



Laboratório 1 **- Assembly RISC-V -**

Objetivos:

- Familiarizar o aluno com o Simulador/Montador Rars;
- Desenvolver a capacidade de codificação de algoritmos em linguagem Assembly;
- Desenvolver a capacidade de análise de desempenho de algoritmos em Assembly;

(2.0) 1) Simulador/Montador Rars

Faça o download e deszip o arquivo Lab1.zip disponível no Moodle.

1.1)(0.5) No diretório System_Rars, abra o Rars14_Custom1 e carregue o programa de ordenamento `sort.s`. Dado o vetor: $V[30] = \{9, 2, 5, 1, 8, 2, 4, 3, 6, 7, 10, 2, 32, 54, 2, 12, 6, 3, 1, 78, 54, 23, 1, 54, 2, 65, 3, 6, 55, 31\}$, ordená-lo em ordem crescente e decrescente (modifique o código), contar o número de instruções por tipo e o número total exigido pelo procedimento `sort`. Qual o tamanho em bytes do código executável? E da memória de dados usada?

1.2)(1.5) Considere a execução deste algoritmo em um processador RISC-V com frequência de *clock* de 50MHz que necessita 1 ciclo de *clock* para a execução de cada instrução ($CPI=1$). Para os vetores de entrada de 100 elementos já ordenados $V_0[100] = \{1, 2, 3, 4, \dots, 100\}$ e ordenados inversamente $V_i[100] = \{100, 99, 98, \dots, 2, 1\}$ calcule os tempos de execução para o ordenamento crescente dos vetores V_0 e V_i .

1.3)(0.0) Sabendo que as chamadas do sistema padrão do Rars usam um console (parte do SO) para entrada e saída de dados, execute o programa testeECALLv17b.s. Essas chamadas usam diretamente as ferramentas KDMIO e Bitmap Display.

(8.5) 3) Problema do Polígono Fechado

Dada um conjunto de pontos (x,y) , implementar um algoritmo que ligue graficamente os pontos de forma a gerar um polígono fechado, isto é, sem cruzamento de arestas.

(1.0) 3.1) Crie um procedimento `V=SORTEIA(N)` que sorteie aleatoriamente $N \geq 3$ conjuntos de coordenadas (x,y) , $x \in [0,319]$ e $y \in [0,239]$, e devolva um ponteiro `V` para a estrutura composta por `N` e o conjunto de vértices na memória de dados.

Para economizar memória armazene o par de coordenadas (x,y) em uma word `0x1234ABCD` onde `0x1234` é a coordenada x e `0xABCD` a coordenada y em hexadecimal.

Exemplo para um retângulo da tela $(0,0)$ $(319,0)$ $(319,239)$ $(0,239)$ temos na memória a seguinte estrutura `V`:

```
.data
```

```
V: .word 4, 0x00000000, 0x013F0000, 0x013F00EF, 0x000000EF
```

(1.0) 3.2) Crie um procedimento `ORDENA(V)` que ordene as `N` coordenadas (x,y) da estrutura `V` de forma que a ligação de um vértice ao seguinte não ocorra cruzamentos de arestas, formando um polígono fechado.

(1.0) 3.3) Crie um procedimento `POLIGONO(V,A)` que dado os vértices ordenados na estrutura `V`, desenhe no display gráfico (Bitmap Display Tool) um polígono fechado com `N` lados de cor `A`.

Dica: use o `ecall 47 draw line`.

(0.5) 3.4) Para a estrutura `V` definida abaixo, conte o número de instruções (`I`) necessárias para a execução dos seus procedimentos `ORDENA(V)` e `POLIGONO(V,A)`, com `A=0x00000007`.

```
V: .word 12, 0x00E6001C, 0x00A40052, 0x00A00042, 0x007B006D, 0x00760061, 0x004C007F, 0x003200D3, 0x005C00CB, 0x0080008C, 0x008600A4, 0x00A80068, 0x00AD007D
```

(5.0) 3.5) Faça um programa principal que chame as rotinas desenvolvidas para $N = \{3, 4, 5, \dots, 20\}$, com limpeza da tela (em branco) antes de cada chamada e de cor aleatória (diferente de branco). Apresente em vídeo o resultado obtido.

No arquivo `GrupoX_Lab1.zip` a ser enviado no Moodle coloque apenas o arquivo `GrupoX_Lab1.pdf` do relatório e o(s) arquivo(s) fonte(s) do seu programa.

Para a apresentação da verificação dos laboratórios (e projeto) nesta disciplina, crie um canal para o seu grupo no YouTube e poste os vídeos dos testes (sempre com o nome 'UnB – OAC Turma Verão - 2020/0 – Grupo X - Laboratório Y - <palavras-chaves que identifiquem este vídeo em uma busca>'), coloque os links clicáveis no relatório.

Passos do vídeo:

i) Apresente o grupo e seus membros; ii) Explique os procedimentos a serem realizados; iii) Apresente os testes solicitados; iv) Apresente suas conclusões.