Checkin 33

Give an example of a snippet of x64 code that benefits from two peephole optimizations

Flipped Wednesday



Administrivia

Project 6 oracle is finally posted

This is the last flipped Wednesday in the schedule

Topics

Final code generation
postcompilation

W10 – Question 1

Draw the X64 assembly that dsc might generate for the following code snippet:

```
fn_foo:  enter foo
         [a] := 2
         [b] := 3
  L_1:   [c] := [a] LT64 [b]
         ifz [c] goto L_2
         call bar
         goto L_1
  L_2:   nop
  L_3:   leave
```

W10 – Question 1 Solution

Draw the X64 assembly that dsc might generate for the following code snippet:

```
fn_foo:  enter foo
         [a] := 2
         [b] := 3
 L_1:    [c] := [a] LT64 [b]
         ifz [c] goto L_2
         call bar
         goto L_1
 L_2:    nop
 L_3:    leave
```

```
fn_foo: pushq %rbp
        movq %rsp, %rbp
        addq $16, %rbp
        subq $32, %rsp
        movq $2, -24(%rbp)
        movq $3, -32(%rbp)
 L_1:   movq -24(%rbp), %rax
        movq -32(%rbp), %rbx
        movq $0, %rcx
        cmpq %rbx, %rax
       setlt %cl
        movq %rcx, -40(%rbp)
        movq -40(%rbp), %r11
        cmpq $0, %r11
          je L_2
       callq fn_bar
         jmp L_1
 L_2:   nop
 L_3: addq $32, %rsp
        popq %rbp
        retq
```

W10 – Question 2

Assume a function printString has been defined that prints a string. Show the x64 code corresponding to the following snippet and explain how it would be run:

```
void v(){
        print "Hello world!";
}
```

W10 – Question 2 Solution

Assume a function printString has been defined that prints a string. Show the x64 code corresponding to the following snippet and explain how it would be run:

```
void v(){
        print "Hello world!";
}
```

```
.data
…
STR_1: .asciz "Hello world!"
…

.text
…
fn_v: pushq %rbp
       movq %rsp, %rbp
       addq $16, %rbp
       movq $STR_1, %rdi
      callq printString
       popq %rbp
       retq
```

W10 – Question 3

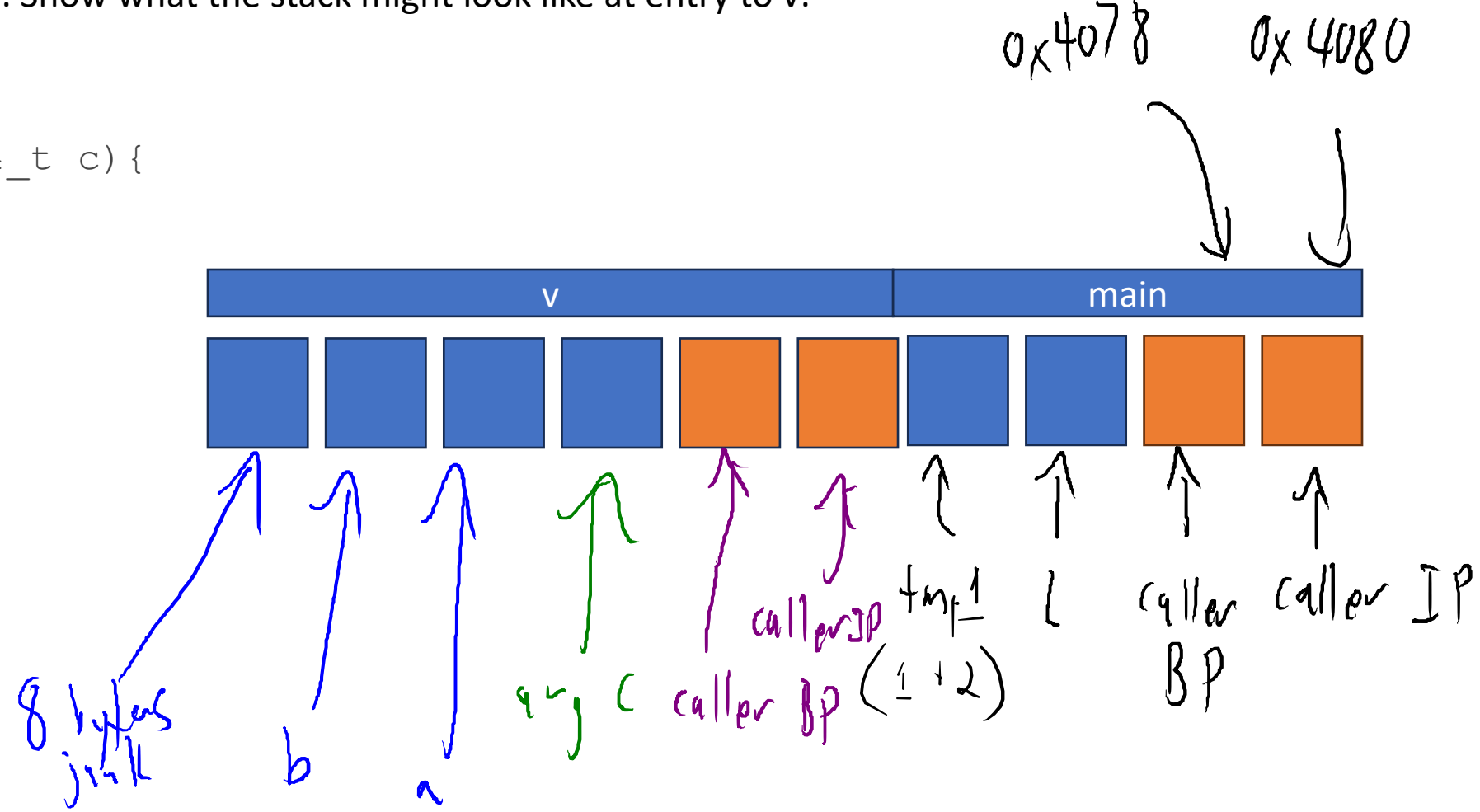
Assume that there is a C compiler that uses the same allocation record scheme that we have presented in class. Show what the stack might look like at entry to v:

```
int64_t v(int64_t c){
    int64_t a;
    int64_t b;
}
void main(){
    int64_t l;
    v(l + 2);
}
```

W10 – Question 3 Solution

Assume that there is a C compiler that uses the same allocation record scheme that we have presented in class. Show what the stack might look like at entry to v:

```
int64_t v(int64_t c){
    int64_t a;
    int64_t b;
}
void main(){
    int64_t l;
    v(l + 2);
}
```



W10 – Question 4

What is an ABI? What is the benefit of allowing two languages to have compatible ABIs?

W10 – Question 4 Solution

What is an ABI? What is the benefit of allowing two languages to have compatible ABIs?