W7-Logical Instructions

1.Create run.sh file
Terminal: nano run.sh

```
#!/bin/bash
nasm -f elf ./$1.asm
ld -m elf_i386 ./$1.o -o ./$1

./$1
```

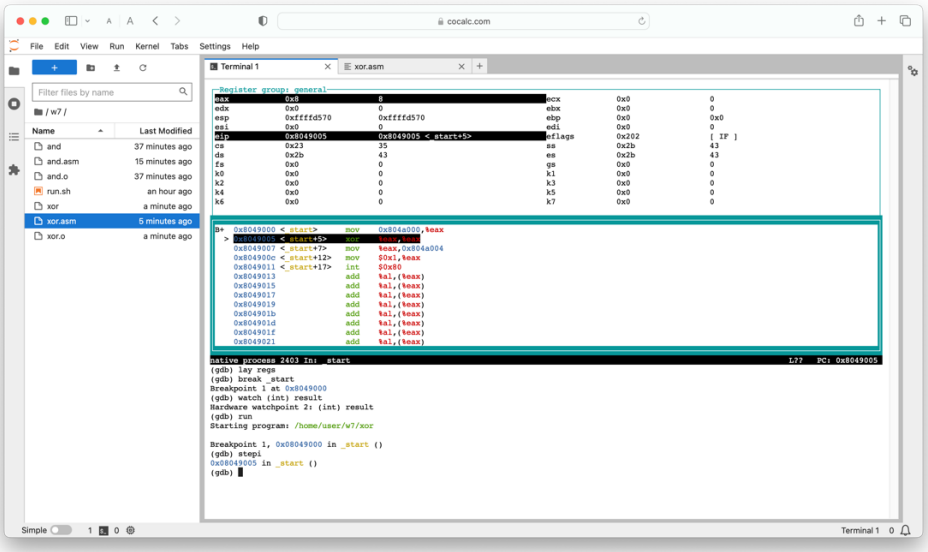
2. Change Access permission for run.sh
Terminal: chmod 777 run.sh

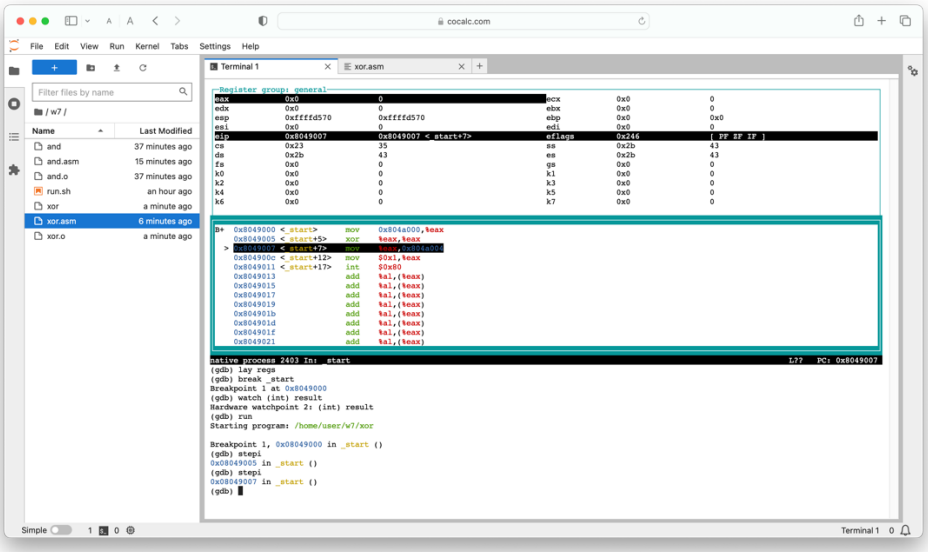
3-1. Create file in Assembly Language code to run
Terminal : nano xor.asm
result = 0

```
section .text
	global _start

_start:
	mov eax,[var]
	xor eax, eax	;XOR to clear register
	mov [result], eax

	mov eax,1
	int 0x80

section .data
	var DD 8	;assign 8 to var

segment .bss
	result resb 1	;uninitialized variable
```

3-2. Run the result code with run.sh
Terminal: ./run.sh xor

3-3. GDB debugging and checking register process
gdb xor
layout asm
layout regs
watch (int) result
break _start
run
stepi <execute step by step.>





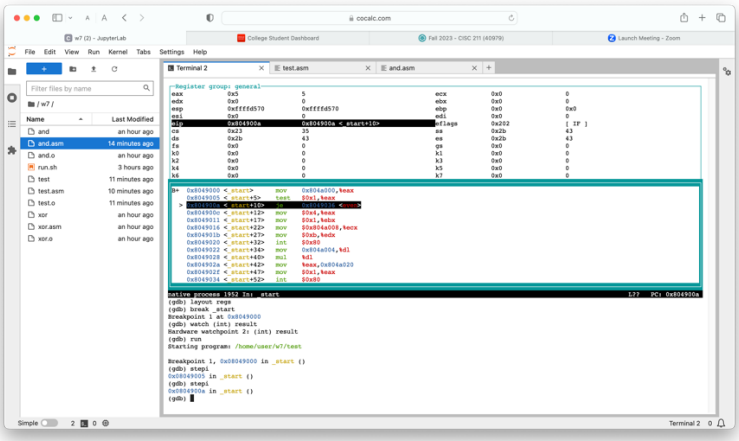
result is 0 after xoring the same number.

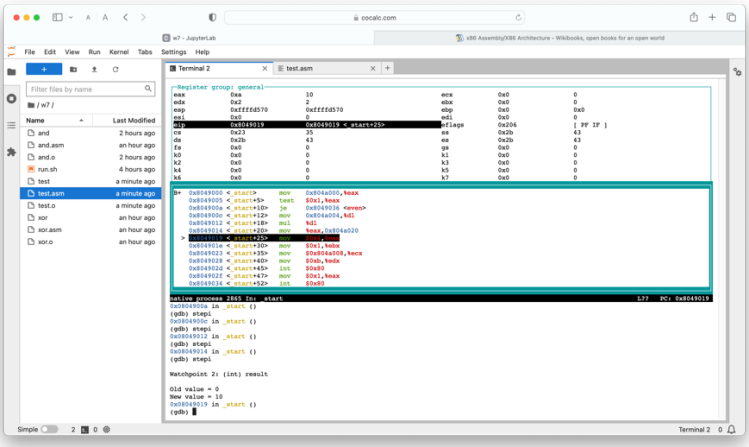
4-1. Create file in Assembly Language code to run
Terminal : nano test.asm

```
section .text
	global _start

_start:
	mov eax,[var]
	test eax,1
	jz even

	mov dl,[var2]		;if odd, save <result = var * 2> and display "Odd Number"
	mul dl
```

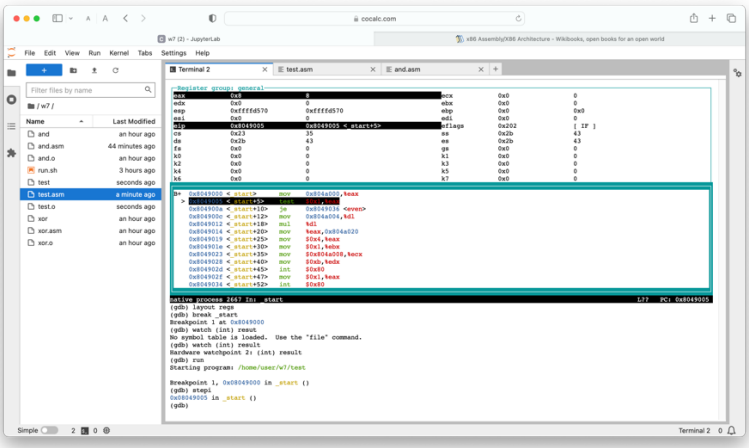
```
        mov [result], eax

        mov eax,4
        mov ebx,1
        mov ecx,msg1
        mov edx,len1
        int 0x80

        mov eax,1
        int 0x80

even:                   ;if even, save <result = var/ 2> and display "Even Number"

        mov bl,[var2]
        div bl
        mov [result], eax

        mov eax,4
        mov ebx,1
        mov ecx,msg2
        mov edx,len2
        int 0x80

        mov eax,1
        int 0x80

section .data
        var DD 8        ;assign value to var
        var2 DD 2
        msg1 db 'Odd Number', 0xa
        len1 equ $ - msg1
        msg2 db 'Even Number', 0xa
        len2 equ $ - msg2


segment .bss
        result resb 1       ;uninitialized variable
```

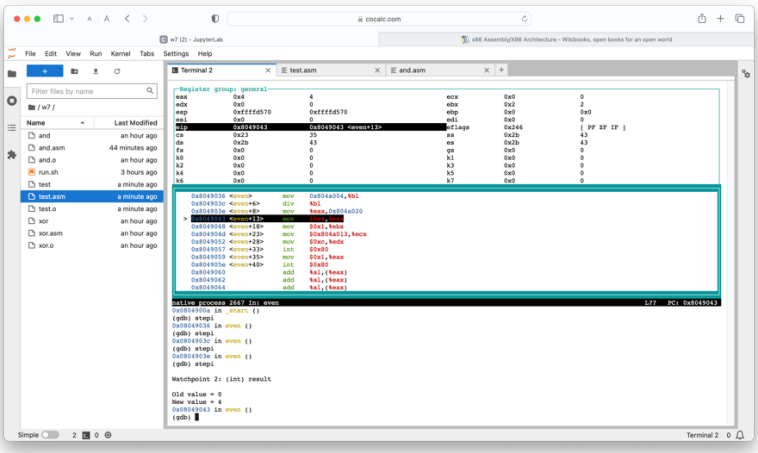
4-2. Run the result code with run.sh
Terminal: ./run.sh test
4-3. GDB debugging and checking register process
gdb test
layout asm
layout regs
watch (int) result
break _start
run
stepi <execute step by step.>





If odd (var = 5), then multiple 2, save to result (10), and print “Odd number” on console.





If even (var = 8), then divide by 2, save to result (4), and print “Even number” on console.

***Challenge:

1. For easy displaying the results, I tried to print result along with message, but not sure how to concatenate variable/number with strings.
2. After the TEST logical operation, I was about to print msg ahead of calculation, and then saving the result. Tried to put “mov eax,4….” before the multiplication and division. Since msg and calculation both using the same eax register, eax value overwritten for printing message. If do calculation after message output, the calculated result is not correct. Using different register should work, but not sure which register (eax, ebx, ecx, edx all used) should I use. Thus, here I just changed the order of calculation and print out message.