

Trabalho Prático OAP – Luige Lima, Marcos Zanardi

Para captura das imagens, foi utilizado o teste de Ackermann(3, 4)

Capturas do MARS:

A área de código compilada:

| Text Segment | | | | |
|--------------|------------|------------|------------------|---|
| Bkpt | Address | Code | Basic | Source |
| | 0x00400000 | 0x24020004 | addiu \$2,\$0,4 | 14: li \$v0, 4 |
| | 0x00400004 | 0x3c011001 | lui \$1,4097 | 15: la \$a0, intro_msg |
| | 0x00400008 | 0x342400a3 | ori \$4,\$1,163 | |
| | 0x0040000c | 0x0000000c | syscall | 16: syscall |
| | 0x00400010 | 0x24020004 | addiu \$2,\$0,4 | 18: li \$v0, 4 |
| | 0x00400014 | 0x3c011001 | lui \$1,4097 | 19: la \$a0, authors |
| | 0x00400018 | 0x342400ca | ori \$4,\$1,202 | |
| | 0x0040001c | 0x0000000c | syscall | 20: syscall |
| | 0x00400020 | 0x24020004 | addiu \$2,\$0,4 | 24: li \$v0, 4 |
| | 0x00400024 | 0x3c011001 | lui \$1,4097 | 25: la \$a0, prompt_m |
| | 0x00400028 | 0x34240000 | ori \$4,\$1,0 | |
| | 0x0040002c | 0x0000000c | syscall | 26: syscall |
| | 0x00400030 | 0x24020005 | addiu \$2,\$0,5 | 29: li \$v0, 5 # Lê o valor do usuário |
| | 0x00400034 | 0x0000000c | syscall | 30: syscall |
| | 0x00400038 | 0x04400039 | bltz \$2,57 | 31: bltz \$v0, exit_program # Se m < 0, sai do programa |
| | 0x0040003c | 0x00024021 | addu \$8,\$0,\$2 | 32: move \$t0, \$v0 # Salva m em \$t0 |
| | 0x00400040 | 0x24020004 | addiu \$2,\$0,4 | 35: li \$v0, 4 |
| | 0x00400044 | 0x3c011001 | lui \$1,4097 | 36: la \$a0, prompt_n |
| | 0x00400048 | 0x3424004b | ori \$4,\$1,75 | |
| | 0x0040004c | 0x0000000c | syscall | 37: syscall |
| | 0x00400050 | 0x24020005 | addiu \$2,\$0,5 | 40: li \$v0, 5 # Lê o valor do usuário |
| | 0x00400054 | 0x0000000c | syscall | 41: syscall |
| | 0x00400058 | 0x04400031 | bltz \$2,49 | 42: bltz \$v0, exit_program # Se n < 0, sai do programa |
| | 0x0040005c | 0x00024821 | addu \$9,\$0,\$2 | 43: move \$t1, \$v0 # Salva n em \$t1 |
| | 0x00400060 | 0x0c100025 | jal 0x00400094 | 46: jal ackermann # Chama a função ackermann |

| Text Segment | | | | |
|--------------|------------|------------|--------------------|---|
| Bkpt | Address | Code | Basic | Source |
| | 0x00400064 | 0x24020004 | addiu \$2,\$0,4 | 49: li \$v0, 4 |
| | 0x00400068 | 0x3c011001 | lui \$1,4097 | 50: la \$a0, result_prompt |
| | 0x0040006c | 0x34240096 | ori \$4,\$1,150 | |
| | 0x00400070 | 0x0000000c | syscall | 51: syscall |
| | 0x00400074 | 0x24020001 | addiu \$2,\$0,1 | 53: li \$v0, 1 # Imprimir número |
| | 0x00400078 | 0x000b2021 | addu \$4,\$0,\$11 | 54: move \$a0, \$t3 # Move o resultado de volta para \$a0 |
| | 0x0040007c | 0x0000000c | syscall | 55: syscall |
| | 0x00400080 | 0x24020004 | addiu \$2,\$0,4 | 57: li \$v0, 4 |
| | 0x00400084 | 0x3c011001 | lui \$1,4097 | 58: la \$a0, newline |
| | 0x00400088 | 0x342400a1 | ori \$4,\$1,161 | |
| | 0x0040008c | 0x0000000c | syscall | 59: syscall |
| | 0x00400090 | 0x08100008 | j 0x00400020 | 61: j loop # Volta ao início do loop |
| | 0x00400094 | 0x23bdfbf4 | addi \$29,\$29,-12 | 65: addi \$sp, \$sp, -12 # Aloca espaço na pilha para \$ra, \$t0 e \$t1 |
| | 0x00400098 | 0xafbf0008 | sw \$31,8(\$29) | 66: sw \$ra, 8(\$sp) # Salva \$ra |
| | 0x0040009c | 0xafa80004 | sw \$8,4(\$29) | 67: sw \$t0, 4(\$sp) # Salva \$t0 |
| | 0x004000a0 | 0xafa90000 | sw \$9,0(\$29) | 68: sw \$t1, 0(\$sp) # Salva \$t1 (n original) |
| | 0x004000a4 | 0x11000002 | beq \$8,\$0,2 | 71: beq \$t0, \$zero, base_case |
| | 0x004000a8 | 0x11200007 | beq \$9,\$0,7 | 73: beq \$t1, \$zero, recursive_case1 |
| | 0x004000ac | 0x0810003b | j 0x004000ec | 75: j recursive_case2 |
| | 0x004000b0 | 0x212b0001 | addi \$11,\$9,1 | 78: addi \$t3, \$t1, 1 # A(m, n) = n + 1 |
| | 0x004000b4 | 0x8fbf0008 | lw \$31,8(\$29) | 79: lw \$ra, 8(\$sp) # Recupera \$ra |
| | 0x004000b8 | 0x8fa80004 | lw \$8,4(\$29) | 80: lw \$t0, 4(\$sp) # Recupera \$t0 |
| | 0x004000bc | 0x8fa90000 | lw \$9,0(\$29) | 81: lw \$t1, 0(\$sp) # Recupera \$t1 (n original) |
| | 0x004000c0 | 0x23bd000c | addi \$29,\$29,12 | 82: addi \$sp, \$sp, 12 # Desaloca a pilha |
| | 0x004000c4 | 0x03e00008 | jr \$31 | 83: jr \$ra |

O estado dos registradores ao término de uma execução:

| | Registers | Coproc 1 | Coproc 0 | |
|-----------------|-----------|----------|------------|------------|
| | Name | Number | Value | |
| | \$zero | 0 | 0 | |
| | \$at | 1 | 268500992 | |
| | \$v0 | 2 | 10 | |
| | \$v1 | 3 | 0 | |
| | \$a0 | 4 | 268500992 | |
| | \$a1 | 5 | 0 | |
| | \$a2 | 6 | 0 | |
| | \$a3 | 7 | 0 | |
| | \$t0 | 8 | 3 | Valor de m |
| | \$t1 | 9 | 4 | Valor de n |
| Resultado Temp. | \$t2 | 10 | 124 | |
| Resultado Final | \$t3 | 11 | 125 | |
| | \$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 | 4194404 | |
| | pc | | 4194600 | |
| | hi | | 0 | |
| | lo | | 0 | |

A área de pilha utilizada para a recursividade ao término do programa:

| Data Segment | | | | | | | | |
|--------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
| 0x7ffffefe0 | 4194568 | 61 | 2 | 4194568 | 4 | 3 | 4194404 | 0 |
| 0x7ffff000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff020 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff040 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff060 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff080 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff0a0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff0c0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff0e0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff140 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff160 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff180 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff1a0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x7ffff1c0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Um exemplo de execução do programa (utilizado para as outras imagens também):

Mars Messages

Run I/O

Autores: Luige Lima e Marcos Zanardi

Digite o parâmetro m para calcular A(m, n) ou -1 para abortar a execução: 3

Digite o parâmetro n para calcular A(m, n) ou -1 para abortar a execução: 4

A(m, n) = 125

Digite o parâmetro m para calcular A(m, n) ou -1 para abortar a execução: -1

Clear

Mars Messages

Run I/O

A(m, n) = 125

Digite o parâmetro m para calcular A(m, n) ou -1 para abortar a execução: -1

-- program is finished running --

Clear