

Hello, systems; Hello, C

COSC 208, Introduction to Computer Systems, 2021-08-30

Announcements

- Before next class: read DiS sections and answer **individually** pre-class questions
- Is there any volunteer to switch from 208LC to LB?

Outline

- Syllabus
- Warm-up: Hello, system
- Hello, C

Syllabus

- *Questions*
- *Remarks*
 - advantage 2D layout: facilitate skimming
 - pre-questions: You need to earn $\geq 80\%$ of the available points to get full credit. Trust you are doing right thing to learn.
 - project: in pair by default
 - participation also includes *listening*, we share a space

Warm-up

- Q1: *What are the main components of a computer system?*
 - Operating system (OS)
 - Central processing unit (CPU)
 - Random access memory (RAM)
 - Secondary storage devices — e.g., hard disk drive, solid state drive
 - Input/output (I/O) ports
- Q1b: *What is the role of each of the components?*
 - OS — manages hardware resources
 - CPU — executes assembly instructions
 - RAM — stores program data (code, variables, etc.) at runtime
 - Secondary storage — stores program data (code, documents, etc.) persistently
 - I/O ports — receives user input (e.g., keyboard, mouse, audio); provides output to users (e.g., text, graphics, audio)

- Q2: *What do you think of when you hear the term "Computer Systems"?*
 - C
 - Number representation
 - Program memory
 - Architecture
 - Assembly
 - Efficiency
 - Multiprocessing
 - Networking
- Q3: *Why is it important to learn about computer systems?*
 - To write more efficient programs
 - To leverage parallelism

Hello, C

Primitive types

- Most common types
 - `char` — 1 byte (8-bits); integers -128...127; integers 0...127 correspond to ASCII characters
 - `int` — usually 4 bytes (32-bits); integers -2 billion...2 billion
 - `long` — usually 8 bytes (64-bits); integers -9 quintillion...9 quintillion
- Add `unsigned` in front of these types for a range from 0 to 255, 4 billion, or 18 quintillion, respectively

Operators

- *How would I increment the number stored in a variable `x` by 1?*
 - `x = x + 1`
 - `x += 1`
 - `x++`
 - `++x`
- Q4: *What is the output of this program?*

```
int main() {
    int x = 1;
    int y = 2;
    x = x+5;
    printf("%d ", x);
    x = y*2;
    printf("%d ", x);
    x *= 5;
    printf("%d ", x);
    printf("%d ", x--);
    printf("%d ", x);
    printf("%d ", --x);
}
```

```
    printf("%d", x);  
}
```

```
6 4 20 20 19 18 18
```

Demo

- Write a C program that prints "Hello, C!"

```
// Visual Studio Code demo  
#include <stdio.h>  
#include <stdlib.h>  
int main() {  
    printf("Hello, C!\n");  
    return EXIT_SUCCESS;  
}
```

- *How do you compile the program?*

```
$ clang -g -Wall -o hello hello.c
```

- *How do you run the program?*

```
$ ./hello
```

More practice

- Q5: *What is the output of this program?*

```
int main() {  
    int x = 5;  
    int y = x/2;  
    int z = x%2;  
    printf("%d %d\n", y, z);  
}
```

```
2 1
```

- Q6: *What is the output of this program?*

```
int main() {  
    int x = 5;  
    char y = 'F';  
    y = y - x;  
    printf("%c %d\n", y, y);  
}
```

A 65

Output (if time)

- What is the syntax for `printf`?
 - `printf(FORMAT_STRING, VALUES, ...);`
 - `FORMAT_STRING` is a string constant (sequence of characters surrounded by double quotes) that may optionally include format specifiers
 - Format specifiers define how to convert a value to a string
 - `%d` decimal (i.e., base 10) number
 - `%c` character
 - `%x` hexadecimal (i.e., base 16) number
 - `%s` string — more on this next week
 - After the format string, include a value for each format specifier
 - A compile error will occur if the number of format specifiers does not match the number of values
 - A compile warning will occur if the value type does not match the format specifier
- Next week we'll talk about reading input using `fgets`
 - *Dive into Systems* uses `scanf`, which has several disadvantages

Extra practice

- Q5: Write a program that computes and displays the number of minutes in a year.

```
int main() {  
    int minutes = 365 * 24 * 60;  
    printf("%d minutes\n", minutes);  
    return EXIT_SUCCESS;  
}
```