
SSC0902 - Prof. Fernando Osório / 2019-2 Trabalho Prático Final (TP-Final)

Definição do Trabalho Prático Final (MIPS)
Data Especificação: 31/10/2019 - versao.01

>>>>>> TRABALHO PADRÃO <<<<<<

CALCULADORA MIPS

Implementar uma calculadora capaz de trabalhar com:

- I: Inteiros 32 bits com sinal (nros. negativos em complemento de 2)
- F: Float de 32 bits
- D: Double de 64 bits

Realizando as operações de soma, subtração, multiplicação, divisão e troca de sinal:

Operações: + (soma) - (subtrai) * (multiplica) / (divide) ! (inverte sinal)

Como deve funcionar a Calculadora-MIPS:

- 1. Selecionar o Tipo de Dado:
 - Mensagem para o usuário: "DataType: "
 - Ler resposta: "I" ou "F" ou "D" (em maiúsculas ou minúsculas)
- 2. Selecionar o Tipo de Operação:
 - Mensagem para o usuário: "Operation: "
 - Ler resposta: "+", "-", "*", "/" ou "!"
- 3. Entrada de dados:
 - Ler o Valor1 do tipo selecionado (I,F,D)
 - Exibir o Valor1 em Hexadecimal (32 ou 64 bits)
 - Ler o Valor2 do tipo selecionado (I,F,D)
 - Exibir o Valor2 em Hexadecimal (32 ou 64 bits)
- 4. Realizar a operação selecionada: soma, subtrai, multiplica, divide ou inverte sinal Obs: Inverte o sinal usa apenas o Valor1, o Valor2 deve ser sempre 0
- 5. Exibir o resultado da operação:
 - Exibir a mensagem para o usuário: "Result: "
 - Exibir o resultado de acordo com o tipo do dado (I,F,D)
 - Exibir o resultado em Hexadecimal (32 ou 64 bits)
 - Caso ocorra "estouro" de precisão aritmética, escrever a mensagem: "Overflow"
- 6. Perguntar se o usuário deseja realizar uma nova operação: Repeat? Y = Yes, N = No Se o usuário responder 'Y' (minúsculo ou maiúsculo), voltar ao item 1 Se o usuário responder 'N' (minúsculo ou maiúsculo), ir para o item 7 Se não for nem 'Y'/'y'/'N'/'n' exibir uma mensagem de erro e repetira a pergunta (item 6)
- 7. Exibir a mensagem final:
 - Mensagem para o usuário: "End of Program"

PROCESSAMENTO DE IMAGENS		

Jogo MIPS (usando o Simulador MARS)		
Processamento de Imagens OU		
<<<<<<<<		
>>>> TRABALHOs ALTERNATIVOS <		
>>>>>>>>>>		

- 1. Ler um arquivo em disco contendo um arquivo com uma imagem em formato PGM.

 Será lida uma imagem de tamanho 100 x 100 pixels armazenada em um arquivo em formato texto.

 Esta imagem deverá ser armazenada em memória. Ver detalhes sobre os arquivos PGM abaixo.
- 2. Uma vez lida a imagem para a memória, deve ser executado um processamento de imagens para a detecção de bordas, usando um filtro do "kernel" ou "convolution" de tamanho 3x3. A matriz do filtro deve ser de tamanho 3x3 podendo ser usado um filtro do tipo "Sobel" por exemplo. Exemplos de como realizar uma convolução de uma matriz 3x3 sobre uma imagem: http://setosa.io/ev/image-kernels/

Conceito de Convolução com Kernels:

https://en.wikipedia.org/wiki/Kernel (image_processing) [Multiplicação de Matrizes] https://en.wikipedia.org/wiki/Sobel_operator_Filtro Sobel

- * Uma convolução usando um kernel (filtro), é uma multiplicação de uma pequena matriz (3x3 por exemplo) que vai sendo multiplicada por cada pixel da imagem, e o resultado da multiplicação do kernel 3x3, centrado em um pixel, é armazenado formando uma nova imagem. O kernel vai sendo deslocado pixel a pixel, onde as multiplicações pela matrz 3x3 ocorrem coluna por coluna e linha por linha (desloca e multiplica o kernel por cada um dos pixels da imagem). Atenção as bordas, para que o Kernel não fique para fora da imagem, você pode avançar o kernel de modo a evitar que ele fique para fora das bordas da imagem (não realiza a convolução nos pixels da borda).
- Salvar a imagem obtida com a aplicação do filtro em um arquivo texto em formato PGM tipo
 P2 (o mesmo formato que foi lido).

Ref. Complementares:

http://www.songho.ca/dsp/convolution/convolution2d_example.html

https://rosettacode.org/wiki/Image convolution

https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-

584bc134c1e2 (ver item 2.1: Convolution, a base das Redes Neurais Convolucionais – Deep CNNs).

>> Sobre o formato PGM: http://paulbourke.net/dataformats/ppm/
http://paulbourke.net/dataformats/ppm/
http://paulbourke.net/dataformats/ppm/

Considere o Formato P2, ou seja, imagens em tons de cinza, com 1 byte por pixel (nível de intensidade – tons de cinza). Veja o exemplo de uma imagem PGM no link acima. As imagens PGM podem ter diferentes resoluções e níveis de quantização (quantidade de tons) diferentes, mas no trabalho vamos considerar o formato PGM tipo P2 com 255 tons (valores dos pixels de 0 a 255) e sempre com uma resolução de 100 x 100 pixels. Exemplos de arquivos textos contendo imagens PGM serão fornecidos pelo professor.

Ferramentas que permitem visualizar (e salvar) arquivos em formato PGM:

IrfanView – https://www.irfanview.net/

ImageJ - https://imagej.nih.gov/ij/

Trabalho Alternativo === OU ===			

Implementar - Snake	um jogo usando o simulador MARS		
Referências:	https://www.google.com/search?q=play+snake https://www.youtube.com/watch?v=MSQaWF6og https://www.youtube.com/watch?v=5v1Pz0bzO1g	Ascii Snake Game 8086 Snake Game	
- Pong (Ping-P	ong / Tennis) [Single user ou Dual Players]		
Referência:	https://en.wikipedia.org/wiki/Pong https://www.youtube.com/watch?v=GqEVbjWnqzw https://pong-2.com/	Ascii Pong Game	
=======			
F.Osório Out.2019			
=======			