gdb의 breakpoint

우리집에 GDB 있는데... 메모리 보고갈래?(2)

위의 문서를 읽어보면

> b *[메모리주소]
> breakpoint를 거는 명령
> [메모리 주소]나 [함수의 이름] 혹은
> 이를 기준으로 한 [offset <+0>]으로
> breakpoint를 걸어도 됩니당 * breakpoint를 걸 땐, 주소 앞에 *를 붙이세요!

–

> (gdb) b *main
> Breakpoint 1 at 0x80484e6
> (gdb) b *0x080484e6
> Note: breakpoint 1 also set at pc 0x80484e6
> Breakpoint 2 at 0x80484e6
> (gdb) b *main+0
> Note: breakpoints 1 and 2 also set at pc 0x80484e
> Breakpoint 3 at 0x80484e6

라고 나와있다. main함수에 breakpoint를 걸 때 b *main 으로 걸었다. 그런데 나는 b main 으로도 breakpoint를 걸어본 적이 있었기에 항상 *(애스터리스크)를 붙여야 breakpoint가 걸리는지 의문이 들었다.

```
#include<stdio.h>
int main(){
    char a[4]="Circler";
    puts(a);
    return 0;
}
```

이 코드를 gcc로 컴파일하여 gdb로 main함수를 disassemble한 뒤 두개의 명령어를 통해 breakpoint를 걸어 보겠다.

```
(gdb) disas main
Dump of assembler code for function main:
   0x00401340 <+0>:     push   %ebp
   0x00401341 <+1>:     mov    %esp,%ebp
   0x00401343 <+3>:     and    $0xfffffff0,%esp
   0x00401346 <+6>:     sub    $0x20,%esp
   0x00401349 <+9>:     call   0x401920 <__main>
   0x0040134e <+14>:    movl   $0x786573,0x1c(%esp)
   0x00401356 <+22>:    lea    0x1c(%esp),%eax
   0x0040135a <+26>:    mov    %eax,(%esp)
   0x0040135d <+29>:    call   0x401b90 <puts>
   0x00401362 <+34>:    mov    $0x0,%eax
   0x00401367 <+39>:    leave
   0x00401368 <+40>:    ret
End of assembler dump.
(gdb) b main
Breakpoint 1 at 0x40134e: file test.c, line 3.
(gdb) b *main
Breakpoint 2 at 0x401340: file test.c, line 2.
(gdb) info b
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x0040134e in main at test.c:3
2       breakpoint     keep y   0x00401340 in main at test.c:2
```

두개의 breakpoint가 서로 다른 곳에 걸린 것을 볼 수 있다.
b main 명령어는 함수의 스택프레임이 끝난 지점에서 breakpoint가 걸리게 되고 b *main 명령어는 함수의 스택프레임설정이 시작되는 지점에서 breakpoint가 걸리게 된다.

별(애스터리스크)하나로 breakpoint걸리는 지점이 다르게 변한다.
시스템보안도 웹보안만큼 재밌을 것 같다.