ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH

BÁO CÁO BÀI THỰC HÀNH 2:
CE119 - THỰC HÀNH KIẾN TRÚC MÁY TÍNH

CHỦ ĐỀ
CE119-Lab02/IT012-Lab04

Giảng viên hướng dẫn:ThS. Trần Văn Quang

Sinh viên thực hiện:

Đỗ Minh Hội - 23520550

1. Lý thuyết

Giảng viên hướng dẫn sinh viên về chương trình hợp ngữ MIPS dựa theo tài liệu: Tổng quát về hợp ngữ và kiến trúc MIPS

Danh sách lệnh giả:
bgt(branch if greater than)
bgt $t1,$t2,end # nhảy đến end nếu $t1>$t2
bge (branch if greater than or equal)
bge $t1,$t2,end # nhảy đến end nếu $t1>=$t2
blt(branch if less than)
blt $t1,$t2,end #nhảy đến end nếu $t1<$t2
ble(branch if less than or equal)
ble $t1,$t2,end #nhảy đến end nếu $t1<=$t2

2. Thực hành

a.Chuyển đoạn code trong bảng theo sau sang MIPS và sử dụng MARS để kiểm tra lại kết quả:

```
if (i == j)
 f = g + h;
else
 f = g – h;
```

(Với giá trị của i, j, f, g, h lần lượt chứa trong các thanh ghi $s0, $s1, $s2, $t0, $t1)

```
beq $s0,$s1,ifequal
sub $s2,$t0,$t1
ifequal:
add $s2,$t0,$t1
```

```
int Sum = 0
for (int i = 1; i <=N; ++i){
 Sum = Sum + i;
```

```
}
```

(Với giá trị của i, N, Sum lần lượt chứa trong các thanh ghi $s0, $s1, $s2)

```
li $s2,0
li $s0,1

loop:
bgt $s0,$s1,end
add $s2,$s2,$s0
addi $s0,$s0,1
j loop

end:
```

3. Bài tập

a. Nhập vào một ký tự, xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

✓ Ký tự liền trước và liền sau của ký tự nhập vào

Ví dụ:

Nhap ky tu (chỉ một ký tự): b

Ky tu truoc: a

Ky tu sau: c

✓ Ký tự nhập vào chỉ được phép là ba loại: số, chữ thường và chữ hoa. Nếu ký tự nhập vào rơi vào một trong ba loại, xuất ra cửa sổ đó là loại nào; nếu ký tự nhập không rơi vào một trong ba loại trên, xuất ra thông báo "invalid type"

*Ý tưởng:

1 **Hiển thị lời nhắc nhập ký tự (ms1)**:

- Sử dụng syscall để in chuỗi yêu cầu người dùng nhập ký tự.
- Chờ người dùng nhập vào một ký tự, lưu vào $t0.

2 **Xác định loại ký tự**:

- **Kiểm tra ký tự số**: Nếu giá trị của ký tự trong khoảng từ 48 ('0') đến 57 ('9'), chương trình xác nhận đây là ký tự số và in thông báo (ms2).

- **Kiểm tra ký tự chữ thường**: Nếu giá trị của ký tự trong khoảng từ 97 ('a') đến 122 ('z'), chương trình xác nhận đây là ký tự chữ thường và in thông báo (ms3).
- **Kiểm tra ký tự chữ hoa**: Nếu giá trị của ký tự trong khoảng từ 65 ('A') đến 90 ('Z'), chương trình xác nhận đây là ký tự chữ hoa và in thông báo (ms4).
- **Ký tự không xác định**: Nếu ký tự không thuộc bất kỳ loại nào kể trên, chương trình xác nhận đây là loại ký tự không xác định và in thông báo (ms5).

3 **Hiển thị ký tự liền trước và liền sau**:

- **Ký tự liền trước** (ms6): Trừ 1 từ giá trị ký tự nhập vào và hiển thị ký tự liền trước.
- **Ký tự liền sau** (ms7): Cộng 1 vào giá trị ký tự nhập vào và hiển thị ký tự liền sau.

4 **Kết thúc chương trình**:

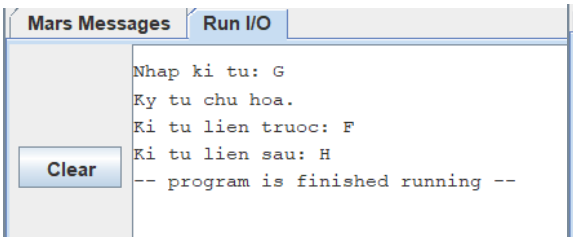
- Gọi syscall để kết thúc chương trình.

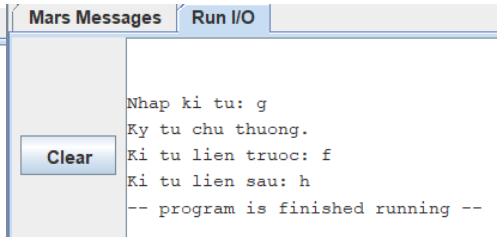
-Thực hiện viết chương trình

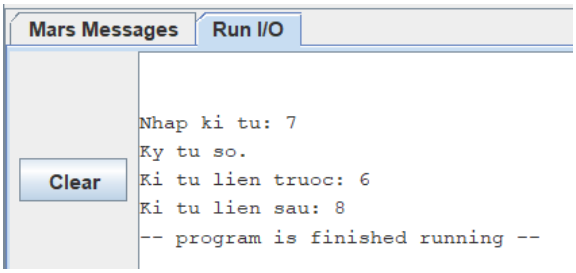
```
.data
  ms1: .asciiz "\nNhap ki tu: "
  ms2: .asciiz "\nKy tu so."
  ms3: .asciiz "\nKy tu chu thuong."
  ms4: .asciiz "\nKy tu chu hoa."
  ms5: .asciiz "\ninvalid type."
  ms6: .asciiz "\nKi tu lien truoc: "
  ms7: .asciiz "\nKi tu lien sau: "
.text
  li $v0,4
  la $a0,ms1
  syscall
  li $v0,12
  syscall
  move $t0,$v0
  li $t1,48
  li $t2,57
  blt $t0, $t1 check_lower
  bgt $t0, $t2 check_lower
  li $v0,4
  la $a0,ms2
  syscall
  j inkytu
 check_lower:
  li $t3,97
  li $t4,123
  blt $t0, $t3 check_upper
  bgt $t0, $t4 check_upper
  li $v0,4
  la $a0,ms3
  syscall
  j inkytu
```

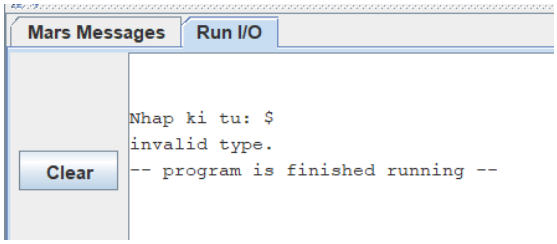
```
check_upper:
  li $t5,65
  li $t6,90
  blt $t0, $t5 invalid
  bgt $t0, $t6 invalid
  li $v0,4
  la $a0,ms4
  syscall
  j inkytu
invalid:
  li $v0,4
  la $a0,ms5
  syscall
  j end
inkytu:
  li $v0,4
  la $a0,ms6
  syscall
  sub $t1,$t0,1
  li $v0,11
  move $a0,$t1
  syscall
  li $v0,4
  la $a0,ms7
  syscall
  add $t2,$t0,1
  li $v0,11
  move $a0,$t2
  syscall
  j end
end:
  li $v0,10
  syscall
```

-Thực hiện chạy chương trình









b. Nhập từ bàn phím 2 số nguyên, in ra cửa sổ I/O của MARS theo từng yêu cầu sau:

✓ Số lớn hơn

✓ Tổng, hiệu, tích và thương của hai số

*Ý tưởng:

1 Nhập hai số nguyên từ người dùng:

- Hiển thị thông báo yêu cầu người dùng nhập số nguyên thứ nhất (nhap1) và nhận giá trị nhập vào, lưu vào thanh ghi $t0.
- Hiển thị thông báo yêu cầu người dùng nhập số nguyên thứ hai (nhap2) và nhận giá trị nhập vào, lưu vào thanh ghi $t1.

2 Tìm số lớn hơn:

- So sánh $t0 và $t1. Nếu $t0 lớn hơn hoặc bằng $t1, thì $t0 là số lớn hơn, và giá trị này được gán cho $t2.
- Nếu $t1 lớn hơn $t0, gán $t1 cho $t2.
- Hiển thị kết quả số lớn hơn (ketqua1) và in giá trị trong $t2.

3 Tính và hiển thị các phép tính:

- **Tổng** (ketqua2): Tính tổng của $t0 và $t1, lưu kết quả vào $t3 và hiển thị kết quả.
- **Hiệu** (ketqua3): Tính hiệu của $t0 trừ $t1, lưu kết quả vào $t4 và hiển thị kết quả.

- **Tích** (ketqua4): Tính tích của $t0 và $t1, lưu kết quả vào $t5 và hiển thị kết quả.
- **Thương** (ketqua5): Trước khi thực hiện phép chia, chương trình kiểm tra $t1 có bằng 0 không (để tránh chia cho 0). Nếu $t1 khác 0, tiến hành phép chia $t0 / $t1, lưu thương vào $t6 và hiển thị kết quả.

4 Kết thúc chương trình:

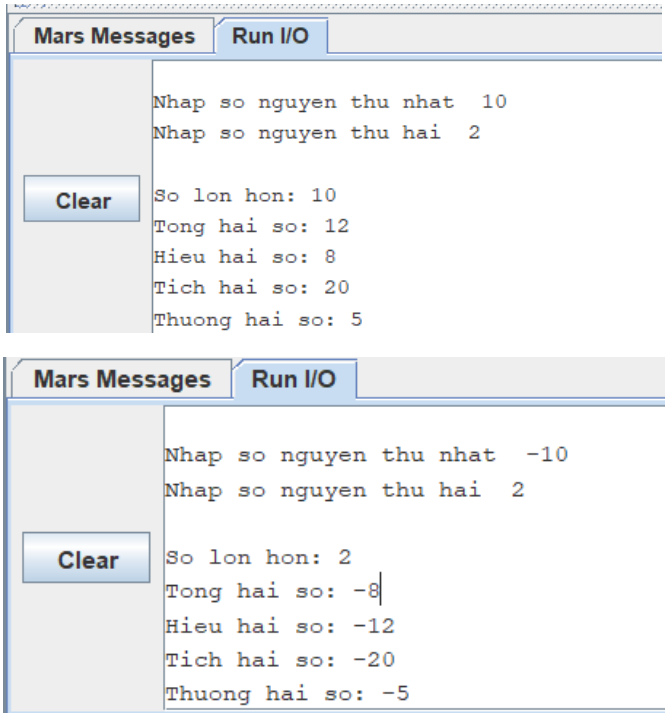
- Gọi syscall để thoát chương trình.

- Thực hiện viết chương trình

```
.data
  nhap1: .asciiz "Nhap so nguyen thu nhat  "
  nhap2: .asciiz "Nhap so nguyen thu hai  "
  ketqua1: .asciiz "\nSo lon hon: "
  ketqua2: .asciiz "\nTong hai so: "
  ketqua3: .asciiz "\nHieu hai so: "
  ketqua4: .asciiz "\nTich hai so: "
  ketqua5: .asciiz "\nThuong hai so: "
.text
 li $v0,4
 la $a0,nhap1
 syscall

 li $v0,5
 syscall
 move $t0,$v0

 li $v0,4
 la $a0,nhap2
 syscall

 li $v0,5
 syscall
 move $t1,$v0
 #so lon hon
 bge $t0,$t1,t0lonhon
 move $t2,$t1
 j insolonhon
```

```
t0lonhon:
  move $t2,$t0
  j insolonhon

insolonhon:
  li $v0,4
  la $a0,ketqua1
  syscall

  li $v0,1
  move $a0,$t2
  syscall

  #tong
  add $t3,$t0,$t1
  li $v0,4
  la $a0,ketqua2
  syscall
  li $v0,1
  move $a0,$t3
  syscall

  #hieu
  sub $t4,$t0,$t1
  li $v0,4
  la $a0,ketqua3
  syscall
  li $v0,1
  move $a0,$t4
  syscall

#tich
mul $t5,$t0,$t1
li $v0,4
la $a0,ketqua4
syscall
li $v0,1
move $a0,$t5
syscall

#thuong
beq $t1,$zero,ketthuc
div $t0,$t1
mflo $t6
li $v0,4
la $a0,ketqua5
syscall
li $v0,1
move $a0,$t6
syscall
j ketthuc
ketthuc:
li $v0,10
syscall
```

- Thực hiện chạy chương trình



III. Nhận xét:

- Thực hành làm quen với phần mềm mô phỏng MARS4_5

- Ôn lại những kiến thức về thanh ghi, memory

- Học được kỹ năng tra cứu thông tin, kĩ năng làm việc nhóm

IV. Tài liệu tham khảo:

Hennessy, John L., và David A. Patterson. *MIPS Architecture for Programmers Volume II-B: microMIPS64 Instruction Set*. Morgan Kaufmann, 2019.

Patterson, David A., và John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 5th ed.,