

ISA do IA-32 (parte 2)

Teste 4

1. Analise **apenas uma das seguintes** afirmações, e identifique eventuais incorrecções:

- (A/R) "Ao analisar o modo como um compilador reserva espaço em memória para arrays de inteiros ou reais, deve-se considerar que (i) a quantidade de espaço ocupado vai depender do sistema operativo usado e da linguagem imperativa de programação, e que (ii) o desempenho na execução do código não vai ser nunca afectado pelo modo como o array vai ser armazenado na memória."
- (A/R) "Quando é preciso aceder ao valor de um campo de uma structure em C, o compilador gera código para aceder à memória, o qual deve usar sempre um modo de endereçamento em que se especificam 2 registos: um com o endereço de base da structure e outro com um valor variável para indicar a distância do início da structure ao início do campo da structure."

2. (A/R) Considere a estrutura de controlo de um ciclo iterativo, num programa escrito numa linguagem imperativa (por exemplo, em C a estrutura `for(i=0, i<Max, i++)`). Quando o compilador tiver de gerar código para implementar esta estrutura em linguagem máquina dum processador do tipo do IA-32 (onde a única instrução disponível é um salto condicional do tipo `if<cond>then`, indique como é que o compilador transforma o código deste ciclo para que ele possa ser traduzido para linguagem máquina.

Nº	Nome
----	------

3. Considere a seguinte expressão aritmética com inteiros, no corpo de uma função em C, para execução num PC (*little endian*),

$$a[i] = b + c$$

em que a é a única variável local que se encontra em memória, (um *array* definido com a dimensão 20), i é o índice do *array* que foi passado como 2º argumento da função (o 1º foi um valor real de precisão dupla), e b e c são variáveis locais (nos registos `%ecx` e `%esi`). Os registos `%eax`, `%ebx` e `%edx` são usados na função para conterem, respectivamente, os valores de retorno da função, de i , e de apontador para início do *array*.

- a) (A/R) Represente a estrutura da *stack frame* desta função, com indicação da dimensão_e_descrição de cada um dos valores nela presente.

- b) (R/B) Considere que:
- o conteúdo de cada célula de memória é igual ao *byte* menos significativo do seu endereço;
 - o código da função que invoca esta se encontra imediatamente a seguir ao código desta função;
 - a execução deste programa com um ponto de paragem antes do início do corpo desta função, permitiu constatar que nesse ponto o conteúdo dos registos `%ecx`, `%esi`, e `%ebp` era, respectivamente, o seguinte conjunto de valores em hexadecimal: `0xffffef`, `0x200`, `0x84012c0`.

Mostre a localização e conteúdo das células que vão conter a instrução que invoca esta função.

- c) (B) Compare a estrutura desta *stack frame* com a que seria necessário construir numa arquitectura RISC, assumindo que esta função era uma função-folha.

Nº	Nome
----	------