MARS: syscall

Prof. Marcelo D. Berejuck

Engenharia de Computação | INE | CTC

Simulador MARS

- O que é o MARS?
  - **M**IPS **A**ssembler and **R**untime **S**imulator
  - É um simulador do processador MIPS
  - Escrito em linguagem Java
  - Projeto desenvolvido na Missouri State University (EUA)

- Como baixar o simulador?
  - http://courses.missouristate.edu/KenVollmar/Mars/download.htm

Formato de um Programa MIPS

- Cada linha contém **no máximo** uma instrução

- O símbolo # indica início de comentário
  - Linguagem Assembly demanda por muitos comentários!
  - Pelo menos um por linha!

Partes de um Programa MIPS

- Identificar o segmento de dados (**.data**) e de programa (**.text**)

```
.data
...     # Declaração variáveis do programa.
...
.text
...     # Início do programa.
...
```

- No simulador MARS, o **segmento de dados** começa no endereço **0x10010000** e o **segmento de programa** começa no endereço **0x00400000**

Partes de um Programa MIPS

- Exemplo de declaração de variáveis:

```
.data
        hello: .asciiz "texto"
        var: .word 7
.text
...
...     # Início do programa.
```

Chamadas de Sistema

- A instrução **syscall** é usada para chamar o **sistema operacional** da máquina (seu notebook!)
  - Entradas (inputs)
  - Saídas (outputs)

- Operação básica
  - Carrega **$v0** com o comando para ser executado
  - Coloca o valor de saída em **$a0** (ou **$f12**)
  - Recebe resultado de entrada em **$v0** (ou **$f0**)

Chamadas de Sistema úteis

Comando ($v0)	Evento	Argumentos	Resultado
1	Escreve INT	$a0 (inteiro)	$a0 escrito no console
2	Escreve FLOAT	$f12 (float)	$f12 escrito no console
3	Escreve DOUBLE	$f12 (double)	$f12 escrito no console
4	Escreve STRING	$a0 (ponteiro)	String escrita no console
5	Ler INT		$v0 armazena inteiro lido
6	Ler FLOAT		$f0 armazena float lido
7	Ler DOUBLE		$f0 armazena double lido
8	Ler STRING	$a0 (buffer), $a1 (tamanho)	String lida do console
10	Sair do programa		
11	Escreve BYTE	$a0 (byte)	Byte escrito no console

Escrevendo um inteiro no Console

- O comando é **1**
- O comando precisa estar no registrador $v0
- Valor a ser escrito no console deve estar no registrador $a0
- Exemplo: escrever o valor 10 no console

```
addi    $v0, $v0, 1     # Comando.
addi    $a0, $a0, 10    # Valor a ser escrito.
syscall
```

Escrevendo strings no Console

- As string são definidas na Seção **.data**
- Faz-se o uso de rótulos (*labels*) para diferentes strings
- Cada *label* representa um endereço de memória (ponteiro)

```
.data
        prompt: .asciiz "Entre com um inteiro: "
        result: .asciiz " O inteiro eh: "
        newline: .asciiz "¥n "
        hello: .asciiz "Oi, alunos de INE5411!"
        var: .word 7
```

Escrevendo string no Console

- O comando é **4**
- O comando precisa estar no registrador $v0
- Valor do ponteiro da string deve estar no registrador $a0

Exemplo: escrever a variável **hello** (string) no console

```
li      $v0, 4          # Comando.
la      $a0, hello      # Carrega string (endereço).
syscall
```

Lendo valores do Console

- O comando é **5**
- O comando precisa estar no registrador $v0
- Valor lido no console é armazenado no próprio $v0

- Exemplo: ler valor digitado no console (número + tecla enter)

```
li      $v0, 5      # Comando para ler inteiro.
syscall
move    $t0, $v0    # Resultado é salvo em $t0.
```

Chamadas úteis - arquivos

Comando ($v0)	Evento	Argumentos	Resultado
13	Abre um arquivo	**$a0**: string com nome do arquivo; **$a1**: flag (0-ler; 1-escrever); **$a2**: modo.	$v0 recebe o descritor do arquivo. Se estiver com erro, será um valor negativo.
14	Leitura de arquivo	**$a0**: descritor do arquivo; **$a1**: buffer de dados; **$a2**: quantidade de caracteres a serem lidos.	$v0 recebe o descritor do arquivo. Se estiver com erro, será um valor negativo.
15	Escrita no arquivo	**$a0**: descritor do arquivo; **$a1**: buffer de dados; **$a2**: quantidade de caracteres a serem escritos.	$v0 recebe o descritor do arquivo. Se estiver com erro, será um valor negativo.
16	Fechar o arquivo	**$a0**: descritor do arquivo.	S.O. fechará o arquivo.

Manipulação de arquivo

```
.data
        fout: .asciiz "testout.txt"     # nome do arquivo de saída.
        buffer: .asciiz "Exemplo de texto sendo escrito no arquivo."

.text
        ################################################################
        # Primeiro passo: abrir um arquivo para escrita (não existente).
        ################################################################
        li $v0, 13                      # Comando para abrir novo arquivo.
        la $a0, fout                    # Carrega nome do arquivo a ser aberto.
        li $a1, 1                       # Aberto para escrita (flags são 0: read, 1: write).
        li $a2, 0                       # Modo ignorado (neste caso).
        syscall                         # Abre arquivo (descritor do arquivo é colocado em $v0).
        move $s6, $v0                   # Salva o descritor do arquivo para uso no fechamento, por exemplo.

        ################################################################
        # Escreve no arquivo aberto.
        ################################################################
        li $v0, 15                      # Comando para escrever no arquivo.
        move $a0, $s6                   # Descritor do arquivo é passado.
        la $a1, buffer                  # Endereço do buffer do qual será copiado para o arquivo.
        li $a2, 44                      # Tmanho do buffer (hardcoded).
        syscall                         # Escreve no arquivo.

        ################################################################
        # Fechar o arquivo após escrever.
        ################################################################
        li $v0, 16                      # Comando para fechamento do arquivo.
        move $a0, $s6                   # Descritor do arquivo é passado.
        syscall                         # Arquivo é fechado pelo sistema operacional.
        ################################################################
```

Contato

E-mail: marcelo.berejuck@ufsc.br

Telefone: +55 (48) 3721-7508

Site: www.ine.ufsc.br