

# CS 24

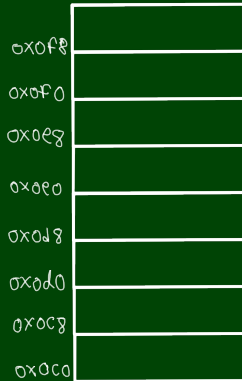
## Introduction to Computing Systems

# x86-64 Stack & Memory

```
movzwl (%rdi),%eax
movzwl (%rsi),%edx
mov %dx,(%rdi)
mov %ax,(%rsi)
retq
```

locals:

```
pushq %rbp
movq  %rsp, %rbp
movq  $10, -8(%rbp)
movq  -8(%rbp), %rax
popq  %rbp
retq
```

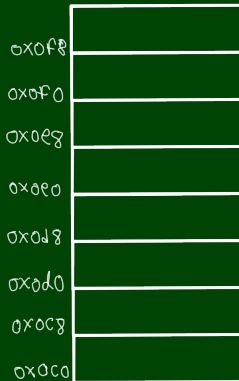


callee:

```
pushq %rbp
movq %rsp, %rbp
movq %rdi, -8(%rbp)
movq -8(%rbp), %rax
popq %rbp
retq
```

caller:

```
pushq %rbp
movq %rsp, %rbp
subq $16, %rsp
movq %rdi, -8(%rbp)
movq -8(%rbp), %rdi
callq callee
addq $1, %rax
addq $16, %rsp
popq %rbp
retq
```

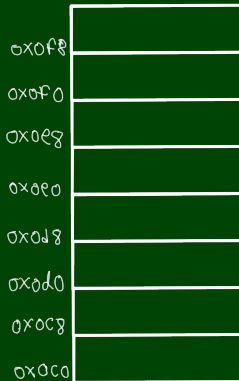


callee:

```
pushq %rbp
movq %rsp, %rbp
movq %rdi, -8(%rbp)
movq -8(%rbp), %rax
popq %rbp
retq
```

caller:

```
pushq %rbp
movq %rsp, %rbp
subq $16, %rsp
movq %rdi, -8(%rbp)
movq -8(%rbp), %rdi
callq callee
addq $1, %rax
addq $16, %rsp
popq %rbp
retq
```



```

gcd:
    mov  %rdi,%rax
    test %rsi,%rsi
    jne  L
    repz retq
L:
    sub  $0x8,%rsp
    mov  %rsi,%rdi
    cqto
    idiv %rsi
    mov  %rdx,%rsi
    callq gcd
    add  $0x8,%rsp
    retq
main:
    sub  $0x8,%rsp
    mov  $0x5,%esi
    mov  $0xa,%edi
    callq gcd
    mov  %rax,%rsi
    ...

```

