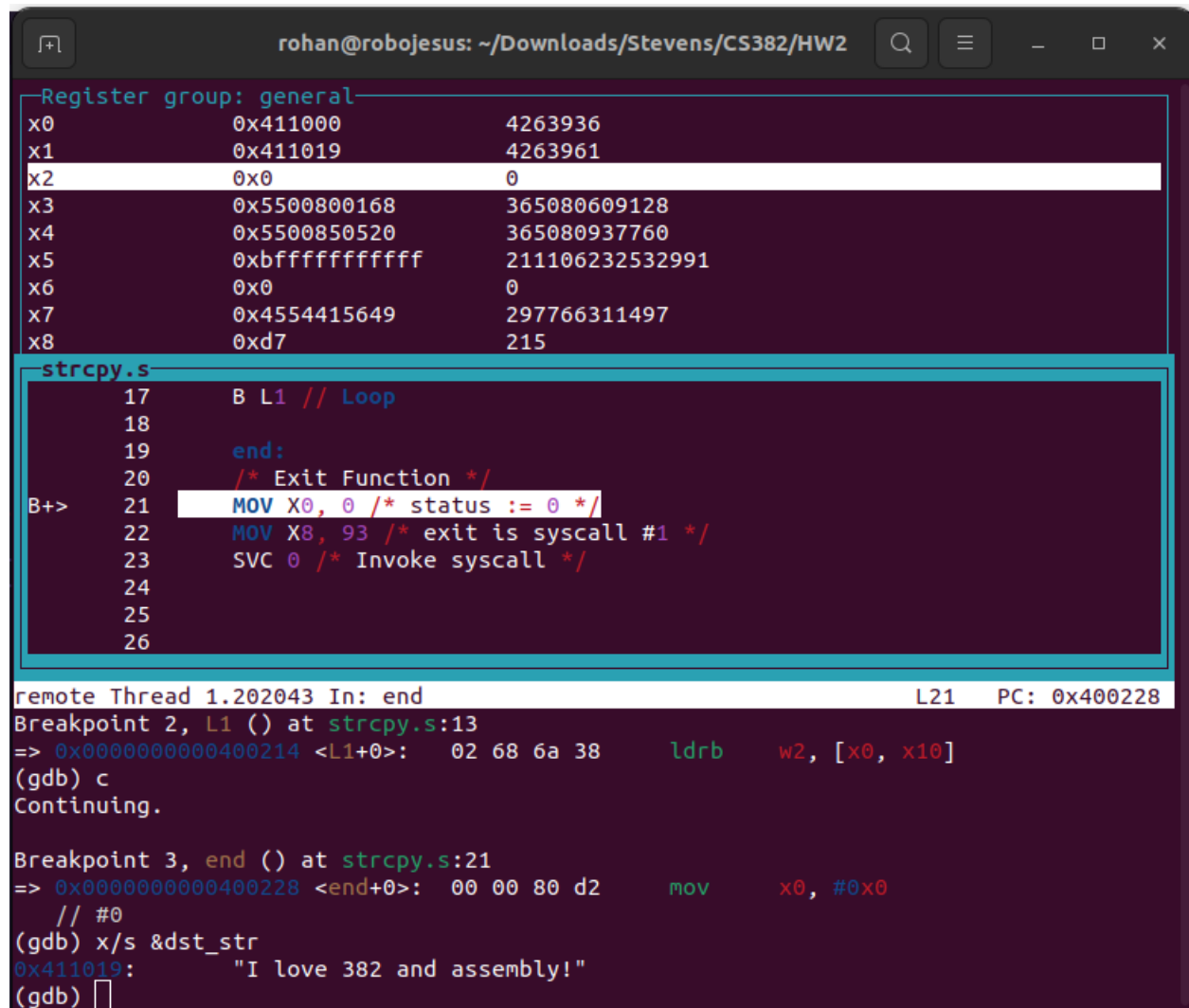


Rohan Kallur

I pledge my honor that I have abided by the Stevens Honor System

Task 1



The screenshot shows a GDB terminal window with the title bar "rohan@robojesus: ~/Downloads/Stevens/CS382/HW2". The window is divided into three main sections. The top section, titled "Register group: general", displays the values of registers x0 through x8. The middle section shows the assembly code for the function "strcpy.s", with line 21 highlighted. The bottom section shows the GDB prompt and the output of the "x/s &dst_str" command.

Register	Address	Value
x0	0x411000	4263936
x1	0x411019	4263961
x2	0x0	0
x3	0x5500800168	365080609128
x4	0x5500850520	365080937760
x5	0xbfffffffffff	211106232532991
x6	0x0	0
x7	0x4554415649	297766311497
x8	0xd7	215

```
strcpy.s
17      B L1 // Loop
18
19      end:
20      /* Exit Function */
21      MOV X0, 0 /* status := 0 */
22      MOV X8, 93 /* exit is syscall #1 */
23      SVC 0 /* Invoke syscall */
24
25
26

remote Thread 1.202043 In: end                                L21    PC: 0x400228
Breakpoint 2, L1 () at strcpy.s:13
=> 0x000000000000400214 <L1+0>:  02 68 6a 38      ldrb      w2, [x0, x10]
(gdb) c
Continuing.

Breakpoint 3, end () at strcpy.s:21
=> 0x000000000000400228 <end+0>:  00 00 80 d2      mov      x0, #0x0
// #0
(gdb) x/s &dst_str
0x411019:  "I love 382 and assembly!"
(gdb)
```

My source string is in X0, Destination str is in X1, and the character whose index im at is in X2
As you can see in the gdb calling x/s on the &dst_str which is its address gives us the string "I love 382 and assembly!" and the character that my currently at register is showing is 0 which indicates the null terminator at the end of the string

Task 2

```
rohan@robojesus: ~/Downloads/Stevens/CS382/HW2

Register group: general
x0      0x411000      4263936
x1      0x41100c      4263948
x2      0x6           6
x3      0x5           5
x4      0x1           1
x5      0xdc          220
x6      0x98          152
x7      0xc           12
x8      0xd0          208
x9      0x8           8

reverse.s
37
38
39 end:
40 /* Exit Function */
B+> 41 MOV X0, 0 /* status := 0 */
42 MOV X8, 93 /* exit is syscall #1 */
43 SVC 0 /* Invoke syscall */
44
45 .data
46 arr: .word 0x12BFDA09, 0x9089CDBA, 0x56788910
47 length: .word 3

remote Thread 1.15606 In: end L41 PC: 0x400264

Breakpoint 2, loop () at reverse.s:16
=> 0x0000000000400220 <loop+0>: 05 68 62 38 ldrb w5, [x0, x2]
(gdb) c
Continuing.

Breakpoint 3, end () at reverse.s:41
=> 0x0000000000400264 <end+0>: 00 00 80 d2 mov x0, #0x0 /
/ #0
(gdb) x/3xw 0x411000
0x411000: 0x01988765 0xabdc9809 0x90adfb21
(gdb) █
```

In this my gdb is at the end of the loop, my x2 which is the index of the front of the array is greater than my x3 which is the value of the back of my array which signifies the end of my loop, if I then print out the value stored in my x0 (arr) it prints out the three elements with reversed nibbles.

Task 3

```
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ aarch64-linux-gnu-as bins.s -o bins.o
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ aarch64-linux-gnu-ld bins.o -lc
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ qemu-aarch64 -L /usr/aarch64-linux-gnu/ a.out
Target -25 is in the array.
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ aarch64-linux-gnu-as bins.s -o bins.o
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ aarch64-linux-gnu-ld bins.o -lc
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$ qemu-aarch64 -L /usr/aarch64-linux-gnu/ a.out
Target 20 is not in the array.
rohan@robojesus:~/Downloads/Stevens/CS382/HW2$
```

Prints -25 is in the array and 20 is not