

Relatório

Laboratório 1: Assembly RISC-V

Grupo 3

Gabriel Alves Castro - 17/0033813

Henrique Mendes de Freitas Mariano - 17/0012280


Luthiery Costa Cavalcante - 17/0040631

Matheus Breder Branquinho Nogueira - 17/0018997

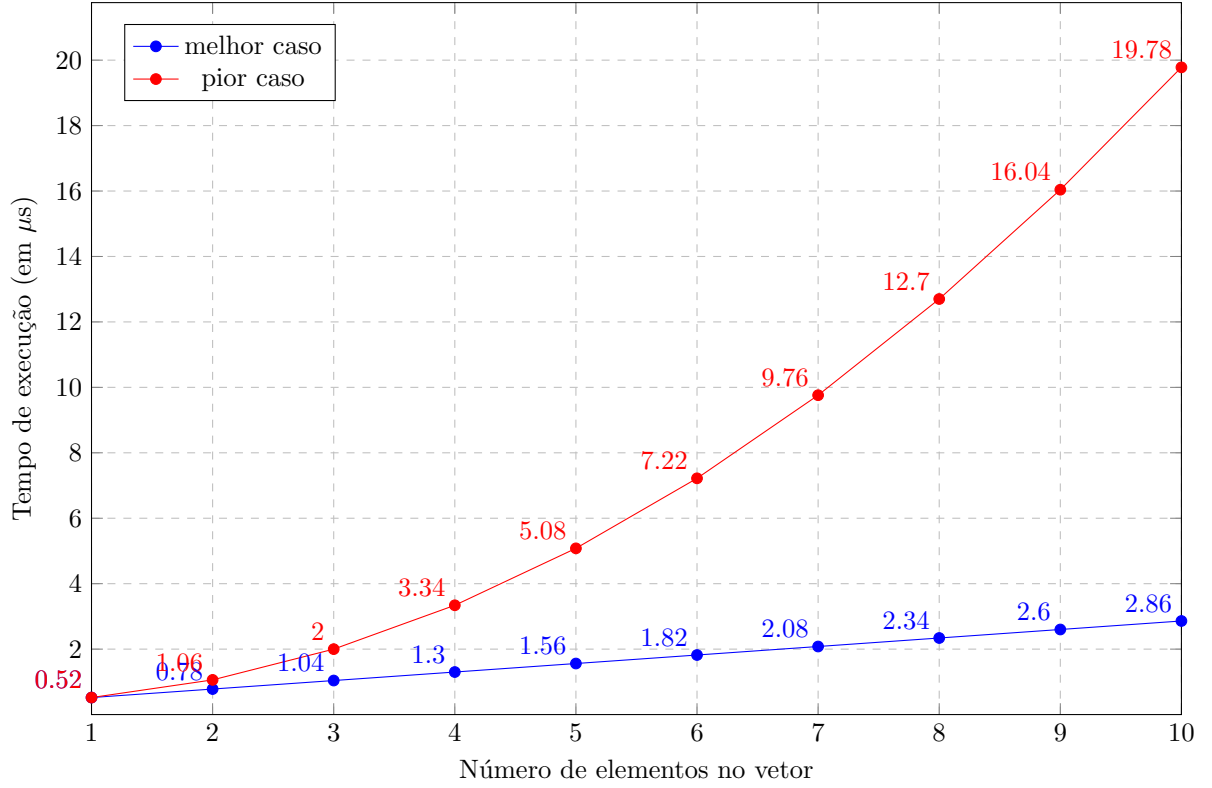
Universidade de Brasília - Dep. Ciência da Computação
Disciplina CIC 116394 - Organização e Arquitetura de Computadores - Turma A

1. Simulador/Montador RARS



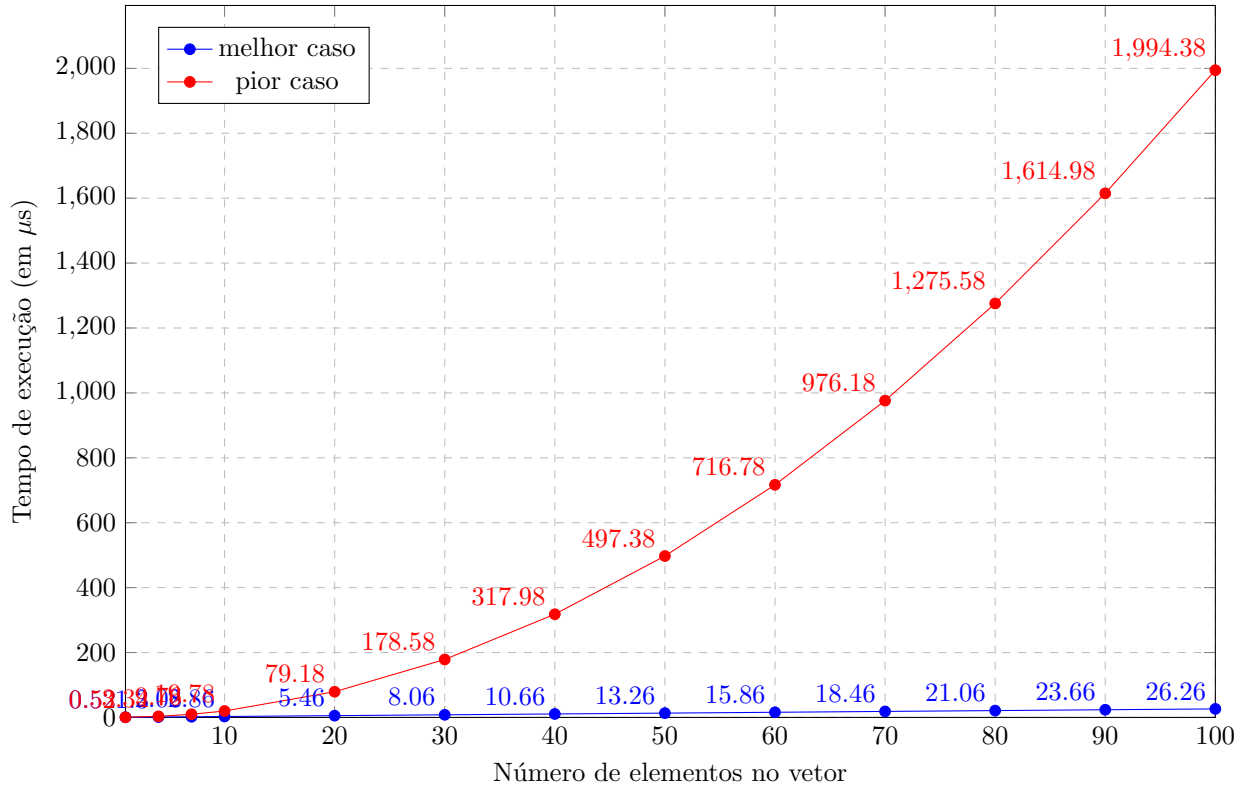
- 1.1 O programa executa no total 803 instruções e o código se estende do endereço 0x00400000 até o 0x0040012b. Como cada endereço aponta para 1 byte, o código contém $(12c)_{16} = 300$ bytes.
- 1.2 Comparando-o com a implementação do MARS podemos melhorar retirando o `nop` na linha 12 do código, o que resulta em um programa $\approx 1,001x$ mais rápido. 

Comparação: algoritmo de ordenação (no arquivo sort.s) ¹



1.3

Comparação: algoritmo de ordenação (no arquivo sort.s)



¹Considerando a equação $t_{exec} = I \cdot CPI \cdot T_{clk}$

Com o gráfico, constatamos que o tempo de execução de um vetor ordenado inversamente (o pior caso possível para o algoritmo de ordenação) cresce **exponencialmente** à medida que aumentamos a quantidade de elementos no vetor, enquanto o de um vetor já ordenado cresce linearmente a uma taxa bem baixa. Para qualquer outra distribuição em um vetor de n elementos, o tempo de execução se encontrará entre essas duas linhas do gráfico.

2. Compilador

- Disponível na pasta "Questao_2"
- 2.2) A implementação sortc possui o melhor desempenho, pois ao definir o vetor fora do escopo da main, o compilador não precisa se preocupar com diretivas que trabalhem com o acesso a um vetor que só pode ser visível na main.
- 2.3) É necessário modificar as instruções de acesso a memória geradas pelo gcc, para as instruções do rars, tais como "lw" e "sw" (ou outra quantidade de bits, dependendo do funcionamento do código e da arquitetura do computador, sendo que, a arquitetura do rars é de 32 bits). As diretivas que não são reconhecidas pelo Rars devem ser excluídas e substituídas por código com equivalência lógica. As operações aritméticas devem ser adaptadas para o formato reconhecido pelo rars, como por exemplo: Não pode haver addiw (apenas addi).

3. Raízes da equação

- Disponível na pasta "Q3" em *questao3.s*
- 3.4. a) $R(1) = 3.1415925$
 $R(2) = -3.1415925$
tempo: 188 ns
- b) $R(1) = 0.0$
 $R(2) = 0.0$
tempo: 137 ns
- c) $R(1) = -49.5 + 2.95804i$
 $R(2) = -49.5 - 2.95804i$
tempo: 178 ns
- d) $R(1) = 1234.0 + 5678.0i$
 $R(2) = 1234.0 - 5678.0i$
tempo: 178 ns
- e) $R(1) = \text{NaN}$
 $R(2) = -\text{Infinity}$
tempo: 188 ns
O que aconteceu pois $a = 0$, então, evidentemente haverá erro ao colocar $2a$ no denominador da fração.

4. Disponível na pasta "Q4" ...

