
MIPS assembler and simulator (Appendix A)

SPIM : MIPS simulator

- Downloadable from <http://spimsimulator.sourceforge.net/>
- Tutorials at https://booksite.elsevier.com/9780124077263/spim_tutorials.php
- Run setup after unzipping the package
- Open the assembly source code.
- F10 : Single step
- Windows :
 - Register window
 - Text Segment window
 - Data Segment window
 - Messages window
 - Console window (separate)

PCSpim

File Simulator Window Help

PC = 00000000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
 Status = 3000ff10 HI = 00000000 LO = 00000000

General Registers

R0 (r0) = 00000000	R8 (t0) = 00000000	R16 (s0) = 00000000	R24 (t8) = 00000000
R1 (at) = 00000000	R9 (t1) = 00000000	R17 (s1) = 00000000	R25 (t9) = 00000000
R2 (v0) = 00000000	R10 (t2) = 00000000	R18 (s2) = 00000000	R26 (k0) = 00000000
R3 (v1) = 00000000	R11 (t3) = 00000000	R19 (s3) = 00000000	R27 (k1) = 00000000
R4 (a0) = 00000000	R12 (t4) = 00000000	R20 (s4) = 00000000	R28 (gp) = 10008000

Register window

[0x00400010] 0x00c23021 addu \$6, \$6, \$2 ; 179: addu \$a2 \$a2 \$v0
 [0x00400014] 0x0c000000 jal 0x00000000 [main] ; 180: jal main
 [0x00400018] 0x00000000 nop ; 181: nop
 [0x0040001c] 0x3402000a ori \$2, \$0, 10 ; 183: li \$v0 10
 [0x00400020] 0x0000000c syscall ; 184: syscall

Text segment window

KERNEL
 [0x80000180] 0x0001d821 addu \$27, \$0, \$1 ; 82: move \$k1 \$at # Save \$at

DATA
 [0x10000000]...[0x10040000] 0x00000000

STACK
 [0x7ffffeffc] 0x00000000

memory contents

address

DATA segment window

KERNEL DATA
 [0x90000000] 0x78452020 0x74706563 0x206e6f69 0x636f2000

Message window

SPIM Version Version 7.2 of August 7, 2005
 Copyright 1990-2004 by James R. Larus (larus@cs.wisc.edu).
 All Rights Reserved.
 DOS and Windows ports by David A. Carley (dac@cs.wisc.edu).
 Copyright 1997 by Morgan Kaufmann Publishers, Inc.
 See the file README for a full copyright notice.
 Loaded: C:\Program Files\PCSpim\exceptions.s

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

3

FP Regs		Int Regs [16]		Data		Text	
PC = 0		Int Regs [16]		User Text Segment [00400000]..[00440000]			
EPC = 0		[00400000]	8fa40000	lw \$4, 0(\$29)	; 183: lw \$a0 0(\$sp) # arg0		
Cause = 0		[00400004]	27a50004	addiu \$5, \$29, 4	; 184: addiu \$a1 \$sp 4 # arg1		
BadVAddr = 0		[00400008]	24a60004	addiu \$6, \$5, 4	; 185: addiu \$a2 \$a1 4 # env		
Status = 3000ff10		[0040000c]	00041080	sll \$2, \$4, 2	; 186: sll \$v0 \$a0 2		
		[00400010]	00c23021	addu \$6, \$6, \$2	; 187: addu \$a2 \$a2 \$v0		
HI = 0		[00400014]	0c100009	jal 0x00400024 [main]	; 188: jal main		
LO = 0		[00400018]	00000000	nop	; 189: nop		
		[0040001c]	3402000a	ori \$2, \$0, 10	; 191: li \$v0 10		
R0 [r0] = 0		[00400020]	0000000c	syscall	; 192: syscall # syscall 10		
R1 [at] = 0		[00400024]	2005000a	addi \$5, \$0, 10	; 4: addi \$5, \$0, 10		
R2 [v0] = 0		[00400028]	20060014	addi \$6, \$0, 20	; 5: addi \$6, \$0, 20		
R3 [v1] = 0		[0040002c]	00a63820	add \$7, \$5, \$6	; 6: add \$7, \$5, \$6		
R4 [a0] = 0		Kernel Text Segment [80000000]..[80010000]					
R5 [a1] = 0		[80000180]	0001d821	addu \$27, \$0, \$1	; 90: move \$k1 \$at # Save \$a		
R6 [a2] = 7ffffe3c		[80000184]	3c019000	lui \$1, -28672	; 92: sw \$v0 s1 # Not re-ent		
R7 [a3] = 0		[80000188]	ac220200	sw \$2, 512(\$1)	; 93: sw \$a0 s2 # But we nee		
R8 [t0] = 0		[8000018c]	3c019000	lui \$1, -28672	; 95: mfc0 \$k0 \$13 # Cause r		
R9 [t1] = 0		[80000190]	ac240204	sw \$4, 516(\$1)	; 96: srl \$a0 \$k0 2 # Extra		
R10 [t2] = 0		[80000194]	401a6800	mfc0 \$26, \$13	; 97: andi \$a0 \$a0 0x1f		
R11 [t3] = 0		[80000198]	001a2082	srl \$4, \$26, 2	; 101: li \$v0 4 # syscall 4		
R12 [t4] = 0		[8000019c]	3084001f	andi \$4, \$4, 31	; 102: la \$a0 __m1__		
R13 [t5] = 0		[800001a0]	34020004	ori \$2, \$0, 4	; 103: syscall		
R14 [t6] = 0		[800001a4]	3c049000	lui \$4, -28672 [__m1__]	; 105: li \$v0 1 # syscall 1		
R15 [t7] = 0		[800001a8]	0000000c	syscall	; 106: srl \$a0 \$k0 2 # Extra		
R16 [s0] = 0		[800001ac]	34020001	ori \$2, \$0, 1	; 107: andi \$a0 \$a0 0x1f		
R17 [s1] = 0		[800001b0]	001a2082	srl \$4, \$26, 2	; 108: syscall		
R18 [s2] = 0		[800001b4]	3084001f	andi \$4, \$4, 31	; 110: li \$v0 4 # syscall 4		
R19 [s3] = 0		[800001b8]	0000000c	syscall	; 111: andi \$a0 \$k0 0x3c		
R20 [s4] = 0		[800001bc]	34020004	ori \$2, \$0, 4	; 112: lw \$a0 __excp(\$a0)		
R21 [s5] = 0		[800001c0]	3344003c	andi \$4, \$26, 60			
R22 [s6] = 0		[800001c4]	3c019000	lui \$1, -28672			
R23 [s7] = 0							

SPIM version 9.1.20 of August 29, 2017

Copyright 1990-2017 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Exercise

- text editor 를 사용하여 다음과 같은 내용으로 addi.s 라는 파일을 만든다.

```
.text
.globl main
main:
    addi $t0, $0, 10 # $t0 is an
    addi $t1, $0, 16 # $t1 is an
    add  $t2, $t0, $t1
```

Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

Assembler

- spim 에서 addi.s 을 open 하여 f10 으로 한 줄씩 수행한다.

Binary machine
language
program
(for MIPS)

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
0000001111100000000000000001000
```

Exercise

```
.text
```

```
.globl main
```

```
main:
```

```
    addi $8, $0, 10 # Q1 : 이 명령어를 실행한 후 8번  
레지스터의 값은 얼마인가? (십진수와 16진수로 쓰시오.)
```

```
    addi $9, $0, 16 # Q2 : 이 명령어를 실행한 후 9번  
레지스터의 값은 얼마인가? (십진수와 16진수로 쓰시오.)
```

```
    add $10, $8, $9 # Q3 : 이 명령어를 실행한 후 10번  
레지스터의 값은 얼마인가? (십진수와 16진수로 쓰시오.)
```

십진수 10을 **32bit** 16진수로 쓰면 0x0000 000A

이진수로는 0000 0000 0000 0000 0000 0000 0000 1010

십진수 16을 **32bit** 16진수로 쓰면 0x0000 0010

이진수로는 0000 0000 0000 0000 0000 0000 0001 0000

File Open

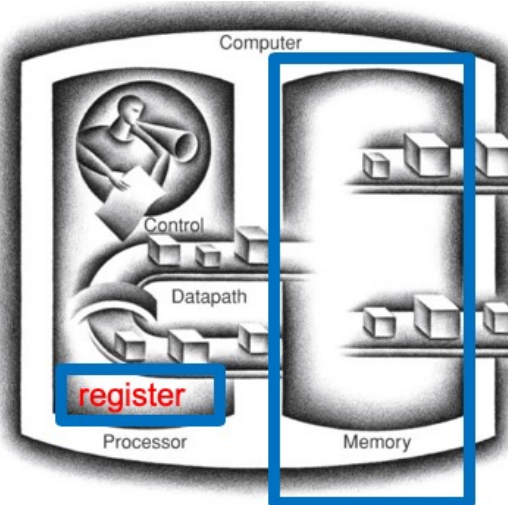


FP Regs Int Regs [16]

Int Regs [16]

PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0

R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 0
R5 [a1] = 0
R6 [a2] = 7ffffe3c
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



Data Text

Text

```
Text Segment [00400000]..[00440000]
; 183: lw $a0 0($sp) # argc
; 184: addiu $a1 $sp 4 # argv
; 185: addiu $a2 $a1 4 # envp
; 186: sll $v0 $a0 2
; 187: addu $a2 $a2 $v0
; 188: jal main
; 189: nop
; 191: li $v0 10
; 192: syscall # syscall 10 (exit)
; 4: addi $t0, $0, 10 # t0 is an alias for $t0
; 5: addi $t1, $0, 16
; 6: add $t2, $t0, $t1

Kernel Text Segment [80000000]..[80010000]
; 90: move $k1 $0 # Save $at
; 92: sw $v0 $1 # not re-entrant and
; 93: sw $a0 $1 # to use

; 95: mfc0 $k0 $13 # Cause register
; 96: srl $a0 $k0 2 # Extract ExcCode
; 97: andi $a0 $a0 0x1f
; 101: li $v0 4 # syscall 4 (print_string)
; 102: la $a0 __m1_
; 103: syscall
; 105: li $v0 1 # syscall 1 (print_int)
; 106: srl $a0 $k0 2 # Extract ExcCode
; 107: andi $a0 $a0 0x1f
; 108: syscall
; 110: li $v0 4 # syscall 4 (print_string)
```

메모리 주소

메모리에 저장된 내용

메모리에 저장된 내용을 disassemble 한 것

source file 에 쓰인 내용

F10 : Single Step Execution

SPIM version 9.1.20 of August 29, 2017
Copyright 1990-2017 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

FP Regs	Int Regs [16]	Data	Text
PC = 400024			User Text Segment [00400000]..[00440000]
EPC = 0			[00400000] 8fa40000 lw \$4, 0(\$29) ; 183: lw \$a0 0(\$sp) # argc
Cause = 0			[00400004] 27a50004 addiu \$5, \$29, 4 ; 184: addiu \$a1 \$sp 4 # argv
BadVAddr = 0			[00400008] 24a60004 addiu \$6, \$5, 4 ; 185: addiu \$a2 \$a1 4 # envp
Status = 3000fff10			[0040000c] 00041080 sll \$2, \$4, 2 ; 186: sll \$v0 \$a0 2
			[00400010] 00c23021 addu \$6, \$6, \$2 ; 187: addu \$a2 \$a2 \$v0
HI = 0			[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
LO = 0			[00400018] 00000000 nop ; 189: nop
			[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
R0 [r0] = 0			[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
R1 [at] = 0			[00400024] 2008000a addi \$8, \$0, 10 ; 4: addi \$t0, \$0, 10 # t0 is an alias
R2 [v0] = c			[00400028] 20090010 addi \$9, \$0, 16 ; 5: addi \$t1, \$0, 16
R3 [v1] = 0			[0040002c] 01095020 add \$10, \$8, \$9 ; 6: add \$t2, \$t0, \$t1
R4 [a0] = 3			
R5 [a1] = 7ffffddc			Kernel Text Segment [80000000]..[80010000]
R6 [a2] = 7ffffdec			[80000180] 0001d821 addu \$27, \$0, \$1 ; 90: move \$k1 \$at # Save \$at
R7 [a3] = 0			[80000184] 3c019000 lui \$1, -28672 ; 92: sw \$v0 \$1 # Not re-entrant and
R8 [t0] = 0			can't trust \$sp
R9 [t1] = 0			[80000188] ac220200 sw \$2, 512(\$1) ; 93: sw \$a0 \$2 # But we need to use
R10 [t2] = 0			[8000018c] 3c019000 lui \$1, -28672 ; 95: mfc0 \$k0 \$13 # Cause register
R11 [t3] = 0			registers
R12 [t4] = 0			[80000190] ac240204 sw \$4, 516(\$1) ; 96: srl \$a0 \$k0 2 # Extract ExcCode
R13 [t5] = 0			[80000194] 401a6800 mfc0 \$26, \$13 ; 97: andi \$a0 \$a0 0x1f
R14 [t6] = 0			[80000198] 001a2082 srl \$4, \$26, 2 ; 101: li \$v0 4 # syscall 4 (print_st
R15 [t7] = 0			[8000019c] 3084001f andi \$4, \$4, 31 ; 102: la \$a0 __m1_
R16 [s0] = 0			[800001a0] 34020004 ori \$2, \$0, 4 ; 103: syscall
R17 [s1] = 0			[800001a4] 3c049000 lui \$4, -28672 [__m1_] ; 105: li \$v0 1 # syscall 1 (print_in
R18 [s2] = 0			[800001a8] 0000000c syscall ; 106: srl \$a0 \$k0 2 # Extract ExcCode
R19 [s3] = 0			[800001ac] 34020001 ori \$2, \$0, 1 ; 107: andi \$a0 \$a0 0x1f
R20 [s4] = 0			[800001b0] 001a2082 srl \$4, \$26, 2 ; 108: syscall
R21 [s5] = 0			[800001b4] 3084001f andi \$4, \$4, 31 ; 110: li \$v0 4 # syscall 4 (print_st
R22 [s6] = 0			[800001b8] 0000000c syscall
R23 [s7] = 0			[800001bc] 34020004 ori \$2, \$0, 4

FP Regs		Int Regs [16]		Data		Text	
Int Regs [16]		Text					
PC	= 400028	User Text Segment [00400000]..[00440000]					
EPC	= 0	[00400000]	8fa40000	lw \$4, 0(\$29)		; 183: lw \$a0 0(\$sp) # argc	
Cause	= 0	[00400004]	27a50004	addiu \$5, \$29, 4		; 184: addiu \$a1 \$sp 4 # argv	
BadVAddr	= 0	[00400008]	24a60004	addiu \$6, \$5, 4		; 185: addiu \$a2 \$a1 4 # envp	
Status	= 3000ff10	[0040000c]	00041080	sll \$2, \$4, 2		; 186: sll \$v0 \$a0 2	
		[00400010]	00c23021	addu \$6, \$6, \$2		; 187: addu \$a2 \$a2 \$v0	
HI	= 0	[00400014]	0c100009	jal 0x00400024 [main]		; 188: jal main	
LO	= 0	[00400018]	00000000	nop		; 189: nop	
		[0040001c]	3402000a	ori \$2, \$0, 10		; 191: li \$v0 10	
R0 [r0]	= 0	[00400020]	0000000c	syscall		; 192: syscall # syscall 10 (exit)	
R1 [at]	= 0	[00400024]	2008000a	addi \$8, \$0, 10		; 4: addi \$t0, \$0, 10 # t0 is an alias	
R2 [v0]	= c	[00400028]	20090010	addi \$9, \$0, 16		; 5: addi \$t1, \$0, 16	
R3 [v1]	= 0	[0040002c]	01095020	add \$10, \$8, \$9		; 6: add \$t2, \$t0, \$t1	
R4 [a0]	= 3	Kernel Text Segment [80000000]..[80010000]					
R5 [a1]	= 7ffffddc	[80000180]	0001d821	addu \$27, \$0, \$1		; 90: move \$k1 \$at # Save \$at	
R6 [a2]	= 7ffffdec	[80000184]	3c019000	lui \$1, -28672		; 92: sw \$v0 \$1 # Not re-entrant and	
R7 [a3]	= 0	can't trust \$sp					
R8 [t0]	= a	[80000188]	ac220200	sw \$2, 512(\$1)			
R9 [t1]	= 0	[8000018c]	3c019000	lui \$1, -28672		; 93: sw \$a0 \$2 # But we need to use	
R10 [t2]	= 0	registers					
R11 [t3]	= 0	[80000190]	ac240204	sw \$4, 516(\$1)			
R12 [t4]	= 0	[80000194]	401a6800	mfc0 \$26, \$13		; 95: mfc0 \$k0 \$13 # Cause register	
R13 [t5]	= 0	[80000198]	001a2082	srl \$4, \$26, 2		; 96: srl \$a0 \$k0 2 # Extract ExcCode	
R14 [t6]	= 0	[8000019c]	3084001f	andi \$4, \$4, 31		; 97: andi \$a0 \$a0 0x1f	
R15 [t7]	= 0	[800001a0]	34020004	ori \$2, \$0, 4		; 101: li \$v0 4 # syscall 4 (print_st	
R16 [s0]	= 0	[800001a4]	3c049000	lui \$4, -28672 [__m1_]		; 102: la \$a0 __m1_	
R17 [s1]	= 0	[800001a8]	0000000c	syscall		; 103: syscall	
R18 [s2]	= 0	[800001ac]	34020001	ori \$2, \$0, 1		; 105: li \$v0 1 # syscall 1 (print_in	
R19 [s3]	= 0	[800001b0]	001a2082	srl \$4, \$26, 2		; 106: srl \$a0 \$k0 2 # Extract ExcCod	
R20 [s4]	= 0	[800001b4]	3084001f	andi \$4, \$4, 31		; 107: andi \$a0 \$a0 0x1f	
R21 [s5]	= 0	[800001b8]	0000000c	syscall		; 108: syscall	
R22 [s6]	= 0	[800001bc]	34020004	ori \$2, \$0, 4		; 110: li \$v0 4 # syscall 4 (print st	

SPIM version 9.1.20 of August 29, 2017

Copyright 1990-2017 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

HOCHIMM University

FP Regs		Int Regs [16]		Data		Text	
Int Regs [16]		Text					
PC	= 40002c	User Text Segment [00400000]..[00440000]					
EPC	= 0	[00400000]	8fa40000	lw \$4, 0(\$29)		; 183: lw \$a0 0(\$sp) # argc	
Cause	= 0	[00400004]	27a50004	addiu \$5, \$29, 4		; 184: addiu \$a1 \$sp 4 # argv	
BadVAddr	= 0	[00400008]	24a60004	addiu \$6, \$5, 4		; 185: addiu \$a2 \$a1 4 # envp	
Status	= 3000ff10	[0040000c]	00041080	sll \$2, \$4, 2		; 186: sll \$v0 \$a0 2	
		[00400010]	00c23021	addu \$6, \$6, \$2		; 187: addu \$a2 \$a2 \$v0	
HI	= 0	[00400014]	0c100009	jal 0x00400024 [main]		; 188: jal main	
LO	= 0	[00400018]	00000000	nop		; 189: nop	
		[0040001c]	3402000a	ori \$2, \$0, 10		; 191: li \$v0 10	
R0 [r0]	= 0	[00400020]	0000000c	syscall		; 192: syscall # syscall 10 (exit)	
R1 [at]	= 0	[00400024]	2008000a	addi \$8, \$0, 10		; 4: addi \$t0, \$0, 10 # t0 is an alias	
R2 [v0]	= c	[00400028]	20090010	addi \$9, \$0, 16		; 5: addi \$t1, \$0, 16	
R3 [v1]	= 0	[0040002c]	01095020	add \$10, \$8, \$9		; 6: add \$t2, \$t0, \$t1	
R4 [a0]	= 3	Kernel Text Segment [80000000]..[80010000]					
R5 [a1]	= 7ffffddc	[80000180]	0001d821	addu \$27, \$0, \$1		; 90: move \$k1 \$at # Save \$at	
R6 [a2]	= 7ffffdec	[80000184]	3c019000	lui \$1, -28672		; 92: sw \$v0 \$1 # Not re-entrant and	
R7 [a3]	= 0	can't trust \$sp					
R8 [t0]	= a	[80000188]	ac220200	sw \$2, 512(\$1)			
R9 [t1]	= 10	[8000018c]	3c019000	lui \$1, -28672		; 93: sw \$a0 \$2 # But we need to use	
R10 [t2]	= 0	registers					
R11 [t3]	= 0	[80000190]	ac240204	sw \$4, 516(\$1)			
R12 [t4]	= 0	[80000194]	401a6800	mfc0 \$26, \$13		; 95: mfc0 \$k0 \$13 # Cause register	
R13 [t5]	= 0	[80000198]	001a2082	srl \$4, \$26, 2		; 96: srl \$a0 \$k0 2 # Extract ExcCode	
R14 [t6]	= 0	[8000019c]	3084001f	andi \$4, \$4, 31		; 97: andi \$a0 \$a0 0x1f	
R15 [t7]	= 0	[800001a0]	34020004	ori \$2, \$0, 4		; 101: li \$v0 4 # syscall 4 (print_st	
R16 [s0]	= 0	[800001a4]	3c049000	lui \$4, -28672 [__m1_]		; 102: la \$a0 __m1_	
R17 [s1]	= 0	[800001a8]	0000000c	syscall		; 103: syscall	
R18 [s2]	= 0	[800001ac]	34020001	ori \$2, \$0, 1		; 105: li \$v0 1 # syscall 1 (print_in	
R19 [s3]	= 0	[800001b0]	001a2082	srl \$4, \$26, 2		; 106: srl \$a0 \$k0 2 # Extract ExcCod	
R20 [s4]	= 0	[800001b4]	3084001f	andi \$4, \$4, 31		; 107: andi \$a0 \$a0 0x1f	
R21 [s5]	= 0	[800001b8]	0000000c	syscall		; 108: syscall	
R22 [s6]	= 0	[800001bc]	34020004	ori \$2, \$0, 4		; 110: li \$v0 4 # syscall 4 (print st	

SPIM version 9.1.20 of August 29, 2017

Copyright 1990-2017 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

HOCHIMM University

FP Regs		Int Regs [16]		Data		Text	
Int Regs [16]		Text					
PC	= 400030	User Text Segment [00400000]..[00440000]					
EPC	= 0	[00400000]	8fa40000	lw \$4, 0(\$29)		; 183: lw \$a0 0(\$sp) # argc	
Cause	= 0	[00400004]	27a50004	addiu \$5, \$29, 4		; 184: addiu \$a1 \$sp 4 # argv	
BadVAddr	= 0	[00400008]	24a60004	addiu \$6, \$5, 4		; 185: addiu \$a2 \$a1 4 # envp	
Status	= 3000ff10	[0040000c]	00041080	sll \$2, \$4, 2		; 186: sll \$v0 \$a0 2	
		[00400010]	00c23021	addu \$6, \$6, \$2		; 187: addu \$a2 \$a2 \$v0	
HI	= 0	[00400014]	0c100009	jal 0x00400024 [main]		; 188: jal main	
LO	= 0	[00400018]	00000000	nop		; 189: nop	
		[0040001c]	3402000a	ori \$2, \$0, 10		; 191: li \$v0 10	
R0 [r0]	= 0	[00400020]	0000000c	syscall		; 192: syscall # syscall 10 (exit)	
R1 [at]	= 0	[00400024]	2008000a	addi \$8, \$0, 10		; 4: addi \$t0, \$0, 10 # t0 is an alias	
R2 [v0]	= c	[00400028]	20090010	addi \$9, \$0, 16		; 5: addi \$t1, \$0, 16	
R3 [v1]	= 0	[0040002c]	01095020	add \$10, \$8, \$9		; 6: add \$t2, \$t0, \$t1	
R4 [a0]	= 3	Kernel Text Segment [80000000]..[80010000]					
R5 [a1]	= 7ffffddc	[80000180]	0001d821	addu \$27, \$0, \$1		; 90: move \$k1 \$at # Save \$at	
R6 [a2]	= 7ffffdec	[80000184]	3c019000	lui \$1, -28672		; 92: sw \$v0 \$1 # Not re-entrant and	
R7 [a3]	= 0	can't trust \$sp					
R8 [t0]	= a	[80000188]	ac220200	sw \$2, 512(\$1)			
R9 [t1]	= 10	[8000018c]	3c019000	lui \$1, -28672		; 93: sw \$a0 \$2 # But we need to use	
R10 [t2]	= 1a	registers					
R11 [t3]	= 0	[80000190]	ac240204	sw \$4, 516(\$1)			
R12 [t4]	= 0	[80000194]	401a6800	mfc0 \$26, \$13		; 95: mfc0 \$k0 \$13 # Cause register	
R13 [t5]	= 0	[80000198]	001a2082	srl \$4, \$26, 2		; 96: srl \$a0 \$k0 2 # Extract ExcCode	
R14 [t6]	= 0	[8000019c]	3084001f	andi \$4, \$4, 31		; 97: andi \$a0 \$a0 0x1f	
R15 [t7]	= 0	[800001a0]	34020004	ori \$2, \$0, 4		; 101: li \$v0 4 # syscall 4 (print_st	
R16 [s0]	= 0	[800001a4]	3c049000	lui \$4, -28672 [__m1_]		; 102: la \$a0 __m1_	
R17 [s1]	= 0	[800001a8]	0000000c	syscall		; 103: syscall	
R18 [s2]	= 0	[800001ac]	34020001	ori \$2, \$0, 1		; 105: li \$v0 1 # syscall 1 (print_in	
R19 [s3]	= 0	[800001b0]	001a2082	srl \$4, \$26, 2		; 106: srl \$a0 \$k0 2 # Extract ExcCod	
R20 [s4]	= 0	[800001b4]	3084001f	andi \$4, \$4, 31		; 107: andi \$a0 \$a0 0x1f	
R21 [s5]	= 0	[800001b8]	0000000c	syscall		; 108: syscall	
R22 [s6]	= 0	[800001bc]	34020004	ori \$2, \$0, 4		; 110: li \$v0 4 # syscall 4 (print st	

SPIM version 9.1.20 of August 29, 2017

Copyright 1990-2017 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

HOCHIMM University

FP Regs Int Regs [16] Data Text

Int Regs [16] Text

PC = 400030

EPC = 0

Cause = 0

BadVAddr = 0

Status = 3000ff10

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 0

R2 [v0] = c

R3 [v1] = 0

R4 [a0] = 3

R5 [a1] = 7ffffddc

R6 [a2] = 7ffffdec

R7 [a3] = 0

R8 [t0] = a

R9 [t1] = 10

R10 [t2] = 1a

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

User Text Segment [00400000]..[00440000]

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 2008000a addi $8, $0, 10 ; 4: addi $t0, $0, 10 # t0 is an alias
[00400028] 20090010 addi $9, $0, 16 ; 5: addi $t1, $0, 16
[0040002c] 01095020 add $10, $8, $9 ; 6: add $t2, $t0, $t1
[80000180] 00000000 ; 70: move $k1 $at # Save $at
[80000184] 00000000 ; 72: sw $v0 $1 # Not re-entrant and
[80000188] 00000000 ; 73: sw $a0 $2 # But we need to use
[8000018c] 00000000 ; 95: mfc0 $k0 $13 # Cause register
[80000194] 401a6800 mfc0 $26, $13 ; 96: srl $a0 $k0 2 # Extract ExcCode
[80000198] 001a2082 srl $4, $26, 2 ; 97: andi $a0 $a0 0x1f
[8000019c] 3084001f andi $4, $4, 31 ; 101: li $v0 4 # syscall 4 (print_st
[800001a0] 34020004 ori $2, $0, 4 ; 102: la $a0 __m1_
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 103: syscall
[800001a8] 0000000c syscall ; 105: li $v0 1 # syscall 1 (print_in
[800001ac] 34020001 ori $2, $0, 1 ; 106: srl $a0 $k0 2 # Extract ExcCode
[800001b0] 001a2082 srl $4, $26, 2 ; 107: andi $a0 $a0 0x1f
[800001b4] 3084001f andi $4, $4, 31 ; 108: syscall
[800001b8] 0000000c syscall ; 110: li $v0 4 # syscall 4 (print_st
[800001bc] 34020004 ori $2, $0, 4

```

Attempt to execute non-instruction at 0x00400030

Abort OK

Copyright 1990-2017, by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Attempt to execute non-instruction at 0x00400030

SPIM 에서 register 값을 이진수로 보려면?

The screenshot shows the QtSpim MIPS simulator interface. The 'Registers' menu is open, with 'Binary' selected. The 'Int Regs [2]' window is active, displaying the following register values:

Register	Value
PC	= 1000000000000000000110000
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 110000000000000111111100010000
HI	= 0
LO	= 0
R0 [r0]	= 0
R1 [at]	= 0
R2 [v0]	= 1100
R3 [v1]	= 0
R4 [a0]	= 11
R5 [a1]	= 111111111111111111111000110000
R6 [a2]	= 111111111111111111111001000000
R7 [a3]	= 0
R8 [t0]	= 1010
R9 [t1]	= 10000
R10 [t2]	= 11010
R11 [t3]	= 0
R12 [t4]	= 0
R13 [t5]	= 0
R14 [t6]	= 0

The 'Text' window is also active, showing the following memory segments:

User Text Segment [00400000]

Address	Hex	Assembly
[00400000]	8fa40000	lw \$4, 0(\$29)
[00400004]	27a50004	addiu \$5, \$29, 4
[00400008]	24a60004	addiu \$6, \$5, 4
[0040000c]	00041080	sll \$2, \$4, 2
[00400010]	00c23021	addu \$6, \$6, \$2
[00400014]	0c100009	jal 0x00400024 [main]
[00400018]	00000000	nop
[0040001c]	3402000a	ori \$2, \$0, 10
[00400020]	0000000c	syscall
[00400024]	2008000a	addi \$8, \$0, 10
[00400028]	20090010	addi \$9, \$0, 16
[0040002c]	01095020	add \$10, \$8, \$9

Kernel Text Segment [80000000]

Address	Hex	Assembly
[80000180]	0001d821	addu \$27, \$0, \$1
[80000184]	3c019000	lui \$1, -28672
[80000188]	ac220200	sw \$2, 512(\$1)
[8000018c]	3c019000	lui \$1, -28672
[80000190]	ac240204	sw \$4, 516(\$1)
[80000194]	401a6800	mfc0 \$26, \$13
[80000198]	001a2082	srl \$4, \$26, 2

HW1 : Due 9/12(화) 자정

- spim 설치
- addi.s 작성
- spim 에서 수행하여 레지스터 8,9,10 의 내용이 보이는 화면을 캡취하여 ecampus 과제 1로 upload 할 것.
- 제출 시 주의 사항 : 캡취 그림 1개만 첨부할 것. 화일 크기는 1MB 미만