



Fundamentos de Arquitetura de Computadores

Trabalho 04

Prof. Tiago Alves

Programação em Linguagem de Montagem MIPS: Implementação de Algoritmos

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 de ponto flutuante.
- 2) Interagir com ferramentas de desenvolvimento para criação, gerenciamento, depuração e testes de projeto de aplicações.

Referências Teóricas

Patterson, David, and John Hennessy. "Computer Organization and Design: The Hardware/Software Interface." 2012. Waltham: Elsevier Inc.

Material Necessário

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



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 capaz de calcular o CRC32 de um número de 32 bits.
 - O CRC (Cyclic Redundancy Check) é um código corretor de erro usado comumente em redes de comunicações digitais e, também, em dispositivos de armazenamento para detectar mudanças acidentais nos dados brutos. Blocos de dados que são processados por esses sistemas recebem um pequeno valor (no nosso caso, um valor de 32 bits) que é calculado baseado no resto da divisão polinomial de seus conteúdos. Na recepção/leitura dos dados, o cálculo deve ser repetido e, caso o valor de checagem (de 32 bits) calculado não seja idêntico ao que foi transmitido/gravado junto com a informação, uma ação corretiva deverá se adotada quanto à provável corrupção dos dados.
 - Os CRCs são assim chamados devido ao fato de o valor usado para a checagem ser uma redundância, ou seja, um artefato expande o tamanho da mensagem sem, porém, adicionar informação. O algoritmo usado na sua concepção é baseado em códigos cíclicos. CRCs são populares pois são simples de implementar em hardware, fáceis de analisar matematicamente e particularmente bons na detecção de erros comuns causados pela transmissão de dados em canais ruidosos. Devido ao fato de seu valor de conferência/checagem possuir um valor fixo, a função que gera o CRC é ocasionalmente aplicada para o cálculo de resumos (hashs).
 - Sua aplicação deverá receber em entrada em console uma string de **até 16 bytes**.
 - Para essa string, demanda-se o cálculo do CRC32.
 - Exemplo:
 - String de entrada: **"Alo mundo."**
 - CRC32: **c11f1783**
 - (Entrada vazia) String de entrada:
 - CRC32: **00000000**
 - String de entrada: **"a123456b"**
 - CRC32: **af0d6cd6**
 - Na sua implementação, esperam-se encontrar as funções:
 - `le_string`, que lerá uma string do console de entrada;
 - `calc_crc32`, que calculará o **crc32**;
 - `imprime_saida`, função que imprimirá o resultado bem sucedido.
 - Outras funções poderão ser criadas, ficando a critério da equipe de implementação.
 - Dicas:
 - Realize a implementação em C da função CRC32 para ter um código de referência de validação.



- Procure implementar em C de acordo com módulos funcionais que possam ser reaproveitados em ASM MIPS.

Instruções e Recomendações

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

Cada Problema do Trabalho 04 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_**trab04.zip**.

Entre os artefatos esperados, listam-se:

- códigos-fonte assembly MIPS 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 binários/executáveis.

Códigos-fonte com erros de compilação/montagem 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.