# Embedded Programmer과정 1,2일차 학습내용

김형주

#### Linux ubuntu 설치

- Iso 파일을 다운로드한다.(16.04.3 ver 권장)
- <a href="https://www.ubuntu.com/download/desktop">https://www.ubuntu.com/download/desktop</a>
- Google에 Universal USB Installer 검색 후 다 운로드한다.
- USB에 ubuntu iso파일을 설치한다.(usb에 자료가 있을 시 백업해둘 것!)
- 부팅옵션을 usb를 최우선으로 설정 후, 설명 에 따라 설치를 진행한다.

주의사항) window도 사용할 경우, aside window 옵션을 선택한다. 다른거 선택 시 window 제거 후 Linux 설치. 내용 읽어보고 진행할것

## Linux 기본 명령어

- pwd : 현재 디렉토리 위치를 보여준다
- Ctrl + Alt T: 터미널 open
- Is : 현재 디렉토리 목록 display(list)
- cd 디렉토리 이동('절대경로'or'상대경로' 이용) (chage directory)
- vi filename.c : .c파일 생성 / '편집모드'or'명령모드' 사용 / :w :q :wq 저장,종료,저장후 종료
- mkdir directoryname : 디렉토리 만들기
- Gcc filename.c : filename.c를 컴파일
- Gcc –o test filename.c : test 이름으로 실행파일생성
- Gcc -g -o debug filename.c : 디버깅 파일 생성

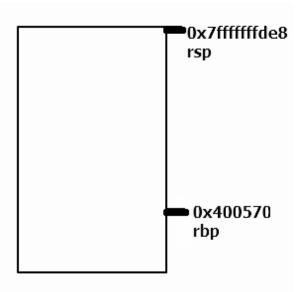
## Linux 기본 명령어 (명령모드)

- d3d, d5d 등 숫자만큼 줄 제거
- y3y, y5y 등 숫자만큼 줄 복사
- pp 붙여넣기
- u 되돌리기(ctrl z)
- ctrl r 되돌리기 취소(ctrl shift z)
- set ~ 기타 편집기 환경설정
   ex) :set number 행 줄수 표시

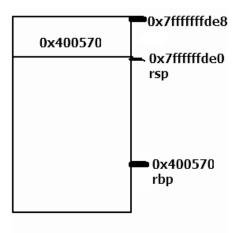
#### C언어 프로그래밍

- 변수 : 메모리로부터 할당받은 data를 저 장할 수 있는 공간
- Data형 : int(정수형), float(실수형), double(실수형)
- 함수: int sum(int num1,int num2)처럼 반 환값, 입력값이 존재

```
🔊 🖃 📵 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
(gdb) disas
Dump of assembler code for function main:
=> 0x0000000000400535 <+0>:
                                push
                                       %rbp
   0x0000000000400536 <+1>:
                                       %rsp,%rbp
                                mov
   0x0000000000400539 <+4>:
                                sub
                                       $0x10,%rsp
   0x000000000040053d <+8>:
                                       $0x3,-0x8(%rbp)
                                movl
   0x0000000000400544 <+15>:
                                       -0x8(%rbp),%eax
                                mov
   0x0000000000400547 <+18>:
                                       %eax.%edi
                                mov
   0x0000000000400549 <+20>:
                                callq
                                       0x400526 <myfunc>
   0x000000000040054e <+25>:
                                       %eax,-0x4(%rbp)
                                mov
                                       -0x4(%rbp),%eax
   0x0000000000400551 <+28>:
                                mov
   0x00000000000400554 <+31>:
                                       %eax,%esi
                                mov
                                       $0x4005f4.%edi
   0x00000000000400556 <+33>:
                                mov
   0x000000000040055b <+38>:
                                       $0x0.%eax
                                MOV
   0x0000000000400560 <+43>:
                                callq 0x400400 <printf@plt>
   0x0000000000400565 <+48>:
                                MOV
                                        $0x0,%eax
   0x0000000000040056a <+53>:
                                leaveq
  0x0000000000040056b <+54>:
                                retq
End of assembler dump.
(gdb) p/x $rbp
$1 = 0x400570
(gdb) p/x $rsp
S2 = 0x7fffffffdce8
```

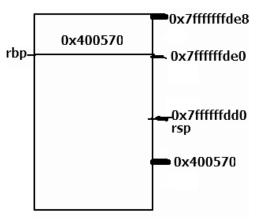


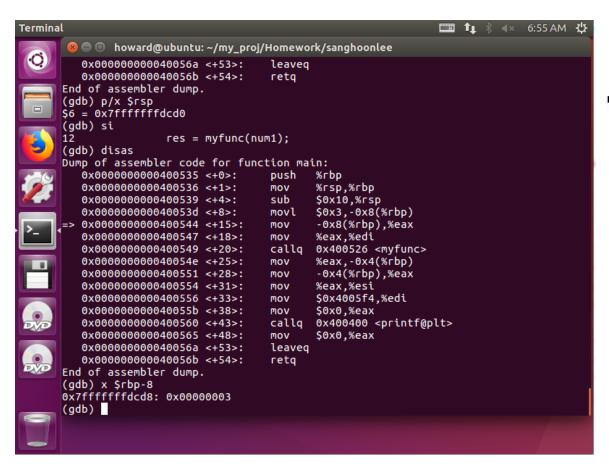
```
🔊 🖨 📵 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
                        10
0x0000000000400536
(qdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                 push
                                        %rbp
=> 0x0000000000400536 <+1>:
                                 mov
                                        %rsp,%rbp
   0x0000000000400539 <+4>:
                                 sub
                                        $0x10,%rsp
                                       $0x3,-0x8(%rbp)
   0x000000000040053d <+8>:
                                movl
                                       -0x8(%rbp),%eax
   0x0000000000400544 <+15>:
                                 MOV
   0x0000000000400547 <+18>:
                                        %eax,%edi
                                 mov
                                 callq 0x400526 <myfunc>
   0x00000000000400549 <+20>:
   0x000000000040054e <+25>:
                                       %eax,-0x4(%rbp)
                                 MOV
   0x0000000000400551 <+28>:
                                        -0x4(%rbp),%eax
                                mov
   0x00000000000400554 <+31>:
                                        %eax,%esi
                                 MOV
   0x0000000000400556 <+33>:
                                        $0x4005f4,%edi
                                 mov
   0x000000000040055b <+38>:
                                 mov
                                        $0x0,%eax
                                callq 0x400400 <printf@plt>
   0x00000000000400560 <+43>:
   0x00000000000400565 <+48>:
                                        $0x0,%eax
                                 mov
   0x0000000000040056a <+53>:
                                 leaveq
   0x000000000040056b <+54>:
                                retq
End of assembler dump.
(qdb) p/x rsp
No symbol "rsp" in current context.
(gdb) p/x $rsp
$3 = 0x7fffffffdce0
(gdb) p/x $rbp
$4 = 0x400570
(gdb) x $rsp
0x7fffffffdce0: 0x00400570
(adb)
```

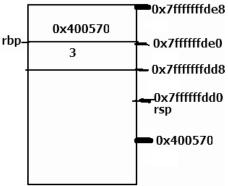


```
🕒 🗊 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
S3 = 0x7fffffffdce0
(gdb) p/x $rbp
                                                                                                            0x7fffffffde8
$4 = 0x400570
                                                                                            0x400570
(gdb) x $rsp
                                                                                    rbp.
                                                                                                            0x7ffffffde0
0x7fffffffdce0: 0x00400570
                                                                                                            rsp
(adb) si
0x0000000000400539
                                 {
                        10
(qdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                 push
                                        %rbp
   0x0000000000400536 <+1>:
                                        %rsp,%rbp
                                 mov
=> 0x0000000000400539 <+4>:
                                 sub
                                        $0x10,%rsp
                                                                                                           ■ 0x400570
   0x000000000040053d <+8>:
                                        $0x3,-0x8(%rbp)
                                 movl
   0x00000000000400544 <+15>:
                                        -0x8(%rbp),%eax
                                 mov
   0x0000000000400547 <+18>:
                                        %eax.%edi
                                 mov
   0x0000000000400549 <+20>:
                                 calla
                                        0x400526 <myfunc>
   0x0000000000040054e <+25>:
                                        %eax,-0x4(%rbp)
                                 mov
   0x0000000000400551 <+28>:
                                        -0x4(%rbp),%eax
                                 mov
                                        %eax.%esi
   0x0000000000400554 <+31>:
                                 mov
   0x0000000000400556 <+33>:
                                        $0x4005f4,%edi
                                 MOV
   0x000000000040055b <+38>:
                                 mov
                                        $0x0.%eax
                                 callq 0x400400 <printf@plt>
   0x0000000000400560 <+43>:
   0x0000000000400565 <+48>:
                                        $0x0, %eax
                                 mov
   0x000000000040056a <+53>:
                                 leaveg
   0x000000000040056b <+54>:
                                 retq
End of assembler dump.
(gdb) p/x $rbp
$5 = 0x7fffffffdce0
(gdb)
```

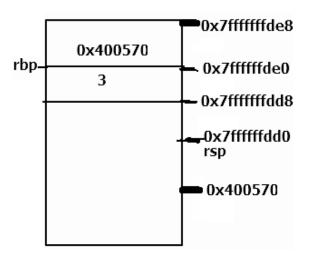
```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
End of assembler dump.
(qdb) p/x $rbp
$5 = 0x7ffffffdce0
(gdb) si
Breakpoint 1, main () at func1.c:11
11
                int num1 = 3, res;
(gdb) disas
Dump of assembler code for function main:
  0x0000000000400535 <+0>:
                                 push
                                        %rbp
                                        %rsp,%rbp
  0x0000000000400536 <+1>:
                                 mov
  0x0000000000400539 <+4>:
                                 sub
                                        $0x10,%rsp
                                        $0x3,-0x8(%rbp)
=> 0x000000000040053d <+8>:
                                 movl
  0x00000000000400544 <+15>:
                                mov
                                        -0x8(%rbp),%eax
  0x00000000000400547 <+18>:
                                MOV
                                        %eax.%edi
                                callq 0x400526 <myfunc>
  0x00000000000400549 <+20>:
                                        %eax,-0x4(%rbp)
  0x0000000000040054e <+25>:
                                MOV
                                        -0x4(%rbp),%eax
  0x0000000000400551 <+28>:
                                 mov
  0x00000000000400554 <+31>:
                                        %eax,%esi
                                mov
                                        S0x4005f4.%edi
  0x0000000000400556 <+33>:
                                mov
  0x000000000040055b <+38>:
                                        $0x0, %eax
                                 mov
                                callq 0x400400 <printf@plt>
  0x0000000000400560 <+43>:
  0x00000000000400565 <+48>:
                                 MOV
                                        $0x0, %eax
  0x0000000000040056a <+53>:
                                 leaveg
  0x000000000040056b <+54>:
                                 retq
End of assembler dump.
(gdb) p/x $rsp
$6 = 0x7ffffffdcd0
(gdb)
```

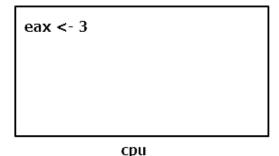




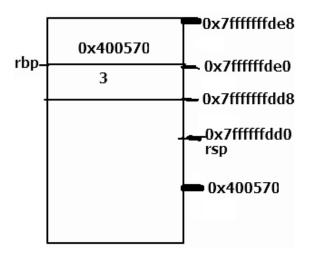


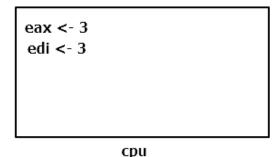
```
howard@ubuntu: ~/my proj/Homework/sanghoonlee
   0x000000000040056a <+53>:
                                 leaveq
   0x000000000040056b <+54>:
                                 reta
End of assembler dump.
(qdb) x Srbp-8
0x7fffffffdcd8: 0x00000003
(qdb) si
                                         res = myfunc(num1);
0x0000000000400547
                        12
(gdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                 push
                                        %rbp
                                        %rsp,%rbp
   0x0000000000400536 <+1>:
                                 MOV
                                        $0x10,%rsp
   0x00000000000400539 <+4>:
                                 sub
                                        $0x3,-0x8(%rbp)
   0x000000000040053d <+8>:
                                 movl
                                        -0x8(%rbp),%eax
   0x00000000000400544 <+15>:
                                 mov
                                        %eax,%edi
=> 0x00000000000400547 <+18>:
                                 mov
                                        0x400526 <mvfunc>
   0x0000000000400549 <+20>:
                                callq
                                        %eax,-0x4(%rbp)
   0x000000000040054e <+25>:
                                 MOV
                                        -0x4(%rbp),%eax
   0x0000000000400551 <+28>:
                                 mov
   0x00000000000400554 <+31>:
                                 MOV
                                        %eax,%esi
   0x0000000000400556 <+33>:
                                        $0x4005f4,%edi
                                 MOV
                                        $0x0,%eax
   0x0000000000040055b <+38>:
                                 MOV
                                 callq
                                        0x400400 <printf@plt>
   0x0000000000400560 <+43>:
   0x0000000000400565 <+48>:
                                        $0x0.%eax
                                 MOV
   0x000000000040056a <+53>:
                                 leaveg
   0x000000000040056b <+54>:
                                 retq
End of assembler dump.
(gdb) p/x $eax
$7 = 0x3
(ddb)
```



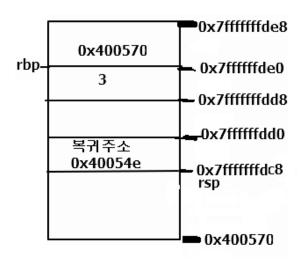


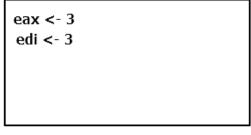
```
noward@ubuntu: ~/my_proj/Homework/sanghoonlee
                                callq 0x400400 <printf@plt>
   0x00000000000400560 <+43>:
                                        $0x0.%eax
   0x0000000000400565 <+48>:
                                MOV
                                leaveg
   0x000000000040056a <+53>:
   0x000000000040056b <+54>:
                                retq
End of assembler dump.
(gdb) p/x $eax
57 = 0x3
(adb) si
                        12
                                         res = myfunc(num1);
0x0000000000400549
(adb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                push
                                       %rbp
                                       %rsp,%rbp
   0x0000000000400536 <+1>:
                                MOV
   0x0000000000400539 <+4>:
                                 sub
                                       $0x10,%rsp
                                       $0x3,-0x8(%rbp)
   0x000000000040053d <+8>:
                                movl
                                        -0x8(%rbp),%eax
   0x00000000000400544 <+15>:
                                mov
   0x0000000000400547 <+18>:
                                       %eax.%edi
                                mov
=> 0x00000000000400549 <+20>:
                                calla
                                       0x400526 <myfunc>
                                       %eax,-0x4(%rbp)
   0x0000000000040054e <+25>:
                                MOV
                                        -0x4(%rbp),%eax
   0x0000000000400551 <+28>:
                                mov
   0x0000000000400554 <+31>:
                                       %eax,%esi
                                mov
                                       $0x4005f4,%edi
   0x0000000000400556 <+33>:
                                mov
   0x000000000040055b <+38>:
                                       $0x0,%eax
                                MOV
                                       0x400400 <printf@plt>
                                calla
   0x0000000000400560 <+43>:
   0x0000000000400565 <+48>:
                                        $0x0,%eax
                                mov
   0x000000000040056a <+53>:
                                 leaveg
   0x000000000040056b <+54>:
                                 retq
End of assembler dump.
(dbp)
```





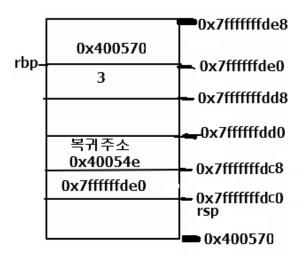
```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
  0x00000000000400549 <+20>:
                                callq 0x400526 <mvfunc>
  0x0000000000040054e <+25>:
                                       %eax,-0x4(%rbp)
                                mov
                                       -0x4(%rbp),%eax
  0x0000000000400551 <+28>:
                                mov
                                       %eax,%esi
  0x0000000000400554 <+31>:
                                MOV
                                       $0x4005f4,%edi
  0x0000000000400556 <+33>:
                                mov
  0x000000000040055b <+38>:
                                       $0x0,%eax
                                mov
                                callq 0x400400 <printf@plt>
  0x0000000000400560 <+43>:
  0x0000000000400565 <+48>:
                                       $0x0,%eax
                                mov
  0x000000000040056a <+53>:
                                leaveg
  0x000000000040056b <+54>:
                                retq
End of assembler dump.
(adb) si
myfunc (num=0) at func1.c:4
(adb) disas
Dump of assembler code for function myfunc:
=> 0x0000000000400526 <+0>:
                                push
                                       %rbp
                                       %rsp,%rbp
  0x00000000000400527 <+1>:
                                mov
                                       %edi,-0x4(%rbp)
  0x0000000000040052a <+4>:
                                mov
                                       -0x4(%rbp),%eax
  0x000000000040052d <+7>:
                                mov
                                       $0x3,%eax
  0x0000000000400530 <+10>:
                                add
                                pop
                                       %гЬр
  0x0000000000400533 <+13>:
  0x0000000000400534 <+14>:
                                retq
End of assembler dump.
(gdb) p/x $rsp
$8 = 0x7ffffffdcc8
(gdb) x $rsp
0x7fffffffdcc8: 0x0040054e
(ddb)
```

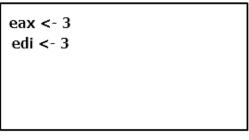




cpu

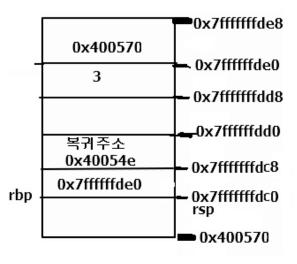
```
🕒 🗊 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
                                        %edi,-0x4(%rbp)
   0x0000000000040052a <+4>:
                                MOV
                                        -0x4(%rbp),%eax
  0x000000000040052d <+7>:
                                mov
  0x00000000000400530 <+10>:
                                add
                                        $0x3,%eax
                                        %гьр
  0x0000000000400533 <+13>:
                                pop
  0x00000000000400534 <+14>:
                                retq
End of assembler dump.
(gdb) p/x $rsp
$8 = 0x7fffffffdcc8
(gdb) x $rsp
0x7fffffffdcc8: 0x0040054e
(gdb) si
0x0000000000400527
(adb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                push
                                        %rbp
=> 0x0000000000400527 <+1>:
                                        %rsp,%rbp
                                MOV
   0x000000000040052a <+4>:
                                        %edi,-0x4(%rbp)
                                mov
  0x000000000040052d <+7>:
                                        -0x4(%rbp),%eax
                                mov
  0x0000000000400530 <+10>:
                                add
                                        $0x3,%eax
                                        %гьр
  0x00000000000400533 <+13>:
                                DOD
  0x00000000000400534 <+14>:
                                retq
End of assembler dump.
(gdb) p/x $rsp
S9 = 0x7fffffffdcc0
(gdb) p/x rsp
No symbol "rsp" in current context.
(qdb) x $rsp
0x7fffffffdcc0: 0xffffdce0
```

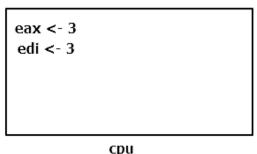




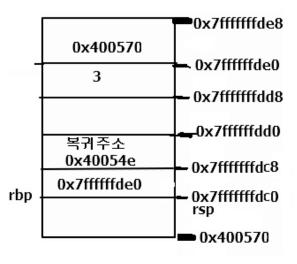
cpu

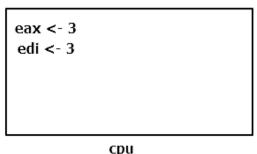
```
🔊 🖨 🗊 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
   0x0000000000400526 <+0>:
                                push
                                        %rbp
                                        %rsp,%rbp
=> 0x0000000000400527 <+1>:
                                MOV
                                        %edi,-0x4(%rbp)
   0x000000000040052a <+4>:
                                MOV
                                        -0x4(%rbp),%eax
   0x000000000040052d <+7>:
                                mov
   0x0000000000400530 <+10>:
                                 add
                                        $0x3,%eax
                                        %гьр
   0x0000000000400533 <+13>:
                                 pop
   0x0000000000400534 <+14>:
                                retq
End of assembler dump.
(gdb) p/x $rsp
$9 = 0x7fffffffdcc0
(gdb) p/x rsp
No symbol "rsp" in current context.
(adb) x Srsp
0x7fffffffdcc0: 0xffffdce0
(adb) si
0x000000000040052a
(qdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                 push
                                        %гьр
                                        %rsp,%rbp
   0x0000000000400527 <+1>:
                                MOV
=> 0x000000000040052a <+4>:
                                        %edi,-0x4(%rbp)
                                MOV
                                        -0x4(%rbp),%eax
   0x0000000000040052d <+7>:
                                MOV
                                add
                                        $0x3,%eax
   0x0000000000400530 <+10>:
                                        %гьр
   0x0000000000400533 <+13>:
                                pop
                                retq
   0x0000000000400534 <+14>:
End of assembler dump.
(gdb) p/x $rbp
$10 = 0x7fffffffdcc0
(dbp)
```



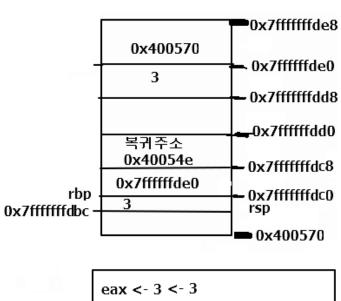


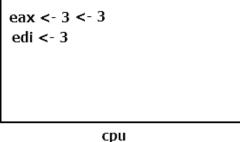
```
🔊 🖨 🗊 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
   0x0000000000400526 <+0>:
                                push
                                        %rbp
                                        %rsp,%rbp
=> 0x0000000000400527 <+1>:
                                MOV
                                        %edi,-0x4(%rbp)
   0x000000000040052a <+4>:
                                MOV
                                        -0x4(%rbp),%eax
   0x000000000040052d <+7>:
                                mov
   0x0000000000400530 <+10>:
                                 add
                                        $0x3,%eax
                                        %гьр
   0x0000000000400533 <+13>:
                                 pop
   0x0000000000400534 <+14>:
                                retq
End of assembler dump.
(gdb) p/x $rsp
$9 = 0x7fffffffdcc0
(gdb) p/x rsp
No symbol "rsp" in current context.
(adb) x Srsp
0x7fffffffdcc0: 0xffffdce0
(adb) si
0x000000000040052a
(qdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                 push
                                        %гьр
                                        %rsp,%rbp
   0x0000000000400527 <+1>:
                                MOV
=> 0x000000000040052a <+4>:
                                        %edi,-0x4(%rbp)
                                MOV
                                        -0x4(%rbp),%eax
   0x0000000000040052d <+7>:
                                MOV
                                add
                                        $0x3,%eax
   0x0000000000400530 <+10>:
                                        %гьр
   0x0000000000400533 <+13>:
                                pop
                                retq
   0x0000000000400534 <+14>:
End of assembler dump.
(gdb) p/x $rbp
$10 = 0x7fffffffdcc0
(dbp)
```



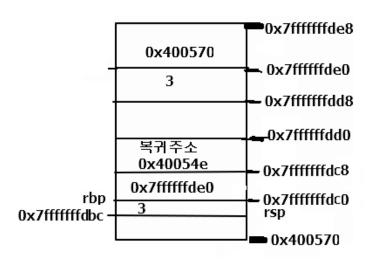


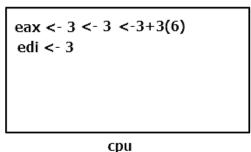
```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
(gdb) si
                return num + 3;
(qdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                push
                                       %гьр
   0x0000000000400527 <+1>:
                                       %rsp,%rbp
                                mov
                                       %edi,-0x4(%rbp)
   0x000000000040052a <+4>:
                                mov
                                       -0x4(%rbp),%eax
=> 0x000000000040052d <+7>:
                                mov
                                       $0x3,%eax
   0x0000000000400530 <+10>:
                                add
   0x0000000000400533 <+13>:
                                pop
                                        %rbp
                                retq
   0x0000000000400534 <+14>:
End of assembler dump.
(gdb) x $rbp-4
0x7fffffffdcbc: 0x00000003
(gdb) si
0x0000000000400530
                                        return num + 3:
                        6
(qdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                push
                                        %rbp
   0x0000000000400527 <+1>:
                                mov
                                       %rsp,%rbp
                                       %edi.-0x4(%rbp)
   0x0000000000040052a <+4>:
                                mov
                                        -0x4(%rbp),%eax
   0x000000000040052d <+7>:
                                mov
                                        $0x3,%eax
=> 0x0000000000400530 <+10>:
                                add
   0x0000000000400533 <+13>:
                                        %rbp
                                pop
   0x0000000000400534 <+14>:
                                retq
End of assembler dump.
(qdb) p/x $eax
$11 = 0x3
(gdb)
```



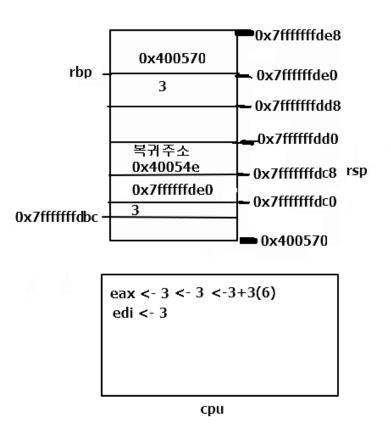


```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
(qdb) si
0x0000000000400530
                                         return num + 3;
                        б
(qdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                push
                                        %гьр
                                       %rsp,%rbp
   0x0000000000400527 <+1>:
                                mov
                                       %edi,-0x4(%rbp)
   0x000000000040052a <+4>:
                                mov
                                       -0x4(%rbp),%eax
   0x000000000040052d <+7>:
                                mov
=> 0x0000000000400530 <+10>:
                                add
                                       $0x3,%eax
   0x0000000000400533 <+13>:
                                        %гьр
                                pop
                                retq
   0x0000000000400534 <+14>:
End of assembler dump.
(gdb) p/x $eax
$11 = 0x3
(adb) si
(adb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                push
                                        %гьр
                                        %rsp,%rbp
   0x00000000000400527 <+1>:
                                MOV
                                       %edi,-0x4(%rbp)
   0x000000000040052a <+4>:
                                mov
                                        -0x4(%rbp),%eax
   0x0000000000040052d <+7>:
                                mov
                                        $0x3,%eax
                                add
   0x0000000000400530 <+10>:
                                        %гьр
=> 0x0000000000400533 <+13>:
                                DOD
   0x0000000000400534 <+14>:
                                reta
End of assembler dump.
(gdb) p/x $eax
$12 = 0x6
(dbp)
```

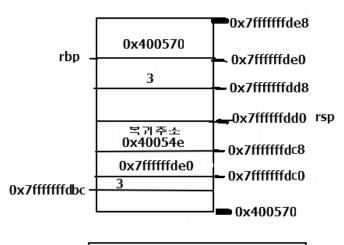


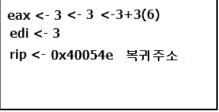


```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
(adb) si
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                 push
                                        %rbp
                                        %rsp,%rbp
  0x00000000000400527 <+1>:
                                mov
   0x000000000040052a <+4>:
                                        %edi,-0x4(%rbp)
                                mov
                                        -0x4(%rbp),%eax
  0x000000000040052d <+7>:
                                mov
  0x0000000000400530 <+10>:
                                add
                                        S0x3.%eax
=> 0x0000000000400533 <+13>:
                                 pop
                                        %rbp
   0x0000000000400534 <+14>:
                                reta
End of assembler dump.
(gdb) p/x $eax
$12 = 0x6
(gdb) si
0x0000000000400534
(gdb) disas
Dump of assembler code for function myfunc:
   0x0000000000400526 <+0>:
                                 push
                                        %rbp
                                        %rsp,%rbp
  0x00000000000400527 <+1>:
                                mov
                                        %edi,-0x4(%rbp)
  0x000000000040052a <+4>:
                                mov
                                        -0x4(%rbp),%eax
  0x000000000040052d <+7>:
                                mov
  0x0000000000400530 <+10>:
                                 add
                                        $0x3,%eax
  0x0000000000400533 <+13>:
                                 pop
                                        %rbp
=> 0x0000000000400534 <+14>:
                                 retq
End of assembler dump.
(gdb) p/x $rbp
$13 = 0x7fffffffdce0
(gdb)
```



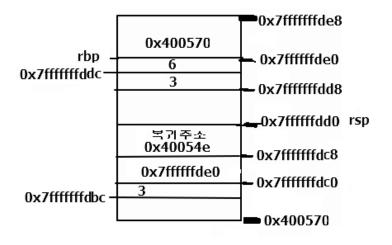
```
(gdb) p/x $rsp
$14 = 0x7fffffffdcc8
(qdb) si
0x000000000040054e in main () at func1.c:12
12
                res = myfunc(num1);
(qdb) disas
Dump of assembler code for function main:
   0x00000000000400535 <+0>:
                                 push
                                        %rbp
   0x00000000000400536 <+1>:
                                        %rsp,%rbp
                                 MOV
   0x00000000000400539 <+4>:
                                 sub
                                        $0x10,%rsp
                                        $0x3,-0x8(%rbp)
   0x000000000040053d <+8>:
                                 movl
   0x00000000000400544 <+15>:
                                 mov
                                        -0x8(%rbp),%eax
   0x0000000000400547 <+18>:
                                        %eax,%edi
                                 mov
   0x00000000000400549 <+20>:
                                 callq 0x400526 <myfunc>
                                        %eax,-0x4(%rbp)
=> 0x0000000000040054e <+25>:
                                 mov
   0x0000000000400551 <+28>:
                                        -0x4(%rbp),%eax
                                 MOV
   0x00000000000400554 <+31>:
                                        %eax,%esi
                                 mov
   0x0000000000400556 <+33>:
                                        $0x4005f4,%edi
                                 MOV
   0x000000000040055b <+38>:
                                 mov
                                        S0x0.%eax
                                 callq 0x400400 <printf@plt>
   0x00000000000400560 <+43>:
   0x00000000000400565 <+48>:
                                 mov
                                        $0x0,%eax
   0x0000000000040056a <+53>:
                                 leaveg
   0x0000000000040056b <+54>:
                                 retq
End of assembler dump.
(gdb) p/x $rip
$15 = 0x40054e
(gdb) p/x $rsp
$16 = 0x7ffffffdcd0
(gdb)
```

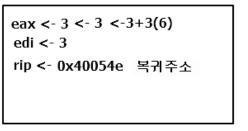




cpu

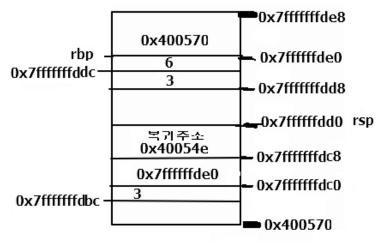
```
howard@ubuntu: ~/my_proj/Homework/sanghoonlee
0x7fffffffdcc8: 0x0040054e
(qdb) si
                printf("res = %d\n", res);
13
(qdb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                        %rbp
                                 push
   0x0000000000400536 <+1>:
                                        %rsp,%rbp
                                mov
                                        $0x10,%rsp
   0x0000000000400539 <+4>:
                                 sub
                                       $0x3,-0x8(%rbp)
   0x000000000040053d <+8>:
                                 movl
   0x0000000000400544 <+15>:
                                        -0x8(%rbp),%eax
                                 MOV
                                       %eax.%edi
   0x0000000000400547 <+18>:
                                 mov
   0x0000000000400549 <+20>:
                                callq 0x400526 <myfunc>
                                       %eax,-0x4(%rbp)
   0x0000000000040054e <+25>:
                                 MOV
                                        -0x4(%rbp),%eax
=> 0x0000000000400551 <+28>:
                                mov
   0x0000000000400554 <+31>:
                                        %eax.%esi
                                 MOV
   0x0000000000400556 <+33>:
                                MOV
                                        $0x4005f4.%edi
   0x000000000040055b <+38>:
                                        S0x0.%eax
                                 MOV
                                callq 0x400400 <printf@plt>
   0x0000000000400560 <+43>:
   0x0000000000400565 <+48>:
                                 MOV
                                        $0x0,%eax
   0x000000000040056a <+53>:
                                 leaveq
   0x000000000040056b <+54>:
                                 reta
End of assembler dump.
(qdb) x $rbp - 8
0x7fffffffdcd8: 0x00000003
(qdb) x %rpb - 4
A syntax error in expression, near `%rpb - 4'.
(qdb) x $rbp - 4
0x7fffffffdcdc: 0x00000006
(ddb)
```

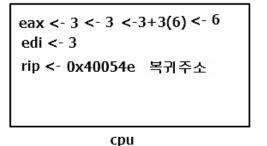




cpu

```
🔊 🖃 🗊 howard@ubuntu: ~/my_proj/Homework/sanghoonlee
A syntax error in expression, near `%rpb - 4'.
(qdb) x Srbp - 4
0x7fffffffdcdc: 0x00000006
(adb) si
                                         printf("res = %d\n", r
0x0000000000400554
                        13
(adb) disas
Dump of assembler code for function main:
   0x0000000000400535 <+0>:
                                push
                                        %rbp
                                        %rsp,%rbp
   0x00000000000400536 <+1>:
                                mov
   0x0000000000400539 <+4>:
                                sub
                                        $0x10,%rsp
   0x000000000040053d <+8>:
                                movl
                                        $0x3.-0x8(%rbp)
   0x00000000000400544 <+15>:
                                mov
                                        -0x8(%rbp),%eax
                                        %eax,%edi
   0x00000000000400547 <+18>:
                                mov
                                callq 0x400526 <myfunc>
   0x0000000000400549 <+20>:
                                        %eax,-0x4(%rbp)
   0x000000000040054e <+25>:
                                mov
                                        -0x4(%rbp),%eax
   0x0000000000400551 <+28>:
                                mov
                                        %eax,%esi
=> 0x0000000000400554 <+31>:
                                mov
                                        $0x4005f4.%edi
   0x0000000000400556 <+33>:
                                mov
   0x000000000040055b <+38>:
                                        $0x0,%eax
                                mov
                                callq 0x400400 <printf@plt>
   0x0000000000400560 <+43>:
   0x0000000000400565 <+48>:
                                        $0x0.%eax
                                MOV
   0x0000000000040056a <+53>:
                                leaveg
   0x0000000000040056b <+54>:
                                retq
End of assembler dump.
(gdb) p/x %eax
A syntax error in expression, near `%eax'.
(qdb) p/x $eax
$19 = 0x6
(gdb)
```





#### 포인터 크기

- 포인터란 주소를 저장하는 변수로 그 크기 는 자료형과 관계가 없고, os(system)의 비 트수에 따라 결정된다.
- 8bit : 1 byte
- 16bit : 2byte
- 32bit : 4byte
- 64bit : 8byte
- Sizeof(int\*) 등을 통해 확인해볼것

#### 2진수,16진수 변환

- 2진수란 0,1로 이루어진 숫자로, 비트를 표현할 때 사용한다.
- 제일 작은 수부터, 2^0,2^1,2^2, ... 을 나타낸다.
- 예) 1101(2)는 2^3+2^2+2^0=13(10)을 나타낸다.
- 16진수란 0~15로 이루어진 숫자로, 10부터는 알 파벳을 통해 표현한다.
- 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
- 2진수 -> 16진수 변환 시, 4bit씩 끊어서 변환한다.
- 예)1101 1001 0001 0010 0001 0001(2)
- -> d a 1 2 1 1 = 0xda1211