

Entrega número 1

Tarefa 3.4 e 3.5

3.4

Desenvolvemos as funções da seguinte forma:

3.4.2 Função `int asciitoint(char * cval)`: Esta função recebe como argumento(em `a0`) um endereço de uma array de 3 caracteres, os quais vai percorrer com apoio do comando **`lbu`**(load byte unsigned) e subtrair pelo inteiro 48, transformando assim um algarismo ASCII no inteiro que representa. Para ver se o símbolo que o antecede é + ou - , a função subtrai 43(Decimal do ASCII "+") ao primeiro carácter, e, se der 0, conclui-se que é de facto um símbolo +, caso contrário, será um - ou qualquer outro carácter.

3.4.3 Função `setpixel(int linha, int coluna, unsigned red, unsigned green, unsigned blue)`: Como esta função recebe mais de 4 argumentos, temos de usar o stack pointer para passar o quinto, guardando então o byte que guarda a informação da cor azul em **`0($sp)`**. Basicamente esta função vai carregar o endereço do display global num registo, e preparar a informação a enviar para este endereço através de comandos como **`sll`** e **`ori`**. Depois, irá imprimir nas coordenadas (x,y) este byte com as informações de cor dadas nos argumentos, sendo x,y as linhas e as colunas, respectivamente.

3.4.4 Função `clear(unsigned red, unsigned green, unsigned blue)`: Esta função recebe nos registos `a0..a2` os dados em RGB da cor que queremos preencher o display todo. Depois, irá calcular o número de pixels que vai pintar($64*64$) e vai percorrer o display e pintá-lo, até a um contador chegar a esse número, indicando o fim do corrimento.

3.4.5 Função `drawaxis()`: Basicamente esta função utiliza a função `setpixel` com a variável linhas e as 3 cores fixas, sendo estas 32 e 0,0,0, respectivamente. Assim, desenha uma linha negra a meio do display.

3.4.6 Função `plot(int *valores, unsigned red, unsigned green, unsigned blue)`: Esta função recebe o endereço de uma array de inteiros, ao que cada inteiro vai corresponder a linha em que o píxel correspondente terá de ser impresso. A coluna do mesmo vai aumentar de 1 em 1, para preencher o ecrã inteiro. Os restantes argumentos são apenas códigos RGB para definir a cor em que vamos imprimir os pixels.

3.4.6 main.s: É a função main do programa, que irá imprimir para o utilizador as suas escolhas(menus), receber os valores de RGB, bem como o nome do ficheiro, abrir o ficheiro e transformar os conteúdos das strings contidas nos ficheiros .txt em inteiros, para então criar uma array destes, e poder passar à função `plot`. Procura também o carácter "\n" no ficheiro e substitui-o por 0, de forma a evitar resultados inesperados.

3.5 3_5.asm: Este programa tem como objectivo ler um ficheiro do tipo PGM e mostrar a imagem no display, caso a imagem fosse menor que o display. Os conhecimentos aplicados são idênticos aos das perguntas anteriores e foi conseguido aplicar tudo. A parte de desenho no display é composta por 3 partes diferentes. Uma primeira parte que ignora os 13 primeiros caracteres lidos, uma segunda parte que calcula a intensidade de cinza e faz display dela e por fim uma que enche o resto do display com preto.