Dove — user-defined functions, compile-time errors

Syntax

```
<program> ::= <declaration_list> ? <expr>
<declaration_list> ::= <declaration>
                     | <declaration> <declaration_list>
<declaration> ::= def <identifier>(<param_list>) <expr> end   } ← fn decl
                | def <identifier> () <expr> end
<param_list> ::= <param>
               | <param> , <param_list>
<param> ::= <identifier>
<expr> ::= ...
         | <identifier>(<arg_list>)   ← fn call

<arg_list> ::= <arg>
             | <arg> , <arg_list>
<arg> ::= <expr>
```

def f(x,y)

f(2,4)

f(if 6 then 3 else false)

Semantics

* call fn: evaluate arg exprs left to right
* assign results to fn params
* evaluate fn body
* result is fn result

printValue(0x8)   8 → rdi
f(4)   4 → stack

Bird Calling Conventions

* Just like x86-64 POSIX C conventions
* Except that all arguments are passed on stack
* And that rdi and rsi are caller-saved instead

	C	Bird
caller-saved	rax, rcx, rdx, r8-r11	rax, rcx, rdx, r8-r11, rsi, rdi
callee-saved	rbx, rsp, rbp, r12-r15, rsi, rdi	rbx, rsp, rbp, r12-r15

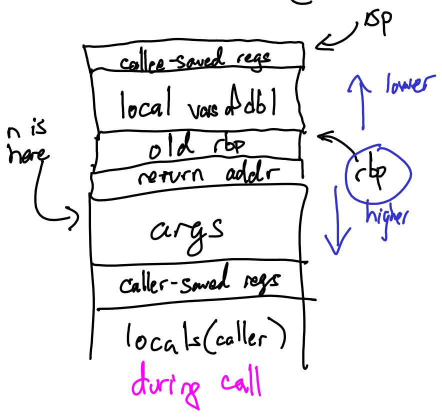
Example

```
def dbl(n)
    n * 2
end

dbl(4)
```

compiled in an env with n ↦ [rbp+16] and offset = -8

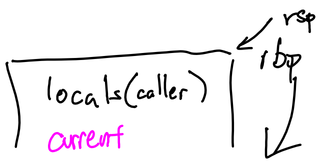
call instruction:
1. Save RIP to top stack
2. Jump to label



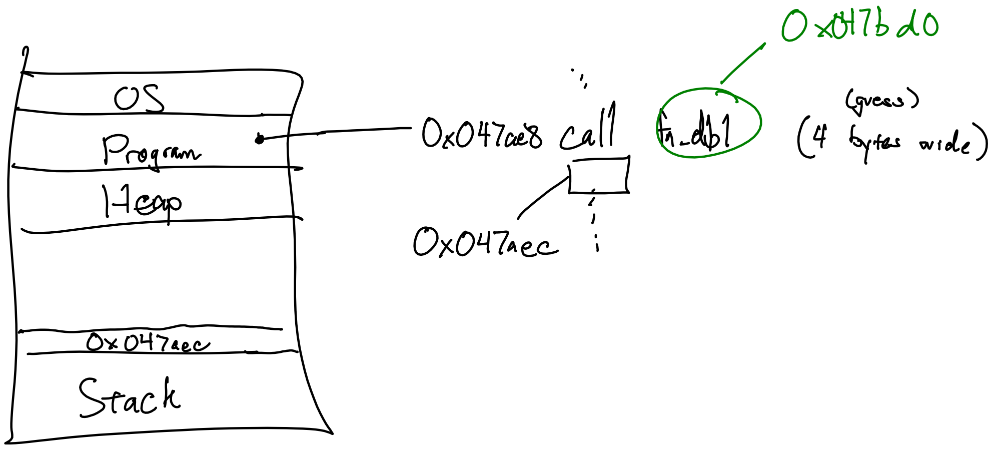
```
fn_dbl:
  push rbp          ; save rbp
  mov  rbp, rsp     ; new stack frame
  sub  rsp, 16      ; make locals space
  ....              ; save callee-saved regs
  ....              ; calculate n*2, leave in rax
  ....              ; restore callee-saved regs
  mov  rsp, rbp     ; remove stack frame
  pop  rbp
  ret               ; return

bird_main:

  mov rax, 4        ; eval arg     } eval args
  mov [rbp-8], rax  ; store arg    }
  ....              ; save caller-saved regs
  mov eax, [rbp-8]  ; put args on stack
  push eax
  call fn_dbl       ; call fn
  ....              ; remove args from stack
  ....              ; restore caller-saved regs
```



Notes from a question asked after lecture:


OS
Program
Heap
0x047aec
Stack
0x047ae8 call fn_dbl
0x047bd0
(guess)
(4 bytes wide)
0x047aec