

Atividade de Laboratório 7

[Objetivo](#)

[Descrição](#)

[Formato PBM](#)

[Cabeçalho](#)

[Imagem](#)

[Atividade](#)

[Infraestrutura](#)

[Entrega](#)

[Dicas e Observações](#)

Objetivo

O objetivo desta atividade é exercitar o uso de operações com *bits*, controle de fluxo, e acesso a vetores utilizando o conjunto de instruções da arquitetura RISC-V.

Descrição

Nesta atividade você deverá implementar um controle simples para guiar um “automóvel” a se manter dentro de uma pista representada por uma imagem PBM. O ponto inicial juntamente com a imagem serão fornecidos através da entrada padrão e você deve dizer a cada instante de tempo se o carro deve ir a direita, esquerda ou continuar reto para que chegue até o final da pista, sem sair ou esbarrar nela.

Formato PBM

O *Portable Bit Map* é um formato simples para representação de imagens monocromáticas ou em escala de cinza, que consiste em uma matriz ASCII, de tamanho $L \times C$, onde cada elemento da matriz corresponde a um pixel e seu valor denota a intensidade de cinza. A estrutura do arquivo para uma imagem de exemplo (escrita FEED) é mostrada abaixo:



```

P2
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

O arquivo PBM consiste de duas partes: o cabeçalho e a matriz de dados, com a imagem.

Cabeçalho

O cabeçalho é formado pelas 3 primeiras linhas do arquivo. A primeira linha ($P2$) mostra que a matriz de dados representa uma imagem em escala de cinza. A segunda linha é uma tupla, com dois números naturais C e L , representando a quantidade de colunas e linhas da matriz de dados (24 colunas - C - e 7 linhas - L - para este exemplo). Por fim, a terceira linha representa o valor máximo da escala de cinza (C_{MAX}). Os valores dos elementos da matriz de dados devem variar entre 0 a C_{MAX} (15, neste exemplo).

Imagem

Após a quarta linha, tem-se a imagem, representada por uma matriz de números inteiros (de 0 a C_{MAX}) de tamanho $L \times C$. Valores próximos a 0 tendem a ser mais escuros (preto) e valores próximos a C_{MAX} tendem a ser claros (branco).

Atividade

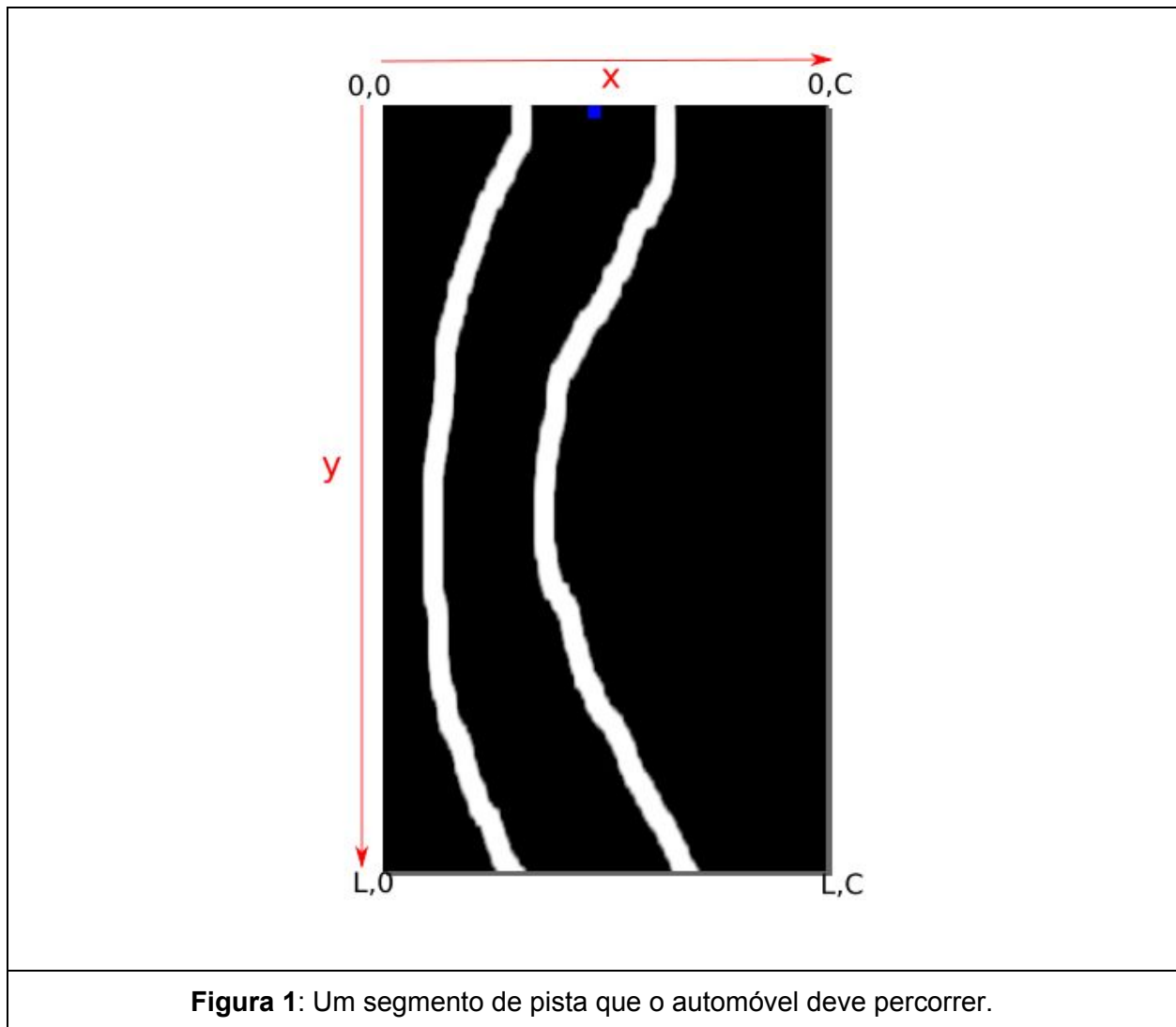
Nesta atividade, um automóvel hipotético equipado com sensores de luminosidade em sua dianteira, deseja se mover autonomamente dentro de uma pista sem esbarrar ou sair dela. Uma imagem no formato PBM representa o ambiente, uma matriz de tamanho $L \times C$ onde cada elemento da matriz próximo de C_{MAX} representa a faixa, como ilustrado na Figura 1.

O automóvel, ilustrado pelo ponto azul na imagem, é representado por um ponto (x, y) tal que $0 \leq x \leq C$, $0 \leq y \leq L$, que indica sua posição no ambiente. No início, este sempre se inicia com $y=0$ e se desloca em uma unidade na direção y . Antes de se mover na direção y , deve-se decidir se o automóvel também moverá uma unidade na direção x , isto é, se ele deve ir para esquerda, direita ou permanecer no centro. Cabe a você determinar se o valor de x deve ser incrementado/diminuído em uma unidade ou mantido o mesmo.

Para tomar esta decisão, antes que qualquer movimento seja tomado na direção y , uma linha da imagem PBM deve ser lida. A linha a ser lida, deve ser a linha de posição $y+1$ da imagem. Nesta linha, que é um vetor de C inteiros variando de 0 a C_{MAX} , o valor de cada

elemento corresponde a luminosidade obtida pelos sensores na frente do automóvel. Portanto, antes de cada deslocamento em y , os sensores devem ser lidos e a decisão de se mover ou não uma unidade na direção x deve ser tomada com base nos sensores e na posição do atual do automóvel, mantendo sempre o carro dentro da pista. Note que a primeira linha da imagem (quando $y=0$) deve ser ignorada.

Assim, o objetivo é que o automóvel se desloque da posição $y=0$ até a posição $y=L$ sem cruzar ou esbarrar nas linhas brancas.



Entrada

Para esta atividade você receberá os valores através da entrada padrão, sendo eles:

- Uma linha com dois números inteiros, separados por espaço, representando a posição inicial do carro (x_i, y_i) .
- Uma imagem no formato PBM, conforme na seção anterior.

Os arquivos de podem ser obtidos em: [pistas.zip](#). Note que, a posição inicial juntamente com a imagem estão contidos todos em um único arquivo, conforme listado acima.

Por exemplo, para seguinte entrada:

```
30 0
P2
64 200
255
0 0 0 0 0 0 0 0 130 240 240 240 240 132 0... 132 0 0 0 0 0 0 0 0
<outras 199 linhas >
...
```

Temos $x_i=30$, $y_i=0$, $C=64$, $L=200$ e $C_{MAX}=255$. A partir da quinta linha tem-se a matriz $L \times C$ da imagem PBM. Por padrão nesta atividade, C sempre terá o valor de 64 e C_{MAX} de 255. Desta forma, cada valor de luminosidade pode ser interpretado como um *byte* sem sinal, variando de 0 a 255, onde valores próximos de 0 são pretos e próximos de 255 são brancos.

Um pseudo-algoritmo para esta atividade é mostrado abaixo.

```
x, y = ler_linha()          // Le a primeira linha com os valores de xi, yi
ler_linha()                 // Linha 'P2' deve ser ignorada
L, C = ler_linha()          // Le a linha com os valores de L e C
C_MAX = ler_linha()         // Le a linha com o valor de C_MAX
ler_linha()                 // Ignora a primeira linha (y=0)

while(y < L-1)
    sensores = ler_linha()   // Lê uma linha com os valores dos sensores
    dir = decidir(sensores) // Decide se vai para esquerda, direita ou
                           // mantém-se no centro. Decidir retorna -1
                           // (esquerda), 1 (direita) ou 0 (centro)
    y = y+1                 // Incrementa y em 1
    x = x+dir               // Incrementa x com -1, 0 ou 1
    imprimir_pos(x, y)      // Imprime a nova posição
```

Saída

Após cada movimento tomado, as coordenadas (x, y) deve ser impressa na saída padrão. Note que, a coordenada inicial não deve ser impressa e, ao todo, $L-1$ linhas devem ser impressas. As linhas da saída devem ter o seguinte formato: 15 caracteres, onde os 4 primeiros são a palavra 'POS: ', seguido do valor de x , seguido de um espaço, seguido de mais 4 caracteres para o valor de y e por fim uma quebra de linha ($\backslash n$).

Por exemplo, uma saída seria:

```
POS: 0040 0001
POS: 0041 0002
POS: 0042 0003
POS: 0043 0004
...
```

POS: 0050 0127

POS: 0050 0128

Uma ilustração dos pontos da trajetória do automóvel pode ser vista na Figura 2. A linha vermelha representa a trajetória e os pontos vermelhos são tem seu valor amostrado em $y=100$ e $y=200$.

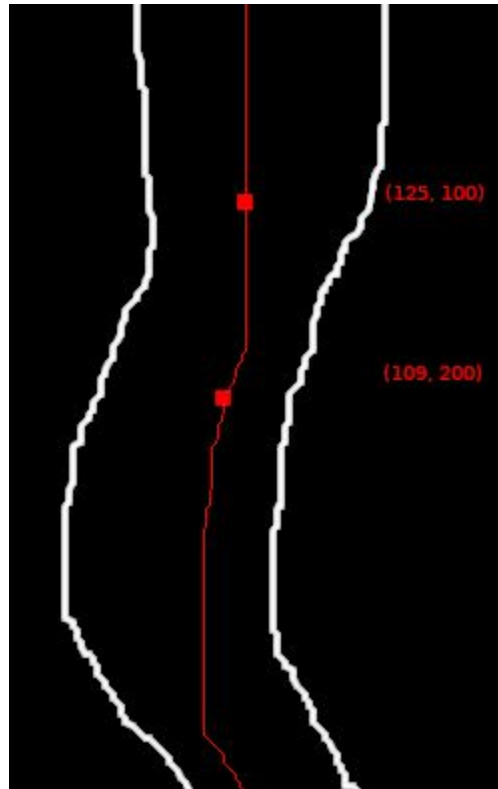


Figura 2: Ilustração da trajetória (em vermelho) do automóvel em uma pista

Infraestrutura

Para montar seu código em linguagem de montagem, use o comando:

```
clang -g -O0 --target=riscv32 -march=rv32gc -mabi=ilp32d  
-mno-relax arquivo_de_entrada.s -c -o arquivo_de_saida.o
```

Note o parâmetro na ferramenta clang: `--target=riscv32 -march=rv32gc -mabi=ilp32d`. Esse parâmetro indica que queremos que seja gerado um objeto que não é da arquitetura do seu computador (provavelmente x86_64) mas sim para RISC-V de 32 bits. Após essa etapa, tendo o arquivo objeto em mãos, podemos executar o ligador (*linker*) através do seguinte comando:

```
ld.lld -g arquivo_de_saida.o -o executavel.x
```

Entrega

Você deve submeter **APENAS** um arquivo denominado **raXXXXXX.s** (em que XXXXXX é seu RA com 6 dígitos) que contenha o código em linguagem de montagem.

Dicas e Observações

- As linhas de entrada contém caracteres no formato ASCII. Os caracteres na saída também devem estar no formato ASCII
- Nesta atividade, não haverá pistas com interseções, bifurcações ou curvas com ângulos agudos ou retos. As pistas sempre serão contínuas e seu tamanho (entre as duas faixas brancas) pode variar ao longo de y.
 - Note que isso não significa que de uma linha para outra haverá no máximo 1 pixel branco a mais horizontalmente. Suponha a pista a seguir onde 1 representa branco e 0 preto.

1	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1

A pista anterior é válida e poderá ser testada, nesta situação, se seu carro estivesse muito próximo do muro esquerdo, seu carro ficaria encurralado. Sua lógica deve evitar que esse cenário ocorra.

- O automóvel sempre iniciará com um valor de x dentro da pista.
- Valores dos sensores acima de 100 são considerados linhas brancas