

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO

Bài tập thực hành tuần 7

Học phần: Thực hành kiến trúc máy tính

Giảng viên hướng dẫn: Lê Bá Vui

Sinh viên thực hiện: Phạm Huy Cảnh - 20194490

Mã lớp: 130938

Hà Nội, tháng 5 năm 2022

1. Assignment 1: Tính giá trị tuyệt đối

Trước khi chạy lệnh: `jal abs`

D:\Courses\2021\2\ThucHanhKienTrucMayTinh\Assignments\Week7\HomeAssignment1.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x2404fffd3	addiu \$4,\$0,0xffff...	3: main: li \$a0, -45	#load input parameter
0x00400004	0x0c100006	jal 0x00400018	4: jal abs	#jump and link to abs pr...
0x00400008	0x00000000	nop	5: nop	
0x0040000c	0x00028020	add \$16,\$0,\$2	6: add \$a0, \$zero, \$v0	
0x00400010	0x2402000a	addiu \$2,\$0,0x0000...	7: li \$v0, 10	#terminate
0x00400014	0x00000005	syscall	8: syscall	
0x00400018	0x00041022	sub \$2,\$0,\$4	15: abs: sub \$v0,\$zero,\$a0	#put -(a0) in v0; in cas...
0x0040001c	0x04800002	bltz \$a0,done	16: bltz \$a0,done	#if (a0)<0 then done
0x00400020	0x00000000	nop	17: nop	
0x00400024	0x00801020	add \$2,\$4,\$0	18: add \$v0,\$a0,\$zero	#else put (a0) in v0
0x00400028	0x03e00008	jr \$ra	19: done: jr \$ra	

Labels

Label	Address
HomeAssignment1.asm	
main	0x00400000
endmain	0x00400018
abs	0x00400018
done	0x00400028

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Clear

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0xffffffff
\$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
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x00000000
\$fp	29	0x7ffffeffc
\$tp	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00400004
\$hi		0x00000000
\$lo		0x00000000

Sau khi chạy lệnh: `jal abs`

D:\Courses\2021\2\ThucHanhKienTrucMayTinh\Assignments\Week7\HomeAssignment1.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x2404fffd3	addiu \$4,\$0,0xffff...	3: main: li \$a0, -45	#load input parameter
0x00400004	0x0c100006	jal 0x00400018	4: jal abs	#jump and link to abs pr...
0x00400008	0x00000000	nop	5: nop	
0x0040000c	0x00028020	add \$16,\$0,\$2	6: add \$a0, \$zero, \$v0	
0x00400010	0x2402000a	addiu \$2,\$0,0x0000...	7: li \$v0, 10	#terminate
0x00400014	0x00000005	syscall	8: syscall	
0x00400018	0x00041022	sub \$2,\$0,\$4	15: abs: sub \$v0,\$zero,\$a0	#put -(a0) in v0; in cas...
0x0040001c	0x04800002	bltz \$a0,done	16: bltz \$a0,done	#if (a0)<0 then done
0x00400020	0x00000000	nop	17: nop	
0x00400024	0x00801020	add \$2,\$4,\$0	18: add \$v0,\$a0,\$zero	#else put (a0) in v0
0x00400028	0x03e00008	jr \$ra	19: done: jr \$ra	

Labels

Label	Address
HomeAssignment1.asm	
main	0x00400000
endmain	0x00400018
abs	0x00400018
done	0x00400028

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Clear

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0xffffffff
\$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
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x00000000
\$fp	29	0x7ffffeffc
\$tp	30	0x00000000
\$ra	31	0x00400008
\$pc		0x00400018
\$hi		0x00000000
\$lo		0x00000000

Kết quả chạy chương trình:

The screenshot shows the MARS 4.5 MIPS simulator interface. The main window displays the assembly code for 'HomeAssignment1.asm'. The Text Segment window shows the following code:

```

0x00400000: addiu $a0, -45      3: main: li $a0, -45      #load input parameter
0x00400004: jal abs             4:      jal abs          #jump and link to abs pr...
0x00400008: pop                5:      pop
0x0040000c: add $a0, $zero, $v0 6:      add $a0, $zero, $v0
0x00400010: li $v0, 10          7:      li $v0, 10       #terminate
0x00400014: syscall            8:      syscall
0x00400018: sub $2, $0, $a0     15: abs: sub $v0, $zero, $a0 #put -(a0) in v0; in cas...
0x0040001c: bltz $a0, done      16:      bltz $a0, done    #if (a0)<0 then done
0x00400020: pop                17:      pop
0x00400024: add $v0, $a0, $zero 18:      add $v0, $a0, $zero #else put (a0) in v0
0x00400028: jr $ra             19: done: jr $ra
  
```

The Data Segment window shows memory addresses and their corresponding values. The Registers window on the right shows the state of MIPS registers. The MARS Messages window at the bottom shows the message '-- program is finished running --'.

Nhận xét: Khi chạy lệnh `jal abs` (địa chỉ lệnh `0x00400004`) thì thanh ghi `$ra` được gán bằng địa chỉ của câu lệnh tiếp theo (`0x00400008`) và thanh ghi `pc` được gán bằng địa chỉ `0x00400018` (địa chỉ tại nhãn `abs`)

Kết quả cuối cùng ta lấy được giá trị tuyệt đối của số được nạp vào trong thanh ghi `$a0` và ghi kết quả đó vào thanh ghi `$s0`

\$a0	4	-45
\$s0	16	45

2. Assignment 2: Tìm MAX của 3 số nguyên

Kết quả chạy chương trình mẫu:

The screenshot shows the MARS MIPS simulator interface. The main window displays the assembly code for a program that finds the maximum of three numbers stored in registers \$a0, \$a1, and \$a2. The registers window on the right shows the current values: \$a0 = 9, \$a1 = 10, and \$a2 = 6. The console at the bottom shows the output 'MAX = 10'.

Thanh ghi $\$s0 = 9$ đang lưu giá trị MAX

Sau khi sửa giá trị trong thanh ghi $\$a0$, $\$a1$, $\$a2$:

The screenshot shows the MARS MIPS simulator interface with the same assembly code as the previous image. However, the registers window shows modified input values: \$a0 = 10, \$a1 = 7, and \$a2 = -3. The console at the bottom shows the output 'MAX = 10'.

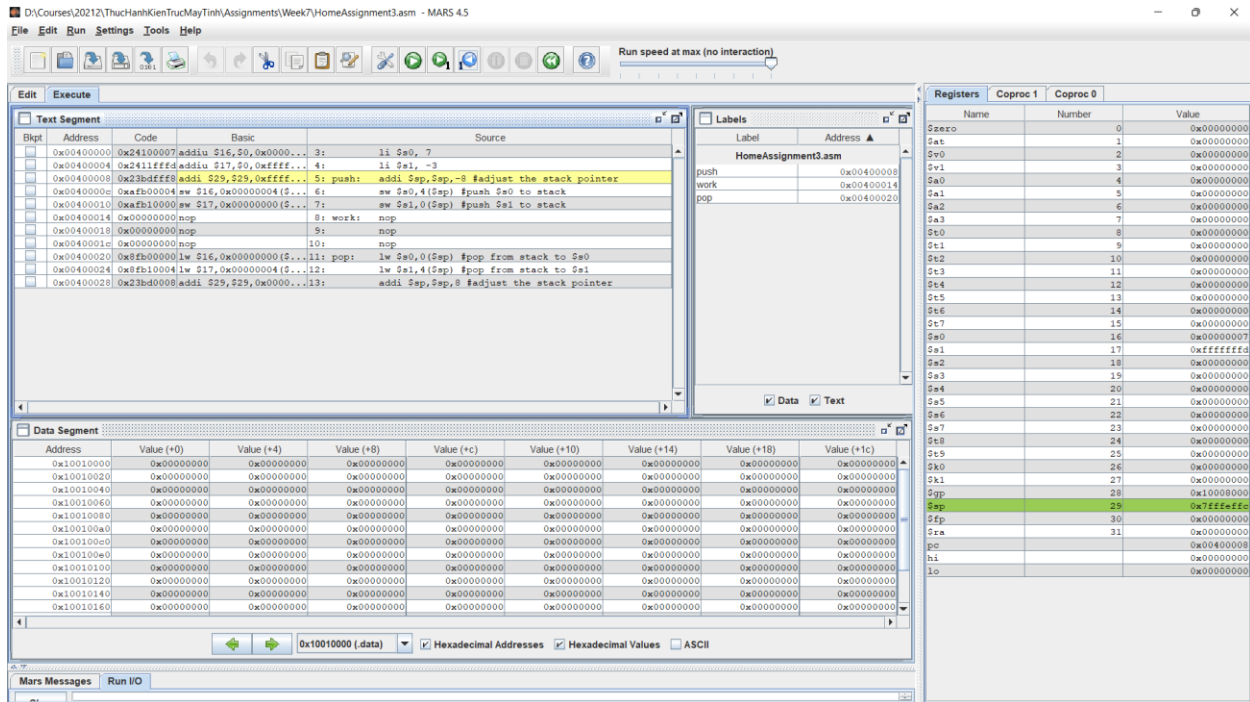
$\$a0 = -12, \$a1 = 7, \$a2 = -3$

Kết quả thu được $\$s0 = 7$

Nhận xét: Khi chạy lệnh `jal` thì thanh ghi $\$ra$ được gán bằng giá trị của địa chỉ của câu lệnh tiếp theo sau `jal` trong nhãn `main`. Thanh ghi `pc` được gán bằng địa chỉ của nhãn `max` để câu lệnh tiếp tục được thực hiện bắt đầu từ nhãn `max`. Sau khi chạy đến `jr` $\$ra$ thì `pc` được gán bằng địa chỉ trong $\$ra$ (địa chỉ của `nop`)

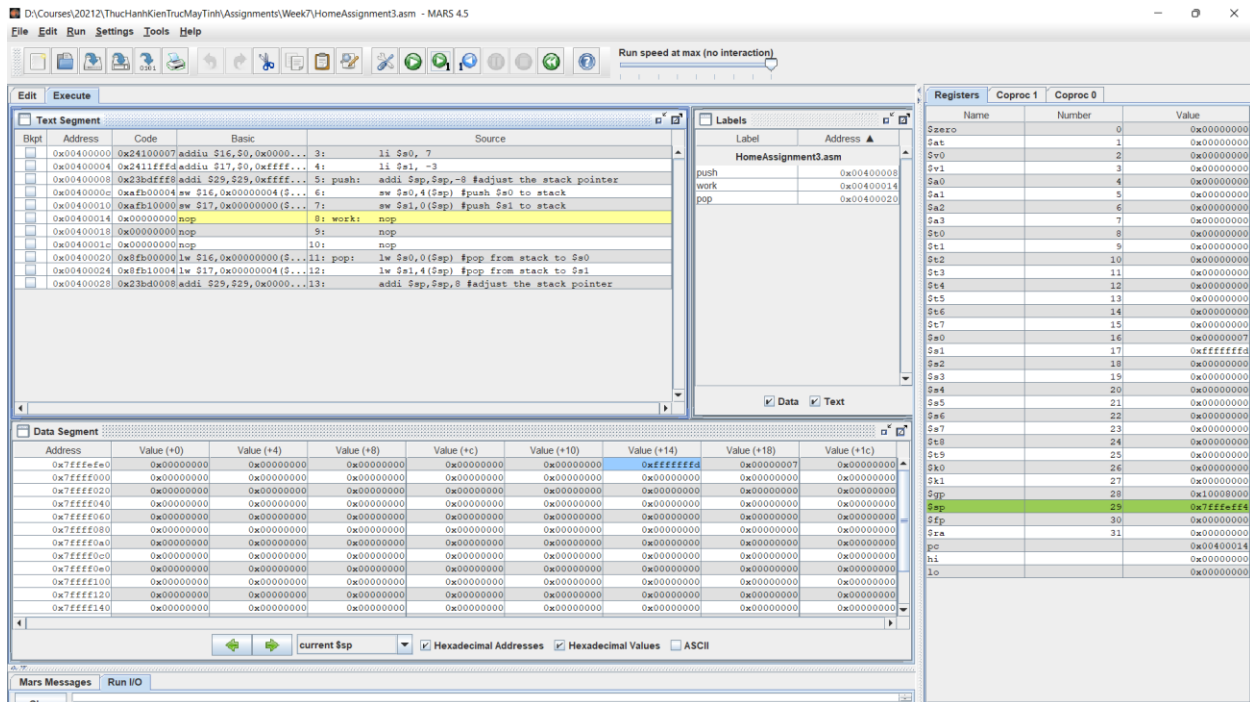
3. Assignment 3: Đổi chỗ hai số nguyên

Trước khi chạy lệnh `add` ở nhãn `push`:



`$sp = 7ffffeffc`

Sau khi chạy lệnh `add` ở nhãn `push`:

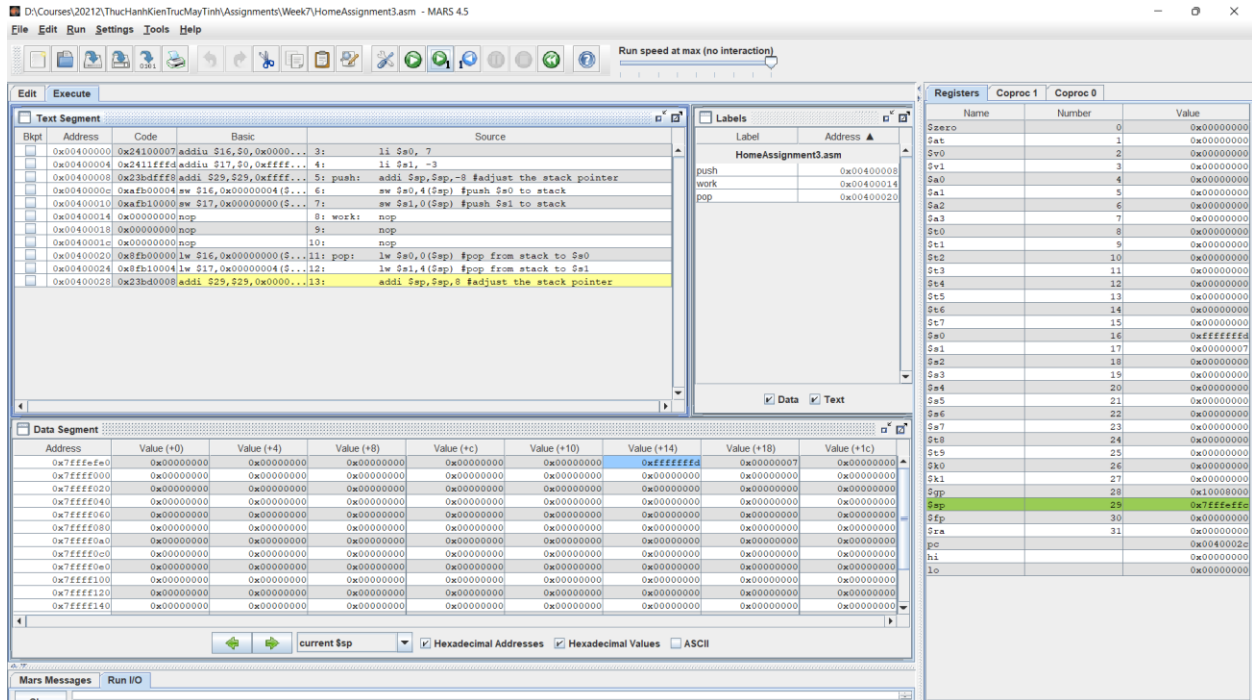


$\$sp = 0x7ffffeff4$

Thanh ghi $\$sp$ được giảm đi 8 byte (tức cấp phát cho bộ nhớ stack 8 byte)

Sau đó lần lượt ghi giá trị trong $\$s0$ vào $\$sp + 4$, giá trị trong $\$s1$ vào $\$sp + 0$

Sau khi chạy lệnh `add` ở nhãn `pop`:



Thực hiện đổi chỗ hai số bằng cách load giá trị tại địa chỉ $\$sp + 0$ vào $\$s1$, load giá trị tại địa chỉ $\$sp + 4$ vào $\$s0$

Lệnh `add $sp, $sp, 8` (giúp giải phóng stack, trả lại đỉnh stack)

4. Assignment 4: Tính giai thừa

D:\Courses\20212\ThucHanhKienTrucMayTinh\Assignments\Week7\HomeAssignment4.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

Labels

HomeAssignment4.asm

Registers

Coproc 1 Coproc 0

Value

\$zero 0 0x00000000

\$at 1 0x00000000

\$v0 2 0x00000000

\$a0 4 0x00000000

\$a1 5 0x00000000

\$a2 6 0x00000000

\$a3 7 0x00000000

\$t1 9 0x00000000

\$t2 10 0x00000000

\$t3 11 0x00000000

\$t4 12 0x00000000

\$t5 13 0x00000000

\$t6 14 0x00000000

\$t7 15 0x00000000

\$w0 16 0x00000000

\$w1 17 0x00000000

\$w2 18 0x00000000

\$w3 19 0x00000000

\$w4 20 0x00000000

\$w5 21 0x00000000

\$w6 22 0x00000000

\$w7 23 0x00000000

\$t8 24 0x00000000

\$t9 25 0x00000000

\$d0 26 0x00000000

\$k1 27 0x00000000

\$gp 28 0x10000000

\$sp 29 0x7ffffefc

\$fp 30 0x00000000

\$ra 31 0x00000000

\$lo 0x00400000

\$hi 0x00000000

\$lo 0x00000000

Text Segment

Address Code Basic Source

0x00400000 0x00100008 jal 0x00400020 5: main: jal WARP

0x00400004 0x00402120 add \$5,\$2,\$0 7: print: add \$a1,\$v0,\$zero # \$a0 = result from N!

0x00400008 0x24020038 addiu \$2,\$0,0x00000000 8: li \$v0,\$5

0x0040000c 0x3c011001 lui \$1,0x00001001 9: la \$a0,\$message

0x00400010 0x34240000 ori \$4,\$1,0x00000000

0x00400014 0x0000000c syscall 10: syscall

0x00400018 0x2402000a addiu \$2,\$0,0x00000000 11: quit: li \$v0,10 # terminate

0x0040001c 0x0000000c syscall 12: syscall

0x00400020 0x2402000a addiu \$2,\$0,0x00000000 17: WARP: sw \$fp,-4(\$sp) # save frame pointer (1)

0x00400024 0x23bc0000 addi \$29,\$29,0x00000000 18: addi \$fp,\$sp,0 # new frame pointer point to the t...

0x00400028 0x23bc0000 addi \$29,\$29,0x00000000 19: addi \$fp,\$sp,-8 # adjust stack pointer (3)

0x0040002c 0x2402000a addiu \$2,\$0,0x00000000 20: sw \$ra,0(\$sp) # save return address (4)

0x00400030 0x2402000a addiu \$4,\$0,0x00000000 21: li \$a0,6 # load test input

0x00400034 0x00100113 jal 0x0040004c 22: jal fact # call fact proc

0x00400038 0x00000000 nop 23: nop

0x0040003c 0x2402000a addiu \$2,\$0,0x00000000 25: li \$ra,0(\$sp) # restore re

0x00400040 0x23bc0000 addi \$29,\$29,0x00000000 26: addi \$fp,\$sp,0 # return ad

0x00400044 0x2402000a addiu \$2,\$0,0x00000000 27: li \$fp,-4(\$sp) # return fr

0x00400048 0x00100113 jal 0x0040004c 28: jal fact # call fact proc

0x0040004c 0x2402000a addiu \$2,\$0,0x00000000 35: WTB: sw \$fp,-4(\$sp) # save fra

Labels

Label Address

main 0x00400000

print 0x00400004

quit 0x00400018

endmain 0x00400020

WARP 0x00400020

wrap_end 0x0040004c

FACT 0x0040004c

recursive 0x00400078

done 0x00400090

fact_end 0x004000a4

Message 0x10010000

Registers

Coproc 1 Coproc 0

Value

\$zero 0 0x00000000

\$at 1 0x00000000

\$v0 2 0x00000000

\$a0 4 0x00000000

\$a1 5 0x00000000

\$a2 6 0x00000000

\$a3 7 0x00000000

\$t1 9 0x00000000

\$t2 10 0x00000000

\$t3 11 0x00000000

\$t4 12 0x00000000

\$t5 13 0x00000000

\$t6 14 0x00000000

\$t7 15 0x00000000

\$w0 16 0x00000000

\$w1 17 0x00000000

\$w2 18 0x00000000

\$w3 19 0x00000000

\$w4 20 0x00000000

\$w5 21 0x00000000

\$w6 22 0x00000000

\$w7 23 0x00000000

\$t8 24 0x00000000

\$t9 25 0x00000000

\$d0 26 0x00000000

\$k1 27 0x00000000

\$gp 28 0x10000000

\$sp 29 0x7ffffefc

\$fp 30 0x00000000

\$ra 31 0x00000000

\$lo 0x00400000

\$hi 0x00000000

\$lo 0x00000000

Data Segment

Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)

0x10010000 0x2074654b 0x20617571 0x656e6974 0x61656720 0x68742069 0x6c206175 0x00203a61 0x00000000

0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x100100e0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010100 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010120 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010140 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x10010160 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

Mars Messages Run I/O

Kết quả khi chạy chương trình: $6! = 720$ (Đúng)

Nhận xét: Sự thay đổi của thanh ghi $\$sp$

Data Segment

Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)

0x7ffffea0 0x00000000 0x00000000 0x00000000 0x00000001 0x00400080 0x7ffffec4 0x00000002 0x00400080

0x7ffffec0 0x00000000 0x00000003 0x00400080 0x7ffffedc 0x00000004 0x00400080 0x7ffffee8 0x00000005

0x7ffffee0 0x00400080 0x7ffffef4 0x00000006 0x00400038 0x7ffffefc 0x00400004 0x00000000 0x00000000

0x7ffffe00 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffe20 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffe40 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffe60 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffe80 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffea0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffec0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7ffffee0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff00 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff20 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff40 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff60 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff80 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffffa0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffffc0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffffe0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

0x7fffff00 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

current \$sp

Hexadecimal Addresses Hexadecimal Values ASCII

Chạy trường hợp n = 3 (Tính 3!)

Bảng thể hiện giá trị ngăn xếp:

0x7ffeff8	\$fp = 0x00000000
0x7ffeff4	\$ra = 0x00400004
0x7ffeff0	\$fp = 0x7ffeffc
0x7ffefec	\$ra = 0x00400038
0x7ffefe8	\$a0 = 0x00000003
0x7ffefe4	\$fp = 0x7ffeff4
0x7ffefe0	\$ra = 0x00400080
0x7ffefdc	\$a0 = 0x00000002
0x7ffefd8	\$fp = 0x7ffefe8
0x7ffefd4	\$ra = 0x00400080
0x7ffefd0	\$a0 = 0x00000001

Code Tính giai thừa không sử dụng thanh ghi \$fp

```
# Laboratory Exercise 7, Assignment 4 Version 2
# Author: Pham Huy Canh

.data
    message1: .asciiz "Nhap so N (N khong am): "
    message2: .asciiz "Ket qua tinh giai thua la: "

.text
main:    li $v0, 51
        la $a0, message1          # Hien message1 & Nhap so N
        syscall
        bltz $a0, main            # Kiem tra N khong am
        nop

        jal fact
        nop

        add $a1, $v0, $zero       # $a1 = ket qua
        li $v0, 56
        la $a0, message2         # Hien message2 va ket qua
        syscall

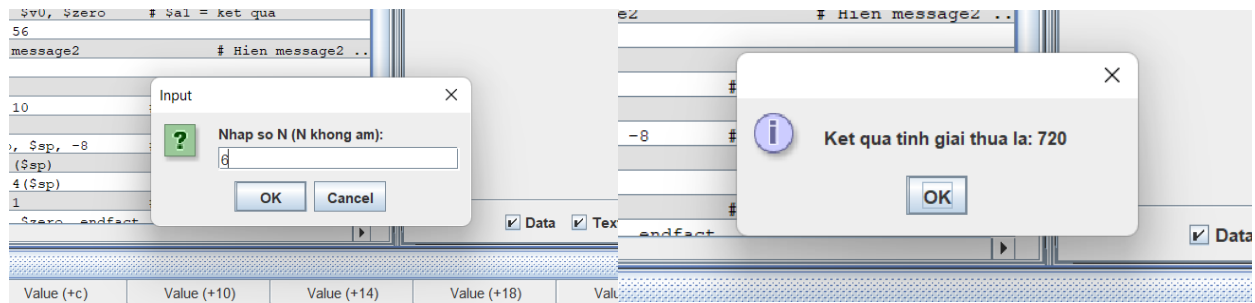
        li $v0, 10               # Exit
        syscall

fact:    addi $sp, $sp, -8         # Cap phat Stack
        sw $ra, ($sp)
        sw $s0, 4($sp)

        li $v0, 1                # v0 = 1
        beq $a0, $zero, endfact   # N = 0 branch to the endfact
        nop
        add $s0, $a0, $zero       # s0 = N
        addi $a0, $a0, -1         # N = N - 1
        jal fact
        nop
        mul $v0, $v0, $s0         # N! = N * (N-1) * (N-2) * ... * 1

endfact: lw $ra, ($sp)
        lw $s0, 4($sp)
        addi $sp, $sp, 8         # Giai phong Stack
        jr $ra
```

Kết quả:



5. Assignment 5:

Source code:

<https://drive.google.com/drive/folders/1CP9Z-YxCokRWsQw6N1kVMwdhjD584cXr?usp=sharing>

Minh hoạ kết quả chạy chương trình:

D:\Courses\2021\ThucHanhKienTrucMayTinh\Assignments\Week7\Assignment5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Execute

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x24100005	addiu	\$16,\$0,0x0000...	8: main: li \$a0, 5
0x00400004	0x2411ffff	addiu	\$17,\$0,0xffff...	9: li \$a1, -123
0x00400008	0x24120038	addiu	\$18,\$0,0x0000...	10: li \$a2, 56
0x0040000c	0x2413000c	addiu	\$19,\$0,0x0000...	11: li \$a3, 12
0x00400010	0x24140057	addiu	\$20,\$0,0x0000...	12: li \$a4, 87
0x00400014	0x2415fe50	addiu	\$21,\$0,0xffff...	13: li \$a5, -432
0x00400018	0x2416fff2	addiu	\$22,\$0,0xffff...	14: li \$a6, -212
0x0040001c	0x24170061	addiu	\$23,\$0,0x0000...	15: li \$a7, 97
0x00400020	0x0c18000a	jal	0x00400068	17: jal saveNumbers
0x00400024	0x00000000	nop		18: nop
0x00400028	0x24020004	addiu	\$2,\$0,0x0000...	19: li \$v0, 4
0x0040002c	0x3c011001	lui	\$1,0x00001001	20: la \$a0, largest
0x00400030	0x34240000	ori	\$4,\$1,0x00000000	
0x00400034	0x00000000	syscall		21: syscall

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x6772614c	0x3a747365	0x530a0020	0x6c6c616d	0x3a747365	0x202c0020	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages

Run IO

largest: 97, 7
smallest: -432, 5
-- program is finished running --

Clear

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000005
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0xfffffe50
\$t2	10	0x00000007
\$t3	11	0x00400024
\$t4	12	0x00000000
\$t5	13	0x00000007
\$t6	14	0x00000005
\$t7	15	0x00000000
\$t8	16	0x00000005
\$t9	17	0xffffffff
\$a2	18	0x00000038
\$a3	19	0x0000000c
\$a4	20	0x00000057
\$a5	21	0xfffffe50
\$a6	22	0xffffffff
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x7ffffefc
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00400024
\$pc		0x00400034
\$hi		0x00000000
\$lo		0x00000000