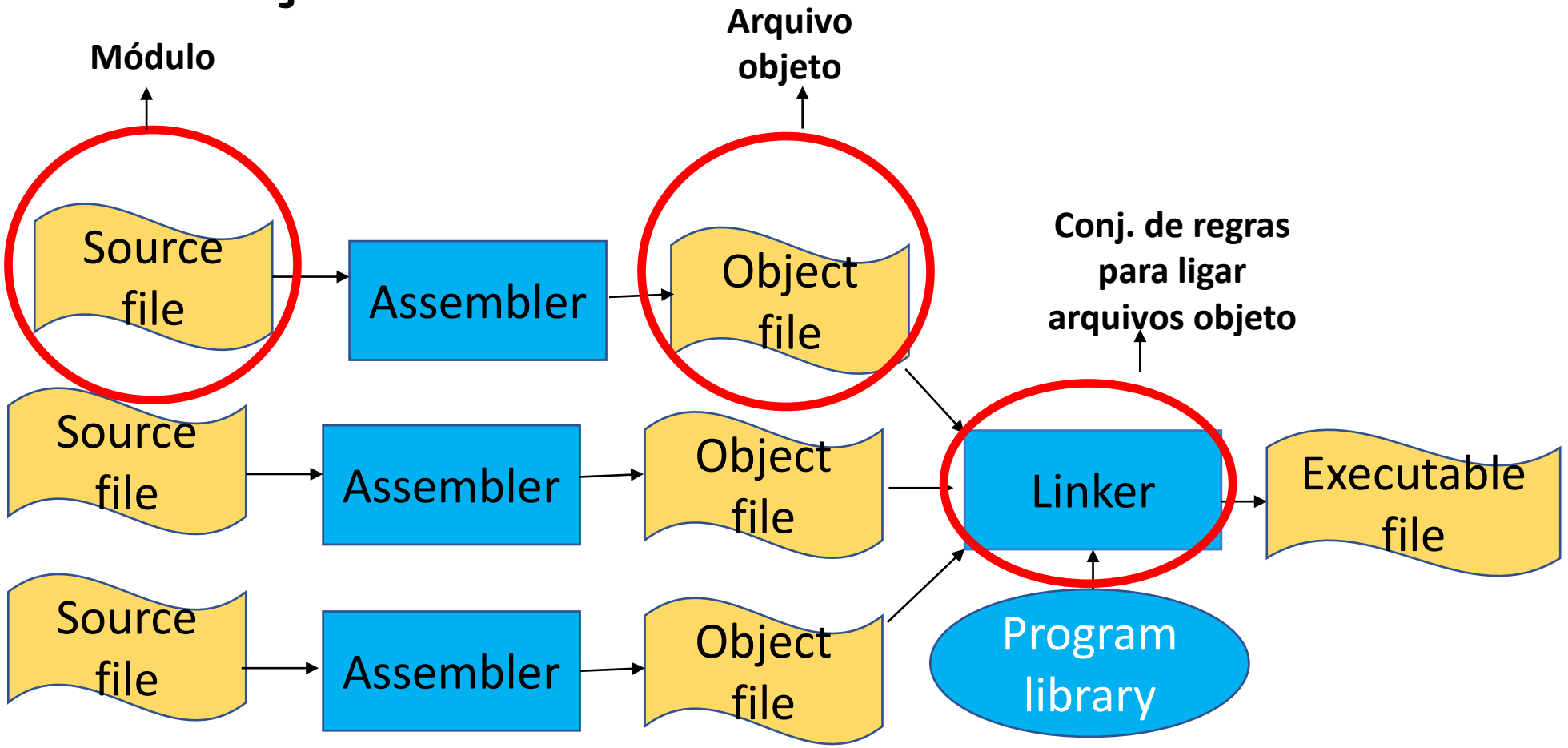
Programação em Baixo Nível (Registradores)

Porfª Mª Cleane Nascimento

Combinação de módulos menores



CONJUNTO DE INSTRUÇÕES MIPS

- Instrução é uma palavra da linguagem;
- ISA MIPS → Instruções possuem até 3 operandos;
- E o que é ISA?
- Intruction Set Architecture (Conjunto de Instruções da Arquitetura);
- No MIPS os operandos das instruções são chamados registradores ($);
- Há 32 registradores de 32 bits;
- Cada registrador possui o símbolo $ antecedendo seu nome.

Load e Store

- **Load:** instrução de movimentação de dados da memória para o registrador;

Operação de **leitura da memória**.

- **Store:** instrução de movimentação de dados do registrador para a memória;

Operação **escrita na memória**.

- **Move:** instrução para passar o conteúdo de um registrador para outro;

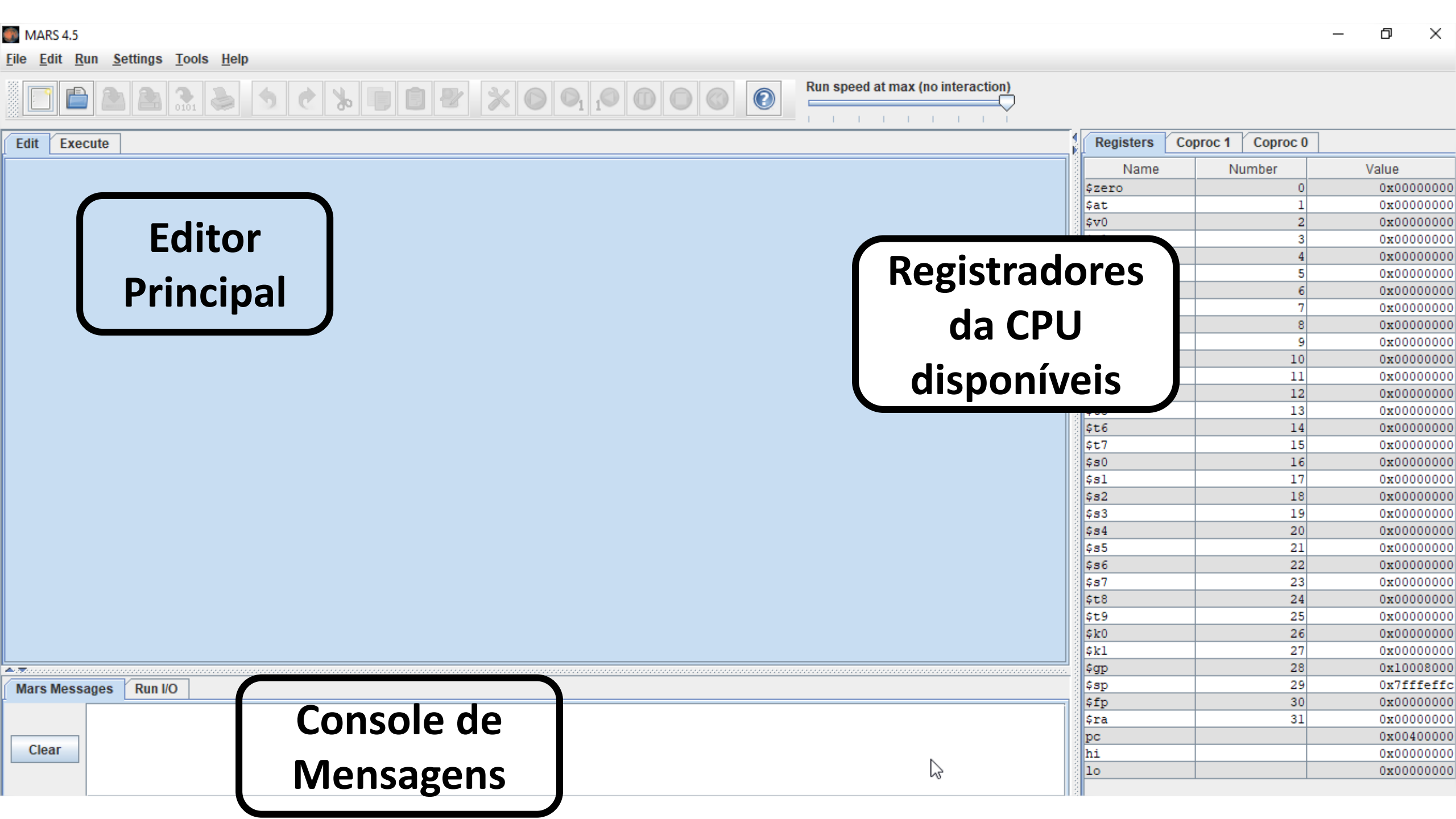
Memória RAM não é envolvida.

E qual programa utilizaremos?

- MARS – Programar e Depurar (debug);

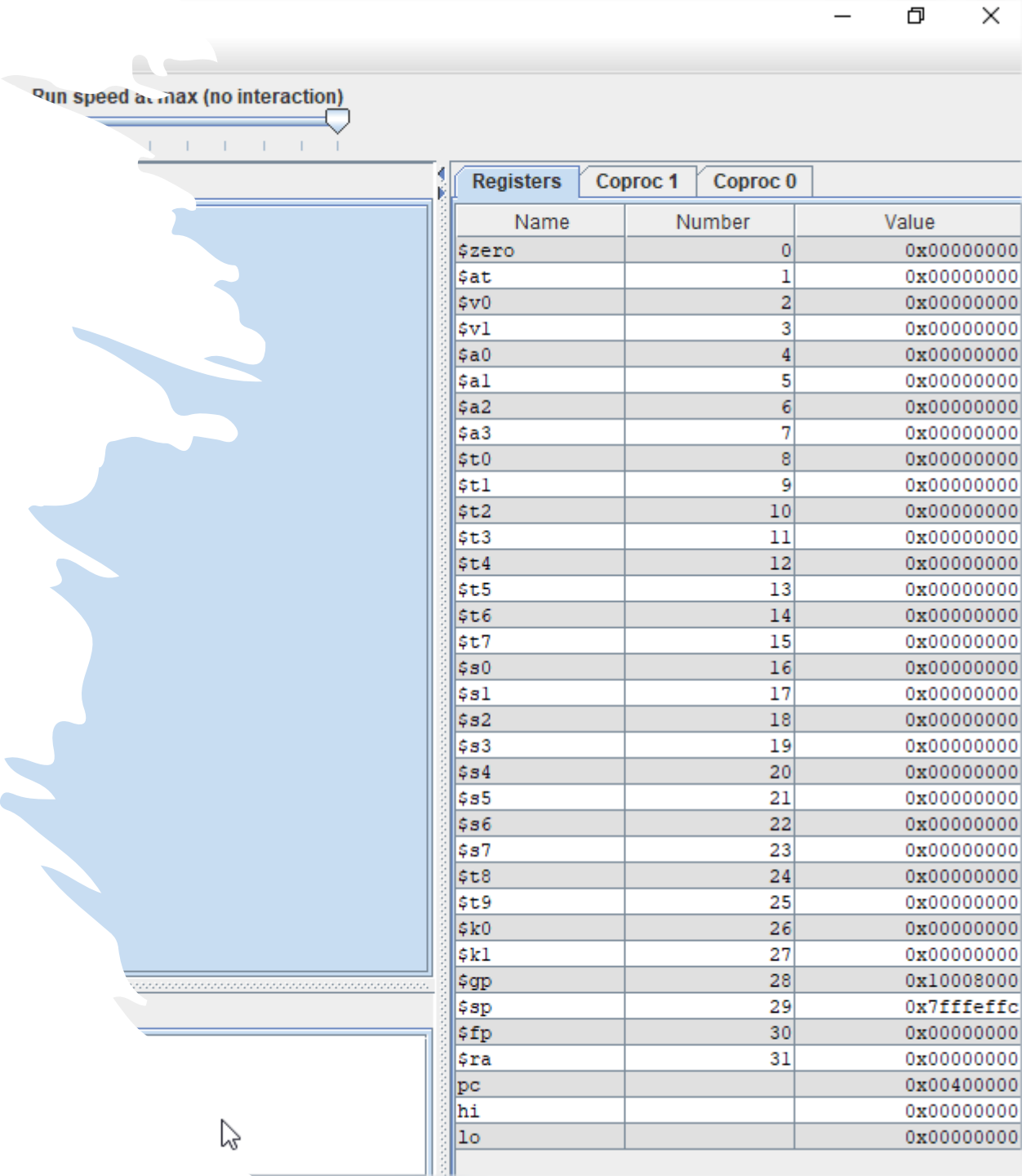
- É um arquivo .jar, portanto é necessário ter o JAVA instalado.

(O link com os arquivos de instalação se encontram no SIGAA).


MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Editor Principal
Registradores da CPU disponíveis
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
3 0x00000000
4 0x00000000
5 0x00000000
6 0x00000000
7 0x00000000
8 0x00000000
9 0x00000000
10 0x00000000
11 0x00000000
12 0x00000000
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 Messages Run I/O
Clear
Console de Mensagens

A princípio não usaremos:

- $at – assember temporary;
- $k0 e $kI – registradores do kernel;
- $gp – registradores de valores globais;
- $sp – stack pointer (aponta para o início do stack e muda progressivamente);
- $fp – frame pointer (aponta para o início da pilha e não muda até que a função seja executada).



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

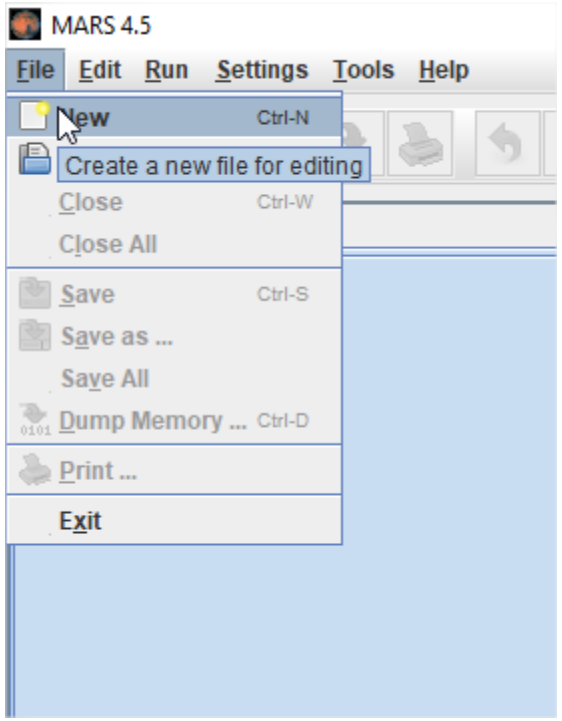
Comando li $v0

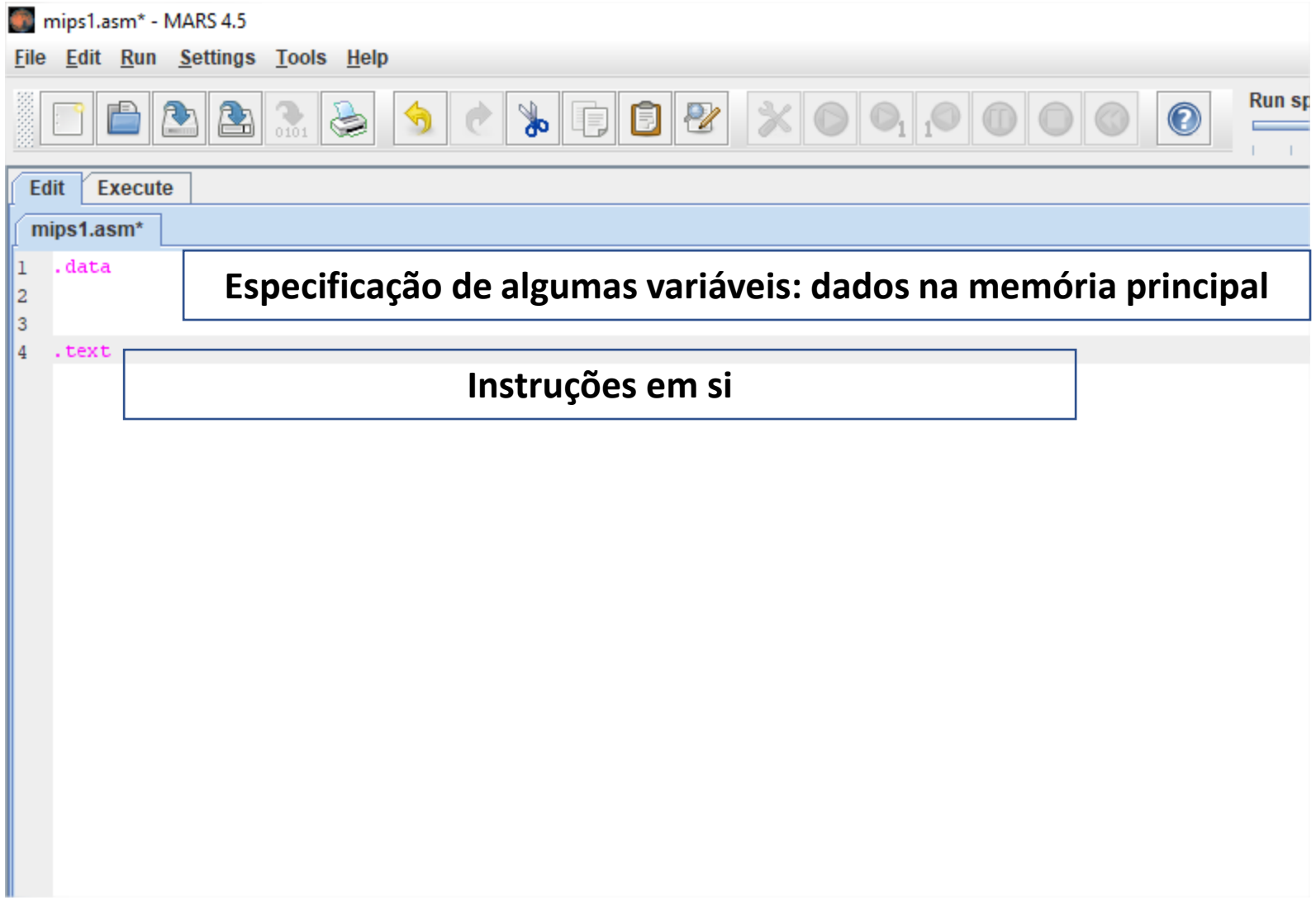
Comando	Significado
Li $v0, 1	Imprimir inteiro
Li $v0, 2	Imprimir float
Li $v0, 3	Imprimir double
Li $v0, 4	Imprimir String ou char
Li $v0, 5	Ler inteiro
Li $v0, 6	Ler float
Li $v0, 7	Ler double
Li $v0, 8	Ler String ou char
Li $v0, 10	Encerrar programa principal

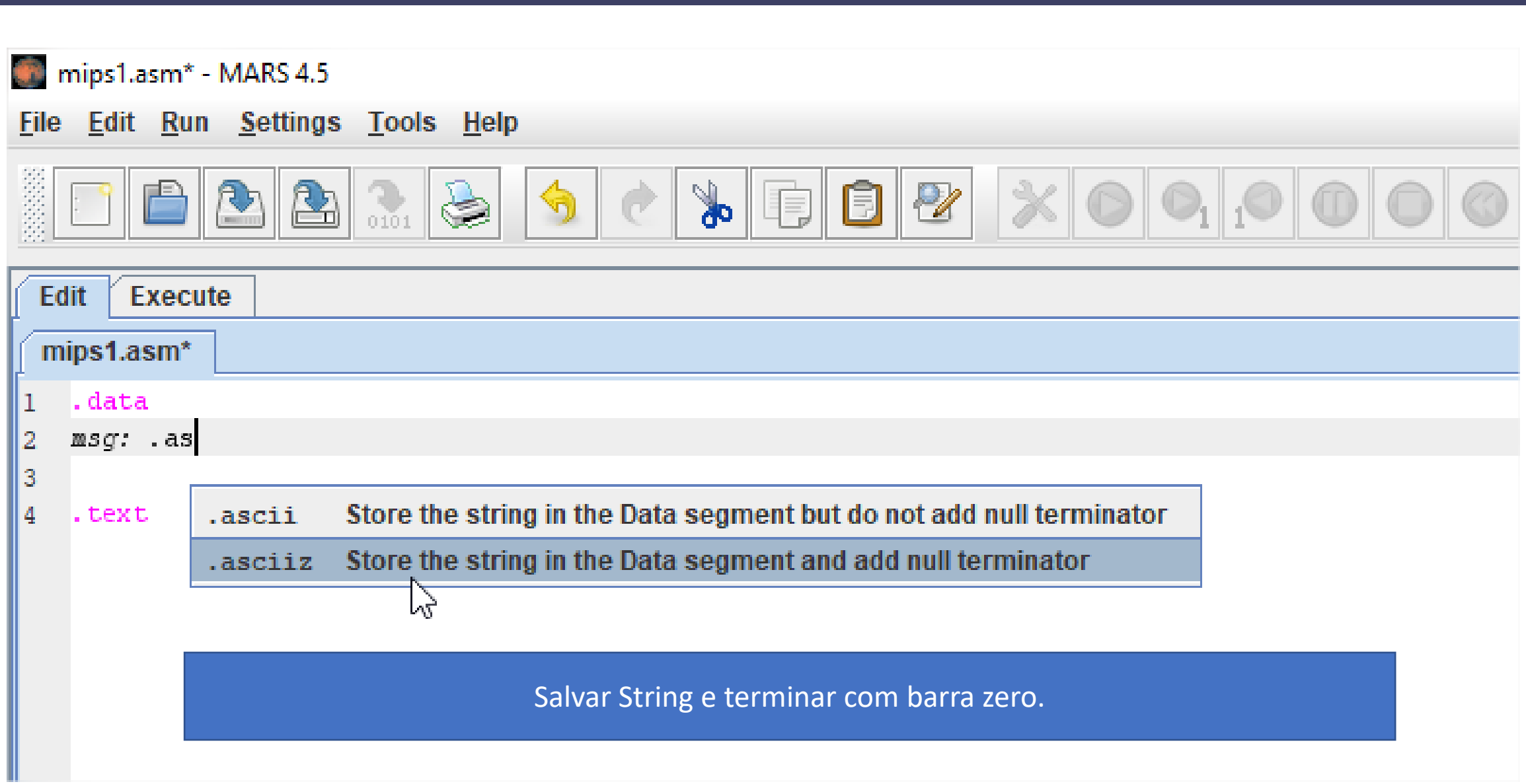
Atribuir valor inteiro à um registrador é usada a instrução li

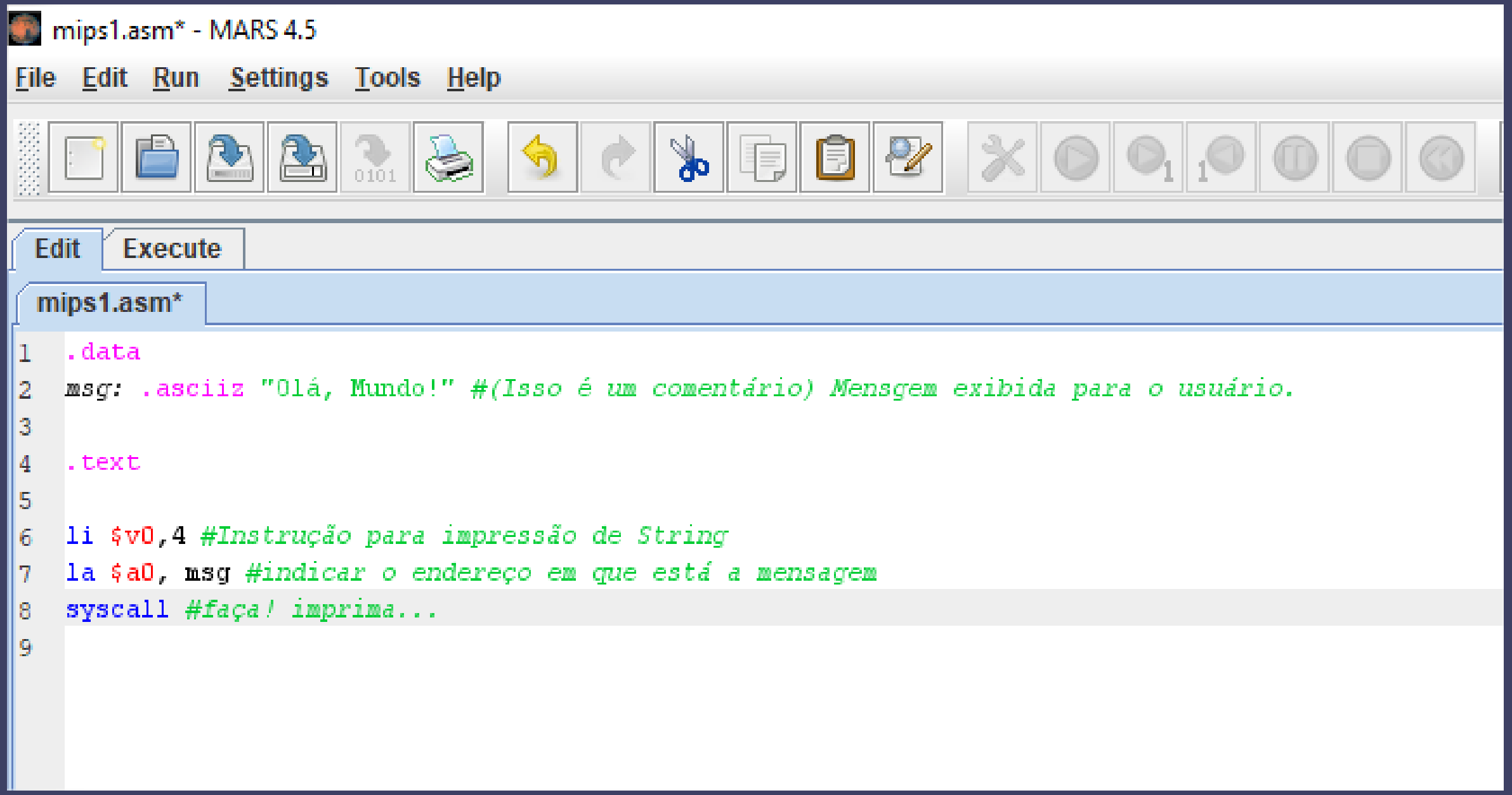
Códigos de 1 a 4 – **impressão;**
Códigos de 5 a 8 – **Leitura.**

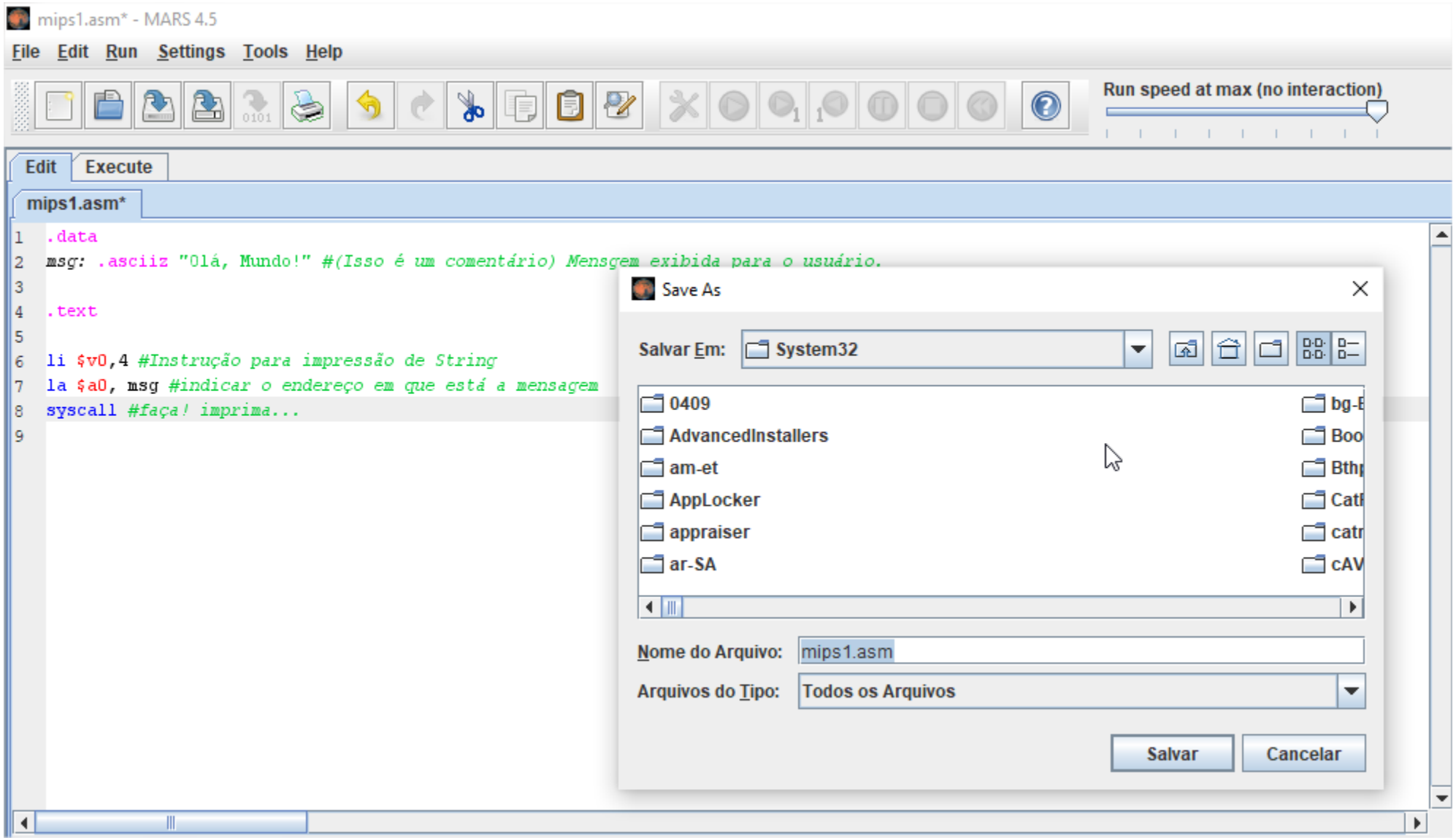
Nosso primeiro programa...

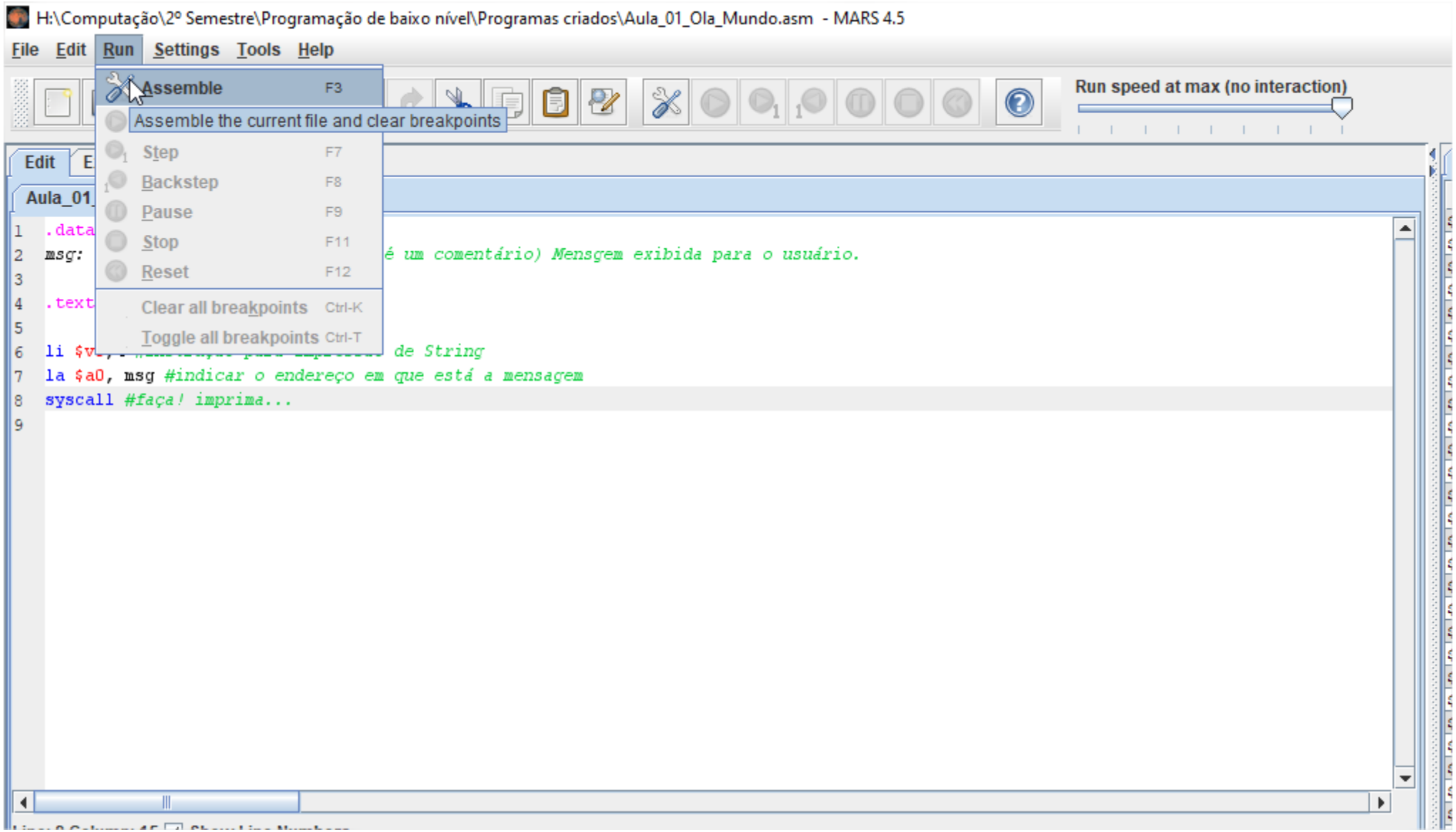





mips1.asm* - MARS 4.5
File Edit Run Settings Tools Help
Edit Execute
mips1.asm*
1 .data
2 msg: .as
3
4 .text
.ascii Store the string in the Data segment but do not add null terminator
.asciiz Store the string in the Data segment and add null terminator
Salvar String e terminar com barra zero.


mips1.asm* - MARS 4.5
File Edit Run Settings Tools Help
Edit Execute
mips1.asm*
1 .data
2 msg: .asciiz "Olá, Mundo!" #(Isso é um comentário) Mensgem exibida para o usuário.
3
4 .text
5
6 li $v0,4 #Instrução para impressão de String
7 la $a0, msg #indicar o endereço em que está a mensagem
8 syscall #faça! imprima...
9


mips1.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips1.asm*
1 .data
2 msg: .asciiz "Olá, Mundo!" #(Isso é um comentário) Mensgem exibida para o usuário.
3
4 .text
5
6 li $v0,4 #Instrução para impressão de String
7 la $a0, msg #indicar o endereço em que está a mensagem
8 syscall #faça! imprima...
9
Save As
Salvar Em: System32
0409
AdvancedInstallers
am-et
AppLocker
appraiser
ar-SA
Nome do Arquivo: mips1.asm
Arquivos do Tipo: Todos os Arquivos
Salvar
Cancelar


H:\Computação\2º Semestre\Programação de baixo nível\Programas criados\Aula_01_Ola_Mundo.asm - MARS 4.5
File Edit Run Settings Tools Help
Assemble F3
Assemble the current file and clear breakpoints
Step F7
Backstep F8
Pause F9
Stop F11
Reset F12
Clear all breakpoints Ctrl-K
Toggle all breakpoints Ctrl-T
Run speed at max (no interaction)
Edit
Aula_01
1 .data
2 msg:
3
4 .text
5
6 li $v
7 la $a0, msg #indicar o endereço em que está a mensagem
8 syscall #faça! imprima...
9
é um comentário) Mensgem exibida para o usuário.
de String

Referências

MANZANO, J. A. N. G. **Fundamentos em Programação Assembly**. 1. ed. Editora Érica, 2004.

HENNESSY, J. L.; PATTERSON D. A. **Organização e Projeto de Computadores:** A Interface Hardware/Software. 4. ed. Editora Elsevier, 2013.