



CS 208

F, 17 Oct 2025

(function1 from today's class notes)

```
int function1(char *p) {
```

p in %rdi

```
    int retval ...
```

retval in %eax

```
    return retval;
```

```
}
```

testq %rdi, %rdi

size (64 bits)

bitwise AND of two args (here %rdi + itself)

je .L6

if (p != 0) or maybe

if (p == 0)

pretend function1("cow") (P is a pointer to cow)

testq P, P  
je .L6

is  $P \& P == 0$ ?

NO. so keep going

movl \$0, %eax

jmp .L2

int ret val = 0

↙  
.L2 cmpl \$0, (%rdi)

(%rdi) ← pointer dereferencing  
"the thing whose address  
is in %rdi"

cmpb \$0, (%rdi)

subtract "byte pointed to by %rdi"  
— 0

jne .L4

→ ret if the subtraction gave != 0 value  
jump. Otherwise,

cmpb \$0, (%rdi)  
jne .L4

if (\*p != '\0')  
jump up.

char s[10] = "goat";

int index = 2;

s[index] = 'u';

