ENGG1003 - Monday Week 12

The C programming language & version control with Git

Brenton Schulz and Steve Weller

University of Newcastle

24 May 2021

Last compiled: May 23, 2021 4:41pm +10:00

Lecture overview

1. Context
   - ENGG1003
   - what is C?
   - do we even need C?
2. C programming language
   - features and philosophy
   - key language details of C
3. version control with Git
   - principles
   - practical demonstration (live demo by Brenton)

1) Context

- Recall from last week...
- $\leq$ 2020, ENGG1003 used *MATLAB* and *C*
  - **from 2021, ENGG1003 uses Python only**
  - ...yet some students will use MATLAB &/or C in later courses
- Thursday week 11: overview of MATLAB
- today's lecture: overview of C

What is C?

- C is a general-purpose, high-level programming language
  - other high-level languages: Python, MATLAB, Java, C++, FORTRAN, ...
- C has been around since early-1970s and still *very* popular
  - IEEE language ranking 2020: #1: Python **#3: C** #10: MATLAB
- C is native language of Linux, Microsoft Windows, OS X, iOS, & Android kernels
- even Python language is written in C
- **C language constructs map efficiently to computer hardware (machine instructions)**

Do we even need C?

- **C is not assessable in ENGG1003**
- BUT... C is currently used in some courses in some Engineering programs:
  - Aerospace Systems, Computer Systems, Electrical, Mechatronics and Medical Engineering
- key courses which use C language:
  - ELEC2720, ELEC3730, AERO3600, MCHA3400, MCHA3500
  - common theme of all these courses is *embedded systems*

Embedded systems

- embedded system: coupled hardware and software designed for a *specific task*
- eg: washing machine
  - inputs: buttons, water-level, water temperature
  - output: LED display, motor control
  - control unit: microprocessor & memory
- *many* other examples: smart TVs, gaming consoles, medical devices, WiFi routers, automotive ...
- **C language is well-suited to tight coupling of hardware and software in embedded systems**

2) C programming language

Quick recap of *computer language hierarchy*, week 1:

- **machine code**
  - a microprocessor can only understand machine code
  - eg: one instruction for an x86-based processor:
    0110 0110 1000 0011 1100 0000 0000 1010
  - very processor-specific!
- **assembly language**
  - human-readable abbreviations ("mnemonics")
  - eg: machine code
    0110 0110 1000 0011 1100 0000 0000 1010
  - same instruction in assembly language
    `ADD AX, 10`
  - very processor-specific

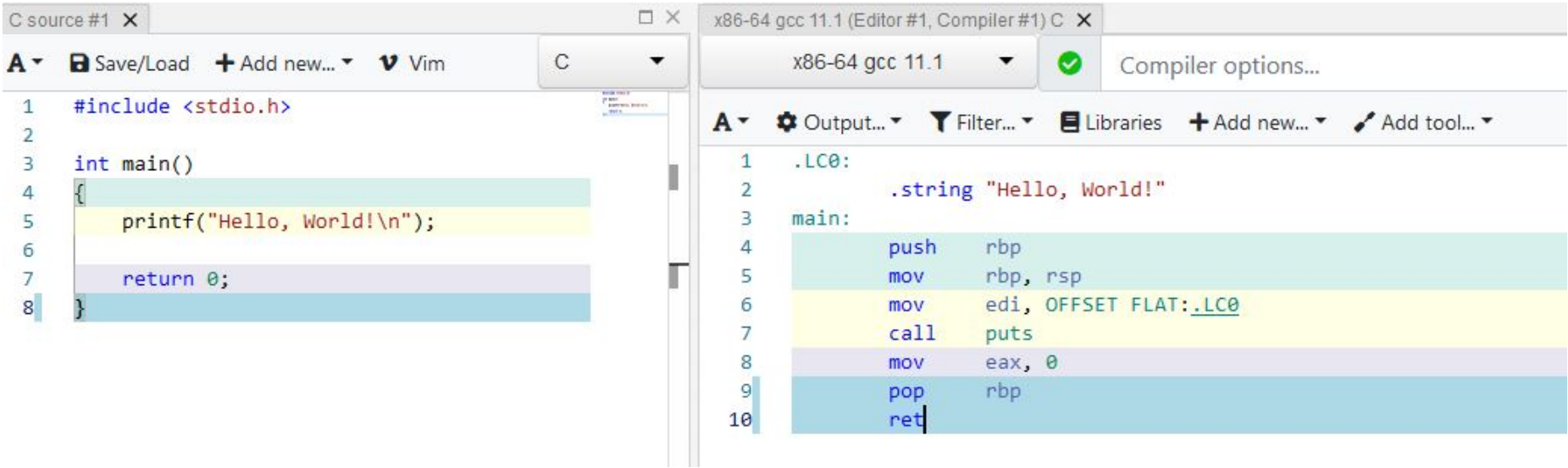
- **high-level languages** eg: C, Python, MATLAB ...
  - human-readable text-based code
  - increased complexity of each high-level instruction
  - eg: machine code
    0110 0110 1000 0011 1100 0000 0000 1010
  - same instruction in assembly language
    ```
    ADD AX, 10
    ```
  - same instruction in C
    ```
    x = x + 10;
    ```
  - *not* processor-specific (that's a good thing)
- *compilation process* converts source code (eg: C) to machine code
  - source code → pre-processor → compiler → assembler → linker → machine code ("executable")

Hello, World! . . . in the C language

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, World!\n");
6     return 0;
7 }
```

- line 1: header file `stdio.h`, like Python `import`
- line 3: function definition for `main()`, returns `int`
- line 5: formatted print, `\n` prints "newline"
- line 6: returns "all good" (error-free) message
- lines 4 & 7: $\{\cdot\}$ begin/end of function `main()`

Hello, World! program in C and assembly



- left panel: C program
- right panel: corresponding assembly language code for 64-bit Intel x86
  - x86 instructions: `push`, `mov`, `call`, `pop`, `ret`
- try it yourself at: https://godbolt.org/

The C language combines all the power of assembly language with all the ease-of-use of assembly language.

— Mark Pearce —

AZ QUOTES

Learn to think like the computer hates you, because it does.

https://learncodethehardway.org/c/

gdb online

- screenshot of gdb https://www.onlinegdb.com/
- link to code blocks
- embedded systems IDEs in other courses

C: key language details

- syntax
- data types
- arithmetic and relational operators
- flow control
- functions
- arrays

3) Version control with Git

- principles—SRW
- live demo of practice—Brenton

Next steps

C is not assessable in ENGG1003

- getting started with C, if you need it *for later courses*

- Code::Blocks integrated development environment
  - https://www.codeblocks.org/downloads/binaries/
  - runs on various platforms:
    - Windows XP / Vista / 7 / 8.x / 10
    - Linux 32- and 64-bit
    - Mac OS X
- OnlineGDB