**Laboratory Exercise 7  
Procedure calls, stack and parameters**

**Nguyễn Hải Dương - 20194530**

**Assignment 1**

Text

Description automatically generated

**Chương trình:**

**Graphical user interface, application, table

Description automatically generated**

**Ý nghĩa các thanh ghi:**

|  |  |
| --- | --- |
| $a0 | Đối số đầu vào |
| $v0 | Lưu giá trị tuyệt đối của đối số |
| $s0 | Lưu giá trị tuyệt đối để nhường thanh ghi $v0 cho syscall |
| $ra | Return, kết thúc procedure abs |

**Ý nghĩa chương trình:**

* Input: một số nguyên ($a0) – MSSV: 20194530
* Ouput: giá trị tuyệt đối của số đó ($s0)
* Thuật toán:

+ gọi thủ tục abste

+ lưu số đối của số nguyên –(a0) vào thanh ghi $v0, nếu số đó là số âm -> done -> return về vị trí sau lệnh jal

+ nếu không thì ta lưu số nguyên đó (a0) vào $v0 -> done -> return về sau lệnh jal

+ lưu giá trị ở $v0 vào $s0 sau đó sử dụng $v0 để gọi syscall

* Giải thích các lệnh quan trọng:
* Jal <target> : nhảy đến target, đồng thời set thanh ghi $ra bằng thanh ghi $pc ( tức là lệnh tiếp theo ngay sau lệnh jal )
* Jr <register> : nhảy đến địa chỉ lưu trong thanh ghi, lệnh jal và jr $ra thường đi kèm với nhau để tạo nên một thủ tục

**Debug từng dòng:**

|  |  |  |
| --- | --- | --- |
| Step | $pc | Giá trị thanh ghi thay đổi |
| 1 | 0x00400004 | $at = 0x01340000 |
| 2 | 0x00400008 | $a0 = 0x013424e2 |
| 3 | 0x0040001c | $ra = 0x0040000c |
| 4 | 0x004000020 | $v0 = 0xfecbdb1e |
| 5 | 0x004000014 | $s0 = 0x013424e2 |
| 6 | 0x004000018 | $v0 = 0x0000000a |
| 7 | 0x00400001c | $v0 = 0x0000000a |

**Assignment 2**

Text

Description automatically generated

**Chương trình:**

**Graphical user interface, application, table

Description automatically generated**

**Ý nghĩa các thanh ghi:**

|  |  |
| --- | --- |
| $a0 | Đối số đầu vào |
| $a1 | Đối số đầu vào |
| $a2 | Đối số đầu vào |
| $v0 | Lưu số lớn nhất |
| $t0 | Biến sử dụng cho phép so sánh <0 ( hiệu của hai số) |

**Ý nghĩa chương trình:**

* Input: 3 số nguyên ($a0,$a1,$a2)
* Ouput: số nguyên lớn nhất trong 3 số ($v0)
* Thuật toán:

+ gọi thủ tục max :

- gán $v0 = $a0 ( tạm coi $a0 là max)  
 - so sánh $v0 với $a1, nếu $a1 lớn hơn thì đưa $a1 lên làm max

- ngược lại tiếp tục so sánh $v0 với $a2 nếu $a2 lớn hơn thì đưa $a2 lên làm max

- return về vị trí sau lệnh jal

**- Debug từng dòng:**

|  |  |  |
| --- | --- | --- |
| Step | $pc | Giá trị thanh ghi thay đổi |
| 1 | 0x00400004 | $a0 = 0x00000005 |
| 2 | 0x00400008 | $a1 = 0x00000003 |
| 3 | 0x0040000c | $a2 = 0x00000000 |
| 4 | 0x004000018 | $ra = 0x00000010 |
| 5 | 0x00400001c | $v0 = 0x00000005 |
| 6 | 0x004000020 | $t0 = 0xfffffffe |
| 7 | 0x00400002c | $t0 = 0xfffffffe |
| 8 | 0x004000030 | $t0 = 0xfffffffb |
| 9 | 0x00400003c | $t0 = 0xfffffffb |
| 10 | 0x004000010 | $t0 = 0xfffffffb |
| 11 | 0x004000014 | $t0 = 0xfffffffb |
| 12 | 0x00400003c | $t0 = 0xfffffffb |
| 13 | 0x004000010 | $t0 = 0xfffffffb |

**Kết quả:** Số lớn nhất trong {5, 3, 0} là 5

**Assignment 3**

Text

Description automatically generated

**Chương trình:**

**Graphical user interface, application, table

Description automatically generated**

**Ý nghĩa các thanh ghi:**

|  |  |
| --- | --- |
| $s0 | Đối số đầu vào (4 số đầu MSSV) |
| $s1 | Đối số đầu vào (4 số cuối MSSV) |
| $sp | Đóng vai trò như một ngăn xếp để lưu các giá trị |

**Ý nghĩa chương trình:**

* Input: 2 số nguyên ($s0,$s1)
* Ouput: đổi vị trí 2 số nguyên đó trong 2 thanh ghi $s0, $s1
* Thuật toán:

+ khai báo một stack chứa được 2 số nguyên

+ lưu lần lượt 2 số nguyên vào stack

+ pop chúng ra theo thứ tự ngược lại vào địa chỉ của 2 thanh ghi $s1 và $s0

|  |  |  |
| --- | --- | --- |
| Step | $pc | Giá trị thanh ghi thay đổi |
| 1 | 0x00400004 | $s0 = 0x000007e3 |
| 2 | 0x00400008 | $s1 = 0x000011b2 |
| 3 | 0x0040000c | $sp = 0x7fffeff4 |
| 4 | 0x00400010 | $sp = 0x7fffeff4 |
| 5 | 0x00400014 | $sp = 0x7fffeff4 |
| 6 | 0x00400018 | $sp = 0x7fffeff4 |
| 7 | 0x0040001c | $sp = 0x7fffeff4 |
| 8 | 0x00400020 | $sp = 0x7fffeff4 |
| 9 | 0x00400024 | $sp = 0x7fffeff4 |
| 10 | 0x00400028 | $s1 = 0x000011b2 |
| 11 | 0x0040002c | $sp = 0x7fffeffc |

**Home Assignment 4**

Text, letter

Description automatically generated

Text

Description automatically generated

**Chương trình:**

**Graphical user interface, text, application

Description automatically generated**

**Kết quả:** 0! = 4 (0 là số cuối MSSV)

**Ý nghĩa các thanh ghi:**

|  |  |
| --- | --- |
| $sp | Con trỏ stack pointer |
| $fp | Lưu con trỏ đến khung trang frame pointer |
| $a0 | Kết quả n! |
| $v0 | Lưu số lớn nhất, đồng thời sử dụng cho syscall |
| $ra | Thanh ghi chứa địa chỉ return |
| $t0 | Thanh ghi chứa biến so sánh làm điều kiện kết thúc |
| $v1 | Thanh ghi chứa giá trị lấy ra từ stack |
| $a1 | Lưu kết quả cuối cùng sử dụng cho hiển thị của lệnh syscall 56 |

**Ý nghĩa chương trình:**

* Input: số nguyên n>0
* Ouput: n!
* Thuật toán:

+ gọi thủ tục wrap :

- trong thủ tục này ta khai báo 2 vị trí cho khung trang và địa chỉ return

- load giá trị n vào thanh ghi $a0

- gọi đến thủ tục fact

+ thủ tục fact:

* + khai báo 3 vị trí cho khung trang, địa chỉ return và số nguyên dùng cho việc tính n!
  + Khung trang trỏ đến vị trí số nguyên trước đó được lưu lại ở mỗi lần lặp
  + Khi đã lưu xong các số nguyên dùng cho việc tính n! Thì nhảy đến done
* Khi nhảy đến nhãn done ta lần lượt pop ra các giá trị đã lưu ra từ trước để tính toán. Các giá trị thanh ghi $ra trước đó đã lưu trong stack giúp ta return về vị trí lệnh jal qua đó kết thúc từng thủ tục một và quay về với chương trình chính (main). Các giá trị của thanh ghi $fp đưa ta đến từng khung trang, nơi mà mỗi vị trí tương ứng sẽ có 3 tham số n, $ra, $fp

\*mô tả trong hình dưới

Timeline

Description automatically generated

**Debug từng dòng:**

|  |  |  |
| --- | --- | --- |
| Step | $pc | Giá trị thanh ghi thay đổi |
| 1 | 0x00400020 | $ra = 0x00400004 |
| 2 | 0x00400024 | $ra = 0x00400004 |
| 3 | 0x00400028 | $fp = 0x7fffeffc |
| 4 | 0x0040002c | $sp = 0x7fffeff4 |
| 5 | 0x00400030 | $sp = 0x7fffeff4 |
| 6 | 0x00400034 | $a0 = 0x00000000 |
| 7 | 0x0040004c | $ra = 0x00000038 |
| 8 | 0x00400050 | $ra = 0x00000038 |
| 9 | 0x00400054 | $fp = 0x7fffeff4 |
| 10 | 0x00400058 | $sp = 0x7fffeff8 |
| 11 | 0x00400064 | $t0 = 0x00000001 |
| 12 | 0x00400070 | $v0 = 0x00000001 |
| 13 | 0x00400094 | $ra = 0x00000038 |
| 14 | 0x00400098 | $a0 = 0x00000000 |
| 15 | 0x0040009c | $fp = 0x7fffeff4 |
| 16 | 0x004000a0 | $fp = 0x7fffeffc |
| … | … | … |

**Assignment 5**

**Code:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**A picture containing table

Description automatically generated**

**Text

Description automatically generated**

**Chương trình:**

**Graphical user interface, application, table

Description automatically generated**

**Output:**

**Text

Description automatically generated**

**Debug từng dòng:**

|  |  |  |
| --- | --- | --- |
| Step | $pc | Giá trị thanh ghi thay đổi |
| 1 | 0x00400004 | $s0 = 0x00000002 |
| 2 | 0x00400008 | $s1 = 0x00000000 |
| 3 | 0x0040000c | $s2 = 0x00000001 |
| 4 | 0x0040002c | $sp = 0x7fffeff4 |
| 4 | 0x00400010 | $s3 = 0xfffffff7 |
| 3 | 0x00400014 | $s4 = 0x00000004 |
| 4 | 0x00400018 | $s5 = 0xffffffb |
| 7 | 0x0040001c | $s6 = 0x00000003 |
| 8 | 0x00400020 | $s7 = 0x0000000 |
| 9 | 0x00400088 | $ra = 0x00400024 |
| … | … | … |
| 4 | 0x004000ac | $fp = 0x7fffefdc |
| 3 | 0x004000b0 | $ra = 0x0040002c |
| 4 | 0x004000b4 | $v1 = 0x00000002 |
| 7 | 0x004000b8 | $sp = 0x7fffeff8 |
| 8 | 0x004000c0 | $t0 = 0x00000000 |
| 9 | 0x004000c8 | $t1 = 0x00000002 |
| 4 | 0x004000b4 | $v1 = 0x00000000 |
| 7 | 0x004000b8 | $sp = 0x7fffeff4 |
| 8 | 0x004000c0 | $t0 = 0x00000001 |
| 9 | 0x004000b4 | $v1 = 0x00000001 |
| 7 | 0x004000b8 | $sp = 0x7fffeff0 |
| … | … | … |