



Fundamentos de Arquitetura de Computadores

Trabalho 02

Prof. Tiago Alves

Programação em Linguagem de Montagem MIPS: Aritmética de Inteiros

Introdução

A disciplina de Fundamentos de Arquitetura de Computadores trata de diversos tópicos que nos ajudam a compreender como sistemas eletrônicos de computação são construídos. Esse tipo de conhecimento ajudará profissionais de áreas afetas a tecnologias de informação e comunicação a aplicarem, adequadamente, um computador digital na realização de tarefas que, devido à sua natureza, serão melhores conduzidas por um sistema automatizado.

Além de identificar a conveniência da aplicação dos computadores digitais, a disciplina ajudará a desenvolver competências necessárias para a solução de problemas em sistemas computacionais em operação, principalmente problemas decorrentes de análise de desempenho.

Para construir ou adicionar funcionalidades a esses sistemas computacionais, é necessário conhecimento de linguagens de programação e ferramentas de desenvolvimento. Em nosso curso, o domínio de linguagens de montagem é um pré-requisito para o devido acompanhamento das atividades da disciplina.

Objetivos

- 1) Exercitar conceitos da linguagem de montagem para arquitetura MIPS, especialmente aqueles referentes à implementação de solução de problemas em aritmética inteira.
- 2) Interagir com ferramentas de desenvolvimento para criação, gerenciamento, depuração e testes de projeto de aplicações.

Referências Teóricas

Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. Advanced linux programming. New Riders, 2001.

Material Necessário

- Computador com sistema operacional programável
- Ferramentas de desenvolvimento GNU/Linux ou similares: MARS.



Roteiro

- 1) Revisão de técnicas e ferramentas de desenvolvimento usando linguagem de montagem MIPS.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os princípios e técnicas de desenvolvimento de aplicações usando linguagem de montagem MIPS

- 2) Realizar as implementações solicitadas no questionário do trabalho.

Implementações e Questões para Estudo

- 1) Escreva um programa em linguagem de montagem para MIPS usando, preferencialmente, o simulador MARS como plataforma de desenvolvimento e validação. A sua aplicação deverá calcular o **inverso multiplicativo modular**. Seguem os requisitos de implementação:
 - Sua aplicação deverá receber em entrada em console dois números inteiros e imprimir, como resultado da operação, uma mensagem.
 - O primeiro número inteiro será o **provável número primo** que definirá a classe de resíduos, ou seja, o **módulo**.
 - O segundo número será o número inteiro cujo inverso multiplicativo é demandado.
 - A mensagem de saída poderá ser:
 - O inverso multiplicativo eh ZZ.
 - O modulo nao eh primo.
 - Na sua implementação, esperam-se encontrar as funções:
 - `le_inteiro`, que lerá um primo do console de entrada;
 - `eh_primo`, que testará se o inteiro indicado é, de fato, um número primo;
 - `calc_inverso`, que calculará o inverso multiplicativo;
 - `imprime_erro`, função que imprimirá o erro;
 - `imprime_saida`, função que imprimirá o resultado bem sucedido.
 - Outras funções poderão ser criadas, ficando a critério da equipe de implementação.
 - Dicas:
 - Em Teoria dos Números, o inverso multiplicativo modular de um inteiro **a** é o inteiro **x** que garante que o produto **ax** seja congruente a 1 com respeito ao módulo **m**. A notação padrão de aritmética modular representa essa congruência como:
$$a \cdot x \equiv 1 \pmod{m}$$
 - Para o caso em particular da nossa implementação, o valor do módulo **m** deverá ser um **número primo**. Caso contrário, uma mensagem de erro será indicada ao usuário.
 - A classe de resíduos módulo 5 é o conjunto $\{0, 1, 2, 3, 4\}$. Esses são os restos de divisão inteira de qualquer número inteiro pelo divisor 5. Como 5 também é primo, cada elemento dessa classe de resíduos (exceto o 0) possui um inverso multiplicativo modular: 1 é inverso de 1, 2 é inverso de 3 e 4 é inverso de 4, módulo 5.
 - $4 \cdot 4 = 16$; $16 \% 5 = 16 \bmod 5 = 1$, logo 4 é inverso de 4
 - Exemplos:
 - Exemplo de invocação 1:
5
2
O inverso multiplicativo é 3.
 - Exemplo de invocação 2:



12

3

O modulo nao eh primo.

Instruções e Recomendações

A submissão das respostas aos problemas dos trabalhos deverá ser feita através do Moodle da disciplina.

Cada Problema do Trabalho 02 deverá ser entregue em um pacote ZIP. A dupla de alunos deverá nomear o pacote ZIP da seguinte forma: nome_sobrenome_matricula_nome_sobrenome_matricula_**trab02.zip**.

Entre os artefatos esperados, listam-se:

- códigos-fonte ASM das soluções dos problemas;
- documentação mínima da aplicação:
 - o qual sistema operacional foi usado na construção do sistema;
 - o qual ambiente de desenvolvimento foi usado;
 - o quais são as telas (instruções de uso)
 - o quais são as limitações conhecidas

Não devem ser submetidos executáveis.

Códigos-fonte com erros de sintaxe serão desconsiderados (anulados).

Os trabalhos poderão ser realizados em duplas; a identificação de cópia ou plágio irá provocar anulação de todos os artefatos em recorrência.