

UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

116394 ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Trabalho III: Programação Assembler

OBJETIVO

Este trabalho objetiva a prática da programação em *assembler* do MIPS. O trabalho consiste no desenvolvimento de algumas funções para manipulação de imagens. As operações a serem implementadas são:

- leitura de um arquivo binário com uma descrição de uma figura colorida, no formato RGB utilizado pelo MIPS
- exibição da figura no mostrador gráfico do MIPS
- leitura e escrita de pixels da imagem
- conversão para cinza

Estas operações deverão ser disponibilizadas a partir de uma interface textual com o usuário. O arquivo contendo a imagem deve chamar-se “image.raw”.

DESCRIÇÃO

A. Display gráfico mapeado em memória

- O MARS oferece, dentre as suas ferramentas de apoio ao desenvolvimento de aplicações em *assembler* MIPS, uma janela gráfica baseada em *pixels*. Suas principais características são as seguintes:
 - Resolução configurável, *default* 512 x 256 *pixels*.
 - Cada *pixel* representado por uma palavra de 32 bits, no formato RGB, um *byte* para cada cor. *Red* = 0x00FF0000, *Green* = 0x0000FF00, *Blue* = 0x000000FF. O *byte* mais significativo é ignorado.
 - *Display* mapeado em memória. O ponto superior esquerdo da tela corresponde ao pixel com coordenadas (0, 0). A coordenada *x* cresce para a direita e a coordenada *y* cresce para baixo. Uma imagem bidimensional é vista como uma matriz de *pixels*, onde *x* designa a linha e *y* a coluna da matriz. Uma imagem de 512 x 512 no MARS é constituída, portanto, de 512 linhas com 512 palavras de 4 *bytes* em cada linha.
 - O endereço correspondente ao primeiro pixel é configurável. Opções: *global data*, *global pointer*, *static data*, *heap*, *memory map*.
 - O desenho de um pixel na tela é realizado pela escrita de uma palavra contendo a descrição de sua cor RGB na posição de memória correspondente.
- Configuração do MARS para este trabalho:
 - Memória: *default*
 - Unidade de largura em *pixels*: 4
 - Unidade de altura em *pixels*: 4
 - Largura do display: 256
 - Altura do display: 256
 - Endereço base do display: 0x10040000 (*heap*)

B. Leitura de uma Imagem em Arquivo

- Dentre as chamadas de sistema do MARS existem funções para acesso à arquivos. A imagem fornecida em arquivo é “image.raw”, no formato RGB com 3 bytes por *pixel*, em resolução de 64 x 64 *pixels*. Não há nenhum cabeçalho no arquivo, apenas os *pixels* da imagem. Cabe salientar que, no arquivo, os *pixels* são representados por 3 *bytes* consecutivos, sem espaço entre eles. Ao serem escritos na área de memória do *display* deve-se convertê-los para o formato de 4 *bytes* do MARS.

C. Interface com o usuário

- Utilizando as chamadas do sistema, escrever uma mensagem com um cardápio de opções para o usuário. Ex:

1. Load Img

Carregar imagem do arquivo “image.raw” para a área de exibição (*display*)

2. Ler pixel

Função *get_pixel(x, y)* imprime na console os valores R, G e B do pixel indicado

3. Seta pixel

Função *set_pixel(x, y, val)* solicita ao usuário os valores das componentes R, G e B via teclado (números entre 0 e 255) e escreve o valor do pixel no endereço (x, y) indicado.

4. 255 tons de cinza

Converte a imagem para tons de cinza. Para cada pixel, deve-se atribuir a cada componente de cor o valor médio de R, G e B, ou seja, $R = G = B = \text{médica}(R, G, B)$

5. Encerra

ENTREGA

Entregar no Moodle um arquivo compactado tendo como título o nome do aluno e o seu número de matrícula, ex: fulano_00123456.zip. Deve conter:

- Relatório de implementação:
 - *cabeçalho*: com título do trabalho, nome e matrícula do aluno, identificação da turma
 - *objetivo*: summarize os objetivos principais do trabalho
 - *documentação do código*: indique quais as funções implementadas (todas), seus parâmetros e funcionamento.
- código assembler: arquivo asm

Prazo de entrega: 29/09/16