

MC404: Organização de Computadores e Linguagem de Montagem

2ª Prova (24/11/2011)

Nome: _____

RA: _____

Questão	Valor	Nota
1	5,0	
2	1,0	
3	1,0	
4	2,0	
5	1,0	
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. (5,0 pontos)

A população da Baitelândia, uma pequena cidade do interior, cresceu rapidamente e, como consequência, o tráfego de carros se tornou um problema. Para melhorar o trânsito, o prefeito Iscuzi decidiu instalar semáforos em todos os cruzamentos da cidade. Um projetista de *hardware*, contratado para ajudar no projeto do semáforo, desenvolveu um dispositivo de saída, para controlar as luzes do semáforo, e o conectou ao barramento de um sistema com um processador ARM. O dispositivo permite ao programador ligar e desligar as luzes do semáforo através da escrita (saída) em um registrador de controle do dispositivo, chamado `LIGHTS_CTRL`. O registrador de controle possui 8 *bits* e pode ser acessado no endereço `500C_0404`. Os *bits* 0, 1 e 2 do registrador `LIGHTS_CTRL` controlam as luzes verde, amarela e vermelha da primeira rua do cruzamento. Se o programador escrever 1 no *bit* a luz associada ao mesmo é ligada, se escrever 0 a luz é desligada. Analogamente os *bits* 3, 4 e 5 controlam as luzes verde, amarela e vermelha da segunda rua do cruzamento. Faça um programa em linguagem de montagem ARM para controlar o semáforo. O programa deve:

- Após 1 minuto, ligar a luz amarela da rua 2.
- Então, após 10 segundos, ligar a luz verde da rua 1 e a luz vermelha da rua 2.
- Então, após 1 minuto, ligar a luz amarela da rua 1.
- Então, após 10 segundos, ligar a luz vermelha da rua 1 e a luz verde da rua 2.
- Retornar ao passo inicial (a).

Suponha que seu programa rode em modo usuário e o sistema ARM já possua um sistema operacional baseado em POSIX/UNIX com as chamadas de sistema `alarm()` e `sigaction()` implementadas. Como visto em um dos trabalhos da disciplina, você pode utilizar as chamadas de sistema `sigaction` e `alarm` para controlar o tempo. A interface (simplificada) das mesmas é descrita abaixo.

- sigaction:** O número dessa chamada de sistema é 67. Os argumentos são passados através de `r0` e `r1`: `r0` contém o código do sinal e `r1` contém o ponteiro para uma função que será chamada quando o sinal acontecer. O código do sinal `SIGALRM` é 14.
- alarm:** O número dessa chamada de sistema é 27. Ela solicita que o sistema operacional dispare um sinal `SIGALRM` após um determinado tempo. O tempo, especificado em segundos, é informado em `r0`.

Questão 2. (1,0 ponto)

Determine o **maior** e o **menor** valor que podem ser representados usando-se 11 *bits*. Mostre sua resposta em decimal.

Complemento de 2		Sinal e Magnitude		Complemento de 1		Sem Sinal	
Maior	Menor	Maior	Menor	Maior	Menor	Maior	Menor

Questão 3. (1,0 ponto) Preencha as lacunas em branco da tabela de acordo com a representação da coluna. Preencha o espaço com um traço se o número não puder ser representado no formato da coluna.

Decimal	Binário de 7 <i>bits</i>		
	Sem sinal	Complemento de 2	Sinal e Magnitude
65			
	001 0000		
		100 0000	
			100 0001

Questão 4. (2,0 pontos) Traduza o procedimento em C abaixo para linguagem de montagem ARM. Considere que os parâmetros são passados nos registradores **r0–r2** (**val1** em **r0**, **val2** em **r1** e **result** em **r2**).

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

Questão 5. (1,0 ponto) Escreva um exemplo de chamada do procedimento **dif_abs** acima, tendo como parâmetros as variáveis de nomes **x**, **y** e **z**, definidas como inteiros de 32 bits.