Centro Federal de Educação Tecnológica de Minas Gerais

CEFET-MG

DEPARTAMENTO DE COMPUTAÇÃO (DECOM)
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I
Professor: Juliana Santiago Teixeira

Aluno: Igor Luciano de Paula

PRÁTICA 3: IMPLEMENTAÇÃO DE FUNÇÕES EM ASSEMBLY DO MIPS

1) Quais conceitos foram utilizados para implementar esta tradução?

R.: Foram utilizados conceitos aritméticos, conceitos de constantes e operadores imediatos, bem como, registradores temporários, registradores de retorno, registradores de argumento, registradores de valores salvos, registrador de endereço de retorno e o registrador zero. Além disso foram utilizados conceitos de organização de memória do MIPS, instruções de transferência de dados (li, la, lw, sw), indexamento de array, operações de controle de fluxo, desvios condicionais, desvios incondicionais, procedimentos (funções) e chamadas de S.O..

2) O que acontece se o valor de y for 34? Qual o valor retornado para uma chamada g(31,34)? Aponte os valores intermediários assumidos por “a[i]” e por “i” durante uma chamada com esses valores (g(31,34)).

R.: Em y igual a 34, o algorítimo trabalhará de forma regular, mas como o vetor possui apenas 32 posições, o valor de x não pode ser maior que 31, caso contrário o programa irá tentar somar uma posição que não pertence ao vetor.

Para uma chamada g(31, 34) o valor de retorno será 66, resultante de todas as operações do programa.

O valor final que “i” irá assumir após a execução do programa será 34, presente no registrador temporário “$t1”. Seu valor se inicia com 31 (valor de x), e a cada execução do Loop for o mesmo é acrescido em uma unidade. Já valor de “a[i]”, na primeira chamada do for, tem como valor a soma de x mais y, resultando em 65, e a cada execução do Loop, o mesmo é acrescido em uma unidade (pois a cada execução do Loop x é acrescido em um). O valor final de a[i] é 67, armazenado no registrador “$t3”.

3) O que poderia acontecer se a função “g” guardasse o seu endereço de retorno na pilha (para poder efetuar uma chamada a outro procedimento, por exemplo)?

R.: Isso permitiria a chamada de uma função recursiva, pois a cada chama um novo registrador de endereço de retorno(“$ra”) seria necessário. Caso o registrador de endereço de retorno do procedimento “g” fosse salvo em uma pilha, seria possível chamar uma outra função com o comando “jal” (Jump and Link) dentro da função “g”.