

Wed Jan 18

Declare function before calling!!

Declare, also called a function prototype: before used, declare as many times (same exact thing tho), have to be declared at beginning or in header file

Void function (int x);

Define: only once

Void function(int x) {

 Sfjvbjks

 Kjsbd

 Return void

}

Return type (use void if void) function name(parameters-use void if void){

 //declarations (declared vars are only local)

 //assignment statements (sets/resets the value of a var)

}

Return values

- Try not to return struct
- Can't return array

Parameters

- Call by value: the value (scalar or pointer) of the parameter is passed, no effect on the actual value of the variable
- Formal: name as used in function
- Actual: the name passed when calling
- Pointers kinda act like call by reference, the actual value is changed

Swap only really works if pointers are used

#include preprocessor, basically pastes the code in

#define macro, replaces the defined things prior to compilation

Conditional directives,

- define, ifdef, endif - execute only if it is defined
- ifndef, endif - execute if not defined

Header files .h

- Doesn't have actual code
- Has function declarations and macro defs
- Structs, globals, includes
- Header guard
- ifndef or pragma once

Standard header files (look up list)

Extern is like global

Fri Jan 20 - Numbers

representation (many ways to write the same number)

We use base 10

Kinds of numbers

- Natural numbers

- Integers
- Rational numbers
- Real numbers
- Complex

Computers use fixed-width ints

- Depends on the bits
- 2^x bits

Positional number systems

- Different bases for different representations of the same number

Ascii is 7 bits, 8 would make it negative

#include <stdint.h> basically always use these other than loops

- Signed int8_t to int64_t
- Unsigned uint8_t to uint64_t
- Sizes 8 to 64

Twos complement

Computers can't represent real numbers, can represent some of them

Real is infinite

Floating point numbers are an approximation, never use equality with these

- Float, single precision, IEEE 754 single, 32 bits
- Double, double precision, IEEE 754 double, 64 bits
- Long double, extended precision, IEEE 754 quad, 128 bits

big/little endian

- Little - least significant byte is low address
- Big - most significant byte is low address