Computer Architecture Assignment 1

2018320205 신대성

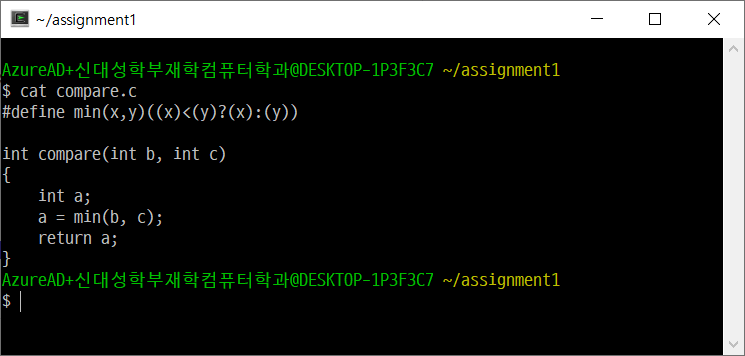
# Explanation of the role of Makefile in Eclipse project

Makefile is setting file for ‘make’ which is program in linux. To make simple, it’s just set of commands in linux. When we execute ‘make’ program with ‘Makefile’ in linux, it compiles with commands in that setting file.

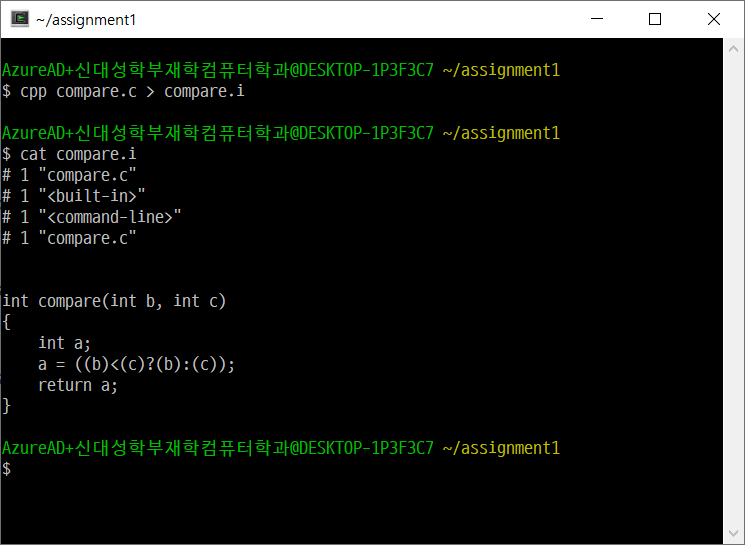
In Eclipse project, Makefile has similar role but has a little difference. We use build button to use ‘Makefile’ setting, not using ‘make’ command.

# Outputs of each compilation step in the native compilation

1. Example code(compare.c)



1. Preprocessing(compare.i)



1. Compilation(compare.s)

$gcc -S compare.i

.file "compare.c"

.text

.globl \_compare

.def \_compare; .scl 2; .type 32; .endef

\_compare:

LFB0:

.cfi\_startproc

pushl %ebp

.cfi\_def\_cfa\_offset 8

.cfi\_offset 5, -8

movl %esp, %ebp

.cfi\_def\_cfa\_register 5

subl $16, %esp

movl 8(%ebp), %eax

cmpl %eax, 12(%ebp)

cmovle 12(%ebp), %eax

movl %eax, -4(%ebp)

movl -4(%ebp), %eax

leave

.cfi\_restore 5

.cfi\_def\_cfa 4, 4

ret

.cfi\_endproc

LFE0:

.ident "GCC: (GNU) 7.4.0"

1. Assembler(compare.o > compare.dump)

$as compare.s -o compare.o

$objdump -SD compare.o

compare.o: file format pe-i386

Disassembly of section .text:

00000000 <\_compare>:

0: 55 push %ebp

1: 89 e5 mov %esp,%ebp

3: 83 ec 10 sub $0x10,%esp

6: 8b 45 08 mov 0x8(%ebp),%eax

9: 39 45 0c cmp %eax,0xc(%ebp)

c: 0f 4e 45 0c cmovle 0xc(%ebp),%eax

10: 89 45 fc mov %eax,-0x4(%ebp)

13: 8b 45 fc mov -0x4(%ebp),%eax

16: c9 leave

17: c3 ret

Disassembly of section .rdata$zzz:

00000000 <.rdata$zzz>:

0: 47 inc %edi

1: 43 inc %ebx

2: 43 inc %ebx

3: 3a 20 cmp (%eax),%ah

5: 28 47 4e sub %al,0x4e(%edi)

8: 55 push %ebp

9: 29 20 sub %esp,(%eax)

b: 37 aaa

c: 2e 34 2e cs xor $0x2e,%al

f: 30 00 xor %al,(%eax)

11: 00 00 add %al,(%eax)

...

Disassembly of section .eh\_frame:

00000000 <.eh\_frame>:

0: 14 00 adc $0x0,%al

2: 00 00 add %al,(%eax)

4: 00 00 add %al,(%eax)

6: 00 00 add %al,(%eax)

8: 01 7a 52 add %edi,0x52(%edx)

b: 00 01 add %al,(%ecx)

d: 7c 08 jl 17 <.eh\_frame+0x17>

f: 01 1b add %ebx,(%ebx)

11: 0c 04 or $0x4,%al

13: 04 88 add $0x88,%al

15: 01 00 add %eax,(%eax)

17: 00 1c 00 add %bl,(%eax,%eax,1)

1a: 00 00 add %al,(%eax)

1c: 1c 00 sbb $0x0,%al

1e: 00 00 add %al,(%eax)

20: 04 00 add $0x0,%al

22: 00 00 add %al,(%eax)

24: 18 00 sbb %al,(%eax)

26: 00 00 add %al,(%eax)

28: 00 41 0e add %al,0xe(%ecx)

2b: 08 85 02 42 0d 05 or %al,0x50d4202(%ebp)

31: 54 push %esp

32: c5 0c 04 lds (%esp,%eax,1),%ecx

35: 04 00 add $0x0,%al

...

# Outputs of each compilation step in the cross-compilation for MIPS

1. Example Code(compare.c) and Makefile

**#define** min(x, y) ((x) < (y) ? (x) : (y))

**int** **compare**(**int** b, **int** c)

{

**int** a;

a = min(b, c);

**return** a;

} // end of compare.c

...

all: testvec

testvec: testvec.o compare.o

$(LD) $(LDFLAGS) testvec.o compare.o -o testvec

$(OBJDUMP) -xS testvec > testvec.dump

$(OBJCOPY) -O binary testvec testvec.bin

./bin2hex.perl > testvec.hex

./hex2mif.perl

# ./mipsel-readelf -a testvec > testvec.r

# ./mipsel-nm testvec > testvec.n

testvec.o: testvec.s

$(AS) $(ASFLAGS) testvec.s -o testvec.o

compare.o: compare.c

$(CPP) compare.c > compare.i

$(CC) -Wall -S compare.i

$(AS) $(ASFLAGS) compare.s -o compare.o

# $(OBJDUMP) -xS compare.o > compare.dump (this part is executed on Cygwin not eclipse)

# $(CC) $(CCFLAGS) compare.c

...

1. Preprocessing(compare.i)

# 1 "compare.c"

# 1 "<built-in>"

# 1 "<command line>"

# 1 "compare.c"

int compare(int b, int c)

{

int a;

a = ((b) < (c) ? (b) : (c));

return a;

}

1. Compilation(compare.s)

**.file** 1 "compare.c"

**.section** .mdebug.abi32

.previous

**.text**

**.align** 2

**.globl** compare

.ent compare

**compare:**

.frame $fp,24,$31 # vars= 16, regs= 1/0, args= 0, gp= 0

.mask 0x40000000,-8

.fmask 0x00000000,0

**.set** noreorder

**.set** nomacro

addiu $sp,$sp,-24

sw $fp,16($sp)

move $fp,$sp

sw $4,24($fp)

sw $5,28($fp)

lw $2,24($fp)

nop

sw $2,12($fp)

lw $3,28($fp)

nop

sw $3,8($fp)

lw $4,8($fp)

lw $3,12($fp)

nop

slt $2,$3,$4

beq $2,$0,$L2

nop

lw $4,12($fp)

nop

sw $4,8($fp)

**$L2:**

lw $2,8($fp)

nop

sw $2,0($fp)

lw $2,0($fp)

move $sp,$fp

lw $fp,16($sp)

addiu $sp,$sp,24

j $31

nop

**.set** macro

**.set** reorder

.end compare

**.size** compare, .-compare

.ident "GCC: (GNU) 4.1.1"

1. Assembler(compare.o > compare.dump)

../../assignment1/compare.o: file format elf32-bigmips

../../assignment1/compare.o

architecture: mips:3000, flags 0x00000011:

HAS\_RELOC, HAS\_SYMS

start address 0x00000000

private flags = 1: [no abi set] [mips1] [not 32bitmode] [noreorder]

Sections:

Idx Name Size VMA LMA File off Algn

0 .text 00000074 00000000 00000000 00000034 2\*\*2

CONTENTS, ALLOC, LOAD, READONLY, CODE

1 .data 00000000 00000000 00000000 000000a8 2\*\*0

CONTENTS, ALLOC, LOAD, DATA

2 .bss 00000000 00000000 00000000 000000a8 2\*\*0

ALLOC

3 .reginfo 00000018 00000000 00000000 000000a8 2\*\*2

CONTENTS, READONLY, LINK\_ONCE\_SAME\_SIZE

4 .pdr 00000020 00000000 00000000 000000c0 2\*\*2

CONTENTS, RELOC, READONLY

5 .mdebug.abi32 00000000 00000000 00000000 000000e0 2\*\*0

CONTENTS, READONLY

6 .comment 00000012 00000000 00000000 000000e0 2\*\*0

CONTENTS, READONLY

SYMBOL TABLE:

00000000 l d .text 00000000 .text

00000000 l d .data 00000000 .data

00000000 l d .bss 00000000 .bss

00000000 l d .mdebug.abi32 00000000 .mdebug.abi32

00000000 l d .reginfo 00000000 .reginfo

00000000 l d .pdr 00000000 .pdr

00000000 l d .comment 00000000 .comment

00000000 l df \*ABS\* 00000000 compare.c

00000000 g F .text 00000074 compare

Disassembly of section .text:

00000000 <compare>:

0: 27bdffe8 addiu sp,sp,-24

4: afbe0010 sw s8,16(sp)

8: 03a0f021 move s8,sp

c: afc40018 sw a0,24(s8)

10: afc5001c sw a1,28(s8)

14: 8fc20018 lw v0,24(s8)

18: 00000000 nop

1c: afc2000c sw v0,12(s8)

20: 8fc3001c lw v1,28(s8)

24: 00000000 nop

28: afc30008 sw v1,8(s8)

2c: 8fc40008 lw a0,8(s8)

30: 8fc3000c lw v1,12(s8)

34: 00000000 nop

38: 0064102a slt v0,v1,a0

3c: 10400004 beqz v0,50 <compare+0x50>

40: 00000000 nop

44: 8fc4000c lw a0,12(s8)

48: 00000000 nop

4c: afc40008 sw a0,8(s8)

50: 8fc20008 lw v0,8(s8)

54: 00000000 nop

58: afc20000 sw v0,0(s8)

5c: 8fc20000 lw v0,0(s8)

60: 03c0e821 move sp,s8

64: 8fbe0010 lw s8,16(sp)

68: 27bd0018 addiu sp,sp,24

6c: 03e00008 jr ra

70: 00000000 nop

1. Linker(testvec > testvec.dump)

testvec: file format elf32-bigmips

testvec

architecture: mips:3000, flags 0x00000012:

EXEC\_P, HAS\_SYMS

start address 0x00000000

Program Header:

LOAD off 0x00000060 vaddr 0x00000000 paddr 0x00000000 align 2\*\*4

filesz 0x00000084 memsz 0x00000084 flags rwx

private flags = 1: [no abi set] [mips1] [not 32bitmode] [noreorder]

Sections:

Idx Name Size VMA LMA File off Algn

0 .text 00000084 00000000 00000000 00000060 2\*\*4

CONTENTS, ALLOC, LOAD, CODE

1 .reginfo 00000018 00000000 00000000 000000e4 2\*\*2

CONTENTS, READONLY, LINK\_ONCE\_SAME\_SIZE

2 .pdr 00000020 00000000 00000000 000000fc 2\*\*2

CONTENTS, READONLY

3 .comment 00000012 00000000 00000000 0000011c 2\*\*0

CONTENTS, READONLY

SYMBOL TABLE:

00000000 l d .text 00000000 .text

00000000 l d .reginfo 00000000 .reginfo

00000000 l d .pdr 00000000 .pdr

00000000 l d .comment 00000000 .comment

00000000 l df \*ABS\* 00000000 compare.c

00000010 g F .text 00000074 compare

Disassembly of section .text:

00000000 <compare-0x10>:

0: 0c000004 jal 10 <compare>

4: 00000000 nop

8: ac020054 sw v0,84(zero)

c: 00000000 nop

00000010 <compare>:

10: 27bdffe8 addiu sp,sp,-24

14: afbe0010 sw s8,16(sp)

18: 03a0f021 move s8,sp

1c: afc40018 sw a0,24(s8)

20: afc5001c sw a1,28(s8)

24: 8fc20018 lw v0,24(s8)

28: 00000000 nop

2c: afc2000c sw v0,12(s8)

30: 8fc3001c lw v1,28(s8)

34: 00000000 nop

38: afc30008 sw v1,8(s8)

3c: 8fc40008 lw a0,8(s8)

40: 8fc3000c lw v1,12(s8)

44: 00000000 nop

48: 0064102a slt v0,v1,a0

4c: 10400004 beqz v0,60 <compare+0x50>

50: 00000000 nop

54: 8fc4000c lw a0,12(s8)

58: 00000000 nop

5c: afc40008 sw a0,8(s8)

60: 8fc20008 lw v0,8(s8)

64: 00000000 nop

68: afc20000 sw v0,0(s8)

6c: 8fc20000 lw v0,0(s8)

70: 03c0e821 move sp,s8

74: 8fbe0010 lw s8,16(sp)

78: 27bd0018 addiu sp,sp,24

7c: 03e00008 jr ra

80: 00000000 nop