

Disciplina: Fundamentos e Arquitetura de Computadores (FAC)

Professor: Tiago Alves

Trabalho 3 - Fundamentos de Arquitetura de Computadores

Alunos	Matrícula	Sistema Operacional
Djorkaeff Alexandre	16/0026822	macOS Sierra
Daniel Maike	16/0117003	Windows 10

O ambiente de desenvolvimento utilizado pelos dois alunos foi o MARS v4.5.

Instruções de uso:

Para simular os arquivos de resposta para as questões deste trabalho foi usado o aplicativo Mars versão 4.5 que pode ser baixado através de (<http://courses.missouristate.edu/kenvollmar/mars/>).

Após iniciar o aplicativo Mars, devemos abrir o arquivo de resposta da questão, na pasta zipada entregue neste trabalho temos um arquivo, com o nome " *floatFac.asm* ", estes arquivos são gerados pelo Mars e podem ser lidos pelo mesmo.

Após abrir um dos arquivos será lhe apresentado o código MIPS do mesmo.

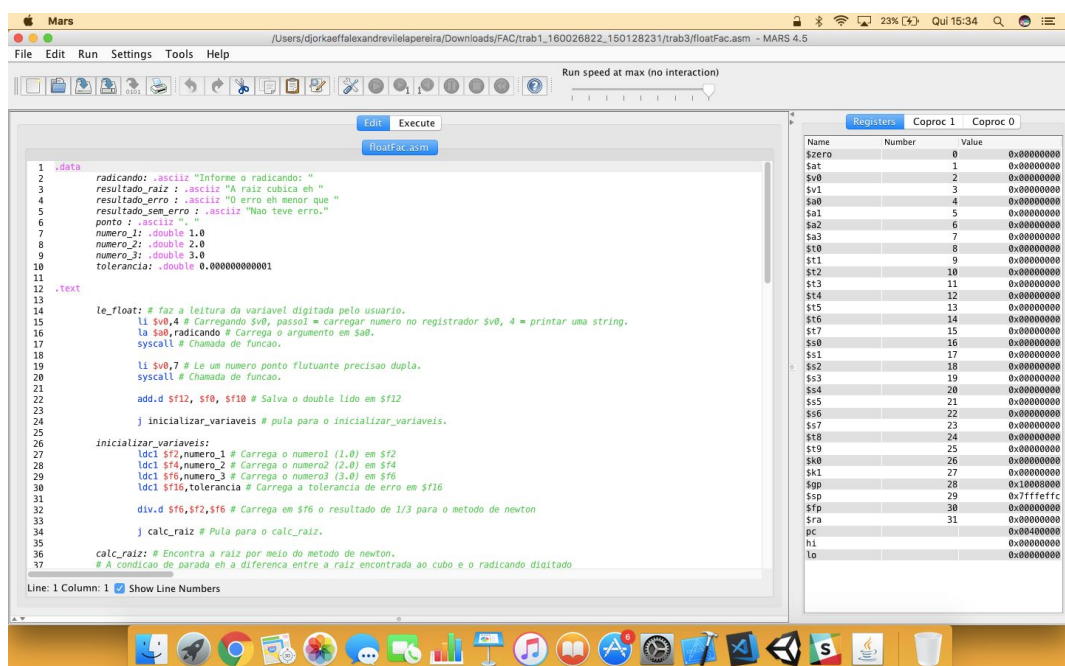


Figura 1 - Abertura de arquivo .asm no simulador Mars v4.5 código "*floatFac.asm*"

Após aberto o arquivo .asm devemos montá-lo (simulação), para isso selecionamos no menu superior a opção Run > Assemble.

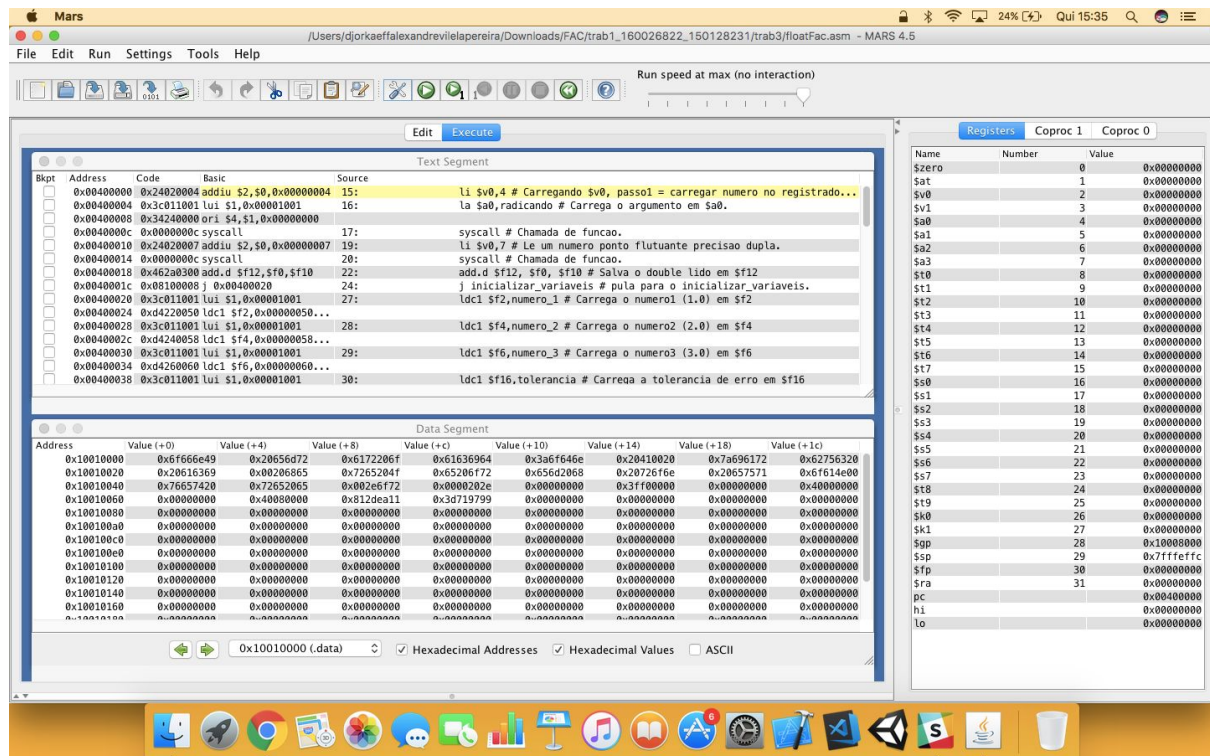


Figura 2 - Lista de comandos executados com o arquivo presente em *Text Segment*

Após clicar na opção Assemble presente na aba Run do menu superior, será apresentada a tela de Text Segment onde estarão presentes as instruções que serão realizadas por este arquivo sobre a memória simulada.



Deve-se clicar no botão indicado ao lado dentro do aplicativo Mars para que as instruções sejam simuladas.

O radicando utilizado no programa para gerar a saída a seguir nos registradores foi o valor 8.

Registers			Coproc 1	Coproc 0
Name	Float	Double		
\$f0	0x00000000	0x4020000000000000		
\$f1	0x40200000			
\$f2	0x00000000	0x4000000000000000		
\$f3	0x40000000			
\$f4	0x00000000	0x4000000000000000		
\$f5	0x40000000			
\$f6	0x55555555	0x3fd5555555555555		
\$f7	0x3fd55555			
\$f8	0x00006a19	0x4010000000006a19		
\$f9	0x40100000			
\$f10	0x0000d432	0x401000000000d432		
\$f11	0x40100000			
\$f12	0x00000000	0x4000000000000000		
\$f13	0x40000000			
\$f14	0x00000000	0x4018000000000000		
\$f15	0x40180000			
\$f16	0x812dea11	0x3d719799812dea11		
\$f17	0x3d719799			
\$f18	0x00000000	0x4020000000000000		
\$f19	0x40200000			
\$f20	0x00000000	0x0000000000000000		
\$f21	0x00000000			
\$f22	0x00000000	0x0000000000000000		
\$f23	0x00000000			
\$f24	0x00000000	0x0000000000000000		
\$f25	0x00000000			
\$f26	0x00000000	0x0000000000000000		
\$f27	0x00000000			
\$f28	0x00000000	0x0000000000000000		
\$f29	0x00000000			
\$f30	0x00000000	0x0000000000000000		
\$f31	0x00000000			

Condition Flags			
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7

Figura 3 - Saídas produzidas nos registradores pelo código

Após clicar no botão que simula as instruções são geradas saídas nos registradores, os registradores.

Funcionamento do código:

A raiz cúbica é calculada utilizando o método de Newton para que seja possível encontrar um resultado exato ou com erro menor que 10^{-12} , quando o usuário digitar o radicando, iremos fazer o cálculo da raiz entrando em um loop no método de Newton, a condição de parada é o erro ser menor que 10^{-12} ou ser nulo, quando houver erro será impresso juntamente com o resultado da raiz, se não houver erro será impressa uma mensagem indicando que não houve erro. Para utilização de mais casas decimais o projeto foi feito em double.

```
calc_raiz: # Encontra a raiz por meio do metodo de newton.  
# A condicao de parada eh a diferenca entre a raiz encontrada ao cubo e o radicando digitado  
# ser menor do que 10^-12.  
# A diferenca eh calculada em calc_erro.
```

```
mul.d $f8,$f2,$f4 # 2 * f2  
mul.d $f10,$f2,$f2 # f2 * f2  
div.d $f14,$f12,$f10 # f12 / (f2*f2)  
  
add.d $f14,$f14,$f8 # (2*f2) + (f12 / (f2*f2))  
  
mul.d $f2,$f6,$f14 # 1/3 * [(2*f2) + (f12 / (f2*f2))]  
  
# Metodo de newton -> Xi+1 = 1/3 * [(2*Xi) + (C / (Xi*Xi))]  
  
j calc_erro # Pula para o calc_erro.
```

```
calc_erro: # Faz o calculo do erro entre a raiz encontrada ao cubo e o radicando digitado.
```

```
mul.d $f18,$f2,$f2 # Eleva ao quadrado a raiz encontrada.  
mul.d $f18,$f18,$f2 # Eleva ao cubo a raiz encontrada.  
  
sub.d $f20,$f18,$f12 # subtrai a raiz encontrada ao cubo e o radicando.  
  
abs.d $f20,$f20 # faz o modulo de $f20.  
  
c.lt.d $f16,$f20 # compara se $f16 eh menor que $f20, se for falso marca a flag 0.  
  
bc1t calc_raiz # se for true a condição, ele volta pro calc_raiz  
  
c.eq.d 2,$f18,$f12  
  
bc1t 2,imprime_saida_sem_erro  
  
j imprime_saida # pula para o label imprime_saida
```

Limitações conhecidas:

Não existem limitações conhecidas diante dos limites de entrada e saída que foram ditados para o trabalho.

Bibliografia:

<https://educacao.uol.com.br/disciplinas/matematica/raiz-cubica-como-calcular-na-mao.htm>