Procedure (IA32) – Bài tập 3a

Code assembly

```
1   proc:
2     pushl   %ebp
3     movl    %esp, %ebp
4     subl    $40, %esp
5     leal    -4(%ebp), %eax
6     movl    %eax, 8(%esp)
7     leal    -8(%ebp), %eax
8     movl    %eax, 4(%esp)
9     movl    $.LC0, (%esp)     Pointer
10    call    scanf
      Diagram stack frame at this point
11    movl    -4(%ebp), %eax
12    subl    -8(%ebp), %eax
13    leave
14    ret
```

Code C

```
1   int proc(void)
2   {
3       int x,y;
4       scanf("%x %x", &y, &x);
5       return x-y;
6   }
```

Giả sử:

- Khi mới bắt đầu thực thi **proc** (dòng 1):

Register	Value
%esp	0x800040
%ebp	0x800060

a. Giá trị của **%ebp** sau dòng lệnh thứ 3 (có giải thích)?

Dòng 2 thực hiện việc push giá trị **%ebp** vào stack, giá trị **%esp** không thay đổi.

Dòng 3 đưa giá trị **%esp** vào **%ebp**, lúc này cả hai thanh ghi **%esp** và **%ebp** trỏ vào đầu stack, stack mới được khởi tạo rỗng phần tử

Giá trị **%ebp** lúc này là: **0x80040**

b. Giá trị của **%esp** sau dòng lệnh thứ 4 (có giải thích)?

Sau 3 dòng đầu, theo như trên thì thanh ghi **%esp** không có gì thay đổi.

Đến dòng thứ 4 thì **%esp** bị trừ đi **$40** (**0x28**).

Giá trị **%esp = 0x800060 – 0x28 = 0x800038**

c. Đoạn code truyền tham số và gọi **scanf**? Giải thích?

Đoạn code truyền tham số: 5 – 9

Theo stack frame thì **x,y** lần lượt ở vị trí **-4(%esp**) và **-8(%esp**)

Tính từ trái sang:

- Tham số thứ 3: sẽ lấy địa chỉ tại **-4(%esp**) tức là x đưa vào (thông qua thanh ghi **%eax**)

- Tham số thứ 2: lấy địa chỉ tại **-8(%esp**) tức là y đưa vào (thông qua thanh ghi **%eax**)

- Tham số thứ 1: chỗ này hơi đặc biệt, đại khái theo như gg và comment bên trên thì ta có thể biết được là lấy địa chỉ của chuỗi **"%x %x"** để đưa vào. Nhưng đồng thời phải có label của **.LCO**

→ Sau đó gọi hàm **scanf**

d. Xác định vị trí lưu của **x** và **y**? Giải thích?

Trích lại (lỡ giải thích bên trên rồi :v)

"Theo stack frame thì **x,y** lần lượt ở vị trí **-4**(**%ebp**) và **-8**(**%ebp**)"

Vì x, y là biến cục bộ được khai báo ở hàm proc **nên x,y** được lưu theo thứ tự địa chỉ giảm dần, liền kề và dưới thanh ghi cố định **%ebp**

e. Vẽ stack sau khi thực hiện lệnh **call scanf**

Trước khi thực hiện **call scanf**:

Địa chỉ cao

Địa chỉ thấp

Chắc đề muốn vẽ thời điểm này, còn lúc sau stack thì nó lấy giá trị x – y. Rồi thực hiện leave (mov lại %esp vào %esp và pop old %ebp ra khỏi stack) và ret (nhảy lại đến địa chỉ trả về) nên stack chẳng còn gì 😊