

Forum:[Hack Forums](#)

Thread: [x86-64 Toggle a bit](#)

theChive - November 10, 2017, 8:22 am

Anyone have any suggestions for toggling (EDIT (clarifying): switch from a 1 to a 0 or vice versa) a given bit within a number in 64 bit x86 assembly?

Something like:

Code toggle:

```
mov rax, 0x0 # zero rax
```

```
mov rax, rdi # move first argument (the number whose bit will be toggled) to rax
```

```
mov rbx, 0x0 # zero rbx
```

```
mov rbx, rsi # move second argument (the bit number within the first argument to be toggled) to rbx
```

toggle the bit?

ret # return the value after having the specified bit toggled (which is sitting in rax)

I'm not quite sure how to manipulate the number. I figured I could do a loop and jump when I am sitting on the specified bit, but I'm thinking there is an easier way to do this that I'm unaware of.

Any and all suggestions are greatly appreciated!

Gorzoid - November 10, 2017, 3:22 pm

Easiest way to toggle a bit is using bitwise XOR the value with the nth bit. Formula would be like $(x \wedge (1 \ll y))$ where x is your value, y is the bit number, $\wedge$ is bitwise xor and $\ll$ is bitwise lshift.

Haxalot - November 11, 2017, 5:39 am

As @Gorzoid mentioned, XOR'ing would be the simplest solution. I'm not quite sure why you're nullifying the registers first before moving data to them, or why you are moving the second arg from rsi to rbx, though. I'd just do the following:

Code toggle:

```
mov rax, rdi
```

```
xor rax, rsi
```

```
ret
```