

Lab Report

**ECPE 170 - Computer Systems and Networks -
Fall 2015**

Name: Zhiyun Yan

Lab Topic: MIPS Assembly Programming (Basic)

Question(1) Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

Before running

FP Regs

Int Regs [10]

Int Regs [10]

PC = 0

EPC = 0

Cause = 0

BadVAddr = 0

Status = 805371664

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 0

R2 [v0] = 0

R3 [v1] = 0

R4 [a0] = 1

R5 [a1] = 2147480192

R6 [a2] = 2147480200

R7 [a3] = 0

R8 [t0] = 0

R9 [t1] = 0

R10 [t2] = 0

R11 [t3] = 0

R12 [t4] = 0

R13 [t5] = 0

R14 [t6] = 0

R15 [t7] = 0

R16 [s0] = 0

R17 [s1] = 0

R18 [s2] = 0

R19 [s3] = 0

R20 [s4] = 0

R21 [s5] = 0

R22 [s6] = 0

R23 [s7] = 0

R24 [t8] = 0

R25 [t9] = 0

R26 [k0] = 0

R27 [k1] = 0

After running

FP Regs

Int Regs [10]

Int Regs [10]

PC = 4194412

EPC = 0

Cause = 0

BadVAddr = 0

Status = 805371664

HI = 0

LO = 10

R0 [r0] = 0

R1 [at] = 268500992

R2 [v0] = 10

R3 [v1] = 0

R4 [a0] = 1

R5 [a1] = 2147480192

R6 [a2] = 2147480200

R7 [a3] = 0

R8 [t0] = 2

R9 [t1] = 25

R10 [t2] = 10

R11 [t3] = 10

R12 [t4] = 15

R13 [t5] = 25

R14 [t6] = 23

R15 [t7] = 0

R16 [s0] = 10

R17 [s1] = 15

R18 [s2] = 5

R19 [s3] = 2

R20 [s4] = 7

R21 [s5] = -3

R22 [s6] = 0

R23 [s7] = 0

R24 [t8] = 0

R25 [t9] = 0

R26 [k0] = 0

R27 [k1] = 0

Question(2) Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, **circle the memory location (not register) that contains the final calculated value of Z.**

Answer:

Before

User data segment [10000000]..[10040000]
 [10000000]..[1003ffff] 00000000

User Stack [7ffff27c]..[80000000]

[7ffff27c]	0000000001			
[7ffff280]	2147480447	0000000000	2147483637	2147483619
[7ffff290]	2147483597	2147483580	2147483528	2147483513 y . . .
[7ffff2a0]	2147483460	2147483449	2147483433	2147483410	D . . . 9 . . .)
[7ffff2b0]	2147483393	2147483375	2147483307	2147483255 w . . .
[7ffff2c0]	2147483208	2147483194	2147481881	2147481823	H . . . :
[7ffff2d0]	2147481771	2147481723	2147481641	2147481590 { . . . }
[7ffff2e0]	2147481522	2147481428	2147481405	2147481387 T . . . = . . . + . . .
[7ffff2f0]	2147481354	2147481345	2147481326	2147481306
[7ffff300]	2147481283	2147481266	2147481251	2147481197 m . . .
[7ffff310]	2147481179	2147481150	2147481132	2147481106	[. . . > . . . ,
[7ffff320]	2147481091	2147481071	2147481063	2147481048
[7ffff330]	2147481004	2147480987	2147480961	2147480882 2 . . .
[7ffff340]	2147480864	2147480804	2147480772	2147480762
[7ffff350]	2147480741	2147480710	2147480699	2147480673 { . . . a . . .
[7ffff360]	2147480654	2147480620	2147480587	2147480562	N . . . ,
[7ffff370]	2147480524	2147480506	0000000000	0788529152 /
[7ffff380]	1701670760	2038069551	2038069601	1768042338	h o m e / y z y a y z y b / b i
[7ffff390]	1668637300	0796157291	0892416050	1818322527	t b u c k e t / 2 0 1 5 _ f a l
[7ffff3a0]	1667587948	0925984112	1634479920	0791687522	l _ e c p e 1 7 0 / l a b 1 0 /
[7ffff3b0]	1953653104	1935748657	1029636205	1920169263	p a r t l . a s m . _ = / u s r
[7ffff3c0]	1852400175	1937010991	0007170416	1414873432	/ b i n / q t s p i m . X A U T
[7ffff3d0]	1230131016	0792549716	1701670760	2038069551	H O R I T Y = / h o m e / y z y
[7ffff3e0]	2038069601	1479421794	1752462689	1953067631	a y z y b / . X a u t h o r i t
[7ffff3f0]	1329791097	1414680396	1028477509	1836019303	y . C O L O R T E R M = g n o m
[7ffff400]	1702112613	1852403058	1409313889	1146378309	e - t e r m i n a l . T E X T D
[7ffff410]	1229016399	1380533326	1937059645	1752379250	O M A I N D I R = / u s r / s h
[7ffff420]	0795177569	1633906540	0003106156	1397966156	a r e / l o c a l e / . L E S S
[7ffff430]	1327705705	1866000140	1647376650	1815047705	C T O R P / . . . / b i - /

After:

User data segment [10000000]..[10040000]
 [10000000]..[1000ffff] 00000000
 [10010000] 0000000023 0000000000 0000000000 0000000000
 [10010010]..[1003ffff] 00000000

User Stack [7ffff27c]..[80000000]
 [7ffff27c] 0000000001
 [7ffff280] 2147480447 0000000000 2147483637 2147483619
 [7ffff290] 2147483597 2147483580 2147483528 2147483513 y . . .
 [7ffff2a0] 2147483460 2147483449 2147483433 2147483410 D . . . 9 . . .)
 [7ffff2b0] 2147483393 2147483375 2147483307 2147483255 w . . .
 [7ffff2c0] 2147483208 2147483194 2147481881 2147481823 H . . . :
 [7ffff2d0] 2147481771 2147481723 2147481641 2147481590 { . . . }
 [7ffff2e0] 2147481522 2147481428 2147481405 2147481387 T . . . = . . . + . . .
 [7ffff2f0] 2147481354 2147481345 2147481326 2147481306
 [7ffff300] 2147481283 2147481266 2147481251 2147481197 m . . .
 [7ffff310] 2147481179 2147481150 2147481132 2147481106 [. . . > . . . ,
 [7ffff320] 2147481091 2147481071 2147481063 2147481048
 [7ffff330] 2147481004 2147480987 2147480961 2147480882 2 . . .
 [7ffff340] 2147480864 2147480804 2147480772 2147480762
 [7ffff350] 2147480741 2147480710 2147480699 2147480673 { . . . a . . .
 [7ffff360] 2147480654 2147480620 2147480587 2147480562 N . . . ,
 [7ffff370] 2147480524 2147480506 0000000000 0788529152 /

Question(3) Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

before:

Int Regs [10]		
PC	=	0
EPC	=	0
Cause	=	0
BadVAddr	=	0
Status	=	805371664
HI	=	0
LO	=	0
R0	[r0]	= 0
R1	[at]	= 0
R2	[v0]	= 0
R3	[v1]	= 0
R4	[a0]	= 1
R5	[a1]	= 2147480192
R6	[a2]	= 2147480200
R7	[a3]	= 0
R8	[t0]	= 0
R9	[t1]	= 0
R10	[t2]	= 0
R11	[t3]	= 0
R12	[t4]	= 0
R13	[t5]	= 0
R14	[t6]	= 0
R15	[t7]	= 0
R16	[s0]	= 0
R17	[s1]	= 0
R18	[s2]	= 0
R19	[s3]	= 0
R20	[s4]	= 0
R21	[s5]	= 0
R22	[s6]	= 0
R23	[s7]	= 0
R24	[t8]	= 0
R25	[t9]	= 0
R26	[k0]	= 0
R27	[k1]	= 0

After:

FP Regs		
Int Regs [10]		
Int Regs [10]		
PC	=	4194540
EPC	=	0
Cause	=	0
BadVAddr	=	0
Status	=	805371664
HI	=	-1
LO	=	-2
R0	[r0]	= 0
R1	[at]	= 268500992
R2	[v0]	= 10
R3	[v1]	= 0
R4	[a0]	= 1
R5	[a1]	= 2147480192
R6	[a2]	= 2147480200
R7	[a3]	= 0
R8	[t0]	= 1
R9	[t1]	= 2
R10	[t2]	= 3
R11	[t3]	= -1
R12	[t4]	= -2
R13	[t5]	= 0
R14	[t6]	= 10
R15	[t7]	= 15
R16	[s0]	= 0
R17	[s1]	= 0
R18	[s2]	= 0
R19	[s3]	= 0
R20	[s4]	= 0
R21	[s5]	= 0
R22	[s6]	= 0
R23	[s7]	= 0
R24	[t8]	= 5
R25	[t9]	= 2
R26	[k0]	= 0
R27	[k1]	= 0

Question(4) Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, **circle the memory location (not register) that contains the final calculated value of Z.**

Answer:

before:

Data					
User data segment [10000000]..[10040000]					
[10000000]..[10010003]	00000000				
[10010004]	0000000010	0000000015	0000000005	.	.
[10010010]..[1003ffff]	00000000			.	.
User Stack [7ffff27c]..[80000000]					
[7ffff27c]	0000000001			.	.
[7ffff280]	2147480447	0000000000	2147483637	2147483619	.
[7ffff290]	2147483597	2147483580	2147483528	2147483513	.
[7ffff2a0]	2147483460	2147483449	2147483433	2147483410	D . . . 9 . . .) . . . y . . .
[7ffff2b0]	2147483393	2147483375	2147483307	2147483255 w . . .
[7ffff2c0]	2147483208	2147483194	2147481881	2147481823	H . . . :
[7ffff2d0]	2147481771	2147481723	2147481641	2147481590	. . . { . . .)
[7ffff2e0]	2147481522	2147481428	2147481405	2147481387	. . . T . . . = . . . + . . .
[7ffff2f0]	2147481354	2147481345	2147481326	2147481306
[7ffff300]	2147481283	2147481266	2147481251	2147481197 m . . .
[7ffff310]	2147481179	2147481150	2147481132	2147481106	[. . . > . . . ,
[7ffff320]	2147481091	2147481071	2147481063	2147481048
[7ffff330]	2147481004	2147480987	2147480961	2147480882 2 . . .

after:

Data					
User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	-2 0000000010	0000000015	0000000005	.	.
[10010010]..[1003ffff]	00000000			.	.
User Stack [7ffff27c]..[80000000]					
[7ffff27c]	0000000001			.	.
[7ffff280]	2147480447	0000000000	2147483637	2147483619	.
[7ffff290]	2147483597	2147483580	2147483528	2147483513	.
[7ffff2a0]	2147483460	2147483449	2147483433	2147483410	D . . . 9 . . .) . . . y . . .
[7ffff2b0]	2147483393	2147483375	2147483307	2147483255 w . . .
[7ffff2c0]	2147483208	2147483194	2147481881	2147481823	H . . . :
[7ffff2d0]	2147481771	2147481723	2147481641	2147481590	. . . { . . .)
[7ffff2e0]	2147481522	2147481428	2147481405	2147481387	. . . T . . . = . . . + . . .
[7ffff2f0]	2147481354	2147481345	2147481326	2147481306

Question(5) Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

before:

after:

FP Regs		Int Regs [10]	
nt Regs [10]			
PC	=	0	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	0
R2	[v0]	=	0
R3	[v1]	=	0
R4	[a0]	=	1
R5	[a1]	=	2147480192
R6	[a2]	=	2147480200
R7	[a3]	=	0
R8	[t0]	=	0
R9	[t1]	=	0
R10	[t2]	=	0
R11	[t3]	=	0
R12	[t4]	=	0
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0

Int Regs [10]	
PC	= 4194428
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 805371664
HI	= 0
LO	= 0
R0	[r0] = 0
R1	[at] = 268500992
R2	[v0] = 10
R3	[v1] = 0
R4	[a0] = 1
R5	[a1] = 2147480192
R6	[a2] = 2147480200
R7	[a3] = 0
R8	[t0] = 10
R9	[t1] = 10
R10	[t2] = 10
R11	[t3] = 0
R12	[t4] = 10
R13	[t5] = 0
R14	[t6] = 0
R15	[t7] = 0
R16	[s0] = 0
R17	[s1] = 0
R18	[s2] = 0
R19	[s3] = 0
R20	[s4] = 0
R21	[s5] = 0
R22	[s6] = 0
R23	[s7] = 0
R24	[t8] = 0
R25	[t9] = 0
R26	[k0] = 0
R27	[k1] = 0

Question(6) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. **Circle the memory location (not register) that contains the final calculated values of I and Z.**

Answer:

before:

User data segment [10000000]..[10040000]

```
[10000000]..[1000ffff] 00000000
[10010000] 0000000005 0000000000 0000000000 0000000000 . . . . .
[10010010]..[1003ffff] 00000000
```

User Stack [7ffff27c]..[80000000]

```
[7ffff27c] 0000000001 . . . .
[7ffff280] 2147480447 0000000000 2147483637 2147483619 . . . .
[7ffff290] 2147483597 2147483580 2147483528 2147483513 . . . . y . . .
[7ffff2a0] 2147483460 2147483449 2147483433 2147483410 D . . . 9 . . . ) . . . .
[7ffff2b0] 2147483393 2147483375 2147483307 2147483255 . . . . w . . .
[7ffff2c0] 2147483208 2147483194 2147481881 2147481823 H . . . : . . . .
[7ffff2d0] 2147481771 2147481723 2147481641 2147481590 . . . . { . . . ) . . . .
[7ffff2e0] 2147481522 2147481428 2147481405 2147481387 . . . . T . . . = . . . + . . .
[7ffff2f0] 2147481354 2147481345 2147481326 2147481306 . . . . . . . .
[7ffff300] 2147481283 2147481266 2147481251 2147481197 . . . . . . . m . . .
[7ffff310] 2147481179 2147481150 2147481132 2147481106 [ . . . > . . . , . . . .
[7ffff320] 2147481091 2147481071 2147481063 2147481048 . . . . . . . .
[7ffff330] 2147481004 2147480987 2147480961 2147480882 . . . . . . . 2 . . .
[7ffff340] 2147480864 2147480804 2147480772 2147480762 . . . . . . . .
[7ffff350] 2147480741 2147480710 2147480699 2147480673 . . . . . { . . . a . . .
```

after:

Data

User data segment [10000000]..[10040000]

```
[10000000]..[1000ffff] 00000000
[10010000] 0000000010 0000000010 0000000000 0000000000 . . . . .
[10010010]..[1003ffff] 00000000
```

User Stack [7ffff27c]..[80000000]

```
[7ffff27c] 0000000001 . . . .
[7ffff280] 2147480447 0000000000 2147483637 2147483619 . . . .
[7ffff290] 2147483597 2147483580 2147483528 2147483513 . . . . y . . .
[7ffff2a0] 2147483460 2147483449 2147483433 2147483410 D . . . 9 . . . ) . . . .
[7ffff2b0] 2147483393 2147483375 2147483307 2147483255 . . . . w . . .
[7ffff2c0] 2147483208 2147483194 2147481881 2147481823 H . . . : . . . .
[7ffff2d0] 2147481771 2147481723 2147481641 2147481590 . . . . { . . . ) . . . .
[7ffff2e0] 2147481522 2147481428 2147481405 2147481387 . . . . T . . . = . . . + . . .
[7ffff2f0] 2147481354 2147481345 2147481326 2147481306 . . . . . . . .
[7ffff300] 2147481283 2147481266 2147481251 2147481197 . . . . . . . m . . .
[7ffff310] 2147481179 2147481150 2147481132 2147481106 [ . . . > . . . , . . . .
[7ffff320] 2147481091 2147481071 2147481063 2147481048 . . . . . . . .
[7ffff330] 2147481004 2147480987 2147480961 2147480882 . . . . . . . 2 . . .
[7ffff340] 2147480864 2147480804 2147480772 2147480762 . . . . . . . .
```

Question(7) Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

before:

after:

Int Regs [10]		
PC	=	0
EPC	=	0
Cause	=	0
BadVAddr	=	0
Status	=	805371664
HI	=	0
LO	=	0
R0	[r0]	= 0
R1	[at]	= 0
R2	[v0]	= 0
R3	[v1]	= 0
R4	[a0]	= 1
R5	[a1]	= 2147480192
R6	[a2]	= 2147480200
R7	[a3]	= 0
R8	[t0]	= 0
R9	[t1]	= 0
R10	[t2]	= 0
R11	[t3]	= 0
R12	[t4]	= 0
R13	[t5]	= 0
R14	[t6]	= 0
R15	[t7]	= 0
R16	[s0]	= 0
R17	[s1]	= 0
R18	[s2]	= 0
R19	[s3]	= 0
R20	[s4]	= 0
R21	[s5]	= 0

FP Regs

Int Regs [10]

nt Regs [10]

PC = 4194456

EPC = 0

Cause = 0

BadVAddr = 0

Status = 805371664

HI = 0

LO = 26

R0 [r0] = 0

R1 [at] = 1

R2 [v0] = 10

R3 [v1] = 0

R4 [a0] = 1

R5 [a1] = 2147480192

R6 [a2] = 2147480200

R7 [a3] = 0

R8 [t0] = 268500992

R9 [t1] = 268501012

R10 [t2] = -4

R11 [t3] = 268500992

R12 [t4] = 268501028

R13 [t5] = 19

R14 [t6] = 26

R15 [t7] = 5

R16 [s0] = 12

R17 [s1] = 0

R18 [s2] = 0

R19 [s3] = 0

R20 [s4] = 0

R21 [s5] = 0

R22 [s6] = 0

Question(8) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. **Circle the final values of array A.**

Answer:

before:

Data									
User data segment [10000000]..[10040000]									
[10000000]..[10010013]	00000000								
[10010014]	0000000001	0000000002	0000000003						
[10010020]	0000000004	0000000005	0000000012	0000000000					
[10010030]..[1003ffff]	00000000								
User Stack [7ffff27c]..[80000000]									
[7ffff27c]	0000000001								
[7ffff280]	2147480447	0000000000	2147483637	2147483619					
[7ffff290]	2147483597	2147483580	2147483528	2147483513					
[7ffff2a0]	2147483460	2147483449	2147483433	2147483410	D	.	.	9	.
[7ffff2b0]	2147483393	2147483375	2147483307	2147483255	w
[7ffff2c0]	2147483208	2147483194	2147481881	2147481823	H	.	.	:	.
[7ffff2d0]	2147481771	2147481723	2147481641	2147481590	.	.	.	{	.
[7ffff2e0]	2147481522	2147481428	2147481405	2147481387	.	.	.	T	.
[7ffff2f0]	2147481354	2147481345	2147481326	2147481306	.	.	.	=	.
[7ffff300]	2147481283	2147481266	2147481251	2147481197	m
[7ffff310]	2147481179	2147481150	2147481132	2147481106	[.	.	>	.
[7ffff320]	2147481091	2147481071	2147481063	2147481048
[7ffff330]	2147481004	2147480987	2147480961	2147480882	2
[7ffff340]	2147480864	2147480804	2147480772	2147480762
[7ffff350]	2147480741	2147480710	2147480699	2147480673	{
[7ffff360]	2147480654	2147480620	2147480587	2147480562	N

after:

Data									
User data segment [10000000]..[10040000]									
[10000000]..[1000ffff]	00000000								
[10010000]	0000000026	0000000028	0000000030	0000000032
[10010010]	0000000034	0000000001	0000000002	0000000003	"
[10010020]	0000000004	0000000005	0000000012	0000000000
[10010030]..[1003ffff]	00000000								
User Stack [7ffff27c]..[80000000]									
[7ffff27c]	0000000001								
[7ffff280]	2147480447	0000000000	2147483637	2147483619
[7ffff290]	2147483597	2147483580	2147483528	2147483513	y
[7ffff2a0]	2147483460	2147483449	2147483433	2147483410	D	.	.	9	.
[7ffff2b0]	2147483393	2147483375	2147483307	2147483255	w
[7ffff2c0]	2147483208	2147483194	2147481881	2147481823	H	.	.	:	.
[7ffff2d0]	2147481771	2147481723	2147481641	2147481590	.	.	.	{	.
[7ffff2e0]	2147481522	2147481428	2147481405	2147481387	.	.	.	T	.
[7ffff2f0]	2147481354	2147481345	2147481326	2147481306	.	.	.	=	.

Question(9) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Hex mode (right-click), since Decimal mode will not allow us to distinguish between bytes. **Circle two things: the final value of the pointer 'result' in memory, and the corresponding location that result points to.** Does that location in memory contain the ASCII code for the character 'e'? (If not, you had better check your work!)

Answer:

Data

User data segment [10000000]..[10040000]

[10000000]..[1000ffff]	00000000				
[10010000]	0000000a	00000000	00000000	00000000
[10010010]..[10010107]	00000000				
[10010108]	72694620	6d207473			F i r s t m
[10010110]	68637461	20746120	72646461	20737365	a t c h a t a d d r e s s
[10010120]	6f4e2000	74616d20	66206863	646e756f	. N o m a t c h f o u n d
[10010130]	54200020	6d206568	68637461	20676e69	. T h e m a t c h i n g
[10010140]	72616863	65746361	73692072	00000020	c h a r a c t e r i s . . .
[10010150]..[1003ffff]	00000000				

User Stack [7ffff27c]..[80000000]

[7ffff27c]	00000001			
[7ffff280]	7ffff37f	00000000	7fffffff5	7fffffe3
[7ffff290]	7fffffcd	7fffffbc	7fffff88	7fffff79 y . . .
[7ffff2a0]	7fffffaa	7fffff20	7fffff20	7fffff12	n n v