



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

BÁO CÁO PROJECT CUỐI KỲ

Thực hành Kiến trúc máy tính

Bài 3 - Bài 9

GIẢNG VIÊN HƯỚNG DẪN

ThS. Lê Bá Vui

SINH VIÊN THỰC HIỆN

Đỗ Hoàng Minh Hiếu

Nguyễn Ngọc Vinh

PHẦN A: KIỂM TRA TỐC ĐỘ VÀ ĐỘ CHÍNH XÁC KHI GÕ VĂN BẢN

Đỗ Hoàng Minh Hiếu

1. Mô tả đề bài

Thực hiện chương trình đo tốc độ gõ bàn phím và hiển thị kết quả bằng 2 đèn led 7 đoạn. Nguyên tắc:

- Cho một đoạn văn bản mẫu, cố định sẵn trong mã nguồn. Ví dụ “bo mon ky thuat may tinh”
- Sử dụng bộ định thời Timer (trong bộ giả lập Digital Lab Sim) để tạo ra khoảng thời gian để đo. Đây là thời gian giữa 2 lần ngắt, chu kì ngắt.
- Người dùng nhập các kí tự từ bàn phím. Ví dụ nhập “bo mOn ky 5huat may tinh”. Chương trình cần phải đếm số kí tự đúng (trong ví dụ trên thì người dùng gõ sai chữ O và 5) mà người dùng đã gõ và hiển thị lên các đèn led.
- Chương trình đồng thời cần tính được tốc độ gõ: thời gian hoàn thành và số từ trên một đơn vị thời gian.

2. Ý tưởng thuật toán

- Vòng lặp xử lý trước khi ngắt
 - + Ta tạo một vòng lặp kiểm tra KEY_READY, nếu bằng một thì ta tạo ngắt (trap) bằng lệnh teqi.
 - + Mỗi lần lặp ta cho sleep 4ms, cứ mỗi lần lặp thì cộng biến index \$s5 thêm 1, đủ 250 lần sẽ bằng 1s thì tăng biến đếm thời gian thêm 1s và reset lại index.
 - + Vòng lặp sẽ tự động kết thúc và nhảy đến tính toán thời gian và show ra đèn LED khi ta nhập đủ số ký tự.
- Bên trong .ktext
 - + Tính địa chỉ dựa vào địa chỉ cơ sở và offset.
 - + Lấy ký tự từ địa chỉ tương ứng và so sánh với nhau. Nếu sai thì không cộng biến count và nhảy đến Show_Key (In ký tự ra màn hình Display).
 - + Xóa thanh ghi chứa mã lỗi và tăng địa chỉ thêm 4 đơn vị để nhảy đến lệnh tiếp theo và trả lại thanh ghi PC.
- In ra thanh led 7 đoạn bằng cách lấy số ký tự đúng chia cho 10, thanh ghi HI lưu trữ giá trị sẽ được in ra ở thanh led bên phải và thanh ghi LO lưu trữ giá trị hàng chục in ra ở thanh led bên trái.
 - + Khi kiểm tra để lấy mã của chữ số tương ứng, ta check từng cái một, bắt đầu từ num_0. Nếu sai thì kiểm tra tiếp số tiếp theo. Khi tìm được số tương ứng, ta return rồi in ra trên thanh LED.
- Hàm speed_calc chuyển về dạng số dấu phẩy động để tính toán. Ta lấy thời gian chia cho tổng số ký tự để ra được số từ trên một giây.
- In ra thông báo và hộp thoại hỏi. Nếu chọn Yes thì thực hiện lại chương trình, chọn No hoặc Cancel để terminate.

3. Mã nguồn

```

1  .eqv SEVENSEG_RIGHT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn phải.
2  .eqv SEVENSEG_LEFT 0xFFFF0011  # Địa chỉ của đèn led 7 đoạn trái
3  .eqv KEY_CODE 0xFFFF0004        # ASCII code from keyboard, 1 byte
4  .eqv KEY_READY 0xFFFF0000       # =1 if has a new keycode ?
5  .eqv DISPLAY_CODE 0xFFFF000C    # ASCII code to show, 1 byte
6  .eqv DISPLAY_READY 0xFFFF0008   # =1 if the display has already to do
7
8  .data
9  default_mess: .asciiz "if i can stop one heart from breaking"
10 message1: .asciiz "Thời gian hoàn thành: "
11 message2: .asciiz "(s) \nTốc độ gõ trung bình của bạn là: "
12 message3: .asciiz " ký tự/s \n"
13 ask_again: .asciiz "Bạn muốn thử lại lần nữa không?"
14 .text
15 main:
16     li $k0, KEY_CODE
17     li $k1, KEY_READY
18     li $s0, DISPLAY_CODE
19     li $s1, DISPLAY_READY
20
21     li $s2, 37 # length = 37
22     la $s3, default_mess
23     li $t4, 0 # count_correct = 0
24     li $t3, 0 # set i = 0
25     li $s5, 0 # time_index, $s5 = 1000 => 1s
26     li $s6, 0 # time (s)
27
28 Loop:
29     beq $t3, $s2, Print_SoKyTu
30 WaitForKey:
31     lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
32     teqi $t1, 1 # if $t0 = 1 then raise an Interrupt
33     nop
34     li $v0, 32
35     li $a0, 4 # sleep 4ms
36     syscall
37     blt $s5, 250, continue # Slepp 4ms 250 lần -> 1s
38     li $s5, -1 # Du 1s gan s5 = -1
39     addi $s6, $s6, 1 # Tang bien thoi gian len 1
40 continue:
41     addi $s5, $s5, 1
42     j loop
43 .ktext 0x80000180
44 ReadKey:
45     lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
46 WaitForDis:
47     lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
48     beq $t2, $zero, WaitForKey # if $t2 == 0 then Polling
49 Encrypt:
50     add $t6, $s3, $t3 # Địa chỉ ký tự $t6 = địa chỉ cơ sở $s3 + offset $t3
51     lb $s4, 0($t6)
52     addi $t3, $t3, 1 # Tang offset them 1
53     bne $s4, $t0, ShowKey # Neu ky tu nhap vao so voi goc khong dung thi khong tang bien count
54     addi $t4, $t4, 1

```

```

54 ShowKey:
55     sw $t0, 0($s0) # show key
56     nop
57 next_pc:
58     mtc0 $zero, $13 # Clear interrupt cause
59     mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
60     addi $at, $at, 4 # $at = $at + 4 (next instruction)
61     mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
62     eret # Return from exception
63
64 Print_SoKyTu:
65     li $t1, 0xA # Dua dau \n de man hinh display co xuong dong sau moi lan nhap
66     sw $t1, 0($s0)
67     move $s0, $t4
68     li $t2, 10
69     div $s0, $t2 # Thuc hien phep chia 10 thi thanh ghi hi chua so ben phai va thanh lo la so ben trai
70     mfhi $t1
71     jal num_0 # Kiem tra tu 0-9
72     nop
73     jal SHOW_7SEG_RIGHT # show so ra thanh ghi led phai
74     nop
75     mflo $t1
76     jal num_0
77     nop
78     jal SHOW_7SEG_LEFT # show so ra thanh ghi led trai
79     nop
80     j speed_calc

```

```

81 num_0: # Kiem tra co phai so 0 khong
82     bne $t1, $zero, num_1 # Nhay sang kiem tra so 1
83     li $a0, 0x3F
84     jr $ra
85 num_1: # Kiem tra so 1
86     li $s1, 1
87     bne $t1, $s1, num_2
88     li $a0, 0x6
89     jr $ra
90 num_2: # Kiem tra so 2
91     li $s1, 2
92     bne $t1, $s1, num_3
93     li $a0, 0x5B
94     jr $ra
95 num_3: # Kiem tra so 3
96     li $s1, 3
97     bne $t1, $s1, num_4
98     li $a0, 0x4F
99     jr $ra
100 num_4: # Kiem tra so 4
101     li $s1, 4
102     bne $t1, $s1, num_5
103     li $a0, 0x66
104     jr $ra
105 num_5: # Kiem tra so 5
106     li $s1, 5
107     bne $t1, $s1, num_6

```

```

105 num_5: # Kiem tra so 5
106     li $s1, 5
107     bne $t1, $s1, num_6
108     li $a0, 0x6D
109     jr $ra
110 num_6: # Kiem tra so 6
111     li $s1, 6
112     bne $t1, $s1, num_7
113     li $a0, 0x7D
114     jr $ra
115 num_7: # Kiem tra so 7
116     li $s1, 7
117     bne $t1, $s1, num_8
118     li $a0, 0x7
119     jr $ra
120 num_8: # Kiem tra so 8
121     li $s1, 8
122     bne $t1, $s1, num_9
123     li $a0, 0x7F
124     jr $ra
125 num_9: # Kiem tra so 9
126     li $s1, 9
127     li $a0, 0x6F
128     jr $ra
129
130 SHOW_7SEG_LEFT:
131     li $t0, SEVENSEG_LEFT # assign port's address
132     sb $a0, 0($t0) # assign new value
133     nop
134     jr $ra
135     nop
136
137 SHOW_7SEG_RIGHT:
138     li $t0, SEVENSEG_RIGHT # assign port's address
139     sb $a0, 0($t0) # assign new value
140     nop
141     jr $ra
142     nop
143 speed_calc: # Tinh toan thoi gian hoan thanh va trung binh
144     mtc1 $s6, $f1 # Dua thoi gian vao thanh ghi f1
145     cvt.s.w $f1, $f1 # Chuyen doi sang dau phay dong chinh xac don
146     mul $s5, $s5, 4 # Lay time_index nhan 4 vi 1 lan sleep la 4ms
147     mtc1 $s5, $f3 # Dua vao f3
148     cvt.s.w $f3, $f3
149     li $s5, 1000 # Gan 1000 de chuyen doi ms -> s
150     mtc1 $s5, $f5
151     cvt.s.w $f5, $f5
152     div.s $f3, $f3, $f5 # Chuyen sang (s)
153     add.s $f1, $f1, $f3 # Cong lai -> $f1 la thoi gian go phim
154     mtc1 $s2, $f3
155     cvt.s.w $f3, $f3

```

```
153     add.s $f1, $f1, $f3 # Cong lai -> $f1 la thoi gian go phim
154     mtc1 $s2, $f3
155     cvt.s.w $f3, $f3
156     div.s $f3, $f3, $f1 # $f3 la so ky tu tren 1s
157 exit:
158     li $v0, 4
159     la $a0, message1 # Syscall hien thi thong bao 1
160     syscall
161     li $v0, 2
162     mov.s $f12, $f1 # In ra thoi gian hoan thanh
163     syscall
164     li $v0, 4
165     la $a0, message2 # Syscall hien thi thong bao 2
166     syscall
167     li $v0, 2
168     mov.s $f12, $f3 # In ra thoi gian trung binh
169     syscall
170     li $v0, 4
171     la $a0, message3 # Syscall hien thi thong bao 3
172     syscall
173     la $a0, ask_again # Hoi lai xem co thuc hien tiep hay khong
174     li $v0, 50
175     syscall
176     beq $a0, $zero, main
177     li $v0, 10
178     syscall
179 endmain:
```

4. Thực thi chương trình

4.1. Cách thực hiện chương trình

- Mở Digital Lab Sim và Keyboard and Display MMIO Simulator trong Tools.
- Chọn Connect to MIPS và RESET cả hai cái trên.
- Nhập chuỗi “if i can stop one heart from breaking” vào phần Keyboard trong Keyboard and Display MMIO Simulator.
- Số ký tự đúng sẽ hiện trên Digital Lab Sim và Thời gian sẽ được hiển thị ở cửa sổ Run I/O.
- Khi chương trình chạy và in xong, lựa chọn thực hiện lại chương trình hoặc kết thúc chương trình.

4.2. Kết quả thực thi chương trình

- Nhập chính xác:

The screenshot displays the 'Keyboard and Display MMIO Simulator' interface. The main window contains a table of memory segments with columns for 'Bkpt', 'Address', 'Code', 'Basic', and 'Source'. A 'Digital Lab Sim' window is overlaid, featuring a 7-segment display showing '88' and a 4x4 keypad. A 'Tool Control' window is also open, showing 'Disconnect from MIPS', 'Reset', 'Help', and 'Close' buttons. A 'Mars Messages' window is at the bottom, showing a list of messages including 'Reset: reset completed.' and 'Thoi gian hoan thanh: 5.992(s)'. The interface is in Vietnamese.

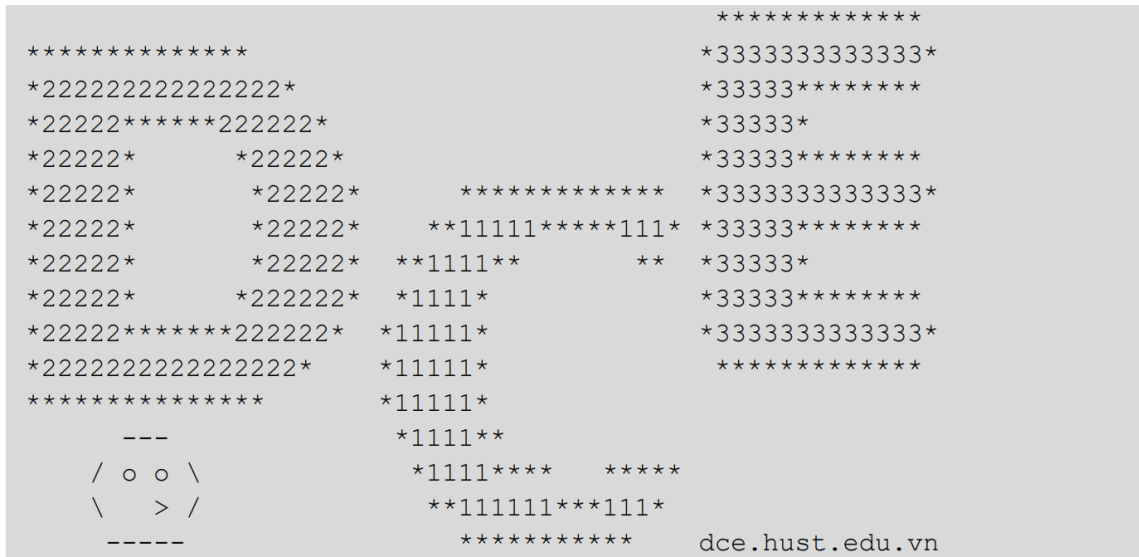
PHẦN B: VẼ HÌNH BẰNG KÝ TỰ ASCII

Nguyễn Ngọc Vinh

Đề 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.



- 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 họa 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.

Mã nguồn


```

1 .data
2 String1: .asciiz "*****\n"
3 String2: .asciiz "*****\n"
4 String3: .asciiz "*****\n"
5 String4: .asciiz "*****\n"
6 String5: .asciiz "*****\n"
7 String6: .asciiz "*****\n"
8 String7: .asciiz "*****\n"
9 String8: .asciiz "*****\n"
10 String9: .asciiz "*****\n"
11 String10: .asciiz "*****\n"
12 String11: .asciiz "*****\n"
13 String12: .asciiz "*****\n"
14 String13: .asciiz "*****\n"
15 String14: .asciiz " / o o \\\n"
16 String15: .asciiz " \\\n"
17 String16: .asciiz "-----\n"
18 EndString: .asciiz "-----SELECTION TABLE-----\n"
19 Message0: .asciiz "1. Hien thi hinh anh\n"
20 Request1: .asciiz "2. In ra hinh anh chi co vien\n"
21 Request2: .asciiz "3. Hoan doi vi tri cac chu\n"
22 Request3: .asciiz "4. Doi mau chu\n"
23 Request4: .asciiz "5. Thoat\n"
24 Terminate: .asciiz "Choose your option: "
25 Choose: .asciiz "Nhap mau cho chu D (Tru ky tu '*'):"
26 DColor: .asciiz "Nhap mau cho chu C(Tru ky tu '*'):"
27 CColor: .asciiz "Nhap mau cho chu E(Tru ky tu '*'):"

```

```

28 EColor: .asciiz "Nhap mau cho chu E(Tru ky tu '*'):"
29 newline: .asciiz "\n"
30 .text
31 main:
32     la $t0, String1 # Tai dia chi String1 vao $t0
33     la $a1, EndString # Tai dia chi EndString vao $a1
34
35     la $a0, Message0 # Tai dia chi Message0 vao $a0
36     li $v0, 4 # Dat ma syscall cho print_string vao $v0
37     syscall
38     la $a0, Request1 # Tai dia chi Request1 vao $a0
39     syscall
40     la $a0, Request2 # Tai dia chi Request2 vao $a0
41     syscall
42
43     la $a0, Request3 # Tai dia chi Request3 vao $a0
44     syscall
45     la $a0, Request4 # Tai dia chi Request4 vao $a0
46     syscall
47     la $a0, Terminate # Tai dia chi Terminate vao $a0
48     syscall
49     la $a0, Choose # Tai dia chi Choose vao $a0
50     syscall
51     li $v0, 5 # Dat ma syscall cho read_int vao $v0
52     syscall # Goi he thong de doc Lua chon nguoi dung
53

```

```

54 Case1menu:
55     li $v1, 1
56     bne $v0, $v1, Case2menu    # Neu $v0 == 1, nhay den Print_Image
57     j Print_Image
58 Case2menu:
59     li $v1, 2
60     bne $v0 $v1 Case3menu      # Neu $v0 == 2, nhay den Remove_Background
61     j Remove_Background
62 Case3menu:
63     li $v1, 3
64     bne $v0 $v1 Case4menu      # Neu $v0 == 3, nhay den Change_Position
65     j Change_Position
66 Case4menu:
67     li $v1, 4
68     bne $v0 $v1 Case5menu      # Neu $v0 == 4, nhay den Fill_Color
69     j Fill_Color
70 Case5menu:
71     li $v1, 5
72     bne $v0 $v1 default        # Neu $v0 == 5, nhay den Termination
73     j Termination
74 default:
75     j main

```

```

77 Print_Image:
78     beq $t0, $a1, end1    # Neu $t0 == $a1, ket thuc
79     lb $a0, 0($t0)    # Tai byte tai dia chi $t0 vao $a0
80     li $v0, 11        # Dat ma syscall cho print_char vao $v0
81     syscall            # Goi he thong de in ky tu
82     addi $t0, $t0, 1    # Tang $t0 Len 1
83     j Print_Image      # Quay Lai Print_Image
84 end1:
85     j main
86 Remove_Background:
87     # Loai bo phan ben trong cac chu cai, chi de lai vien
88     li $a0, ' '
89     lb $t2, 0($t0)
90     beq $t0, $a1, end2    # Neu $t0 == $a1, ket thuc
91     bge $t2, 48, check_upper_bound    # Neu $t2 >= '0',kiem tra ky tu so
92     j print                # Neu khong, in ky tu hien tai
93 check_upper_bound:
94     ble $t2, 57, skip_char    # Neu $t2 <= '9', bo qua ky tu nay
95 print:
96     move $a0, $t2            # Di chuyen ky tu hien tai vao $a0
97 skip_char:
98     li $v0, 11
99     syscall                # Goi he thong de in ky tu
100    addi $t0, $t0, 1
101    j Remove_Background
102 end2:

```

```

103 j main
104 Change_Position:
105     # Hoan doi vi tri cac chu
106     bge $t0, $a1, end3      # Neu $t0 >= $a1, ket thuc
107     li $v0, 11              # Dat ma syscall cho print_char vao $v0
108     addi $t3, $t0, 43        # Tinh vi tri cho chu C
109     li $t2, 17              # So Luong ky tu cho chu C
110     jal print_loop           # Goi print_loop de in chu C
111     addi $t3, $t0, 22        # Tinh vi tri cho chu E
112     li $t2, 21              # So Luong ky tu cho chu E
113     jal print_loop           # Goi print_loop de in chu E
114     addi $t3, $t0, 0         # Tinh vi tri cho chu D
115     li $t2, 21              # So Luong ky tu cho chu D
116     jal print_loop           # Goi print_loop de in chu D
117     addi $t0, $t0, 0x43      # Tinh vi tri cho hang ke tiep
118     li $a0, '\n'            # Ky tu xuong dong
119     syscall                  # Goi he thong de in xuong dong
120     j Change_Position        # Quay lai Change_Position
121 print_loop:
122     # Vong Lap de in ky tu
123     beqz $t2, end_print_loop # Neu $t2 == 0, ket thuc
124     lb $a0, 0($t3)           # Tai ky tu tai dia chi $t3 vao $a0
125     syscall                  # Goi he thong de in ky tu
126     addi $t3, $t3, 1         # Tang $t3 len 1
127     addi $t2, $t2, -1        # Giam $t2 xuong 1
128     j print_loop             # Quay lai print_loop
129 end_print_loop:
130     jr $ra                  # Quay lai dia chi tra ve
131 end3:|
132 j main                    # Quay lai main
133
134 Fill_Color:
135     # Doi mau chu D, C, E
136 EnterDColor:
137     li $v0, 4                # Dat ma syscall cho print_string vao $v0
138     la $a0, DColor           # Tai dia chi DColor vao $a0
139     syscall                  # Goi he thong de in DColor
140     li $v0, 12               # Dat ma syscall cho read_char vao $v0
141     syscall                  # Goi he thong de doc ky tu
142     beq $v0, 42, EnterDColor # Neu ky tu la '*', nhap lai
143     move $t3, $v0            # Luu mau D vao $t3
144     li $v0, 4                # Dat ma syscall cho print_string vao $v0
145     la $a0, newline          # Tai dia chi newline vao $a0
146     syscall                  # Goi he thong de in newline
147 EnterCColor:
148     li $v0, 4                # Dat ma syscall cho print_string vao $v0
149     la $a0, CColor           # Tai dia chi CColor vao $a0
150     syscall                  # Goi he thong de in CColor
151     li $v0, 12               # Dat ma syscall cho read_char vao $v0
152     syscall                  # Goi he thong de doc ky tu
153     beq $v0, 42, EnterCColor # Neu ky tu la '*', nhap lai
154     move $t4, $v0            # Luu mau C vao $t4
155     li $v0, 4                # Dat ma syscall cho print_string vao $v0
156     la $a0, newline          # Tai dia chi newline vao $a0

```

```

157     syscall                # Goi he thong de in newline
158 EnterEColor:
159     li $v0, 4              # Dat ma syscall cho print_string vao $v0
160     la $a0, EColor         # Tai dia chi EColor vao $a0
161     syscall                # Goi he thong de in EColor
162     li $v0, 12             # Dat ma syscall cho read_char vao $v0
163     syscall                # Goi he thong de doc ky tu
164     beq $v0, 42, EnterEColor # Neu ky tu la '*', nhap lai
165     move $t5, $v0          # Luu mau E vao $t5
166     li $v0, 4              # Dat ma syscall cho print_string vao $v0
167     la $a0, newline        # Tai dia chi newline vao $a0
168     syscall                # Goi he thong de in newline
169 Redraw:
170     # Ve lai hinh anh voi mau moi
171     beq $t0, $a1, end4      # Neu $t0 == $a1, ket thuc
172     lb $a0, 0($t0)          # Tai ky tu tai dia chi $t0 vao $a0
173 CheckDColor:
174     bne $a0, '2', CheckCColor # Neu khong phai '2',kiem tra ky tu '1'
175     move $a0, $t3           # Neu la '2', dat $a0 = mau cua D
176     j print_color          # In ky tu voi mau moi
177 CheckCColor:
178     bne $a0, '1', CheckEColor # Neu khong phai '1',kiem tra ky tu '3'
179     move $a0, $t4           # Neu la '1', dat $a0 = mau cua C
180     j print_color          # In ky tu voi mau moi

181 CheckEColor:
182     bne $a0, '3', print_color # Neu khong phai '3', in ky tu
183     move $a0, $t5           # Neu la '3', dat $a0 = mau cua E
184 print_color:
185     li $v0, 11             # Dat ma syscall cho print_char vao $v0
186     syscall                # Goi he thong de in ky tu
187     addi $t0, $t0, 1        # Tang $t0 len 1
188     j Redraw               # Quay lai Redraw
189 end4:
190     j main                 # Quay lai main
191
192 Termination:
193     li $v0, 10             # Dat ma syscall cho exit vao $v0
194     syscall                # Goi he thong de ket thuc chuong trinh

```

Bước 1: Khởi tạo và Quản lý Tùy chọn

1. Khai báo dữ liệu

Trong phần `.data`, các chuỗi ASCII đại diện cho từng dòng của hình ảnh được khai báo. Mỗi chuỗi kết thúc bằng ký tự xuống dòng `\n` để đảm bảo rằng khi in ra, mỗi chuỗi sẽ hiển thị trên một dòng riêng biệt.

```

String1: .asciiz "*****\n"
String2: .asciiz "*****\n"
String3: .asciiz "*2222222222222*\n"
String4: .asciiz "*22222*****22222*\n"
String5: .asciiz "*22222* *22222*\n"
String6: .asciiz "*22222* *22222* *****\n"
String7: .asciiz "*22222* *22222* **11111*****111*\n"
String8: .asciiz "*22222* *22222* **1111** **\n"
String9: .asciiz "*22222* *22222* *1111*\n"
String10: .asciiz "*22222*****22222* *1111*\n"
String11: .asciiz "*2222222222222222* *1111*\n"
String12: .asciiz "*****\n"
String13: .asciiz " --- *1111**\n"
String14: .asciiz " / o o \\\n"
String15: .asciiz " \\ > /\n"
String16: .asciiz " ---- ***** dce.hust.edu.vn\n"
EndString: .asciiz

```

2. Hiện thị menu và nhận lựa chọn của người dùng

Chương trình sử dụng các lệnh syscall để in ra menu và nhận đầu vào từ người dùng. Mỗi tùy chọn trong menu sẽ dẫn đến một hành động khác nhau. Các mục trong menu được khai báo như Message0, Request1, Request2, ...

```

Message0: .asciiz "-----SELECTION TABLE-----\n"
Request1: .asciiz "1. Hiện thị hình ảnh\n"
Request2: .asciiz "2. In ra hình ảnh chỉ có viên\n"
Request3: .asciiz "3. Hoán đổi vị trí các chu\n"
Request4: .asciiz "4. Đổi màu chu\n"
Terminate: .asciiz "5. Thoát\n"
Choose: .asciiz "Choose your option: "

```

Khi chương trình bắt đầu, nó sẽ hiển thị menu các tùy chọn để người dùng chọn lựa hành động cần thực hiện.

3. Xử lý lựa chọn của người dùng

Sau khi nhận lựa chọn từ người dùng, chương trình kiểm tra lựa chọn này và nhảy đến nhãn tương ứng để thực hiện hành động được chọn. Các nhãn như Case1menu, Case2menu, ..., sẽ xử lý từng lựa chọn cụ thể.

Ví dụ:

```

Case1menu:
    li $v1, 1
    bne $v0, $v1, Case2menu
    j Print_Image

```

Sau khi chạy chương trình, người dùng nhập số từ 1 đến 5 tương ứng với yêu cầu

```
-----SELECTION TABLE-----
1. Hien thi hinh anh
2. In ra hinh anh chi co vien
3. Hoan doi vi tri cac chu
4. Doi mau chu
5. Thoat
Choose your option: |
```

Bước 2: "Xử lý yêu cầu"

1. In ra toàn bộ hình ảnh

Mục tiêu :

- In ra toàn bộ hình ảnh ASCII đã được định nghĩa trong các chuỗi liên tiếp từ `String1` đến `String16`.
- Vòng lặp `Print_Image` liên tục đọc và in từng ký tự từ các chuỗi ASCII, bắt đầu từ `String1` và kết thúc khi gặp `EndString`.

Giải thích chi tiết:

- So sánh địa chỉ hiện tại với địa chỉ kết thúc để kiểm tra xem đã hết chuỗi dữ liệu chưa.
- Lấy ký tự tại địa chỉ hiện tại và thiết lập `syscall` để in ký tự đó ra màn hình.
- Sau khi in ký tự, tăng địa chỉ để trở đến ký tự tiếp theo.
- Quá trình này lặp lại cho đến khi gặp `EndString`.
-

Khi người dùng chọn tùy chọn 1 (như hình dưới):

```
-----SELECTION TABLE-----
1. Hien thi hinh anh
2. In ra hinh anh chi co vien
3. Hoan doi vi tri cac chu
4. Doi mau chu
5. Thoat
Choose your option: 1
```

⇒ chương trình sẽ in ra toàn bộ hình ảnh ASCII lưu trong các chuỗi từ `String1` đến `String16`. Các chuỗi này được nối tiếp nhau trong bộ nhớ, và vòng lặp `Print_Image` sẽ duyệt qua toàn bộ các ký tự này, đảm bảo rằng toàn bộ hình ảnh được in ra đúng như định dạng ban đầu.

⇒ **Kết quả**

Choose your option: 1

```

*****
*2222222222222222*
*22222*****22222*
*22222*      *22222*
*22222*      *22222*      *****
*22222*      *22222*      **11111*****111*
*22222*      *22222*      **1111**      **
*22222*      *22222*      *1111*
*22222*****22222*      *1111*
*2222222222222222*      *1111*
*****
---
/ o o \
\   > /
-----
*****
*3333333333333333*
*33333*****
*33333*
*33333*****
*3333333333333333*
*33333*****
*33333*
*33333*****
*3333333333333333*
*****
*1111**
*1111****      *****
**111111**111*
*****
dce.hust.edu.vn

```

2. In ra hình ảnh chỉ có viền

Choose your option: 2

Xác định viền của chữ cái

- Đọc từng ký tự trong chuỗi ASCII.
- Nếu ký tự không phải là số (đại diện cho viền), in ký tự đó ra.
- Nếu ký tự là số (đại diện cho các ký tự bên trong), thay thế bằng khoảng trắng trước khi in ra.

In ra hình ảnh chỉ có viền

- Sử dụng vòng lặp để duyệt qua từng ký tự trong chuỗi ASCII.
- Nếu ký tự là số, thay thế bằng khoảng trắng.
- In ra ký tự đã được xử lý ra màn hình.
-

⇒ Kết quả cuối cùng sẽ là viền của hình ảnh, mà không có bất kỳ ký tự nào nằm bên trong chữ cái D, C, E.

Choose your option: 2

[illegible]

3. Hoán đổi vị trí các chữ cái

Người dùng nhập 3

Choose your option: 3

Xác định vị trí của các chữ cái D, C, E

- Duyệt qua từng ký tự trong chuỗi ASCII.
- Xác định vị trí của các ký tự D, C, E và lưu lại.

Hoán đổi vị trí của các chữ cái

- Xác định vị trí mới cho các chữ cái D, C, E theo thứ tự mới (ECD).
- Thay thế các ký tự tại các vị trí ban đầu bằng khoảng trắng.
- Đặt các ký tự D, C, E vào vị trí mới.

In ra hình ảnh sau khi hoán đổi vị trí

- Sử dụng vòng lặp để duyệt qua từng ký tự trong chuỗi ASCII.
- In ra ký tự đã được xử lý ra màn hình.


```

Choose your option: 3
*****
*3333333333333*
*33333*****
*33333*
*33333*****
*3333333333333*
*33333*****
*33333*
*33333*****
*3333333333333*
*33333*****
*33333*
*33333*****
*3333333333333*
*****
*11111*
*11111*
*11111*
*11111*
*11111*
*11111*
*11111*
*11111*
*11111*****
*1111111111111*
*****

*****
*2222222222222*
*22222*****22222*
*22222*
*22222*
*22222*
*22222*
*22222*
*22222*
*22222*
*22222*****22222*
*222222222222222*
*****
---
/ o o \
\ > /
-----

dce.hust.edu.vn
*****

```

4. Hiện thị ảnh trên với màu mới

Nhập màu mới cho các chữ cái D, C, E:

- Sử dụng syscall số 4 để hiển thị các thông báo yêu cầu nhập màu cho các chữ cái D, C, E.
- Sử dụng syscall số 12 để nhận màu mới từ người dùng và lưu trữ chúng vào các thanh ghi tạm thời.

Thực hiện đổi màu:

- Duyệt qua chuỗi ASCII và xác định vị trí của các chữ cái D, C, E.
- Sử dụng các lệnh để thay đổi màu của các chữ cái này thành màu mới đã được nhập từ người dùng.

Hiển thị hình ảnh với màu mới:

- Sử dụng các lệnh và vòng lặp để in ra hình ảnh sau khi đã thay đổi màu.
- Ví dụ nhập màu mới là D là 4, C là 5, E là 6

```

Choose your option: 4
Nhap mau cho chu D (Tru ky tu '*'): 4
Nhap mau cho chu C(Tru ky tu '*'): 5
Nhap mau cho chu E(Tru ky tu '*'): 6

*****
*****
*4444444444444444*
*44444*****444444*
*44444*      *44444*
*44444*      *44444*      *****
*44444*      *44444*      **55555*****555*
*44444*      *44444*      **5555**      **
*44444*      *444444*      *5555*
*44444*****444444*      *55555*
*4444444444444444*      *55555*
*****
      *55555*
      *5555**
      *5555****      *****
      **555555***555*
      *****
*****
/  o  o  \
\    >  /
-----
*****
dce.hust.edu.vn

```

5. Thoát

- Nếu người dùng chọn tùy chọn "Thoát", thoát khỏi chương trình.
- Nếu không, tiếp tục thực hiện các lựa chọn khác.

```

Choose your option: 5

```

```

-- program is finished running --

```