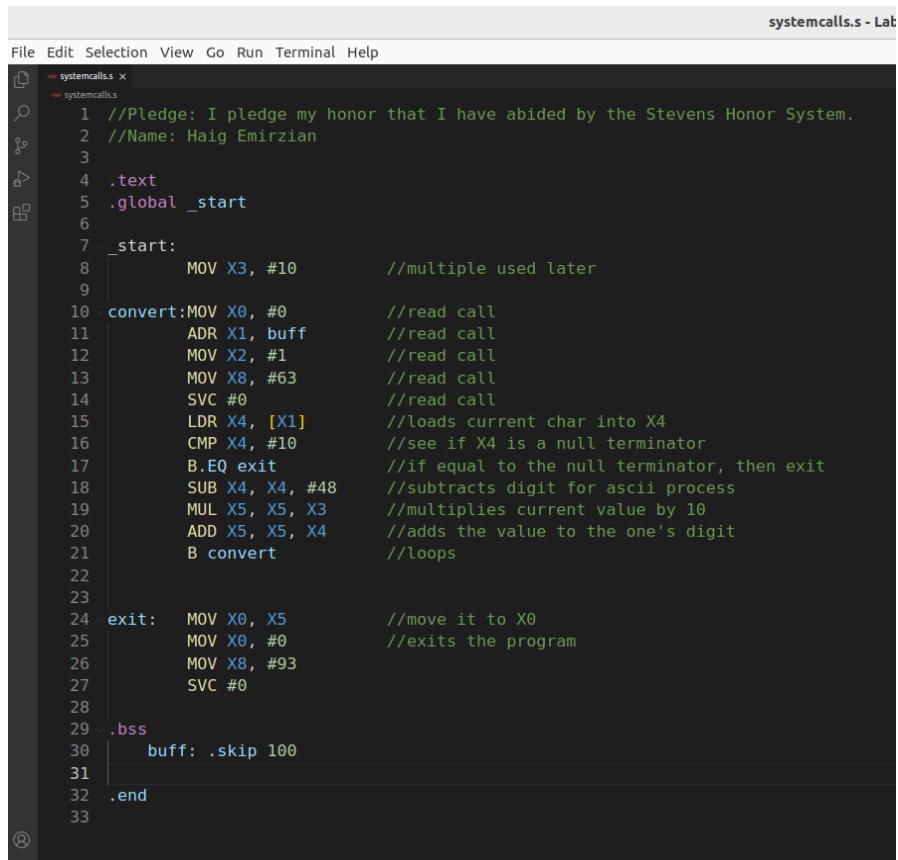
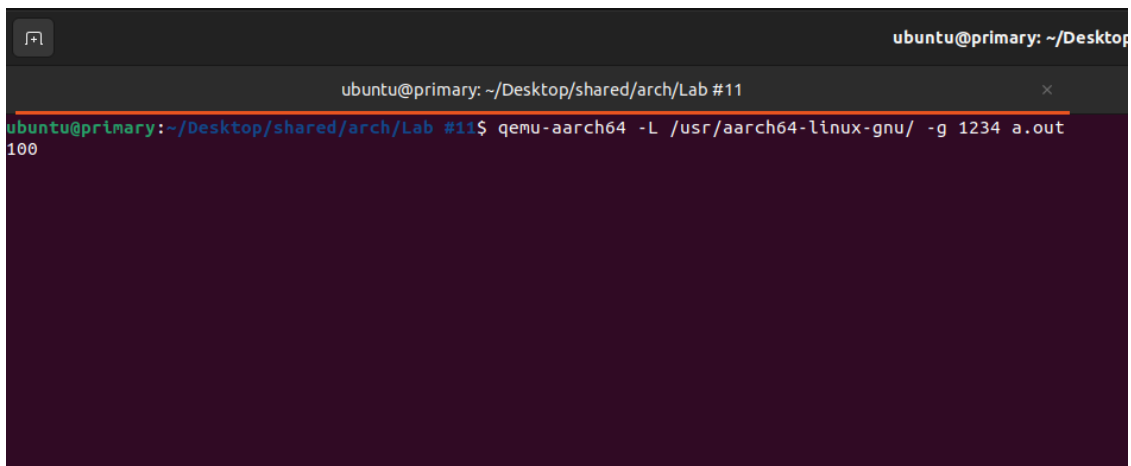


Pledge: I pledge my honor that I have abided by the Stevens Honor System.
Name: Haig Emirzian



```
1 //Pledge: I pledge my honor that I have abided by the Stevens Honor System.
2 //Name: Haig Emirzian
3
4 .text
5 .global _start
6
7 _start:
8     MOV X3, #10        //multiple used later
9
10 convert: MOV X0, #0     //read call
11         ADR X1, buff    //read call
12         MOV X2, #1      //read call
13         MOV X8, #63     //read call
14         SVC #0          //read call
15         LDR X4, [X1]     //loads current char into X4
16         CMP X4, #10     //see if X4 is a null terminator
17         B.EQ exit       //if equal to the null terminator, then exit
18         SUB X4, X4, #48  //subtracts digit for ascii process
19         MUL X5, X5, X3   //multiplies current value by 10
20         ADD X5, X5, X4   //adds the value to the one's digit
21         B convert       //loops
22
23
24 exit:   MOV X0, X5       //move it to X0
25         MOV X0, #0      //exits the program
26         MOV X8, #93
27         SVC #0
28
29 .bss
30     buff: .skip 100
31
32 .end
33
```

Screenshot of program



```
ubuntu@primary: ~/Desktop
ubuntu@primary: ~/Desktop/shared/arch/Lab #11
ubuntu@primary:~/Desktop/shared/arch/Lab #11$ qemu-aarch64 -L /usr/aarch64-linux-gnu/ -g 1234 a.out
100
```

Input string is 100
I put in 100 after running it

```
ubuntu@primary: ~/Desktop/shared/arch/Lab #  
Register group: general  
x0      0x64      100  
x3      0xa       10  
x6      0x0       0  
x9      0x0       0  
x12     0x0       0  
x15     0x0       0  
x18     0x0       0  
x21     0x0       0  
x24     0x0       0  
x27     0x0       0  
x30     0x0       0  
cpsr    0x60000000 1610612736  
systemcalls.s  
18      SUB X4, X4, #48 //subtracts digit for ascii  
19      MUL X5, X5, X3 //multiplies current value b  
20      ADD X5, X5, X4 //adds the value to the one  
21      B convert //loops  
22  
23  
B+ 24      exit: MOV X0, X5 //move it to X0  
    > 25      MOV X0, #0 //exits the program  
26      MOV X8, #93  
27      SVC #0  
28  
29      .bss  
30      buff: .skip 100  
31      dest: .ascii ""  
remote Thread 1.3079 In: exit  
(gdb) b _start  
Breakpoint 1 at 0x4000b0: file systemcalls.s, line 8.  
(gdb) b exit  
Breakpoint 2 at 0x4000e4: file systemcalls.s, line 24.  
(gdb) c  
Continuing.  
Breakpoint 2, exit () at systemcalls.s:24  
=> 0x00000000004000e4 <exit+0>: e0 03 05 aa    mov    x0, x5  
(gdb) s  
=> 0x00000000004000e8 <exit+4>: 00 00 80 d2    mov    x0, #0x0  
(gdb)
```

X0 is 100

- I created two breakpoints: one at `_start` and one at `exit`
- I used the `continue` command to go to the second breakpoint
- I then used the `step` command to see what X0 is after I move the conversion into X0