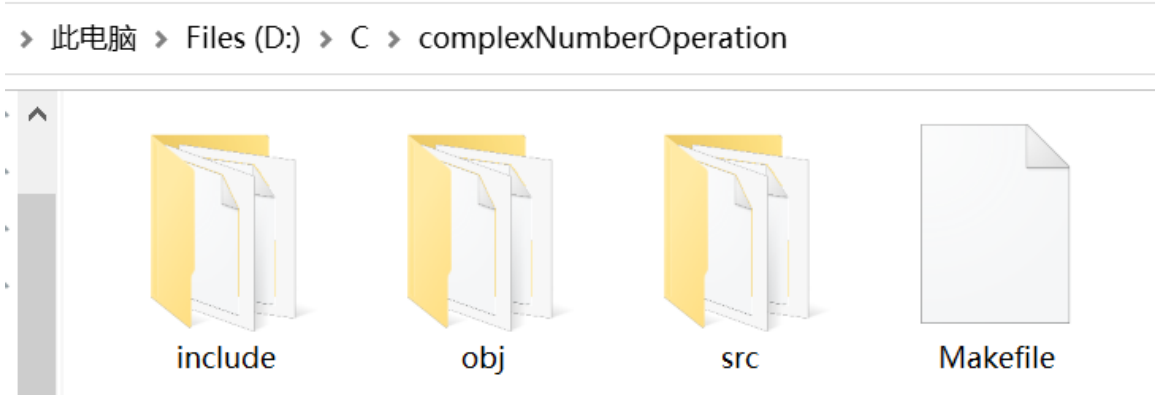
```
IDIR=include
ODIR=obj
SDIR=src

CC=gcc

_OBJS=main.o utils.o operations.o
OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
_MAINDEPS=utils.h operations.h
MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
CFLAGS=-c -o
DEPFLAGS=-I$(IDIR)
BINARY=complexSum

all: $(OBJS)
	$(CC) -o $(BINARY) $(OBJS) $(DEPFLAGS)

$(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

$(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

clean:
	rm -f $(OBJS)
	rm $(BINARY)
```

Makefile V5：增加clean目标

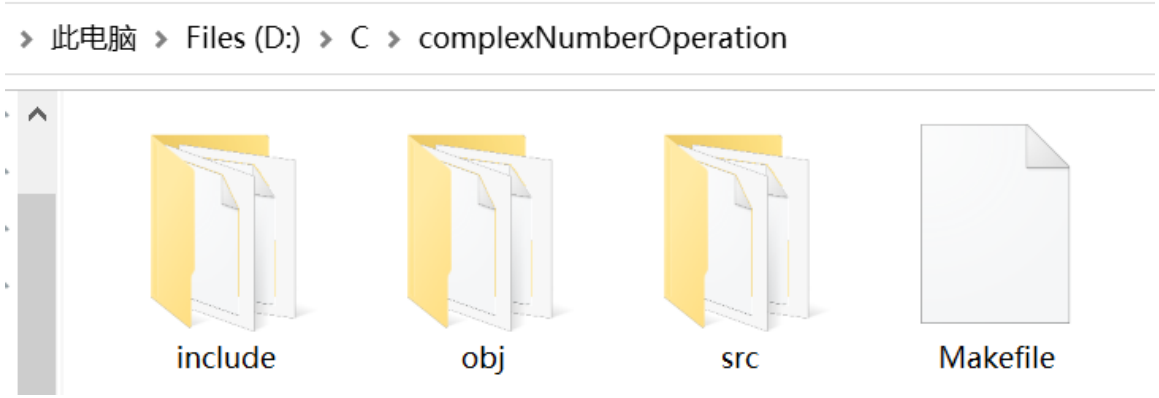
```
IDIR=include
ODIR=obj
SDIR=src

CC=gcc

_OBJS=main.o utils.o operations.o
OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
_MAINDEPS=utils.h operations.h
MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
CFLAGS=-c -o
DEPFLAGS=-I$(IDIR)
BINARY=complexSum

all: $(OBJS)
	$(CC) -o $(BINARY) $(OBJS) $(DEPFLAGS)

$(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

$(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
	$(CC) $(CFLAGS) $@ $< $(DEPFLAGS)

clean:
	rm -f $(OBJS)
	rm $(BINARY)
```

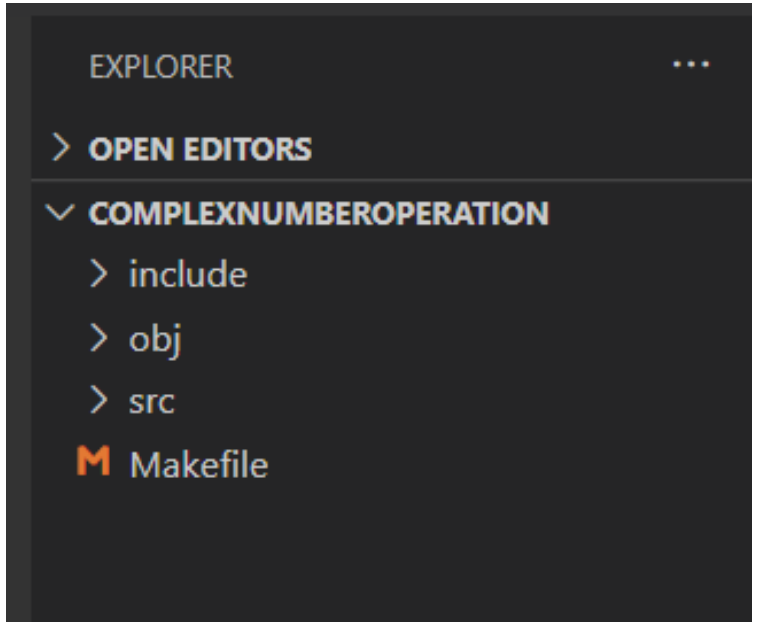
```
luxuesong@10-24-0-196:~/complexNumberOperation$ make clean
rm -f  obj/main.o  obj/utils.o  obj/operations.o
rm complexSum
luxuesong@10-24-0-196:~/complexNumberOperation$ tree
.
├── include
│   ├── operations.h
│   └── utils.h
├── Makefile
├── obj
└── src
    ├── main.c
    ├── operations.c
    └── utils.c

3 directories, 6 files
luxuesong@10-24-0-196:~/complexNumberOperation$
```

VSCode中使用Makefile

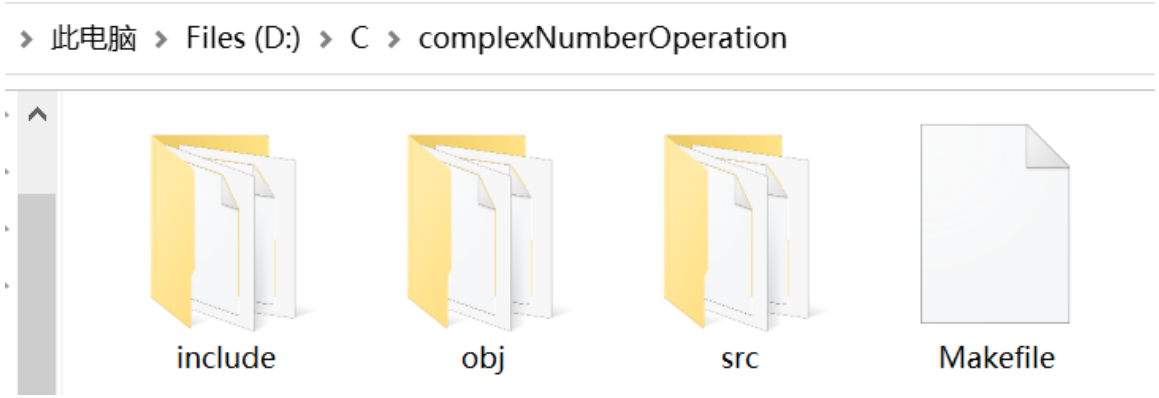


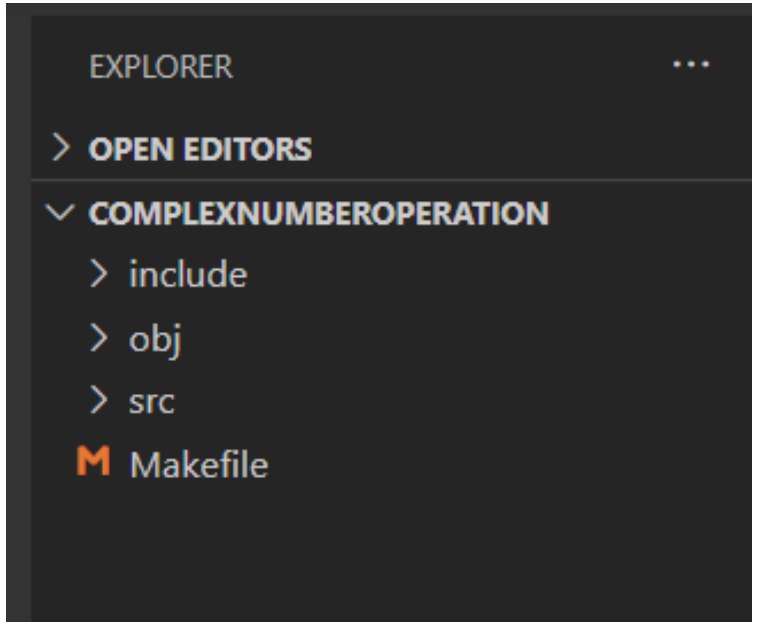
VSCode中使用Makefile





VSCode中使用Makefile





Makefile

```
1  IDIR=include
2  ODIR=obj
3  SDIR=src
4  CC=gcc
5  _OBJS=main.o utils.o operations.o
6  OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
7  DELOBJS=$(patsubst %, $(ODIR)\%, $(_OBJS))
8  _MAINDEPS=utils.h operations.h
9  MAINDEPS=$(patsubst %, $(IDIR)/%, $(_MAINDEPS))
10 CFLAGS=-c -o
11 DEPFLAGS=-I$(IDIR)
12 BINARY=complexSum.exe
13 RM=del
14
15 all: $(OBJS)
16     $(CC) -o $(BINARY) $(OBJS) $(DEPFLAGS)
17
18 $(ODIR)/main.o: $(SDIR)/main.c $(MAINDEPS)
19     $(CC) $(CFLAGS) $@ $< $(DEPFLAGS)
20
21 $(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
22     $(CC) $(CFLAGS) $@ $< $(DEPFLAGS)
23
24 clean:
25     $(RM) $(DELOBJS)
26     $(RM) $(BINARY)
```

VSCode中使用Makefile

Makefile

```
1  IDIR=include
2  ODIR=obj
3  SDIR=src
4  CC=gcc
5  _OBJS=main.o utils.o operations.o
6  OBJS=$(patsubst %, $(ODIR)/%, $(_OBJS))
7  DELOBJS=$(patsubst %, $(ODIR)\%, $(_OBJS))
                                    erations.h
                                %, $(IDIR)/%, $(_MAINDEPS))

                                e

                                Y) $(OBJS) $(DEPFLAGS)

                                IR)/main.c $(MAINDEPS)
                                $@ $< $(DEPFLAGS)

21 $(ODIR)/%.o: $(SDIR)/%.c $(IDIR)/%.h
22     $(CC) $(CFLAGS) $@ $< $(DEPFLAGS)
23
24 clean:
25     $(RM) $(DELOBJS)
26     $(RM) $(BINARY)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS D:\C\complexNumberOperation> mingw32-make
gcc -c -o obj/main.o src/main.c -Iinclude
gcc -c -o obj/utils.o src/utils.c -Iinclude
gcc -c -o obj/operations.o src/operations.c -Iinclude
gcc -o complexSum.exe  obj/main.o  obj/utils.o  obj/operations
.o -Iinclude
PS D:\C\complexNumberOperation>
```

Makefile参考资料

- https://seisman.github.io/how-to-write-makefile/

Makefile参考资料

- https://www.gnu.org/software/make/manual/make.html

GNU make

Short Table of Contents

静态链接 vs 动态链接

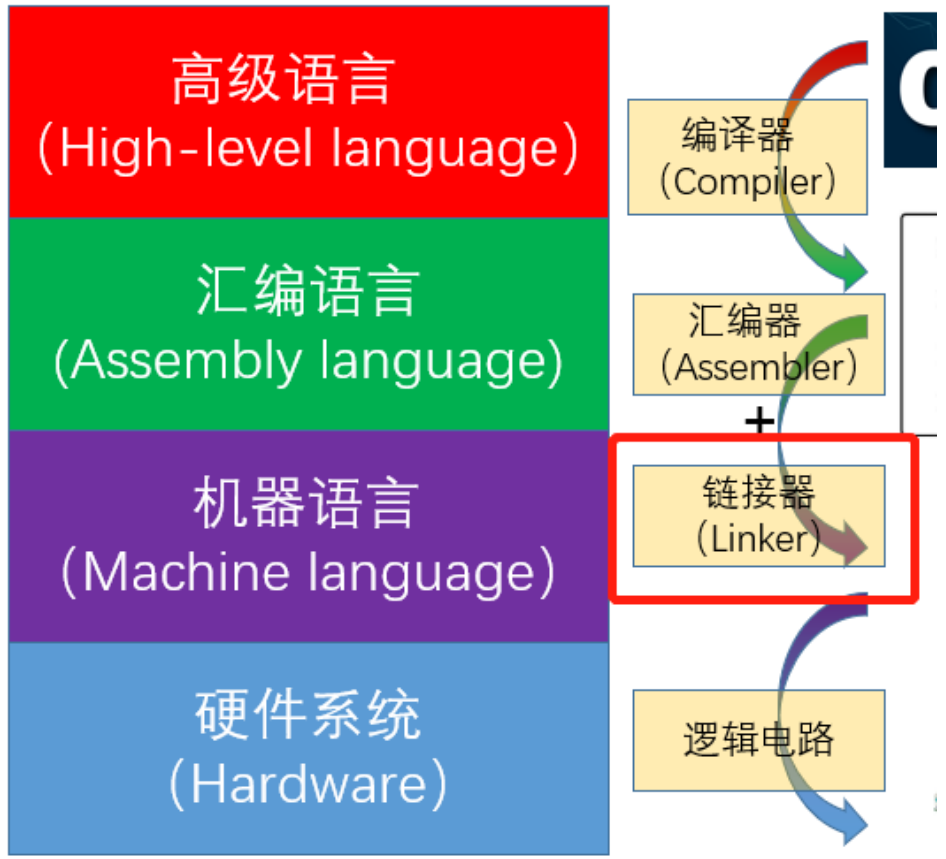
- 链接：将已经编译好的二进制文件组合成可执行文件，例如

```
gcc -o complexSum main.o utils.o operations.o
```

- 常用的目标文件会转换成库文件，以便随时调用。库文件分为静态库文件（.a, .lib）和动态库文件(.so, .dll)，相应的链接方式称为静态链接和动态链接

- 静态链接：编译可执行文件时进行链接

- 动态链接：执行可执行文件时进行调用



gcc中链接option： -l 和 -L

-l: 指定库文件

-L: 指定库文件所在文件夹

```
luxuesong@10-24-0-196:~/complexNumberOperation$ ls
libs  main.c  operations.c  operations.h  utils.c  utils.h
luxuesong@10-24-0-196:~/complexNumberOperation$ gcc -c utils.c
luxuesong@10-24-0-196:~/complexNumberOperation$ ar rcs libutils.a utils.o
luxuesong@10-24-0-196:~/complexNumberOperation$ mv libutils.a libs/
luxuesong@10-24-0-196:~/complexNumberOperation$ tree
.
├── libs
│   └── libutils.a
├── main.c
├── operations.c
├── operations.h
├── utils.c
├── utils.h
└── utils.o

1 directory, 7 files
luxuesong@10-24-0-196:~/complexNumberOperation$ gcc main.c operations.c -Llibs -lutils -o complexNumber
luxuesong@10-24-0-196:~/complexNumberOperation$ ./complexNumber 1 1 1 1 1 1
**************************
The Sum of the Array is 3+3i
**************************
luxuesong@10-24-0-196:~/complexNumberOperation$
```

gcc中链接option： -l 和 -L

-l: 指定库文件

-L: 指定库文件所在文件夹

```
luxuesong@10-24-0-196:~/complexNumberOperation$ ls
libs  main.c  operations.c  operations.h  utils.c  utils.h
luxuesong@10-24-0-196:~/complexNumberOperation$ gcc -c utils.c
luxuesong@10-24-0-196:~/complexNumberOperation$ ar rcs libutils.a utils.o
luxuesong@10-24-0-196:~/complexNumberOperation$ mv libutils.a libs/
luxuesong@10-24-0-196:~/complexNumberOperation$ tree
.
├── libs
│   └── libutils.a
├── main.c
├── operations.c
├── operations.h
├── utils.c
├── utils.h
└── utils.o

1 directory, 7 files
luxuesong@10-24-0-196:~/complexNumberOperation$ gcc main.c operations.c -Llibs -lutils -o complexNumber
```

Linux下系统库文件存放位置：/usr/lib

```
luxuesong@10-24-0-196:~/complexNumberOperation$
```