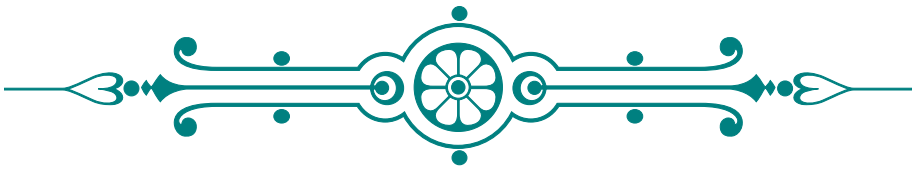

시스템 S/W 실습8

학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2016년 11월 16일

1. 다음에 주어진 C 프로그램 sum.c를 Linux 환경에서 컴파일해서 어셈블리 파일 및 object 파일을 생성하고, 어셈블리어 파일 sum.s는 editor로 확인하고, object 파일 sum.o는 objdump를 사용해서 확인하시오.

Listing 1: sum.c

```
int accum=0;
int sum(int x, int y) {
    int t = x + y;
    accum += t;
    return t;
}
```

Listing 2: sum.s

```
    .file   "sum.c"
    .text
    .globl  sum
    .type   sum, @function
sum:
.LFB0:
    .cfi_startproc
    leal    (%rdi,%rsi), %eax
    addl    %eax, accum(%rip)
    ret
    .cfi_endproc
.LFE0:
    .size   sum, .-sum
    .globl  accum
    .bss
    .align 4
    .type   accum, @object
    .size   accum, 4
accum:
    .zero   4
    .ident  "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609"
    .section    .note.GNU-stack,"",@progbits
```

Listing 3: objdump -d sum.o

```
sum.o:     file format elf64-x86-64
```

```
Disassembly of section .text:

0000000000000000 <sum>:
   0:   8d 04 37                lea    (%rdi,%rsi,1),%eax
   3:   01 05 00 00 00 00       add    %eax,0x0(%rip)        # 9 <sum+0x9>
   9:   c3                      retq
```

2. 다음에 주어진 main.c를 컴파일하여 main.o를 얻고, main.o와 sum.o를 link하여 실행파일 prog를 얻고, prog로서 Linux debugger program인 'gdb'를 사용해서 1명령씩 실행하면서 필요한 register들의 값을 확인하시오.

Listing 4: main.c

```
int main() {
    return sum(10,20);
}
```

Listing 5: gdb 실행

```
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from prog...done.
(gdb) r
Starting program: /home/zezeon/Programming/SIC/prog
[Inferior 1 (process 4311) exited with code 036]
(gdb) b sum
Breakpoint 1 at 0x4004d6
(gdb) disas sum()
(gdb) disas sum()[K[K
Dump of assembler code for function sum:
```

```
   0x00000000004004d6 <+0>: lea     (%rdi,%rsi,1),%eax
   0x00000000004004d9 <+3>: add     %eax,0x200b55(%rip)        # 0x601034 <accum>
   0x00000000004004df <+9>: retq
End of assembler dump.
(gdb) print [K[K[K[K /x $eax
(gdb) r
Starting program: /home/zezeon/Programming/SIC/prog

Breakpoint 1, 0x00000000004004d6 in sum ()
(gdb) p /x $eax
$1 = 0x0
(gdb) info frame
Stack level 0, frame at 0x7fffffffdd70:
 rip = 0x4004d6 in sum; saved rip = 0x4004f8
 called by frame at 0x7fffffffdd80
 Arglist at 0x7fffffffdd60, args:
 Locals at 0x7fffffffdd60, Previous frame's sp is 0x7fffffffdd70
 Saved registers:
  rip at 0x7fffffffdd68
(gdb) info registers
rax            0x0    0
rbx            0x0    0
rcx            0x0    0
rdx            0      0x7fffffffde68    140737488346728
rsi            0x14   20
rdi            0xa    10
rbp            0x400500 0x400500 <__libc_csu_init>
rsp            0      0x7fffffffdd68    0x7fffffffdd68
r8             0x400570 4195696
r9             0      0x7ffff7de78e0    140737351940320
r10            0x846  2118
r11            0      0x7ffff7a2e740    140737348036416
r12            0x4003e0 4195296
r13            0      0x7fffffffde50    140737488346704
r14            0x0    0
r15            0x0    0
rip            0x4004d6 0x4004d6 <sum>
eflags         0x202  [ IF ]
cs             0x33   51
ss             0x2b   43
```

```
ds             0x0   0
es             0x0   0
fs             0x0   0
gs             0x0   0
(gdb) x/20b sum
0x4004d6 <sum>: 0x8d    0x04    0x37    0x01    0x05    0x55    0x0b    0x20
0x4004de <sum+8>:  0x00    0xc3    0x48    0x83    0xec    0x08    0xbe    0x14
0x4004e6 <main+6>: 0x00    0x00    0x00    0xbf
(gdb) l
1     int main() {
2           return sum(10,20);
3     }
(gdb) stepi
0x00000000004004d9 in sum ()
(gdb) n
Single stepping until exit from function sum,
which has no line number information.
main () at main.c:3
3     }
(gdb)
 __libc_start_main   (main=0x4004e0 <main>, argc=1, argv=0 x7fffffffde58 ,
      init =<optimized out>, fini =<optimized out>,  rtld_fini =<optimized out>,
      stack_end=0 x7fffffffde48 ) at  ../ csu/ libc - start . c:325
(gdb) q
A debugging session is active .

      Inferior 1 [process 4354] will be killed .

Quit anyway? (y or n) y
```

소감

gdb나 컴파일러가 아직도 모르는 옵션들이 매우 많다는 것을 느꼈다.