



PUCRS

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

ARTHUR D'ARISBO

LUCAS RIBEIRO

Arquitetura de Processadores na Prática – TP2

Um breve relatório

Porto Alegre – RS
2018

Respostas às perguntas propostas

1. Qual o número do registrador \$sp no conjunto de registradores do MIPS e qual o seu valor inicial (atribuído pelo simulador MARS)? O número do registrador é o 29 e o valor atribuído é o endereço na memória de dados 0x7ffffcfc.
2. Qual é o primeiro valor escrito na pilha, e qual o significado do mesmo? O primeiro valor escrito na pilha é o endereço contido no registrador \$ra durante a chamada da função.
3. Mostre o conteúdo da pilha ao entrar na terceira chamada aninhada de alguma recursão (use a opção File→Dump Memory do simulador MARS).

Text Segment	Code	Basic	Source
0x00400100	0x00400100	jr \$ra	154: jr \$ra
0x00400104	0x00400104	lw \$s0, 0(\$sp)	155: lw \$s0, 0(\$sp) # Carrega o endereço do RA
0x00400108	0x00400108	beq \$s0, zero, cap_ra_ins	156: beq \$s0, zero, cap_ra_ins # Caso RA não tenha sido capturado salta para capturar
0x0040010c	0x0040010c	lw \$s1, 4(\$sp)	157: lw \$s1, 4(\$sp) # Carrega o endereço do apontador da lista
0x00400110	0x00400110	lw \$s2, 0(\$s1)	158: lw \$s2, 0(\$s1) # Carrega o dado na posição
0x00400114	0x00400114	beq \$s2, zero, ins_el	159: beq \$s2, zero, ins_el # Verifica se existe dado na posição : salta para inserir
0x00400118	0x00400118	addiu \$s1, \$s1, 4	160: addiu \$s1, \$s1, 4 # Incrementa o apontador para o próximo item
0x0040011c	0x0040011c	sw \$s1, 4(\$sp)	161: sw \$s1, 4(\$sp) # Atualiza o endereço do próximo elemento na pilha
0x00400120	0x00400120	jal ins_elt	162: jal ins_elt # Chama a função novamente
0x00400124	0x00400124	lw \$s4, 8(\$sp)	163: lw \$s4, 8(\$sp) # Carrega o dado a ser inserido
0x00400128	0x00400128	sw \$s4, 0(\$s1)	164: sw \$s4, 0(\$s1) # Adiciona o dado na lista
0x0040012c	0x0040012c	addiu \$s1, \$s1, 4	165: addiu \$s1, \$s1, 4 # Endereço para o próximo elemento
0x00400130	0x00400130	sw \$s2, 0(\$s1)	166: sw \$s2, 0(\$s1) # Grava o endereço do próximo elemento
0x00400134	0x00400134	sw \$s2, 0(\$s1)	167: sw \$s2, 0(\$s1) # Salva o endereço para próximo elemento
0x00400138	0x00400138	end_ins	168: end_ins # Salta finalizar a função
0x0040013c	0x0040013c	sw \$ra, 0(\$sp)	169: sw \$ra, 0(\$sp) # Salva o RA para retornar
0x00400140	0x00400140	main_ins	170: main_ins # Retorna para a função

Data Segment	Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
	0x7ffffcfc	0x00000000	0x00000000	0x00000000	0x00000000	0x00400090	0x10040008	0x0000001e	0x00000000
	0x7fffffd0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7fffffd4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7fffffd8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	0x7ffffdc8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Figura 1: Captura de tela do simulador MARS.

4. Qual o conteúdo do registrador \$sp neste momento? Para entender o conteúdo do \$sp (Stack Pointer) é necessário mencionar que para execução das funções é armazenado o endereço de retorno, o endereço do primeiro elemento e o elemento a ser inserido ou o somatório dos elementos. A chamada capturada na imagem acima o registrador \$sp tem armazenado neste momento o endereço de retorno da função, o endereço para o próximo elemento da lista e o elemento a ser inserido ao final da lista.
5. Isto implica quanto espaço alocado na pilha? Para armazenar esses dados foi alocado 12 bytes na pilha equivalentes a três .word.

6. **Observar o retorno do procedimento recursivo. O valor do registrador \$sp volta ao valor original? (Se isto não ocorrer seu programa está incorreto, pois sua execução deixa lixo na pilha).** O retorno da função recursiva somente desaloca (move) o registrador \$sp ao endereço inicial. Porém os dados utilizados na chamada continuam “salvos” nas posições alocadas.
7. **Em qual linha de código este valor é re-estabelecido?** O registrador \$sp (Stack Pointer) é retornado ao endereço original ao finalizar a função e recuperar o retorno (caso haja) na pilha. Assim na chamada descrita acima a linha em que este valor é re-estabelecido é a 54.

```
addiu    $sp, $sp, 12           # Desaloca espaco na pilha
```

Figura 2: Linha de código do programa.