

Relatório do Trabalho de Organização de Computadores

Confecção de um disassembler em Assembly

Mathias Recktenvald e Guilherme Brizzi

26 de outubro, 2023

Sumário

1	Introdução	2
2	Objetivos	2
3	Revisão Bibliográfica	2
4	Metodologia	2
5	Experimento	3
6	Resultados	3
7	Discussão	3
8	Conclusões e perspectivas	3
	Referências	4

1 Introdução

A linguagem Assembly é uma representação das instruções de um processador, a qual permite que essas sejam mais legíveis e compreensíveis em comparação ao código de máquina que é realmente usado pelo processador. Em vez de zeros e uns, são usadas palavras-chave de instrução, que permitem criar programas de uma maneira mais similar à programação convencional.

Um *assembler* (ou montador) é um *software* que converte da linguagem Assembly para linguagem de máquina. Em contrapartida, um *disassembler* (ou desmontador) é um *software* que realiza o oposto - ou seja, converte código de máquina nas suas correspondentes instruções Assembly. Tal projeto - um *disassembler* - é objeto de estudo de trabalho.

Para tal, adotou-se o processador MIPS como objeto de estudo - juntamente às suas instruções em código de máquina e ao seu Assembly.

2 Objetivos

Sucintamente, o objetivo principal deste trabalho foi de desenvolver, em linguagem Assembly, um *disassembler* para o código de máquina do processador MIPS. O *disassembler* visa a converter código de máquina de um arquivo binário em um arquivo legível contendo o endereço da instrução, seu valor em base hexadecimal e sua tradução em Assembly.

No intuito de evitar a complexificação do trabalho, o *software* desenvolvido ateu-se tão somente às instruções presentes nos arquivos de teste (*ex-000-012.asm* e *ex-000-012.bin*) fornecidos pelo professor.

Outrossim, também se objetivou aprofundar o aprendizado da linguagem Assembly e da compreensão dos aspectos de programação de baixo nível e do funcionamento do processador.

3 Revisão Bibliográfica

Utilizou-se como bibliografia base o livro-texto "Organização e Projeto de Computadores", de John Hennessy e David Patterson. Secundariamente, usou-se um guia simplificado das instruções MIPS da University of Minnesota.¹

4 Metodologia

Adotou-se como metodologia para o desenvolvimento do *software disassembler* o simulador do processador MIPS MARS, o qual interpreta em um ambiente virtual o comportamento do processador. Valeu-se, ademais, do conteúdo da bibliografia apresentada para auxílio na codificação da tradução das instruções, seus respectivos campos e códigos.

¹Disponível em: <https://www.d.umn.edu/~gshute/mips/instruction-types.xhtml>

5 Experimento

O experimento realizado foi a conversão de um arquivo binário contendo código de máquina - o qual foi fornecido pelo professor da disciplina - em um arquivo texto com as correspondentes endereços, valor em hexadecimal das instruções e as próprias instruções textuais em Assembly. Comparou-se a saída do *software* com a saída esperada - presente também em um arquivo de código Assembly fornecido pelo professor. Estão anexados, na entrega do trabalho, os respectivos arquivos contendo o *software*, os *inputs* e *outputs*.

6 Resultados

O *software*, para as instruções compreendidas no arquivo binário fornecido, teve êxito na tradução. Dessarte, pode-se afirmar que o objetivo principal - de construir um *disassembler* para a linguagem Assembly do processador MIPS foi cumprido.

Nessa toada, a realização do trabalho solidificou a compreensão dos participantes da linguagem Assembly, por meio da prática extensiva da programação nesta linguagem - cumprindo-se, assim, também os objetivos secundários da proposta.

7 Discussão

Foram discutidas, durante a implementação do trabalho, possíveis otimizações de programação e também as melhores maneiras de utilizar algoritmos convencionais de mais alto nível na linguagem Assembly. Tais discussões permitiram o incremento do funcionamento do trabalho a cada iteração, possibilitando a programação de um *software* funcional.

É inegável, contudo, que há diversas maneiras de construir um *disassembler*, e, assim, outras implementações devem vir a ser discutidas entre os discentes e o professor da disciplina. Buscar-se-á, pois, o aprofundamento da compreensão da linguagem e, ademais, de maneiras de tornar a programação mais limpa e eficiente.

8 Conclusões e perspectivas

Conclui-se, em suma, que o trabalho do *disassembler* do processador MIPS é integral ao aprendizado na disciplina de Organização de Computadores, ao passo que possibilita um contato dos alunos com a linguagem em um projeto real, complexo e exigente - nos quesitos de tempo, estudo e dedicação.

Referências

HENNESSY, John. Organização e Projeto de Computadores. Barueri, SP: Grupo GEN, 2017. E-book. ISBN 9788595152908. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595152908/>. Acesso em: 20 out. 2023.

MIPS Instruction Types. [S. l.]. Disponível em: <https://www.d.umn.edu/~gshute/mips/instruction-types.xhtml>. Acesso em: 23 out. 2023.