

CPSC 240: Computer Organization and Assembly Language

Assignment 05, Spring Semester 2023

CWID: 886880327 Name: Andrew Saldana

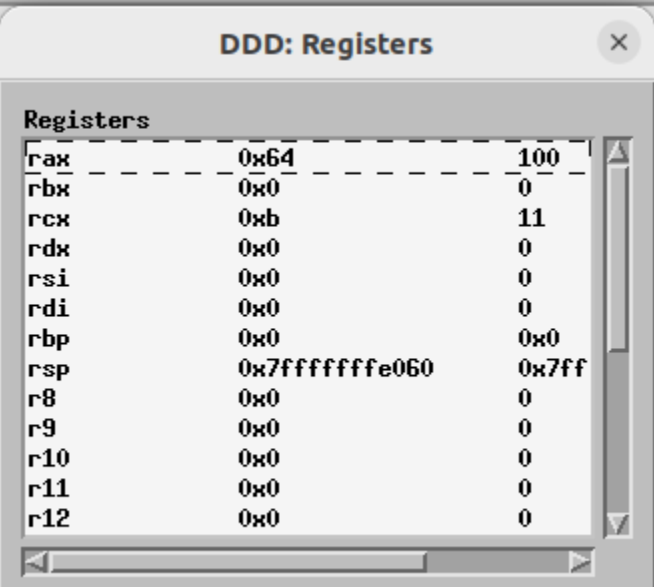
1. Download the "CPSC-240 Assignment05.docx" document.
2. Design the "doWhile.asm" program, and use assembly language to realize the function of the following C++ instructions.
register char cl = 1;
unsigned short sum = 0;
do {
 sum = sum + (cl * cl);
 cl++;
} while(cl <= 10);
3. Assemble the "doWhile.asm" file and link the "doWhile.o" file to get the "doWhile" executable file.
4. Run the "doWhile" file with the DDD debugger to display the simulation results of **sum**.
5. Insert source code (doWhile.asm) and simulation results (GDB window) of the memory (**sum**) in the document. Write an analysis to verify simulation results.
6. Save the file in pdf or docx format and submit the pdf or docx file to Canvas before 23:59 pm on 03/15/2023.

[Insert the source of doWhile.asm here]

```
1 ; register char cl = 1;
2 ; unsigned short sum = 0;
3 ; do {
4 ;     sum = sum + (cl * cl);
5 ;     cl++;
6 ; } while(cl <= 10);
7
8 section .data
9 sum     dw     0
10
11 section .text
12     global _start
13
14 _start:
15     mov     cl, 1                ; cl = 0001h only runs once
16 doloop
17     mov     al, cl                ; al = cl
18     mul     cl                    ; al = cl * al = cl * cl
19
20     ;mov from ax bc sum is a word and we need to use a word size register
21     ;has same size as al
22     add     word[sum], ax         ;sum = sum + ax = sum + (cl * cl)
23     inc     cl                    ;cl++
24
25     cmp     cl, 10                ;compare cl with 0
26     jbe     doloop                ;jump to doloop if cl is less than or equal to 10
27
28 exit:
29     mov     rax, 60
30     mov     rdi, 0
31     syscall
```

[Insert doWhile simulation result (GDB window with sum) here]

```
(gdb) x/xw &sum
0x402000: 0x00000181
(gdb) x/uw &sum
0x402000: 385
(gdb) I
```



Registers

Register	Value	Comment
rax	0x64	100
rbx	0x0	0
rcx	0xb	11
rdx	0x0	0
rsi	0x0	0
rdi	0x0	0
rbp	0x0	0x0
rsp	0x7fffffff060	0x7ff
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x0	0
r12	0x0	0

Integer registers All registers

[Insert doWhile simulation result analysis here]

		<i>mul cl</i>	<i>add word[sum],al</i>	<i>inc cl</i>	<i>cmp cl,10</i>	<i>jbe doloop</i>
cl	al	al * cl	Sum + al	cl + 1	cl <= 10?	jump?
1	1	1	0 + 1 = 1	1 + 1 = 2	T	jbe doloop
2	2	4	1 + 4 = 5	2 + 1 = 3	T	jbe doloop
3	3	9	5 + 9 = 14	3 + 1 = 4	T	jbe doloop
4	4	16	14 + 16 = 30	4 + 1 = 5	T	jbe doloop
5	5	25	30 + 25 = 55	5 + 1 = 6	T	jbe doloop
6	6	36	55 + 36 = 91	6 + 1 = 7	T	jbe doloop
7	7	49	91 + 49 = 140	7 + 1 = 8	T	jbe doloop
8	8	64	140 + 64 = 204	8 + 1 = 9	T	jbe doloop
9	9	81	204 + 81 = 285	9 + 1 = 10	T	jbe doloop
10	10	100	285 + 100 = 385	10 + 1 = 11	F	finish

Sum = 385