ICS Homework 8

November 19,2021

1

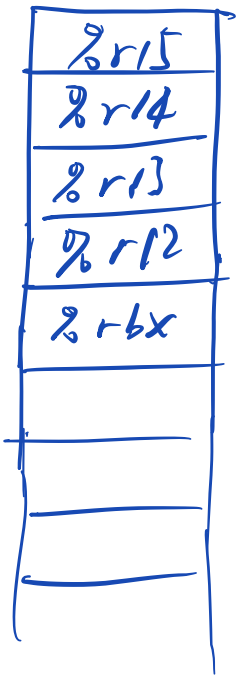
Suppose we have two function `A` and `B` and their corresponding assembly code as below. And we also have another function `C` which takes 8 parameters and function `D` which takes 1 parameter are omitted here. Read the code and answer the question below.

```
long A(long x) {
    long a0 = x;
    long a1 = x + 1;
    long a2 = x + 2;
    long a3 = x + 3;
    long a4 = x + 4;
    long a5 = x + 5;
    long a6 = x + 6;
    long a7 = x + 7;

    a5 += C(a0,a1,a2,a3,a4,a5,(char)a6,&a7);
    return a5;
}

long B(long n)
{
    long result;

    if (n <= 1)
        result = 1;
    else
        result = n * D(n-1);
    return result;
}
```




```
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
```

Save callee-saved registers

Caller-saved register

Allocate stack for local variables.

Save %r11 as a Caller-saved register and will be used afterwards.

pushq %r12

作为"Callee-saved"register，但是返回时它存的值不是原来的值了.

movq $1, %rax

popq %r12

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

4. movq %rbx, %rdi
movq %r15, %rsi
movq %r14, %rdx
movq %r13, %rcx
movq %r12, %r8
movq %r11, %r9

movq (%rsp), %rbx
leaq 8(%rsp), %r11
pushq %r11
pushq %rbx

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)

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

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

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

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

2

For a C function having the general structure

```
1  typedef long long unsigned u64;
2  u64 foo(u64 x) {
3          return (?);
4  }
```

GCC generates the following assembly code:

```
1  foo:
2      pushq   %rbx
3      movq    %rdi, %rbx
4      testq   %rdi, %rdi
5      jne     .L4
6  .L2:
7      movq    %rbx, %rax
8      popq    %rbx
9      ret
10 .L4:
11     leaq    -1(%rdi), %rdi
12     call    foo
13     imulq   %rax, %rbx
14     jmp     .L2
```

Please fill in the missing expressions (?) in the C code shown above.

x ? foo(x-1)*x : x