#6 : MIPS Programming II

Computer Architecture 2020/2021

Ricardo Rocha

Computer Science Department, Faculty of Sciences, University of Porto

Program Structure

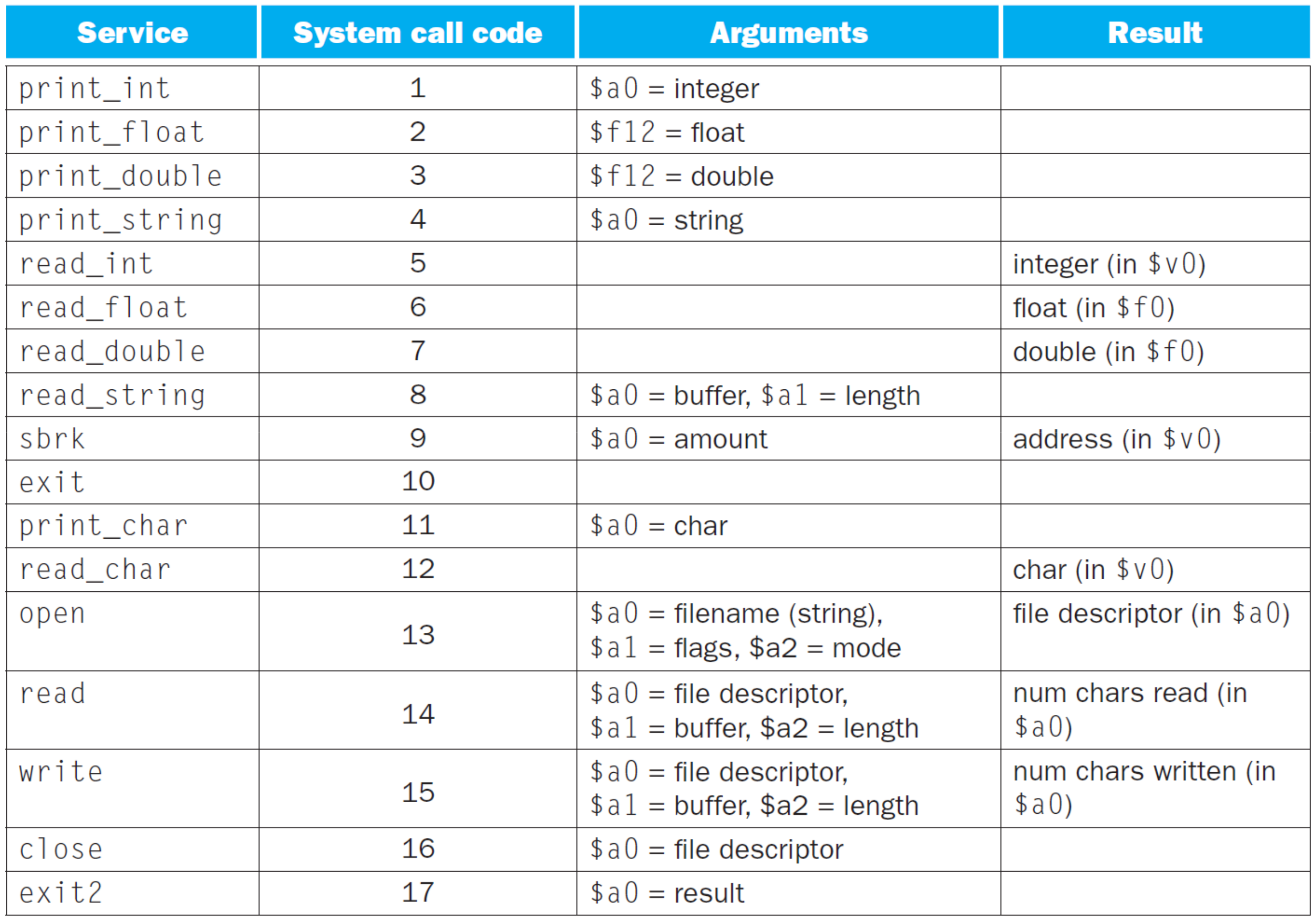
```
        .data                 # data segment (constants and global variables)
_b1:    .byte 1               # byte (8 bits) with value 1
_h1:    .half 10              # half word (16 bits) with value 10
_w1:    .word 100             # word (32 bits) with value 100
_a1:    .byte 1, 2, 3, 4      # array of 4 bytes with values 1, 2, 3 and 4
_a2:    .word 0:100           # array of 100 words with values 0
_s1:    .ascii  “abc\n”       # string not null terminated
_s2:    .asciiz “123”         # string null terminated”
_e1:    .space 100            # leave 100 bytes of space

        .text                 # text segment (program instructions)
_main:                        # main procedure
        ...
        li $v0, 10            # load code 10 for system call exit()
        syscall               # exit()
```

System Calls

To request a service, **load the system call code into register $v0** and **arguments into registers $a0–$a3 or $f12** (floating point values).

Return values are put in **register $v0 or $f0** (floating-point results).



Service	System call code	Arguments	Result
print_int	1	$a0 = integer	
print_float	2	$f12 = float	
print_double	3	$f12 = double	
print_string	4	$a0 = string	
read_int	5		integer (in $v0)
read_float	6		float (in $f0)
read_double	7		double (in $f0)
read_string	8	$a0 = buffer, $a1 = length	
sbrk	9	$a0 = amount	address (in $v0)
exit	10		
print_char	11	$a0 = char	
read_char	12		char (in $v0)
open	13	$a0 = filename (string), $a1 = flags, $a2 = mode	file descriptor (in $a0)
read	14	$a0 = file descriptor, $a1 = buffer, $a2 = length	num chars read (in $a0)
write	15	$a0 = file descriptor, $a1 = buffer, $a2 = length	num chars written (in $a0)
close	16	$a0 = file descriptor	
exit2	17	$a0 = result	

MARS – MIPS Simulator

Main functionalities:

- Edit programs (**assembly**)
- Compile (**assembler**)
- Run and/or execute step by step
- See the memory contents and the values in the set of registers

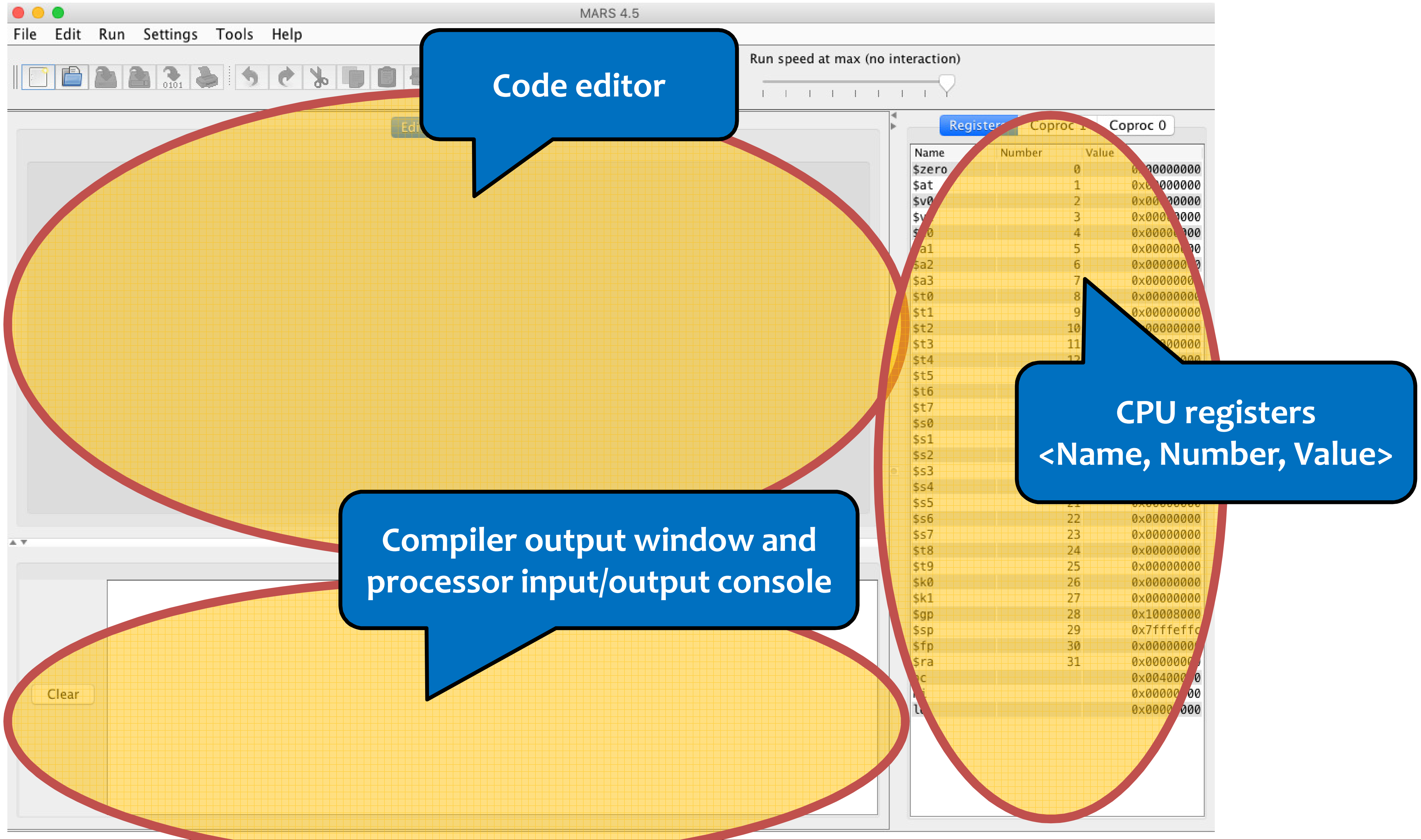
Download Mars4_5.jar:

- http://www.softpedia.com

Command to execute:

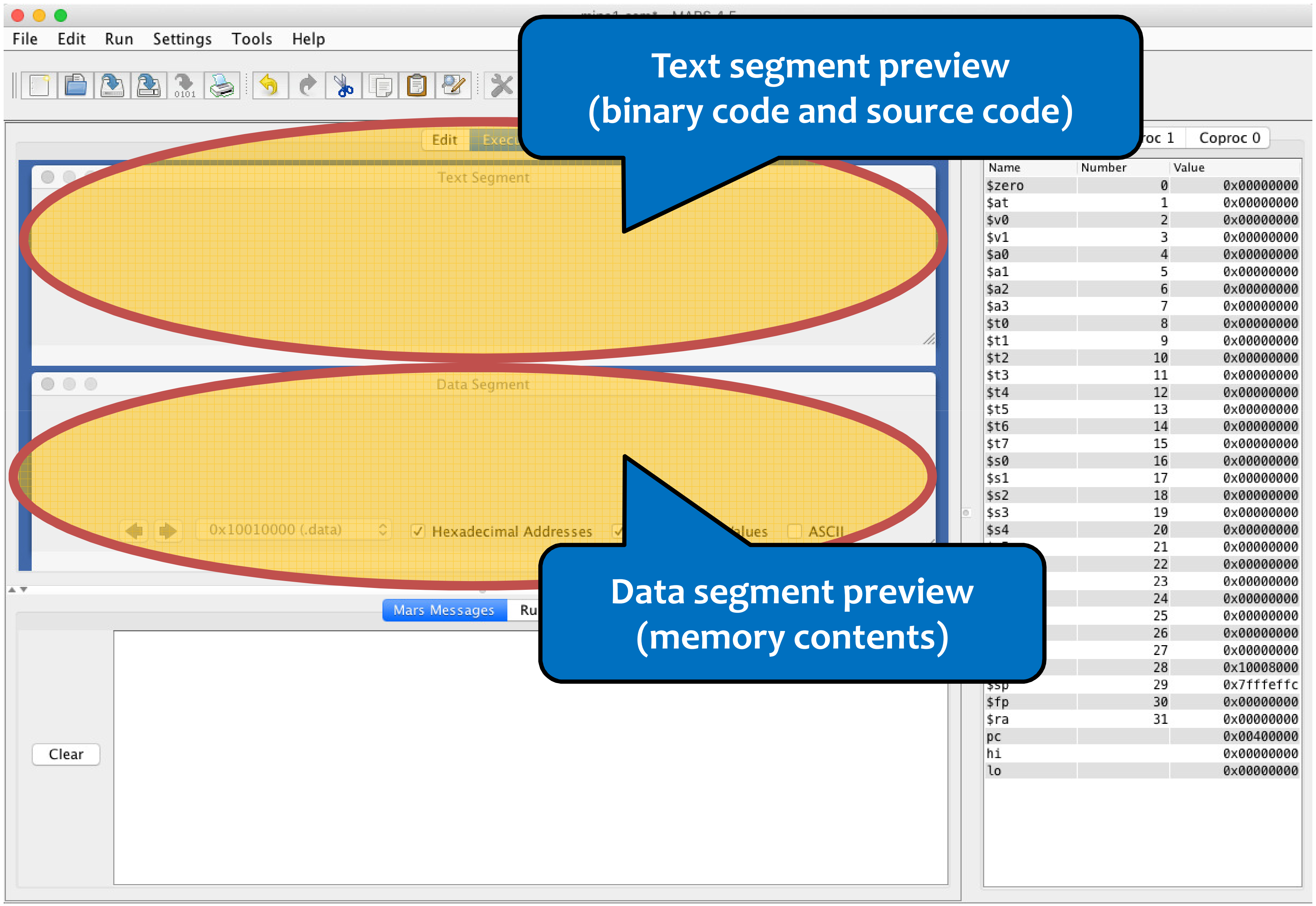
- java –jar Mars4_5.jar

MARS – MIPS Simulator


MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit
Registers
Coproc 1
Coproc 0
Clear
Code editor
CPU registers <Name, Number, Value>
Compiler output window and processor input/output console

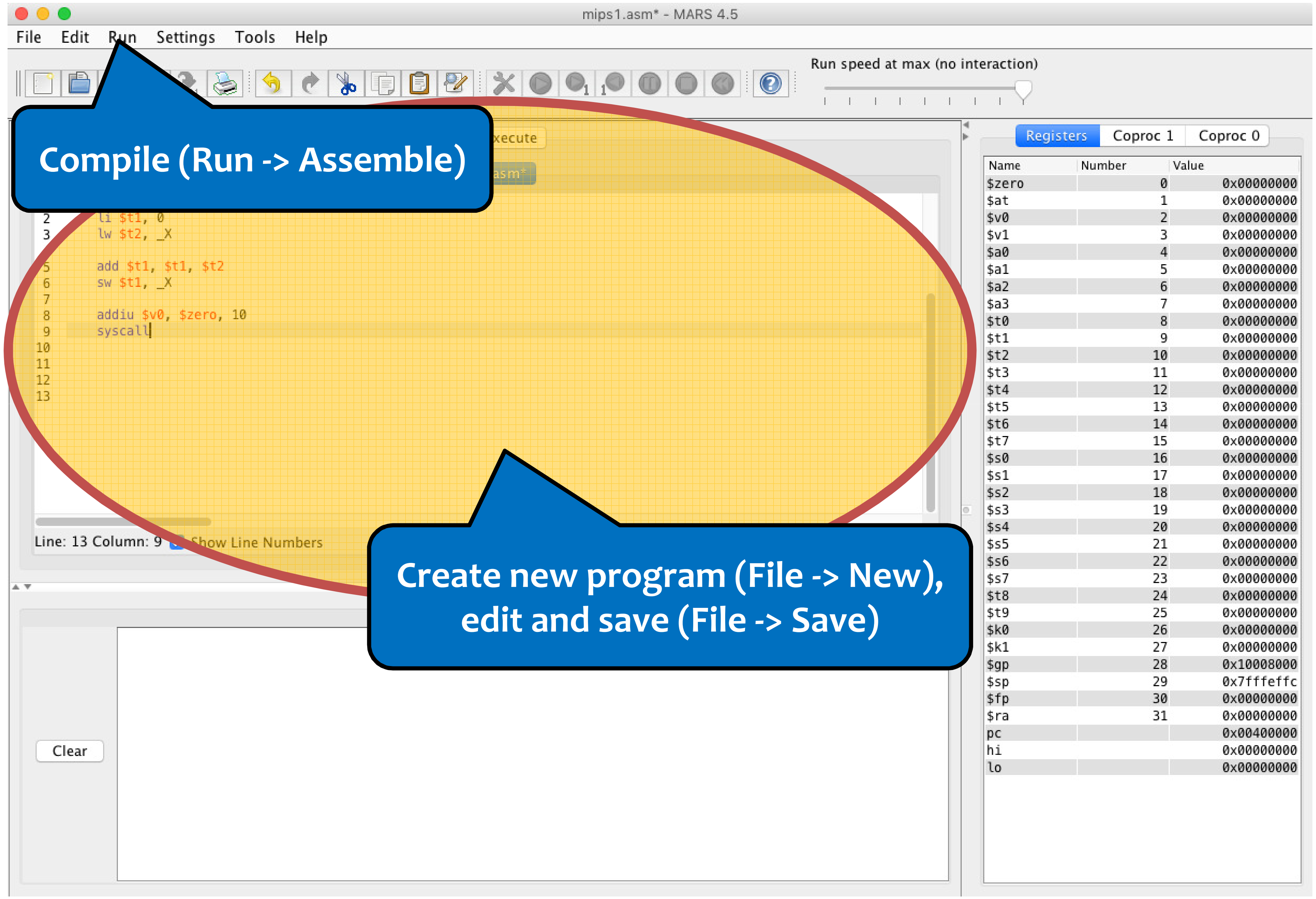
Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	[illegible]
$t5	[illegible]	[illegible]
$t6	[illegible]	[illegible]
$t7	[illegible]	[illegible]
$s0	[illegible]	[illegible]
$s1	[illegible]	[illegible]
$s2	[illegible]	[illegible]
$s3	[illegible]	[illegible]
$s4	[illegible]	[illegible]
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator


File Edit Run Settings Tools Help
Edit Execu
Text Segment
Data Segment
0x10010000 (.data)
Hexadecimal Addresses
alues
ASCII
Mars Messages
Ru
Clear
roc 1
Coproc 0
Text segment preview (binary code and source code)
Data segment preview (memory contents)

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
	21	0x00000000
	22	0x00000000
	23	0x00000000
	24	0x00000000
	25	0x00000000
	26	0x00000000
	27	0x00000000
	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator


mips1.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Execute
asm*
Registers Coproc 1 Coproc 0
Line: 13 Column: 9
Show Line Numbers
Clear

Compile (Run -> Assemble)

Create new program (File -> New), edit and save (File -> Save)

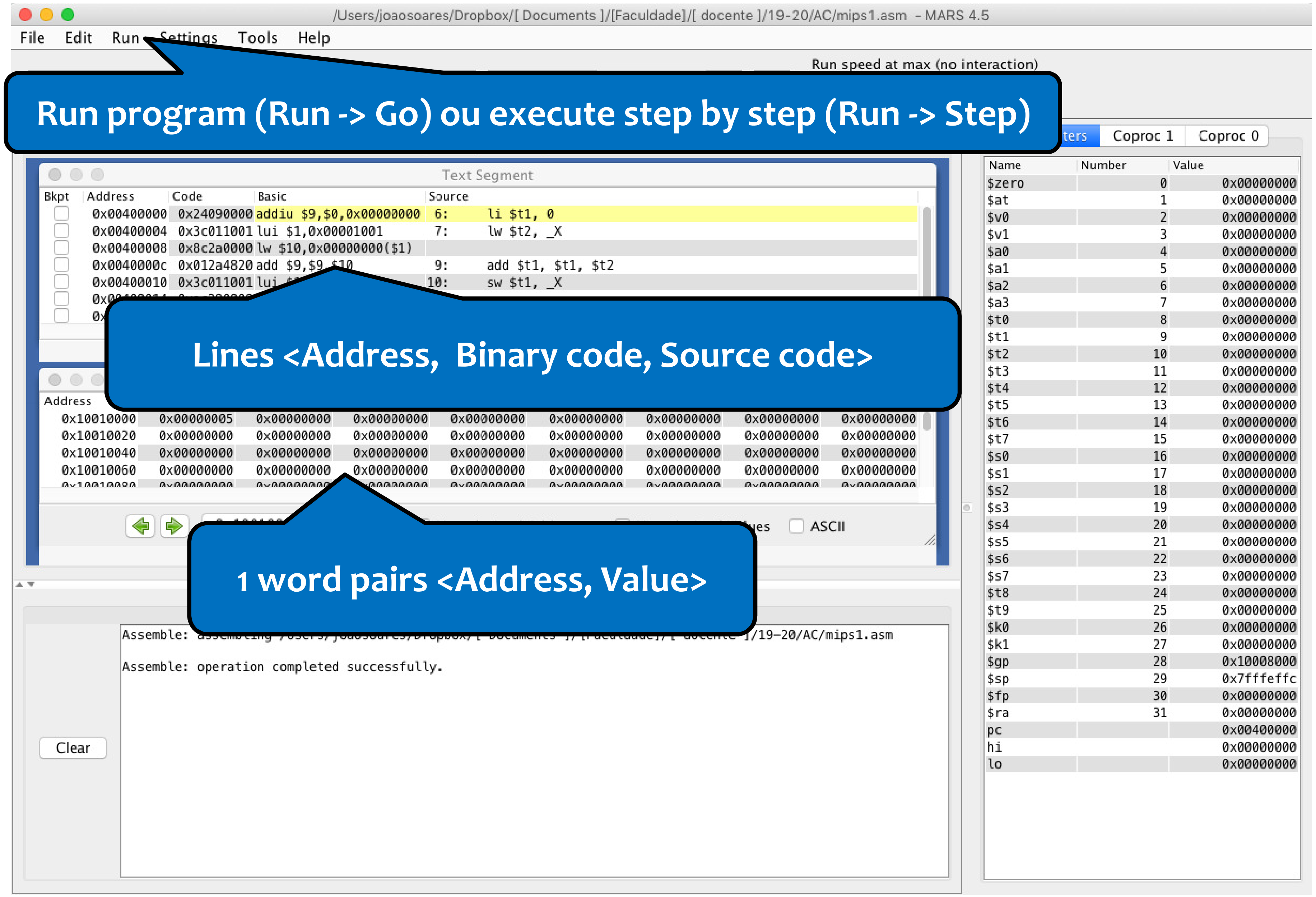
```
    li $t1, 0
    lw $t2, _X

    add $t1, $t1, $t2
    sw $t1, _X

    addiu $v0, $zero, 10
    syscall
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator


/Users/joaosoares/Dropbox/[Documents]/[Faculdade]/[docente]/19-20/AC/mips1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Coproc 1
Coproc 0
Text Segment
ASCII
Clear

Run program (Run -> Go) ou execute step by step (Run -> Step)

Lines <Address, Binary code, Source code>

1 word pairs <Address, Value>

```
Bkpt  Address     Code       Basic                      Source
      0x00400000  0x24090000 addiu $9,$0,0x00000000     6:      li $t1, 0
      0x00400004  0x3c011001 lui $1,0x00001001          7:      lw $t2, _X
      0x00400008  0x8c2a0000 lw $10,0x00000000($1)
      0x0040000c  0x012a4820 add $9,$9,$10              9:      add $t1, $t1, $t2
      0x00400010  0x3c011001 lui                        10:     sw $t1, _X
```

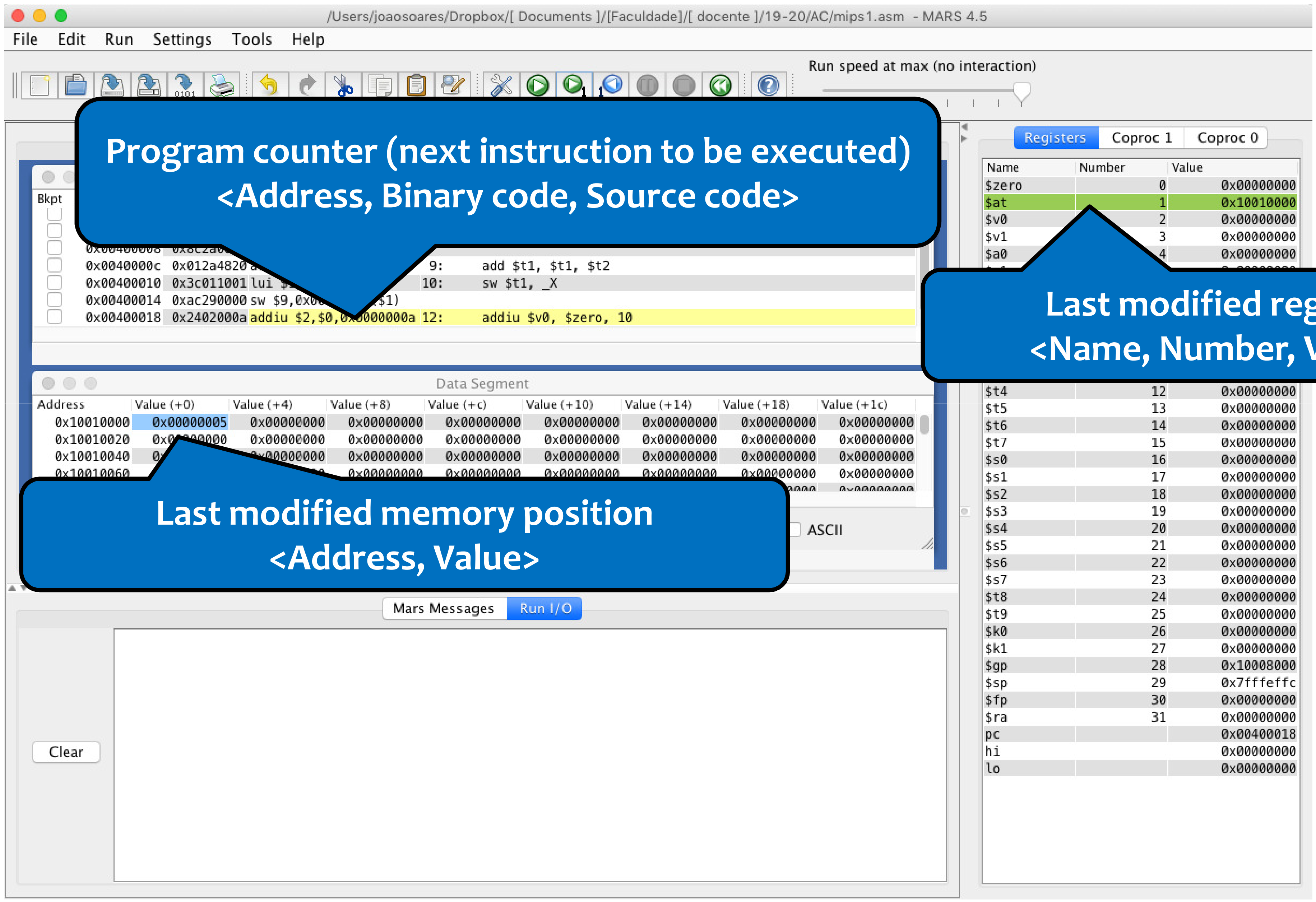
```
Address
  0x10010000  0x00000005  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010020  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010040  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010060  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
```

```
Assemble: ]/19-20/AC/mips1.asm

Assemble: operation completed successfully.
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator


/Users/joaosoares/Dropbox/[Documents]/[Faculdade]/[docente]/19-20/AC/mips1.asm - MARS 4.5
File
Edit
Run
Settings
Tools
Help
Run speed at max (no interaction)
Bkpt
Data Segment
ASCII
Mars Messages
Run I/O
Clear
Registers
Coproc 1
Coproc 0

Program counter (next instruction to be executed)
<Address, Binary code, Source code>

```
0x0040000c  0x012a4820 a                        9:      add $t1, $t1, $t2
0x00400010  0x3c011001 lui $                    10:     sw $t1, _X
0x00400014  0xac290000 sw $9,0x0          $1)
0x00400018  0x2402000a addiu $2,$0,0x0000000a 12:      addiu $v0, $zero, 10
```

```
Address      Value (+0)  Value (+4)  Value (+8)  Value (+c)  Value (+10) Value (+14) Value (+18) Value (+1c)
  0x10010000  0x00000005  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010020  0x0 00000   0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010040  0x          x00000000   0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
  0x10010060                          0x00000000  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
```

Last modified memory position
<Address, Value>

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400018
hi		0x00000000
lo		0x00000000

Last modified register
<Name, Number, Value>