

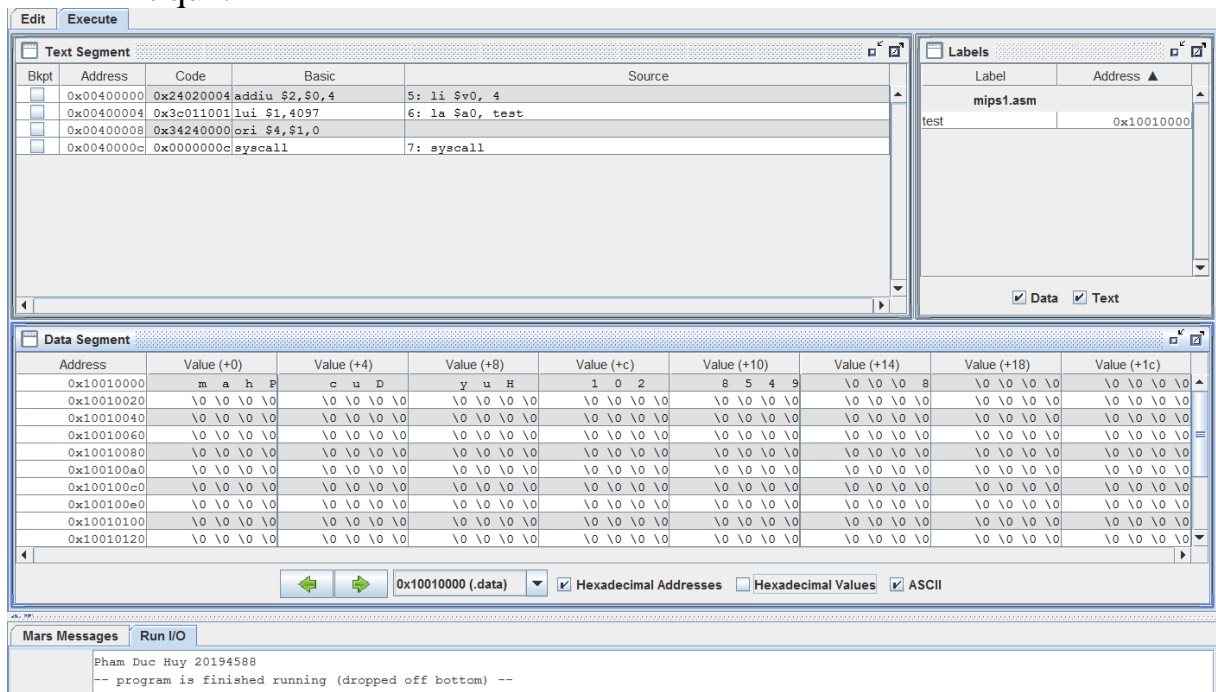
Báo cáo tuần 5

Assignment 1

- Mã nguồn:

```
mips1.asm
1  #Laboratory Exercise 5, Home Assignment 1
2  .data
3  test: .asciiz "Pham Duc Huy 20194588"
4  .text
5  li $v0, 4
6  la $a0, test
7  syscall
```

- Kết quả:



- Giải thích:
 - Dòng 3: Khai báo chuỗi ký tự “Pham Duc Huy 20194588” với nhãn test
 - Dòng 5: Gán $v0 = 4$
 - Dòng 6: Đưa địa chỉ biết test vào thanh ghi a0
 - Syscall \Rightarrow Gọi hàm $v0 \Rightarrow$ in ra chuỗi ký tự (do $v0$ có giá trị 4)

Assignment 2

- Mã nguồn:

```

mips1.asm mips2.asm
1  # print the sum of two register $s0 and $s1: "The sum of (s0) and (s1) is (result)"
2  .data
3      theSum:      .asciiz      "The sum of "
4      andWord:     .asciiz      " and "
5      is:          .asciiz      " is "
6  .text
7      addi    $s0, $zero, 2019
8      addi    $s1, $zero, 4588
9      add     $t0, $s0, $s1
10
11     li      $v0, 4              # print string
12     la      $a0, theSum         # print The sum of
13     syscall
14
15     li      $v0, 1              # print number
16     la      $a0, 0 ($s0)        # print s0
17     syscall
18
19     li      $v0, 4              # print string
20     la      $a0, andWord        # print and
21     syscall
22
23     li      $v0, 1              # print number
24     la      $a0, 0 ($s1)        # print s1
25     syscall
26
27     li      $v0, 4              # Print string
28     la      $a0, is             # print is
29     syscall
30
31     li      $v0, 1              # Print number
32     la      $a0, 0 ($t0)        # print t0 = s0 + s1
33     syscall
34

```

- Kết quả:

The screenshot shows the MARS MIPS simulator interface. The main window displays the assembly code with line numbers 1 through 34. The 'Registers' window on the right shows the state of registers: \$s0 is 2019, \$s1 is 4588, and \$t0 is 6607. The 'Data Segment' window shows memory addresses and values. The 'Mars Messages' window at the bottom displays the output: 'The sum of 2019 and 4588 is 6607' and '-- program is finished running (dropped off bottom) --'.

- Giải thích:
 - Dòng 3,4,5: Khai báo các chuỗi ký tự với nhãn tương ứng
 - Dòng 7,8: Khai báo s0,s1
 - Dòng 9: Tính tổng t0 = s0+s1
 - Dòng 11: Gán v0 = 4 (4 là lệnh in)
 - Dòng 12: load địa chỉ biến theSum vào thanh ghi a0

- Dòng 13: Gọi hàm v0 => In chuỗi ký tự
- Dòng 15: Gán thanh ghi v0 = 1 (1 là lệnh in số nguyên)
- Dòng 16: load địa chỉ s0 vào thanh ghi a0
- Dòng 17: Gọi hàm v0 => In 2019 ra màn hình
- Dòng 19 -25: Tương tự dòng 9-17 (thay theSum thành andWord)
⇒ In 4685 ra màn hình
- Dòng 27 -33: Tương tự dòng 9-17(thay theSum bằng s1)
⇒ In 6704 ra màn hình

Assignment 3:

- Màn hình chạy:

```

1  #Laboratory Exercise 5, Home Assignment 2
2  .data
3  x:      .space    1000
4  y:      .asciiz   "Pham Duc Huy"
5  .text
6  la      $a0, x
7  la      $a1, y
8  strcpy: add      $s0, $zero, $zero0
9  L1: add  $t1, $s0, $a1
10 lb      $t3, 0 ($t1)
11
12 add      $t0, $s0, $a0
13 sb      $t3, 0 ($t0)
14
15 beq      $t3, $zero, endOfStrcpy
16 nop
17 addi     $s0, $s0, 1
18 j        L1
19 endOfStrcpy: li    $v0, 4                # Print string
20 la      $a0, x                          # print the sum of
21 syscall

```

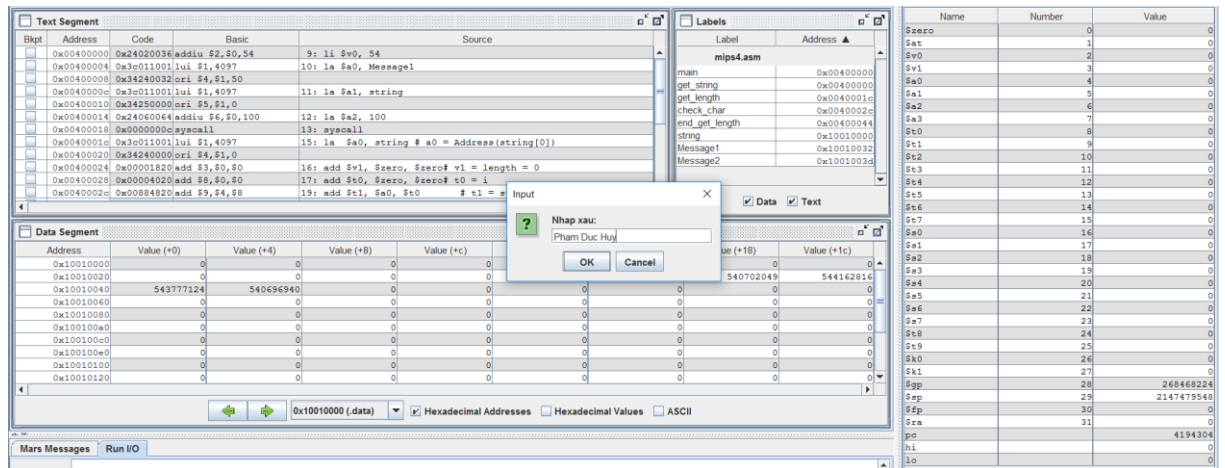
- Kết quả:

Assignment 4

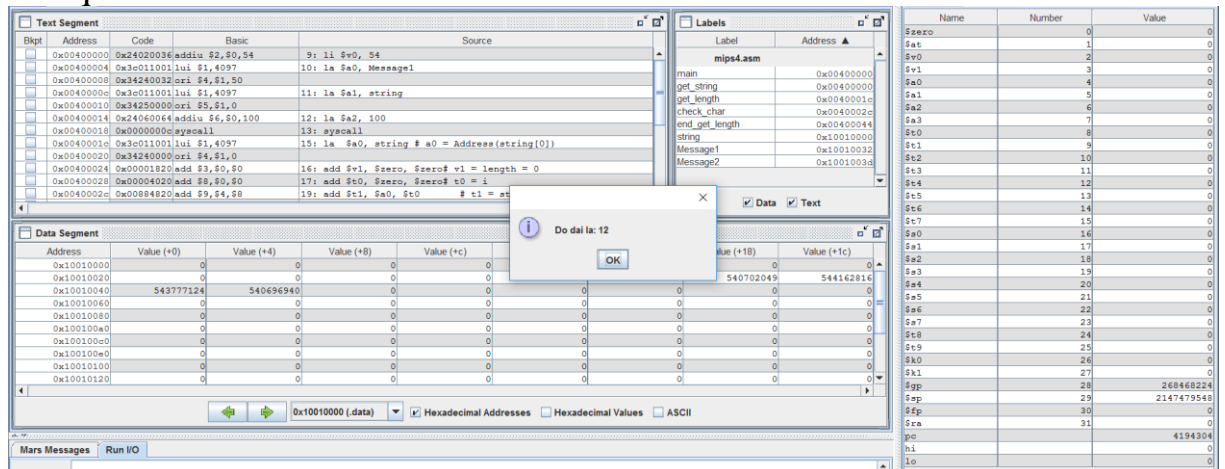
- Mã nguồn:

```
mips2.asm  mips3.asm  mips4.asm
1  #Laboratory Exercise 5, Home Assignment 3
2  .data
3  string:          .space   50
4  Message1:        .asciiz  "Nhap xau: "
5  Message2:        .asciiz  "Do dai la: "
6  .text
7  main:
8  get_string:
9  li $v0, 54
10 la $a0, Message1
11 la $a1, string
12 la $a2, 100
13 syscall
14 get_length:
15 la $a0, string # a0 = Address(string[0])
16 add $v1, $zero, $zero # v1 = length = 0
17 add $t0, $zero, $zero # t0 = i
18 check_char:
19 add $t1, $a0, $t0 # t1 = string[i]'s address
20 lb $t3, 0 ($t1) # t3 = string[i]
21
22 beq $t3, $zero, end_get_length
23
24 addi $v1, $v1, 1 # v1 = length ++
25 addi $t0, $t0, 1 # t0 = i ++
26 j      check_char
27 end_get_length:
28 li $v0, 56
29 la $a0, Message2 # Print the string
30 addi $v1, $v1, -1
31 la $a1, 0 ($v1) # Print the length of string
32 syscall
33
```

- Kết quả:
Nhập chuỗi ký tự = Phạm Duc Huy



Kết quả = 12:



• Giải thích:

- Dòng 3: Khai báo chuỗi có độ dài 50
- Dòng 4: Khai báo chuỗi ký tự “Nhập xau: ” với nhãn Message1
- Dòng 5: Khai báo chuỗi ký tự “Do dai la: ” với nhãn Message2
- Dòng 9: Gán thanh ghi v0 = 54 (54 là hiện hộp thoại nhập chuỗi)
- Dòng 10,11: load địa chỉ biến Message1,string vào thanh ghi a0,a1
- Dòng 12: khai báo a2 có thể đọc tối đa 100 ký tự
- Dòng 13: Gọi hàm v0
- Dòng 15: load địa chỉ biến string vào thanh ghi a0
- Dòng 16: khởi tạo v1 = 0 lưu độ dài chuỗi ký tự
- Dòng 17: khởi tạo biến chạy t0 = 0 (= i)
- Dòng 19: gán t1 bằng địa chỉ của string[i]
- Dòng 20: t3 = t1 (= string[i])
- Dòng 22: check if t3 = 0 (đọc hết string) => đến nhãn end_get_length
- Dòng 24: v1=v1+1
- Dòng 25: t0=t0+1
- Dòng 26: jump đến nhãn check_char
- Dòng 28: gán thanh ghi v0 = 56 (56 là lệnh hiển thị hộp thoại cho số nguyên)
- Dòng 29: load địa chỉ biến Message2 vào thanh ghi a0
- Dòng 30: v1 = v1- 1

- Dòng 31: load giá trị của v1 gán vào thanh ghi a1
- Dòng 32: Gọi hàm v0
⇒ in “Do dai la “ và số nguyên ra màn hình

Assignment 5

- Mã nguồn:

```

1
2 .data
3     string:      .space    50
4     message: .asciiz  "\nChuoai dao nguoc:\n"
5 .text
6     addi    $t1, $zero, 21
7     la      $s1, string
8 read_char:
9     li      $v0, 12
10    syscall
11
12    addi    $t1, $t1, -1
13    beq     $v0, 10, end_read_char
14    beqz    $t1, end_read_char
15
16    add     $t3, $t1, $s1
17    sb      $v0, 0 ($t3)
18
19    j       read_char
20 end_read_char:
21    li      $v0, 4                # read string $v0 luu charactor
22    la      $a0, message
23    syscall
24    li      $v0, 4
25    la      $a0, 0 ($t3)         # in ra chuoai dao nguoc
26    syscall

```

- Kết quả:
Khi chuỗi <20 ký tự

The screenshot shows the MARS MIPS simulator interface. The **Text Segment** window displays the assembly code with addresses and source code. The **Data Segment** window shows memory values. The **Registers** window shows the state of registers, with \$a0 containing the address 268500992. The **Run I/O** window shows the output: "Chuoi dao nguoc: yuH cuD mahP".

Khi vượt quá 20 ký tự => Tự động ngưng

Mars Messages

Run I/O

Clear

```
abcdthfihdsiilhsidhfií
Chuoi dao nguoc:
ifhdishiiisdhifhtdcba
-- program is finished running (dropped off bottom) --
```

- Giải thích:
 - Dòng 3: Khai báo chuỗi ký tự có độ dài 50
 - Dòng 4: Khai báo chuỗi ký tự “Chuoi dao nguoc:” với nhãn message
 - Dòng 6: khai báo t1 là chiều dài tối đa của string (ko vượt quá 20 ký tự)
 - Dòng 7: load địa chỉ biến string vào thanh ghi s1
 - Dòng 9: gán thanh ghi v0 = 12 (12 tức là lệnh đọc ký tự)
 - Dòng 10: thực hiện hàm v0
 - Dòng 12: t1 = t1 - 1
 - Dòng 13: if v0 == 10 => end_read_char
 - Dòng 14: if t1 == 0 (tức quá 20 ký tự) => end_read_char
 - Dòng 16: lsave địa chỉ ô ghi tiếp theo vào t3
 - Dòng 17: v0 = t3
 - Dòng 19: jump đến read_char
 - Dòng 21: gán thanh ghi v0 = 4 (4 là lệnh in chuỗi)
 - Dòng 22: load địa chỉ biến message vào thanh ghi a0
 - Dòng 23: Gọi hàm v0
 - ⇒ in chuỗi “Chuoi dao nguoc:” ra màn hình
 - Dòng 24: gán thanh ghi v0 = 4 (4 là lệnh in chuỗi)
 - Dòng 25: load địa chỉ thanh ghi t3 vào thanh ghi a0
 - Dòng 26: Gọi hàm v0
 - ⇒ in chuỗi ra màn hình