

Stage 3 Codegen Example 2

Monday, May 6, 2024 11:02 AM

```
extern foo(int, int, int, int, int, int, int) -> int
```

```
fn main() -> int {  
  let x:int
```

```
  entry:  
    x = $call_ext foo(1, 2, 3, 4, 5, 6, 7)  
    $ret x  
}
```

```
.data  
.text  
.globl main
```

```
main:  
  pushq %rbp  
  movq %rsp, %rbp  
  subq $16, %rsp  
  movq $0, -8(%rbp)  
  jmp main_entry
```

```
main_entry:  
  movq $1, %rdi      // first 6 args in regs  
  movq $2, %rsi  
  movq $3, %rdx  
  movq $4, %rcx  
  movq $5, %r8  
  movq $6, %r9  
  pushq $7           // 7th arg on stack  
  subq $8, %rsp      // alignment  
  call foo  
  movq %rax, -8(%rbp) // storing return value  
  addq $16, %rsp     // restoring stack  
  movq -8(%rbp), %rax  
  jmp main_epilogue
```

```
main_epilogue:  
  movq %rbp, %rsp  
  popq %rbp  
  ret
```