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

David A. Patterson, John L. Hennessy , *Computer Organization and Design: the Hardware/Software Interface*, The Morgan Kaufmann series in computer architecture and design , 5th ed

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 a **exponenciação modular**. Seguem os requisitos de implementação:
   - Sua aplicação deverá receber em entrada em console três números inteiros positivos e imprimir, como resultado da operação, uma mensagem.
     - Os três números deverão ser inferiores a 65535.
     - O primeiro número será a base, ou seja, o número inteiro cuja potência (modular) será demandada.
     - O segundo número representará o expoente usado no cálculo da exponencial modular.
     - O terceiro número inteiro será o ***provável* número primo** que definirá a classe de resíduos, ou seja, o **módulo**.
     - As mensagens de saída seguirão três formatos:
       - Caso algum dos parâmetros não atendam à restrição de entrada (números positivos menores que 65535), espera-se a seguinte mensagem:
         - `Entradas invalidas.`
       - Caso o módulo não seja primo, espera-se a seguinte mensagem:
         - `O modulo nao eh primo.`
       - Caso os parâmetros de entrada não apresentem problemas, espera-se a seguinte mensagem:
         - `A exponencial modular AA elevado a BB (mod PP) eh ZZ.`
   - Na sua implementação, recomenda-se a criação das seguintes 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_exp`, que calculará a exponencial modular;
     - `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:
     - A exponenciação modular é um tipo de exponenciação realizada em relação a um módulo. Essa operação é útil em ciência da computação, especialmente no campo de criptografia de chave pública.

- A exponenciação modular calcula o resto quando um inteiro b (a base) é elevada à e-ésima potência (expoente), b^e, e, em seguida, é dividido por um inteiro positivo m (o módulo). Em símbolos, a exponenciação modular pode ser representada por: c ≡ b^e (mod m)
- Por exemplo, se b=5, e=3 e m=13, c=8 é o valor da potência 5^3 (mod 13).
- Dados inteiros b e e, e um inteiro positivo m, há uma solução única c com a propriedade de que 0 <= c <= m.

- Exemplos:
  - Exemplo de invocação 1:
    5
    3
    13
    A exponencial modular 5 elevado a 3 (mod 13) eh 8.
  - Exemplo de invocação 2:
    5
    3
    4
    O modulo nao eh primo.
  - Exemplo de invocação 3:
    -1
    3
    4
    Entradas invalidas.
  - Exemplo de invocação 4:
    3
    0
    4
    Entradas invalidas.
  - Exemplo de invocação 5
    3
    5
    -13
    Entradas invalidas.

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:
  - qual sistema operacional foi usado na construção do sistema;
  - qual ambiente de desenvolvimento foi usado;
  - quais são as telas (instruções de uso)

- 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.