



✓ call of compiler

✓ string in memory

✓ CC change

✓ sizeof

id mem error

✓ reg table in  → none

✓ call vs jmp

✓ variable names — lex

✓ ASM → C

✓ string bytes



rax

eax

ax al

Cond Codes

- CMP

- test

all math/logic named inst

add
sub
and
xor

\$0

\$0

\$01

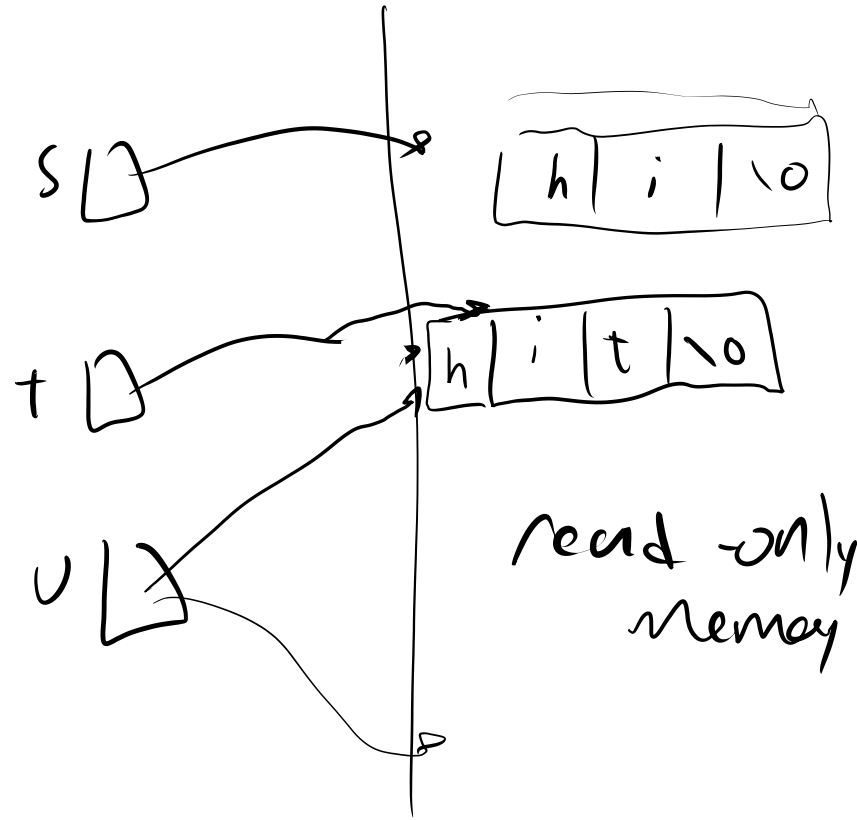
\$-1

s = "hi"

const char *t = "it"

u = "hij"

v = _____



Size of

Compiler

i32x0!y

x+-+*y

((x);

32x7z

CPP

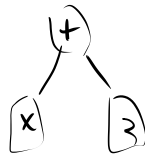
#include
#if
#ifdef
#ifndef
#define

C → C

lex
int

int

Parse



type check

int x = '3'

~~int x = 3.14~~

code gen

asm

link

load

Symbol undefined

→ callq f — pushq address
 jmp f

jmp f

Program register

retq — popq %rip

a base:



loop

-
Jmp above
Jle
Jg

Jbe fem



if

fem

a base:

→ loop

-
jmp above
jle
jg

jbe fem

→ if

fem

a base:

→ loop

-
jmp above
jle
jg

jbe fem

→ if

fem

error find by inspection

1. be pedantic annoyingly precise

2. pick a bug and ask

1. where could it happen

2. how could that line be that bug (conditions)

3. work backward

Struct - malloc?

- array dynamic (run-time size)

- dynamic number of values

not malloc

Stack - it destroyed before returning
↳ in func & not here

global - it known in advance size & count
↳ not in func

man

```

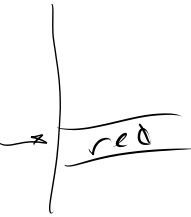
| add $3, %r12
| call baz
| cmp $3, %r12
|

```



Caller save
 man's promise
 you can change
 w/o making me
 mad

Callee save
 baz's promise:
 I won't change register



rdi rsi rdx rcx r8 r9