

ssh - secure shell

Git → track files

- can go back to earlier versions

Type vimtutor



into terminal

Readme

- what it does
- how to build
- how to run
- errors

Binary arithmetic

Bits (manip.)

Addresses

Pointers

Kollected

Code book

Vicious

Latech

for drawing
pics

↓
on
overleaf

git add design.pdf to computer

git pull to VM

emacs

vi

VScode / Sublime Text



text editor

GOOGLE

Markdown

syntax

`<stdio.h>` → standard I/O package
`printf` → print, 0 → success (return)
`cc -o hello hello.c` ← compile (clang)
`./hello` ← run, 11 → comments
% 3.0%f 6.1f\n ← look in MAN printf EOF
use for loop when you can
Need to declare type of var and char
initialize (char → small ints) ''
Variables of same name cause problems
can read from one file to another

International Obfuscated C Competition

#define CONSTANT number
processor when you put CONSTANT, number
will be in that place

Python → interpreter (slower)
C → compiler → asm → machine code
Use clang to compile

Xkcd
↑
OMICS

Makefile → has rules on how to compile

Make sure code works on Linux
SOMEHOW

SIGN UP FOR TUTORING ON TUTORTRAC !!!

Booleans:

AND	\wedge	conjunction	$\&\&$	inclusive in C
OR	\vee	disjunction	$\ $	
NOT	\neg	negation	!	
XOR	\oplus	$A \oplus B = (A \vee B) \wedge \neg(A \wedge B)$		
		$A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$		

- one or the other, not both

$$A \oplus A = 0, A \oplus 0 = A, A \oplus 1 = \neg A$$

XOR	T	F
T	F	T
F	T	F

CHECK

- nonzero
 \Rightarrow TRUE

0 = FALSE (only thing that is)

- logical expressions have type int

scanf("%d", &n)

Short circuit eval. \rightarrow when result of bool expr. is known, evaluation will stop

Switch():

switch(___) {

case 0:

use break
after statement

case 1:

- goto could also be used

DON'T

} default:

Loops:

while()

- top test loops (test evaluated before body)
- evaluated 0 or more times

@ 9:35 in lecture asgn2

tip for

for()

- top test loop
- parts → init., test, increment
- i = control variable (disappears after execution)
- init. and test don't need same variable

do {

- execute at least once

} while()

use for pig if u can

- bottom-test loop

switch statements
are good for
rolling dice

infinite loops

- to end, use break

Ex: while(1), for(;;)

- could use goto for major errors

= assignment, == equality

Continue : like break, but it goes back to the beginning of the loop

LOOK MORE INTO SWITCH

LOOK UP
CALL BY
NAME

FUNCTIONS:

void

- may or may not return a value
- defined once / declared & call ≥ 1 times
- support functions before main
- main() run when program starts

return type func-name (parameters) {
 // declarations / assignment
}

- return type can be void / anything except array
- no parameters = VOID
- declared variables inside func. = local
- use snake case \rightarrow (mm-mm) lowercase

Parameters:

- call-by-value \rightarrow take value of parameter, make a copy of x, and give it to the function (x is not modified @ end)

formal: name of par. as used in func. body
actual: ^{supplied / passed} value copied to formal parameters

LOOK BACK @ CALL BY REFERENCE SLIDE

GO BACK TO  PART

call by reference → can be emulated using
pointers (addresses are passed as arguments)
function prototype:

return type func-name (parameters);

- declaration of function needed if
functions not defined before main

Macros in C use text replacement

#include → pastes code of given file ^{into current file}

#define → defines macro for program ^{replace}

#indefn / #out ↡ C00K

Headers:

- contain func. declarations / macros / etc.
- ends in .h

extern: write def later

static: variable persists across function calls
only visible inside that file

utility functions: Func. that do small things

Recursion: take factoring
code & run it

HEADERS
10:08

MAKE COPY OF PUBLIC
SSH KEY and PUT IN FILE