



Fundamentos de Arquitetura de Computadores

Trabalho 01

Prof. Tiago Alves

Introdução à Programação em Linguagem Assembly MIPS

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.

Objetivos

- 1) Exercitar conceitos da linguagem de montagem (assembly) MIPS.
- 2) Interagir com ferramentas de desenvolvimento para criação, gerenciamento, depuração e testes de projeto de aplicações.

Referências Teóricas

David A. Patterson; John Hennessy, Organização e Projeto de Computadores, Campus, 3ª Edição, 2005.

Material Necessário

- Computador com sistema operacional programável
- Ambiente de simulação para arquitetura MIPS: MARS ou SPIM.

Roteiro

- 1) Revisão de conceitos básicos da arquitetura MIPS.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os conceitos básicos da arquitetura MIPS.



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

Implementações e Questões para Estudo

- 1) Implemente um programa em *assembly* MIPS que atenda aos seguintes comandos:
- Leia três números inteiros a partir do terminal e menores do que 255. Por exemplo:
9
2
36
 - Processe os números de entrada de forma a gerar a seguinte tela saída:
ADD: 11
SUB: 7
AND: 0
OR: 11
XOR: 11
MASK: 4
SLL(4): 144
SRL(4): 0
 - A operação MASK resulta do AND entre o valor 0x0000001F e o terceiro parâmetro da linha de comando. Seu resultado para a entrada 36 é 4.
 - A operação SLL(4) resulta do deslocamento de 4 bits (a saída de MASK) à esquerda do **primeiro operando** (9) enquanto SRL consiste em deslocar 4 bits (a saída de MASK) à direita o **segundo operando** (2).
 - Outro exemplo:
15
4
2
=====
ADD: 19
SUB: 11
AND: 4
OR: 15
XOR: 11
MASK: 2
SLL(2): 76
SRL(2): 1

Instruções e Recomendações

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

Cada Problema do Trabalho 01 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_**trab01**.zip.

Entre os artefatos esperados, listam-se:



- códigos-fonte 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 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.