

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*\*\*



**SOICT**

**BÁO CÁO**  
**FINAL PROJECT**  
**HỌC PHẦN: THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**Giảng viên hướng dẫn : TS. Phạm Ngọc Hưng**

**Sinh viên thực hiện : Phạm Văn Anh**

**Mã lớp : 139365**

**Hà Nội, tháng 7 năm 2023**

## BÀI 9: VẼ HÌNH BẰNG KÝ TỰ ASCII

Cho hình ảnh đã được chuyển thành các ký tự ascii như hình vẽ. Đây là hình của chữ DCE có viền \* và màu là các con số.

```
*****
*****
*222222222222222*
*22222*****22222*
*22222*      *22222*
*22222*      *22222*      *****
*22222*      *22222*      **11111*****111*
*22222*      *22222*      **1111**      **
*22222*      *222222*      *1111*
*22222*****222222*      *11111*
*2222222222222222*      *11111*
*****      *11111*
-----      *1111**
/  o  o  \      *1111****      *****
\    >  /      **111111***111*
-----      *****      dce.hust.edu.vn
```

- Hãy hiển thị hình ảnh trên lên giao diện console (hoặc giao diện Display trong công cụ giả lập Keyboard and Display MMIO Simulator)
- Hãy sửa ảnh để các chữ cái DCE chỉ còn lại viền, không còn màu số ở giữa, và hiển thị
- Hãy sửa ảnh để hoán đổi vị trí của các chữ, thành ECD, và hiển thị. Để đơn giản, các hoạt tiết đính kèm cũng được phép di chuyển theo.
- Hãy nhập từ bàn phím ký tự màu cho chữ D, C, E, rồi hiển thị ảnh trên với màu mới. Chú ý: ngoài vùng nhớ lớn chứa ảnh được chứa sẵn trong code, không được tạo thêm vùng nhớ mới để chứa ảnh hiệu chỉnh

## 1. Mã nguồn

```
1  .data
2      String1: .asciiz "                      ***** \n"
3      String2: .asciiz "*****                *3333333333333* \n"
4      String3: .asciiz "2222222222222222*        *33333***** \n"
5      String4: .asciiz "22222*****22222*        *33333* \n"
6      String5: .asciiz "22222*          *22222*        *33333***** \n"
7      String6: .asciiz "22222*          *22222*          ***** *3333333333333* \n"
8      String7: .asciiz "22222*          *22222*          **11111*****111* *33333***** \n"
9      String8: .asciiz "22222*          *22222*          **1111**          ** *33333* \n"
10     String9: .asciiz "22222*          *22222*          *1111*        *33333***** \n"
11     String10: .asciiz "22222*****22222*        *1111*        *3333333333333* \n"
12     String11: .asciiz "2222222222222222*        *1111*        ***** \n"
13     String12: .asciiz "*****                *1111*        \n"
14     String13: .asciiz "          ---          *1111**        \n"
15     String14: .asciiz "          ( o o )          *111****        ***** \n"
16     String15: .asciiz "          ( > )          **11111***111* \n"
17     String16: .asciiz "          -----          *****        dce.hust.edu.vn \n"
18     Message0: .asciiz "-----IN CHU-----\n"
19     Case1_text: .asciiz " 1. In ra chu\n"
20     Case2_text: .asciiz " 2. In ra chu rong\n"
21     Case3_text: .asciiz " 3. Thay doi vi tri\n"
22     Case4_text: .asciiz " 4. Doi mau cho chu\n"
23     Thoat: .asciiz " 5. Thoat\n"
24     Nhap: .asciiz " Nhap chu nang muon thuc hien gia tri: "
25     Chu_D: .asciiz " Nhap mau cho chu D(0->9): "
26     Chu_C: .asciiz " Nhap mau cho chu C(0->9): "
27     Chu_E: .asciiz " Nhap mau cho chu E(0->9): "
28 .text
29     # Set up color default characters
30     addi $t5, $zero, 50      # Mau hien tai cua chu D
31     addi $t6, $zero, 49      # Mau hien tai cua chu C
32     addi $t7, $zero, 51      # Mau hien tai cua chu E
```

Khai báo mảng hiện thị các ký tự trên cửa sổ Run I/O, menu chức năng và mã màu cho các ký tự DCE

```

33  main:
34      li $v0, 4
35      la $a0, Message0          # In ra Menu
36      syscall
37      li $v0, 4
38      la $a0, Case1_text        # In ra Case1
39      syscall
40      li $v0, 4
41      la $a0, Case2_text        # In ra Case2
42      syscall
43      li $v0, 4
44      la $a0, Case3_text        # In ra Case3
45      syscall
46      li $v0, 4
47      la $a0, Case4_text        # In ra Case4
48      syscall
49      li $v0, 4
50      la $a0, Thoat             # In ra Thoat
51      syscall
52      li $v0, 4
53      la $a0, Nhap              # In ra Nhap
54      syscall
55
56      li $v0, 5
57      syscall

59  Case1_menu:
60      bne $v0, 1, Case2_menu    # Khac chuc nang 1 => 2
61      j Case1
62  Case2_menu:
63      bne $v0, 2, Case3_menu    # Khac chuc nang 2 => 3
64      j Case2
65  Case3_menu:
66      bne $v0, 3, Case4_menu    # Khac chuc nang 3 => 4
67      j Case3
68  Case4_menu:
69      bne $v0, 4, Case5_menu    # Khac chuc nang 4 => 5
70      j Case4
71  Case5_menu:
72      bne $v0, 5, main          # Khac chuc nang 5 => Chon lai chuc nang
73      j Exit                    # Chuc nang 5 => Exit

```

In ra Menu và các lựa chọn

## 2. Chức năng 1: Hiển thị hình ảnh DCE

```
75 # Chức năng 1: In ra giao diện chu
76 Case1:
77     addi $t0, $0, 0           # Biến đếm số dòng: $t0 = i = 0
78     addi $t1, $0, 16         # Tổng số dòng: $t1 = 16
79     la $a0, String1
80 Loop_Case1:
81     beq $t1, $t0, main        # $t0 = $t1 => main. Kết thúc chức năng 1
82     li $v0, 4
83     syscall
84     addi $a0, $a0, 63        # 1 dòng có 63 ký tự. Chuyển in dòng tiếp theo
85     addi $t0, $t0, 1         # Số dòng i += 1
86     j Loop_Case1
```

- Các chuỗi Message đều có 63 ký tự.
- Có 16 dòng tất cả của các chữ cần hiện
- Thực hiện vòng lặp in ra đủ 16 dòng Message
- Vòng lặp kết thúc khi số dòng in ra là 16 (biến đếm \$t0 ban đầu bằng 0)

## 3. Chức năng 2: Hiển thị hình ảnh DCE chỉ còn lại viền, không còn màu số ở giữa

```
88 # Chức năng 2: In ra giao diện chu không có màu (chỉ có viền)
89 Case2:
90     addi $s0, $0, 0           # Biến đếm số dòng: $s0 = i = 0
91     addi $s1, $0, 16         # Tổng số dòng: $s1 = 16
92     la $s2, String1          # Lấy ra địa chỉ của String1 vào $s2
93
94 Loop_Case2:
95     beq $s1, $s0, main        # $s0 = $s1 => main. Kết thúc chức năng 2
96     addi $t0, $zero, 0        # Biến đếm ký tự trong 1 dòng $t0 = j = 0
97     addi $t1, $zero, 63       # 1 dòng có 63 ký tự. Chuyển in dòng tiếp theo
98 KiemTraMau:
99     # $t0 = $t1 => End. Kết thúc in ra 1 hàng
100     lb $t2, 0($s2)            # Load byte từ địa chỉ 0 của thanh ghi $s2
101     slti $t3, $t2, 48         # $t3 = $t2 < 48 ? 1 : 0
102     beq $t3, $zero, LonHon_0  # Lớn hơn 0 => LonHon_0
103     j In_Ky_Tu
104 LonHon_0:
105     slti $t3, $t2, 59         # 48 <= $t3 = $t2 < 59 ? 1 : 0
106     beq $t3, $zero, In_Ky_Tu  # Lớn hơn 9 => In_Ky_Tu
107     addi $t2, $zero, 32       # Đổi màu bằng " "
108     j In_Ky_Tu
109 In_Ky_Tu:
110     li $v0, 11                # In từng ký tự
111     addi $a0, $t2, 0
112     syscall
113     addi $s2, $s2, 1          # Tăng địa chỉ của $s2. Chuyển tới ký tự tiếp theo
114     addi $t0, $t0, 1          # Đếm ký tự j += 1
115     j KiemTraMau
116 End:
117     addi $s0, $s0, 1          # Số dòng i += 1
118     j Loop_Case2
```

- Thực hiện kiểm tra xem đó có phải kí tự chữ ( $0 \leq \$t2 \leq 9$ ) thì thực hiện thay đổi thành khoảng trắng.
- Thực hiện tương tự chức năng 1 khi in ra DCE
- Hàm *kiemtramau*: Kiểm tra có phải kí tự màu (màu <0) không?
  - Gán giá trị địa chỉ của \$s2 vào thanh ghi \$t2
  - So sánh \$t2 và 48 (mã ASCII của 0)
    - + Nếu  $\$t2 > 48$  ( $t2 > 0$ )  $\rightarrow$  *LonHon\_0*
    - + Nếu  $\$t2 < 48$  ( $t2 < 0$ )  $\rightarrow$  *In\_Ky\_Tu*
- Hàm *LonHon\_0*: Kiểm tra có phải kí tự màu (màu > 9) không?
  - So sánh \$t2 và 59 (mã ASCII của 9)
    - + Nếu  $\$t2 > 59$  ( $t2 > 9$ )  $\rightarrow$  *In\_Ky\_Tu*
    - + Nếu  $\$t2 < 59$  ( $t2 < 9$ )  $\rightarrow$  Thực hiện đổi tất cả các kí tự màu thành khoảng trắng ( $0 \times 20 = 32$  ASCII) rồi in ra màn hình.
- Hàm *In\_Ky\_Tu*: In từng ký tự sau khi xóa màu số ở giữa

#### 4. Chức năng 3: Đảo vị trí các chữ

```

120 # Chuc nang 3: Dao vi tri cac chu
121 Case3:
122     addi $s0, $zero, 0      # Bien dem so dong $s0 = i = 0
123     addi $s1, $zero, 16
124     la   $s2, String1      # Lay ra dia chi cua String1 vao $s2
125 Loop_Case3:
126     beq  $s1, $s0, main    # $s0 = $s1 => main. Ket thuc chuc nang 3
127     # Tach chuoai lon tao ra 3 chuoai nhophan chia 3 chu
128     sb   $zero, 22($s2)    # Truoc chu C = "null"
129     sb   $zero, 43($s2)    # Truoc chu E = "null"
130     sb   $zero, 60($s2)    # Truoc \n = "null"
131     # In doi vi tri
132     li   $v0, 4
133     la   $a0, 44($s2)      # In chu E
134     syscall
135
136     li   $v0, 4
137     la   $a0, 23($s2)      # In chu C
138     syscall
139
140     li   $v0, 4
141     la   $a0, 0($s2)       # In chu D
142     syscall
143
144     li   $v0, 4
145     la   $a0, 61($s2)      # Inphan con lai \n
146     syscall

```

```

147      # Gop lai thanh chuoi lon ban dau
148      addi      $t1, $zero, 32      # " "
149      sb        $t1, 22($s2)        # Truoc chu C = " "
150      sb        $t1, 43($s2)        # Truoc chu E = " "
151      sb        $t1, 60($s2)        # Truoc \n = " "
152
153      addi      $s0, $s0, 1          # So dong i += 1
154      addi      $s2, $s2, 63          # Gan lai dia chi moi xuong dong tiep theo += 63
155      j         Loop_Case3

```

- Tách Message gốc thành 3 chuỗi mới và lưu thành 3 phần riêng của các chữ D, C, E lần lượt ở 22, 43, 60 bằng cách thay đổi vị trí đó bằng NULL
- In ra các chữ cái lần lượt theo kí tự của 3 chuỗi mới. In ra E, C, D theo thứ tự cần thực hiện in từ địa chỉ:
  - 44: Vị trí của chữ E
  - 23: Vị trí của chữ C
  - 0: Vị trí của chữ D
  - 61: Phần còn lại của 1 message gốc “\n”
- Sau khi in các chữ vào vị trí mới thì ghép các chuỗi nhỏ thành chuỗi ban đầu bằng cách thay các vị trí trước đó khi tách thành khoảng trắng (0x20 = 32)

## 5. Chức năng 4: Đổi màu chữ

```
157 # Chức năng 4: Đổi màu chữ
158 Case4:
159     j Input_D
160 Input_D:
161     li $v0, 4
162     la $a0, Chu_D
163     syscall
164
165     li $v0, 5 # Nhập vào màu của chu D muốn đổi
166     syscall
167
168     addi $t2, $v0, 0
169     slti $t3, $t2, -1 # $t3 = $t2 < -1 ? 1 : 0
170     bne $t3, $zero, Input_D # Nho hơn 0 => Input_D: Nhập lại
171     slti $t3, $t2, 10 # 0 <= $t3 = $t2 < 10 ? 1 : 0
172     beq $t3, $zero, Input_D # Lớn hơn 9 => Input_D: Nhập lại
173     addi $s3, $v0, 48 # Lưu lại giá trị màu của chu D = $s3
174 Input_C:
175     li $v0, 4
176     la $a0, Chu_C
177     syscall
178
179     li $v0, 5 # Nhập vào màu của chu C muốn đổi
180     syscall
181
182     addi $t2, $v0, 0
183     slti $t3, $t2, -1 # $t3 = $t2 < -1 ? 1 : 0
184     bne $t3, $zero, Input_C # Nho hơn 0 => Input_C: Nhập lại
185     slti $t3, $t2, 10 # 0 <= $t3 = $t2 < 10 ? 1 : 0
186     beq $t3, $zero, Input_C # Lớn hơn 9 => Input_C: Nhập lại
187     addi $s4, $v0, 48 # Lưu lại giá trị màu của chu C = $s4
188 Input_E:
189     li $v0, 4
190     la $a0, Chu_E
191     syscall
192
193     li $v0, 5 # Nhập vào màu của chu E muốn đổi
194     syscall
195
196     addi $t2, $v0, 0
197     slti $t3, $t2, -1 # $t3 = $t2 < -1 ? 1 : 0
198     bne $t3, $zero, Input_E # Nho hơn 0 => Input_E: Nhập lại
199     slti $t3, $t2, 10 # 0 <= $t3 = $t2 < 10 ? 1 : 0
200     beq $t3, $zero, Input_E # Lớn hơn 9 => Input_E: Nhập lại
201     addi $s5, $v0, 48 # Lưu lại giá trị màu của chu E = $s5
202 Set_Default:
203     addi $s0, $zero, 0 # Biến đếm số dòng $s0 = i = 0
204     addi $s1, $zero, 16
205     la $s2, String1 # Lay ra địa chỉ của String1 vào $s2
206     addi $a1, $zero, 48 # Giá trị của số 0
207     addi $a2, $zero, 58 # Giá trị của số 9
208 Loop_Case4:
209     beq $s1, $s0, Update_Mau # $s0 = $s1 => Update_Mau: Cập nhật lại màu mọi chỗ chữ
210     addi $t0, $zero, 0 # Biến đếm ký tự trong 1 dòng $t0 = j = 0
211     addi $t1, $zero, 63 # 1 dòng có 63 ký tự. Chuyển in dòng tiếp theo
```



```

212 Check:
213     beq      $t1, $t0, Complete
214     lb       $t2, 0($s2)                # Lay ra gia chi cua dia chi 0($s2) vao $t2
215 Check_D:
216     slti     $s6, $t0, 23              # $t0 < 23 ? 1 : 0
217     beq      $s6, $zero, Check_C      # $s6 = 0 => Check_C: Thay doi het mau cua chu D
218     beq      $t2, $t5, Update_D
219     j        ChuyenKyTu
220 Check_C:
221     slti     $s6, $t0, 44              # $t0 < 44 ? 1 : 0
222     beq      $s6, $zero, Check_E      # $s6 = 0 => Check_E: Thay doi het mau cua chu C
223     beq      $t2, $t6, Update_C
224     j        ChuyenKyTu
225 Check_E:
226     beq      $t2, $t7, Update_E
227     j        ChuyenKyTu
228 Update_D:
229     sb       $s3, 0($s2)
230     j        ChuyenKyTu
231 Update_C:
232     sb       $s4, 0($s2)
233     j        ChuyenKyTu
234 Update_E:
235     sb       $s5, 0($s2)
236     j        ChuyenKyTu
237 ChuyenKyTu:
238     addi     $s2, $s2, 1                # Chuyen toi ky tu tiep theo
239     addi     $t0, $t0, 1                # Dem ky tu j += 1
240     j        Check
241 Complete:
242     li       $v0, 4
243     addi     $a0, $s2, -63              # Chuyen ve ky tu dau cua dong
244     syscall
245     addi     $s0, $s0, 1                # So dong i += 1
246     j        Loop_Case4
247 Update_Mau:
248     move     $t5, $s3
249     move     $t6, $s4
250     move     $t7, $s5
251     j        main
252 Exit:

```

- Lần lượt nhập các màu cho các chữ D, C, E và kiểm tra xem màu đó có hợp lệ hay không
  - Nếu không hợp lệ ( $\$t2 < 0$  hoặc  $\$t2 > 9$ ) → yêu cầu nhập lại
  - Nếu hợp lệ ( $0 \leq \$t2 \leq 9$ ) → chuyển tới nhập màu cho chữ tiếp theo
- Thực hiện vòng lặp để đổi các ký tự màu trong Message
  - Kiểm tra ký tự đó thuộc chữ nào
  - Thực hiện chuyển tới từng hàm đến thay đổi màu tương ứng với từng chữ
  - Kiểm tra từng ký tự xem ký tự đó có phải là ký tự màu hay không
    - + Nếu là ký tự màu → lưu màu mới để thay thế màu cũ
    - + Nếu không là ký tự màu → chuyển tới ký tự tiếp theo để kiểm tra

- In ra màn hình giao diện mới đã thay đổi màu các chữ tương tự Chức năng 1
- Sau khi in ra màn hình, lưu lại các giá trị màu để thực hiện chức năng về sau

