

Lecture 8:

Using Pointers and Hashing

CSE 29: Systems Programming and Software Tools

Aaron Schulman (Shalev)



Pointers let you pass/return data w/o copy

- How do you pass an array to a function and return it?
 - Passing an array C lets you take a `char[]` as input.
 - » **`int func(char[] blah)`**
 - » The compiler just makes `blah` a pointer:
 - » `int func(char* blah)`
 - C does not let you return an array! Why?
 - » `char[] func()` **// NOT POSSIBLE**
 - » Returning an array is inherently dangerous!

Demo



□ Big Idea:

Returning an array is generally a bad idea because you may not be guaranteed some other data will end up in that array.

How to “return” an array in C

1. Pass a pointer to the array (and often its length also) as a variable
2. Modify the array in the function

Example: C's String libraries

```
char* strncpy(char *dst, const char * src, size_t count);
```

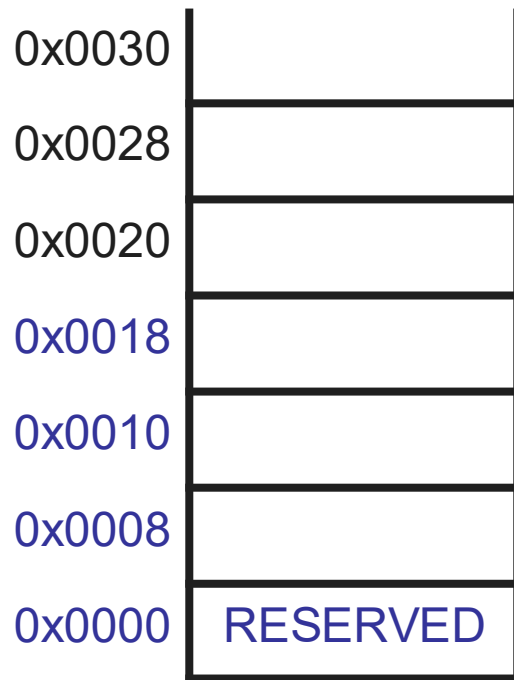
size_t is just a renamed unsigned long (int)

How do you point to “nothing”

```
int *p = NULL;
```

```
// C always has this  
#define NULL 0
```

```
*p = 5;  
SEGMENTATION FAULT!!!!
```



stack
(strachpad memory) 5

Hashing: A way to fingerprint arbitrary bytes of data in memory



Everything stored in the memory of a computer is just arrays of bytes

- » Strings, numbers, images, videos, etc.

If there was a way to produce an integer as a fingerprint any array of bytes we could:

- **Create a unique digital fingerprint for any unique file/data:**

- » A (relatively) small integer can uniquely represent any data
- » We can validate that a file/data has not changed by checking the fingerprint
- » Often used to check if a downloaded file is correct

- **Application: Store passwords securely**

- » Applications can store fingerprints of passwords, instead of passwords
- » These fingerprints can not easily be reversed if the password fingerprint file is compromised

Hash: An integer as a fingerprint for bytes

- Hash: A linear mathematical function that maps bytes to an integer
 - ...a really really big integer: 2^{256} (256-bits)
- Hashes are fingerprints that represent those bytes
 - » However, they are one-way functions
 - » The integer that makes up the fingerprint can not be reversed to find the bytes that made it
 - » This makes them particularly useful for use in password security
 - » They must *uniquely* represent those bytes or they are insecure
- There are many hash functions that you can come up with:
 - MD5 [Rivest] (1992) – 128-bits (broken in 2008)
 - SHA-1 [NSA] (1993) – 160-bits (broken in 2017)
 - SHA-256 [NSA] (2001) – 256-bits (still secure as far as we know)



Review Questions

1. 1. Why can't you return an array from a function in C?
2. 2. What happens if you dereference a pointer to NULL (addr 0x0) in C?
3. 3. If I take a hash of a video I recorded on my phone, then I modify the video with an AI tool, but I only make a minor change that only affects a few bits of the video, will I get a different hash result?