CSPB 2400 - Park - Computer Systems

Started on	Saturday, 10 February 2024, 10:01 PM
State	Finished
Completed on	Saturday, 10 February 2024, 10:12 PM
Time taken	11 mins 14 secs
Marks	4.00/5.00
Grade	**8.00** out of 10.00 (**80**%)

Question **1**

Correct

Mark 0.50 out of 0.50

The x86-64 calling convention passes arguments to procedures in registers. In what registers are the first two arguments passed?

Select one or more:

- [ ] a. %r12
- [ ] b. %r14
- [ ] c. %rax
- [x] d. %rsi ✔
- [ ] e. %r11
- [ ] f. %r10
- [ ] g. %rbx
- [ ] h. %r8
- [ ] i. %r13
- [ ] j. %r15
- [x] k. %rdi ✔
- [ ] l. %rdx
- [ ] m. %rcx
- [ ] n. %rsp
- [ ] o. %rbp
- [ ] p. %r9

Your answer is correct.

Question **2**

Correct

Mark 0.50 out of 0.50

The x86-64 calling convention uses two registers to manage procedure calls. Select the register that is **always used** for procedure calls.

Select one or more:

- [ ] a. %r15
- [ ] b. %rdi
- [ ] c. %rbp
- [ ] d. %r13
- [ ] e. %rbx
- [ ] f. %r10
- [ ] g. %rsi
- [ ] h. %rdx
- [ ] i. %r12
- [ ] j. %rax
- [ ] k. %r11
- [ ] l. %r8
- [ ] m. %r14
- [ ] n. %r9
- [ ] o. %rcx
- [x] p. %rsp ✔

Your answer is correct.

Question **3**

Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
int a;
int rfoo() {
  int rval;
  if (a == 0)
    return 1;
  rval = rfoo(a>>1);
  return rval * a;
   }
}
```

```
foo:
   movl    a, %ebx
   cmpl    $0, %ebx
   jne     L3
   movl    %ebx,%eax
   shrl    %eax
   pushl   %eax
   call    rfoo
   imull   %ebx, %eax
   jmp     L2
.L3:
   movl    $1, %eax
.L2:
   ret
```

+ | - | / | >> | <<

Your answer is correct.

Question **4**

Correct

Mark 1.00 out of 1.00

Given the following assembly code:

```
rfun:
        movl    $0, %eax
        cmpq    $100, %rdi
        ja      .L18
        pushq   %rbx
        movq    %rdi, %rbx
        leaq    (%rdi,%rdi), %rdi
        call    rfun
        subq    %rax, %rbx
        movq    %rbx, %rax
        popq    %rbx
.L18:
        rep ret
```

Fill in the blanks by dragging the appropriate entries below:

```
          unsigned lon
long rfun(      g       ✔  x){
  if ( x > 100 ✔  ) return 0;
  unsigned lon
       g       ✔  nx = x * 2 ✔ ;
  long rv = rfun(nx);
  return x - rv ✔ ;
}
```

long | int | unsigned int

x == 0 | x < 0 | x > 100 | x * 2 | x >> 2 | x / 8 | x * rv | x - rv | x + rv

Your answer is correct.

Question **5**

Correct

Mark 1.00 out of 1.00

Given the following C code:

```
long rfun(int x){
  if ( x == 0 ) return 0;
   int nx = x >> 2;
  long rv = rfun(nx);
  return x - rv;
}
```

Fill in the blanks by dragging the appropriate entries below:

```
rfun:
        movl    $0, %eax
        testl %edi, %ed
                i
        je .L6
        pushq   %rbx
        movl    %edi, %ebx
        sarl
             $2, %edi
        movslq  %edi, %rdi
        call    rfun
        movslq  %ebx, %rbx
        subq
           %rax, %rbx
        movq    %rbx, %rax
        popq    %rbx
.L6:
        rep ret
```

testq %rdi, %rdi | testl %edi, %edi | cmpq $100, %rdi | imulq %rdi, %rax | unsigned int | subq %rax, %rbx | addq %rbx, %rax | imulq %rbx, %rax

shrq $2,%rdi | sarq $2, %rdi | sarl $2, %edi | je .L6 | sarq $3, %rdi | leaq (%rdi,%rdi), %rdi | js .L6 | ja .L6 | shrl $2, %edi

Your answer is correct.

Question **6**

Incorrect

Mark 0.00 out of 1.00

We have Function callee is called by caller as below left and there assembly code on the right:

```
void caller(int x, int y)
{
  int z;
  z = callee(x,y);
  z = z << 2;
  return;
}
int callee(int x, int y)
{
  if (x>y)
        return x;
  else
        return y;
}
```

```
caller:
    endbr64
    push   %rbp
    mov    %rsp,%rbp
    sub    $0x18,%rsp
    mov    %edi,-0x14(%rbp)
    mov    %esi,-0x18(%rbp)
    mov    -0x18(%rbp),%edx
    mov    -0x14(%rbp),%eax
    mov    %edx,%esi
    mov    %eax,%edi
    callq  1129 <callee>
    mov    %eax,-0x4(%rbp)
    shll   $0x2,-0x4(%rbp)
    leaveq
    retq

callee:
    endbr64
    push   %rbp
    mov    %rsp,%rbp
    mov    %edi,-0x4(%rbp)
    mov    %esi,-0x8(%rbp)
    mov    -0x4(%rbp),%eax
    cmp    -0x8(%rbp),%eax
    jle    1144 <callee+0x1b>
    mov    -0x4(%rbp),%eax
    jmp    1147 <callee+0x1e>
    mov    -0x8(%rbp),%eax
    pop    %rbp
    retq
```

When caller starts executing(before **push %rbp**), we have:
%rbp 0x7fffffffdea0
%rsp 0x7fffffffde98

What value does %rsp get when callee starts executing?

Select one:

- a. 0x7fffffffde90 ✖
- b. 0x7fffffffde70
- c. 0x7fffffffde80
- d. 0x7fffffffde78

Your answer is incorrect.