

1 - Introdução

O professor Giovani Baratto, docente da disciplina Organização de computadores, ELC1011, desafiou os seus alunos a aprofundarem seus conhecimentos sobre o funcionamento de um processador e como é feita a tradução da linguagem de alto nível para a de baixo nível, através primeiro trabalho da disciplina, um “disassembler”, um programa escrito na linguagem de máquina assembly, usando o software Mars 4.5. Este programa realiza a leitura de um arquivo binário e apresenta os seguintes dados: o valor do pc, a instrução em binário e a instrução traduzida para assembly.

2 - Objetivos:

Ao entender como um processador realiza as ordens dadas, cria-se uma base sólida para que novos conteúdos importantes sejam edificados ao longo do tempo.

3 - Revisão bibliográfica:

BARATTO, Giovani, 2.5_representando_instrucoes_computador.pdf, 22 páginas.

PATTERSON, David A.; HENNESSY, John L. Organização e projetos de computadores, a interface hardware/software. 5ª edição. 2017.

4 - Metodologia:

O trabalho foi realizado com implementação de código, testes, verificação dos resultados e aprimoramento. A implementação de código é a comunicação das necessidades do desenvolvedor, através da linguagem de programação assembly, com o software Mars. Em seguida foram realizados os testes e a verificação dos resultados obtidos, com a finalidade de descobrir se o programador está comunicando corretamente as suas necessidades ao simulador, caso contrário, iniciava-se a fase de aprimoramento de software.

5 - Experimento:

Foi feito com base no exemplo de código ex-000-091.asm da pasta de procedimentos. Os conhecimentos necessários para a realização deste trabalho foram obtidos em aula, através da pesquisa no livro base desta disciplina, discussões entre o grupo composto pelos graduandos Leandro Dalla Nora e Lorenzo Facco Comunello, estudo dos códigos presentes na pasta de procedimentos, dos slides disponibilizados pelo professor Giovani, lista de exercícios e da pesquisa de informações complementares na internet.

6 - Resultados:

0x00400000	0x27bdfff8	addiu \$sp , \$sp , 0x0000fff8
0x00400004	0x24080005	addiu \$t0 , \$zero , 0x00000005
0x00400008	0xafa80004	sw \$t0 , 0x00000004(\$sp)
0x0040000c	0x00082020	add \$sp , \$zero , \$t0
0x00400010	0x0c10000f	jal 0x0040003c
0x00400014	0x8fa80004	lw \$t0 , 0x00000004(\$sp)
0x00400018	0x00024821	addu \$sp , \$zero , \$v0
0x0040001c	0xafa20000	sw \$v0 , 0x00000000(\$sp)
0x00400020	0x00082021	addu \$sp , \$zero , \$t0
0x00400024	0x00092821	addu \$t0 , \$zero , \$t1
0x00400028	0x0c10001c	jal 0x00400070
0x0040002c	0x27bd0008	addiu \$sp , \$sp , 0x00000008
0x00400030	0x24020011	addiu \$v0 , \$zero , 0x00000011

0x00400034	0x24040000	addiu \$a0 , \$zero , 0x00000000
0x00400038	0x0000000c	syscall
0x0040003c	0x27bdfff8	addiu \$sp , \$sp , 0x0000fff8
0x00400040	0xafbf0004	sw \$ra , 0x00000004(\$sp)
0x00400044	0xafa40000	sw \$a0 , 0x00000000(\$sp)
0x00400048	0x14040002	bne \$zero , \$a0 , 0x00000002
0x0040004c	0x20020001	addi \$v0 , \$zero , 0x00000001
0x00400050	0x0810001a	j 0x00400068
0x00400054	0x2084ffff	addi \$a0 , \$a0 , 0x0000ffff
0x00400058	0x0c10000f	jal 0x0040003c
0x0040005c	0x8fa40000	lw \$a0 , 0x00000000(\$sp)
0x00400060	0x70821002	[ERRO] Erro instrução desconhecida
0x00400064	0x8fbf0004	lw \$ra , 0x00000004(\$sp)
0x00400068	0x23bd0008	addi \$sp , \$sp , 0x00000008
0x0040006c	0x03e00008	jr \$ra
0x00400070	0x27bdfffc	addiu \$sp , \$sp , 0x0000fffc
0x00400074	0xafa40000	sw \$a0 , 0x00000000(\$sp)
0x00400078	0x24020004	addiu \$v0 , \$zero , 0x00000004
0x0040007c	0x3c011001	lui \$at , 0x00001001(\$zero)
0x00400080	0x34240000	ori \$a0 , 0x00000000(\$at)
0x00400084	0x0000000c	syscall
0x00400088	0x8fa40000	lw \$a0 , 0x00000000(\$sp)
0x0040008c	0x24020001	addiu \$v0 , \$zero , 0x00000001
0x00400090	0x0000000c	syscall
0x00400094	0x3c011001	lui \$at , 0x00001001(\$zero)
0x00400098	0x3424000f	ori \$a0 , 0x0000000f(\$at)
0x0040009c	0x24020004	addiu \$v0 , \$zero , 0x00000004
0x004000a0	0x0000000c	syscall
0x004000a4	0x00052021	addu \$zero , \$zero , \$a1
0x004000a8	0x24020001	addiu \$v0 , \$zero , 0x00000001
0x004000ac	0x0000000c	syscall
0x004000b0	0x2404000a	addiu \$a0 , \$zero , 0x0000000a
0x004000b4	0x2402000b	addiu \$v0 , \$zero , 0x0000000b
0x004000b8	0x0000000c	syscall
0x004000bc	0x27bd0004	addiu \$sp , \$sp , 0x00000004
0x004000c0	0x03e00008	jr \$ra

7 - Discussão:

Os resultados obtidos são a tradução de cada instrução de linguagem de máquina para a linguagem assembly do simulador Mars.

8 - Conclusões:

Os resultados estão coerentes com o resultado que é esperado com o arquivo ex-000-012.asm presente na pasta dados. Os graduandos Leandro Dalla Nora e Lorenzo Facco Comunello aprofundaram seus conhecimentos relacionados ao funcionamento do processador.