MC404: Organização de Computadores e Linguagem de Montagem

1º Semestre de 2013 - Turma E - Prof. Edson Borin

2ª Prova (20/06/2013)

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

Nome:

RA:

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. (1 ponto) Descreva os passos do mecanismo de tratamento de interrupções dos processadores ARM.

Questão 2. (3,0 pontos) Traduza o procedimento em C abaixo para linguagem de montagem do ARM. O código deve estar de acordo com a ABI vista no curso.

```
int do_something(int v1, int v2, int v3, int v4, int v5)
{
  return do_something_else(v5, v1, v2, v3, v4);
}
```

Questão 3. (3,0 pontos) Escreva, em linguagem de montagem do ARM, a função

```
void rev(short* v, int sz);
```

que reordena os elementos do vetor, colocando-os de trás para frente. Ou seja, dado o vetor: `v = {0 1 2 3}`, a função `rev` modifica o conteúdo do vetor para `v = {3 2 1 0}`. O código deve estar de acordo com a ABI vista no curso.

Questão 4. (3,0 pontos) Um grupo de alunos da disciplina de MC404 decidiu desenvolver um robô para explorar a superfície do planeta Marte. O robô é controlado por um sistema de computação, baseado em um processador ARM, que possui diversos dispositivos de entrada e saída, incluindo câmeras, radares, sensores de toque, motores, lâmpadas, e um dispositivo que faz bip! bip! Além dos dispositivos de entrada e saída mencionados, o sistema possui um *watchdog timer*, um dispositivo conectado ao barramento que serve para monitorar a sanidade do sistema. Este dispositivo funciona da seguinte maneira:

- Durante o processo de inicialização, o *software* programa o contador do *watchdog timer* para gerar uma interrupção de *reset* após 1,2 segundos.
- O *software* do sistema reinicia o contador do *watchdog timer* periodicamente, evitando que o mesmo atinja o tempo limite e produza uma interrupção de *reset*, re-iniciando o sistema.

Caso o sistema trave devido a uma falha de *software* ou *hardware*, o contador do *watchdog timer* não será reiniciado em tempo e o mesmo produzirá uma interrupção de *reset*, re-iniciando o sistema. O seguinte trecho de código mostra o laço principal do sistema:

```
...
bl programa_watchdog_timer
...
main:
  bl le_sensores
  bl computa_proximo_passo
  bl realiza_acao
  bl faz_bip_bip
  bl reinicia_watchdog_timer
  b  main
```

A rotina `programa_watchdog_timer` habilita o *watchdog timer* e configura o seu contador para produzir uma interrupção de *reset* após 1,2 segundos. A rotina `reinicia_watchdog_timer` apenas reinicia o contador do *watchdog timer* para que o mesmo só produza a interrupção daqui a 1,2 segundos.
O *watchdog timer* possui duas portas, uma de controle, associada ao endereço 0x40405000, e outra de dados, associada ao endereço 0x40405001. A programação do dispositivo é feita através da escrita de um *byte* de controle na porta de controle e de uma palavra com o tempo para *reset*, em milisegundos, na porta de dados. O valor escrito na porta de dados é transferido para o contador do *watchdog timer*. A escrita do valor 0 no *byte* de controle desabilita o *watchdog timer* enquanto que a escrita do valor 1 habilita o mesmo.
Implemente as funções `programa_watchdog_timer` e `reinicia_watchdog_timer`. (Utilize o verso da folha para sua resposta)