

Integrantes: Arthur R. S. Ferreira, Marcelo A. F. Martins e Mateus F. Charloto

Área compilada no código que lê m e n:

Text Segment					Source	
Bkpt	Address	Code	Basic			
	4194344	0x24020004	addiu \$2,\$0,4	14:	li \$v0, 4	# Imprime msgInicial
	4194348	0x0000000c	syscall	15:	syscall	
	4194352	0x24020005	addiu \$2,\$0,5	17:	li \$v0, 5	# Ler o 'm' do teclado
	4194356	0x0000000c	syscall	18:	syscall	
	4194360	0x04400049	bltz \$2,73	19:	bltz \$v0, fim	# Se o 'm' for negativo encerra o programa
	4194364	0x00024021	addu \$8,\$0,\$2	20:	move \$t0, \$v0	# \$t0 será nosso registrador para o 'm'
	4194368	0x24020005	addiu \$2,\$0,5	22:	li \$v0, 5	# Ler o 'n' do teclado
	4194372	0x0000000c	syscall	23:	syscall	
	4194376	0x04400045	bltz \$2,69	24:	bltz \$v0, fim	# Se o 'n' for negativo encerra o programa
	4194380	0x00024821	addu \$9,\$0,\$2	25:	move \$t1, \$v0	# \$t1 será nosso registrador para o 'n'
	4194384	0x2001000e	addi \$1,\$0,12	27:	subi \$sp, \$sp, 12	# cria pilha de 3 espaços(12 bytes)
	4194388	0x03a1e822	sub \$29,\$29,\$1			
	4194392	0xafaf8000	sw \$8,0(\$sp)	28:	sw \$t0, 0(\$sp)	# \$t0 <- no inicio da fila('m') sw = save word
	4194396	0xafaf9004	sw \$9,4(\$sp)	29:	sw \$t1, 4(\$sp)	# \$t1 <- no meio da fila('n') sw = save word
	4194400	0xafafb008	sw \$31,8(\$sp)	30:	sw \$ra, 8(\$sp)	# \$ra <- no final da fila sw = save word
	4194404	0xc100032	jal 4194504	32:	jal ackermann	# vai para a função

Área compilada do código que faz a função Ackerman:

Text Segment					Source	
Bkpt	Address	Code	Basic			
	4194504	0x8fa40000	lw \$4,0(\$29)	65:	lw \$a0, 0(\$sp)	# \$a0 <- \$t0('m') lw = load word
	4194508	0x8fa50001	lw \$5,4(\$29)	66:	lw \$a1, 4(\$sp)	# \$a1 <- \$t1('n') lw = load word
	4194512	0x10800000	bgez \$4,\$0,2	67:	bgez \$a0, m_equals_zero	# se 'm' = 0 vai para m_equals_zero
	4194516	0x1c800003	bgtz \$4,\$0,3	68:	bgtz \$a0, m_bigger_zero	# se 'm' > 0 vai para m_bigger_zero
	4194520	0x03e00008	jr \$31	69:	jr \$ra	# retorno
	4194524	0x20a50001	addi \$5,\$5,1	72:	addi \$a1, \$a1, 1	# 'n' + 1
	4194528	0x03e00008	jr \$31	73:	jr \$ra	# retorno

Área da pilha na memória:

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)	
2147479520	0	0	0	0	0	5	0	0	
2147479552	0	0	0	0	0	0	0	0	
2147479584	0	0	0	0	0	0	0	0	
2147479616	0	0	0	0	0	0	0	0	
2147479648	0	0	0	0	0	0	0	0	
2147479680	0	0	0	0	0	0	0	0	
2147479712	0	0	0	0	0	0	0	0	
2147479744	0	0	0	0	0	0	0	0	
2147479776	0	0	0	0	0	0	0	0	
2147479808	0	0	0	0	0	0	0	0	
2147479840	0	0	0	0	0	0	0	0	
2147479872	0	0	0	0	0	0	0	0	
2147479904	0	0	0	0	0	0	0	0	
2147479936	0	0	0	0	0	0	0	0	

Em 2147479520, Value +16 é salvo o valor de m, o Value +20 é salvo o n, e o Value +24 é salvo o \$ra.

Estado dos registradores após a execução:

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	1
\$v1	3	0
\$a0	4	6
\$a1	5	6
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	5
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194500
hi		0
lo		0

O Resultado da Função Ackermann é salvo no registrador \$a1 para ser usado no restante do código.

Interface com o exemplo de execução acima ( $m=0$  e  $n=5$ )

Mars Messages

Run I/O

Programa Ackermann  
Componentes: <Arthur, Marcelo e Mateus>

Digite os parâmetros m e n para calcular A(m, n) ou -1 para abortar a execução:  
0  
5  
A(0,5)= 6  
Digite os parâmetros m e n para calcular A(m, n) ou -1 para abortar a execução:

Clear