

Laboratory 1: Prelab

Date: 09/13/2020 Section: CS402

Name: Shansi Dong

The last digit of my SSN is 4.

Test cases for *lab1.1.asm*

Input number	Output number
-4	-64
-3	-48
-2	-32
-1	-16
0	0
1	16
2	32
3	48
4	64
5	80

Step 7

What is the formula that describes the relation between the output and the input?

Output=input*16

Laboratory 1: Inlab

Date: 09/13/2020

Section: CS402

Name: Shansi Dong

Step 2

Symbol	Address
g__eoth	0x00400024
g__start	0x00400000
g main	0x00400024

Step 4

Symbol	Address
g__eoth	0x00400024
g__start	0x00400000
g main	0x00400024

Q 1:

What is the size of an instruction (in bytes)?

Instruction size =4

Step 6

Label	Address	Native instruction	Source instruction
Main	[00400000]	lw \$4, 0(\$29)	: lw \$a0 0(\$sp) # argc
Main	[00400004]	addiu \$5, \$29, 4	: addiu \$a1 \$sp 4 # argv
Main	[00400008]	addiu \$6, \$5, 4	: addiu \$a2 \$a1 4 # envp
Main	[0040000c]	sll \$2, \$4, 2	: sll \$v0 \$a0 2
Main	[00400010]	addu \$6, \$6, \$2	: addu \$a2 \$a2 \$v0
Main	[00400014]	Jal 0x00400024 [main]	: jal main
Main	[00400018]	nop	: nop
Main	[0040001c]	ori \$2, \$0, 10	: li \$v0 10
Main	[00400020]	syscall	: syscall # syscall 10 (exit)
Main	[00400024]	addu \$16, \$31, \$0	addu \$s0, \$ra, \$0 # save \$31 in \$16
Label1	[00400028]	ori \$2, \$0, 4	li \$v0, 4 # system call for print_str
Main	[0040002c]	lui \$4, 4096 [msg1]	la \$a0, msg1 # address of string to pri
Main	[00400030]	syscall	syscall
Main	[00400034]	ori \$2, \$0, 5	li \$v0, 5 # system call for read_int
Main	[00400038]	syscall	syscall # the integer placed in \$v0
Main	[0040003c]	addu \$8, \$2, \$0	addu \$t0, \$v0, \$0 # move the number in
Main	[00400040]	sll \$8, \$8, 4	sll \$t0, \$t0, 4 # last digit of your SS
Main	[00400044]	ori \$2, \$0, 1	li \$v0, 1 # system call for print_int
Main	[00400048]	addu \$4, \$8, \$0	addu \$a0, \$t0, \$0 # move number to prin
Main	[0040004c]	syscall	syscall
Main	[00400050]	addu \$31, \$0, \$16	addu \$ra, \$0, \$s0 # return address back
Main	[00400054]	jr \$31	jr \$ra # return from main

Step 7

Label	Address (PC)	Native instruction	Source instruction
Main	[00400000]	lw \$4, 0(\$29)	: lw \$a0 0(\$sp) # argc
Main	[00400004]	addiu \$5, \$29, 4	: addiu \$a1 \$sp 4 # argv
Main	[00400008]	addiu \$6, \$5, 4	: addiu \$a2 \$a1 4 # envp
Main	[0040000c]	sll \$2, \$4, 2	: sll \$v0 \$a0 2

Main	[00400010]	addu \$6, \$6, \$2	: addu \$a2 \$a2 \$v0
Main	[00400014]	Jal 0x00400024 [main]	: jal main
Main	[00400018]	nop	: nop
Main	[0040001c]	ori \$2, \$0, 10	: li \$v0 10
Main	[00400020]	syscall	: syscall # syscall 10 (exit)
Main	[00400024]	addu \$16, \$31, \$0	addu \$s0, \$ra, \$0 # save \$31 in \$16
Label1	[00400028]	ori \$2, \$0, 4	li \$v0, 4 # system call for print_str
Main	[0040002c]	lui \$4, 4096 [msg1]	la \$a0, msg1 # address of string to pri
Main	[00400030]	syscall	syscall
Main	[00400034]	ori \$2, \$0, 5	li \$v0, 5 # system call for read_int
Main	[00400038]	syscall	syscall # the integer placed in \$v0
Main	[0040003c]	addu \$8, \$2, \$0	addu \$t0, \$v0, \$0 # move the number in
Main	[00400040]	sll \$8, \$8, 4	sll \$t0, \$t0, 4 # last digit of your SS
Main	[00400044]	ori \$2, \$0, 1	li \$v0, 1 # system call for print_int
Main	[00400048]	addu \$4, \$8, \$0	addu \$a0, \$t0, \$0 # move number to prin
Main	[0040004c]	syscall	syscall
Main	[00400050]	addu \$31, \$0, \$16	addu \$ra, \$0, \$s0 # return address back
Main	[00400054]	jr \$31	jr \$ra # return from main

Q 2:

Why does this table differ from the table you got at step 6?

Because for step 6, it starts from the address of the start_ symbol, but for step7, it's step by step.

Step 9

Register number	Register name	Before the syscall	After the syscall	Changed
0	zero	0	0	0
1	\$at	0	0	0
2	\$v0	0	4	4*
3	\$v1	0	0	0
4	\$a0	1	10000000	9999999*
5	\$a1	7ffffe14	7ffffe14	0
6	\$a2	7ffffe1c	7ffffe1c	0
7	\$a3	0	0	0
8	\$t0	0	0	0
9	\$t1	0	0	0
10	\$t2	0	0	0
11	\$t3	0	0	0
12	\$t4	0	0	0

13	\$t5	0	0	0
14	\$t6	0	0	0
15	\$t7	0	0	0
16	\$s0	0	400018	400018*
17	\$s1	0	0	0
18	\$s2	0	0	0
19	\$s3	0	0	0
20	\$s4	0	0	0
21	\$s5	0	0	0
22	\$s6	0	0	0
23	\$s7	0	0	0
24	\$t8	0	0	0
25	\$t9	0	0	0
26	\$k0	0	0	0
27	\$k1	0	0	0
28	\$gp	10008000	10008000	0
29	\$sp	7ffffe10	7ffffe10	0
30	\$fp	0	0	0
31	\$ra	0	400018	400018*

Q 3:

Some registers have changed during the syscall execution. Can you assume that syscall uses only these registers? Explain.

Yes. Because during the calculation, only these registers have been changed(their color turned red).

Q 4:

The first instruction in your program moves the content of register **\$ra** to register **\$s0**. The content of that register is a memory address. What is stored in memory at that address?

400018

Q 5:

This question is related to the previous one. When will be executed the instruction stored at the address in **\$s0**? Indicate the instruction that immediately precedes it in execution.

nop

Laboratory 1: Postlab

Date: 09/16/2020

Section: CS402

Name: Shansi Dong

Step 2

Register	Single precision	Double precision
\$f0	10000001000011001110110 11001001	10000000001000011001110110110010001011010000111 0010101100000010
\$f2	0	0
\$f4	0	0
\$f6	0	0
\$f8	0	0
\$f10	0	0
\$f12	10000001000011001110110 11001001	10000000001000011001110110110010001011010000111 0010101100000010
\$f14	0	0
\$f16	0	0
\$f18	0	0
\$f20	0	0
\$f22	0	0
\$f24	0	0
\$f26	0	0
\$f28	0	0
\$f30	0	0

For codes lab1.1-4.asm, see:

<https://dls0122@bitbucket.org/dls0122/lab1.git>

Email: sdong14@hawk.iit.edu