ICS Homework 9

November 27, 2020

1

Suppose we have two function `A` and `B` and their corresponding assembly code as below. And we also have another function `C` which takes 8 parameters and function `D` which takes 1 parameter are omitted here. Read the code and answer the question below.

```
1  long A(long x) {
2      long a0 = x;
3      long a1 = x + 1;
4      long a2 = x + 2;
5      long a3 = x + 3;
6      long a4 = x + 4;
7      long a5 = x + 5;
8      long a6 = x + 6;
9      long a7 = x + 7;
10
11     a5 += C(a0,a1,a2,a3,a4,a5,(char)a6,&a7);
12     return a5;
13 }
14
15 long B(long n)
16 {
17     long result;
18
19     if (n <= 1)
20         result = 1;
21     else
22         result = n * D(n-1);
23     return result;
24 }
```

```
1  A:
2      pushq %r15                    /* Comment 1 */
3      pushq %r14
4      pushq %r13
5      pushq %r12
6      /* Comment 2: Skip %r11 as ... */
7      pushq %rbx
8      subq $24, %rsp                /* Comment 3 */
9      movq %rdi, %rbx
10     leaq 1(%rdi), %r15
11     leaq 2(%rdi), %r14
12     leaq 3(%rdi), %r13
13     leaq 4(%rdi), %r12
14     leaq 5(%rdi), %r11
15     leaq 6(%rdi), %rax
16     movq %rax, (%rsp)
17     leaq 7(%rdi), %rdx
18     movq %rdx, 8(%rsp)
19     pushq %r11                    /* Comment 4 */
20     /* CODE HERE: Passing parameters to C */
21     ...
22     call C
23     ...
24
25 B:
26     movq %rdi, %r12
27     movl $1, %eax
28     cmpq $1, %rdi
29     jle .L35
30     leaq -1(%rdi), %rdi
31     call D
32     imulq %r12, %rax
33 .L35:
34     ret
```

1. Fill the `Comment 1,2,3,4` to describe the purpose of the instruction.

   Comment 1: Save 'Callee-saved' registers.

   Comment 2: Skip `%r11` as it is a 'Caller-saved' register.

   Comment 3: Allocate stack for local variables.

   Comment 4: Save `%r11` as it is a 'Caller-saved' register and will be used afterwards.

2. Where are the local variables `a0-a7` in function `A` stored before **line 18**? Write the register name or memory address (use `%rsp` to represent it).

variable	location	variable	location
a0	%rbx	a4	%r12
a1	%r15	a5	%r11
a2	%r14	a6	(%rsp)
a3	%r13	a7	8(%rsp)

3. Where the passing parameters a0-a7 should be stored right after calling C? Write the register name or memory address (use %rsp to represent it).

variable	location	variable	location
a0	%rdi	a4	%r8
a1	%rsi	a5	%r9
a2	%rdx	a6	8(%rsp)
a3	%rcx	a7	16(%rsp)

4. Write the assembly code before call C (CODE HERE) to make it function right.

   Any implementation which obeys the calling convention is right. Be careful with pushq %rbx as all data sizes are rounded up to be multiples of eight.

```
1  movq %rbx, %rdi
2  movq %r15, %rsi
3  movq %r14, %rdx
4  movq %r13, %rcx
5  movq %r12, %r8
6  movq %r11, %r9
7  movq (%rsp), %rbx
8  leaq 8(%rsp), %r11
9  pushq %r11
10 pushq %rbx
```

5. What is the possible value of the 8 bytes begin from %rsp + 8 at the beginning of function C and why?

   The lower 1 byte is the same with the lower 1 byte in (x + 0x6). Other bits may variate. It is because that all passing arguments have to be rounded up to the multiples of eight.

6. There is a problem in B. Find the problem and fix it.

   It uses 'callee-saved' register %r12 but when it returns it does not restore its value to original one. Add pushq %r12 at the beginning of the function and add popq %r12 before the ret.

2

For a C function having the general structure

```
typedef long long unsigned u64;
u64 foo(u64 x) {
        return x?foo(x-1)*x:x;
        /* or */
        /* return x?x*foo(x-1):0; */
}
```

GCC generates the following assembly code:

```
foo:
    pushq   %rbx
    movq    %rdi, %rbx
    testq   %rdi, %rdi
    jne     .L4
.L2:
    movq    %rbx, %rax
    popq    %rbx
    ret
.L4:
    leaq    -1(%rdi), %rdi
    call    foo
    imulq   %rax, %rbx
    jmp     .L2
```

Please fill in the missing expressions in the C code shown above.