

MC404: Organização de Computadores e Linguagem de Montagem

1º Semestre de 2012 - Turmas A e B - Prof. Edson Borin

2ª Prova (20/06/2012)

Nome: _____

RA: _____

Questão	Valor	Nota
1	2,5	
2	2,5	
3	0,5	
4	1,0	
5	1,5	
6	2,0	
Total	10,0	

Instruções: A duração da prova é de uma hora e quarenta minutos. *Consulta exclusivamente às folhas de resumos de instruções do ARM.* Comente seu código! Qualquer tentativa de fraude será punida com zero para todos os envolvidos.

Questão 1. (2,5 pontos) O procedimento a seguir deve somar o conteúdo de dois vetores de inteiros e gravar o resultado em um terceiro vetor. Os argumentos para o procedimento são organizados da seguinte forma: **r0** e **r1** contém o apontador para o primeiro e o segundo vetor, respectivamente. **r2** contém o apontador para o vetor resultado e **r3** contém um número indicando o tamanho dos vetores (o tamanho do vetor é sempre maior ou igual a 1). Você deve preencher as lacunas abaixo para completar o procedimento. Salve e recupere apenas os registradores necessários.

soma_vetor:

```

    -----          @ Salva registradores callee-save
loop:
    ldr r4, -----  @ Carrega valor do primeiro vetor e incrementa apontador
    ldr r5, -----  @ Carrega valor do segundo vetor e incrementa apontador
    add r4, r4, r5    @ Soma valores dos vetores
    str r4, -----  @ Armazena o resultado da soma no vetor resultado
    sub r3, r3, #1    @ Decrementa o contador e salta para loop enquanto
    -----          @ há elementos para somar
    -----          @ Recupera registradores callee-save e retorna
```

Questão 2. (2,5 pontos) A função **atualiza** deve atualizar o elemento da posição **i** do vetor de inteiros **vetor** com o valor **novo**. O valor antigo deve ser retornado. A assinatura da função na linguagem de programação C é: `int atualiza(int i, int novo, int* vetor);`

Você deve implementar a função **atualiza** observando as seguintes regras:

- Modifique apenas registradores *caller-save*.
- Implemente sua função com exatamente quatro instruções.
- A terceira instrução deve ser `mov r0, r3`

atualiza:

```

-----
-----
mov r0, r3
-----
```

Questão 3. (0,5 ponto) O que é o vetor de interrupções, onde ele fica armazenado e para que ele serve?

Questão 4. (1,0 ponto) O que é a leitura com “espera ocupada” (*Busy waiting*) e quais são os problemas com esta abordagem?

Questão 5. (1,5 pontos) Com relação ao mecanismo de tratamento de exceções da arquitetura ARM, responda:

- a) para onde e como é modificado o fluxo de execução?
- b) o que acontece com os valores dos registradores que estavam sendo utilizados pela aplicação?
- c) onde o CPSR é salvo e o que acontece com seu valor quando acontece uma exceção?

Questão 6. (2,0 pontos) Traduza o procedimento em C abaixo para linguagem de montagem ARM. Considere que os parâmetros são passados nos registradores de acordo com a convenção da ABI do ARM. **Você não pode utilizar instruções de salto**, utilize o mecanismo de execução condicional de instruções para implementar o corpo da sentença **if-then-else**.

```
void dif_abs(unsigned val1, unsigned val2, unsigned *result)
{
    if (val1 > val2)
        *result = val1 - val2;
    else
        *result = val2 - val1;
}
```