

## **Relatório da implementação**

### **1. Descrição do problema:**

Foi implementado um programa em linguagem C que simula as operações do MIPS. Primeiramente foi criado a partir do MARS um .ASM e carregados os arquivos .data e .text para ser lido no programa a ser escrito. Dentro do programa foi armazenado os valores em um array mem em seus respectivos endereços de memória.

### **2. Descrição das funções:**

void fetch():

Pega a instrução apontada pelo PC e armazena em ri acrescentando 4 bytes.

void decode():

Analisa a instrução do processamento.

void execute():

Executa as instruções decodificadas pela função decode.

void step():

Executa uma instrução de cada vez.

void run():

Executa todas as instruções.

void dump\_mem(inicio, fim, formato):

Lista os valores da memória, passando como parâmetro, em bytes, o início, o tamanho a ser lido e o formato a exibir (se em decimal ou hexadecimal).

void dump\_reg:

Lista os registradores e seus respectivos valores armazenados.

As instruções foram implementadas dentro do execute():

### **3. Testes e Resultados**

O código do trabalho anterior foi corrigido e utilizado na main deste programa.

Foi desenvolvida todas as funções descritas acima e executadas.

Para realização dos testes foi utilizado dois códigos ASM. Um dado pelo roteiro, onde retorna números primos, e o outro que retorna strings informando resultados de testes das funções da instrução.

Primeiramente foi gerado dois arquivos para cada ASM e colocado nas respectivas pastas. O mesmo código foi usado para testá-las. Um chamado "text.bin" e outro chamado "data.bin". E carregamos o array de memória (mem[size]).

Imprimido-se um contador de passos, o valor do registrador apontado, a função dump\_reg() em hexadecimal e a função step() conseguimos debugar comparando os valores do banco de registradores mostrados no MARS ao rodar o mesmo programa e analisando um por um os registradores.

Segue-se os valores obtidos:

Programa imprimindo código que retorna números primos:

```
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S: ~/Documentos/OAC/Trab2_OAC_150115326/primos
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$ ./naib
Sucesso na leitura do data.binSucesso na leitura do text.bin

Print dos primeiros 7 registradores:
men[0] = 20082000
men[1] = 20092020
men[2] = 8d290000
men[3] = 24020004
men[4] = 20042024
men[5] = c
men[6] = 11200009
Digite:
1.Para rodar o passo a passo dos registradores: step()
2.Para rodar todo o programa: run()
2
Os oito primeiros numeros primos sao : 1 3 5 7 11 13 17 19
-- Program is finished running --
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$
```

Messages Run I/O

-- program is finished running --

Reset: reset completed.

Os oito primeiros numeros primos sao : 1 3 5 7 11 13 17 19

-- program is finished running --

Comparação dos valores dos registradores:

```
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$ ./naib
Registrador ri: 2402000a
Digite:
Registradores:
Reg:      Valores (hexadecimal):
$zero:    0
$at:      0
$v0:      0xa
$v1:      0
$a0:      0x204c
$a1:      0
$a2:      0
$a3:      0
$t0:      0x2020
$t1:      0
$t2:      0
$t3:      0
$t4:      0
$t5:      0
$t6:      0
$t7:      0
$s0:      0
$s1:      0
$s2:      0
$s3:      0
$s4:      0
$s5:      0
$s6:      0
$s7:      0
$t8:      0
$t9:      0
$sk0:     0
$sk1:     0
$gp:      0x1800
$sp:      0x3ffc
$fp:      0
$ra:      0
pc:        0x44
hi:        0
lo:        0

-- Program is finished running --
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$
```

Coproc 1		Coproc 0	
Registers			
Name	Number	Value	
\$zero	0	0x00000000	
\$at	1	0x00000000	
\$v0	2	0x0000000a	
\$v1	3	0x00000000	
\$a0	4	0x0000204c	
\$a1	5	0x00000000	
\$a2	6	0x00000000	
\$a3	7	0x00000000	
\$t0	8	0x00002020	
\$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	0x00001800	
\$sp	29	0x00003ffc	
\$fp	30	0x00000000	
\$ra	31	0x00000000	
pc		0x00000044	
hi		0x00000000	
lo		0x00000000	

Programa lendo ASM que imprime números primos:

```
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$ ./main
Sucesso na leitura do data.binSucesso na leitura do text.bin

Print dos primeiros 7 registradores:
men[0] = 20082000
men[1] = 20092020
men[2] = 8d290000
men[3] = 24020004
men[4] = 20042024
men[5] = c
men[6] = 11200009
Digite:
1.Para rodar o passo a passo dos registradores: step()
2.Para rodar todo o programa: run()
2
Os oito primeiros numeros primos sao : 1 3 5 7 11 13 17 19
-- Program is finished running --
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/primos$
```

Programa lendo testador.asm e executando run():

```
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/testador$ ./main
Sucesso na leitura do data.binSucesso na leitura do text.bin

Print dos primeiros 7 registradores:
men[0] = 20082000
men[1] = 8d090000
men[2] = 3c01f0fe
men[3] = 342af0f0
men[4] = 24050001
men[5] = 112a0002
men[6] = c00012f
Digite:
1.Para rodar o passo a passo dos registradores: step()
2.Para rodar todo o programa: run()
2
Teste1 OK
Teste2 OK
Teste3 OK
Teste4 OK
Teste5 OK
Teste6 OK
Teste7 OK
Teste8 OK
Teste9 OK
Teste10 OK
Teste11 OK
Teste12 OK
Teste13 OK
Teste14 OK
Teste15 OK
Teste16 OK
Teste17 OK
Teste18 OK
Teste19 OK
Teste20 OK
Teste21 OK
Teste22 OK
Teste23 OK
Teste24 OK
Teste25 OK
Teste26 OK
Teste27 OK
Teste28 OK
Teste29 OK
Teste30 OK
Teste31 OK
-- Program is finished running --
ryo@ryo-500R4K-500R5H-5400RK-501R5H-5500RH-500R5S:~/Documentos/OAC/Trab2_OAC_150115326/testador$
```

## **Configurações**

### **1 Compilador:**

gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)

### **2 Sistema Operacional:**

Ubuntu 16.04.5 LTS

### **3 IDE:**

Foi utilizado o compilador gcc via terminal pelo comando: gcc simulador.c -o simulador