



Departamento de
Sistemas e
Computação

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Disciplina: *Organização e Arquitetura de Computadores I*
Profa. *Joseana Macêdo Fachine Régis de Araújo*

Simulador MIPS

SIMULAÇÃO DE UMA MÁQUINA MIPS

Simulador: SPIM. É um simulador que executa programas em linguagem de montagem escritos para processadores que implementam a arquitetura MIPS-32 (correspondente ao processador descrito no livro "Digital Design and Computer Architecture", Harris/Harris).

Opções: PCSPIM ou QtSPIM.

```
PCSPIM
File Edit Options Help
PC = 00000000 EPC = 00000000 Cause = 00000000 BadAddr = 00000000
Status = 00000000 HI = 00000000 LO = 00000000
General Registers
$0 ($zero) = 00000000 $1 ($at) = 00000000 $2 ($v0) = 00000000 $3 ($v1) = 00000000
$4 ($a0) = 00000000 $5 ($a1) = 00000000 $6 ($a2) = 00000000 $7 ($a3) = 00000000
$8 ($t0) = 00000000 $9 ($t1) = 00000000 $10 ($t2) = 00000000 $11 ($t3) = 00000000
$12 ($t4) = 00000000 $13 ($t5) = 00000000 $14 ($t6) = 00000000 $15 ($t7) = 00000000
$16 ($s0) = 00000000 $17 ($s1) = 00000000 $18 ($s2) = 00000000 $19 ($s3) = 00000000
$20 ($s4) = 00000000 $21 ($s5) = 00000000 $22 ($s6) = 00000000 $23 ($s7) = 00000000
$24 ($s8) = 00000000 $25 ($s9) = 00000000 $26 ($s10) = 00000000 $27 ($s11) = 00000000
$28 ($s12) = 00000000 $29 ($s13) = 00000000 $30 ($s14) = 00000000 $31 ($s15) = 00000000
00000000: Data400000 lw $t0, 0($t0) ; 100: lw $t0, 0($t0) # 0
00000004: Data500000 addiu $t1, $t0, 4 ; 101: addiu $t1, $t0, 4 #
00000008: Data600004 addiu $t2, $t1, 4 ; 104: addiu $t2, $t1, 4 #
0000000c: Data700000 ori $t3, $t2, 0 ; 105: ori $t3, $t2, 0
00000010: Data800000 andi $t4, $t3, 0 ; 106: andi $t4, $t3, 0
00000014: Data900000 jal 0x00000000 [main] ; 107: jal main $l 0x0
DATA
[00000000]...[00000000] 0x00000000
STACK
[00000000] 0x00000000
All Rights Reserved.
GCC and Windows ports by David A. Carley (dcarley@wisc.edu).
Copyright 1997 by Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Location: C:\Program Files\PCSPIM\tray_handler
PC=00000000 EPC=00000000 Cause=00000000
```

```
QtSPIM
File Edit Options Help
PC = 00000000 EPC = 00000000 Cause = 00000000 BadAddr = 00000000
Status = 00000000 HI = 00000000 LO = 00000000
General Registers
$0 ($zero) = 00000000 $1 ($at) = 00000000 $2 ($v0) = 00000000 $3 ($v1) = 00000000
$4 ($a0) = 00000000 $5 ($a1) = 00000000 $6 ($a2) = 00000000 $7 ($a3) = 00000000
$8 ($t0) = 00000000 $9 ($t1) = 00000000 $10 ($t2) = 00000000 $11 ($t3) = 00000000
$12 ($t4) = 00000000 $13 ($t5) = 00000000 $14 ($t6) = 00000000 $15 ($t7) = 00000000
$16 ($s0) = 00000000 $17 ($s1) = 00000000 $18 ($s2) = 00000000 $19 ($s3) = 00000000
$20 ($s4) = 00000000 $21 ($s5) = 00000000 $22 ($s6) = 00000000 $23 ($s7) = 00000000
$24 ($s8) = 00000000 $25 ($s9) = 00000000 $26 ($s10) = 00000000 $27 ($s11) = 00000000
$28 ($s12) = 00000000 $29 ($s13) = 00000000 $30 ($s14) = 00000000 $31 ($s15) = 00000000
00000000: Data400000 lw $t0, 0($t0) ; 100: lw $t0, 0($t0) # 0
00000004: Data500000 addiu $t1, $t0, 4 ; 101: addiu $t1, $t0, 4 #
00000008: Data600004 addiu $t2, $t1, 4 ; 104: addiu $t2, $t1, 4 #
0000000c: Data700000 ori $t3, $t2, 0 ; 105: ori $t3, $t2, 0
00000010: Data800000 andi $t4, $t3, 0 ; 106: andi $t4, $t3, 0
00000014: Data900000 jal 0x00000000 [main] ; 107: jal main $l 0x0
DATA
[00000000]...[00000000] 0x00000000
STACK
[00000000] 0x00000000
All Rights Reserved.
GCC and Windows ports by David A. Carley (dcarley@wisc.edu).
Copyright 1997 by Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Location: C:\Program Files\QtSPIM\tray_handler
PC=00000000 EPC=00000000 Cause=00000000
```

Observações:

1. Janelas do simulador

O simulador apresenta quatro janelas (ver Figuras).

- a) A janela *Registers* mostra os valores dos registradores
- Registradores de uso geral (identificados pelo nome e pelo número) e outros registradores, a exemplo do PC.
- b) A janela *Text Segment* mostra o segmento de texto. Os endereços onde estão carregadas as instruções são seguidos pelos códigos dessas instruções em linguagem máquina (em hexadecimal) e em linguagem Assembly.
- c) A janela *Data Segment* mostra o segmento de dados. São mostrados os valores guardados em endereços de memória.
- d) Finalmente, na janela *Messages* são mostradas mensagens que incluem mensagens de erro ou exceções.

2. Para criar um programa:

- a) Abra o notepad;
- b) Escreva o seu programa;
- c) Salve o arquivo com extensão "asm" ou "s".

3. Para executar um programa:

- a) Abra o PCSPIM (ou QtSPIM);
- b) Selecione a opção *File* do menu;
- c) Selecione a opção *Open*;
- d) Escolha o arquivo com o código que quer executar;
- e) Verifique na janela *Messages* se o programa foi carregado com sucesso;
- f) Selecione a opção *Simulator*;
- g) Selecione a opção *GO*.

4. Para executar um programa passo a passo:

- a) Repita os passos de 3a) a 3f);
- b) Selecione a opção *Single Step*.

Importante:

- a) O "**main:**" indica ao assembler o início do código a ser carregado no segmento de texto..
- b) O programa deve terminar com a instrução `jr $ra` (ou `jr $31`)

Fontes:

1. **SPIM A MIPS32 Simulator**

Disponível em <http://pages.cs.wisc.edu/~larus/spim.html>. Último acesso em 05 de abril de 2017.

2. **Computer Fundamentals and MIPS assembly language programming**

Disponível em <http://www.cl.cam.ac.uk/~rdm34/fundamentals.html>. Último acesso em 05 de abril de 2017.

3. **Exercício 1 - Noções Básicas do Simulador**

Disponível em <http://www.ic.unicamp.br/~rodolfo/Cursos/mc723/1s2012/ex1.html>. Último acesso em 05 de abril de 2017.

4. **Arquitectura de Computadores**

Disponível em http://w3.ualg.pt/~mmadeira/ensino/ArgComp/Guias_v11.pdf. Último acesso em 05 de abril de 2017.

5. **Conjunto de Instruções**

Disponível em

http://www.univasf.edu.br/~leonardo.campos/Arquivos/Disciplinas/Micros_2008_2/Micros_Aula_03.pdf. Último acesso em 05 de abril de 2017.

6. **Conjunto de Instruções**

Disponível em

http://www.univasf.edu.br/~leonardo.campos/Arquivos/Disciplinas/Micros_2008_2/Micros_Aula_03.pdf. Último acesso em 05 de abril de 2017.