

Stage 1: no structs, globals, pointers, calls, or functions other than main

Stage 2: add globals



.data
→ .global g
g: .zero 8

g(%rip)

Stage 3: add extern calls

L additional lir: \$call-ext

$x = \$call_ext \text{ foo}(op_1, \dots, op_n)$

• place 1st 6 one-word args into

6 registers: %rdi, %rsi, %rdx, %rcx, %r8, %r9

• if > 6 args, push rest on stack (right to left) of

• call foo

• if we pushed to stack, pop them off

taking
care of
alignment

- if we pushed to stack, pop them off
 - callee's return value is in %rax,
store to X

አንቀጽ ፭

Stage 4 : internal functions & calls

L additional lir : \$call-dir, \$call-idr

Calling convention

- pre-call : sets up call
 - function prologue : accept info. from call
 - function epilogue : tear down callee + return
 - post-call : clean up

caller- vs callee-save registers

L caller-save : caller's responsibility to save this reg. before call if they want the value to last pass the call

L coffee-saver: the seller can assume these are
unouched - coffee is responsible
for restoring value of mg. if
it uses it

• pre-call :  It can raise

- pre-call:
 - save any in-use caller-save reg's.
 - put 7th & on-word args in reg.
 - place any remaining args on stack
 - account for alignment
- ret {
 - push return address to stack
 - jump to callee

- prologue:
 - save any callee-save reg we're going to use
 - set up stack frame
- epilogue:
 - place return value in designated reg
 - deallocate stack frame
 - restore callee-save reg
 - pop return address
 - jump to it
- post-return:
 - pop any args from stack
 - restore caller-save reg.

our calling convention

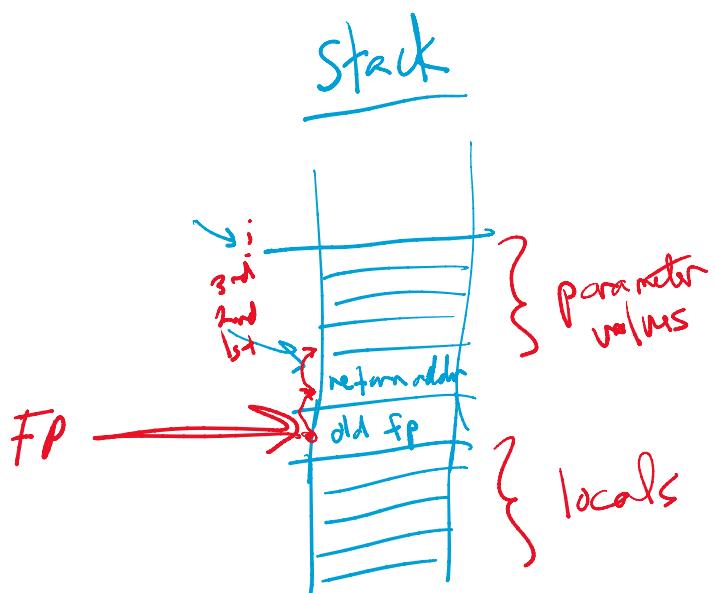
- always push all args to stack

$x = \$call_dir \text{ foo}(op_1, \dots, op_n) \text{ then } bb$

- push $op_1 \dots op_n$ to stack in reverse order
- fix stack alignment
- call foo
- restore stack ptr
- store $\%rax$ to x
- jump to bb

$x = \$call_idr \text{ fp}(op_1, \dots, op_n) \text{ then } bb$

- call $*fp$



implicit function pts

↳ mangle global names (add "-" to end)

L mangle global names (add "-" to end)

L initialize to function add