

Bài thực hành số 5

Lớp: 139365 – Học phần: Thực hành Kiến Trúc Máy Tính

Họ và tên: Đinh Thị Hồng Phúc

MSSV: 20215118

Bài 1.

```
1 #Assignment 1
2 .data
3     test: .asciiz "Hello World"
4 .text
5     li $v0, 4
6     la $a0, test
7     syscall
```

Thực hiện gõ chương trình vào công cụ MARS

- Thực hiện dòng lệnh 3: Khai báo chuỗi test: Hello World
- Thực hiện dòng lệnh số 5: Thanh ghi v0 được gán giá trị 4 (in ra chuỗi)
- Thực hiện dòng lệnh số 6: Dùng thanh ghi a0 lấy địa chỉ của chuỗi test
- Thực hiện dòng lệnh số 7: Gọi hàm in chuỗi ra màn hình (giá trị cần in lưu trong thanh ghi a0)
- Kết quả: Chương trình in ra chuỗi: Hello World

The screenshot displays the MARS MIPS simulator interface. The **Text Segment** window shows the assembly code being executed. The **Data Segment** window shows the memory layout, with the 'test' string stored at address 0x10010000. The **Register Window** shows the state of registers, with \$v0 containing 4 and \$a0 containing the address of the 'test' string. The **Console** window shows the output 'Hello World'.

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000004
\$v1	3	0x00000000
\$a0	4	0x10010000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$t9	25	0x00000000
\$t0	26	0x00000000
\$t1	27	0x00000000
\$t2	28	0x10000000
\$t3	29	0x7ffffcfc
\$t4	30	0x00000000
\$t5	31	0x00000000
\$t6	32	0x00000000
\$t7	33	0x00000000
\$t8	34	0x00000000
\$t9	35	0x00000000
\$s0	36	0x00000000
\$s1	37	0x00000000
\$s2	38	0x00000000
\$s3	39	0x00000000
\$s4	40	0x00000000
\$s5	41	0x00000000
\$s6	42	0x00000000
\$s7	43	0x00000000
\$s8	44	0x00000000
\$s9	45	0x00000000
\$t0	46	0x00000000
\$t1	47	0x00000000
\$t2	48	0x00000000
\$t3	49	0x00000000
\$t4	50	0x00000000
\$t5	51	0x00000000
\$t6	52	0x00000000
\$t7	53	0x00000000
\$t8	54	0x00000000
\$t9	55	0x00000000
\$s0	56	0x00000000
\$s1	57	0x00000000
\$s2	58	0x00000000
\$s3	59	0x00000000
\$s4	60	0x00000000
\$s5	61	0x00000000
\$s6	62	0x00000000
\$s7	63	0x00000000
\$s8	64	0x00000000
\$s9	65	0x00000000
\$t0	66	0x00000000
\$t1	67	0x00000000
\$t2	68	0x00000000
\$t3	69	0x00000000
\$t4	70	0x00000000
\$t5	71	0x00000000
\$t6	72	0x00000000
\$t7	73	0x00000000
\$t8	74	0x00000000
\$t9	75	0x00000000
\$s0	76	0x00000000
\$s1	77	0x00000000
\$s2	78	0x00000000
\$s3	79	0x00000000
\$s4	80	0x00000000
\$s5	81	0x00000000
\$s6	82	0x00000000
\$s7	83	0x00000000
\$s8	84	0x00000000
\$s9	85	0x00000000
\$t0	86	0x00000000
\$t1	87	0x00000000
\$t2	88	0x00000000
\$t3	89	0x00000000
\$t4	90	0x00000000
\$t5	91	0x00000000
\$t6	92	0x00000000
\$t7	93	0x00000000
\$t8	94	0x00000000
\$t9	95	0x00000000
\$s0	96	0x00000000
\$s1	97	0x00000000
\$s2	98	0x00000000
\$s3	99	0x00000000
\$s4	100	0x00000000
\$s5	101	0x00000000
\$s6	102	0x00000000
\$s7	103	0x00000000
\$s8	104	0x00000000
\$s9	105	0x00000000
\$t0	106	0x00000000
\$t1	107	0x00000000
\$t2	108	0x00000000
\$t3	109	0x00000000
\$t4	110	0x00000000
\$t5	111	0x00000000
\$t6	112	0x00000000
\$t7	113	0x00000000
\$t8	114	0x00000000
\$t9	115	0x00000000
\$s0	116	0x00000000
\$s1	117	0x00000000
\$s2	118	0x00000000
\$s3	119	0x00000000
\$s4	120	0x00000000
\$s5	121	0x00000000
\$s6	122	0x00000000
\$s7	123	0x00000000
\$s8	124	0x00000000
\$s9	125	0x00000000
\$t0	126	0x00000000
\$t1	127	0x00000000
\$t2	128	0x00000000
\$t3	129	0x00000000
\$t4	130	0x00000000
\$t5	131	0x00000000
\$t6	132	0x00000000
\$t7	133	0x00000000
\$t8	134	0x00000000
\$t9	135	0x00000000
\$s0	136	0x00000000
\$s1	137	0x00000000
\$s2	138	0x00000000
\$s3	139	0x00000000
\$s4	140	0x00000000
\$s5	141	0x00000000
\$s6	142	0x00000000
\$s7	143	0x00000000
\$s8	144	0x00000000
\$s9	145	0x00000000
\$t0	146	0x00000000
\$t1	147	0x00000000
\$t2	148	0x00000000
\$t3	149	0x00000000
\$t4	150	0x00000000
\$t5	151	0x00000000
\$t6	152	0x00000000
\$t7	153	0x00000000
\$t8	154	0x00000000
\$t9	155	0x00000000
\$s0	156	0x00000000
\$s1	157	0x00000000
\$s2	158	0x00000000
\$s3	159	0x00000000
\$s4	160	0x00000000
\$s5	161	0x00000000
\$s6	162	0x00000000
\$s7	163	0x00000000
\$s8	164	0x00000000
\$s9	165	0x00000000
\$t0	166	0x00000000
\$t1	167	0x00000000
\$t2	168	0x00000000
\$t3	169	0x00000000
\$t4	170	0x00000000
\$t5	171	0x00000000
\$t6	172	0x00000000
\$t7	173	0x00000000
\$t8	174	0x00000000
\$t9	175	0x00000000
\$s0	176	0x00000000
\$s1	177	0x00000000
\$s2	178	0x00000000
\$s3	179	0x00000000
\$s4	180	0x00000000
\$s5	181	0x00000000
\$s6	182	0x00000000
\$s7	183	0x00000000
\$s8	184	0x00000000
\$s9	185	0x00000000
\$t0	186	0x00000000
\$t1	187	0x00000000
\$t2	188	0x00000000
\$t3	189	0x00000000
\$t4	190	0x00000000
\$t5	191	0x00000000
\$t6	192	0x00000000
\$t7	193	0x00000000
\$t8	194	0x00000000
\$t9	195	0x00000000
\$s0	196	0x00000000
\$s1	197	0x00000000
\$s2	198	0x00000000
\$s3	199	0x00000000
\$s4	200	0x00000000
\$s5	201	0x00000000
\$s6	202	0x00000000
\$s7	203	0x00000000
\$s8	204	0x00000000
\$s9	205	0x00000000
\$t0	206	0x00000000
\$t1	207	0x00000000
\$t2	208	0x00000000
\$t3	209	0x00000000
\$t4	210	0x00000000
\$t5	211	0x00000000
\$t6	212	0x00000000
\$t7	213	0x00000000
\$t8	214	0x00000000
\$t9	215	0x00000000
\$s0	216	0x00000000
\$s1	217	0x00000000
\$s2	218	0x00000000
\$s3	219	0x00000000
\$s4	220	0x00000000
\$s5	221	0x00000000
\$s6	222	0x00000000
\$s7	223	0x00000000
\$s8	224	0x00000000
\$s9	225	0x00000000
\$t0	226	0x00000000
\$t1	227	0x00000000
\$t2	228	0x00000000
\$t3	229	0x00000000
\$t4	230	0x00000000
\$t5	231	0x00000000
\$t6	232	0x00000000
\$t7	233	0x00000000
\$t8	234	0x00000000
\$t9	235	0x00000000
\$s0	236	0x00000000
\$s1	237	0x00000000
\$s2	238	0x00000000
\$s3	239	0x00000000
\$s4	240	0x00000000
\$s5	241	0x00000000
\$s6	242	0x00000000
\$s7	243	0x00000000
\$s8	244	0x00000000
\$s9	245	0x00000000
\$t0	246	0x00000000
\$t1	247	0x00000000
\$t2	248	0x00000000
\$t3	249	0x00000000
\$t4	250	0x00000000
\$t5	251	0x00000000
\$t6	252	0x00000000
\$t7	253	0x00000000
\$t8	254	0x00000000
\$t9	255	0x00000000

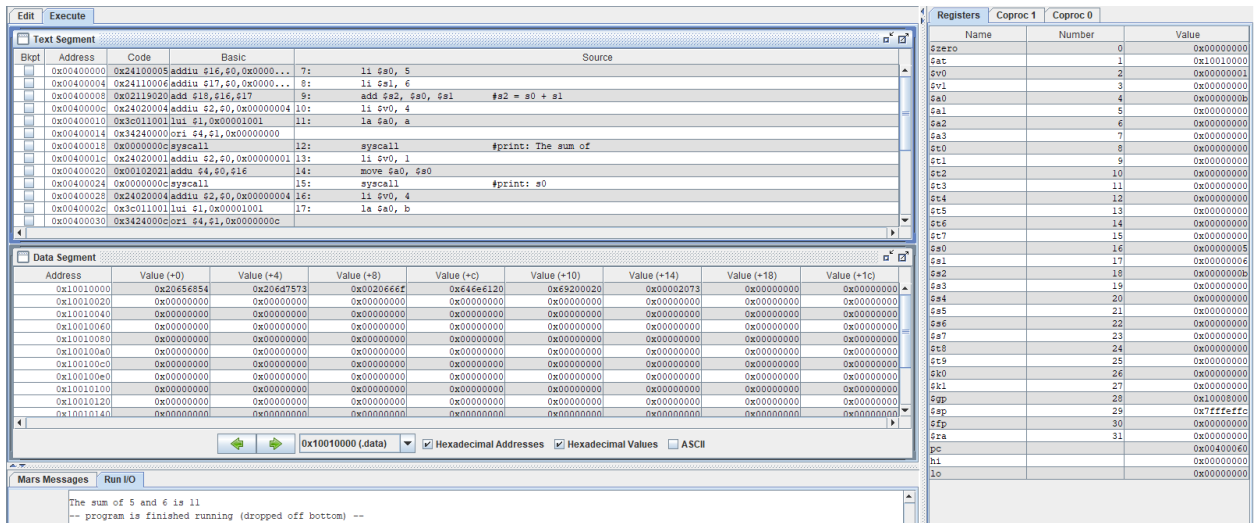
Thực hiện chạy chương trình với MARS

Bài 2.

```
1 #Assignment 2
2 .data
3     a: .asciiz "The sum of "
4     b: .asciiz " and "
5     c: .asciiz " is "
6 .text
7     li $s0, 5
8     li $s1, 6
9     add $s2, $s0, $s1      #s2 = s0 + s1
10    li $v0, 4
11    la $a0, a
12    syscall                #print: The sum of
13    li $v0, 1
14    move $a0, $s0
15    syscall                #print: s0
16    li $v0, 4
17    la $a0, b
18    syscall                #print: and
19    li $v0, 1
20    move $a0, $s1
21    syscall                #print: s1
22    li $v0, 4
23    la $a0, c
24    syscall                #print: is
25    li $v0, 1
26    move $a0, $s2
27    syscall                #print: s2
```

Thực hiện gõ chương trình vào công cụ **MARS**

- Dòng 3, 4, 5: Khai báo 3 chuỗi a, b, c lần lượt là “The sum of ”, “ and ”, “is ”
- Dòng 7, 8: Thanh ghi s0, s1 được gán giá trị lần lượt là 5, 6
- Dòng 9: Tính tổng s0 và s1, gán giá trị vào s2
- Dòng 10, 16, 22: Thanh ghi v0 được gán giá trị 4 (in ra chuỗi)
- Dòng 11: Dùng thanh ghi a0 lấy địa chỉ của chuỗi a
- Dòng 12, 15, 18, 21, 24, 27: Gọi hàm in chuỗi ra màn hình (giá trị cần in lưu trong thanh ghi a0)
- Dòng 13, 19, 25: Thanh ghi v0 được gán giá trị 1 (in ra số)
- Dòng 14: Di chuyển giá trị từ s0 vào a0
- Dòng 17: Dùng thanh ghi a0 lấy địa chỉ của chuỗi b
- Dòng 20: Di chuyển giá trị từ s1 vào a0
- Dòng 23: Dùng thanh ghi a0 lấy địa chỉ của chuỗi c
- Dòng 26: Di chuyển giá trị từ s2 vào a0



Thực hiện chạy chương trình với MARS

Bài 3.

```

1  #Assignment 3
2  .data
3      x: .space 1000      # destination string x, empty
4      y: .asciiz "Hello"  # source string y
5  .text
6  strcpy:
7      add $s0, $zero, $zero  #s0 = i=0
8  L1:
9      la $a1, y
10     add $t1, $s0, $a1      #t1 = s0 + a1 = i + y[0]
11                                # = address of y[i]
12     lb $t2, 0($t1)         #t2 = value at t1 = y[i]
13     la $a0, x
14     add $t3, $s0, $a0      #t3 = s0 + a0 = i + x[0]
15                                # = address of x[i]
16     sb $t2, 0($t3)         #x[i]= t2 = y[i]
17     beq $t2, $zero, end_of_strcpy  #if y[i]==0, exit
18     nop
19     addi $s0, $s0, 1       #s0=s0 + 1 <-> i=i+1
20     j L1                  #next character
21     nop
22 end_of_strcpy:
23
24     li $v0, 4
25     la $a0, x
26     syscall

```

Thực hiện gõ chương trình vào công cụ MARS

- Dòng 3: Khai báo chuỗi x rỗng có 1000 phần tử
- Dòng 4: Khai báo chuỗi y: Hello
- Dòng 7: Khai báo biến đếm i = 0
- Dòng 9: Dùng thanh ghi a1 lấy địa chỉ của phần tử đầu trong chuỗi y (y[0])
- Dòng 10: Thanh ghi t1 chứa địa chỉ của y[i]
- Dòng 12: Thanh ghi t2 chứa giá trị của phần tử thanh ghi t1 trỏ đến (y[i])

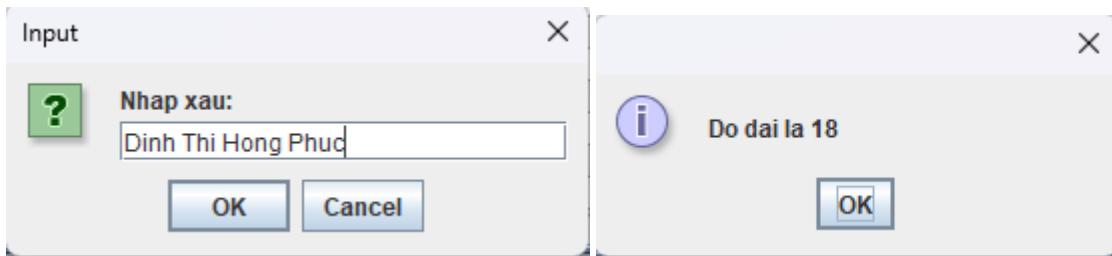
Bài 4.

```
1 #Assignment 4
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    xor $v0, $zero, $zero  # v0 = length = 0
17    xor $t0, $zero, $zero  # t0 = i = 0
18 check_char:
19    add $t1, $a0, $t0      # t1 = a0 + t0
20                                # = Address(string[0]+i)
21    lb $t2, 0($t1)         # t2 = string[i]
22    beq $t2, $zero, end_of_str # Is null char?
23    addi $v0, $v0, 1       # v0=v0+1->length=length+1
24    addi $t0, $t0, 1       # t0=t0+1->i = i + 1
25    j check_char
26 end_of_str:
27 end_of_get_length:
28 print_length:
29    sub $s1, $v0, 1
30    li $v0, 56
31    la $a0, Message2
32    move $a1, $s1
33    syscall
```

Thực hiện gõ chương trình vào công cụ **MARS**

- Dòng 3: Khai báo chuỗi string rỗng, 50 phần tử
- Dòng 4, 5: Khai báo chuỗi Message1, Message2 lần lượt là “Nhap xau:”, “Do dai la ”
- Dòng 9: Thanh ghi v0 được gán giá trị 54 (nhập chuỗi vào Dialog)
- Dòng 10: Dùng thanh ghi a0 lấy địa chỉ chuỗi Message1
- Dòng 11: Dùng thanh ghi a1 lấy địa chỉ chuỗi string
- Dòng 12: Số lượng tối đa của chuỗi
- Dòng 13: Nhập chuỗi string từ bàn phím vào Dialog
- Dòng 15: Dùng thanh ghi a0 lấy địa chỉ phần tử đầu của chuỗi string

- Dòng 16, 17: Gán chiều dài, biến đếm i bằng 0
- Dòng 19: Thanh ghi t0 chứa địa chỉ string[i]
- Dòng 21: Thanh ghi t2 chứa giá trị string[i]
- Dòng 22: Kiểm tra đã đến kí tự cuối hay chưa, nếu là kí tự cuối thì kết thúc vòng lặp
- Dòng 23, 24, 25: Nếu chưa đến kí tự cuối thì tăng chiều dài, biến đếm thêm 1, tiếp tục vòng lặp
- Dòng 29: Trừ chiều dài xâu đi 1, vì đã tính cả kí tự ENTER cuối cùng
- Dòng 30: Thanh ghi v0 được gán giá trị 56 (in chuỗi, số vào Dialog)
- Dòng 31: Dùng thanh ghi a0 lấy địa chỉ chuỗi Message2
- Dòng 32: Di chuyển giá trị từ s1 vào a1
- Dòng 33: In chuỗi ra Dialog



Thực hiện chạy chương trình với **MARS**

Bài 5.

```
1  #Assignment 5
2  .data
3      string: .space 100
4  .text
5  main:
6  get_string:
7      li $v0, 8
8      la $a0, string
9      la $a1, 21
10     syscall
11 get_length:
12     li $v0, 0      # v0 = length = 0
13     li $t0, 0      # t0 = i = 0
14 check_char:
15     add $t1, $a0, $t0      # t1 = a0 + t0
16                             #      = Address(string[0]+i)
17     lb $t2, 0($t1)         # t2 = string[i]
18     beq $t2, $zero, end_of_str      # Is null char?
19     addi $v0, $v0, 1      # v0=v0+1->length=length+1
20     addi $t0, $t0, 1      # t0=t0+1->i = i + 1
21     j check_char
22
23 end_of_str:
24     addi $t0, $v0, -1      #t0 = length (ignore '\n')
25     move $s0, $a0          #s0 = address pf str
26 print_reverse:
27     li $v0, 11
28     add $t1, $s0, $t0
29     lb $a0, 0($t1)
30     syscall
31     addi $t0, $t0, -1
32     blt $t0, $zero, end
33     j print_reverse
34 end:
```

Thực hiện gõ chương trình vào công cụ **MARS**

Giải thích: Dựa trên assignment 4 để nhập xâu (dòng 7-10), đếm chiều dài chuỗi (trừ 1 do đã đếm cả kí tự ENTER) (dòng 12-24). Sau đó in chuỗi đảo ngược bằng cách di chuyển bắt đầu từ phần tử cuối cùng của chuỗi ($t1 = s0 + t0$), in lần lượt các kí tự lấy được ra màn hình (dòng 28). Kiểm tra nếu $t0 < 0$ thì kết thúc vòng lặp, không thì lặp đến hết chuỗi

<input type="button" value="Clear"/>	Dinh Thi Hong Phuc
	cuhP gnoH ihT hniD

Thực hiện chạy chương trình với **MARS**