

RELATÓRIO TRABALHO 1 - OAC

MATEUS LUIZ FREITAS BARROS

1. Descrição do Problema. O Assembly é uma linguagem de baixo nível que está entre a linguagem de máquina e a linguagem de alto nível, como C ou Java, e o MIPS, processador usado na disciplina de Organização e Arquitetura de Computadores, pode ler instruções Assembly. Neste trabalho o intuito é implementar um emulador de instruções do MIPS, em que as instruções implementadas recebem o endereço e fazem a sua operação na memória virtual definida pelo programa.

2. Descrição das Instruções. As instruções implementadas são:

- Load word - Lê 32 bits a partir de um endereço na memória
- Load half-word - Lê 16 bits a partir de um endereço na memória
- Load half-word unsigned - Lê 16 bits a partir de um endereço na memória sem levar consideração o sinal
- Load byte - Lê 8 bits a partir de um endereço na memória
- Load byte unsigned - Lê 8 bits a partir de um endereço na memória sem levar consideração o sinal
- Store word - Salva 32 bits a partir de um endereço na memória
- Store half-word - Salva 16 bits a partir de um endereço na memória
- Store byte - Salva 8 bits a partir de um endereço na memória

3. Testes e resultados. Todos os testes previstos no documento de especificação do trabalho passaram, e alguns outros testes manuais foram feitos para verificar a corretude do código. Os resultados foram todos como o esperado e as instruções se comportaram como no MIPS.

4. Plataforma utilizada e como inicializar o programa.

- Sistema operacional: Ubuntu 16.04
- Compilador C: gcc (Ubuntu 6.2.0-5ubuntu12) 6.2.0 20161005

Para iniciar o programa basta ter o make e o gcc instalado. Comandos do make:

- "make go" compila tudo e inicia a main
- "make all" compila tudo