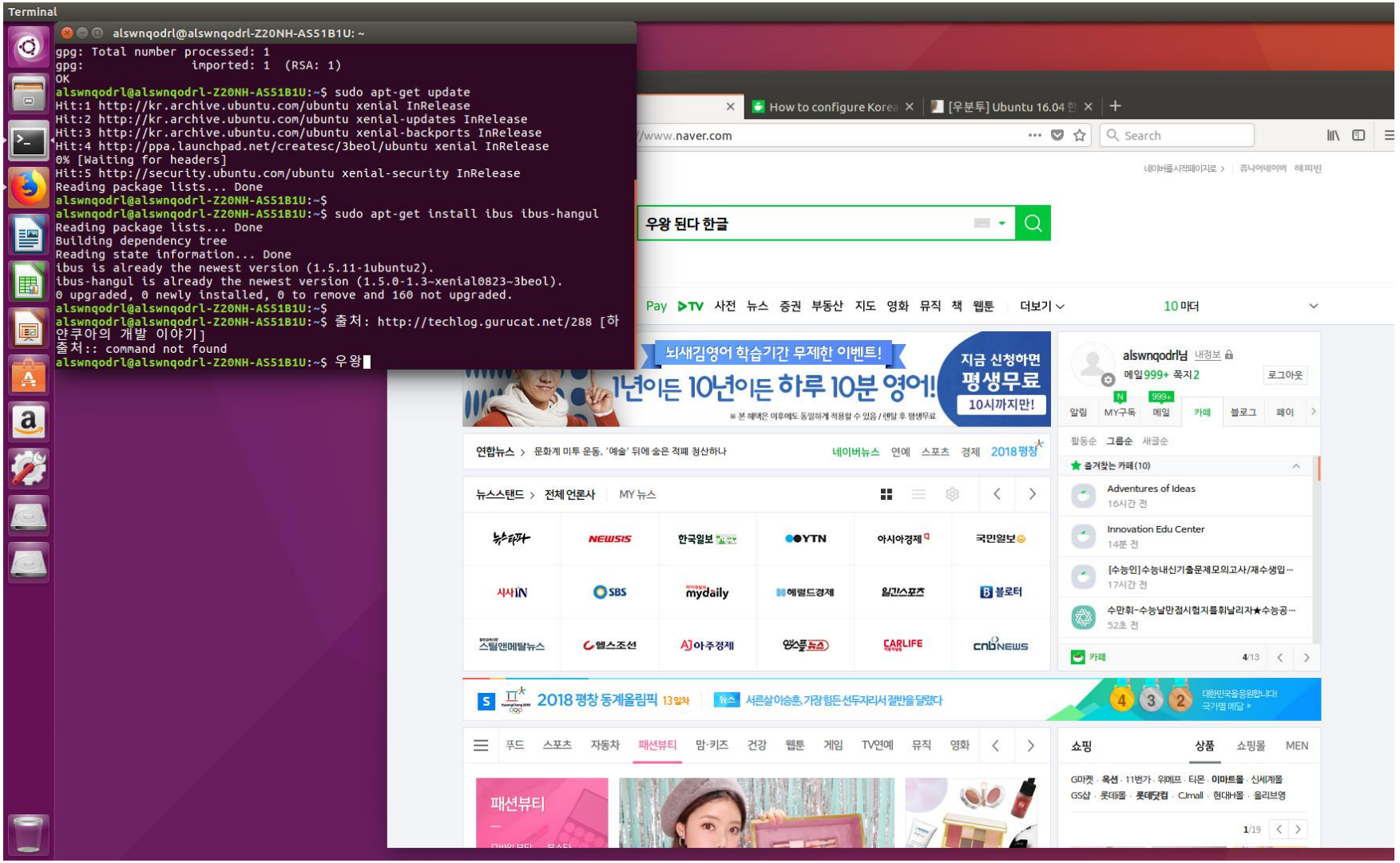
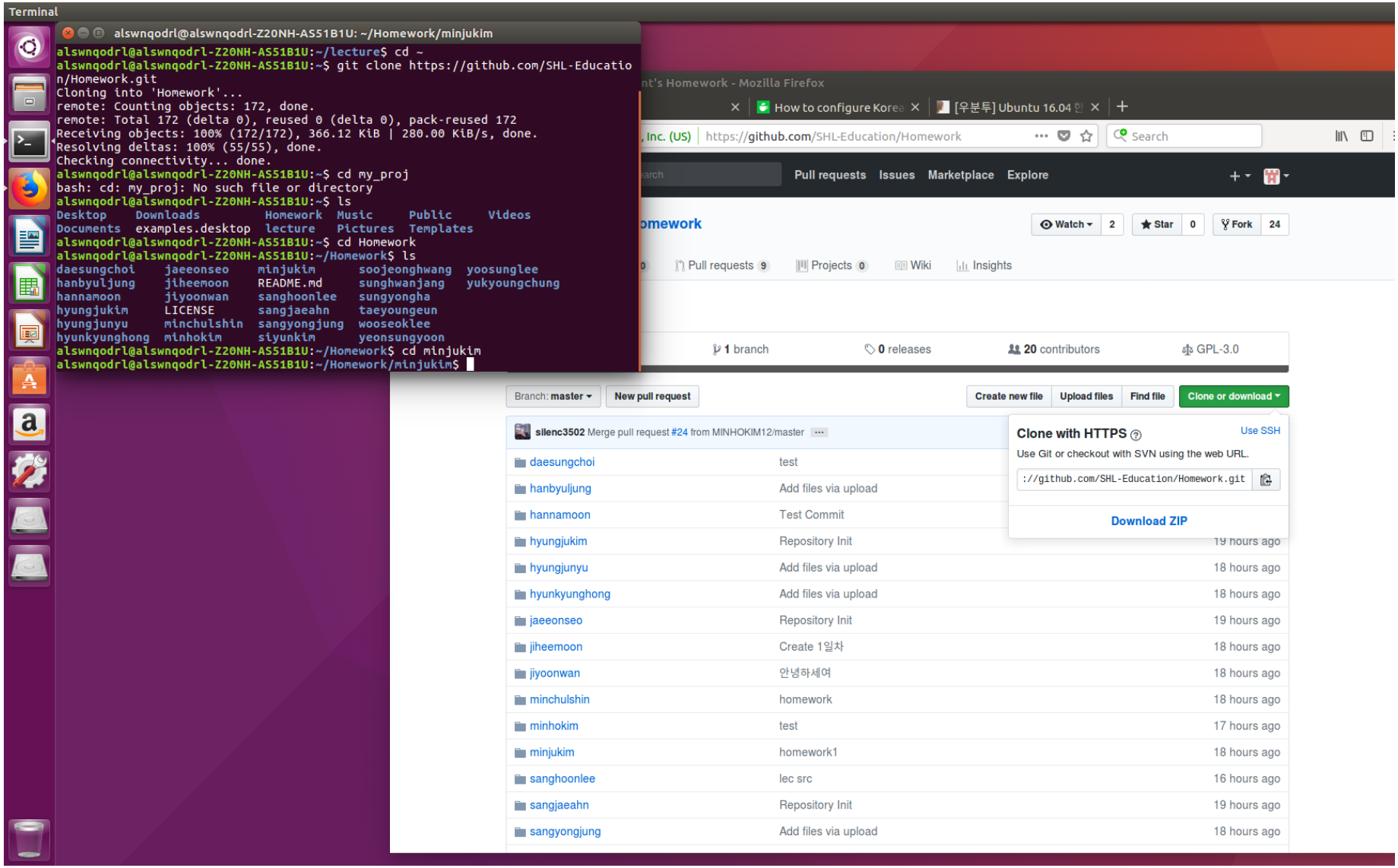
Ubuntu Homework2

김민주

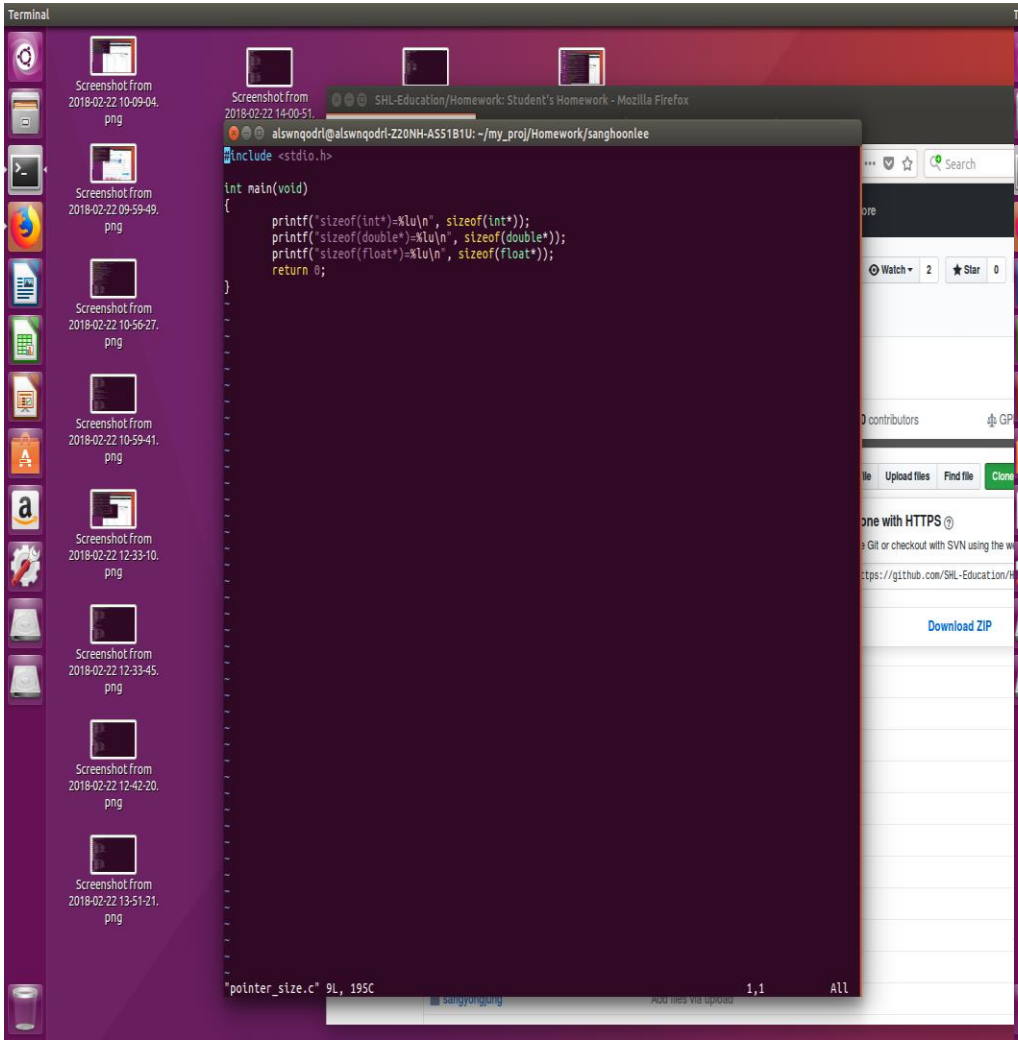
01 한글 설치 (super+space)

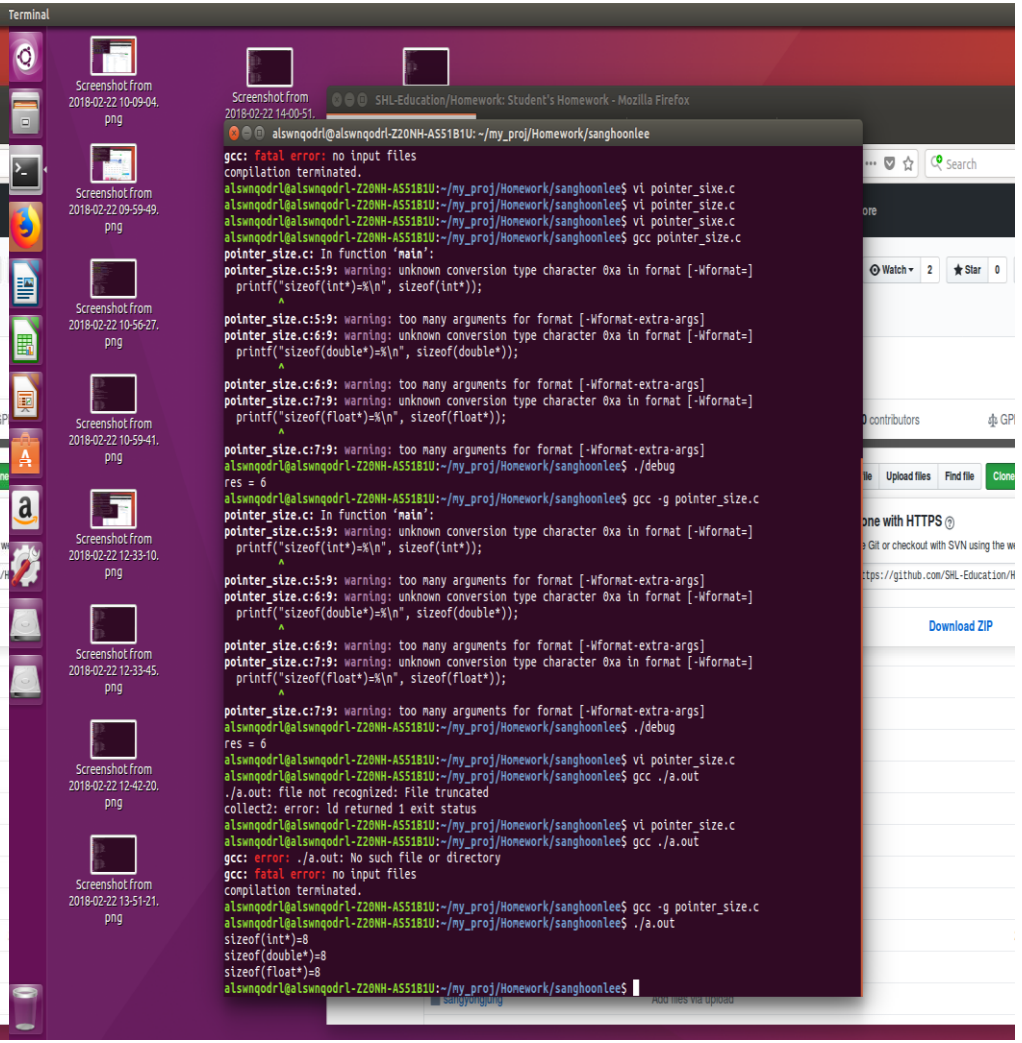


02 Git clone으로 원하는 위치로 이동

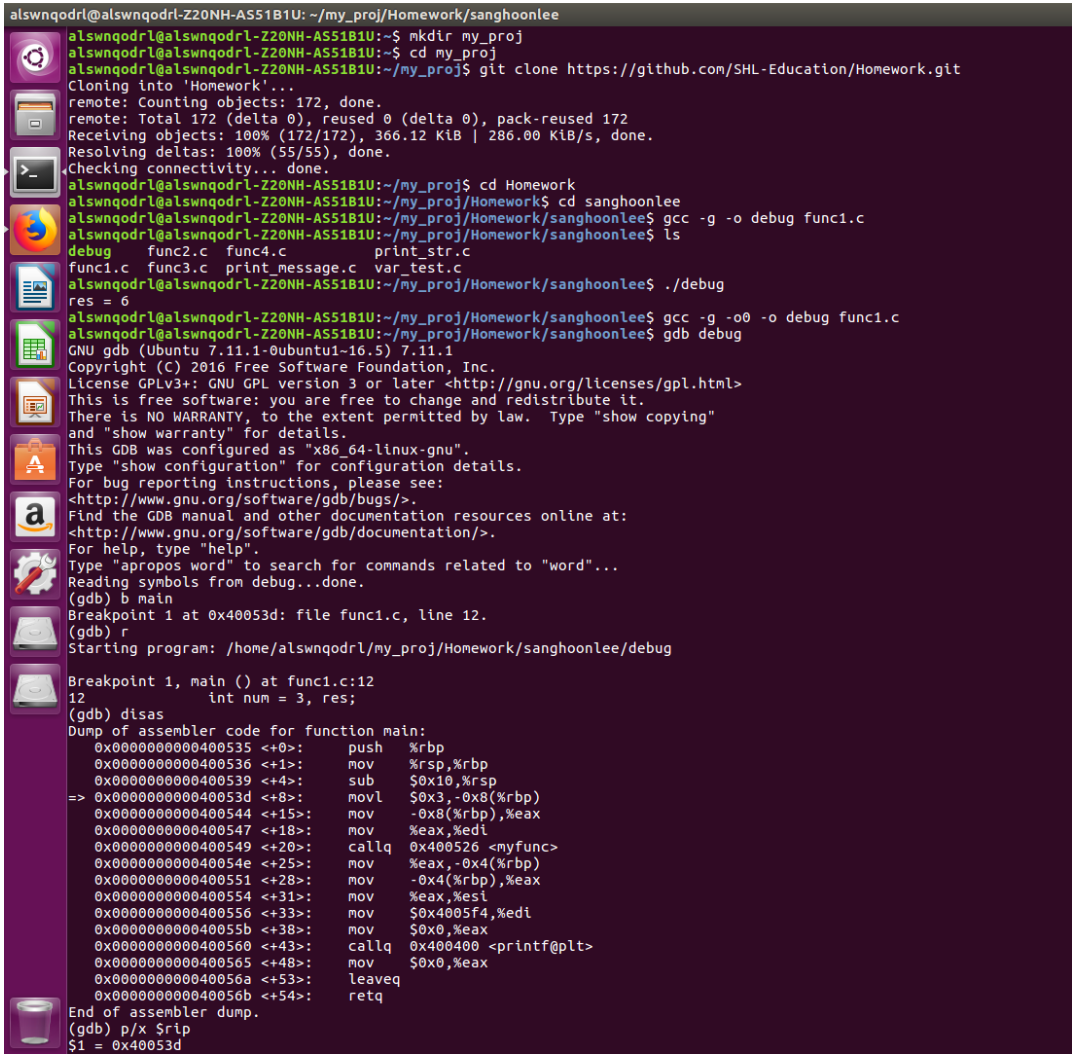


03 Bit 확인 – 64bit





04 디버깅



Sanghoonlee 폴더 안의 func1.c 디버깅

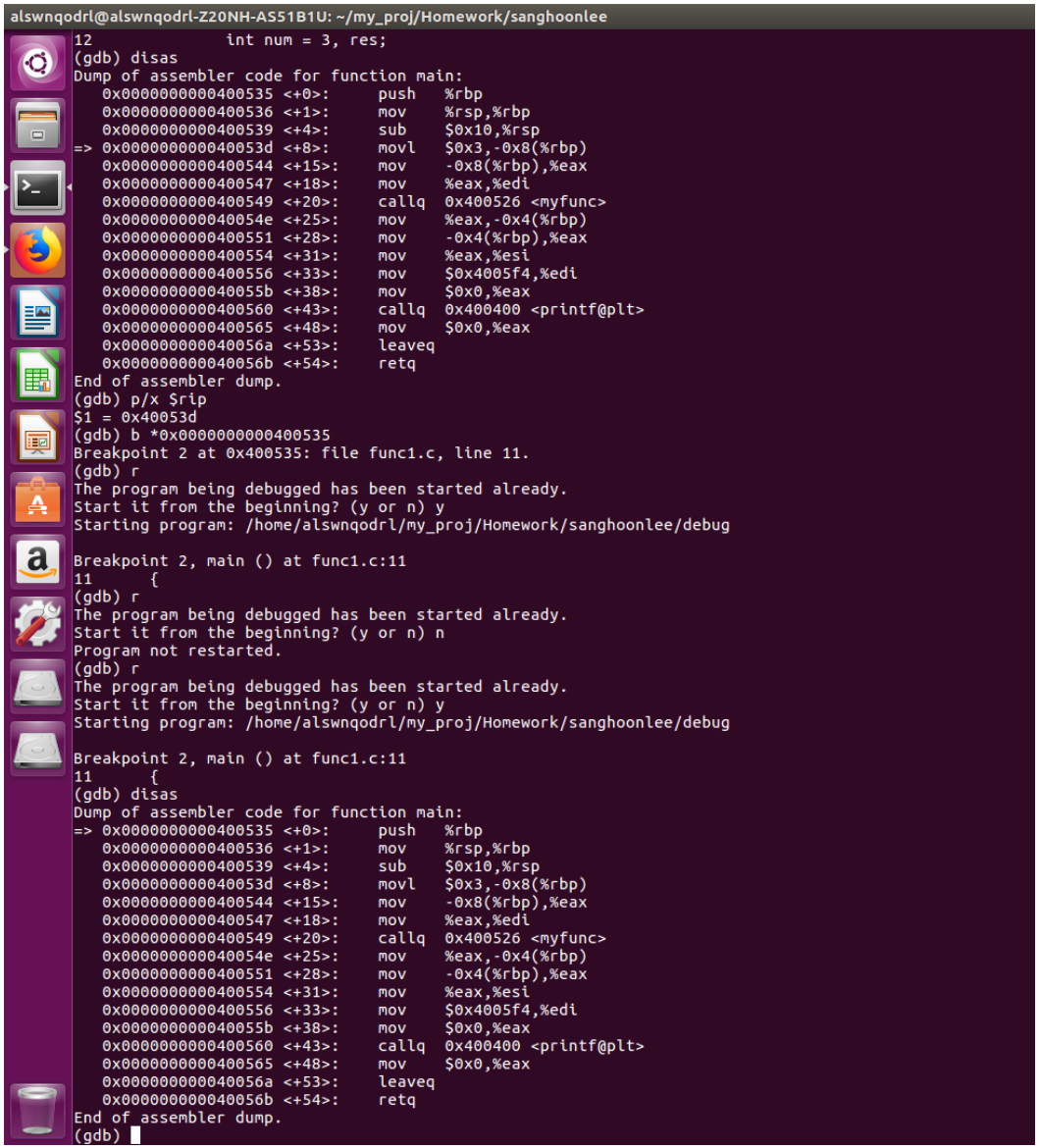
해석이 난해해지지 않도록 -o0를 통해 최적화를 방지

Gdb를 통해 디버깅 파일 실행하고 멈춰준 후 main실행(b main)

실행(run)을 위해 r입력

Disas(display to assemble) 실행

05 화살표 위치 바꾸기



복귀주소(rip) 확인

가장 위의 주소값에서 멈추게 함(b *)

실행(run)을 위해 r입력

Disas(display to assemble) 실행

06 기계어 분석

Si-> disas 실행을 통해

Push (현재 sp에 쌓음-64비트 하에선 rsp)

Mov(movl) a b (a를 b로 이동)

- mov와 movl은 길이의 차이만 있을뿐 역할동일

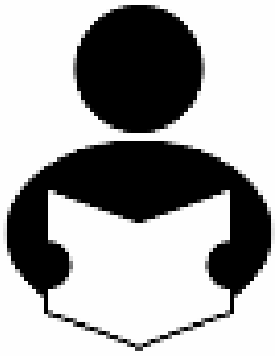
Callq 주소값 <지정함수> = push+jmp(힘수시행)

(해당 주소값에서 지정함수 실행 후 sp에 쌓기)

Pop a (현재 값을 유지하고 a로 돌아감)

Retq = pop rip (현재 값을 유지하고

rip(복귀주소)로 돌아감



06 기계어 분석

Sub a b (b에서 a를 뺌)

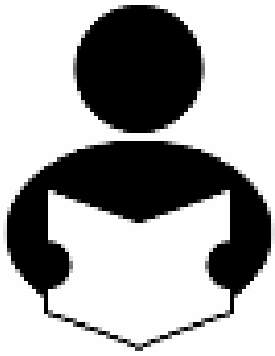
c/x a (enter 입력 시 마다 배열된 값을 출력)

p/x a (a의 크기를 출력)

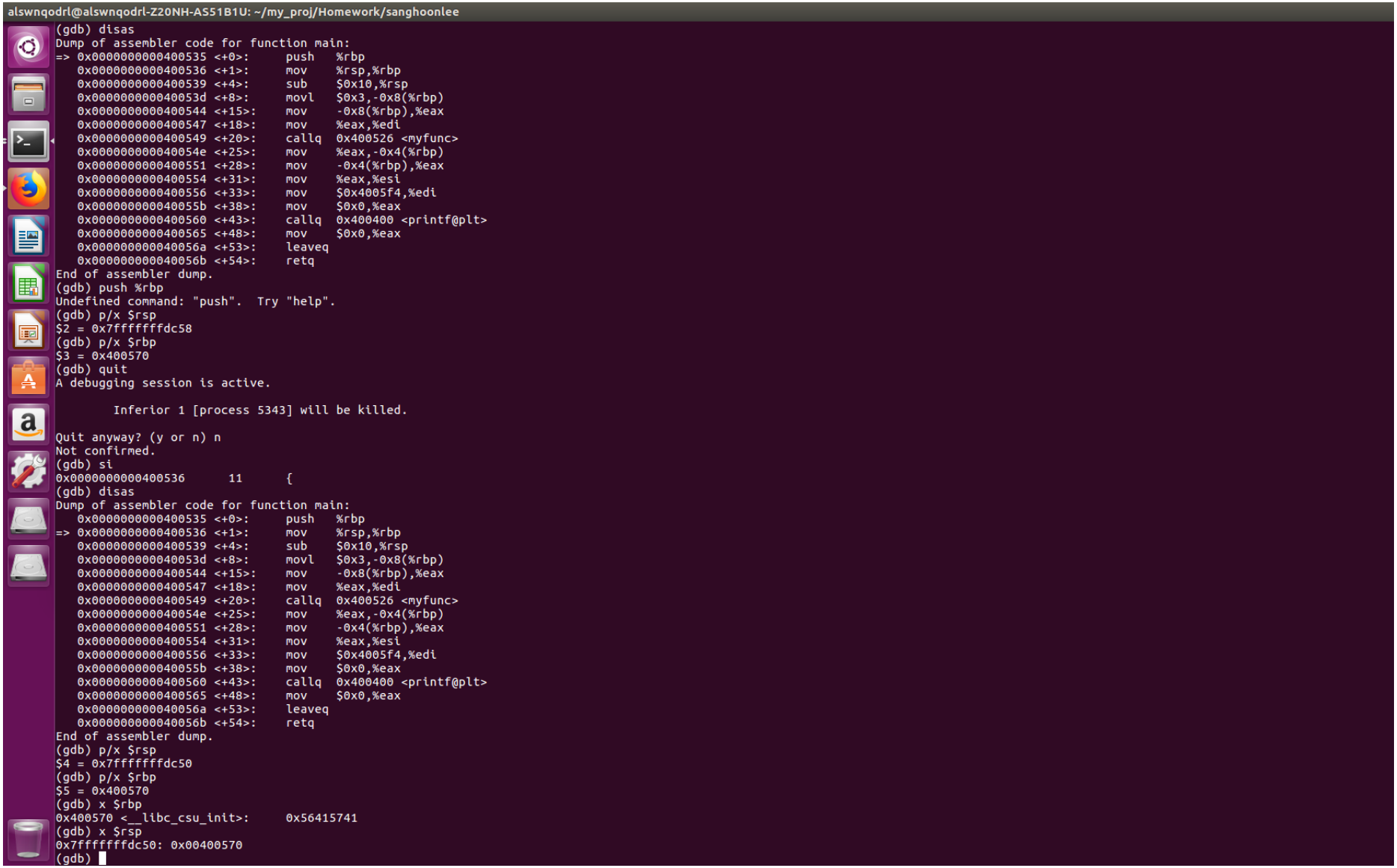
X a (a의 주소값과 크기를 모두 출력)

Bt (callq에서 빠져나와 다음 주소값으로 이동)

Ni (함수를 실해하지 않음)



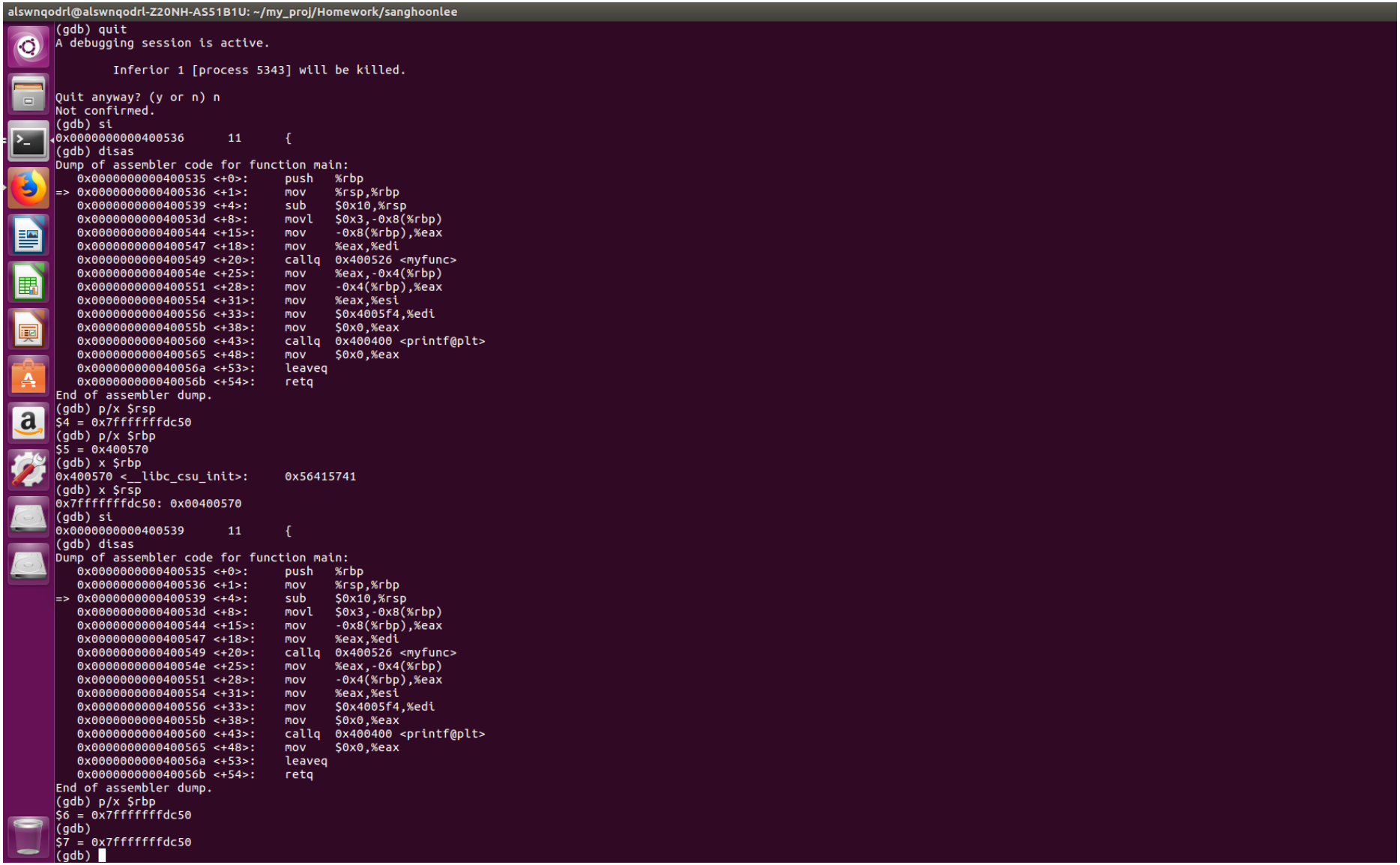
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
(gdb) disas
Dump of assembler code for function main:
=> 0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) push %rbp
Undefined command: "push". Try "help".
(gdb) p/x $rsp
$2 = 0x7fffffffdc58
(gdb) p/x $rbp
$3 = 0x400570
(gdb) quit
A debugging session is active.

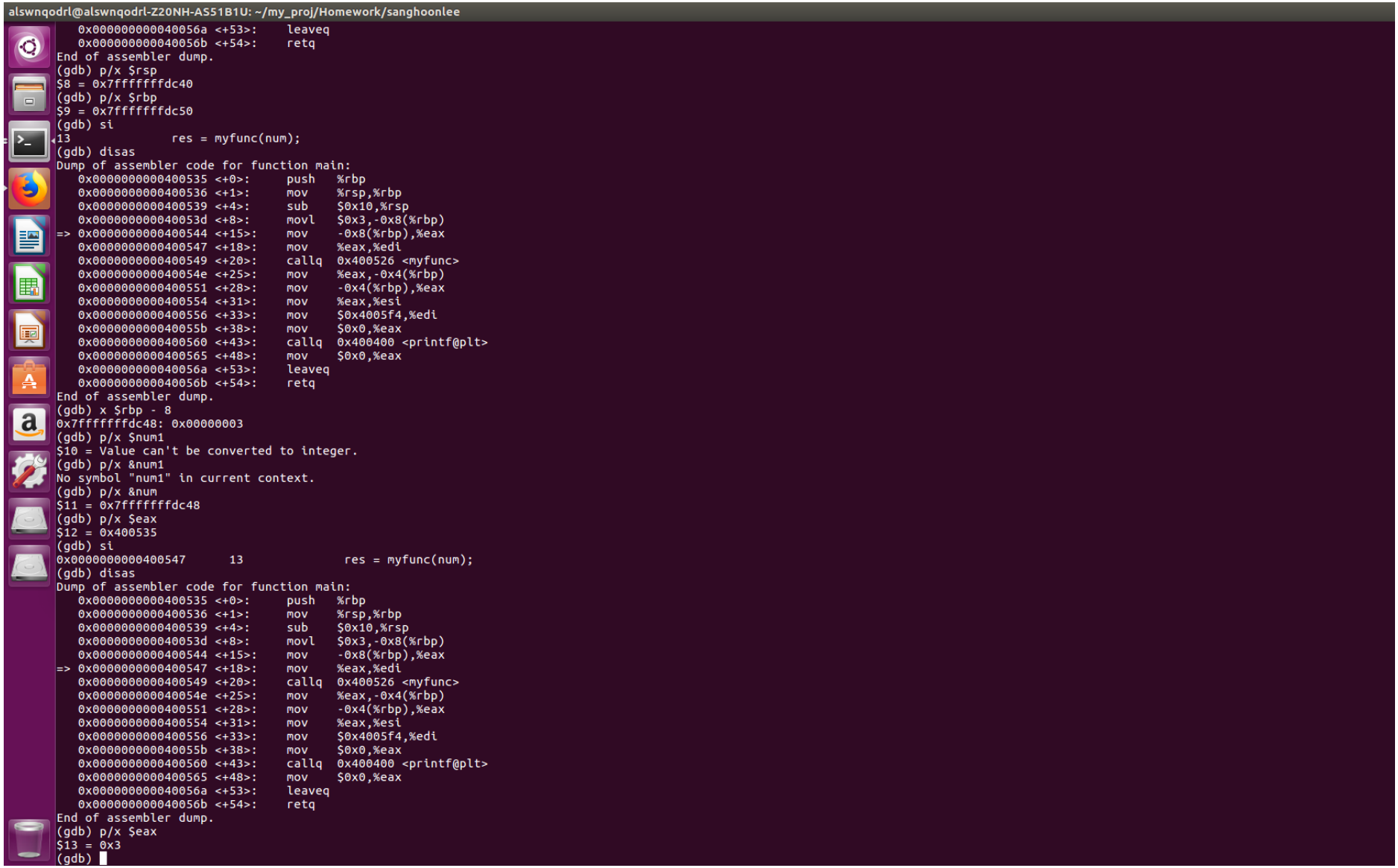
        Inferior 1 [process 5343] will be killed.

Quit anyway? (y or n) n
Not confirmed.
(gdb) si
0x0000000000400536   11   {
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
=> 0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $rsp
$4 = 0x7fffffffdc50
(gdb) p/x $rbp
$5 = 0x400570
(gdb) x $rbp
0x400570 <__libc_csu_init>:   0x56415741
(gdb) x $rsp
0x7fffffffdc50: 0x00400570
(gdb)

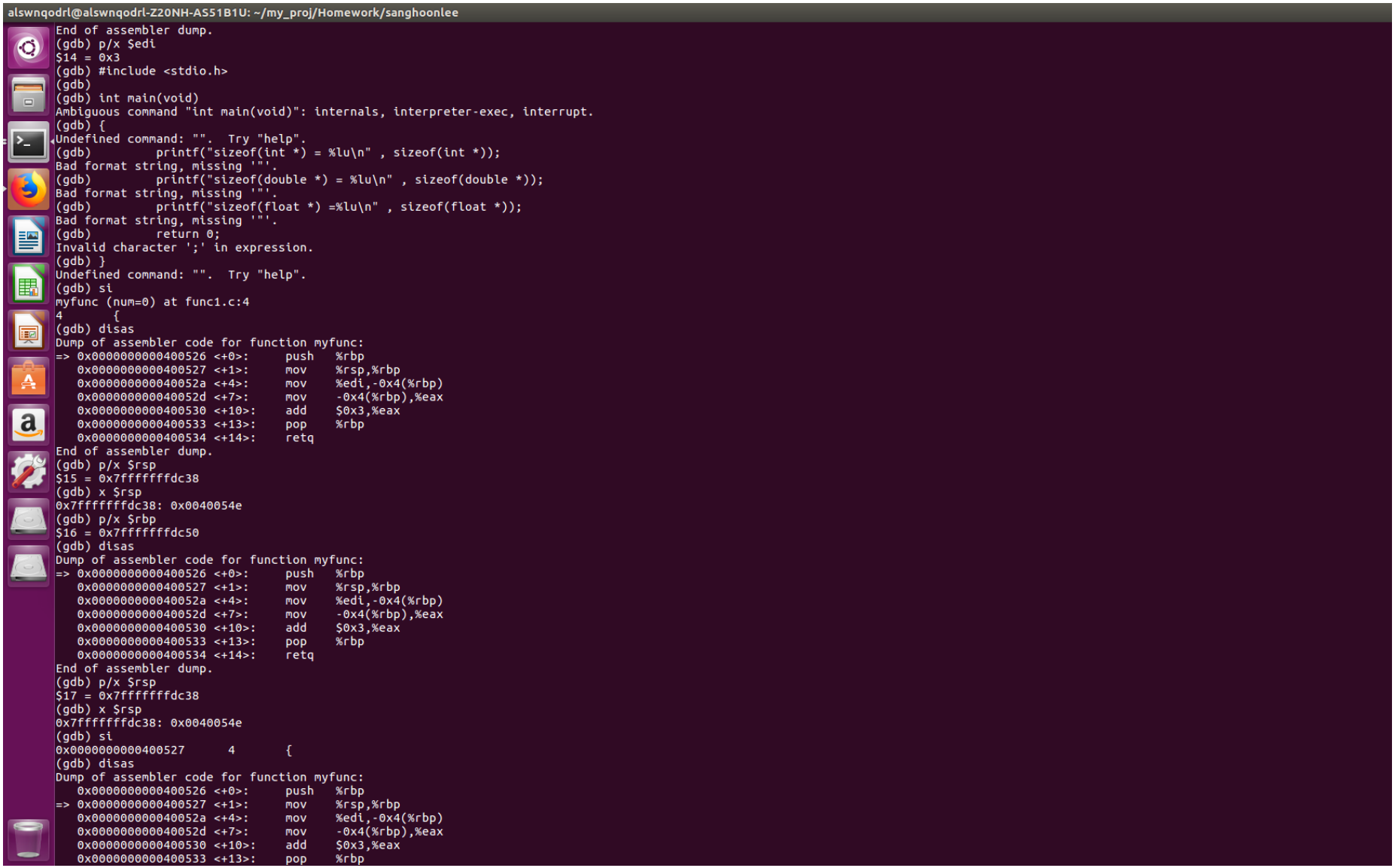
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
(gdb) quit
A debugging session is active.
Inferior 1 [process 5343] will be killed.
Quit anyway? (y or n) n
Not confirmed.
(gdb) si
0x0000000000400536 11 {
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
=> 0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rsp
$4 = 0x7fffffffdc50
(gdb) p/x $rbp
$5 = 0x400570
(gdb) x $rbp
0x400570 <__libc_csu_init>: 0x56415741
(gdb) x $rsp
0x7fffffffdc50: 0x00400570
(gdb) si
0x0000000000400539 11 {
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
=> 0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$6 = 0x7fffffffdc50
(gdb)
$7 = 0x7fffffffdc50
(gdb)

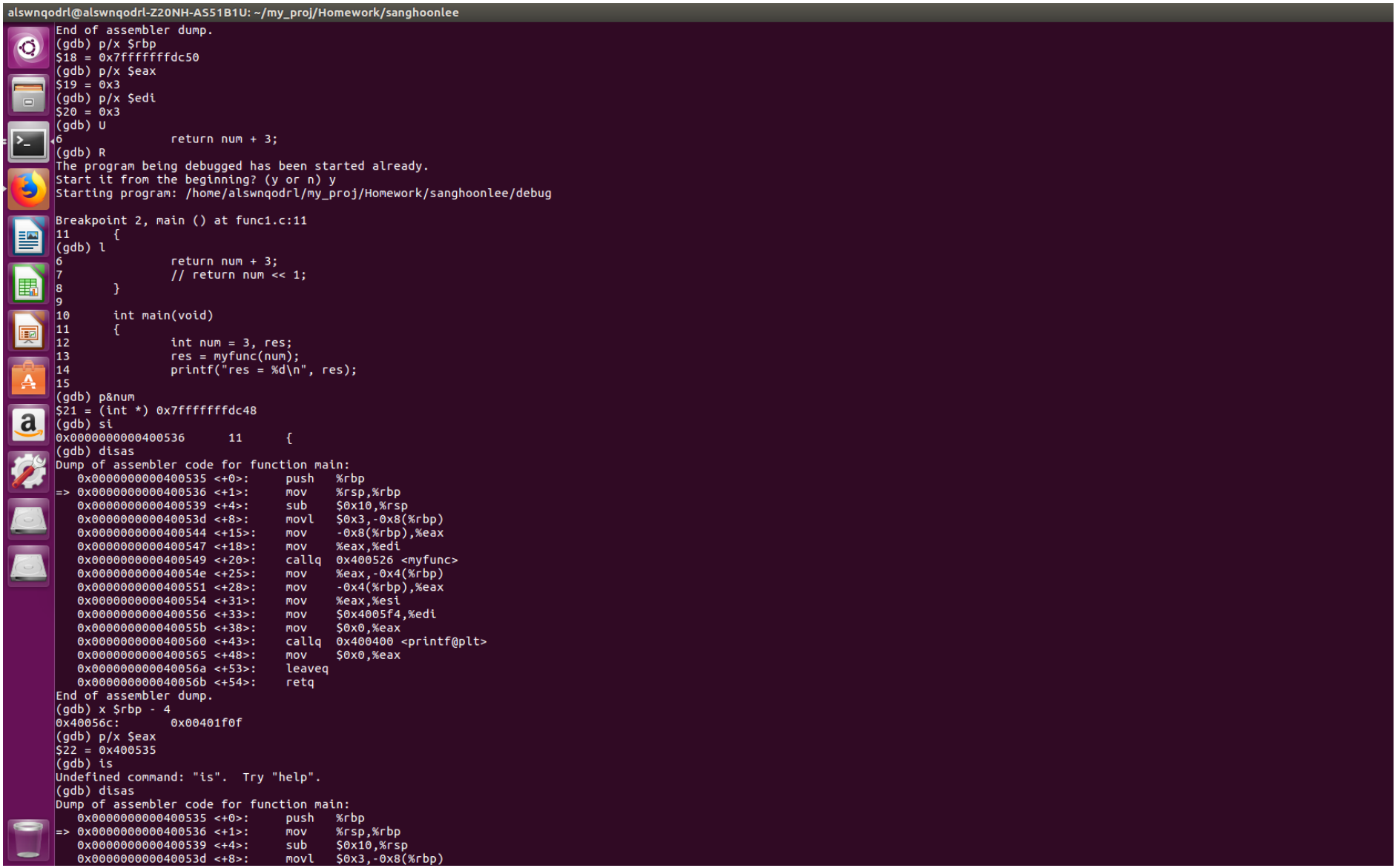
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $rsp
$8 = 0x7fffffffdc40
(gdb) p/x $rbp
$9 = 0x7fffffffdc50
(gdb) si
13   res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
=> 0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) x $rbp - 8
0x7fffffffdc48: 0x00000003
(gdb) p/x $num1
$10 = Value can't be converted to integer.
(gdb) p/x &num1
No symbol "num1" in current context.
(gdb) p/x &num
$11 = 0x7fffffffdc48
(gdb) p/x $eax
$12 = 0x400535
(gdb) si
0x0000000000400547   13   res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
=> 0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $eax
$13 = 0x3
(gdb)

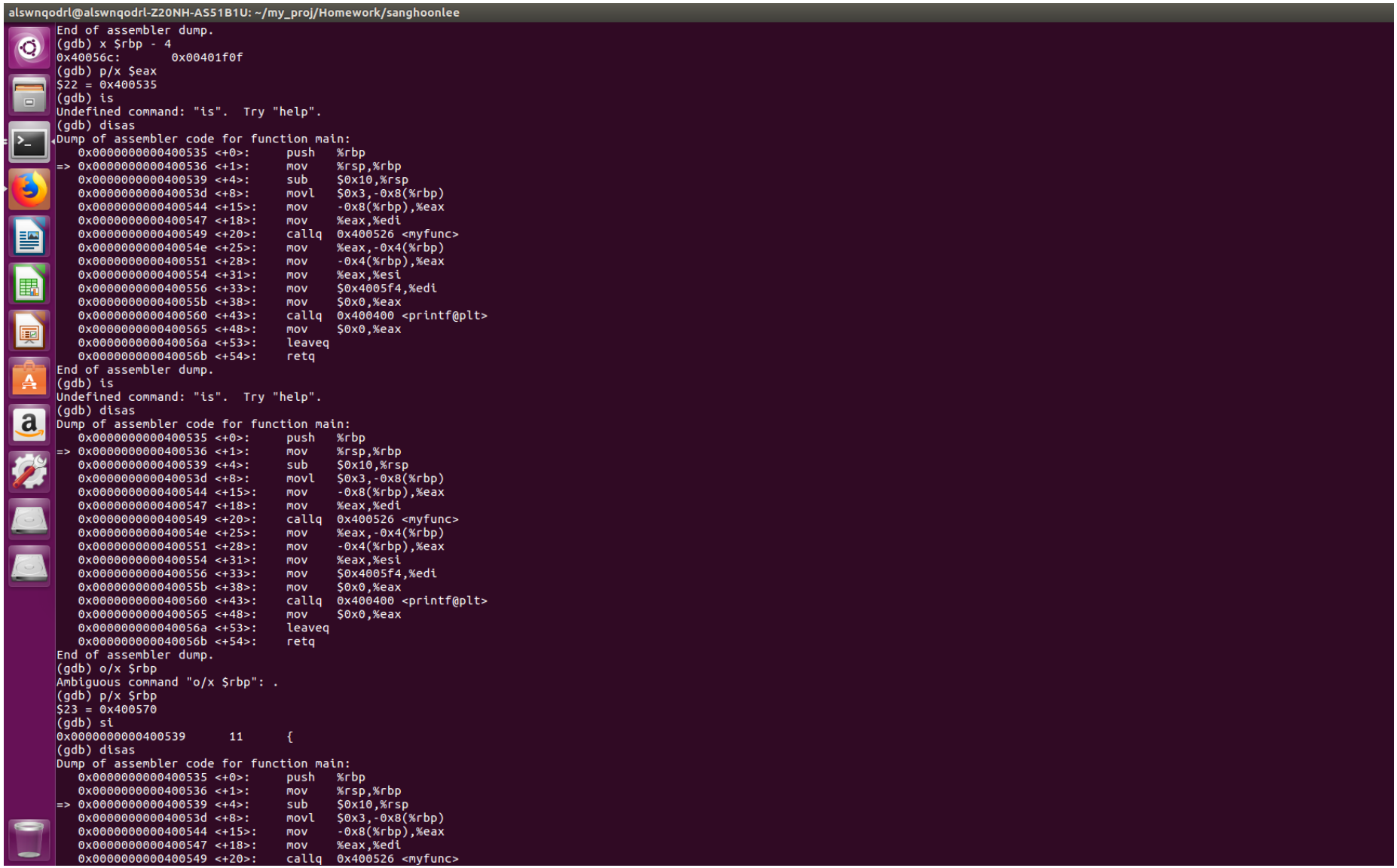
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
End of assembler dump.
(gdb) p/x $edi
$14 = 0x3
(gdb) #include <stdio.h>
(gdb)
(gdb) int main(void)
Ambiguous command "int main(void)": internals, interpreter-exec, interrupt.
(gdb) {
Undefined command: "". Try "help".
(gdb)   printf("sizeof(int *) = %lu\n" , sizeof(int *));
Bad format string, missing '"'.
(gdb)   printf("sizeof(double *) = %lu\n" , sizeof(double *));
Bad format string, missing '"'.
(gdb)   printf("sizeof(float *) =%lu\n" , sizeof(float *));
Bad format string, missing '"'.
(gdb)   return 0;
Invalid character ';' in expression.
(gdb) }
Undefined command: "". Try "help".
(gdb) si
myfunc (num=0) at func1.c:4
4   {
(gdb) disas
Dump of assembler code for function myfunc:
=> 0x0000000000400526 <+0>:   push   %rbp
   0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
   0x0000000000400530 <+10>:   add $0x3,%eax
   0x0000000000400533 <+13>:   pop   %rbp
   0x0000000000400534 <+14>:   retq
End of assembler dump.
(gdb) p/x $rsp
$15 = 0x7fffffffdc38
(gdb) x $rsp
0x7fffffffdc38: 0x0040054e
(gdb) p/x $rbp
$16 = 0x7fffffffdc50
(gdb) disas
Dump of assembler code for function myfunc:
=> 0x0000000000400526 <+0>:   push   %rbp
   0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
   0x0000000000400530 <+10>:   add $0x3,%eax
   0x0000000000400533 <+13>:   pop   %rbp
   0x0000000000400534 <+14>:   retq
End of assembler dump.
(gdb) p/x $rsp
$17 = 0x7fffffffdc38
(gdb) x $rsp
0x7fffffffdc38: 0x0040054e
(gdb) si
0x0000000000400527   4   {
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:   push   %rbp
=> 0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
   0x0000000000400530 <+10>:   add $0x3,%eax
   0x0000000000400533 <+13>:   pop   %rbp

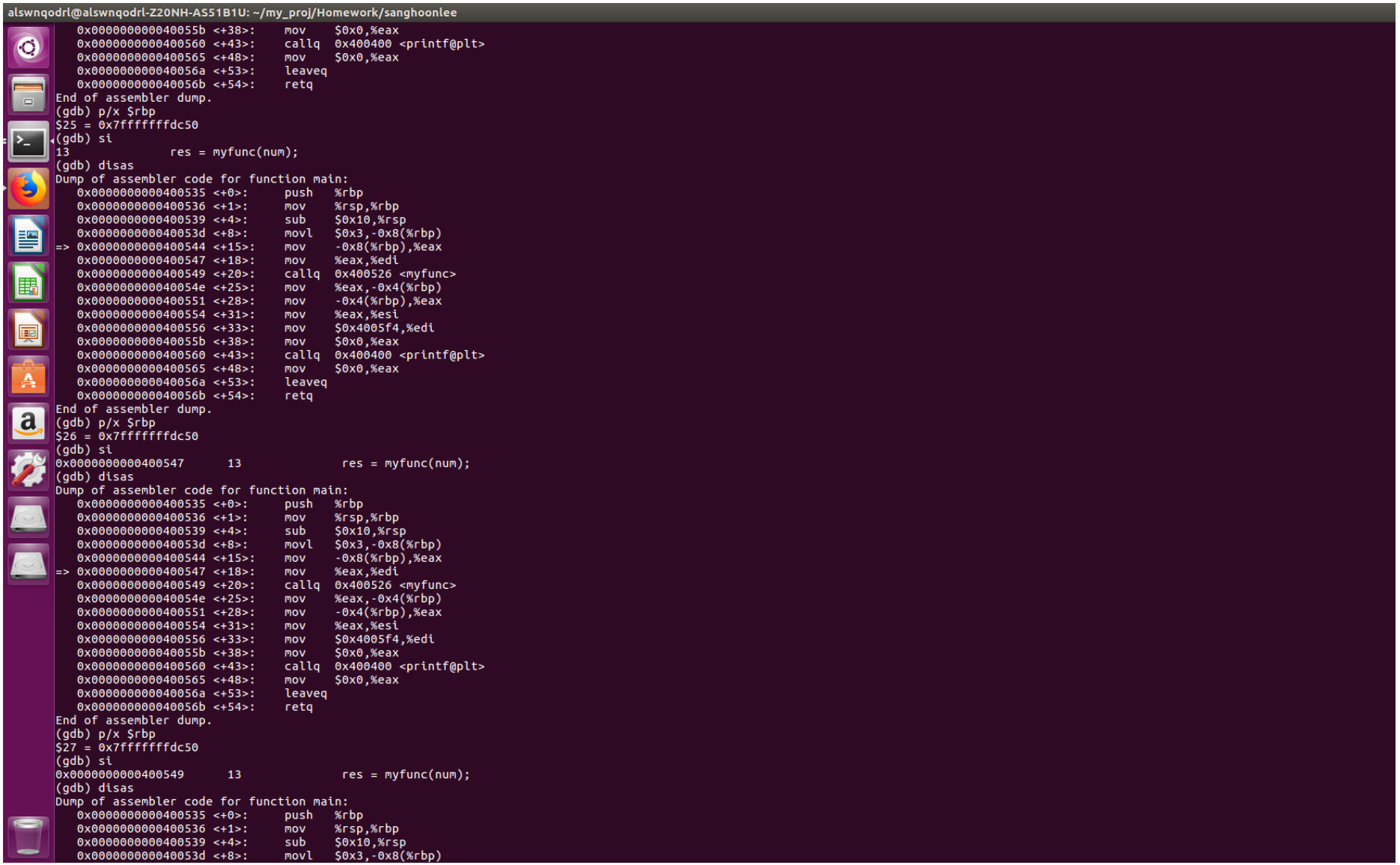
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
End of assembler dump.
(gdb) p/x $rbp
$18 = 0x7fffffffdc50
(gdb) p/x $eax
$19 = 0x3
(gdb) p/x $edi
$20 = 0x3
(gdb) U
6   return num + 3;
(gdb) R
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/alswnqodrl/my_proj/Homework/sanghoonlee/debug
Breakpoint 2, main () at func1.c:11
11   {
(gdb) l
6   return num + 3;
7   // return num << 1;
8   }
9
10   int main(void)
11   {
12   int num = 3, res;
13   res = myfunc(num);
14   printf("res = %d\n", res);
15
(gdb) p&num
$21 = (int *) 0x7fffffffdc48
(gdb) si
0x0000000000400536   11   {
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
=> 0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) x $rbp - 4
0x40056c: 0x00401f0f
(gdb) p/x $eax
$22 = 0x400535
(gdb) is
Undefined command: "is". Try "help".
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
=> 0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)

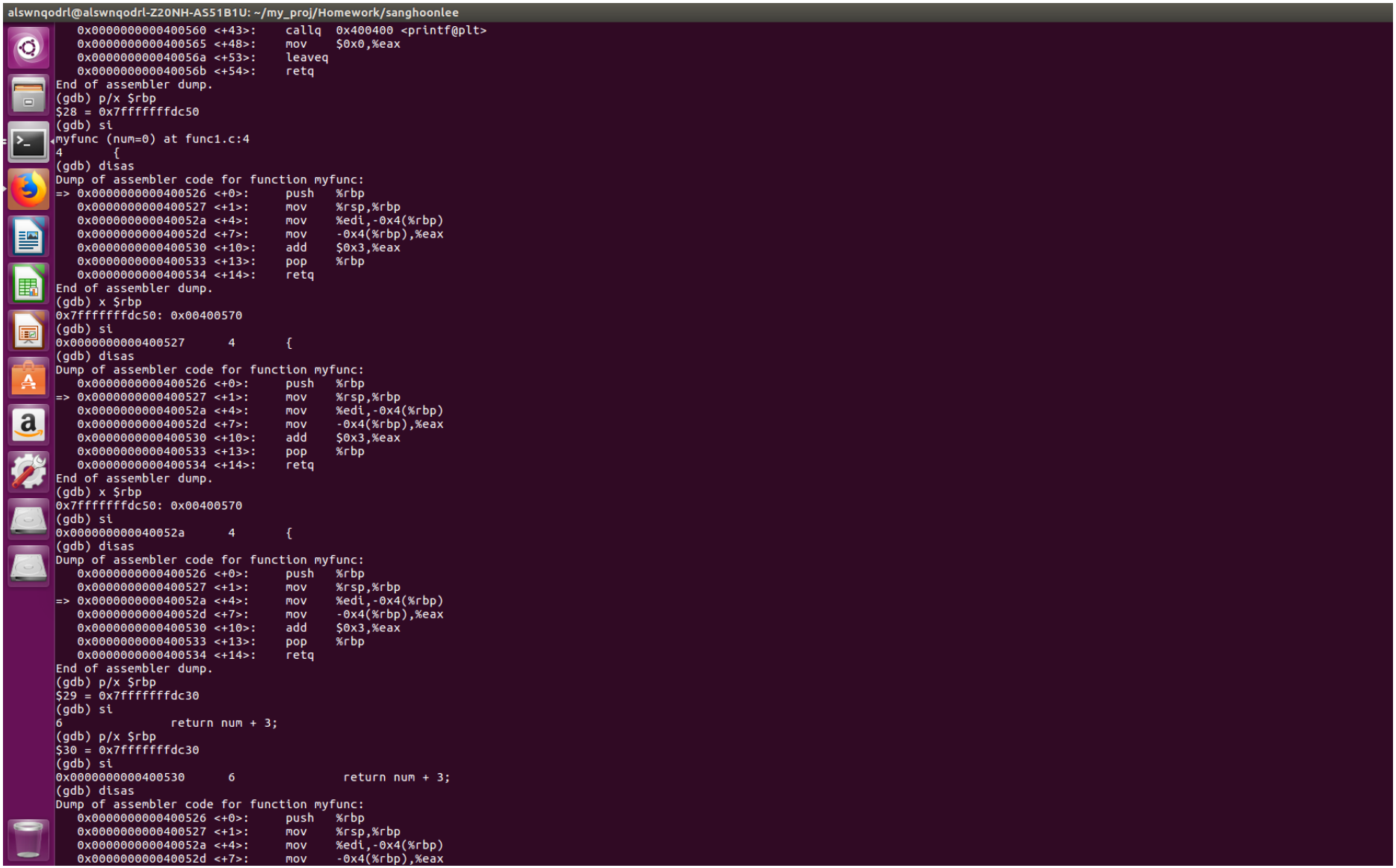
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
End of assembler dump.
(gdb) x $rbp - 4
0x40056c:	0x00401f0f
(gdb) p/x $eax
$22 = 0x400535
(gdb) is
Undefined command: "is". Try "help".
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:	push %rbp
=> 0x0000000000400536 <+1>:	mov %rsp,%rbp
   0x0000000000400539 <+4>:	sub $0x10,%rsp
   0x000000000040053d <+8>:	movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:	mov -0x8(%rbp),%eax
   0x0000000000400547 <+18>:	mov %eax,%edi
   0x0000000000400549 <+20>:	callq 0x400526 <myfunc>
   0x000000000040054e <+25>:	mov %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:	mov -0x4(%rbp),%eax
   0x0000000000400554 <+31>:	mov %eax,%esi
   0x0000000000400556 <+33>:	mov $0x4005f4,%edi
   0x000000000040055b <+38>:	mov $0x0,%eax
   0x0000000000400560 <+43>:	callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:	mov $0x0,%eax
   0x000000000040056a <+53>:	leaveq
   0x000000000040056b <+54>:	retq
End of assembler dump.
(gdb) is
Undefined command: "is". Try "help".
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:	push %rbp
=> 0x0000000000400536 <+1>:	mov %rsp,%rbp
   0x0000000000400539 <+4>:	sub $0x10,%rsp
   0x000000000040053d <+8>:	movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:	mov -0x8(%rbp),%eax
   0x0000000000400547 <+18>:	mov %eax,%edi
   0x0000000000400549 <+20>:	callq 0x400526 <myfunc>
   0x000000000040054e <+25>:	mov %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:	mov -0x4(%rbp),%eax
   0x0000000000400554 <+31>:	mov %eax,%esi
   0x0000000000400556 <+33>:	mov $0x4005f4,%edi
   0x000000000040055b <+38>:	mov $0x0,%eax
   0x0000000000400560 <+43>:	callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:	mov $0x0,%eax
   0x000000000040056a <+53>:	leaveq
   0x000000000040056b <+54>:	retq
End of assembler dump.
(gdb) o/x $rbp
Ambiguous command "o/x $rbp": .
(gdb) p/x $rbp
$23 = 0x400570
(gdb) si
0x0000000000400539	11	{
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:	push %rbp
   0x0000000000400536 <+1>:	mov %rsp,%rbp
=> 0x0000000000400539 <+4>:	sub $0x10,%rsp
   0x000000000040053d <+8>:	movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:	mov -0x8(%rbp),%eax
   0x0000000000400547 <+18>:	mov %eax,%edi
   0x0000000000400549 <+20>:	callq 0x400526 <myfunc>

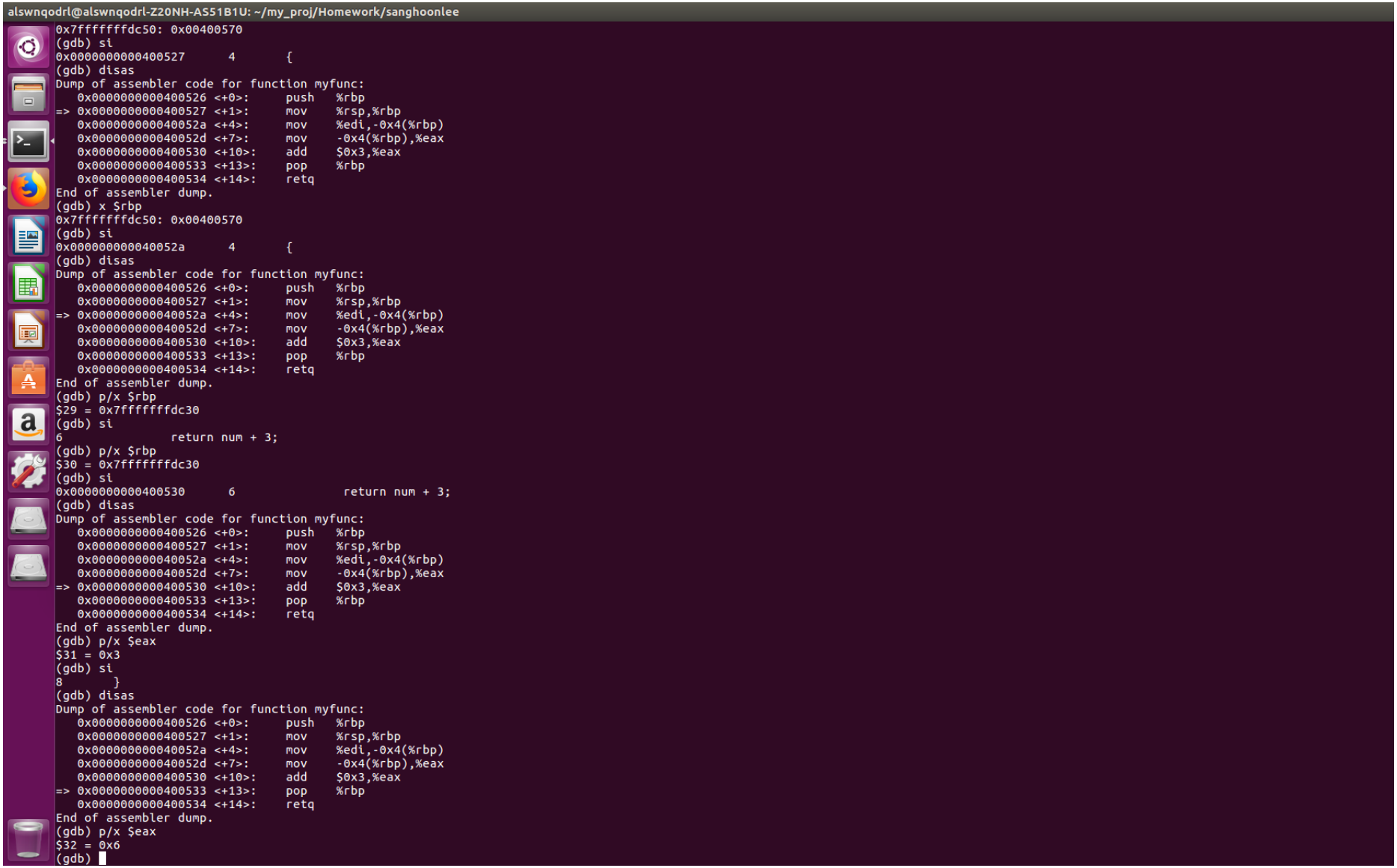
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$25 = 0x7fffffffdc50
(gdb) si
13 res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
=> 0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$26 = 0x7fffffffdc50
(gdb) si
0x0000000000400547 13 res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
=> 0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$27 = 0x7fffffffdc50
(gdb) si
0x0000000000400549 13 res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)

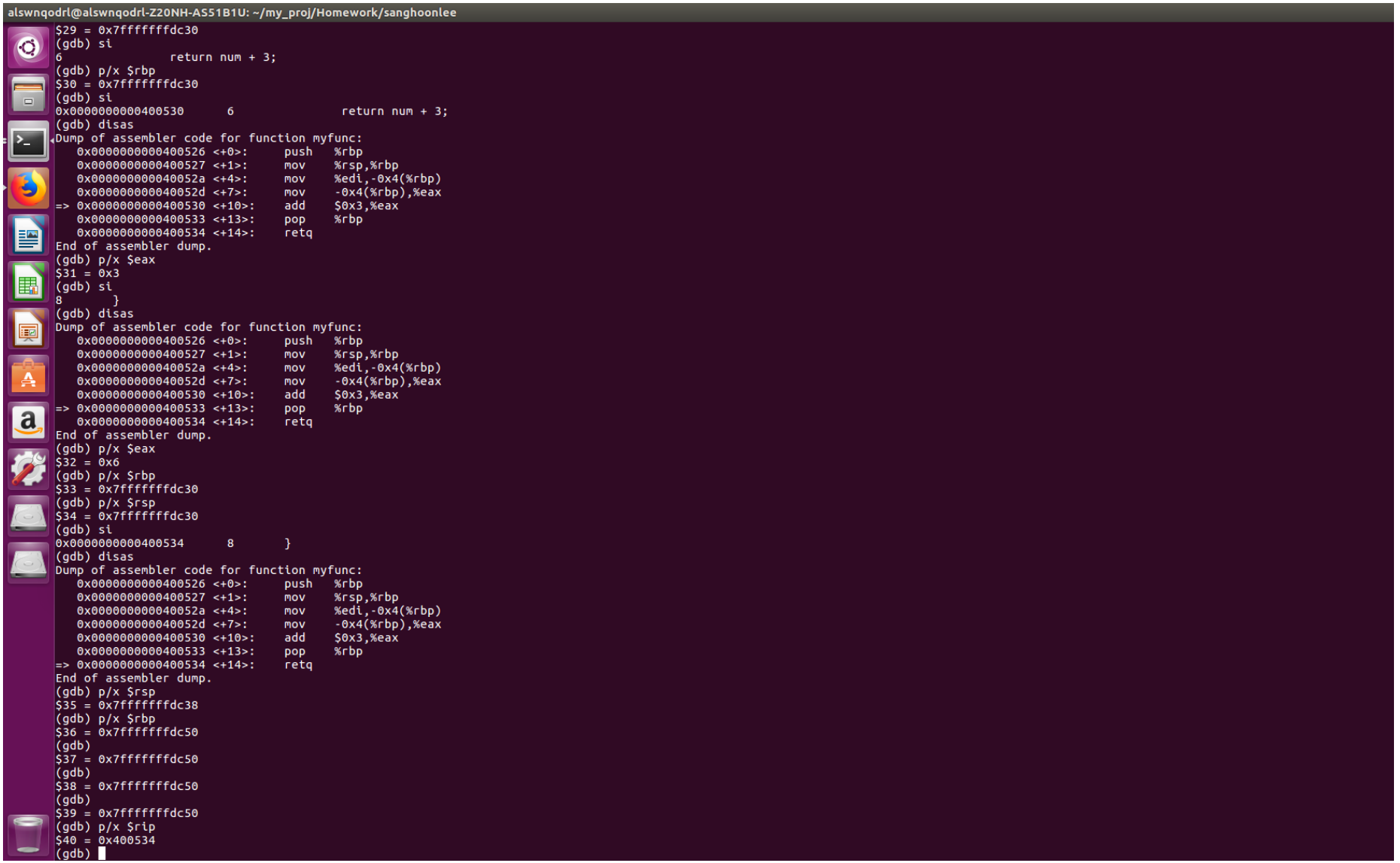
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$28 = 0x7fffffffdc50
(gdb) si
myfunc (num=0) at func1.c:4
4 {
(gdb) disas
Dump of assembler code for function myfunc:
=> 0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) x $rbp
0x7fffffffdc50: 0x00400570
(gdb) si
0x0000000000400527 4 {
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
=> 0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) x $rbp
0x7fffffffdc50: 0x00400570
(gdb) si
0x000000000040052a 4 {
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
=> 0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) p/x $rbp
$29 = 0x7fffffffdc30
(gdb) si
6 return num + 3;
(gdb) p/x $rbp
$30 = 0x7fffffffdc30
(gdb) si
0x0000000000400530 6 return num + 3;
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax

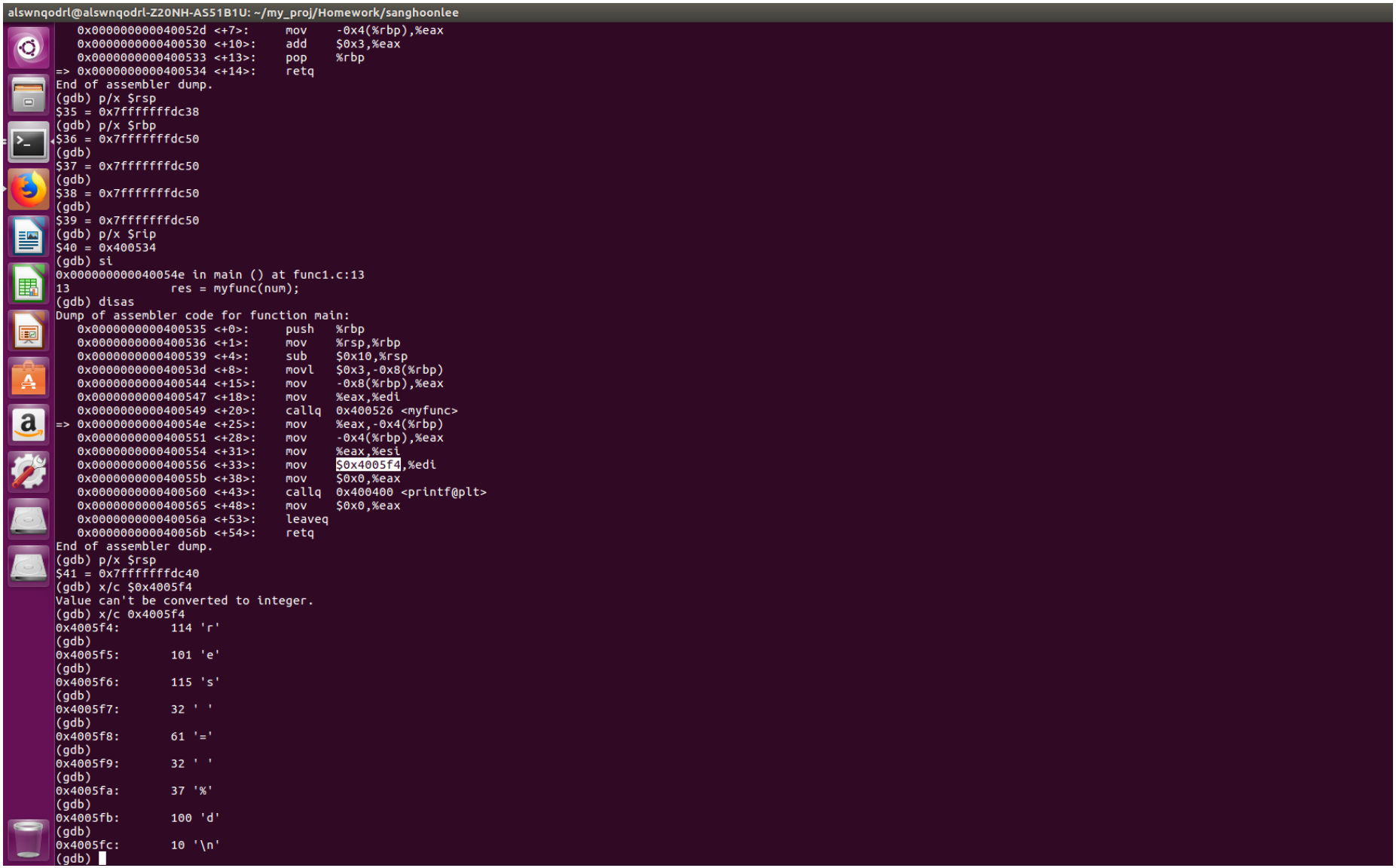
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x7fffffffdc50: 0x00400570
(gdb) si
0x0000000000400527 4 {
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
=> 0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) x $rbp
0x7fffffffdc50: 0x00400570
(gdb) si
0x000000000040052a 4 {
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
=> 0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) p/x $rbp
$29 = 0x7fffffffdc30
(gdb) si
6 return num + 3;
(gdb) p/x $rbp
$30 = 0x7fffffffdc30
(gdb) si
0x0000000000400530 6 return num + 3;
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
=> 0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) p/x $eax
$31 = 0x3
(gdb) si
8 }
(gdb) disas
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
=> 0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) p/x $eax
$32 = 0x6
(gdb)

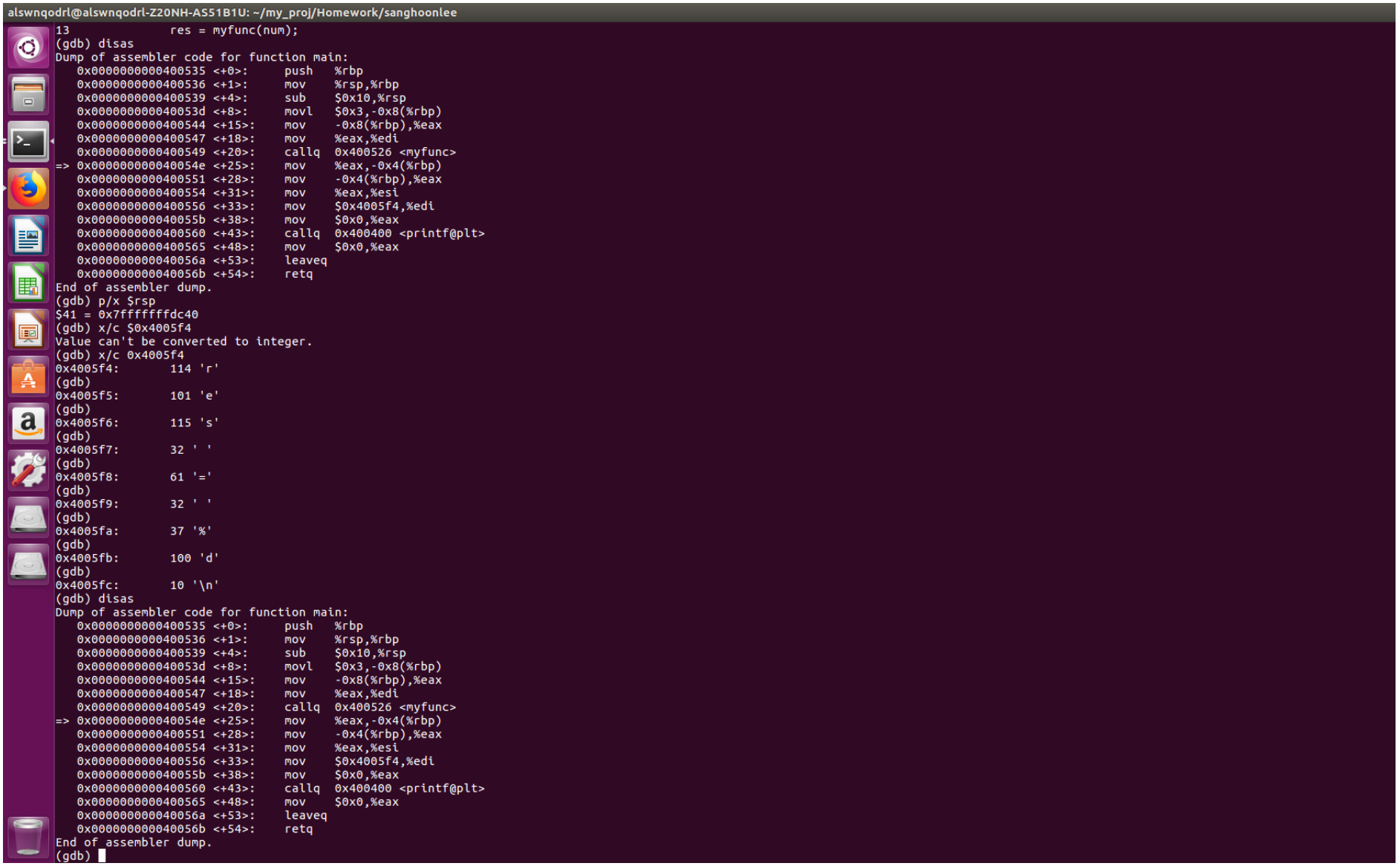
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
$29 = 0x7fffffffdc30
(gdb) si
6   return num + 3;
(gdb) p/x $rbp
$30 = 0x7fffffffdc30
(gdb) si
0x0000000000400530   6   return num + 3;
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:   push   %rbp
   0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
=> 0x0000000000400530 <+10>:   add $0x3,%eax
   0x0000000000400533 <+13>:   pop   %rbp
   0x0000000000400534 <+14>:   retq
End of assembler dump.
(gdb) p/x $eax
$31 = 0x3
(gdb) si
8   }
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:   push   %rbp
   0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
   0x0000000000400530 <+10>:   add $0x3,%eax
=> 0x0000000000400533 <+13>:   pop   %rbp
   0x0000000000400534 <+14>:   retq
End of assembler dump.
(gdb) p/x $eax
$32 = 0x6
(gdb) p/x $rbp
$33 = 0x7fffffffdc30
(gdb) p/x $rsp
$34 = 0x7fffffffdc30
(gdb) si
0x0000000000400534   8   }
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:   push   %rbp
   0x0000000000400527 <+1>:   mov   %rsp,%rbp
   0x000000000040052a <+4>:   mov   %edi,-0x4(%rbp)
   0x000000000040052d <+7>:   mov   -0x4(%rbp),%eax
   0x0000000000400530 <+10>:   add $0x3,%eax
   0x0000000000400533 <+13>:   pop   %rbp
=> 0x0000000000400534 <+14>:   retq
End of assembler dump.
(gdb) p/x $rsp
$35 = 0x7fffffffdc38
(gdb) p/x $rbp
$36 = 0x7fffffffdc50
(gdb)
$37 = 0x7fffffffdc50
(gdb)
$38 = 0x7fffffffdc50
(gdb)
$39 = 0x7fffffffdc50
(gdb) p/x $rip
$40 = 0x400534
(gdb)

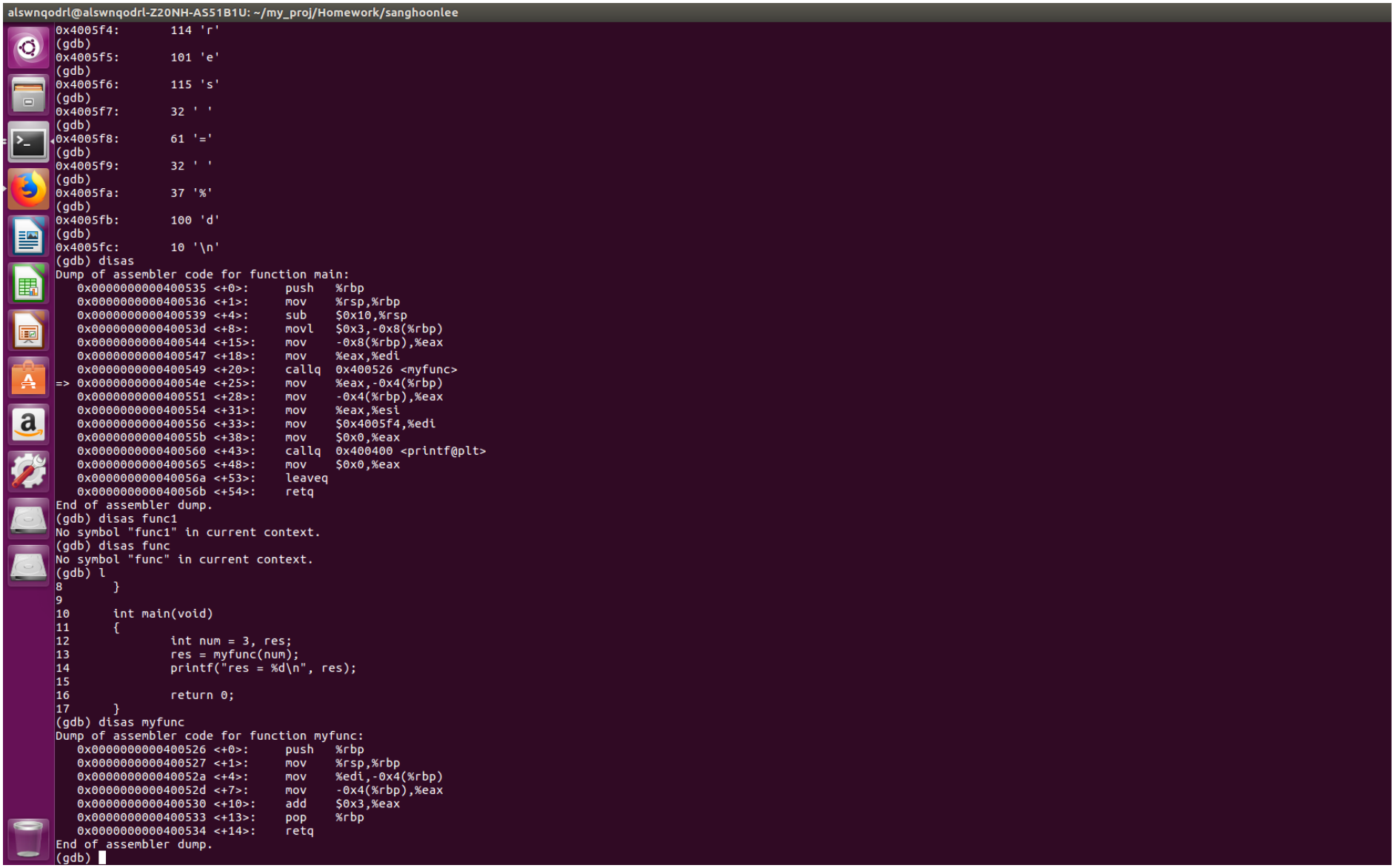
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
=> 0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) p/x $rsp
$35 = 0x7fffffffdc38
(gdb) p/x $rbp
$36 = 0x7fffffffdc50
(gdb)
$37 = 0x7fffffffdc50
(gdb)
$38 = 0x7fffffffdc50
(gdb)
$39 = 0x7fffffffdc50
(gdb) p/x $rip
$40 = 0x400534
(gdb) si
0x000000000040054e in main () at func1.c:13
13 res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rsp
$41 = 0x7fffffffdc40
(gdb) x/c $0x4005f4
Value can't be converted to integer.
(gdb) x/c 0x4005f4
0x4005f4: 114 'r'
(gdb)
0x4005f5: 101 'e'
(gdb)
0x4005f6: 115 's'
(gdb)
0x4005f7: 32 ' '
(gdb)
0x4005f8: 61 '='
(gdb)
0x4005f9: 32 ' '
(gdb)
0x4005fa: 37 '%'
(gdb)
0x4005fb: 100 'd'
(gdb)
0x4005fc: 10 '\n'
(gdb)

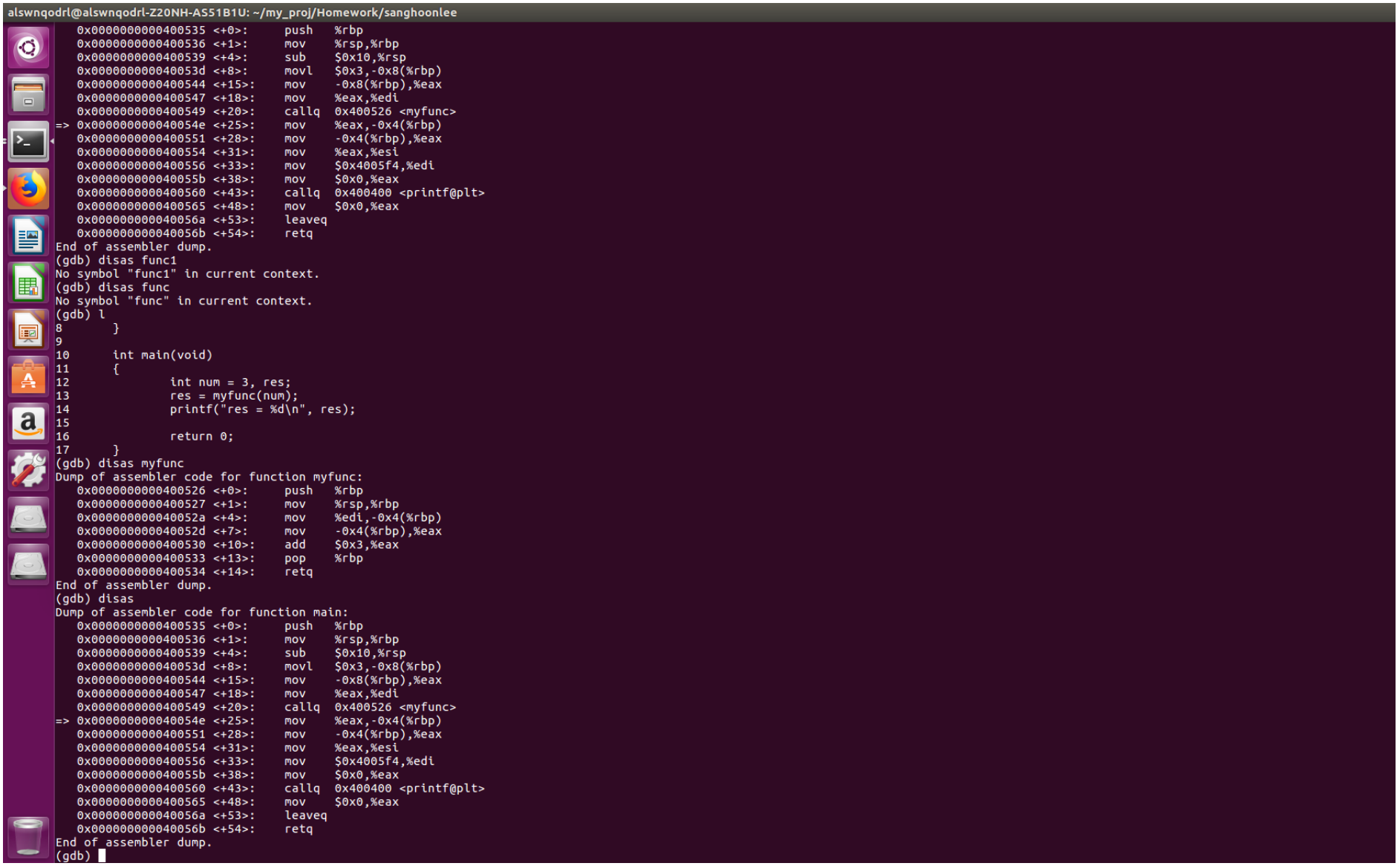
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
13   res = myfunc(num);
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $rsp
$41 = 0x7fffffffdc40
(gdb) x/c $0x4005f4
Value can't be converted to integer.
(gdb) x/c 0x4005f4
0x4005f4:   114 'r'
(gdb)
0x4005f5:   101 'e'
(gdb)
0x4005f6:   115 's'
(gdb)
0x4005f7:   32 ' '
(gdb)
0x4005f8:   61 '='
(gdb)
0x4005f9:   32 ' '
(gdb)
0x4005fa: 37 '%'
(gdb)
0x4005fb: 100 'd'
(gdb)
0x4005fc: 10 '\n'
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb)

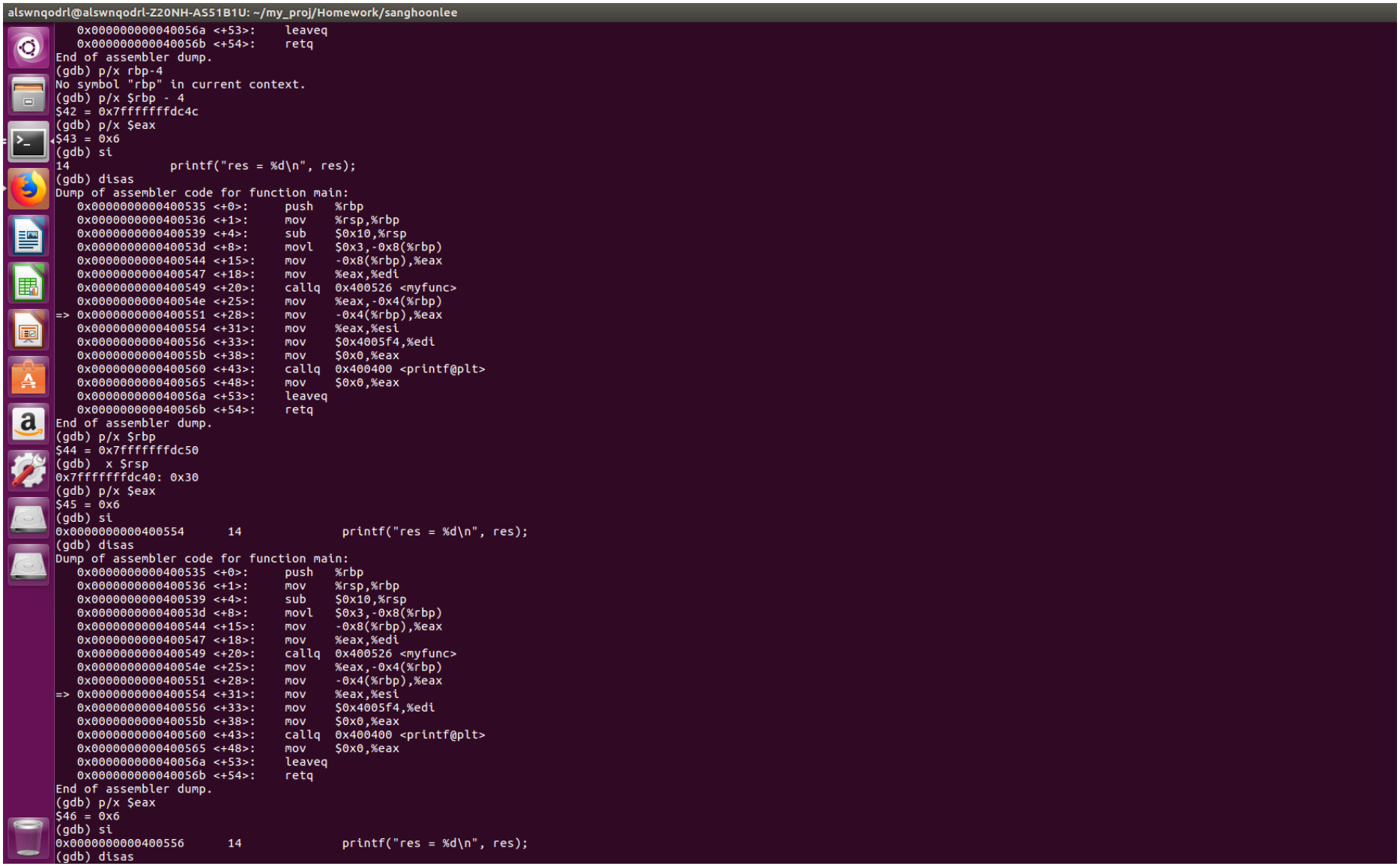
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x4005f4:	114 'r'
(gdb)
0x4005f5:	101 'e'
(gdb)
0x4005f6:	115 's'
(gdb)
0x4005f7:	32 ' '
(gdb)
0x4005f8:	61 '='
(gdb)
0x4005f9:	32 ' '
(gdb)
0x4005fa:	37 '%'
(gdb)
0x4005fb:	100 'd'
(gdb)
0x4005fc:	10 '\n'
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) disas func1
No symbol "func1" in current context.
(gdb) disas func
No symbol "func" in current context.
(gdb) l
8 }
9
10 int main(void)
11 {
12 int num = 3, res;
13 res = myfunc(num);
14 printf("res = %d\n", res);
15
16 return 0;
17 }
(gdb) disas myfunc
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb)

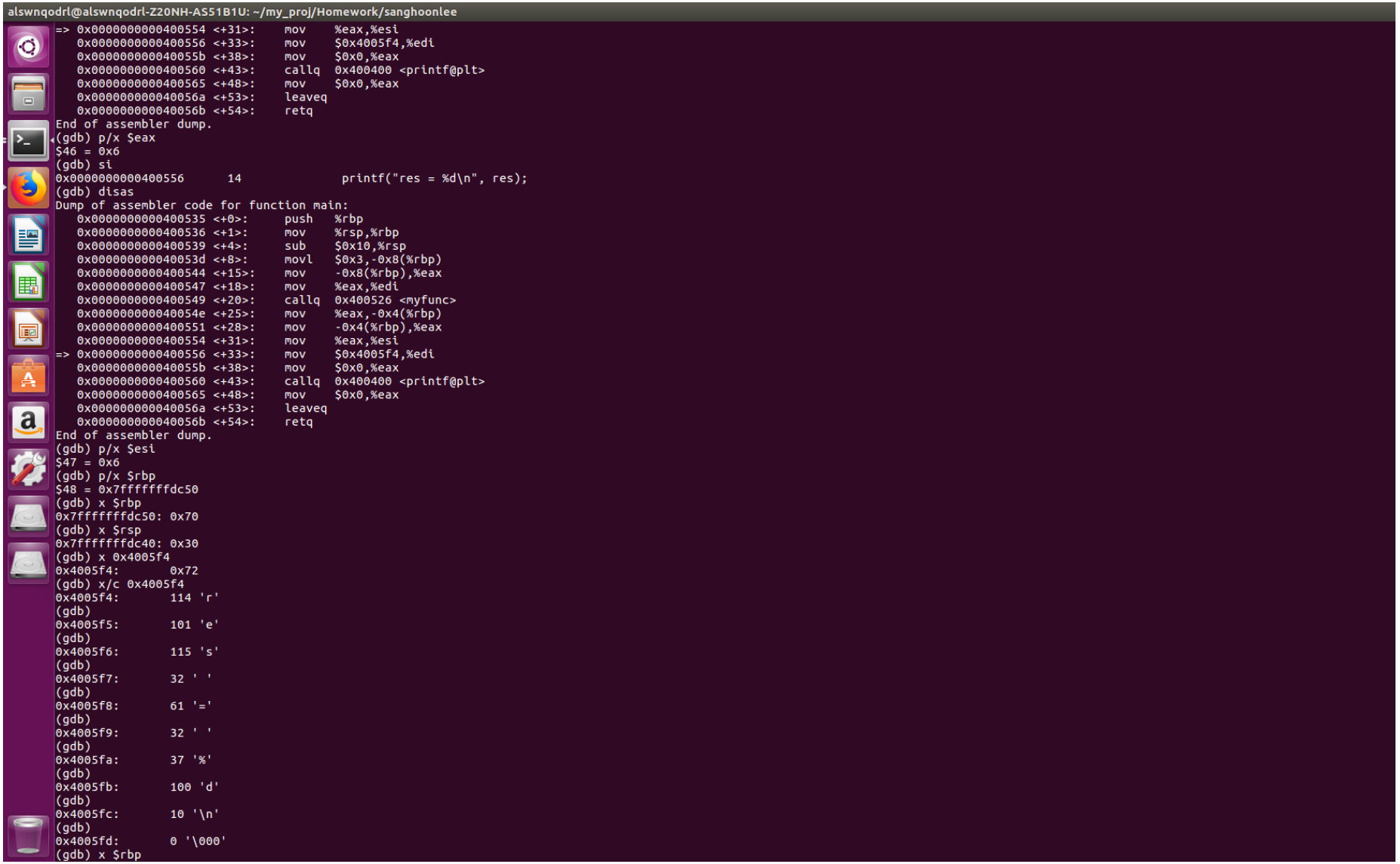
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) disas func1
No symbol "func1" in current context.
(gdb) disas func
No symbol "func" in current context.
(gdb) l
8 }
9
10 int main(void)
11 {
12 int num = 3, res;
13 res = myfunc(num);
14 printf("res = %d\n", res);
15
16 return 0;
17 }
(gdb) disas myfunc
Dump of assembler code for function myfunc:
0x0000000000400526 <+0>: push %rbp
0x0000000000400527 <+1>: mov %rsp,%rbp
0x000000000040052a <+4>: mov %edi,-0x4(%rbp)
0x000000000040052d <+7>: mov -0x4(%rbp),%eax
0x0000000000400530 <+10>: add $0x3,%eax
0x0000000000400533 <+13>: pop %rbp
0x0000000000400534 <+14>: retq
End of assembler dump.
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
=> 0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb)

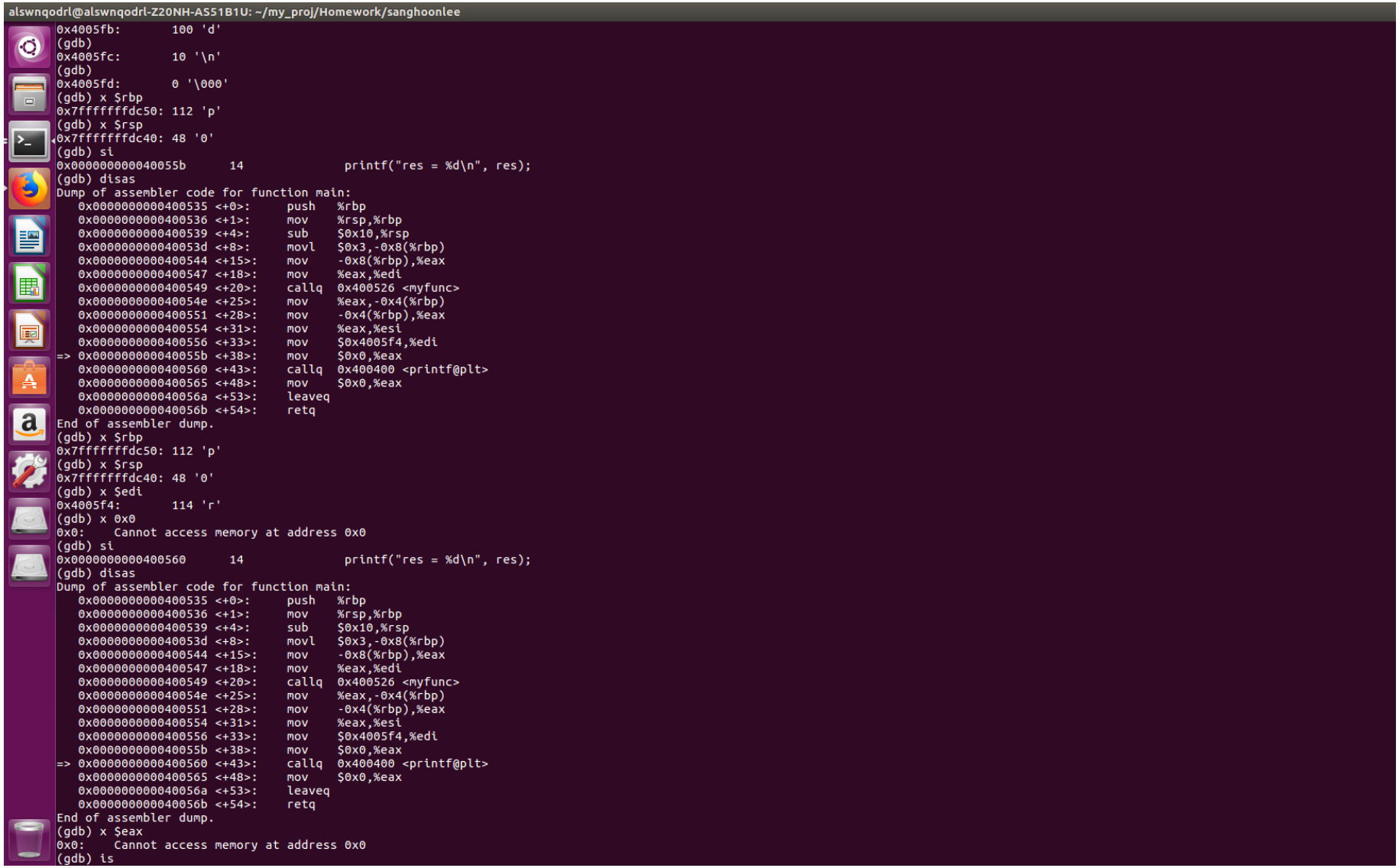
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x rbp-4
No symbol "rbp" in current context.
(gdb) p/x $rbp - 4
$42 = 0x7fffffffdc4c
(gdb) p/x $eax
$43 = 0x6
(gdb) si
14 printf("res = %d\n", res);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
=> 0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $rbp
$44 = 0x7fffffffdc50
(gdb) x $rsp
0x7fffffffdc40: 0x30
(gdb) p/x $eax
$45 = 0x6
(gdb) si
0x0000000000400554 14 printf("res = %d\n", res);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
=> 0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb) p/x $eax
$46 = 0x6
(gdb) si
0x0000000000400556 14 printf("res = %d\n", res);
(gdb) disas

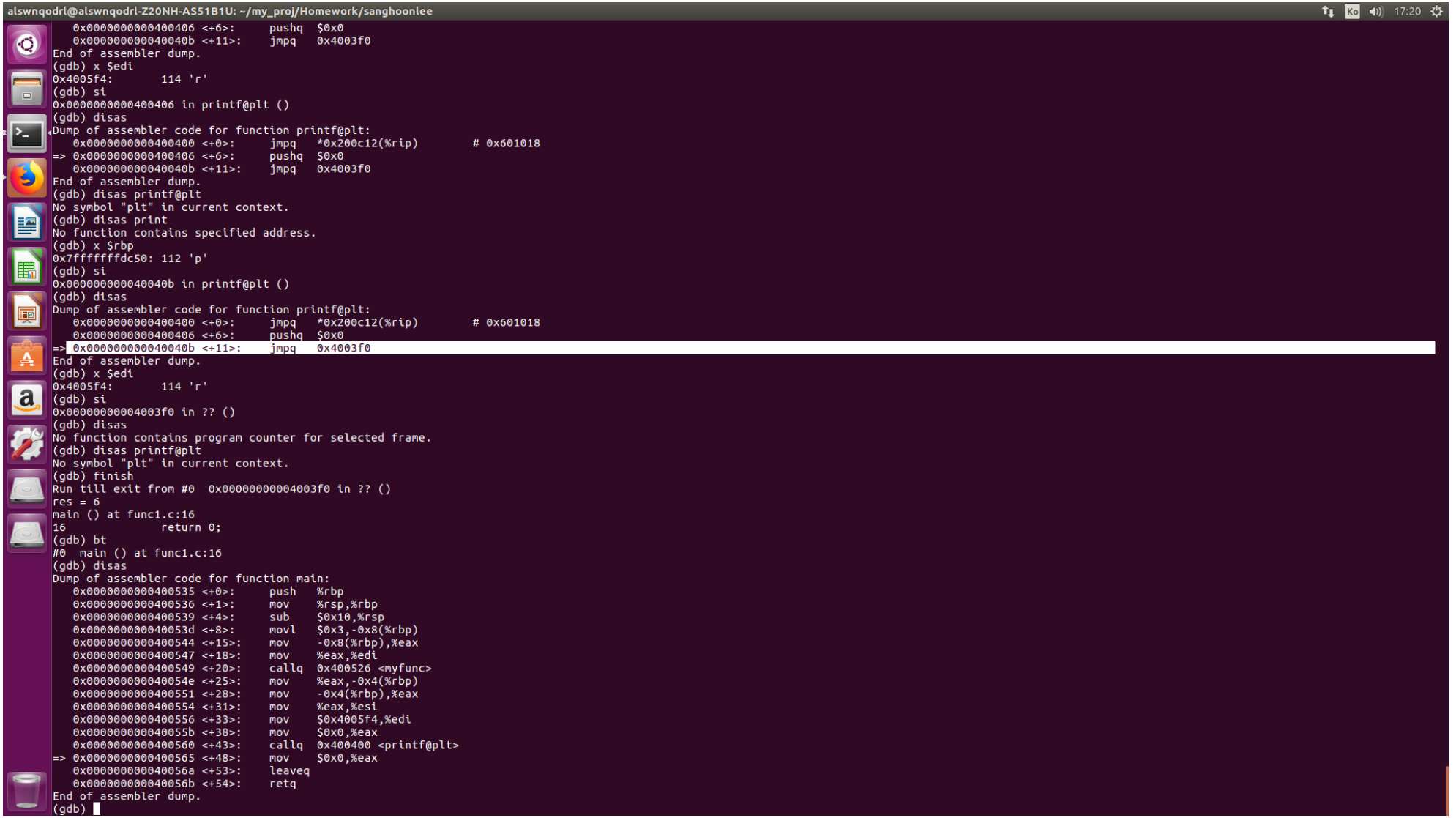
06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
=> 0x0000000000400554 <+31>:   mov   %eax,%esi
   0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $eax
$46 = 0x6
(gdb) si
0x0000000000400556   14   printf("res = %d\n", res);
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:   push   %rbp
   0x0000000000400536 <+1>:   mov   %rsp,%rbp
   0x0000000000400539 <+4>:   sub $0x10,%rsp
   0x000000000040053d <+8>:   movl $0x3,-0x8(%rbp)
   0x0000000000400544 <+15>:   mov   -0x8(%rbp),%eax
   0x0000000000400547 <+18>:   mov   %eax,%edi
   0x0000000000400549 <+20>:   callq 0x400526 <myfunc>
   0x000000000040054e <+25>:   mov   %eax,-0x4(%rbp)
   0x0000000000400551 <+28>:   mov   -0x4(%rbp),%eax
   0x0000000000400554 <+31>:   mov   %eax,%esi
=> 0x0000000000400556 <+33>:   mov $0x4005f4,%edi
   0x000000000040055b <+38>:   mov $0x0,%eax
   0x0000000000400560 <+43>:   callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:   mov $0x0,%eax
   0x000000000040056a <+53>:   leaveq
   0x000000000040056b <+54>:   retq
End of assembler dump.
(gdb) p/x $esi
$47 = 0x6
(gdb) p/x $rbp
$48 = 0x7fffffffdc50
(gdb) x $rbp
0x7fffffffdc50: 0x70
(gdb) x $rsp
0x7fffffffdc40: 0x30
(gdb) x 0x4005f4
0x4005f4:   0x72
(gdb) x/c 0x4005f4
0x4005f4:   114 'r'
(gdb)
0x4005f5:   101 'e'
(gdb)
0x4005f6:   115 's'
(gdb)
0x4005f7:   32 ' '
(gdb)
0x4005f8:   61 '='
(gdb)
0x4005f9:   32 ' '
(gdb)
0x4005fa: 37 '%'
(gdb)
0x4005fb: 100 'd'
(gdb)
0x4005fc: 10 '\n'
(gdb)
0x4005fd: 0 '\000'
(gdb) x $rbp

06 기계어 분석


alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
0x4005fb:	100 'd'
(gdb)
0x4005fc:	10 '\n'
(gdb)
0x4005fd:	0 '\000'
(gdb) x $rbp
0x7fffffffdc50: 112 'p'
(gdb) x $rsp
0x7fffffffdc40: 48 '0'
(gdb) si
0x000000000040055b	14		printf("res = %d\n", res);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>:	push %rbp
0x0000000000400536 <+1>:	mov %rsp,%rbp
0x0000000000400539 <+4>:	sub $0x10,%rsp
0x000000000040053d <+8>:	movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>:	mov -0x8(%rbp),%eax
0x0000000000400547 <+18>:	mov %eax,%edi
0x0000000000400549 <+20>:	callq 0x400526 <myfunc>
0x000000000040054e <+25>:	mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>:	mov -0x4(%rbp),%eax
0x0000000000400554 <+31>:	mov %eax,%esi
0x0000000000400556 <+33>:	mov $0x4005f4,%edi
=> 0x000000000040055b <+38>:	mov $0x0,%eax
0x0000000000400560 <+43>:	callq 0x400400 <printf@plt>
0x0000000000400565 <+48>:	mov $0x0,%eax
0x000000000040056a <+53>:	leaveq
0x000000000040056b <+54>:	retq
End of assembler dump.
(gdb) x $rbp
0x7fffffffdc50: 112 'p'
(gdb) x $rsp
0x7fffffffdc40: 48 '0'
(gdb) x $edi
0x4005f4:	114 'r'
(gdb) x 0x0
0x0:	Cannot access memory at address 0x0
(gdb) si
0x0000000000400560	14		printf("res = %d\n", res);
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>:	push %rbp
0x0000000000400536 <+1>:	mov %rsp,%rbp
0x0000000000400539 <+4>:	sub $0x10,%rsp
0x000000000040053d <+8>:	movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>:	mov -0x8(%rbp),%eax
0x0000000000400547 <+18>:	mov %eax,%edi
0x0000000000400549 <+20>:	callq 0x400526 <myfunc>
0x000000000040054e <+25>:	mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>:	mov -0x4(%rbp),%eax
0x0000000000400554 <+31>:	mov %eax,%esi
0x0000000000400556 <+33>:	mov $0x4005f4,%edi
0x000000000040055b <+38>:	mov $0x0,%eax
=> 0x0000000000400560 <+43>:	callq 0x400400 <printf@plt>
0x0000000000400565 <+48>:	mov $0x0,%eax
0x000000000040056a <+53>:	leaveq
0x000000000040056b <+54>:	retq
End of assembler dump.
(gdb) x $eax
0x0:	Cannot access memory at address 0x0
(gdb) is

06 기계어 분석

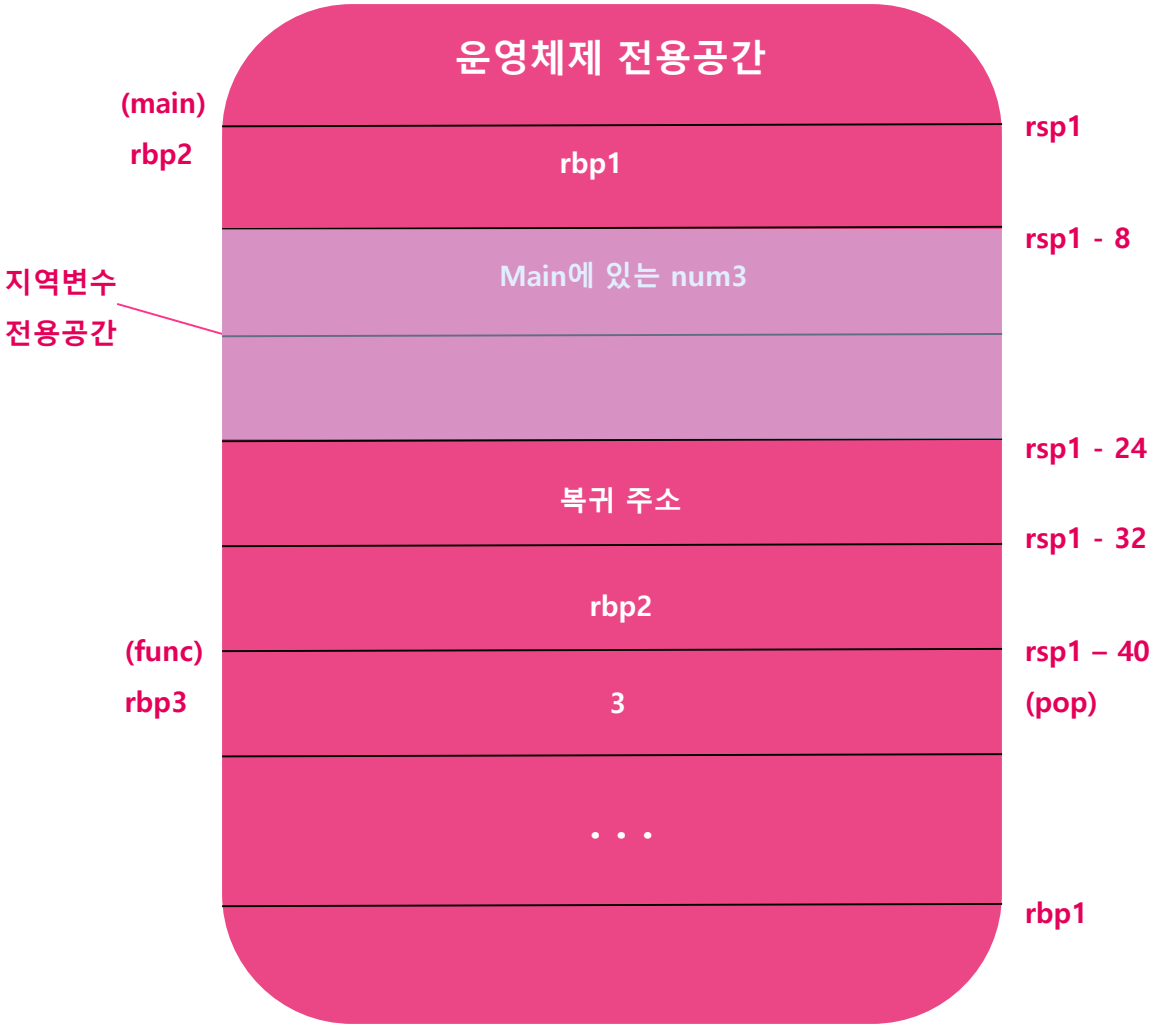

alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/sanghoonlee
Ko 17:20
0x0000000000400406 <+6>: pushq $0x0
0x000000000040040b <+11>: jmpq 0x4003f0
End of assembler dump.
(gdb) x $edi
0x4005f4: 114 'r'
(gdb) si
0x0000000000400406 in printf@plt ()
(gdb) disas
Dump of assembler code for function printf@plt:
0x0000000000400400 <+0>: jmpq *0x200c12(%rip) # 0x601018
=> 0x0000000000400406 <+6>: pushq $0x0
0x000000000040040b <+11>: jmpq 0x4003f0
End of assembler dump.
(gdb) disas printf@plt
No symbol "plt" in current context.
(gdb) disas print
No function contains specified address.
(gdb) x $rbp
0x7fffffffdc50: 112 'p'
(gdb) si
0x000000000040040b in printf@plt ()
(gdb) disas
Dump of assembler code for function printf@plt:
0x0000000000400400 <+0>: jmpq *0x200c12(%rip) # 0x601018
0x0000000000400406 <+6>: pushq $0x0
=> 0x000000000040040b <+11>: jmpq 0x4003f0
End of assembler dump.
(gdb) x $edi
0x4005f4: 114 'r'
(gdb) si
0x00000000004003f0 in ?? ()
(gdb) disas
No function contains program counter for selected frame.
(gdb) disas printf@plt
No symbol "plt" in current context.
(gdb) finish
Run till exit from #0 0x00000000004003f0 in ?? ()
res = 6
main () at func1.c:16
16 return 0;
(gdb) bt
#0 main () at func1.c:16
(gdb) disas
Dump of assembler code for function main:
0x0000000000400535 <+0>: push %rbp
0x0000000000400536 <+1>: mov %rsp,%rbp
0x0000000000400539 <+4>: sub $0x10,%rsp
0x000000000040053d <+8>: movl $0x3,-0x8(%rbp)
0x0000000000400544 <+15>: mov -0x8(%rbp),%eax
0x0000000000400547 <+18>: mov %eax,%edi
0x0000000000400549 <+20>: callq 0x400526 <myfunc>
0x000000000040054e <+25>: mov %eax,-0x4(%rbp)
0x0000000000400551 <+28>: mov -0x4(%rbp),%eax
0x0000000000400554 <+31>: mov %eax,%esi
0x0000000000400556 <+33>: mov $0x4005f4,%edi
0x000000000040055b <+38>: mov $0x0,%eax
0x0000000000400560 <+43>: callq 0x400400 <printf@plt>
=> 0x0000000000400565 <+48>: mov $0x0,%eax
0x000000000040056a <+53>: leaveq
0x000000000040056b <+54>: retq
End of assembler dump.
(gdb)

07 분석 그림

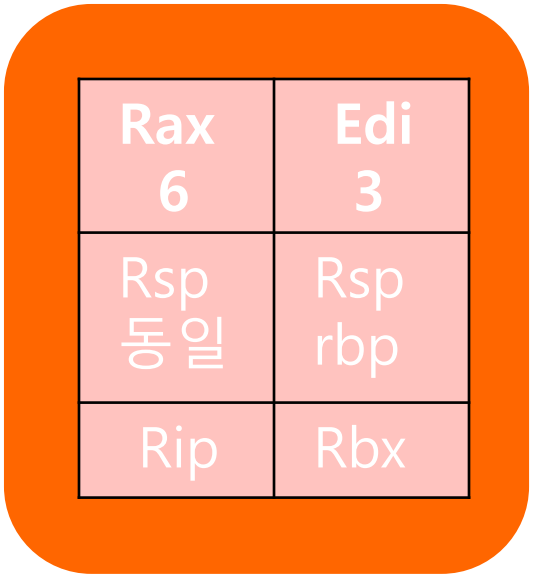
Push %rbp mov %rsp %rbp - stack Frame 생성

Pop %rbp retq - stack Frame 해제

=> 모든 함수는 위의 구조를 가지기 때문에
stack없이는 언어구동 불가능



메모리



CPU

08 진수 시스템 정리

진수시스템

16진수 0-f까지 총 16개로 컴퓨터가 사용

10진수 0-9까지 총 10개로 사람이 사용

8진수 0-7까지 총 8개로 리눅스 권한에 사용

3진수 0-2까지 총 3개로 RNA 분석에 사용

2진수 0-1까지 총 2개로 컴퓨터가 사용

둘 다 컴퓨터가 사용하지만
16진수가 가독성이 좋기 때문에
사람과 컴퓨터의 혼용어

2진수 16진수 변환 정리

33 = 16^1 * 2 + 16^0 *1 = 0x21

= 2^5 * 1 + 2^0 *1 = 100001(2)

2568 = 16^2 *a + 16^0 *8 = 0xa08

= 2^11 *1 + 2^9 *1 + 2^3 *1 = 101000001000(2)

8421 8421 8421 8421 8421 8421 8421 8421

0x48932110 = 0100 1000 1001 0011 0010 0001 0001 0001 (2)

감사합니다.