

Họ và tên: Thái Ngọc Diễm Trinh	
MSSV: 22521541	Lớp: NT209.O22

## BÀI TẬP BONUS

### MACHINE PROGRAMMING BASIC

Giả sử ta có đoạn mã assembly như bên dưới:

<p><i>x tại ô nhớ (%ebp+8), n tại ô nhớ (%ebp+12)</i></p> <ol style="list-style-type: none"> <li>1. <code>movl 12(%ebp), %ecx //n</code></li> <li>2. <code>movl 8(%ebp), %edx //x</code></li> <li>3. <code>xorl %eax, %eax</code></li> <li>4. <code>addl \$1, %eax</code></li> <li>5. <code>sall %ecx, %eax</code></li> <li>6. <code>subl \$1, %eax</code></li> <li>7. <code>andl %edx, %eax</code></li> </ol>
--

Trả lời các câu hỏi sau:

#### 1. Instruction thứ 3 (lệnh xor) có tác dụng gì?

Ta có bảng chân trị của XOR như sau:

0	0	0
0	1	1
1	0	1
1	1	0

Lệnh `xorl %eax, %eax` nghĩa là  $\%eax = \%eax \text{ xor } \%eax = 0$ . Hay có nghĩa là ta gán giá trị của 1 biến tại ô nhớ `%eax` bằng 0. Ví dụ `int result = 0`.

#### 2. Instruction thứ 5 thực hiện các phép dịch bit (`sall`) với số bit cần dịch lưu trong thanh ghi `%ecx`, tuy nhiên đang bị lỗi. Lý giải nguyên nhân bị lỗi và sửa lại cho đúng?

Lệnh `sall %ecx, %eax` thực hiện phép dịch bit sang trái (`<<`) trên giá trị của thanh ghi `%eax`, số lượng bit dịch được xác định bởi giá trị `%ecx`.

Lỗi có thể xảy ra nếu giá trị của `%ecx` vượt quá số lượng bit trong `%eax`. Trong Intel x86 assembly yêu cầu số lượng bit cần dịch nằm trong thanh ghi `%cl` hoặc 1 giá trị immediate.

Giải pháp: có thể đổi thành `sall %cl, %eax`. Khi đó sẽ dịch `%eax` sang trái, số lượng bit được chỉ định bởi 8 bit thấp nhất trong thanh ghi `%ecx`.

#### 3. Viết hàm C tương ứng với mã assembly trên: `int bonus(int x, int n)`. Thử dự đoán chức năng của đoạn mã này?

```
int bonus (int x, int n)
```

```
{
```

```
    int result = 0;
```

```
    result = result + 1;  
    result = result << n;  
    result = result - 1;  
    result = result & x;  
}
```

Chức năng: Tính giá trị của  $(2^n - 1) \& x$