Simuladores MIPS

Chrístofer Caetano de Oliveira

SPIM

- Simulador “clássico” multi-plataformas open-source, suas versões mais novas são desenvolvidas em Qt.

- Download: http://spimsimulator.sourceforge.net/

- Documentação: http://pages.cs.wisc.edu/~larus/spim.html#information

MARS

- Simulador “moderno” multi-plataformas, desenvolvido em JAVA é open-source.

- Download: http://courses.missouristate.edu/KenVollmar/MARS/download.htm

- Documentação: http://courses.missouristate.edu/KenVollmar/MARS/Help/MarsHelpIntro.html

MARS

- A entrada e saída padrão do sistema é o console, através dele o usuário pode interagir com o programa lendo as mensagens e escrevendo dados.

- Para se utilizar o console é necessário o conhecimento de algumas “syscalls” (as básicas são compatíveis com o SPIM).

O que são *syscalls?*

- São serviços de sistema, principalmente para I/O.

- Para a chamada destes serviços é necessário indicar qual o serviço, através do registrador $v0. E passar seus parâmetros, geralmente pelos registradores $a0 .. $a3.

Syscalls básicas

Nome	ID	Parâmetros (Entradas)	Saídas
print_int	1	$a0: Inteiro à ser escrito	-
print_str	4	$a0: Ponteiro para string	-
read_int	5	-	$v0: Valor lido
Exit	10	-	-
print_char	11	$a0: Caractere à ser escrito	-
read_char	12	-	$v0: Caractere lido

Lista completa: http://courses.missouristate.edu/KenVollmar/MARS/Help/SyscallHelp.html

Exemplo de uso das syscalls

```
...
move $a0, $t0 # Valor à ser impresso no console
li $v0, 1 # ID syscall (1 = print_int)
syscall # Dispara syscall
...
```

Pseudo Instructions

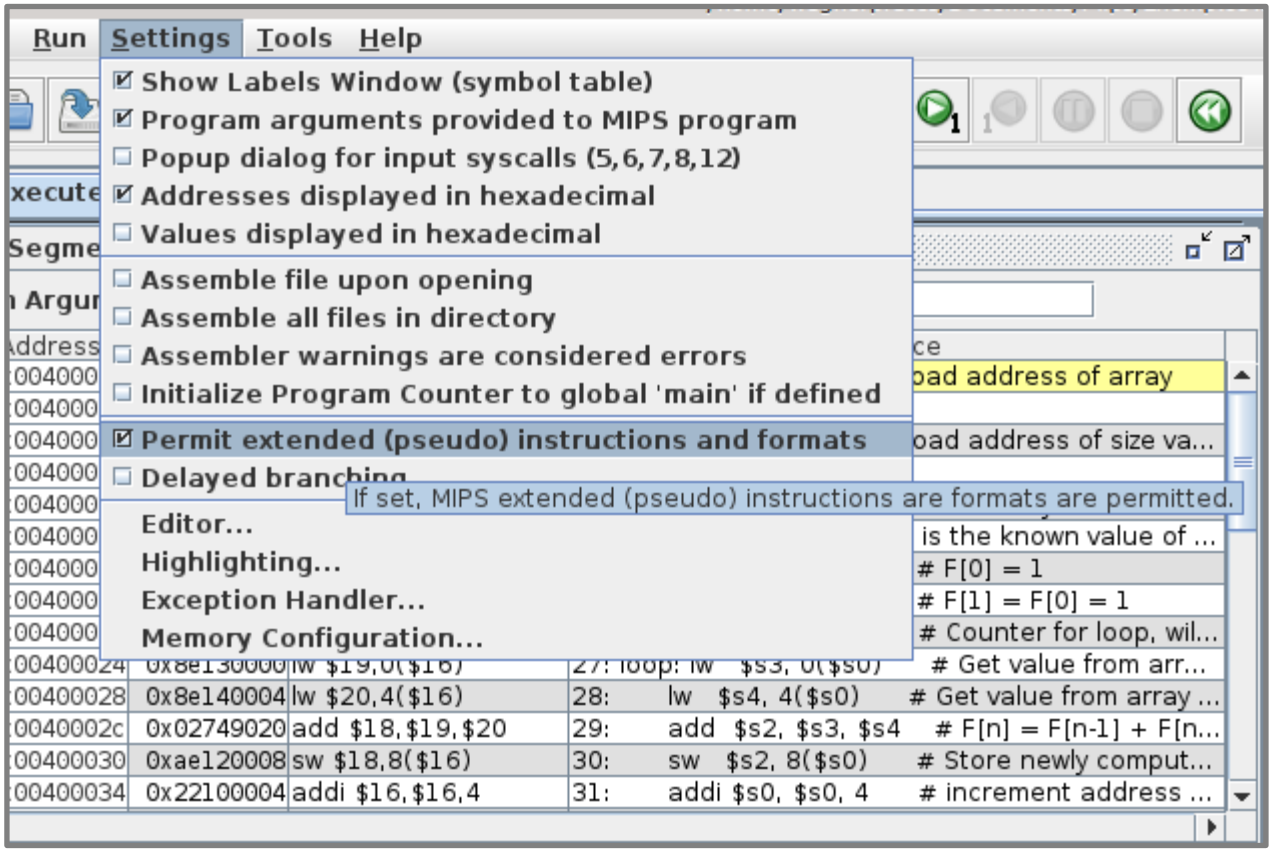
- São similares as macros, no momento da compilação são subtituídas por uma ou mais instruções reais.

- Ex:
  - li $t0, 0x80013002 *# Load Immediate*
    - lui $at, 0x8001 *# $at ← 0x80010000*
    - ori $t0, $at, 0x3002 *# $t0 ← 0x80013002*

Pseudo Instructions

- Podem-se adicionar novas pseudo instruções facilmente no MARS.
- Podem ser incluídas no arquivo PseudoOps.txt. Este arquivo esta dentro do MARS 4_2.jar (abrir como .ZIP).
- Instruções no cabeçalho do arquivo, podem-se utilizar as pseudo instruções já inclusas como exemplo.

Habilitando Pseudo-Instructions



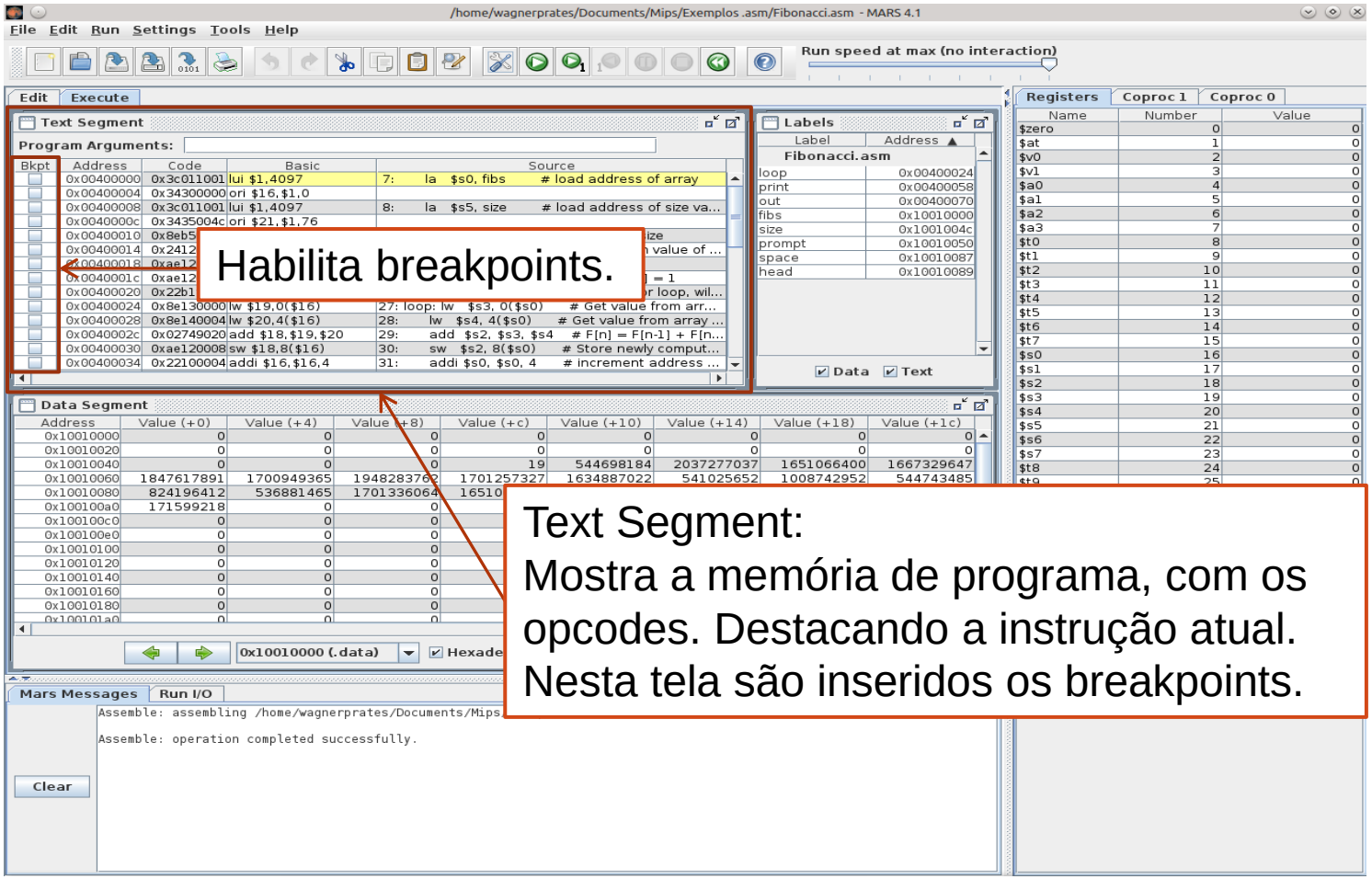
Iniciando a simulação

- Para se realizar a simulação é necessário ‘montar’ o programa, para isso deve-se clicar no ícone:

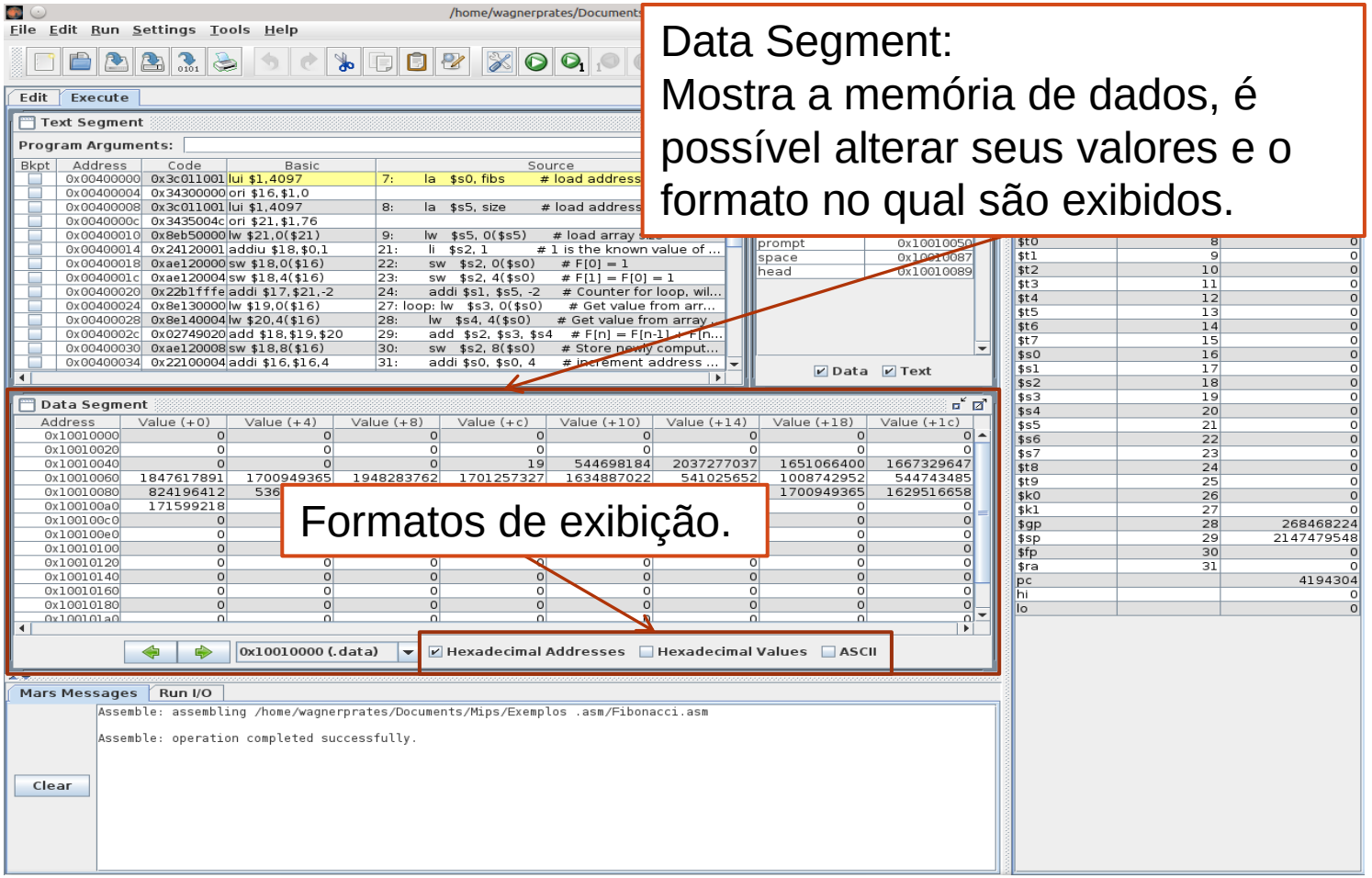
- Se o programa contiver pseudoinstruções é necessário habilitá-las.

- O Mars NÃO tem suporte a pipeline, porém pode simular o atraso nos saltos.

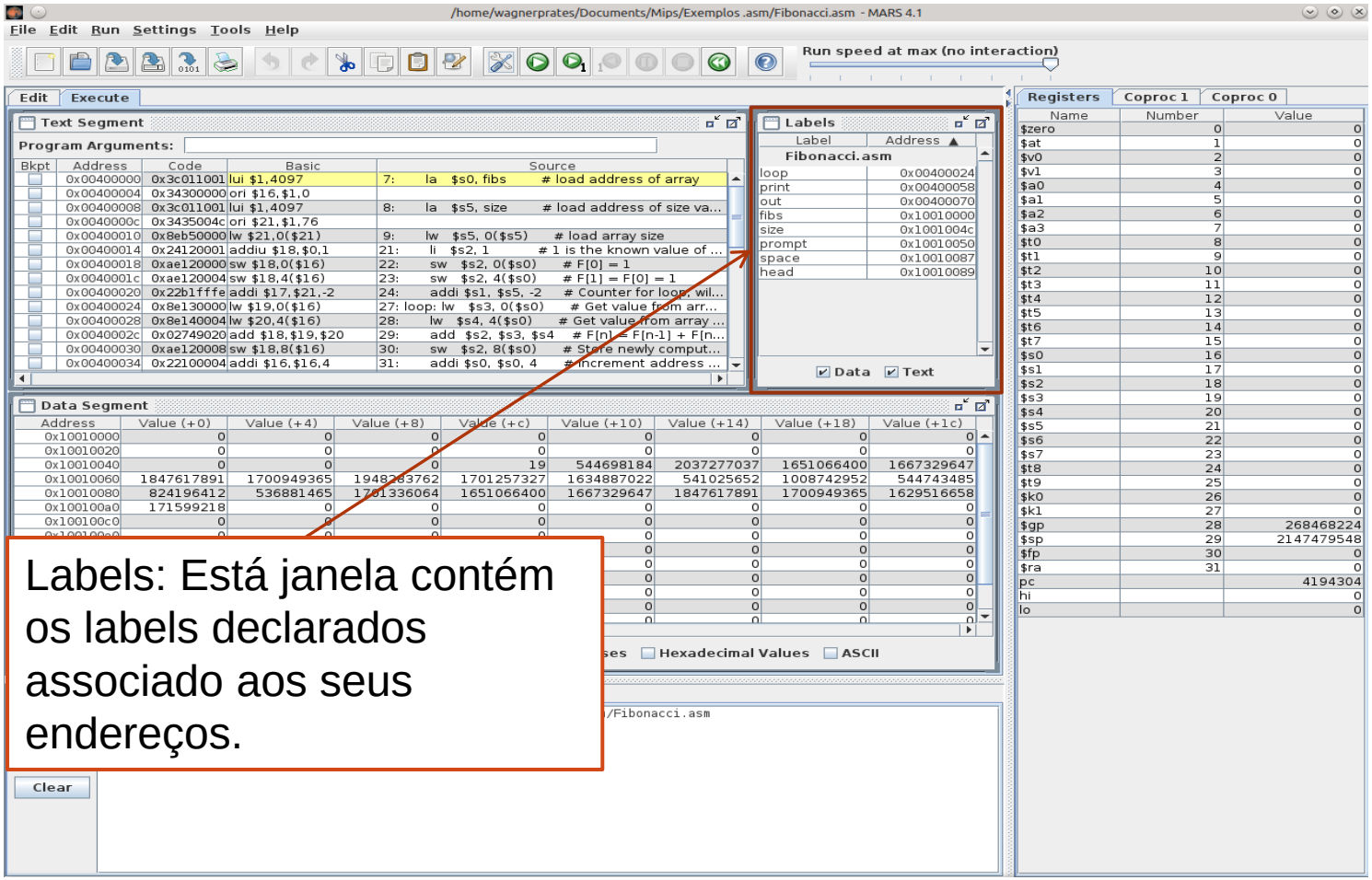
Interface do Mars



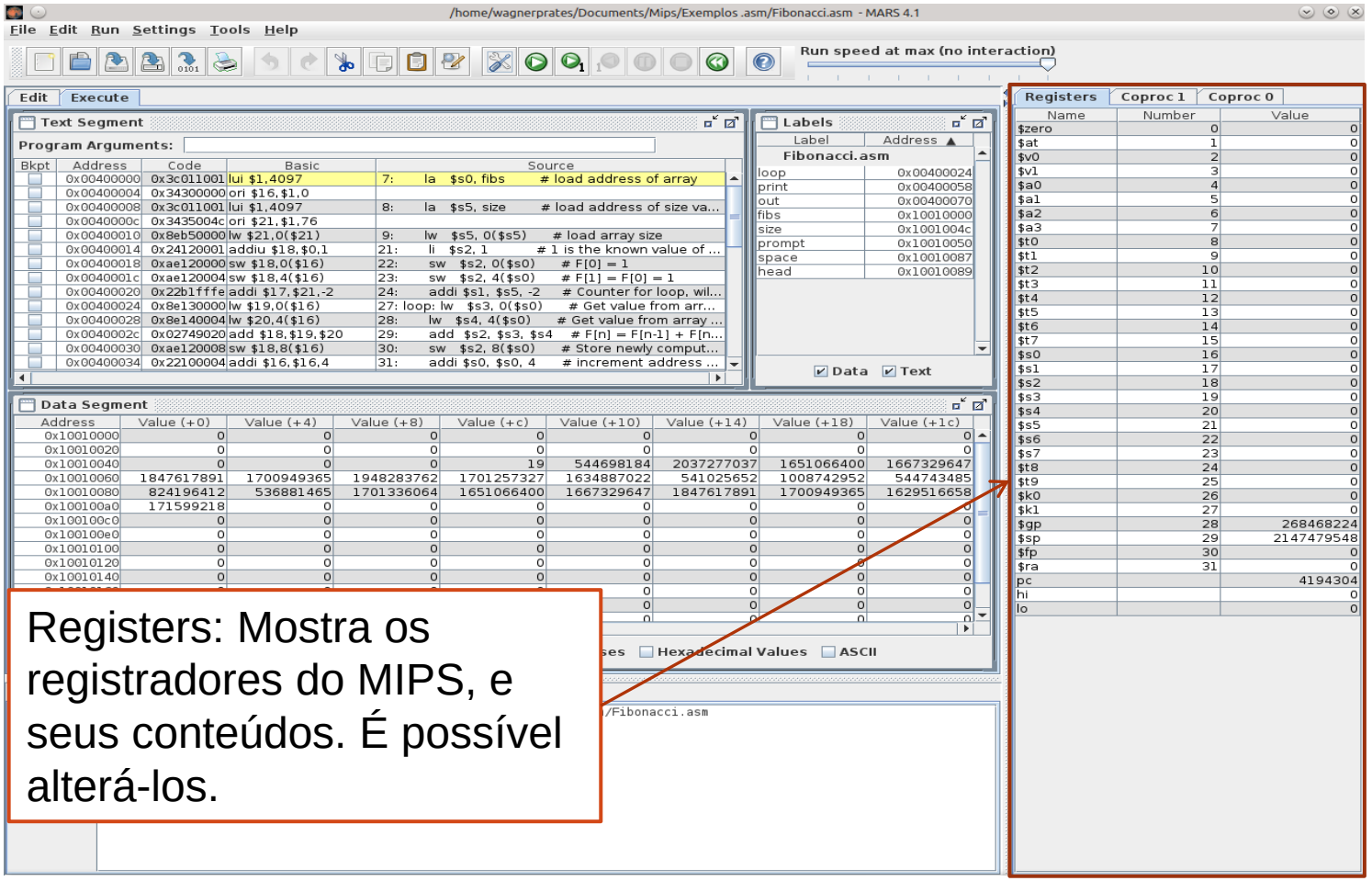
Interface do Mars



Interface do Mars

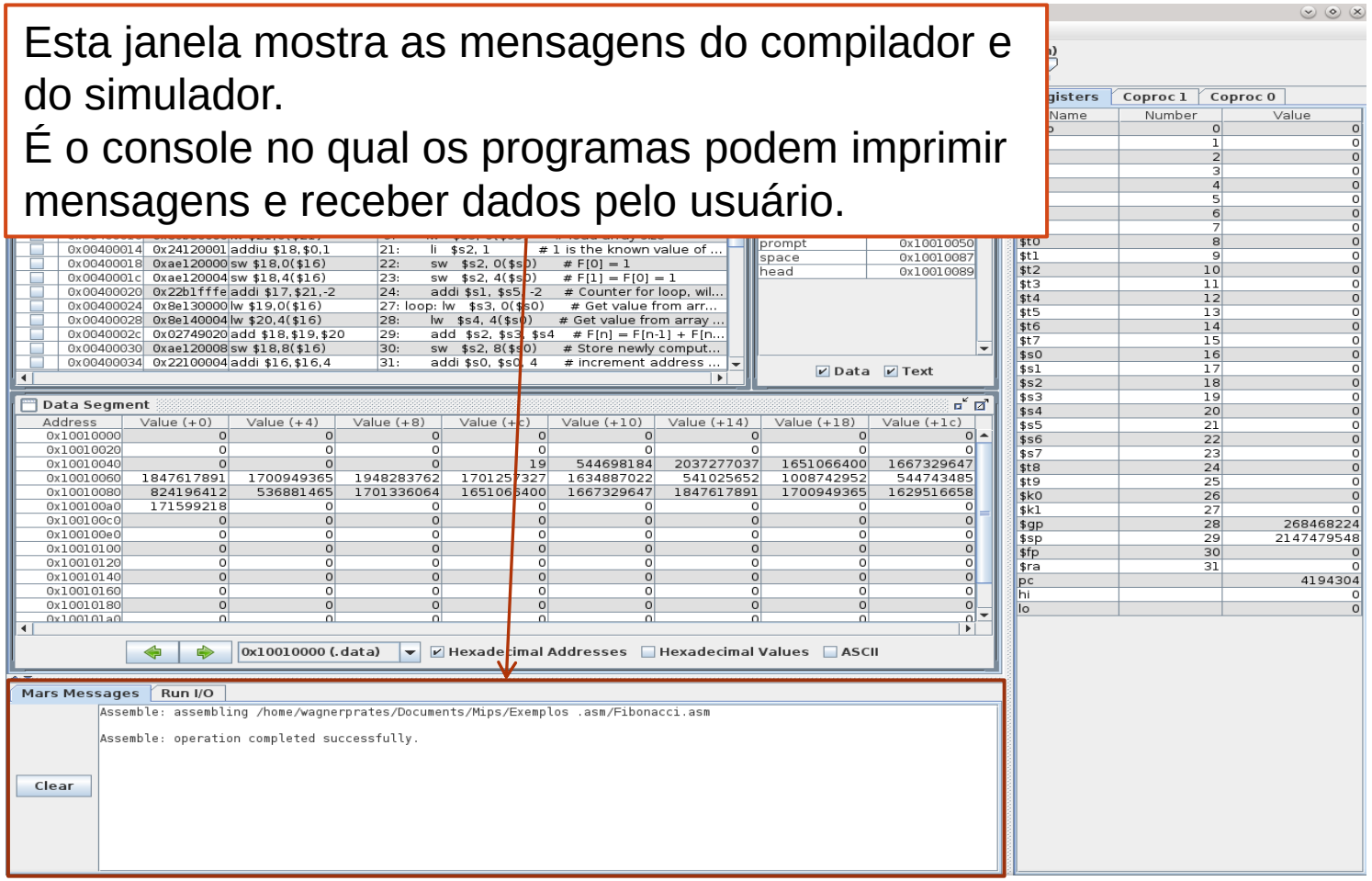


Interface do Mars



Interface do Mars

Esta janela mostra as mensagens do compilador e do simulador.
É o console no qual os programas podem imprimir mensagens e receber dados pelo usuário.



Simulação

Executa o programa até o fim ou próximo breakpoint.

Executa a instrução atual.

Desfaz última instrução.

Reinicia o simulador, registradores e memória.

Simulação

A barra deslizante 

controla a velocidade de execução.

Permitindo assim “ver a ação” ao invés do resultado final somente se a velocidade for reduzida.

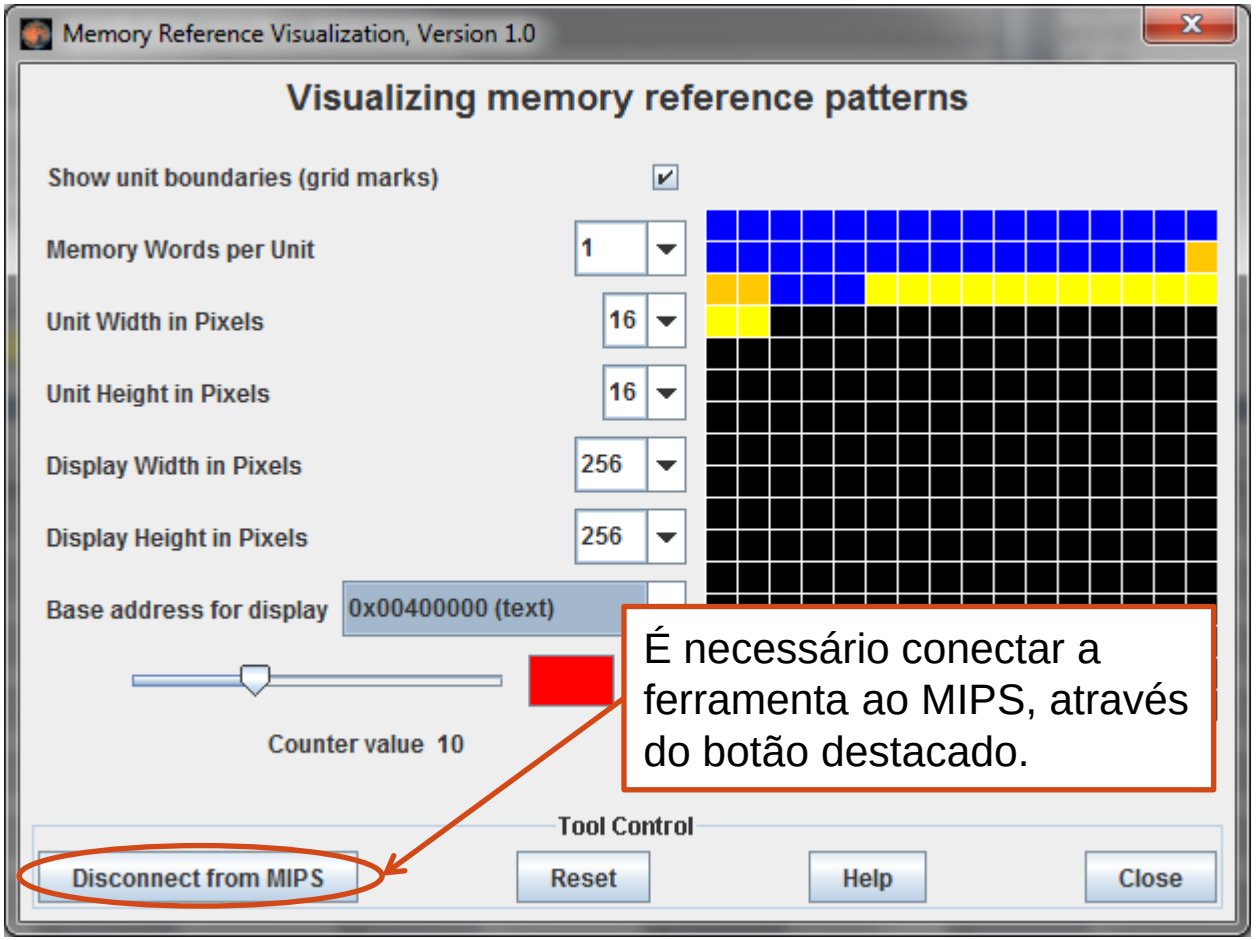
Ferramentas do MARS

- O Mars possuí algumas ferramentas já inclusas além de um modelo para ajudar na geração de novas ferramentas.

- Estas ferramentas possuem as mais diversas funcionalidades, como mostrar o padrão de acesso à memória de dados ou de programa.

- Podem ser acessadas através do menu *Tools*.

Ferramentas do MARS



Organização do programa

- Os programas são organizados em seções.
  - .text: Esta seção contém as instruções em ASM.
  - .data: Esta seção contém os dados, estes são inicializados na memória de dados.

    Ex: str_1: .asciiz “Simulador MARS\n”

Tipos de dados

Tipo	Descrição
.word	Tipo de dados genérico de 32 bits.
.half	Tipo de dados genérico de 16 bits.
.byte	Tipo de dados genérico de 8 bits.
.ascii	Sequência de caractéres: Ex: “ABCD\n”.
.asciiz	String com terminação nula: Ex: “ABCD\n”.
.float	Ponto flutuante de precisão símples.
.double	Ponto flutuante de precisão dupla.
.space	Reserva espaço de ‘n’ bytes.
.align	Alinha o próximo dado para endereço múltiplo de 2^n

Exemplos:

```
prompt: .asciiz “Digite sua idade: ”
.align 2
vetor_1: .word 1, 2, 3, 4
```

Chamadas de função

- Os programas ASM são sequenciais, por isso o programador deve tomar cuidado com os saltos (jumps).
- Os parâmetros podem ser passados através de registradores, ou através da pilha (stack).
- Funções recursivas devem salvar seus endereços de retorno ($ra), antes de chamar outra função.

Chamadas de função

- Para chamar uma função devem-se utilizar as instruções de saltos com ‘link’, é comum utilizar a JAL, porém é possível utilizar-se também as condicionais BGEZAL ou BLTZAL.

- Para o retorno utiliza-se a instrução “jr”, usualmente associada ao registrador $ra.

Chamadas de função - Exemplo

```
main:
        ...
        move $a0, $t0 # some parameter
        jal func  # call func and writes the return
                  # address on $ra
        ...
func:
         ... Func statements ...
        jr $ra              # return to caller
        ...
```

Exemplos

- 01 Console.asm:

  Este exemplo mostra como se utilizar o console.

- O que faz:
  - Lê dois números informados pelo usuário ($s1 e $s2)
  - Troca-os de registrador (swap)
  - Imprime no console

Exemplos

- 02 Factorial.asm:

  Este exemplo demonstra o uso de funções recursivas e pilha (stack).

- O que faz:
  - Calcula o fatorial do número informado pelo usuário ($s1).
  - Imprime no console o resultado ($s0).

Exemplos

- 03 Fibonnaci.asm:

  Este exemplo gera uma série de Fibonnaci

- O que faz:
  - Calcula os 12 primeiros itens da série de Fibonacci.
  - Imprime a série no console.

Links Úteis

- Site com instruções ASM do MIPS: http://en.wikipedia.org/wiki/MIPS_architecture

- Site com vários exemplos de MIPS Assembly: http://pt.scribd.com/doc/50236313/10/MIPS-using-pseudo-instructions

Exercício

- Modificar Fibonnaci.asm, indicado no link abaixo:
  - http://courses.missouristate.edu/KenVollmar/MARS/Fibonacci.asm

- De modo que o usuário indique o tamanho da série de 1 até 12.
  - Deve mostrar uma mensagem de erro para tamanho fora deste intervalo.

Obrigado.