

Lecture 1:

Binary, Printing, and Strings

CSE 29: Systems Programming and Software Tools

Aaron Schulman (Shalev)



Lecture 2 Overview

- ▣ Reminder about binary number representation
- ▣ How to print variables as text
- ▣ How to access strings and arrays in C
- ▣ String functions and how things can go wrong

Week 1 Announcements

- **Announcements:**

- Lab attendance is **required** tomorrow, and a lot happens there.
- Make sure to go to lab
- Waitlist folks: You can do the lab at home until everything shakes out

Course Website / Syllabus

<https://cse29.site>



Converting integers to/from binary arrays

1 bit		2 bits		3 bits	
<u>0</u> = 0		<u>0</u> <u>0</u> = 0		<u>0</u> <u>0</u> <u>0</u> = 0	
<u>1</u> = 1		0 1 = 1		0 0 1 = 1	
2 values: 2^1		1 0 = 2		0 1 0 = 2	
		1 1 = 3		0 1 1 = 3	
		4 values: 2^2		1 0 0 = 4	
				1 0 1 = 5	
				1 1 0 = 6	
				1 1 1 = 7	
				8 values: 2^3	

Why is 'H' or 72 in ASCII the same as bin 01001000?



$$\begin{aligned} 1001000 &= 1 \times 2^6 + 1 \times 2^3 \\ &= 64 + 8 = 72 \end{aligned}$$

Binary (Base 2) is *similar* to Decimal (Base 10) representation:

$$\begin{aligned} 72 &= 7 \times 10^1 + 2 \times 10^0 \\ &= 70 + 2 = 72 \end{aligned}$$

Demo: Accessing Binary Data in C

- Programs can interpret and present bits (in arrays) **in many ways:**
 - ◆ They can be **Integers**: `printf("%d", arr[i])`
 - » `%d` (signed) `%u` (unsigned)
 - ◆ They can be **Characters**: `printf("%c", arr[i])`
 - » `%c` (signed)
 - ◆ They can be **Strings** of Characters with NULL terminators : `printf("%s", arr)`
 - ◆ There are many other ways `printf()` can interpret arrays of bits too!
- **C gives us direct access to the bit representations of data in memory!**

Demo: How do strings work in C?

- Big Idea:

- ◆ Strings are just arrays of characters
- ◆ Characters are just integers
- ◆ The string is terminated when there is a null character at the end

What is a string, an array of characters!

`char sup[7] = "Hello";`

0	1	2	3	4	5	6
'H'	'e'	'l'	'l'	'o'	' '	'\0'

Special char.
NUL Character
End of string = 0

Demo: What if we forget the NULL Char?



- Big Idea:
 - ◆ There are no training wheels anymore in C, this is not Java
 - ◆ If you tell the computer to do something, it will do exactly what you say.

What happens if the null is not there?

```
char sup[7] = "Hello";  
char hi[2] = {'H', 'i'};
```

0	1	0	1	2	3	4	5	6
'H'	'i'	'H'	'e'	'l'	'l'	'o'	' '	'\0'

← **hi** → ← **sup** →



Lecture Review Questions

1. What is the value written in decimal of the 8-bit binary numbers 0000 and 1101?
2. What is 33 as an 8-bit binary number?
3. `printf("___ __", 70, 70);` Fill in the blanks to make this print F 70