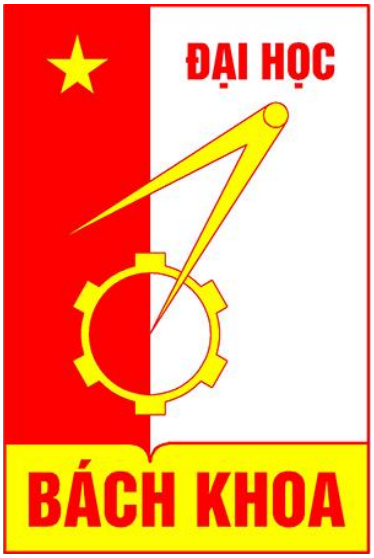
BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ thông tin và truyền thông
----- o0o -----



BÁO CÁO THỰC HÀNH

KIẾN TRÚC MÁY TÍNH

Mã học phần: IT3280

Mã lớp: 130938

Sinh viên thực hiện:

Lê Trung Kiên MSSV: 20194596

Giảng viên hướng dẫn: Lê Bá Vui

Project 8. Mô phỏng ổ đĩa RAID 5

Mô phỏng ổ đĩa RAID 5 Hệ thống ổ đĩa RAID5 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.

1. **Phân tích cách thực hiện o Phân tích đề bài:**
   - ▪ Đầu vào là 1 chuỗi ký tự có độ dài là bội của 8
   - ▪ Đầu ra là in ra màn hình console giả lập hệ thống ổ đĩa RAID 5
   - ➢ Chương trình gồm:
     - ▪ Nhập chuỗi từ bàn phím
     - ▪ Kiểm tra mức độ hợp lệ của dữ liệu nhập từ bàn phím
     - ▪ Chương trình tìm tính mã parity
     - ▪ Chương trình in ra màn hình giả lập hệ thống ổ đĩa RAID 5 o Cách chạy chương trình:
     - ▪ Nhập 1 chuỗi ký tự từ bàn phím độ dài là bội của 8
     - ▪ Chương trình in ra giả lập hệ thống ổ đĩa
     - ▪ Lựa chọn thực hiện lại chương trình hoặc kết thúc chương trình
   - ➢ 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.

Mã Nguồn:

```
.data

start: .asciiz "Nhap chuoi ky tu : "

hex: .byte '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'

d1: .space 4
```

```
d2: .space 4
d3: .space 4
array: .space 32
string: .space 5000
enter: .asciiz "\n"
error_length: .asciiz "Do dai chuoi khong hop le! Nhap lai.\n"
m: .asciiz "      Disk 1              Disk 2              Disk 3\n"
m2: .asciiz "----------------      ----------------      ----------------\n"
m3: .asciiz "|     "
m4: .asciiz "     |       "
m5: .asciiz "[[ "
m6: .asciiz "]]      "
comma: .asciiz ","
ms: .asciiz "Try again?"

.text
        la $s1, d1
        la $s2, d2
        la $s3, d3
        la $a2, array               # dia chi mang chua parity

input:      li $v0, 4               # nhap ten (chuoi)
```

```
        la $a0, start
        syscall
        li $v0, 8
        la $a0, string
        li $a1, 1000
        syscall
        move $s0, $a0           # s0 chua dia chi xau moi nhap
        li $v0, 4
        la $a0, m
        syscall
        li $v0, 4
        la $a0, m2
        syscall

#----------------------kiem tra do dai co chia het cho 8 khong----------------
--------
length: addi $t3, $zero, 0    # t3 = length
        addi $t0, $zero, 0      # t0 = index

check_char: add $t1, $s0, $t0      # t1 = address of string[i]
        lb $t2, 0($t1)            # t2 = string[i]
        nop
```

```
        beq $t2, 10, test_length      # t2 = '\n' ket thuc xau
        nop
        addi $t3, $t3, 1   # length++
        addi $t0, $t0, 1   # index++
        j check_char
        nop
test_length: move $t5, $t3
        and $t1, $t3, 0x0000000f            # xoa het cac byte cua $t3 ve 0, chi giu lai byte cuoi
        bne $t1, 0, test1                   # byte cuoi bang 0 hoac 8 thi so chia het cho 8
        j split1
test1:        beq $t1, 8, split1
        j error1
error1:       li $v0, 4
        la $a0, error_length
        syscall
        j input
#------------------------------ket thuc kiem tra do dai-------------------------------


#-----------------------lay parity---------------------------
HEX:  li $t4, 7
```

```
loopH:      blt $t4, $0, endloopH
        sll $s6, $t4, 2             # s6 = t4*4
        srlv $a0, $t8, $s6          # a0 = t8>>s6
        andi $a0, $a0, 0x0000000f  # a0 = a0 & 0000 0000 0000 0000
0000 0000 0000 1111 => lay byte cuoi cung cua a0
        la $t7, hex
        add $t7, $t7, $a0
        bgt $t4, 1, nextc
        lb $a0, 0($t7)              # print hex[a0]
        li $v0, 11
        syscall
        # lb $t6, 0($t7)
        # beq $t6, 48, in
nextc:      addi $t4,$t4,-1
        j loopH
# in:  bgt $t4, 1, loopH
        # move $a0, $t6
        # li $v0, 11
        # syscall
endloopH: jr $ra
#---------------------------------------------------------
```

```
#-----------------------------mo phong RAID 5-------------------------------
#----------------------xet 6 khoi dau----------------------
#----------------lan 1: luu vao 2 khoi 1,2; xor vao 3-------
split1:     addi $t0, $zero, 0      # so byte duoc in ra (4 byte)
        addi $t9, $zero, 0
        addi $t8, $zero, 0
        la $s1, d1
        la $s2, d2
        la $a2, array
print11:li $v0, 4
        la $a0, m3
        syscall
b11:  lb $t1, ($s0)
        addi $t3, $t3, -1
        sb $t1, ($s1)
b21:  add $s5, $s0, 4
        lb $t2, ($s5)           # t2 chua dia chi tung byte cua dick 2
        addi $t3, $t3, -1
        sb $t2, ($s2)
b31:  xor $a3, $t1, $t2
        sw $a3, ($a2)
        addi $a2, $a2, 4
```

```
        addi $t0, $t0, 1
        addi $s0, $s0, 1
        addi $s1, $s1, 1
        addi $s2, $s2, 1
        bgt $t0, 3, reset
        j b11
reset:  la $s1, d1
        la $s2, d2
print12:lb $a0, ($s1)
        li $v0, 11
        syscall
        addi $t9, $t9, 1
        addi $s1, $s1, 1
        bgt $t9, 3, next11
        j print12
next11: li $v0, 4
        la $a0, m4
        syscall
        li $v0, 4
        la $a0, m3
        syscall
print13:lb $a0, ($s2)
```

```
        li $v0, 11
        syscall
        addi $t8, $t8, 1
        addi $s2, $s2, 1
        bgt $t8, 3, next12
        j print13
next12:     li $v0, 4
        la $a0, m4
        syscall
        li $v0, 4
        la $a0, m5
        syscall

        la $a2, array
        addi $t9, $zero, 0
print14:lb $t8, ($a2)
        jal HEX
        li $v0, 4
        la $a0, comma
        syscall
        addi $t9, $t9, 1
        addi $a2, $a2, 4
```

```
        bgt $t9, 2, end1 # in ra 3 parity dau co dau ",", parity cuoi cung k
co
        j print14
end1:       lb $t8, ($a2)
        jal HEX
        li $v0, 4
        la $a0, m6
        syscall
        li $v0, 4
        la $a0, enter
        syscall
        beq $t3, 0, exit1
#-------------------------------------------------------
split2:     la $a2, array
        la $s1, d1
        la $s3, d3
        addi $s0, $s0, 4
        addi $t0, $zero, 0
print21:li $v0, 4
        la $a0, m3
        syscall
b12:  lb $t1, ($s0)
```

```
        addi $t3, $t3, -1
        sb $t1, ($s1)
b32:    add $s5, $s0, 4
        lb $t2, ($s5)
        addi $t3, $t3, -1
        sb $t2, ($s3)
b22:    xor $a3, $t1, $t2
        sw $a3, ($a2)
        addi $a2, $a2, 4
        addi $t0, $t0, 1
        addi $s0, $s0, 1
        addi $s1, $s1, 1
        addi $s3, $s3, 1
        bgt $t0, 3, reset2
        j b12
reset2:     la $s1, d1
        la $s3, d3
        addi $t9, $zero, 0
print22:lb $a0, ($s1)
        li $v0, 11
        syscall
        addi $t9, $t9, 1
```

```
        addi $s1, $s1, 1
        bgt $t9, 3, next21
        j print22
next21:     li $v0, 4
        la $a0, m4
        syscall
        la $a2, array
        addi $t9, $zero, 0
        li $v0, 4
        la $a0, m5
        syscall
print23:lb $t8, ($a2)
        jal HEX
        li $v0, 4
        la $a0, comma
        syscall
        addi $t9, $t9, 1
        addi $a2, $a2, 4
        bgt $t9, 2, next22
        j print23
next22:     lb $t8, ($a2)
        jal HEX
```

```
        li $v0, 4
        la $a0, m6
        syscall
        li $v0, 4
        la $a0, m3
        syscall
        addi $t8, $zero, 0
print24:lb $a0, ($s3)
        li $v0, 11
        syscall
        addi $t8, $t8, 1
        addi $s3, $s3, 1
        bgt $t8, 3, end2
        j print24

end2:       li $v0, 4
        la $a0, m4
        syscall
        li $v0, 4
        la $a0, enter
        syscall
        beq $t3, 0, exit1
```

```
#-------------------------------------------------------------
split3:     la $a2, array
        la $s2, d2
        la $s3, d3
        addi $s0, $s0, 4
        addi $t0, $zero, 0
print31:li $v0, 4
        la $a0, m5
        syscall
b23:  lb $t1, ($s0)
        addi $t3, $t3, -1
        sb $t1, ($s1)
b33:  add $s5, $s0, 4
        lb $t2, ($s5)
        addi $t3, $t3, -1
        sb $t2, ($s3)
b13:  xor $a3, $t1, $t2
        sw $a3, ($a2)
        addi $a2, $a2, 4
        addi $t0, $t0, 1
        addi $s0, $s0, 1
        addi $s1, $s1, 1
```

```
        addi $s3, $s3, 1
        bgt $t0, 3, reset3
        j b23
reset3:     la $s2, d2
        la $s3, d3
        la $a2, array
        addi $t9, $zero, 0
print32:lb $t8, ($a2)
        jal HEX
        li $v0, 4
        la $a0, comma
        syscall
        addi $t9, $t9, 1
        addi $a2, $a2, 4
        bgt $t9, 2, next31
        j print32
next31:     lb $t8, ($a2)
        jal HEX
        li $v0, 4
        la $a0, m6
        syscall
        li $v0, 4
```

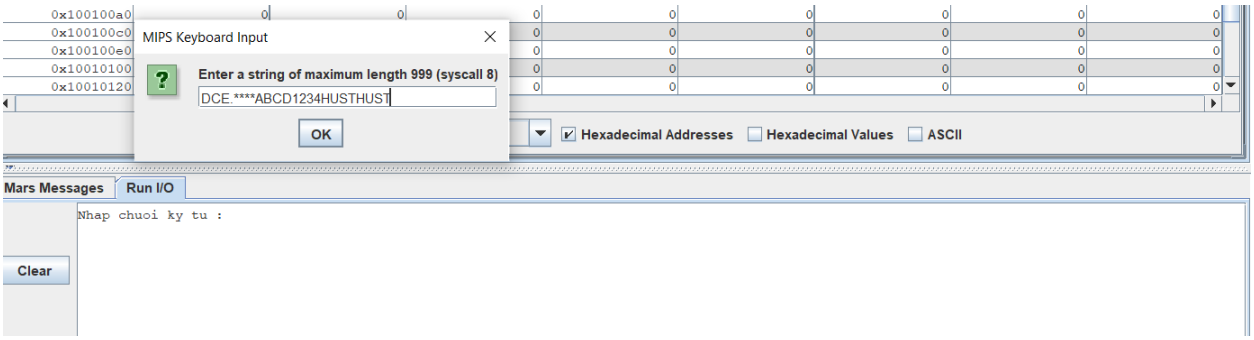
```
        la $a0, m3
        syscall
        addi $t9, $zero, 0
print33:lb $a0, ($s2)
        li $v0, 11
        syscall
        addi $t9, $t9, 1
        addi $s2, $s2, 1
        bgt $t9, 3, next32
        j print33
next32:     addi $t9, $zero, 0
        addi $t8, $zero, 0
        li $v0, 4
        la $a0, m4
        syscall
        li $v0, 4
        la $a0, m3
        syscall
print34:lb $a0, ($s3)
        li $v0, 11
        syscall
        addi $t8, $t8, 1
```

```
        addi $s3, $s3, 1
        bgt $t8, 3, end3
        j print34

end3:   li $v0, 4
        la $a0, m4
        syscall
        li $v0, 4
        la $a0, enter
        syscall
        beq $t3, 0, exit1
#------------------------------end 6 khoi dau-------------------------------------
# chuyen sang 6 khoi tiep theo
nextloop: addi $s0, $s0, 4
        j split1

exit1:  li $v0, 4
        la $a0, m2
        syscall
        j ask
#------------------------------ket thuc mo phong RAID 5-------------------------
-------
```

```
#----------------------------------try again--------------------------------------
ask:   li $v0, 50
       la $a0, ms
       syscall
       beq $a0, 0, clear
       nop
       j exit
       nop
# clear: dua string ve trang thai ban dau de thuc hien lai qua trinh
clear:la $s0, string
       add $s3, $s0, $t5        # s3: dia chi byte cuoi cung duoc su dung
trong string
       li $t1, 0
goAgain: sb $t1, ($s0)          # set byte o dia chi s0 thanh 0
       nop
       addi $s0, $s0, 1
       bge $s0, $s3, input
       nop
       j goAgain
       nop
#---------------------------------end try again-------------------------------------
```

```
exit:  li $v0, 10
       syscall
```

Chạy code:



Disk 1	Disk 2	Disk 3
DCE.	****	[[6e,69,6f,04]]
ABCD	[[70,70,70,70]]	1234
[[00,00,00,00]]	HUST	HUST