

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



BÁO CÁO
FINAL PROJECT 08 - 09

Học phần: Thực hành kiến trúc máy tính

Mã học phần: IT3280

Giảng viên hướng dẫn: Th.S Lê Bá Vui

Nhóm sinh viên thực hiện:

1. Đỗ Gia Huy - 20215060
2. Nguyễn Hoàng Việt - 20215168

Mã lớp: 130938

Hà Nội, tháng 7 năm 2022

I. Phân công công việc

1. Đỗ Gia Huy

- MSSV: 20215060
- Email: huy.dg215060@sis.hust.edu.vn
- Công việc:
 - Làm project 08: Mô phỏng ổ đĩa RAID 5
 - Thiết kế bìa và trang trí file báo cáo

2. Nguyễn Hoàng Việt

- MSSV: 20215168
- Email: viet.nh215168@sis.hust.edu.vn
- Công việc:
 - Làm project 09: Vẽ hình bằng ký tự ASCII
 - Căn chỉnh và sửa chữa file báo cáo

II. Project 8: Mô phỏng ổ đĩa RAID 5

1. Phân tích chung

1.1. Đề bài

Hệ thống ổ đĩa RAID 5 cần tối thiểu 3 ổ đĩa cứng, trong đó phần dữ liệu parity sẽ được chứa lần lượt lên 3 ổ đĩa như trong hình bên. Hãy viết chương trình mô phỏng hoạt động của RAID 5 với 3 ổ đĩa, với giả định rằng, mỗi block dữ liệu có 4 ký tự. Giao diện như trong minh họa dưới. Giới hạn chuỗi ký tự nhập vào có độ dài là bội của 8.

Trong ví dụ sau, chuỗi ký tự nhập vào từ bàn phím (DCE.****ABCD1234HUSTHUST) sẽ được chia thành các block 4 byte. Block 4 byte đầu tiên "DCE." sẽ được lưu trên Disk 1, Block 4 byte tiếp theo "****" sẽ lưu trên Disk 2, dữ liệu trên Disk 3 sẽ là 4 byte parity được tính từ 2 block đầu tiên với mã ASCII là $6e = 'D' \text{ xor } '*'$; $69 = 'C' \text{ xor } '*'$; $6f = 'E' \text{ xor } '*'$; $04 = '.' \text{ xor } '*'$

| **Nhập chuỗi kí tự : DCE.****ABCD1234HUSTHUST | | |
|---|---------------------|---------------------|
| Disk 1 | Disk 2 | Disk 3 |
| ----- | ----- | ----- |
| DCE. | **** | [[6e, 69, 6f, 04]] |
| ABCD | [[70, 70, 70, 70]] | 1234 |
| [[00, 00, 00, 00]] | HUST | HUST |
| ----- | ----- | ----- |

1.2. Phân tích đề bài:

- Đầu vào là một dãy ký tự sao cho độ dài của dãy đó phải chia hết cho 8
- Đầu ra là in ra màn hình console kết quả chạy giả lập ổ đĩa RAID 5

1.3. Các thành phần và ý tưởng thuật toán của chương trình

⇒ Chương trình được chia ra 3 hàm chính

- Hàm nhập chuỗi ký tự và kiểm tra chuỗi đó có số ký tự là bội của 8 hay không hay là chuỗi rỗng.
- Có hàm RAID 5 được chia làm 3 phần:
 - + split1 : 4 byte parity được tính từ 2 block đầu tiên sẽ được lưu vào Disk 3. Nếu còn chuỗi cần lưu, hàm sẽ tiếp tục block2
 - + split2 : 4 byte parity được tính sẽ lưu vào Disk 2. Nếu còn chuỗi cần lưu, hàm sẽ tiếp tục block3
 - + split3 : 4 byte parity được tính sẽ lưu vào Disk 1. Nếu còn chuỗi cần lưu, hàm sẽ quay trở về block1

Kết thúc 3 phần nói trên mà còn chuỗi ký tự và chuỗi party thì lặp lại từ phần 1, cho đến khi hết ký tự để xét thì dừng.

- Hàm hex để chuyển 4 byte parity từ chuẩn ASCII sang Hexa.

1.4. Cách thực hiện

Duyệt toàn bộ chuỗi ký tự, tính mã chẵn lẻ của 2 ký tự cách nhau 4 ký tự. Lưu trữ ký tự đầu tiên vào 1 vùng nhớ trống, lưu trữ ký tự thứ 2 vào 1 vùng nhớ trống khác, lưu giá trị mã ascii của byte parity vào 1 vùng nhớ trống thứ 3. In ra màn hình theo cấu hình thiết lập để được giả lập hệ thống ổ đĩa. Chương trình có thể lặp lại nhiều lần.

1.5. Ý nghĩa của các thanh ghi trong chương trình

- \$s1: địa chỉ của Disk1
- \$s2: địa chỉ của Disk2
- \$s3: địa chỉ của Disk3
- \$t3: độ dài chuỗi input
- \$t0: index
- \$t1: địa chỉ của chuỗi nhập vào
- \$t2: string[i]
- \$t3: length
- \$t4: gán giá trị bằng 7 (cho lặp tới 0 để đủ 8 bit)
- \$t7: địa chỉ của hex
- \$a0: chỉ số của mảng hex
- \$t8: địa chỉ của chuỗi parity

2. Source code

```

1  .data
2      start: .ascii "Nhap chuoai ky tu : "
3      hex: .byte '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'
4      d1: .space 4
5      d2: .space 4
6      d3: .space 4
7      array: .space 32
8      string: .space 5000
9      enter: .ascii "\n"
10     error_length: .ascii "Do dai chuoai khong hop le! Nhap lai.\n"
11     m: .ascii "          Disk 1          Disk 2          Disk 3\n"
12     m2: .ascii "-----\n"
13     m3: .ascii "|          "
14     m4: .ascii "|          "
15     m5: .ascii "[[ "
16     m6: .ascii "]]          "
17     comma: .ascii ","
18     ms: .ascii "Try again?"
19
20 .text
21     la    $s1, d1          # Tuong ung disk 1
22     la    $s2, d2          # Tuong ung disk 2
23     la    $s3, d3          # Tuong ung disk 3
24     la    $a2, array       # dia chi mang chua parity

```

```

24     la    $a2, array                # địa chỉ mảng chứa parity
25
26 input: li    $v0, 4                  # nhập tên (chuỗi)
27         la    $a0, start
28         syscall
29         li    $v0, 8
30         la    $a0, string
31         li    $a1, 1000
32         syscall
33         move   $s0, $a0              # s0 chứa địa chỉ xâu mới nhập
34         li    $v0, 4
35         la    $a0, m
36         syscall
37         li    $v0, 4
38         la    $a0, m2
39         syscall
40
41 #-----kiểm tra độ dài có chia hết cho 8 không-----
42 length: addi $t3, $zero, 0           # t3 = length
43         addi $t0, $zero, 0           # t0 = index
44
45 check_char: add $t1, $s0, $t0         # t1 = address of string[i]
46             lb    $t2, 0($t1)         # t2 = string[i]
47
48         nop
49         beq    $t2, 10, test_length   # t2 = '\n' kết thúc xâu
50         nop
51         addi   $t3, $t3, 1             # length++
52         addi   $t0, $t0, 1             # index++
53         j      check_char
54         nop
55 test_length: move $t5, $t3
56             and    $t1, $t3, 0x0000000f # xóa hết các byte của $t3 về 0, chỉ giữ lại byte cuối
57             bne    $t1, 0, test1       # byte cuối bằng 0 hoặc 8 thì số chia hết cho 8
58             j      split1
59 test1: beq    $t1, 8, split1
60         j      error1
61 error1: li    $v0, 4
62         la    $a0, error_length
63         syscall
64         j      input
65
66 #-----kết thúc kiểm tra độ dài-----
67
68 #-----lay parity-----
69 HEX:  li    $t4, 7
70     loopH: blt    $t4, $0, endloopH
71             sll    $s6, $t4, 2          # s6 = t4*4
72
73     70     srlv   $a0, $t8, $s6          # a0 = t8>>s6
74     71     andi   $a0, $a0, 0x0000000f  # a0 = a0 & 0000 0000 0000 0000 0000 0000 1111 => lay byte cuối
75     72     la     $t7, hex
76     73     add    $t7, $t7, $a0
77     74     bgt    $t4, 1, nextc
78     75     lb     $a0, 0($t7)           # print hex[a0]
79     76     li     $v0, 11
80     77     syscall
81     78     # lb $t6, 0($t7)
82     79     # beq $t6, 48, in
83     80     nextc: addi $t4, $t4, -1
84     81             j      loopH
85     82     # in: bgt $t4, 1, loopH
86     83     # move $a0, $t6
87     84     # li $v0, 11
88     85     # syscall
89     86     endloopH: jr    $ra
90
91 #-----Mo phong RAID 5-----
92 # xet 6 khoi dau
93 # lan 1: luu vao 2 khoi 1,2; xor vao 3-----
94 split1: addi $t0, $zero, 0           # số byte được in ra (4 byte)

```

```

93     addi $t9, $zero, 0
94     addi $t8, $zero, 0
95     la   $s1, d1
96     la   $s2, d2
97     la   $a2, array
98 print11:li $v0, 4
99     la   $a0, m3
100    syscall
101 b11: lb   $t1, ($s0)
102    addi $t3, $t3, -1
103    sb   $t1, ($s1)
104 b21: add  $s5, $s0, 4
105    lb   $t2, ($s5)
106    addi $t3, $t3, -1
107    sb   $t2, ($s2)
108 b31: xor  $a3, $t1, $t2
109    sw   $a3, ($a2)
110    addi $a2, $a2, 4
111    addi $t0, $t0, 1
112    addi $s0, $s0, 1
113    addi $s1, $s1, 1
114    addi $s2, $s2, 1
115    bgt  $t0, 3, reset

```

t2 chưa địa chỉ tung byte của disk 2

```

116    j     b11
117 reset: la   $s1, d1
118    la   $s2, d2
119 print12:lb $a0, ($s1)
120    li   $v0, 11
121    syscall
122    addi $t9, $t9, 1
123    addi $s1, $s1, 1
124    bgt  $t9, 3, next11
125    j     print12
126 next11:li $v0, 4
127    la   $a0, m4
128    syscall
129    li   $v0, 4
130    la   $a0, m3
131    syscall
132 print13:lb $a0, ($s2)
133    li   $v0, 11
134    syscall
135    addi $t8, $t8, 1
136    addi $s2, $s2, 1
137    bgt  $t8, 3, next12
138    j     print13

```

```

139 next12:li $v0, 4
140    la   $a0, m4
141    syscall
142    li   $v0, 4
143    la   $a0, m5
144    syscall
145
146    la   $a2, array
147    addi $t9, $zero, 0
148 print14:lb $t8, ($a2)
149    jal  HEX
150    li   $v0, 4
151    la   $a0, comma
152    syscall
153    addi $t9, $t9, 1
154    addi $a2, $a2, 4
155    bgt  $t9, 2, end1
156    j     print14
157 end1: lb   $t8, ($a2)
158    jal  HEX
159    li   $v0, 4
160    la   $a0, m6
161    syscall

```

in ra 3 parity đầu có dấu ",", parity cuối cùng k có

```

162     li    $v0, 4
163     la    $a0, enter
164     syscall
165     beq    $t3, 0, exit1
166     #-----
167 split2: la    $a2, array
168         la    $s1, d1
169         la    $s3, d3
170         addi  $s0, $s0, 4
171         addi  $t0, $zero, 0
172 print21: li   $v0, 4
173         la    $a0, m3
174         syscall
175 b12:     lb    $t1, ($s0)
176         addi  $t3, $t3, -1
177         sb    $t1, ($s1)
178 b32:     add  $s5, $s0, 4
179         lb    $t2, ($s5)
180         addi  $t3, $t3, -1
181         sb    $t2, ($s3)
182 b22:     xor  $a3, $t1, $t2
183         sw    $a3, ($a2)
184         addi  $a2, $a2, 4

```

```

185         addi  $t0, $t0, 1
186         addi  $s0, $s0, 1
187         addi  $s1, $s1, 1
188         addi  $s3, $s3, 1
189         bgt  $t0, 3, reset2
190         j     b12
191 reset2: la    $s1, d1
192         la    $s3, d3
193         addi  $t9, $zero, 0
194 print22: lb    $a0, ($s1)
195         li    $v0, 11
196         syscall
197         addi  $t9, $t9, 1
198         addi  $s1, $s1, 1
199         bgt  $t9, 3, next21
200         j     print22
201 next21: li    $v0, 4
202         la    $a0, m4
203         syscall
204         la    $a2, array
205         addi  $t9, $zero, 0
206         li    $v0, 4
207         la    $a0, m5

```

```

208     syscall
209 print23: lb    $t8, ($a2)
210         jal    HEX
211         li    $v0, 4
212         la    $a0, comma
213         syscall
214         addi  $t9, $t9, 1
215         addi  $a2, $a2, 4
216         bgt  $t9, 2, next22
217         j     print23
218 next22: lb    $t8, ($a2)
219         jal    HEX
220         li    $v0, 4
221         la    $a0, m6
222         syscall
223         li    $v0, 4
224         la    $a0, m3
225         syscall
226         addi  $t8, $zero, 0
227 print24: lb    $a0, ($s3)
228         li    $v0, 11
229         syscall
230         addi  $t8, $t8, 1

```



```

231     addi    $s3, $s3, 1
232     bgt     $t8, 3, end2
233     j       print24
234
235 end2:  li     $v0, 4
236     la      $a0, m4
237     syscall
238     li      $v0, 4
239     la      $a0, enter
240     syscall
241     beq     $t3, 0, exit1
242     #-
243 split3: la    $a2, array
244         la    $s2, d2
245         la    $s3, d3
246         addi  $s0, $s0, 4
247         addi  $t0, $zero, 0
248 print31: li   $v0, 4
249         la    $a0, m5
250         syscall
251 b23:    lb     $t1, ($s0)
252         addi  $t3, $t3, -1
253         sb     $t1, ($s1)

```

```

254 b33:    add    $s5, $s0, 4
255         lb     $t2, ($s5)
256         addi  $t3, $t3, -1
257         sb     $t2, ($s3)
258 b13:    xor    $a3, $t1, $t2
259         sw     $a3, ($a2)
260         addi  $a2, $a2, 4
261         addi  $t0, $t0, 1
262         addi  $s0, $s0, 1
263         addi  $s1, $s1, 1
264         addi  $s3, $s3, 1
265         bgt   $t0, 3, reset3
266         j     b23
267 reset3: la    $s2, d2
268         la    $s3, d3
269         la    $a2, array
270         addi  $t9, $zero, 0
271 print32: lb    $t8, ($a2)
272         jal    HEX
273         li     $v0, 4
274         la     $a0, comma
275         syscall
276         addi  $t9, $t9, 1

```

```

277         addi  $a2, $a2, 4
278         bgt   $t9, 2, next31
279         j     print32
280 next31: lb     $t8, ($a2)
281         jal    HEX
282         li     $v0, 4
283         la     $a0, m6
284         syscall
285         li     $v0, 4
286         la     $a0, m3
287         syscall
288         addi  $t9, $zero, 0
289 print33: lb    $a0, ($s2)
290         li     $v0, 11
291         syscall
292         addi  $t9, $t9, 1
293         addi  $s2, $s2, 1
294         bgt   $t9, 3, next32
295         j     print33
296 next32: addi  $t9, $zero, 0
297         addi  $t8, $zero, 0
298         li     $v0, 4
299         la     $a0, m4

```



```

300     syscall
301     li     $v0, 4
302     la     $a0, m3
303     syscall
304 print34: lb     $a0, ($s3)
305     li     $v0, 11
306     syscall
307     addi    $t8, $t8, 1
308     addi    $s3, $s3, 1
309     bgt     $t8, 3, end3
310     j       print34
311
312 end3:  li     $v0, 4
313     la     $a0, m4
314     syscall
315     li     $v0, 4
316     la     $a0, enter
317     syscall
318     beq     $t3, 0, exit1
319     #-----end 6 khối đầu-----
320     # chuyển sang 6 khối tiếp theo
321 nextLoop: addi $s0, $s0, 4
322     j       split1

```

```

323
324 exit1: li     $v0, 4
325     la     $a0, m2
326     syscall
327     j       ask
328     #-----kết thúc mô phỏng RAID 5-----
329
330     #-----try again-----
331 ask:   li     $v0, 50
332     la     $a0, ms
333     syscall
334     beq     $a0, 0, clear
335     nop
336     j       exit
337     nop
338     # clear: đưa string về trạng thái ban đầu để thực hiện lại quá trình
339 clear: la     $s0, string
340     add     $s3, $s0, $t5      # s3: địa chỉ byte cuối cùng được sử dụng trong string
341     li     $t1, 0
342 goAgain: sb    $t1, ($s0)      # set byte ở địa chỉ s0 thành 0
343     nop
344     addi    $s0, $s0, 1
345     bge     $s0, $s3, input

```

```

346     nop
347     j       goAgain
348     nop
349     #-----end try again-----
350
351 exit:  li     $v0, 10
352     syscall

```

3. Thực thi

3.1. Cách chạy chương trình

- Nhập 1 chuỗi ký tự từ bàn phím sao cho độ dài chuỗi ký tự được nhập phải chia hết cho 8 và khác rỗng. Nếu không nhập đúng, chương trình sẽ báo lỗi và yêu cầu người dùng nhập lại cho đến khi nhập đúng thì thôi.
- Khi nhập đúng, chương trình sẽ chạy và in ra in ra giả lập ổ đĩa RAID 5.
- Khi chương trình chạy và in xong, lựa chọn thực hiện lại chương trình hoặc kết thúc chương trình.

3.2. Kết quả thực thi

- Trường hợp như ví dụ đề bài:


```

Nhap chuoì ky tu : lebavui
Disk 1          Disk 2          Disk 3
-----
Do dai chuoì khong hop le! Nhap lai.
Nhap chuoì ky tu : kientrucmaytinh
Disk 1          Disk 2          Disk 3
-----
Do dai chuoì khong hop le! Nhap lai.
Nhap chuoì ky tu : kythuatsoict
Disk 1          Disk 2          Disk 3
-----
Do dai chuoì khong hop le! Nhap lai.
Nhap chuoì ky tu : 123456789
Disk 1          Disk 2          Disk 3
-----
Do dai chuoì khong hop le! Nhap lai.
Nhap chuoì ky tu :

```

- Trường hợp nhập xâu có độ dài chia hết cho 8:

```

Nhap chuoì ky tu : dogiahuy
Disk 1          Disk 2          Disk 3
-----
| dogi | | ahuy | | [[ 05,07,12,10]] |
-----
Nhap chuoì ky tu :

```

```

Nhap chuoì ky tu : abcdefgh12345678
Disk 1          Disk 2          Disk 3
-----
| abcd | | efgh | | [[ 04,04,04,0c]] |
| 1234 | | [[ 04,04,04,0c]] | | 5678 |
-----

-- program is finished running --

```

```

Nhap chuoì ky tu : qwertyuiopasdfghjklzxcvbnm,./;'\=-0987654321~!@#%&*()_+{}|:"<>?HUSTSOICTKMT_thay_Le_Ba_Vui
Disk 1          Disk 2          Disk 3
-----
| qwer | | tyui | | [[ 05,0e,10,1b]] |
| opas | | [[ 0b,1e,06,1b]] | | dfgH |
| [[ 12,08,1a,18]] | | jklz | | xcvb |
| nm,. | | /:'[ | | [[ 41,5e,0b,75]] |
| ]\=- | | [[ 6d,65,05,1a]] | | 0987 |
| [[ 04,04,54,4d]] | | 6543 | | 21`~ |
| !@#% | | %^&* | | [[ 04,1e,05,0e]] |
| ()_+ | | [[ 53,54,23,11]] | | {}|: |
| [[ 6a,69,6d,6b]] | | "<>? | | HUST |
| SOIC | | TKMT | | [[ 07,04,1d,0e]] |
| T_th | | [[ 35,2e,2b,24]] | | ay_L |
| [[ 3a,09,37,08]] | | e_Ba | | _Vui |
-----

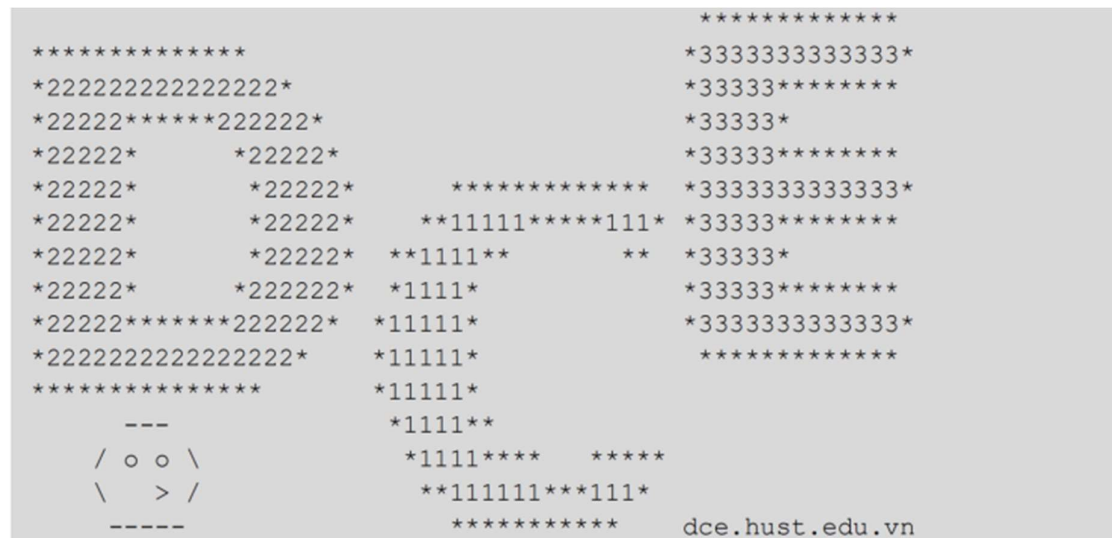
-- program is finished running --

```

III. Vẽ hình bằng kí tự ASCII

1. Đề bài

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



- 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 hoạ tiết đính kèm cũng được phép di chuyển theo.
- Hãy nhập từ bàn phím kí 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.

2. Thực hiện yêu cầu

Ta coi hình vẽ là một mảng hai chiều cỡ $64 * 16$ nhưng được trình bày dưới dạng mảng 1 chiều với kích cỡ $64 * 16 = 1024$ kí tự. Trong đó ở mỗi dòng:

- 21 kí tự dành cho chữ D.
- 21 kí tự dành cho chữ C.

- 21 kí tự dành cho chữ E.
- 1 kí tự cuối là \n.

```

1 .data
2
3 Str: .asciiz "***** \n*****"
4 # Cõi hình ve la mang 2 chieu kích co 64 * 14 nhưng đươc trình bay duoi dang mang 1 chieu.
5 # "***** \n"
6 # "***** *333333333333* \n"
7 # "*2222222222222222* *333333333333* \n"
8 # "*22222*****22222* *33333* \n"
9 # "*22222* *22222* *333333333333* \n"
10 # "*22222* *22222* ***** *3333333333333333* \n"
11 # "*22222* *22222* **11111*****111* *3333333333333333* \n"
12 # "*22222* *22222* **1111** ** *33333* \n"
13 # "*22222* *22222* *1111* *3333333333333333* \n"
14 # "*22222*****222222222222* *11111* *3333333333333333* \n"
15 # "*22222222222222222222* *11111* ***** \n"
16 # "***** *11111* \n"
17 # " --- *1111** \n"
18 # " / o o \ \ *1111**** ***** \n"
19 # " \ \ > / **111111***111* \n"
20 # " ----- ***** dce.hust.edu.vn \n"
21
22 Menu: .asciiz "\n\n ----MENU----\n 1. In hình ve.\n 2. In hình ve chỉ co vien.\n 3. Hoan doi vi tri.\n 4. Doi
23 Msg: .asciiz "Nhap lan luot mau cho chu D, C, E (Tu 0 den 9):\n "
24 Error: .asciiz "So khong phu hop. Nhap lai!\n"

```

Biến Str có giá trị là :

```

"***** \n*****
*333333333333* \n*2222222222222222* *33333*****
\n*22222*****222222* *33333* \n*22222*
*22222* *33333***** \n*22222* *22222*
***** *3333333333333333* \n*22222* *22222* **11111*****111*
*33333***** \n*22222* *22222* **1111** ** *33333*
\n*22222* *22222* *1111* *33333*****
\n*22222*****222222* *11111* *3333333333333333*
\n*22222222222222222222* *11111* ***** \n*****
*11111* \n --- *1111**
\n / o o \ \ *1111**** ***** \n \ \ > /
**111111***111* \n ----- *****
dce.hust.edu.vn \n

```

Lưu ý: Để có được kí tự ‘\’ trên màn hình thì ta phải viết là ‘\\’.

Tiếp theo, ta sẽ tạo menu để thực hiện các yêu cầu đề bài.

- Tạo biến menu:

Menu: .asciiz "\n\n ----MENU----\n 1. In hình ve.\n 2. In hình ve chỉ co vien.\n 3. Hoan doi vi tri.\n 4. Doi mau chu.\n 5. Thoat.\n Nhap lua chon: "

```

28  main:
29
30  la $a0, Menu
31  li $v0, 4
32  syscall
33
34  li $v0, 5
35  syscall
36
37  Case1:
38          addi $v1 $0 1
39          bne $v0 $v1 Case2
40          j c1
41  Case2:
42          addi $v1 $0 2
43          bne $v0 $v1 Case3
44          j c2
45  Case3:
46          addi $v1 $0 3
47          bne $v0 $v1 Case4
48          j c3
49  Case4:
50          addi $v1 $0 4
51          bne $v0 $v1 Case5
52          j c4
53  Case5:
54          addi $v1 $0 5
55          bne $v0 $v1 default
56          j Exit
57  default:
58          j main

```

Kết quả:

```

---MENU---
1. In hình vẽ.
2. In hình vẽ chi co vien.
3. Hoan doi vi tri.
4. Doi mau chu.
5. Thoat.
Nhap lua chon:

```

2.1 In hình vẽ

Vì biến Str đã chứa hoàn toàn hình vẽ nên ta chỉ cần in chuỗi Str ra.

```

60  #1. In hình vẽ.-----
61  c1:
62          li $v0, 4
63          la $a0, Str
64          syscall
65          j main

```

```
Nhap lua chon: 1

*****
*33333333333333*
*22222222222222*
*33333*
*2222*      *22222*
*33333*****
*2222*      *2222*
*33333333333333*
*2222*      *2222*      *****1111*
*33333*****
*2222*      *2222*      *1111*
*33333*****
*2222*      *22222*      *1111*
*33333333333333*
*22222222222222*      *1111*
*****
*1111*
*1111*
/ o o \          *1111***   *****
\ > /            *11111*1111*
_              *****
                dce.hust.edu.vn
```

```

67 #2. In hình vẽ chi có viên.-----
68 c2:
69         la $s0, Str      # $s0 là địa chỉ có số của str
70         li $s1, 0        # $s1 là con trỏ của str
71
72 loop_2:  beq $s1, 1024, main      # Kết thúc khi in đủ số kí tự.
73         lb $t0, 0($s0)          # $t0 chứa giá trị phần tử trong str.
74         bgt $t0, 57, print_2
75         bgt $t0, 47, digit_2    # Các chữ số 0-9 có giá trị ascii là 48-57.
76         j print_2
77
78 digit_2: li $t0, 32            # Thay chữ số bằng kí tự space.
79 print_2:
80         li $v0, 11             # In từng kí tự
81         move $a0, $t0
82         syscall
83
84         addi $s0, $s0, 1       # Tăng địa chỉ str.
85         addi $s1, $s1, 1       # Tăng giá trị con trỏ.
86         j loop_2
87

```

[illegible]

2.4. Đổi màu chữ

2.4.1. Nhập dữ liệu

- Lần lượt nhập các chữ số vào để thu được màu cho 3 chữ D,C,E.
- Nếu phát hiện không phải chữ số thì yêu cầu nhập lại.
- Cộng số vừa nhập vào với 48 để thu được giá trị ASCII của số tương ứng rồi lưu trữ vào các thanh ghi \$t7, \$t8, \$t9.

```
129 #4. Doi mau chu.-----
130 c4:
131 Input:
132 li $v0, 4
133 la $a0, Msg
134 syscall
135
136 li $v0, 5
137 syscall
138
139 bgt $v0, 9, error
140 blt $v0, 0, error
141 addi $t7, $v0, 48          # $t7 chua mau chu D.
142
143 li $v0, 5
144 syscall
145
146 bgt $v0, 9, error
147 blt $v0, 0, error
148 addi $t8, $v0, 48          # $t8 chua mau chu C.
149
150 li $v0, 5
151 syscall
152
153 bgt $v0, 9, error
154 blt $v0, 0, error
155 addi $t9, $v0, 48          # $t9 chua mau chu E.
156 j Input_end
157
158 error:
159 li $v0, 4
160 la $a0, Error
161 syscall
162 j Input
163
```

2.4.2. Xử lý dữ liệu

Giống như bài 3, ta sẽ duyệt trên từng dòng một và sẽ in lần lượt:

21 kí tự chữ D → 21 kí tự chữ C → 21 kí tự chữ E → kí tự \n

Tuy nhiên, trong quá trình in 21 kí tự trên, nếu phát hiện kí tự chữ số (Có giá trị ASCII là 48 - 57) thì ta sẽ thay thế chúng bằng giá trị số mới được lưu trong các thanh ghi \$t7 với chữ D, \$t8 với chữ C, \$t9 với chữ E rồi mới in ra.

```

164 Input_end:
165         la $s0, Str
166         li $s2, -1      # $s2 là tung do con tro.
167
168 loop_4_j:    addi $s2, $s2, 1
169             beq $s2, 16, main      # Ket thuc khi duyet du 16 dong.
170             sll $s1, $s2, 6        # $s1 = $s2 * 64
171             add $s1, $s1, $s0      # $s1 là địa chỉ đầu mỗi dòng của hình vẽ.
172
173 # In D
174         addi $s3, $s1, 0          # Chu D bat dau tu dau dong
175         move $t6, $t7
176         jal print_21_char_c4
177
178 # In C
179         addi $s3, $s1, 21         # Chu C bat dau tu ki tu 21 tren dong
180         move $t6, $t8
181         jal print_21_char_c4
182
183 # In E
184         addi $s3, $s1, 42         # Chu E bat dau tu ki tu 42 tren dong
185         move $t6, $t9
186         jal print_21_char_c4
187
188 # In \n
189         lb $t0, 63($s1)
190         li $v0, 11                # In ki tu
191         move $a0, $t0
192         syscall
193
194         j loop_4_j
195
196 print_21_char_c4:
197         li $t4, 0
198         loop_4_i:
199             lb $t0, 0($s3)        # $t0 lưu giá trị phần tử trong Str
200             bgt $t0, 57, print_4   # Các chữ số 0-9 có giá trị ascii là 48-57.
201             j print_4
202
203 digit_4:   move $t0, $t6          # Đổi kí tự ban đầu thành kí tự mới nhập.
204 print_4:    li $v0, 11            # In ki tu
205             move $a0, $t0
206             syscall
207
208             addi $s3, $s3, 1      # s3 += 1
209             addi $t4, $t4, 1      # t4 += 1
210             bne $t4, 21, loop_4_i
211             jr $ra

```

Kết quả:

```
Nhap lua chon: 4
Nhap lan luot mau cho chu D, C, E (Tu 0 den 9):
6
7
8
*****
*****
*88888888888888*
*66666666666666*
*88888*****
*66666*****66666*
*88888*
*66666*      *66666*
*88888*****
*66666*      *66666*      *****
*88888888888888*
*66666*      *66666*      **77777*****777*
*88888*****
*66666*      *66666*      **77777**      **
*88888*
*66666*      *66666*      *77777*
*88888*****
*66666*****66666*      *77777*
*88888888888888*
*6666666666666666*      *77777*
*****
*77777*
*77777*
/  o  o  \
\   >  /
-----
*****
dce.hust.edu.vn
```