CS201
Homework 02

Le Duong Cong Duc
Student ID: 1651044

November 19, 2017

Contents

1 Exercises 1

Write x86-64 assembly code for swapping the contents of two registers, $\%rax$ and $\%rbx$. You may NOT use any other registers.

Assembly code:

```
xor %rax, %rbx
xor %rbx, %rax
xor %rax, %rbx
```

2 Exercises 2

In the following code, A and B are constants defined with #defined:

```
typedef struct
{
        int x[A][B]; /* Unknown constants A and B */
        long y;
}str1;

typedef struct
{
        char array[B];
        int t;
        short s[A];
        long u;
}str2;

void setVal(str1 *p, str2 *q)
{
        long v1 = q->t;
        long v2 = q->u;
        p->y = v1 + v2;
}
```

GCC generates the following code for setVal:

```
        ; void setVal(str1 *p, str2 *q)
        ; p in %rdi, q in %rsi

.setVal:
        movslq  8(%rsi), %rax
        addq    32(%rsi), %rax
        movq    %rax, 184(%rdi)
        ret
```

What are the values of A and B? (The solution is unique)

Your answer:

2A. A = 9

2B. B = 5

3 Exercises 3

For a function with prototype

```
long decode2(long x, long y, long z);
```

GCC generates the following assembly code:

```
decode2:
        subq    %rdx, %rsi
        imulq   %rsi, %rdi
        movq    %rsi, %rax
        salq    $63, %rax
        sarq    $63, %rax
        xorq    %rdi, %rax
        ret
```

Parameters `x,y,z` are passed in registers `%rdi, %rsi, %rdx`. The code stores the return value in register `%rax`.

Write C code for `decode2` that will have an effect equivalent to the assembly code shown.

C code:

```
//write your answer here
long decode2(long x, long y, long z)
{
    long result;
    y = y - z;
    x = x * y;
    result = y;
    result = result << 63;
    result = result >> 63;
    result = result ^ x;
    return result;
}
```

4 Exercises 4

Consider the following assembly code:

```
        ;long loop(long x, int n)
        ;x in %rdi, n in %esi
loop:
        movl    %esi, %ecx
        movl    $1, %edx
        movl    $0, %eax
        jmp     .L2
.L3:
        movq    %rdi, %r8
        andq    %rdx, %r8
        orq     %r8, %rax
        salq    %cl, %rdx
.L2:
        testq   %rdx, %rdx
        jne     .L3
        rep     ; ret
```

The preceding code was generated by compiling C code that has the the following overall form:

```
//Fill the missing parts
long loop(long x, long n)
{
        long result = 0;
        long mark;
        for (mark = 1; mark != 0; mark = mark << n)
        {
                result |= x & mark;
        }
        return result;
}
```

Your task is to fill in the missing parts of the C code to get a program equivalent to the generated assembly code. Recall that the result of the function is returned in register `%rax`.

5 Exercises 5

The following code transposes the elements of an M x M array, where is constant defined by #defined:

```
void transpose(long A[M][M])
{
        long i, j;
        for (i = 0; i < M; i++)
                for (j = 0; j < i; j++)
                {
                        long t = A[i][j];
                        A[i][j] = A[j][i];
                        A[j][i] = t;
                }
}
```

When compiled with optimization level -01, GCC generates the following code for the inner loop of the function:

```
.L6:
        movq    (%rdx), %rcx
        movq    (%rax), %rsi
        movq    %rsi,   (%rdx)
        movq    %rcx,   (%rax)
        addq    $8, %rdx
        addq    $120, %rax
        cmpq    %rdi, %rax
        jne     .L6
```

We can see that GCC has converted the array indexing to pointer code. Your answer:

5A. The register holds a pointer to array element A[i][j] = %rdx

5B. The register holds a pointer to array element A[j][i] = %rax

5C. M = 15