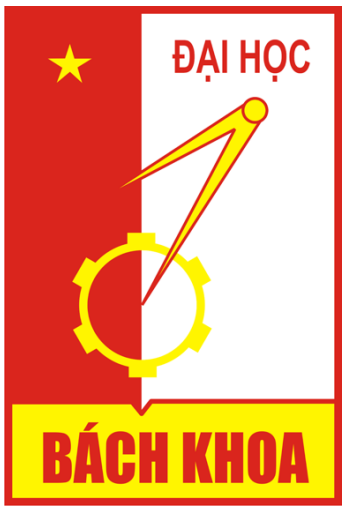
HANOI UNIVERSITY OF SCIENCE & TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



BÁO CÁO PROJECT CUỐI KỲ

IT3280 – THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Thông tin học phần

Mã HP	Tên học phần	Mã lớp
IT3280	Thực hành kiến trúc máy tính	147799

Thông tin sinh viên

Họ và tên	Lớp	Mã sinh viên
Bùi Thu Trang	IT-E6 03 K67	20225938
Lương Văn Khanh	IT-E6 01 K67	20225728

Giảng viên: Lê Bá Vui

Hanoi, tháng 06 2024

Table of Contents

I. BÀI 8: Mô phỏng ổ đĩa RAID5

Người thực hiện: Bùi Thu Trang

1. Mô tả yêu cầu đề bài

a) 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 kí tự. Giao diện như trong minh họa dưới. Giới hạn chuỗi kí tự nhập vào có độ dài là bội của 8.
- Trong ví dụ sau, chuỗi kí 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' xor '*' ; 69='C' xor '*'; 6f='E' xor '*' ; 04='.' xor '*'

```
'*'Nhap chuoi ki tu : 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     |

----------------      ----------------      ----------------
```

b) Phân tích đề bài

- Đầu vào gồm chuỗi kí tự khác rỗng và là bội của 8
- Kết quả trả về là mô phỏng hoạt động của RAID5 với 3 đĩa có giao diện như trên.

2. Thực hiện project

2.1 Thuật toán

- ***Chương trình xử lí các phần sau:***

- *Kiểm tra chuỗi được nhập có thỏa mãi điều kiện là bội của 8 và khác rỗng không:* Chương trình sẽ tính toán độ dài của xâu đã nhập. Nếu độ dài bằng không thì in ra comment yêu cầu nhập lại. Nếu độ dài khác 0 thì kiểm tra bit thấp nhất của độ dài có bằng 0 hoặc 8 hay không, nếu không thì in ra comment yêu cầu nhập lại (vì số chia hết cho 8 thì sẽ có bit cuối cùng bằng 0 hoặc 8).
- *Chuyển kết quả Hexa đã thu được qua phép xor sang cac byte parity tương ứng:* định nghĩ chuỗi gồm các số hexa: { '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f' }, từ kết quả ở dạng hexa của phép xor lấy từng byte khác 0 cộng với địa chỉ của mảng trên để xác định parity tương ứng của nó và in ra.
- *Mô phỏng hoạt động ổ đĩa RAID5:*

  Bước 1: Lưu 2 block đầu vào disk1 và disk 2, lưu kết quả xor của 2 block trên vào disk3 qua hàm chuyển từ kết quả Hexa sang các byte parity tương ứng, nếu xâu vẫn chưa kết thúc chuyển sang bước 2.

  Bước 2: Lưu 2 block kế tiếp vào disk1 và disk 3, lưu kết quả xor của 2 block trên vào disk2 qua hàm chuyển từ kết quả Hexa sang các byte parity tương ứng, nếu xâu vẫn chưa kết thúc chuyển sang bước 3.

Bước 3: Lưu 2 block kế tiếp vào disk2 và disk 3, lưu kết quả xor của 2 block trên vào disk1 qua hàm chuyển từ kết quả Hexa sang các byte parity tương ứng, nếu xâu vẫn chưa kết thúc chuyển sang bước 4

Bước 4: Lặp lại bước 1 với 2 block kế tiếp.

- ***Cách chương trình hoạt động***

- Chương trình đọc một xâu từ người dùng.
- Nó kiểm tra tính hợp lệ của xâu.
- Nếu xâu hợp lệ, nó tiến hành mô phỏng ổ đĩa RAID5 như đề bài.
- Kết quả mô phỏng sau đó được in ra.
- Chương trình hỏi người dùng nếu họ muốn tiếp tục, và vòng lặp lặp lại nếu người dùng chọn tiếp tục.

2.2 Giải thích code

- ***Phần chuẩn bị dữ liệu***

```
.data
        start: .asciiz "Xin moi nhap xau : "
        hextoascii: .byte '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'
        disk1: .space 4
        disk2: .space 4
        disk3: .space 4
        parity: .space 32
        string: .space 2000
        cmt: .asciiz "Do dai xau phai la boi cua 8. Xin moi nhap lai theo dinh dang.\n"
        disk: .asciiz "      Disk 1                Disk 2              Disk 3\n"
        mes1: .asciiz "----------------       ----------------       ----------------\n"
        mes2: .asciiz "|      "
        mes3: .asciiz "     |       "
        mes4: .asciiz "[[ "
        mes5: .asciiz "]]       "
        comma: .asciiz ","
        mutiplechoice: .asciiz "Ban co muon bat dau lai chuong trinh khong? \nAn 1: neu co \nAn 0: neu khong"
```

- **Disk1, disk2, disk3**: 3 mảng mô phỏng các disk trong ổ đĩa RAID5.
- **string:** mảng để lưu trữ chuỗi đầu vào.
- **mess1 đến mess5:** Các chuỗi ký tự phục vụ in kết quả.
- **Start, cmt, mutiplechoice:** in ra các thông báo cho người dùng.

- **Khởi tạo biến**
  - Khởi tạo các thanh ghi để trỏ đến các chuỗi, mảng

```
init:
        la      $s1, disk1                      # disk 1
        la      $s2, disk2                      # disk 2
        la      $s3, disk3                      # disk 3
        la      $a2, parity             # parity
```

- **Chương trình chính**
  - **process:** Quá trình chính để đọc xâu và in ra các ổ như đề yêu cầu.

```
input:  li      $v0, 4
        la      $a0, start
        syscall
        li      $v0, 8                  # nhap xau
        la      $a0, string
        li      $a1, 5000
        syscall
        move    $s0, $a0                # s0 chua dia chi cua string
        li      $v0, 4
        la      $a0, disk               # in "      Disk 1                  Disk 2                  Disk 3"
        syscall
        li      $v0, 4
        la      $a0, mes1
        syscall
```

- **Kiểm tra tính hợp lệ của xâu**
  - **Loopoflength:** duyệt qua và đếm số kí tự trong mảng khi gặp kí tự ‘\n’ thì kết thúc đếm
  - **testinput:** nếu rỗng thì yêu cầu nhập lại, không thì xử lí tiếp như đã nêu ở trong thuật toán.

```
#-------------------------------------------------------------------------
# kiem tra xau co hop le ko
length: addi    $t3, $zero, 0           # t3 = length
        addi    $t0, $zero, 0           # t0 = index

loopoflength: add $t1, $s0, $t0                 # t1 = address of string[i]
        lb      $t2, 0($t1)                     # t2 = string[i]
        nop
        beq     $t2, 10, testinput      # t2 = '\n' ket thuc xau
        nop
        addi    $t3, $t3, 1                     # length++
        addi    $t0, $t0, 1                     # index++
        j       loopoflength
        nop
testinput: move $t5, $t3
        beq     $t5, $0, nhaplai                # xau rong thi nhap lai
        # do dai xau khac 0
        andi    $t1, $t3, 0x0000000f            # lay bit cuoi cua do dai xau
        bne     $t1, 0, t1                      # byte cuoi bang 0 hoac 8 thi so chia het cho 8
        j       R_block1
t1:     beq     $t1, 8, R_block1
        j       nhaplai
nhaplai:li      $v0, 4
        la      $a0, cmt
        syscall
        j       input
```

- **Chương trình chuyển đổi kết quả xor của 2 block sang parity tương ứng**

```
#-----------------------------------------------------------------------------------------------------
#chuyen tu hex sang ascii

chuyendoi:
      addi      $k1, $ra,0
      srl       $v0, $t8,4
      jal       convert
      nop
      addi      $v0,$t8,0
      andi      $v0, $v0,0xf
      jal       convert
      nop
      addi      $ra, $k1,0
      jr        $ra
convert:
      la        $t7, hextoascii
      add       $t7, $t7,$v0
      lb        $a0, 0($t7)
      li        $v0,11
      syscall
      jr $ra
#-----------------------------------------------------------------------------------------------------
```

- **Chuyendoi:** Do kết quả mã hex trong bảng mã ascii của các kí tự chỉ gồm 2 byte nên kết quả xor của các kí tự này cũng chỉ gồm 2 byte nên ta lần lượt xor để lấy lần lượt 2 kí tự củ kết quả này cộng với địa chỉ của mảng hex, rồi truy cập để lấy mã parity tương

- **Xử lý 2 block đầu**

- Lưu 4 byte đầu tiên vào disk1, 4 byte sau vào disk2, và kết quả xor của 2 khối trên vào disk3

```
#-----------------------------------------------------------------------------------------------------
# Ham mo phong o dia
# 2 khoi dau: luu 4 byte dau vao disk1, 4 byte sau vao disk 2, va xor cua 2 khoi vao disk 3
# luu ki tu dau tien cua block1 vao disk 1,
# luu ki tu dau tien cua block2 vao disk 2,
# luu ket qua xor cua 2 ki tu tren vao disk3,
# lan luot nhu vay cho den khi het 4 ki tu thi dung lai,
# neu van con ki tu o ngoai block thi luu tiep sang cac block o dang sau.
R_block1:
        addi    $t0, $zero, 1                   # dem ki tu cua block khi nao du 4 thif dung lai


        la      $s1, disk1                      # dia chi cua disk1
        la      $s2, disk2                      # dia chi cua disk2
        la      $a2, parity                     # dia chi cua mang chua parity
        jal     print1                         # in '|       '
        nop
lasblock1:      lb      $t1, 0($s0)                     # load cac ki tu cua block1
        addi    $t3, $t3, -1                    # luu do dai xau, moi khi duyet qua 1 ki tu -1, xem da het xau ban dau chua
        sb      $t1, 0($s1)                      # luu tung ki tu cua block1 vao disk1
#load and store block2
        add     $s5, $s0, 4                     # doi s0 di 4 dv sau khi da load cac ki tu cua s1
        lb      $t2, 0($s5)                     # loa cac ki tu cua block2
        addi    $t3, $t3, -1
        sb      $t2, 0($s2)                      # luu tung ki tu cua block1 vao disk2
#load and store block1 xor block2
        xor     $a3, $t1, $t2                   # t3 luu ket qua xoa cua tung ki tu trong 2 block tren
        sw      $a3, 0($a2)                      # luu ket qua vao mang parity
        addi    $a2, $a2, 4
        addi    $t0, $t0, 1
        addi    $s0, $s0, 1                     # tang dia chi xau len 1 dv de bo ki tu da xet
        addi    $s1, $s1, 1                     # tang dia chi cua disk1 len 1 dv de luu ki tu tiep theo
        addi    $s2, $s2, 1                     # tang dia chi cua disk2 len 1 dv de luu ki tu tiep theo
        bgt     $t0, 4, reset                   # du 4 ki tu thi xg in 2 o disk1 va disk2
        j       lasblock1                            # khong thi lap lai cho den khi xet du moi 4 ki tu o 2 block
reset:  la      $s1, disk1
        la      $s2, disk2
```

```
        la      $s2, disk2
        addi    $k0, $zero, 1                # dem de in cac ki tu trong disk 1, khi nao du 4 ki tu thi dung
printdau:lb     $a0, 0($s1)
        li      $v0, 11
        syscall
        addi    $s1, $s1, 1
        addi    $k0, $k0, 1
        bgt     $k0, 4, nextf1
        j       printdau
nextf1: jal     printf2
        nop
        addi    $k0, $zero, 1                # dem de in cac ki tu trong disk 1, khi nao du 4 ki tu thi dung
printt2:lb      $a0, 0($s2)
        li      $v0, 11
        syscall
        addi    $s2, $s2, 1
        addi    $k0, $k0, 1
        bgt     $k0, 4, nextf2
        j       printt2
nextf2: jal     printf3
        nop

        la      $a2, parity
        addi    $k0, $zero, 1               # dem cac ki tu in ra trong disk3 khi nao du 4 ki tu thi dung
printt3:lb      $t8, 0($a2)
        jal     chuyendoi
        nop
        addi    $k0, $k0, 1
        bgt     $k0, 4, end2block12                  # in ra cac phan tu trong disk3
        li      $v0, 4
        la      $a0, comma
        syscall
        addi    $a2, $a2, 4
        j       printt3
end2block12:    li      $v0, 4
        la      $a0, mes5
        syscall
end2block12:    li      $v0, 4
        la      $a0, mes5
        syscall
        li      $v0, 11
        la      $a0, '\n'
        syscall
        beq     $t3, 0, exit1
#-------------------------------------------------------------------------------------------------
```

- **Xử lý 2 block tiếp theo**

- Lưu 4 byte tiếp vào disk1, 4 byte tiếp vào disk3, kết quả xor của 2 block trên vào disk2

```
R_block2: la    $a2, parity
        la      $s1, disk1
        la      $s3, disk3
        addi    $s0, $s0, 4
        addi    $t0, $zero, 1                # dem khi nao block du 4 ki tu thi dung lai
        jal     print1                        # in '|       '
        nop
lasblock3:      lb      $t1, 0($s0)
        addi    $t3, $t3, -1
        sb      $t1, 0($s1)                   # luu cac ki tu cua block3 vao disk1
#load and store block 4
        add     $s5, $s0, 4
        lb      $t2, 0($s5)
        addi    $t3, $t3, -1
        sb      $t2, 0($s3)                   # luu cac ki tu cua block4 vao disk3
#load and store block3 xor block4
        xor     $a3, $t1, $t2
        sw      $a3, 0($a2)                   # luu xor cua 2 block tren vao parity de xu ly sau
        addi    $a2, $a2, 4
        addi    $s0, $s0, 1
        addi    $s1, $s1, 1                  # tang dia chi cua disk1 len 1 dv de luu ki tu tiep theo
        addi    $s3, $s3, 1                  # tang dia chi cua disk3 len 1 dv de luu ki tu tiep theo
        addi    $t0, $t0, 1
        bgt     $t0, 4, reset2               # du 4 ki tu thi xuong reset va in
        j       lasblock3
reset2:la       $s1, disk1
        la      $s3, disk3
        addi    $k0, $zero, 1
printhai:lb     $a0, 0($s1)
        li      $v0, 11
        syscall
        addi    $k0, $k0, 1
        addi    $s1, $s1, 1
        bgt     $k0, 4, next21
```

```
        addi    $s1, $s1, 1
        bgt     $k0, 4, next21
        j       printhai
next21: jal     printf3                         # in '|       [['
        nop
        la      $a2, parity
        addi    $k0, $zero, 1                   # dem nao in du 4 ki tu thi dung lai

print23:lb      $t8, 0($a2)
        jal     chuyendoi
        nop
        addi    $k0, $k0, 1
        bgt     $k0, 4, next22
        li      $v0, 4
        la      $a0, comma
        syscall
        addi    $a2, $a2, 4
        j       print23
next22: la      $a0, mes5
        li      $v0, 4
        syscall
        jal     print1                          # in '|'
        nop
        addi    $k0, $zero, 1                   # dem nao in du 4 ki tu thi dung lai
print24:lb      $a0, ($s3)
        li      $v0, 11
        syscall
        addi    $s3, $s3, 1
        addi    $k0, $k0, 1
        bgt     $k0, 4, end2
        j       print24
end2:   li      $v0, 4
        la      $a0, mes3
        syscall
        li      $v0, 11
```

- **Xử lý 2 block cuối của 6 block đầu tiên**

- 2 block cuối cùng lần lượt được lưu vào disk2, disk3, xor của 2 khối trên được lưu vào disk1

```
# neu xau van chua ket thuc thi lap lai tu R_block1
R_block3:la     $a2, parity
        la      $s2, disk2
        la      $s3, disk3
        addi    $s0, $s0, 4
        addi    $t0, $zero, 1               # dem du 4 ki tu thi dung lai
print31:li      $v0, 4
        la      $a0, mes4
        syscall
lasblock5:      lb      $t1, 0($s0)
        addi    $t3, $t3, -1
        sb      $t1, 0($s2)                 # luu cac ki tu cua block5 vao disk2
# load and store block6
        add     $s5, $s0, 4
        lb      $t2, 0($s5)
        addi    $t3, $t3, -1
        sb      $t2, 0($s3)                 # luu cac ki tu cua block6 vao disk3
#load and store block5 xor block6
        xor     $a3, $t1, $t2
        sw      $a3, 0($a2)
        addi    $a2, $a2, 4
        addi    $t0, $t0, 1
        addi    $s0, $s0, 1
        addi    $s2, $s2, 1
        addi    $s3, $s3, 1
        bgt     $t0, 4, reset3
        j       lasblock5
reset3:la       $s2, disk2
        la      $s3, disk3
        la      $a2, parity
        addi    $k0, $zero, 1
print32:lb      $t8, 0($a2)
        jal     chuyendoi
        nop
        addi    $k0, $k0, 1
        bgt     $k0, 4, next31
        li      $v0, 4
        la      $a0, comma
```

```
        li      $v0, 4
        la      $a0, comma
        syscall
        addi    $a2, $a2, 4
        j       print32
next31: li      $v0, 4
        la      $a0, mes5
        syscall
        jal     print1                          # in '|'
        nop
        addi    $k0, $zero, 1
print33:lb      $a0, ($s2)
        li      $v0, 11
        syscall
        addi    $k0, $k0, 1
        addi    $s2, $s2, 1
        bgt     $k0, 4, next32
        j       print33
next32: jal     printf2
        nop
        addi    $k0, $zero, 1
print34:lb      $a0, 0($s3)
        li      $v0, 11
        syscall
        addi    $s3, $s3, 1
        addi    $k0, $k0, 1
        bgt     $k0, 4, end3
        j       print34
end3:   li      $v0,4
        la      $a0, mes3
        syscall
        li      $v0, 11
        li      $a0, '\n'
        syscall
        beq     $t3, 0, exit1
```

- Trường hợp xâu vẫn chưa được duyệt hết
- **Existblock**: jump đến chỗ xử lí 2 block đầu tiên để lặp lại việc xử lý.

```
#-----------------------------------------------------------------------------------------
# neu xau van chua ket thuc
# lap lai cac buoc nhu tren
existblock: addi $s0, $s0, 4
            j R_block1
# in ra ki tu ket thuc chuong trinh
exit1:  li      $v0, 4
        la      $a0, mes1
        syscall
        j       choice
```

- Chọn xem có tiếp tục chương trình hay không.

```
#
choice: li      $v0, 51
        la      $a0, mutiplechoice
        syscall
        beq     $a0, 1, resetprogram
        nop
        j       exit
        nop
# reset lai chuong trinh
resetprogram:   la      $s0, string
                add     $s3, $s0, $t5           # s3: dia chi byte cuoi cung duoc su dung trong string
                li      $t1, 0
loop:   sb      $t1, 0($s0)             # set byte o dia chi s0 thanh 0
        nop
        addi    $s0, $s0, 1
        bge     $s0, $s3, input
        nop
        j       loop
        nop
#-------------------------------------------------------------------------------

exit:   li      $v0, 10
        syscall
```

3. Kết quả

- **Nếu xâu nhập vào là xâu rỗng:**

```
Xin moi nhap xau :
      Disk 1               Disk 2              Disk 3
---------------       ---------------       ---------------
Do dai xau phai la boi cua 8. Xin moi nhap lai theo dinh dang.
Xin moi nhap xau :
```

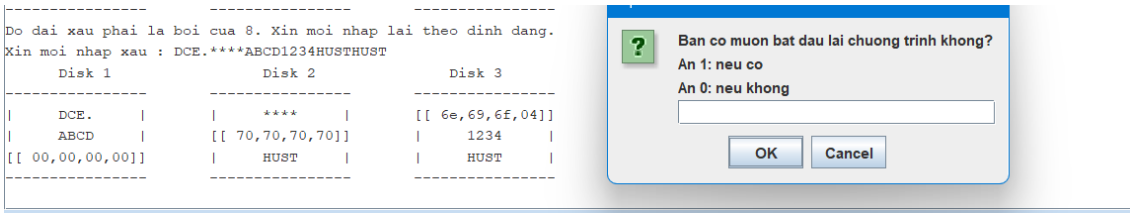
- **Nếu xâu nhập khác rỗng**

- Độ dài xâu không phải bội của 8:

```
Xin moi nhap xau : abc
      Disk 1                Disk 2                Disk 3
----------------       ----------------       ----------------
Do dai xau phai la boi cua 8. Xin moi nhap lai theo dinh dang.
Xin moi nhap xau :
```

- Độ dài xâu là bội của 8:

```
Xin moi nhap xau : 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      |
----------------       ----------------       ----------------
```



- **Nếu ấn 0:**

```
Do dai xau phai la boi cua 8. Xin moi nhap lai theo dinh dang.
Xin moi nhap xau : 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      |
----------------      ----------------      ----------------

-- program is finished running --
```

- **Nếu ấn 1:**

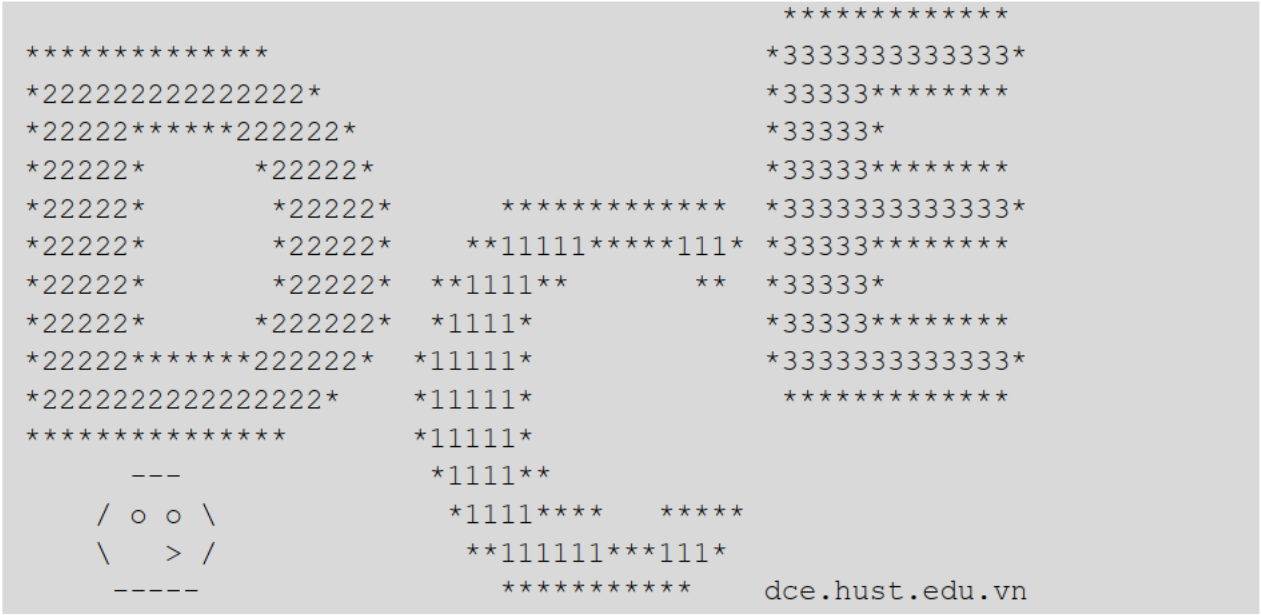
```
Xin moi nhap xau : DCE.****ABCD1234HUSTHUST
      Disk 1                Disk 2               Disk 3
----------------      ----------------      ----------------
|      DCE.      |     |      ****      |     [[ 6e,69,6f,04]]
|      ABCD      |     [[ 70,70,70,70]]  |     |      1234      |
[[ 00,00,00,00]]      |      HUST      |     |      HUST      |
----------------      ----------------      ----------------
Xin moi nhap xau :
```

II. Bài 9: Vẽ hình bằng kí tự ASCII

Người thực hiện: Lương Văn Khanh

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 hoạt 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 project:

2.1 Giải thích code

- Phần *.data* khai báo từng dòng của hình ảnh cần in và menu lựa chọn

```
.data
line1: .asciiz  "                                          *************       \n"
line2: .asciiz  " **************                          *3333333333333*      \n"
line3: .asciiz  " *222222222222222*                        *33333********       \n"
line4: .asciiz  " *22222******222222*                      *33333*              \n"
line5: .asciiz  " *22222*      *22222*                     *33333********       \n"
line6: .asciiz  " *22222*       *22222*        *************  *3333333333333*      \n"
line7: .asciiz  " *22222*       *22222*      **11111*****111* *33333********       \n"
line8: .asciiz  " *22222*       *22222*    **1111**       **  *33333*              \n"
line9: .asciiz  " *22222*      *222222*    *1111*             *33333********       \n"
line10: .asciiz " *22222*******222222*    *11111*             *3333333333333*      \n"
line11: .asciiz " *2222222222222222*      *11111*              *************       \n"
line12: .asciiz " ***************         *11111*                                  \n"
line13: .asciiz "      ---                 *1111**                                 \n"
line14: .asciiz "    / o o \\               *1111****   *****                       \n"
line15: .asciiz "    \\   > /                **111111***111*                       \n"
line16: .asciiz "     -----                   **********      dce.hust.edu.vn    \n"

menu_message: .asciiz "\n\n ----MENU----\n 1. Show picture.\n 2. Show picture with only border.\n 3. Change the order.\n 4. I
error_message: "Input must be a integer from 1 to 5"

input_d_color: .asciiz "Enter color for D (integer from 0-9):"
input_c_color: .asciiz "Enter color for C (integer from 0-9):"
input_e_color: .asciiz "Enter color for E (integer from 0-9):"
```

- Vào phần *.text* , in ra menu lựa chọn, các rẽ nhánh của 5 lựa chọn và cửa sổ yêu cầu người dùng nhập lựa chọn
  - Nếu **lựa chọn 1** (in ra hình DCE) thì nhảy đến menu_func1->print , thực hiện vòng lặp và in ra cửa sổ console hình ảnh DCE
  - Nếu **lựa chọn 2** (in ra chữ DCE chỉ có viền) thì nhảy đến menu_func2. Tại đây sẽ duyệt từng dòng , nếu kí tự được duyệt là chữ số (Có giá trị ASCII là 48 – 57) thì thay thế nó thành kí tự Space (Có giá trị ASCII là 32) rồi in ra

```
menu_func2:
 li $t0,16
 li $t1,68
 li $t2,0
 li $t3,0
 la $s0,line1
loop_row:
 beq $t3,$t0,end_loop_row
 li $t2,0
loop_column:
 beq $t2,$t1,end_loop_column
 lb $a1,0($s0)
 addi $s0,$s0,1
 bgt $a1,57,print_char
 blt $a1,48,print_char
 li $a1,32
print_char:
 li $v0, 11
 move $a0,$a1
 syscall
 addi $t2,$t2,1
 j loop_column
end_loop_column:
 addi $t3,$t3,1
 j loop_row
```

- Nếu **lựa chọn 3** (Hoán đổi vị trí) thì nhảy đến menu func3.Chương trình thực hiện lặp qua các hàng(12) và các cột (22) cụ thể để hoán đổi vị trí các kí tự D,E,C . Lặp lại cho đến khi tất cả các hàng và các cột được xử lý rồi in ra màn hình hình ảnh đã được cập nhật (ECD)

```
func3:
 li $t0,12
 li $t1,22
 li $t2,1
 li $t3,1
 la $a0,line1
 add $a0,$a0,68
change_order_row_loop:
 beq $t2,$t0,end_change_order_row_loop
 li $t3,1
change_order_column_loop:
 beq $t3,$t1,end_change_order_column_loop
 add $a1,$a0,$t3
 lb $s1,0($a1) #char in d
 sub $a2,$a1,24
 lb $s2,0($a2) #char in e
 add $a3,$a1,294
 lb $s3,0($a3) #char in c
 sb $s2,0($a1)
 sb $s1,0($a2)
 addi $t3,$t3,1
 j change_order_column_loop
end_change_order_column_loop:
 addi $a0,$a0,68
 addi $t2,$t2,1
```

- Nếu **lựa chọn 4** (thay màu cho hình) thì nhảy đến menu_func4.

Lần lượt hiển thị thông báo và yêu cầu người dùng nhập màu cho các kí tự D , C , E. Màu sẽ là 1 số từ 0 đến 9 , nếu nhập ngoài khoảng sẽ yêu cầu người dùng nhập lại.

Khi đã có đủ 3 kí tự hợp lệ , chương trình thực hiện cập nhật màu sắc (hàm change_color)

Một vòng lặp qua các hàng , các cột để update lại giá trị

Cuối cùng hàm print in ra hình ảnh mới sau khi cập nhật lại màu cho các kí tự

```
menu_func4:
input_d_co:
 li $v0,4
 la $a0,input_d_color
 syscall
 li $v0,5
 syscall
 bgt $v0,9,input_d_co
 blt $v0,0,input_d_co
 addi $t4,$v0,48 # d color
input_c_co:
 li $v0,4
 la $a0,input_c_color
 syscall
 li $v0,5
 syscall
 bgt $v0,9,input_c_co
 blt $v0,0,input_c_co
 addi $t5,$v0,48
input_e_co:
 li $v0,4
 la $a0,input_e_color
 syscall
 li $v0,5
 syscall
 bgt $v0,9,input_e_co
 blt $v0,0,input_e_co
 addi $t6,$v0,48
change_color:
 li $t0,12
 li $t1,22
 li $t2,1
 li $t3,1
 la $a0,line1
 addi $a0,$a0,68
change_color_row_loop:
 beq $t2,$t0,end_change_color_row_loop
 li $t3,1
change_color_column_loop:
 beq $t3,$t1,end_change_color_column_loop
 add $a1,$a0,$t3
 lb $s1,0($a1) #char in d
 move $t8,$t4
 jal modifycolor
 sb $s1,0($a1)
```

```
177   addi $t3,$t3,1
178   j change_color_column_loop
179  end_change_color_column_loop:
180   addi $a0,$a0,68
181   addi $t2,$t2,1
182   j change_color_row_loop
183  end_change_color_row_loop:
184   jal print
185   j menu
186
187  modifycolor:
188   bgt $s1,57,end_modify
189   blt $s1,48,end_modify
190   move $s1,$t8
191  end_modify:
192   jr $ra
193
194  print:
195   li $t0,16
196   li $t1,0
197   la $a0,line1
198  print_loop:
199   beq $t1,$t0,end_print_loop
200   li $v0,4
201   syscall
202   addi $a0,$a0,68
203   addi $t1,$t1,1
204   j print_loop
205  end_print_loop:
206   jr $ra
```

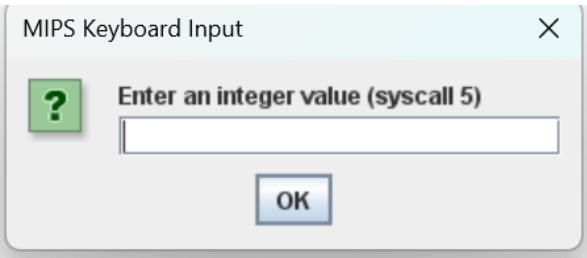
2.2.Kết quả thực hiện

a, Các ngoại lệ

- Ban đầu nếu nhập lựa chọn 6 (ngoài số từ 1 đến 5) sẽ báo lỗi và bắt nhập lại

```
 Enter your choice: **** user input : 6
Input must be a integer from 1 to 5

 ----MENU----
 1. Show picture.
 2. Show picture with only border.
 3. Change the order.
 4. Enter new color number and update.
 5. Exit.
 Enter your choice:
```



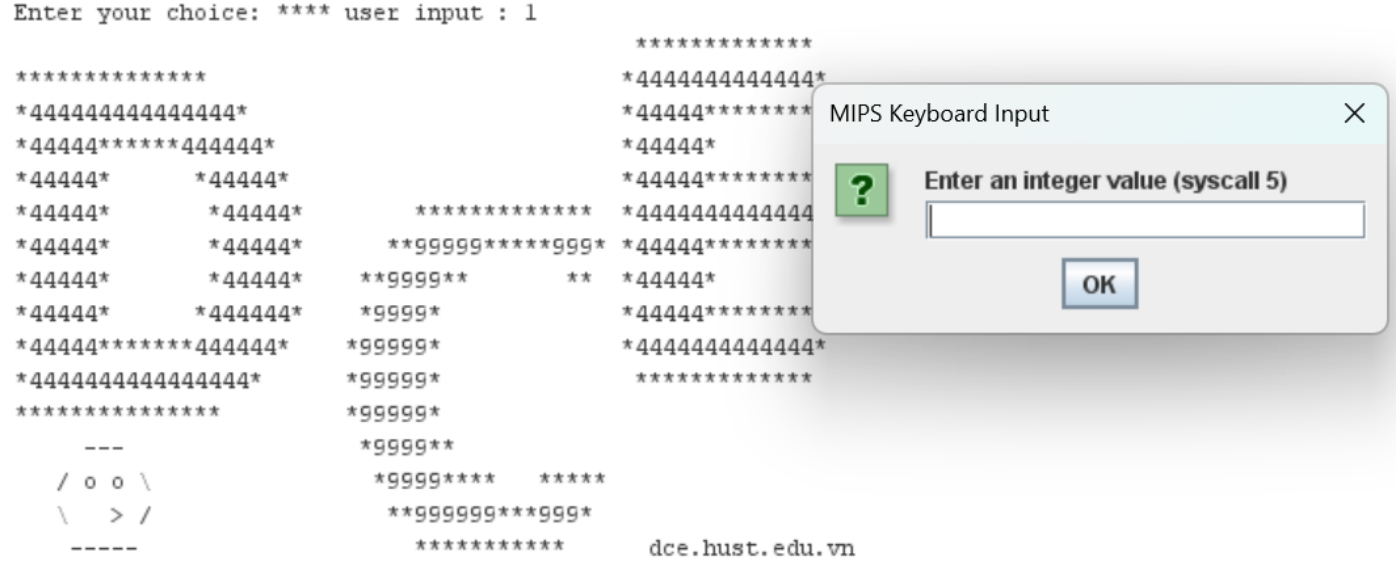
- Khi vào lựa chọn 4 (Thay đổi màu) nếu nhập 1 số ngoài khoảng 0 đến 9 thì sẽ yêu cầu nhập lại cho đến khi đủ 3 màu thỏa mãn

```
 Enter your choice: **** user input : 4
Enter color for D (integer from 0-9):**** user input : 4
Enter color for C (integer from 0-9):**** user input : -1
Enter color for C (integer from 0-9):**** user input : 10
Enter color for C (integer from 0-9):**** user input : 9
Enter color for E (integer from 0-9):**** user input : 4
```

b, Kết quả hợp lệ

- Lựa chọn 1:In ra hình ban đầu

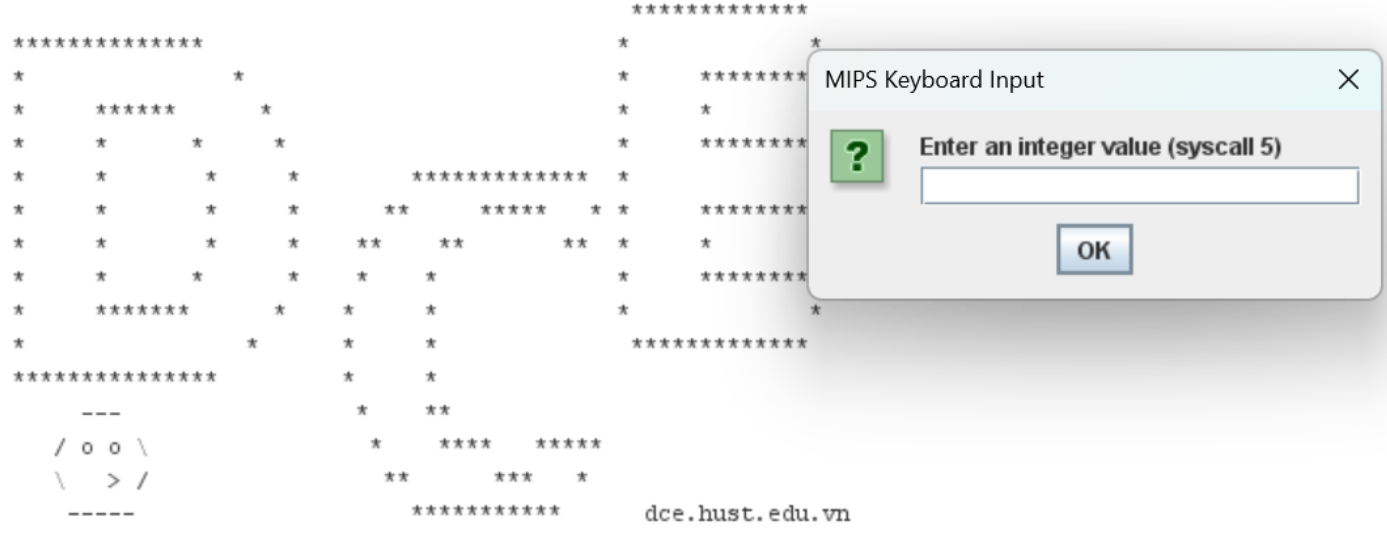
```
1. Show picture.
2. Show picture with only border.
3. Change the order.
4. Enter new color number and update.
5. Exit.
Enter your choice: **** user input : 1
```




```
     dce.hust.edu.vn
```

- Lựa chọn 2:Chỉ in ra viền

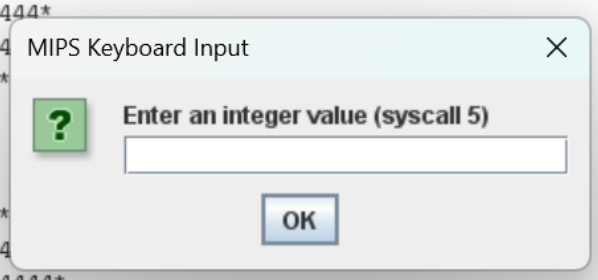
```
1. Show picture.
2. Show picture with only border.
3. Change the order.
4. Enter new color number and update.
5. Exit.
Enter your choice: **** user input : 2
```




```
     dce.hust.edu.vn
```

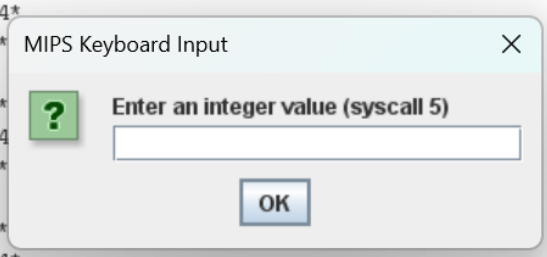
- Lựa chọn 3 : Hoán đổi vị trí

```
Enter your choice: **** user input : 3
                                                **************
 **************                                 *4444444444444444*
*4444444444444*                                 *44444******44
*44444********                                  *44444*      *
*44444*                                         *44444*
*44444********                 *************    *44444*
*4444444444444*              **99999*****999*   *44444*
*44444********              **9999**       **   *44444*      *
*44444*                     *9999*              *44444*******4
*44444********             *99999*              *4444444444444444*
*4444444444444*            *99999*              ***************
 **************            *99999*
     ---                    *9999**
   / o o \                   *9999****   *****
   \   > /                    **99999999999*
    -----                       **********      dce.hust.edu.vn
```



- Lựa chọn 4 : Thay màu , giả sử chọn 1,7,4

```
 Enter your choice: **** user input : 4
Enter color for D (integer from 0-9):**** user input : 1
Enter color for C (integer from 0-9):**** user input : 7
Enter color for E (integer from 0-9):**** user input : 4
                                                 *************
 **************                                 *4444444444444*
 *lllllllllllllll*                              *44444********
 *llllll******llllll*                           *44444*
 *llllll*      *llllll*                         *44444********
 *llllll*       *llllll*        *************   *4444444444444
 *llllll*       *llllll*      **77777******777* *44444********
 *llllll*       *llllll*    **7777**       **   *44444*
 *llllll*      *llllll*     *7777*              *44444********
 *llllll******llllll*     *77777*              *4444444444444*
 *lllllllllllllllll*      *77777*               *************
 ***************          *77777*
      ---                  *7777**
    / o o \                 *7777****   *****
    \   > /                  **77777777***777*
     -----                     **********      dce.hust.edu.vn
```



- Lựa chọn 5 : Exit

```
----MENU----
1. Show picture.
2. Show picture with only border.
3. Change the order.
4. Enter new color number and update.
5. Exit.
Enter your choice: **** user input : 5

-- program is finished running --
```