

DCC/ICEX/UFMG

Curso: Bacharelado em Ciência da Computação

Disciplina: Software Básico

2º. Semestre de 2017

TRABALHO PRÁTICO 2 – Parte A

O trabalho prático 2 consiste em 2 partes (A e B). Neste documento é especificada a parte A.

O objetivo deste trabalho é a prática de conceitos vistos em sala de aula, como interrupções, exceções, pilhas de execução, entre outros.

Para instalação do msp430 no Windows, usar o IAR, disponível em:

<http://www.dcc.ufmg.br/~mmvieira/se/slac050ag.zip>

Instalação do msp430 no ubuntu:

Para instalar o msp430 em um sistema linux utilize os seguintes comandos:

```
sudo apt-get install msp430-libc mspdebug msp430mcu binutils-msp430 gcc-msp430  
gdb-msp430
```

Para compilar um programa 'exemplo.c' escrito para o msp430 utilize os seguintes comando:

```
msp430-gcc -Os -mmcu=msp430g2553 exemplo.c -o exemplo.elf
```

Para fazer o upload do código no msp430 utilize o seguinte comando:

```
mspdebug rf2500
```

```
(mspdebug) prog exemplo.elf
```

```
(mspdebug) run
```

Parte 0 – Conhecendo o MSP430

Esta parte do trabalho tem por objetivo apresentar o MSP430, assim você deve baixar o programa de exemplo fornecido, compilar e executar. Além disso, você deve aprender a utilizar o gdb para depurar o seu programa.

Parte 1 – Interrupções

Nesta parte você deverá desenvolver um pisca-pisca utilizando o LED vermelho do MSP430. O LED deverá ficar 1 segundo aceso e 2 segundos apagado. É importante mencionar que vocês deverão calcular o tempo corretamente a partir da interrupção de tempo.

Parte 2 – Semáforo

Nesta parte, será utilizado o conceito de interrupções para o desenvolvimento de um simulador de semáforos utilizando o MSP430.

O semáforo utilizará os dois LED's contidos no launchpad do MSP430. O LED 0 é a luz vermelha, o LED 1 a verde, e ao acender os dois LED simultaneamente será a luz amarela. Além dos LEDs o semáforo também irá utilizar o botão do kit de desenvolvimento como entrada, simulando um pedestre esperando a luz vermelha para atravessar.

O semáforo funcionará da seguinte forma:

Sem o pedestre apertar o botão: a luz vermelha deverá ficar acesa 2 segundos, a verde 8, e a amarela 1.

Com o pedestre apertando o botão: se as luzes vermelha ou amarela estiverem acesas, os tempos definidos anteriormente devem ser mantidos. Caso a luz esteja em verde, o tempo para a luz verde será decrementado em 5 segundos, entretanto caso o tempo restante para a luz verde seja menor que 3 segundos, o tempo restante será mantido.

Informações importantes

- A data de entrega será especificada através de uma tarefa no Moodle.
- Os trabalhos poderão ser entregues até as 23:55 do dia especificado para a entrega. O horário de entrega deve respeitar o relógio do sistema Moodle. Haverá uma tolerância de 5 minutos de atraso, de forma que os alunos podem fazer a entrega até as 0:00. A partir desse horário, os trabalhos já estarão sujeitos a penalidades. A fórmula para desconto por atraso na entrega do trabalho prático é:

$$\text{Desconto} = 2^{d/0.32}$$

onde d é o atraso em dias úteis. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.

- Além disso, deverá ser entregue uma pequena documentação contendo todas as decisões de projeto que foram tomadas durante a implementação, sobre aspectos não

contemplados na especificação, assim como uma justificativa para essas decisões. Esse documento não precisa ser extenso (mínimo 3 e máximo de 10 páginas). A documentação deve indicar o nome dos alunos. O código fonte não deve ser incluído no arquivo PDF da documentação.

- Todas as dúvidas referentes ao trabalho serão esclarecidas por meio do fórum disponível no ambiente Moodle da disciplina.
- A entrega do trabalho deverá ser realizada pelo Moodle, na tarefa criada especificamente para tal. A entrega deverá ser feita no seguinte formato:
 - O trabalho a ser entregue deverá estar contido em um único arquivo compactado, em formato “.zip”, com o nome no formato “tp1_aluno.zip”
 - O arquivo .zip definido deverá ter três pastas:
 - “src”: Essa pasta deverá conter o código-fonte do montador implementado, juntamente do arquivo Makefile (OBS.: Não devem ser incluídos arquivos .o nem executáveis nessa pasta.)
 - “doc”: Essa pasta deverá conter o arquivo da documentação, em formato PDF. Caso julgue necessário incluir quaisquer outros arquivos a parte, esses deverão ser justificados em um arquivo texto com o nome README.

Atenção: Trabalhos que descumprirem o padrão definido acima serão penalizados

Referências:

<https://sergioprado.org/trabalhando-com-o-msp430-no-linux/>

<http://www.ti.com/lit/ug/slau144j/slau144j.pdf>