

Atividade 6

Anaísa Forti da Fonseca

11811ECP012

Questão 1

Ao executar o assembler com o teste2.asm, o seguinte arquivo hack é gerado:

```
0000000000000000
1111110000010000
0000000000000001
1111000010010000
0000000000010000
1110001100001000
0000000000101010
1110110000010000
0000000000010000
1111010011010000
0000000000010000
1110001100000101
0000000000010001
1110111111001000
0000000000010010
1110101010000111
0000000000010001
1110111010001000
0000000000010010
1110101010000111
```

O programa realiza uma soma de ram_0 e ram_1 e verifica se ela é igual à 42.

a) ram_0 = 11 e ram_1 = 13

Ao inicializar a memória com estes valores, o resultado da soma é 24. Esse resultado é armazenado na memória de endereço 16. Na memória de endereço 17, é escrito o valor -1, indicando que a soma está incorreta (ou seja, é diferente de 42). Então, a memória é finalizada como:

ram_0 = 11

ram_1 = 13

...

ram_16 = 24

ram_17 = -1

b) ram_0 = 31 e ram_1 = 11

Ao inicializar a memória com estes valores, o resultado da soma é 42. Esse resultado é armazenado na memória de endereço 16. Na memória de endereço 17, é escrito o valor 1, indicando que a soma está correta (ou seja, é igual à 42). Então, a memória é finalizada como:

ram_0 = 31

ram_1 = 11

...

ram_16 = 42

ram_17 = 1

Questão 2

Instruções hacks podem conter símbolos que devem ser traduzidos para endereços durante todo o processo. A symboltable serve justamente para isso, ela foi projetada para criar e manter a correspondência entre os símbolos e seus significados (endereços ROM e RAM).

No assembler, duas passadas são necessárias pelo fato de que programas assembly podem usar símbolos antes que eles sejam definidos de fato. Na primeira passada, o assembler constrói a symboltable sem gerar nenhum código. Na segunda passada, todos os símbolos encontrados no programa já foram associados a locais de memória e registrados na symboltable. Com isso, o assembler pode substituir cada símbolo com seu endereço numérico correspondente e, então, gera o código final (em binário).