CS105 – Computer Systems Fall 2020

Problem Session 3: Arithmetic and Control Flow in Assembly

SOLUTION

September 9, 2020

1. Match each snippet of assembly code on the left with the equivalent C function on the right.

```
foo1:
    movl %edi,%eax
    sall $4,%eax
    subl %edi,%eax
    ret


foo2:
    movl %edi,%eax
    testl %eax,%eax
    jge .L4
    addl $15,%eax
.L4:
    sarl $4,%eax
    ret


foo3:
    movl %edi,%eax
    shrl $31,%eax
    ret


foo4:
    movl %edi,%eax
    sall $4,%eax
    addl %edi,%eax
    addl %eax,%eax
    ret
```

```
int choice1(int x){
    return (x < 0);
}

int choice2(int x){
    return (x << 31) & 1;
}

int choice3(int x){
    return 15 * x;
}

int choice4(int x){
    return (x + 15) /4
}

int choice5(int x){
    return x / 16;
}

int choice6(int x){
    return (x >> 31);
}

int choice7(int x){
    return x * 30;
}

int choice8(int x){
    return x * 34;
}

int choice9(int x){
    return a * 18;
}
```

Solution:
foo1 corresponds to choice3
foo2 corresponds to choice5.
foo3 corresponds to choice1
foo4 corresponds to choice8

2. Consider the following assembly code for a C function `looper` and compiled on an x86-64 machine:

```
looper:
    movq    $0, %rax
    movq    $0, %rdx
    jmp     .L2
.L4:
    movq    %rdx, %rcx
    leaq    (%rsi,%rcx,4), %rcx
    cmpq    %rcx, %rax
    jl      .L3
    movq    %rax, %rcx
.L3:
    leaq    1(%rcx), %rax
    addq    $1, %rdx
.L2:
    cmpq    %rdi, %rdx
    jl      .L4
    rep ret
```

(a) For each variable, indicate which register that variable is stored in.

- n: %rdi
- a: %rsi
- x: %rax
- i: %rdx

(b) Based on the assembly code, fill in the blanks in the C source code.

```
int looper(int n, int *a) {
    int i;
    int x = 0;

    for(i = 0; i < n; i++) {
        if (x < a+i*4) {
            x = a + i*4 +1;
        else {
            x++;
        }
    }

    return x;
}
```