

LAB Report On MIPS Assembly Programming (Basic) (Individual)

Lab Part 1 – Arithmetic

Register Panel

Before Program Execution

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304
hi		0
lo		0

After Program Execution

Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0	
\$at	1	268500992	
\$v0	2	10	
\$v1	3	0	
\$a0	4	0	
\$a1	5	0	
\$a2	6	0	
\$a3	7	0	
\$t0	8	15	
\$t1	9	10	
\$t2	10	5	
\$t3	11	2	
\$t4	12	18	
\$t5	13	-3	
\$t6	14	5	
\$t7	15	7	
\$s0	16	53	
\$s1	17	0	
\$s2	18	0	
\$s3	19	0	
\$s4	20	0	
\$s5	21	0	
\$s6	22	0	
\$s7	23	0	
\$t8	24	21	
\$t9	25	3	
\$k0	26	0	
\$k1	27	0	
\$gp	28	268468224	
\$sp	29	2147479548	
\$fp	30	0	
\$ra	31	0	
pc		4194384	
hi		0	
lo		35	

Memory Panel

Before Program Execution

Text Segment

Program Arguments:

Bkpt	Address	Source	Code	Basic
	4194304:27:	li \$t0, 15	#\$t0 = A = 15	0x2408000f addiu \$8,\$0,15
	4194308:28:	li \$t1, 10	#\$t1 = B = 10	0x2409000a addiu \$9,\$0,10
	4194312:29:	li \$t2, 5	#\$t2 = C = 5	0x240a0005 addiu \$10,\$0,5
	4194316:30:	li \$t3, 2	#\$t3 = D = 2	0x240b0002 addiu \$11,\$0,2
	4194320:31:	li \$t4, 18	#\$t4 = E = 18	0x240c0012 addiu \$12,\$0,18
	4194324:32:	li \$t5, -3	#\$t5 = F = -3	0x240dffff addiu \$13,\$0,-3
	4194328:34:	sub \$t6, \$t0, \$t1	\$(A - B)	0x01097022 sub \$14,\$8,\$9
	4194332:35:	add \$t7, \$t2, \$t3	\$(C + D)	0x014b7820 add \$15,\$10,\$11
	4194336:36:	sub \$t8, \$t4, \$t5	\$(E - F)	0x018dc022 sub \$24,\$12,\$13
	4194340:37:	div \$t9, \$t0, \$t2	\$(A / C)	0x15400001 bne \$10,\$0,1
	4194344:			0x0000000d break
	4194348:			0x010a001a div \$8,\$10
	4194352:			0x00000c812 mflo \$25
	4194356:39:	mul \$a0, \$t6, \$t7	\$(A - B) * (C + D)	0x71cf8002 mul \$16,\$14,\$15
	4194360:40:	add \$a0, \$a0, \$t8	\$(A - B) * (C + D) + (E - F)	0x02188020 add \$16,\$16,\$24

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	0	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

After Program Execution

Text Segment

Program Arguments:

Bkpt	Address	Source	Code	Basic
	4194304:27:	li \$t0, 15	#\$t0 = A = 15	0x2408000f addiu \$8,\$0,15
	4194308:28:	li \$t1, 10	#\$t1 = B = 10	0x2409000a addiu \$9,\$0,10
	4194312:29:	li \$t2, 5	#\$t2 = C = 5	0x240a0005 addiu \$10,\$0,5
	4194316:30:	li \$t3, 2	#\$t3 = D = 2	0x240b0002 addiu \$11,\$0,2
	4194320:31:	li \$t4, 18	#\$t4 = E = 18	0x240c0012 addiu \$12,\$0,18
	4194324:32:	li \$t5, -3	#\$t5 = F = -3	0x240dffff addiu \$13,\$0,-3
	4194328:34:	sub \$t6, \$t0, \$t1	\$(A - B)	0x01097022 sub \$14,\$8,\$9
	4194332:35:	add \$t7, \$t2, \$t3	\$(C + D)	0x014b7820 add \$15,\$10,\$11
	4194336:36:	sub \$t8, \$t4, \$t5	\$(E - F)	0x018dc022 sub \$24,\$12,\$13
	4194340:37:	div \$t9, \$t0, \$t2	\$(A / C)	0x15400001 bne \$10,\$0,1
	4194344:			0x0000000d break
	4194348:			0x010a001a div \$8,\$10
	4194352:			0x00000c812 mflo \$25
	4194356:39:	mul \$a0, \$t6, \$t7	\$(A - B) * (C + D)	0x71cf8002 mul \$16,\$14,\$15
	4194360:40:	add \$a0, \$a0, \$t8	\$(A - B) * (C + D) + (E - F)	0x02188020 add \$16,\$16,\$24

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	53	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Lab Part 2 – Branches

Register Panel

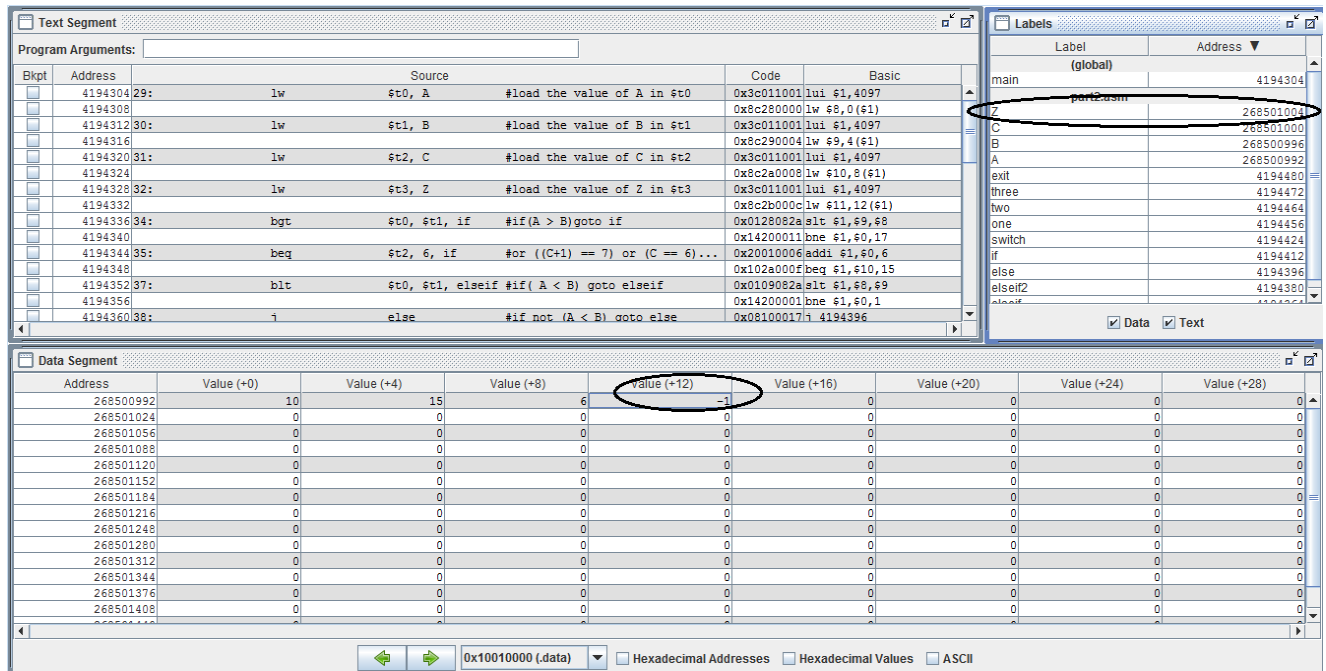
Before Program Execution

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304
hi		0
lo		0

After Program Execution

Registers			Coproc 1	Coproc 0
Name	Number	Value		
\$zero	0	0		
\$at	1	268500992		
\$v0	2	10		
\$v1	3	0		
\$a0	4	0		
\$a1	5	0		
\$a2	6	0		
\$a3	7	0		
\$t0	8	10		
\$t1	9	15		
\$t2	10	6		
\$t3	11	-1		
\$t4	12	0		
\$t5	13	0		
\$t6	14	0		
\$t7	15	0		
\$s0	16	0		
\$s1	17	0		
\$s2	18	0		
\$s3	19	0		
\$s4	20	0		
\$s5	21	0		
\$s6	22	0		
\$s7	23	0		
\$t8	24	0		
\$t9	25	0		
\$k0	26	0		
\$k1	27	0		
\$gp	28	268468224		
\$sp	29	2147479548		
\$fp	30	0		
\$ra	31	0		
pc		4194496		
hi		0		
lo		0		

Before Program Execution



Lab Part 3 – Loops

Register Panel

After Program Execution

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	1
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	76
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194440
hi		0
lo		0

Memory Panel

After Program Execution

Text Segment

Program Arguments:

Bkpt	Address	Source	Code	Basic
	4194304 27:	lw \$t0, Z # load the value of Z in \$t0	0x3c011001 lui \$1,4097	
	4194308		0x8c280000 lw \$8,0(\$1)	
	4194312 28:	lw \$t1, i #load the value of i in \$t1	0x3c011001 lui \$1,4097	
	4194316		0x8c290004 lw \$9,4(\$1)	
	4194320 30:	li \$t1, 0 #initialize \$t1 with 0	0x24090000 addiu \$9,\$0,0	
	4194324 31:	sw \$t1, i #store the value in \$t1 ...	0x3c011001 lui \$1,4097	
	4194328		0xac290004 sw \$9,4(\$1)	
	4194332 34:	bgt \$t1, 21, do #loop till \$t1 <= 21 and then...	0x20010015 addi \$1,\$0,21	
	4194336		0x0029082a slt \$1,\$1,\$9	
	4194340		0x14200007 bne \$1,\$0,7	
	4194344 35:	addi \$t0, \$t0, 1 #increment \$t0 by 1	0x21080001 addi \$8,\$8,1	
	4194348 36:	sw \$t0, Z #store the value of \$t0 in Z	0x3c011001 lui \$1,4097	
	4194352		0xac280000 sw \$8,0(\$1)	
	4194356 38:	addi \$t1, \$t1, 3 #increment the value of \$t1 by 3	0x21290003 addi \$9,\$9,3	
	4194360 39:	sw \$t1, i #store the value of \$t1 in i	0x3c011001 lui \$1,4097	

Labels

Label	Address
(global)	
main	4194304
part3.asm	
i	268500996
Z	268500992
exit	4194432
while	4194392
do	4194372
for	4194332

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	76	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Lab Part 4 – Arrays

Register Panel

After Program Execution

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	4
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	-4
\$t1	9	2
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194428
hi		0
lo		0

Memory Panel

After Program Execution

Memory Panel interface showing program execution results.

Text Segment

Program Arguments:

Bkpt	Address	Source	Code	Basic
	4194304:27:	li \$t0, 0 # i = \$t0 and initialize it w...	0x24080000	addiu \$8,\$0,0
	4194308:30:	beq \$t0, 20, end_for #if \$t0 = 5 * 4 (size of word...	0x20010014	addi \$1,\$0,20
	4194312:		0x1028000a	beq \$1,\$8,10
	4194316:31:	lw \$t1, B(\$t0) #load a certain word from the...	0x3c011001	lui \$1,4097
	4194320:		0x00280821	addu \$1,\$1,\$8
	4194324:		0x8c290014	lw \$9,20(\$1)
	4194328:32:	sub \$t1, \$t1, 1 #decrement the value by 1	0x20010001	addi \$1,\$0,1
	4194332:		0x01214822	sub \$9,\$9,\$1
	4194336:33:	sw \$t1, A(\$t0) #store the value in a certain...	0x3c011001	lui \$1,4097
	4194340:		0x00280821	addu \$1,\$1,\$8
	4194344:		0x8c290000	sw \$9,0(\$1)
	4194348:34:	add \$t0, \$t0, 4 #i++, as the arrays are of wo...	0x21080004	addi \$8,\$8,4
	4194352:35:	j for #continue the loop	0x08100001	j 4194308
	4194356:40:	add \$t0, \$t0, -4 #decrement \$t0 by 4 and so \$t...	0x2108ffff	addi \$8,\$8,-4
	4194360:43:	bltz \$t0, exit #if i = 0 or \$t0 = 0. exit th...	0x0500000e	bltz \$8,14

Labels

Label	Address
(global)	
main	4194304
part4.asm	
B	268501012
A	268500992
exit	4194420
while	4194360
end_for	4194356
for	4194308

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	2	6	14	30	62	1	2	4
268501024	8	16	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Lab Part 5 – I/O, Loops and Arrays

Memory Panel

Address of result and its value (address of the character 'm')

The screenshot displays the Memory Panel of a debugger, divided into two main sections: Text Segment and Data Segment.

Text Segment: This section shows assembly code with columns for Bkpt, Address, Source, Code, and Basic. The code is for a program that takes input as a string and prints it. The address 4194336 is highlighted in the Source column, corresponding to the instruction `addiu $s0, $s0, 1`.

Labels: A list of labels is shown on the right, including `main`, `part5.asm`, `result`, `msg2`, `msg1`, `newline`, `address2`, `address`, `str`, `printString`, `while`, `hexValue`, and `digit`. The label `result` is circled in red, with its address 268501364.

Data Segment: This section shows memory data with columns for Address, Value (+0), Value (+4), Value (+8), Value (+12), Value (+16), Value (+20), Value (+24), and Value (+28). The address 268500994 is circled in red, with its value 10.

Legend: At the bottom, there are checkboxes for `Hexadecimal Addresses`, `Hexadecimal Values`, and `ASCII`.

Location of the character 'm'

Edit Execute

Text Segment

Program Arguments:

Bkpt	Address	Source	Code	Basic
	4194304	34: li \$v0, 8 #Take input as string ...	0x24020008	addiu \$2,\$0,8
	4194308	35: la \$a0, str #load the address of str	0x3c011001	lui \$1,4097
	4194312		0x34240000	ori \$4,\$1,0
	4194316	36: li \$a1, 255 #put the size of str	0x240500ff	addiu \$5,\$0,255
	4194320	37: syscall	0x0000000c	syscall
	4194324	42: li \$t0, 0 #t0 = i, initialize i	0x24080000	addiu \$8,\$0,0
	4194328	43: la \$s0, str	0x3c011001	lui \$1,4097
	4194332		0x34300000	ori \$16,\$1,0
	4194336	46: lb \$t2, (\$s0) #load a character from str ...	0x820a0000	lb \$10,0(\$16)
	4194340	47: beq \$t2, 0, notFound #if \$t2 == 0 or '\0' goto n...	0x20010000	addi \$1,\$0,0
	4194344		0x102a0014	beq \$1,\$10,20
	4194348	48: beq \$t2, 'm', Found # if \$t2 == 'm' goto Found	0x2001006d	addi \$1,\$0,109
	4194352		0x102a0002	beq \$1,\$10,2
	4194356	49: addiu \$s0, \$s0, 1 #i++	0x26100001	addiu \$16,\$16,1
	4194360	50: loop #continue loop	0x08100008	4194336

Labels

Label	Address
(global)	
main	4194304
part5.asm	
null	268501364
result	268501360
msg2	268501342
msg1	268501318
newline	268501316
address2	268501282
address	268501248
str	268500992
printString	4194624
while	4194576
hexValue	4194572
digit	4194552

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	o m s a	\0 \0 \n o	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501024	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501056	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501088	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501152	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501184	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501216	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501248	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501280	0 1 \0 \0	0 0 1 0	\0 \0 2 0	\0 \0 \0 \0	0 0 0 2	1 0 0 1	\0 \0 \0 \0	\0 \0 \0 \0
268501312	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	o N \0
268501344	t a m	f h c	d n u o	\0 \0 \0 \n	. . \0 .	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501376	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
268501408	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII