C Programming: Pointers, Arrays, Memory

Yipeng Huang

Rutgers University

January 28, 2021

Table of contents

Programming assignment

Programming assignment

- Due in 14 days: 11:59pm Thursday, February 11.
- Use Piazza to ask and respond to questions.
- By end of today, you will have everything you need for at least part 1, goldbach, part 2, maximum, and part 3, matMul

Chat box

- Great for everyone to participate.
- Something that would be useful even outside of the online classroom.
- Be respectful.
- Help me monitor for any questions that are going unanswered.

Recap of Tuesday: Stuff we missed.

- header files
- `int fscanf(FILE *stream, const char *format, ...)`

Table of contents

git pull

From the folder 2021_0s_211, type: `git pull`

Lesson 1: What are pointers?

- Pointers are numbers
- The unary operator & gives the “address of a variable”.
- how big is a pointer? 32-bit or 64-bit machine?
- Pointers are typed

Lesson 2: Dereferencing pointers with *

`*pointer`: dereferencing operator: variable in that address

`int* ptr` and `int *ptr`

No difference between `int* ptr` and `int *ptr`

- `int* ptr` emphasizes that `ptr` is `int*` type
- `int *ptr` emphasizes that when you dereference `ptr`, you get a variable of type `int`

Lesson 3: The integer datatype uses four bytes

- Memory is an array of addressable bytes
- Variables are simply names for contiguous sequences of bytes

Lesson 4: Printing each byte of an integer

- Most significant byte (MSB) first $\rightarrow$ big endian
- Least significant byte (LSB) first $\rightarrow$ little endian

Which one is true for the ilab machine?

Lesson 5: Pointers are just variables that live in memory

- Pointers to pointer

Lesson 6: Arrays are just places in memory

- name of array points to first element
- `malloc()` and `free()`
- stack and heap
- using pointers instead of arrays
- pointer arithmetic
- `char* argv[]` and `char** argv` are the same thing

Lesson 7: Passing-by-value

- C functions are entirely pass-by-value

Lesson 8: Passing-by-reference

- You can create the illusion of pass-by-reference by passing pointers

Lesson 9: Passing an array leads to passing-by-reference

Table of contents