Introducing QtSpim

Based on quiz examples

Dr. Kokalj-Filipovic
kokaljfilipovic@rowan.edu
CompOrg 2022

Section 2 Quiz Question

- **Write the MIPS assembly code for the C statement:**
- **A[10] = g + A[i]**
- **The base address of the array A is stored in $s0. Variable i is stored in $s1. Variable g is stored in $s2.**

```
sll $s1, $s1,2   # multiply i by 4
add     $t0, $s1, $s0      # calculate   current address
lw      $t1, 0($t0)        #  load from the current address A[i]
add     $t0, $t1, $s2    # A[i]+g
sw      $t0, 40($s0)    #store the sum to A[10]
```

Creating a file interpretable by QtSim

- Use a simple text editor (Notebook in Windows, vi in Linux, mipster)
- Write pseudo instructions to declare the data (A, i, g)
- Add pseudo instructions to denote the start of the program
- Write pseudo instructions to load the registers given in the question:
  - **The base address of the array A is stored in $s0. Variable i is stored in $s1. Variable g is stored in $s2.**
- Paste the assembly instructions from the solution I posted
- Add a syscall to exit from the program

How MIPS communicates with peripherals

- **syscall (trap)**
- A number of system services, mainly for input and output, are available for use by your MIPS program.
- MIPS register contents are not affected by a system call, except for result registers as specified in the link below.
- **How to use SYSCALL system services**
- Step 1. Load the service number in register $v0.
  Step 2. Load argument values, if any, in $a0, $a1, $a2, or $f12 as specified.
  Step 3. Issue the SYSCALL instruction.
  Step 4. Retrieve return values, if any, from result registers as specified.
- http://courses.missouristate.edu/KenVollmar/Mars/Help/SyscallHelp.html
- https://student.cs.uwaterloo.ca/~isg/res/mips/traps

Our example in a file

```
.data
A: .word 10 11 12 13 14 15 16 17 18 20
21 22 23 24
ii: .word 2
g:  .word 5

.text
.globl main

main:
la      $s0, A
lw      $s1, ii
lw      $s2, g
sll $s1, $s1,2

add     $t0, $s1, $s0

lw      $t1, 0($t0)
add     $t0, $t1, $s2

sw      $t0, 40($s0)

ori $v0, $zero, 10          #exit
syscall
```

Modified file

```
.data
A: .word 10 11 12 13 14 15 16 17 18 20 21 22 23
24
ii: .word 2
g:  .word 5
msg:      .asciiz "Section 2 result:"
nLine:    .asciiz "\n"

.text
.globl main

main:
la        $s0, A
lw        $s1, ii
lw        $s2, g
sll $s1, $s1,2

add     $t0, $s1, $s0

lw      $t1, 0($t0)
add     $t0, $t1, $s2

sw      $t0, 40($s0)

li $v0, 4
la $a0, msg     # argument: string
syscall         # print the string
la $a0, nLine           # argument: string
syscall         # print the string
# Print integer
li $v0, 1
add $a0, $t0, $zero
syscall
ori $v0, $zero, 10                  #exit
syscall
```

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Section 3 Quiz

- **Write the MIPS assembly code for the C statement:**
- **A[i] = g - A[5]**
- **The base address of the array A is stored in $s0. Variable i is stored in $s1. Variable g is stored in $s2.**

```
lw $t1, 20($s0)   #  load from the address A[5]
sub $t0, $s2, $t1   # g-A[5]

sll $s1, $s1,2   # multiple i by 4
add  $t1, $s1, $s0    # calculate   current address

sw  $t0, 0($t1) #store the sum to A[i]
```

Our example in a file

```
.data
A: .word 10 11 12 13 14 15 16 17 18 20
21 22 23 24
ii: .word 2
g:  .word 5

.text
.globl main

main:
la      $s0, A
lw      $s1, ii
lw      $s2, g
lw $t1, 20($s0)

sub $t0, $s2, $t1

sll $s1, $s1,2
add $t1, $s1, $s0

sw  $t0, 0($t1)

ori $v0, $zero, 10      #exit
syscall
```

MIPS Reference Sheet

- https://mathcs.holycross.edu/~csci226/MIPS/MIPSReferenceHO.pdf