Atividade de Laboratório 2

Objetivos

O objetivo desta atividade é a familiarização com a infraestrutura de simulação do computador IAS e programação em linguagem de máquina.

Descrição

Nesta atividade você deverá escrever um pequeno programa em linguagem de máquina do computador IAS para calcular a velocidade inicial de um projétil que precisa atingir um alvo a uma determinada distância, como ilustrado na Figura 1.

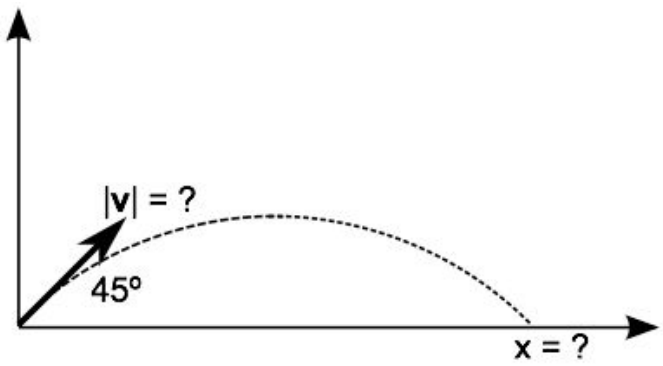

Figura 1: Ilustração da trajetória de um projétil no espaço bidimensional.

Suponha que o projétil seja lançado a 45º do eixo *x*. Então você pode usar a fórmula:

$$v = \sqrt{gx}$$

Para o cálculo (consulte o apêndice para a explicação), em que *v* é a velocidade inicial, *g*, a aceleração da gravidade, e *x*, a distância horizontal do alvo em relação à origem do disparo. Assuma *g* igual a 10 m/s^2. Os cálculos do programa devem usar sempre inteiros. Para testar o seu programa, descubra qual a velocidade inicial, em m/s, para atingir um alvo a 3500 m.

Note que o IAS não possui instrução para a raiz quadrada. Por isso, você deve calcular uma expressão que aproxima o valor da raiz quadrada para números inteiros. Tal expressão aproxima um chute para o valor da raiz quadrada. Seja *k* um chute aproximado para $\sqrt{y}$. Comece supondo que

$$k = \frac{y}{2}.$$

Um passo para aproximar *k* do valor real de $\sqrt{y}$ é calcular um novo *k′* com a fórmula:

$$k' = \frac{k + \frac{y}{k}}{2}$$

A cada vez que você calcular a expressão acima, você descobrirá um valor de *k* mais próximo de $\sqrt{y}$. Para os valores usados neste exercício, uma boa aproximação consiste em computar essa expressão por 10 vezes.

Programando o IAS

No capítulo 5 do documento "Programando o computador IAS" (disponível no Moodle) você encontrará detalhes sobre as operações do IAS e sua codificação em linguagem de máquina e no final desse documento existe um sumário de todas as instruções.

Antes de iniciar a simulação, é necessário carregar o programa na memória principal do IAS. Para carregar as instruções e dados iniciais você deve fornecer um "mapa de memória" na área indicada do simulador. O mapa de memória contém uma lista de valores que serão utilizados para iniciar a memória da máquina antes de a execução iniciar. O mapa de memória é um arquivo no formato texto no qual cada linha especifica o endereço e o valor que deve ser gravado na palavra de memória associada ao endereço. Cada linha consiste de uma coluna representando o endereço e outra representando o valor armazenado na palavra de memória associada àquele endereço, conforme o exemplo abaixo:

```
000 0110015000
001 0D40000000
# ...
100 00000088D8
```

Esse programa-exemplo copia o conteúdo da posição `0x100` de memória para o registrador AC (operação 0x01), faz um deslocamento para a direita (operação 0x15) no dado recém-obtido e por fim realiza um salto para o endereço de memória 0x400 (operação 0D). Note que todos os endereços e valores no mapa de memória estão representados no sistema hexadecimal.

Você pode introduzir espaços em branco entre os caracteres da segunda coluna (dados/instruções), linhas em branco e comentários para facilitar a leitura. No entanto, cada linha que especifica um conteúdo da memória deve conter exatamente 13 dígitos hexadecimais, 3 para o endereço e 10 para o conteúdo da memória. O trecho de código abaixo é equivalente ao anterior:

```
000 01 100 15 000
001 0D 400 00 000   #isso eh um comentario
# ...
100 00 00 00 88 D8
```

Para testar o seu programa você deve usar o Simulador do IAS. A Figura 2 apresenta uma descrição dos elementos da interface do simulador.

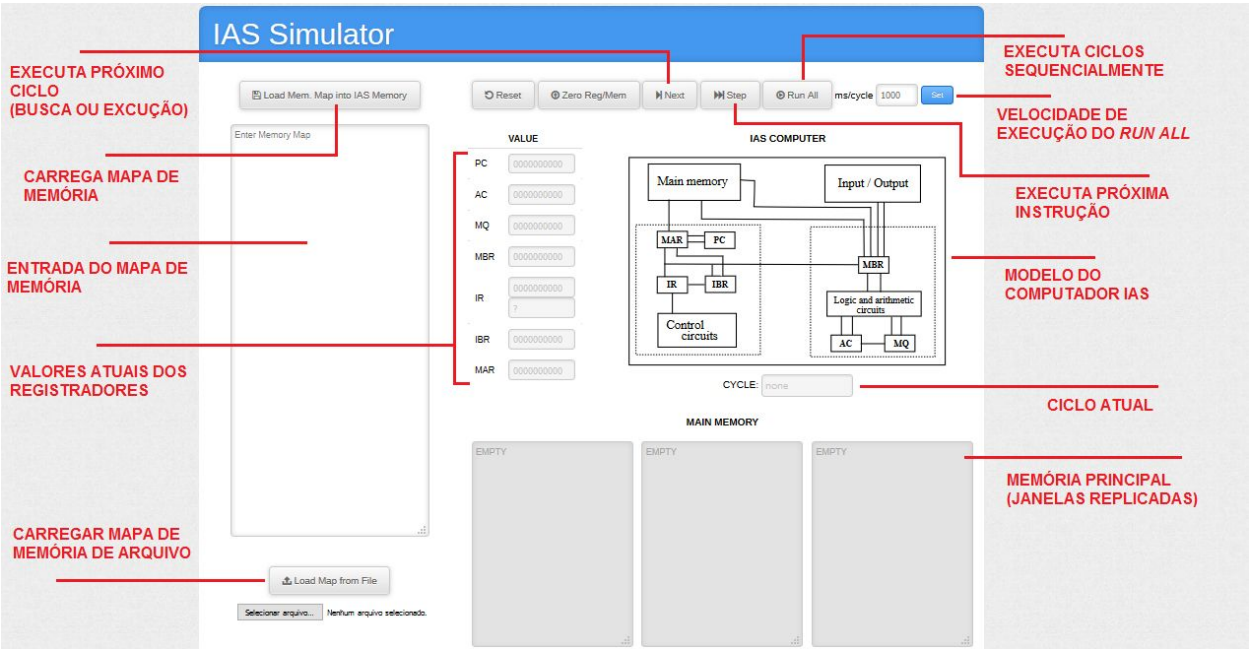

Figura 2: Descrição dos elementos da interface do simulador.

Note que existem três caixas indicadas como memória principal. Elas apresentam o mesmo conteúdo, carregado a partir do mapa de memória, replicado para facilitar a visualização de diversas áreas da memória. Após carregado o programa, pode-se optar por um dos três modos de execução: ciclo a ciclo (através do botão *Next*); instrução a instrução (botão *Step*); ou execução automática (botão *Run All*). Uma vez iniciado o programa, o botão *Reset* reinicia a aplicação, habilitando os demais modos, assim como o carregamento de um novo mapa de memória. Para terminar a execução, você pode realizar um salto para o endereço 0x400 - veja que esse valor representa o número 1024 no sistema decimal, e sabemos que o IAS tem 1024 palavras de memória, que são os endereços de 0x000 a 0x3FF no mapa de memória. Ou seja, ao efetuar um salto para o endereço 0x400, estamos desviando o fluxo de execução para uma posição de memória inexistente, portanto o simulador irá parar a execução e apresentar a mensagem "*PROGRAM ABORTED*", ilustrada na Figura 3.

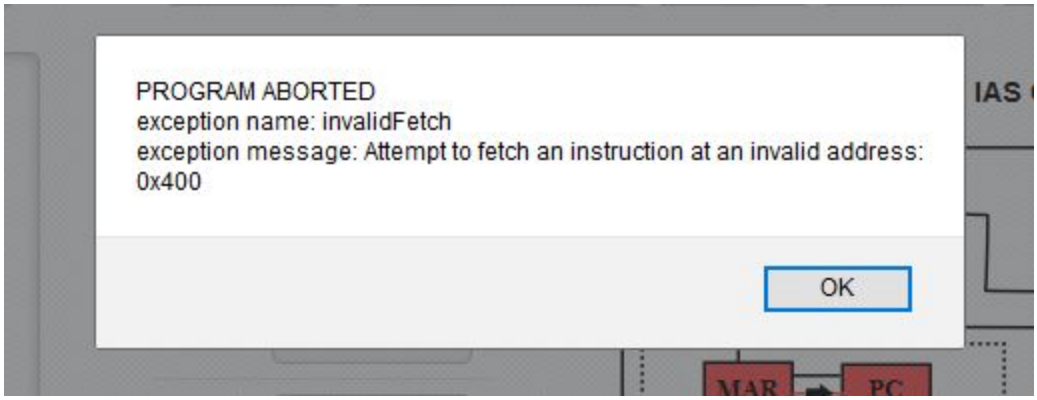

Figura 3: Caixa de texto indicando que o fluxo de execução alcançou um endereço inválido.

Dicas

- Não se esqueça de que o **endereço do início da execução é 000**, e não 001.
- Ao saltar para o endereço 0x400, lembre-se de preencher de fato o campo de endereço da instrução com o valor 400, ou este ficará zerado e o programa entrará em *loop* infinito.
- Atente-se ao escrever o mapa de memória. Programas com o caractere **O (letra)** no lugar do **0 (número zero)** causarão erros de execução, com difícil detecção.
- Procure primeiramente entender as instruções do IAS e escreva seu programa uma instrução por linha, comentando cada linha e entendendo a lógica. Só depois converta esse código para linguagem de máquina usando 2 instruções de 20 *bits* por palavra de memória, no formato do mapa de memória.
- Para facilitar a implementação, você pode considerar que a multiplicação gera apenas resultados de até 40 *bits*.

Entrega

- O exercício a ser submetido é o mapa de memória com o programa para resolver o problema da balística descrito acima.
- **Você deve submeter APENAS um arquivo no Moodle, chamado raXXXXXX.hex, em que XXXXXX é seu ra com 6 dígitos.**
- Seu programa deve terminar com um salto para o endereço 0x400, conforme comentado acima.
- Ao final da execução do seu programa, o valor da velocidade do projétil deve estar no registrador AC.
- O valor da **"entrada" (x)** deverá estar na posição de memória **0x110**. Não utilize esta posição de memória para outros fins. O *script* de correção irá ajustar o mapa de memória para que esta posição contenha o valor da entrada em cada teste.
- Use comentários e os espaços à vontade, pois além de facilitar a leitura de seu arquivo, caso seja necessário, é uma boa prática de programação.

Avaliação

Para avaliar seu programa, executaremos o mesmo com 10 entradas diferentes e verificaremos o resultado armazenado no registrador AC.

A avaliação será realizada com o mecanismo de automação de testes do simulador IAS.

Este mecanismo toma como entrada um mapa de memória (*memory map*), com o programa a ser testado, e um vetor de testes (*test object*), que define os parâmetros de entrada e saída do teste. Os parâmetros de entrada consistem em valores que serão inseridos na memória ou registradores do simulador enquanto que os parâmetros de saída consistem em valores que serão verificados ao término da execução. Por exemplo, o primeiro teste do vetor de testes abaixo indica que a posição de memória 0x110 será iniciada com o valor 0 antes do início da simulação e, ao término da simulação, o valor do registrador AC será comparado ao valor 0.

O vetor de testes a seguir define os 5 casos de testes públicos desta atividade de laboratório. **OBS: O VETOR DE TESTES ABAIXO É O VETOR DE TESTES CORRETO.**

```
[
   {
      "input":  [{"where": "ram", "position": "0x110", "value": 34}],
      "output": [{"where": "reg", "position": "ac",    "value": 18}]
   },
   {
      "input":  [{"where": "ram", "position": "0x110", "value": 1}],
      "output": [{"where": "reg", "position": "ac",    "value": 3}]
   },
   {
      "input":  [{"where": "ram", "position": "0x110", "value": 3218}],
      "output": [{"where": "reg", "position": "ac",    "value": 179}]
   },
   {
      "input":  [{"where": "ram", "position": "0x110", "value": 28306}],
      "output": [{"where": "reg", "position": "ac",    "value": 532}]
   },
   {
         "input":  [{"where": "ram", "position": "0x110", "value": 3}],
         "output": [{"where": "reg", "position": "ac",      "value": 5}]
    }
]
```

Para saber se sua solucão está passando por todos os testes, inclusive os fechados, utilize esse link: https://cloud.lmcad.ic.unicamp.br/test

Apêndice: Desenvolvimento teórico

Nota: esse apêndice tem caráter de curiosidade, não sendo necessária sua leitura para compreensão do laboratório.

Seja v_{y0} a velocidade vertical inicial do projétil, v_x a velocidade horizontal, g a aceleração da gravidade e t o tempo. As equações de movimento são:

$$y = v_{y_0} t - \frac{gt^2}{2} \quad (1)$$

e

$$x = v_x t \quad (2)$$

Isolando o tempo na equação (2), temos:

$$t = \frac{x}{v_x}$$

Fazemos então $y = 0$ pois estamos interessados no momento em que o projétil atinge o chão (altura igual a zero) e simplificamos a equação 1:

$$t(v_{y_0} - \frac{gt}{2}) = 0 \Rightarrow v_{y_0} = \frac{gt}{2} \quad (3)$$

Substituindo 3 em 4 temos:

$$v_{y_0} = \frac{gx}{2v_x} \quad (4)$$

Agora, recompomos as componentes da velocidade inicial (v_x e v_{y0}) para ângulo e norma do vetor v, obtendo

$$v^2 \sin(2\theta) = gx \quad (5)$$

A partir da equação 5, percebemos que o ângulo de 45º é o que minimiza a velocidade inicial necesssária para atingir o alvo, pois leva o termo multiplicativo da velocidade a 1 (qualquer outro ângulo tem o efeito de prejudicar a velocidade inicial). Então, usando este ângulo, obtemos a fórmula:

$$v = \sqrt{gx} \quad (6)$$