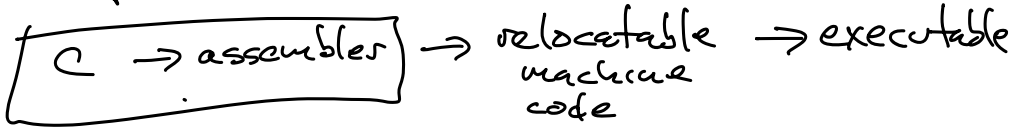


CS 3650 – Computer Systems  
Spring 2024  
Peter Desnoyers

Lecture 3, Tue Jan 16 2024

# C Programming

compilation



file.c file  
(text)

file.o  
(ELF format)

file  
(ELF)

$a = b$   
 $\rightarrow$  LOAD R1 ← R2  
STORE R1  $\rightarrow$  R2

$\boxed{R1 \ 1}$  (addr)  
instructions

F()  $\rightarrow$   $\begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix}$   
...  
}

# relocatable code :

instrs



addresses  
not filled in



tables

addresses

&"var1" = 1234

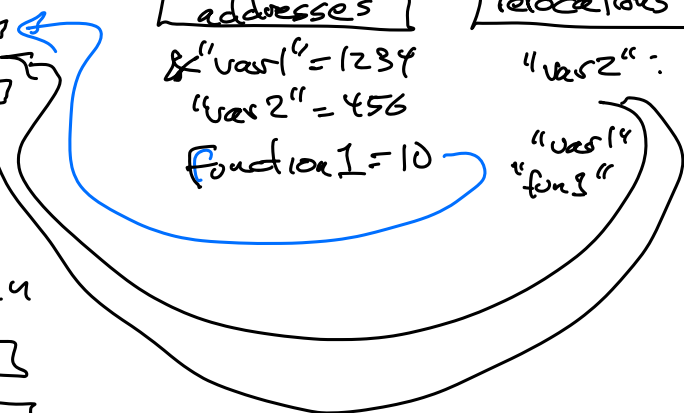
"var2" = 456

function1 = 10

relocations

"var2" :

"var1"  
"func"



gcc -c file1.c -o file1.o

" " file2.c -o file2.o

gcc file1.o file2.o -o my-program

(+ libraries)

output file =

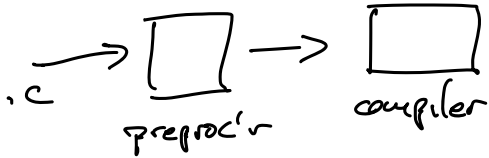
compile,  
don't link

gcc hello.c -o hello  
↑  
gcc is "driver"  
of compilation  
steps

user's view of  
separate compilation & linking

# The C language (origin stuff)

1) Preprocessor



#include <stdio.h>

- 1) file insertion
- 2) text substitution ('macros')

2) single pass compilation

.c

use 'abc'

↑

define 'abc'

.c

~

~

~

~

[symbol  
table]

# C data types

int, long, short, char

4    8    2    1

unsigned < (int, ..., char)

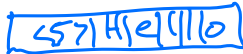
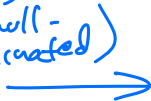
int x = 1;    x = x + 2;

double, float



strings "abc"

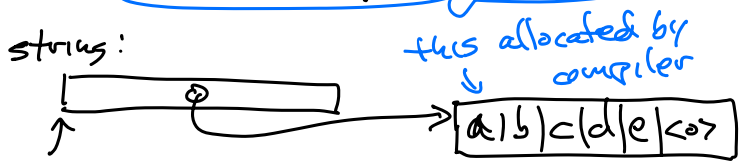
C string  
(zero null-terminated)



how a PL might store strings:



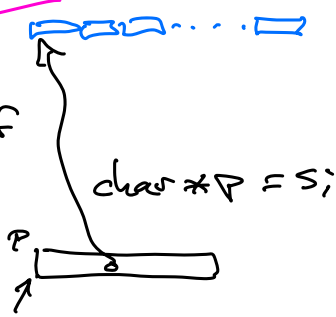
C string: `char *strng = "abcde"`



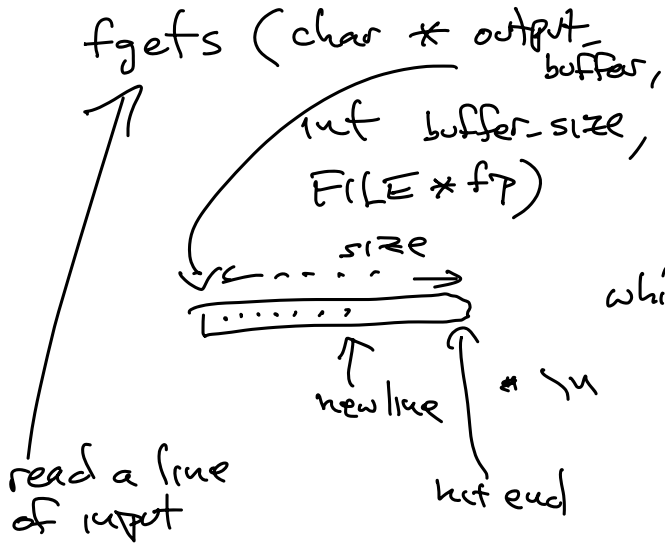
C arrays  $\sum$  `int/long/...`  
`char ci`  
`char s[128];`



name of array  
 $\equiv$  pointer to beginning of array







```
FILE * fp =
  fopen("myfile.txt",
        "r");
char line[256];
```

```
while (fgets(line,
             sizeof(line),
             fp) != NULL)
  printf("%s", line);
```

pattern: caller allocates

← stack allocation

heap allocation:

$\Sigma$  char \*line = malloc(128);

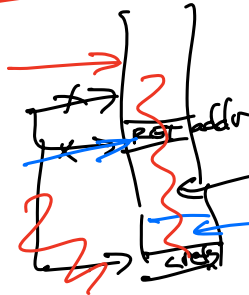
fgets ...

free(line);

explicit alloc

free

f(tree \*a)  
~~x = a~~  
if a == NULL  
return  
else  
f(a->right)  
if a->left == NULL  
...



local:

$\Sigma$  char x[16];

3

locals

arguments

stack