ICS Homework 6

October 31, 2019

1 Arithmetic and Logical Operations

Suppose a 64-bit little endian machine has the following memory and register status:

Address	Value	Register	Value
0x100	0x0000000000002019	%rax	0x2121
0x108	0xffffffffaabb8922	%rbx	0x100
0x110	0x1212121212121212	%rcx	0x2
0x118	0x1300130013001300	%rdx	0x9

Each operation take effect on the status of memory and register, please fill in the blanks in the following table:

Operation	Destination	Value
subq (%rbx),%rax	%rax	0x108
incq -8(%rax)	0x100	0x000000000000201a
decq %rdx	%rdx	0x8
imulq $4,0x100(%rdx,%rcx,4)	0x110	0x4848484848484848
shrq $4,%rax	%rax	0x10
imulq 0x10	%rax, %rdx	0x100, 0x0
notw (%rax,%rdx)	0x100	0xdfe5
andq 0x10(%rax,%rcx,4),%rax	%rax	0x100
leaq 9(%rax,%rcx,8),%rdx	%rdx	0x119

2 Imply C codes from Assembly

You are given prototypes of three functions and assembly codes when they are compiled. Please write C code for the functions that will have equivalent effect as the assembly code shown. Function parameters a, b, c, and d are stored in registers $\%rdi$, $\%rsi$, $\%rdx$, and $\%rcx$, respectively.

Function prototypes:

```
long f1(long a, long b, long c, long d);
long f2(long a, long b, long c);
long f3(long a, long b);
```

Assembly codes:

```
f1:
  leaq  (%rsi,%rsi,2), %rax
  leaq  (%rax,%rdi,2), %rax
  leaq  (%rdx,%rdx,2), %rdx
  addq  %rdx, %rax
  leaq  (%rcx,%rcx,2), %rdx
  addq  %rdx, %rax
  ret
f2:
  leaq  -1(%rsi), %rcx
  imulq %rcx, %rdx
  leaq  (%rdx,%rdi,2), %rax
  imulq %rsi, %rax
  xorq  %rdx, %rdx
  divq  $2
  ret
f3:
  movq  %rsi, %rax
  cmpq  %rsi, %rdi
  jg    .L5
.L4:
  rep ret
.L5:
  movq  %rdi, %rax
  jmp   .L4
```

Please implement $f1$, $f2$, and $f3$:

```
long f1(long a, long b, long c, long d) {
    long t = 2*a + 3*b + 3*c + 3*d;
    return t;
}

long f2(long a, long b, long c) {
    long s = (2*a + (b-1)*c) * b / 2;
    return s;
}

long f3(long a, long b) {
```

```
12      long max;
13      if (a > b)
14          max = a;
15      else
16          max = b;
17      return max;
18  }
```