

# Makefile GCC

- 6. Makefile GCC

# 6. Makefile GCC

## 6.1

### 6.1.1

	PC		PC		ARM				
ARM		ARM	PC	ARM		ARM	GCC	ARM	ARM
	ARM		PC		PC		ARM		
1)	gcc		PC		X86				
2)		ARM							
	X86		ARM						
	"		"						
	PC								
	ARM								
	PC			ARM					
	PC		ARM			"		PC	ARM
						"			"

- 1) Ubuntu arm-linux-gcc
- 2) Windows ADS ARM armcc
- 3) Windows cygwin arm-elf-gcc

### 6.1.2

1)

2)

3)

### 6.1.3

main.c      gcc

main.c

```
01 #include <stdio.h>
02
03 int main()
04 {
05     printf("100ask\n");
06     return 0;
07 }
```

```
$ gcc main.c -o 100ask
$ ./100ask
100ask
$
```

```
$ chmod 777 100ask
$ ./100ask
./100ask: line 1: syntax error: unexpected "("
$
```

X86

Ubuntu   gcc

1.2   SDK

gcc   arm-linu:

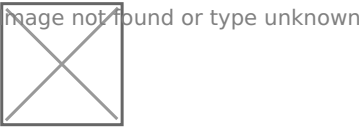
gcc   x86

arm-linux- gcc RSIC      ARM

# 6.2 gcc 1\_gcc \_\_gcc

## 6.2.1 gcc

C/C++ (preprocessing) (compilation) (assembly) (linking) 4



### 6.1.2.1

C/C++ “#” “#include” “#define” “#if” “#ifdef” (inclu

### 6.1.2.2

“.s”

### 6.1.2.3

“.o”

### 6.1.2.4

4

.c	C	
.C	C++	
.cc	C++	
.cxx	C++	
.m	Objective-C	
.i	C	
.ii	C++	
.s		

.S		
.h		

(linker)

.o (Object file OBJ )

.a (Archive file)

“-c” “-S” “-E” ( ) .o “-l” (

6.2.2 gcc

gcc

```
gcc [ ]
```

gcc c gcc gcc gcc

gcc mian.c

main.c:

```
01 #include <stdio.h>
02
03#define HUNDRED 100
04
05int main()
06{
07    printf("%d ask\n", HUNDRED);
08    return 0;
09}
```

: Git NoosProgramProject/ 6\_Makefile GCC/001\_gcc\_01001\_gcc\_01

6.2.2.1

gcc gcc

1 (-E)

C/C++ “#” “#include” “#define” “#if” “#ifdef” (inclu

```
$ gcc -E main.c -o main.i
```

main.i main.i

```
extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;

extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));

# 942 "/usr/include/stdio.h" 3 4

# 2 "main.c" 2

# 5 "main.c"

int main()

{

printf("%d ask\n",100);

return 0;

}
```

printf      HUNDRED

2      (-S)

C/C++ (      ".i" )"      "

```
$ gcc -S main.c -o main.s
```

main.s main.s

```
1      .file    "main.c"
2      .text
3      .section        .rodata
4 .LC0:
5      .string "%d ask\n"
6      .text
7      .globl  main
```

```

8      .type    main, @function
9 main:
10 .LFB0:
11      .cfi_startproc
12      pushq   %rbp
13      .cfi_def_cfa_offset 16
14      .cfi_offset 6, -16
15      movq    %rsp, %rbp
16      .cfi_def_cfa_register 6
17      movl    $100, %esi
18      leaq    .LC0(%rip), %rdi
19      movl    $0, %eax
20      call    printf@PLT
21      movl    $0, %eax
22      popq    %rbp
23      .cfi_def_cfa 7, 8
24      ret
25      .cfi_endproc
26 .LFE0:
27      .size    main, .-main
28      .ident   "GCC: (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0"
29      .section .note.GNU-stack,"",@progbits

```

3 (-c)

“.s”

Linux

ELF

(OBJ )

```
$ gcc -c main.c -o main.o
```

main.o( )

## 6.2.2.2

gcc

1 *filename*

-o*filename*

```
$ gcc main.c -o main
```

main

```
$ ls
main.c main
$ ./main
$ 100 ask
```

a.out

```
$ gcc main.c
```

a.out

```
$ ls
a.out main.c
$ ./a.out
$ 100 ask
```

2 -Wall

## Git NoosProgramProject/(6\_Makefile GCC/001\_gcc\_02)

main.c:

```
01 #include <stdio.h>
02
03 #define HUNDRED 100
04
05 int main()
06 {
07     int a = 0;
08     printf("%d ask\n", HUNDRED);
09     return 0;
10 }
```

-Wall

```
$ gcc main.c -o main.c
```

-Wall

```
$ gcc main.c -Wall -o main.c
```

```
main.c: In function 'main':
```



```
main.c:7:6: warning: unused variable 'a' [-Wunused-variable]
```

```
int a=0;
```

```
^
```

## 6.2.2.3

*dirname*

*dirname*                      gcc                      **Git**                      test.h    main.c  
**NoosProgramProject/(6\_Makefile GCC/001\_gcc\_03)**

```
$ tree
```

```
.
```

```
└─ inc
```

```
  └─ test.h
```

```
└─ main.c
```

```
1 directory, 2 files
```

```
$
```

test.h:

```
01 #ifndef __TEST_H
```

```
02 #define __TEST_H
```

```
03 /*
```

```
04    code
```

```
05 */
```

```
06 #endif
```

```
$ gcc main.c -I inc -o main
```

```
$ gcc main.c -o main
```

```
main.c:2:18: fatal error: test.h: No such file or directory
```

```
compilation terminated.
```

test.h

## 6.2.2.3

gcc 2\_

1) *dirname-L*

*dirname*

2) *name-l*

*name.a libname.so -lm libm.so*

3) *-static*

## 6.2.2.4

gcc *level* 001 2 3 s -O

1 -O

2 -O2 O2

3 -Os

## 6.2.2.5

```
**git (6_Makefile GCC/001_gcc_02)**
```

gcc

-g GDB

,

```
$ gcc main.c -g -o main
```

GDB :

1 run

run

run

run [ ]

```
$ gdb -q main<---
Reading symbols from output...done.
(gdb) run<---
Starting program: /home/100ask/makefile/
100 ask
[Inferior 1 (process 7425) exited normally]
(gdb)
```

2 list

list

list

list [ ]

```
(gdb) list 1<--- 10
#include <stdio.h>

#define HUNDRED 100

int main()
{
    int a = 100;

    printf("%d ask\n", HUNDRED);
    return 0;
```

```
(gdb) <Enter>{}<--- Enter    10
}
(gdb)
```

3

1 break                    break            break < > | < >

```
(gdb) break 7

Breakpoint 1 at 0x40052e: file main.c, line 7.

(gdb)
```

2 info break

```
(gdb) info break

Num  Type    Disp Enb Address          What
---  ---
1   breakpoint keep y  0x000000000040052e in main at main.c:7

(gdb)
```

3 delete breakpoint                    delete breakpoint            delete breakpoint < >

```
(gdb) delete breakpoint 1

(gdb) info break

No breakpoints or watchpoints.

(gdb)
```

4

1 print                    print            print[/ ] < >

2 display                    display            display < >

3 step next                    step next            step < > next < >

4 continue

```

(gdb) break 7
Breakpoint 1 at 0x40052e: file main.c, line 7.
(gdb) break 9
Breakpoint 2 at 0x400535: file main.c, line 9.
(gdb) run
Starting program: /home/100ask/makefile/

Breakpoint 1, main () at main.c: 7
7 int a = 100;
(gdb) continue
Continuing.

Breakpoint 2, main () at main.c: 9
9 printf("%d ask\n", HUNDRED);
(gdb) print a
$1 = 100
(gdb)

```

## 6.2.3

gcc

main.c

```

01 #include <stdio.h>
02 #include "hander.h"
03
04 int main()
05 {
06     float a = 0.0;
07     int b = a;
08     char c = 'a'
09
10     printf("100ask: \n", a);
11
12     return 0;
13 }

```

2

8

10            a

-Wall

```
$ gcc main.c -Wall -o output
main.c: In function 'main':
main.c:2:20: fatal error: hander.h: No such file or directory
compilation terminated.
```

hander.h

hander.h

hander.h

```
$ gcc -Wall main.c -o output
main.c: In function 'main':
main.c:10:2: error: expected ',' or ';' before 'printf'
    printf("100ask: \n",a);
    ^
main.c:8:7: warning: unused variable 'c' [-Wunused-variable]
    char c = 'a'
         ^
main.c:7:6: warning: unused variable 'b' [-Wunused-variable]
    int b = a;
         ^
```

10 printf

7 8

printf

```
$ gcc -Wall main.c -o output
main.c: In function 'main':
main.c:8:9: warning: too many arguments for format [-Wformat-extra-args]
    printf("100ask: \n",a);
    ^
```

printf

a

a

```
$ gcc -Wall main.c -o output
$ tree
.
```

```
|— main.c
|— output
```

## 6.3 gcc 2\_

OBJ OBJ

```
$ gcc main.c -c
$ gcc -o output main.o
$ gcc -o output_static main.o --static
$ ls -alh
drwxrwxr-x 2 tym tym 4.0K 2  20 07:27 .
drwxrwxr-x 6 tym tym 4.0K 2  20 07:25 ..
-rw-rw-r-- 1 tym tym  96 2  20 07:25 main.c
-rw-rw-r-- 1 tym tym 1.5K 2  20 07:26 main.o
-rwxrwxr-x 1 tym tym 8.5K 2  20 07:27 output
-rwxrwxr-x 1 tym tym 892K 2  20 07:27 output_static
```

output\_static output

### 6.3.1

main.c add.c add.h

main.c:

```
#include <stdio.h>
#include "add.h"

int main(int argc, char *argv[])
{
    printf("%d\n", add(10, 10));
}
```

```
printf("%d\n", add(20, 20));  
return 0;  
}
```

add.c:

```
#include "add.h"  
  
int add(int a, int b)  
{  
    return a + b;  
}
```

add.h:

```
#ifndef __ADD_H  
#define __ADD_H  
  
int add(int a, int b);  
  
#endif
```

\*\* \*\* Git NoosProgramProject/(6\_Makefile GCC/001\_gcc\_04)

## 6.3.1.1

“libxxx.a”

1.

2.

1.

2.

```
$ gcc add.c -o add.o -c  
$ ar -rc libadd.a add.o
```



```
$ gcc main.c -o output -ladd -L.
```

```
$ ./output
20
40
```

## 6.3.1.2

“libxxx.so”

\*\* \*\*

\*\* \*\*

```
$ gcc -shared -fPIC lib.c -o libtest.so
```

```
$ sudo cp libtest.so /usr/lib/
```

```
$ gcc main.c -L. -ltest -o output
```

```
$ ./output
20
40
```

# 6.4 Makefile

## 6.4.1 Makefile?

GCC

C

gcc

## 6.4.2 Makefile

Makefile

Makefile

main.c sub.c sub.h ac

main.c

```
#include <stdio.h>
#include "add.h"
#include "sub.h"

int main()
{
    printf("100 ask, add: %d\n", add(10, 10));
    printf("100 ask, sub: %d\n", sub(20, 10));
    return 0;
}
```

add.c

```
#include "add.h"

int add(int a, int b)
{
    return a + b;
}
```

add.h

```
#ifndef __ADD_H
#define __ADD_H

int add(int a, int b);

#endif
```

sub.c

```
#include "sub.h"

int sub(int a, int b)
{
```

```
    return a - b;  
}
```

sub.h

```
#ifndef __SUB_H  
#define __SUB_H  
  
int sub(int a, int b);  
  
#endif
```

## Git NoosProgramProject/(6\_Makefile GCC/001\_Makefile\_01)

gcc

```
$ gcc main.c sub.c add.c -o output  
$ ls  
add.c  add.h  main.c  output  sub.c  sub.h  
$ ./output  
100 ask, add: 20  
100 ask, sub: 10
```

gcc     main.c sub.c add.c

output

```
$ gcc -c main.c  
$ gcc -c sub.c  
$ gcc -c add.c  
$ gcc main.o sub.o add.o -o output
```

```
$ gcc -c add.c  
$ gcc main.o sub.o add.o -o output
```

1

2

## Makefile

## Makefile

## Makefile

```

output: main.o add.o sub.o
    gcc -o output main.o add.o sub.o
main.o: main.c
    gcc -c main.c
add.o: add.c
    gcc -c add.c
sub.o: sub.c
    gcc -c sub.c

clean:
    rm *.o output

```

## Makefile

## make

## make

## Makefile

```

$ ls
add.c  add.h  main.c  Makefile  sub.c  sub.h
$ make
gcc -c main.c
gcc -c add.c
gcc -c sub.c
gcc -o output main.o add.o sub.o
$ ls
add.c  add.h  add.o  main.c  main.o  Makefile  output  sub.c  sub.h  sub.o

```

## make

## .o

## make

```

$ make
make: 'output' is up to date.

```

## make

## add.c

```

$ make
gcc -c add.c
gcc -o output main.o add.o sub.o

```

## .C

# 6.4.3 Makefile

## 6.4.3.1

Makefile    Makefile    makefile    Makefile    Makefile    makefile.linux

```
make -f makefile.linux
```

## 6.4.3.2

target	prerequisites
[ Tab]	command

1 target

2 prerequisites    target

3 command

target    prerequisites    command

```
$ gcc main.c -o main
```

Makefile

```
01 main: main.c

02 gcc main.c -o main
```

**\*\*MakefileTab \*\***

## 6.4.3.3

1

2

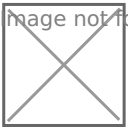
image not found or type unknown



1

2

image not found or type unknown



## Makefile

```
output: main.o add.o sub.o
    gcc -o output main.o add.o sub.o
main.o: main.c
    gcc -c main.c
add.o: add.c
    gcc -c add.c
sub.o: sub.c
    gcc -c sub.c

clean:
    rm *.o output
```

```
$ make
gcc -c main.c
gcc -c add.c
gcc -c sub.c
gcc -o output main.o add.o sub.o
```

make

add.c

make

```
$ make
gcc -c add.c
```

```
gcc -o output main.o add.o sub.o
```

make

## 6.5 Makefile

### 6.5.1

Makefile

Makefile

```
**      **      C
```

```
DIR = ./100ask/
```

```
**      **
```

```
F00 = $(DIR)
```

Makefile

Makefile

```
output: main.o add.o sub.o
        gcc -o output main.o add.o sub.o
main.o: main.c
        gcc -c main.c
add.o: add.c
        gcc -c add.c
sub.o: sub.c
        gcc -c sub.c

clean:
        rm *.o output
```

```
#Makefile
OBJ = main.o add.o sub.o
output: $(OBJ)
        gcc -o output $(OBJ)
main.o: main.c
```

```

    gcc -c main.c
add.o: add.c
    gcc -c add.c
sub.o: sub.c
    gcc -c sub.c

clean:
    rm $(OBJ) output

```

Makefile      Makefile      ‘#’      C      OBJ      “main.o add.o su

Makefile      ‘=’      ‘:=’ ‘?='

## 6.5.2.1      ‘=’

‘=’      Makefile

```

01 ¶ PARA = 100
02 ¶ CURPARA = $( PARA)
03 ¶ PARA = ask
04
05 ¶ print:
06 ¶ @echo $( CURPARA)

```

PARA      “100”      CURPARA      PARA      CURPARA      PARA      PARA

“make print”      Makefile

```

$ make print
ask

```

CURPARA      “100”      PARA      “=”      ,

C

```

01 ¶ int a = 10;
02 ¶ int *b = &a;
03 ¶ a=20;

```

## 6.5.2.2      ‘:=’

‘:=’      Makefile



```
01 PARA = 100
02 CURPARA := $( PARA)
03 PARA = ask
04
05 print:
06 echo $( CURPARA)
```

Makefile      “=”      “:=”

```
$ make print
100
$
```

CURPARA      “100” “=” “:=”      “:=”

## 6.5.2.3      ‘? =’

Makefile      ‘? =’

Makefile

```
PARA = 100
PARA ?= ask

print:
echo $( PARA)
```

```
$ make print
100
$
```

Makefile:

```
PARA ?= ask

print:
echo $( PARA)
```

```
$ make print
ask
$
```

PARA "ask"

## 6.5.2.4 '+='

Makefile , "+="

```
01[OBJ] = main.o add.o
02[OBJ] += sub.o
```

OBJ "main.o add.o sub.o" "+="

## 6.5.2

CC PWD CLFAG

1 CPPFLAGS -I

2 CFLAGS -Wall -g -c

3 LDFLAGS -L -l

CC cc gcc CC=gcc

```
01[OBJ] = main.o add.o sub.o
02[output: $(OBJ)
03[ gcc -o output $(OBJ)
04[main.o: main.c
05[ gcc -c main.c
06[add.o: add.c
07[ gcc -c add.c
08[sub.o: sub.c
09[ gcc -c sub.c
10[
11[clean:
12[ rm $(OBJ) output
```

```

01 CC = gcc
02 OBJ = main.o add.o sub.o
03 output: $(OBJ)
04 $(CC) -o output $(OBJ)
05 main.o: main.c
06 $(CC) -c main.c
07 add.o: add.c
08 $(CC) -c add.c
09 sub.o: sub.c
10 $(CC) -c sub.c
11
12 clean:
13 rm $(OBJ) output

```

CC

gcc

## 6.5.3

Makefile

Makefile

1 \$@

2 \$<

3 \$^

:

```

01 CC = gcc
02 OBJ = main.o add.o sub.o
03 output: $(OBJ)
04 $(CC) -o $@ $^
05 main.o: main.c
06 $(CC) -c $<
07 add.o: add.c
08 $(CC) -c $<
09 sub.o: sub.c
10 $(CC) -c $<
11

```

```
12 clean:
```

```
13 rm $(OBJ) output
```

```
4 $^ OBJ main.o add.o sub.o $< main.c add.c sub.c $@ output
```

## 6.5.4

%

```
01 CC = gcc
```

```
02 OBJ = main.o add.o sub.o
```

```
03 output: $(OBJ)
```

```
04 $(CC) -o $@ $^
```

```
05 %.o: %.c
```

```
06 $(CC) -c $<
```

```
07
```

```
08 clean:
```

```
09 rm $(OBJ) output
```

%.o: %.

1.main.o main.c

2.add.o add.c

3.sub.o sub.c

## 6.5.5

make

make

make clean

```
$make
```

```
gcc -c main.c
```

```
gcc -c add.c
```

```
gcc -c sub.c
```

```
gcc -o output main.o add.o sub.o
```

```
$make clean
```

```
rm *.o output
```

Makefile

clean

make make clean

```
$touch clean
$make
gcc -c main.c
gcc -c add.c
gcc -c sub.c
gcc -o output main.o add.o sub.o
$make clean
make: 'clean' is up to date.
```

clean

Makefile

Makefile clean

```
01 CC = gcc
02 OBJ = main.o add.o sub.o
03 output: $(OBJ)
04 $(CC) -o $@ $^
05 %.o: %.c
06 $(CC) -c $<
07
08 PHONY: clean
09 clean:
10 rm $(OBJ) output
```

```
$make
gcc -c main.c
gcc -c add.c
gcc -c sub.c
gcc -o output main.o add.o sub.o
$make clean
rm *.o output
```

rm

make

make

- 1.
2. Makefile

## 6.5.6 Makefile

Makefile

(wildcard patsubst)

Makefile

src

```
.
├─ Makefile
└─ src
   ├── 100.c
   └─ ask.c
```

```
**      **      Git   NoosProgramProject/(6_Makefile GCC/001_Makefile_02)
```

### 6.5.6.1 wildcard

+

```
$( wildcard      )
```

```
01[ SRC = $( wildcard ./src/*.c )
02
03[ print:
04[      @echo $( SRC)
```

make

```
$ make
./src/ask.c ./src/100.c
```

./src .c SRC SRC ./src/ask.c ./src/100.c

### 6.5.6.2 patsubst

+ + :

```
$( patsubst      ,      ,      )
```

./src .c .o obj

```
SRC = $( wildcard ./src/*.c )
OBJ = $( patsubst %.c, %.o, $( SRC) )
```

```
print:
  @echo $(OBJ)
```

make

```
$ make
./src/ask.o ./src/100.o
```

```
      .c      .o      OBJ      ./src/ask.o ./src/100.o
```

## 6.6 Makefile

inc src \*\*

```
$ tree
.
├── inc
│   ├── add.h
│   └── sub.h
├── Makefile
└── src
    ├── add.c
    ├── main.c
    └── sub.c
```

Makefile

```
01[SOURCE = $(wildcard ./src/*.c)
02[OBJECT = $(patsubst %.c, %.o, $(SOURCE))
03[
04[INCLUDES = -I ./inc
05[
06[TARGET = 100ask
07[CC      = gcc
08[CFLAGS  = -Wall -g
09[
10[$(TARGET): $(OBJECT)
11[      @mkdir -p output/
```

```

12      $(CC) $^ $(CFLAGS) -o output/$(TARGET)
13
14%.o: %.c
15      $(CC) $(INCLUDES) $(CFLAGS) -c $< -o $@
16
17.PHONY: clean
18clean:
19      @rm -rf $(OBJECT) output/

```

```

1      src .c      SOURCE
2  ./src .c      .o      OBJECT
3
4  -I      INCLUDES
5
6      100ask      TARGET
7  CC      cc      gcc
8
9      gdb      CFLAGS
10
11      output
12      100ask      output
13
14
15
16
17      clean
18
19      make clean
20
21      :

```

```

$ make
gcc -I ./inc -c src/main.c -o src/main.o
gcc -I ./inc -c src/add.c -o src/add.o
gcc -I ./inc -c src/sub.c -o src/sub.o
gcc src/main.o src/add.o src/sub.o -o output/100ask
$ tree
.
├── inc
│   └── add.h

```



```

|   └─ sub.h
└─ Makefile
└─ output
|   └─ 100ask
└─ src
    └─ add.c
    └─ add.o
    └─ main.c
    └─ main.o
    └─ sub.c
    └─ sub.o

```

## Makefile

```

01 VERSION = 1.0.0
02 SOURCE  = $(wildcard ./src/*.c)
03 OBJECT   = $(patsubst %.c, %.o, $(SOURCE))
04
05 INCLUDED = -I ./inc
06
07 TARGET   = 100ask
08 CC       = gcc
09 CFLAGS   = -Wall -g
10
11 $(TARGET): $(OBJECT)
12     @mkdir -p output/
13     $(CC) $^ $(CFLAGS) -o output/$(TARGET)_$(VERSION)
14
15 %.o: %.c
16     $(CC) $(INCLUDED) $(CFLAGS) -c $< -o $@
17
18 PHONY: clean
19 clean:
20     @rm -rf $(OBJECT) output/

```

1           VERSION

13

```
$ tree
```

```
.  
├── inc  
│   ├── add.h  
│   └── sub.h  
├── Makefile  
├── output  
│   └── 100ask_1.0.0  
└── src  
    ├── add.c  
    ├── add.o  
    ├── main.c  
    ├── main.o  
    ├── sub.c  
    └── sub.o
```