

Computer Systems & Low-Level Programming

Introduction to Linux and C

Marija Stanojevic
Spring 2019

TA info

Teaching Assistant: Marija Stanojevic

Office: Science Education and Research Center (SERC) 334

Office hours: Tuesday, 3-4 pm and Wednesday, 10-11 am

URL: <https://astro.temple.edu/~tug73611/> (google: Marija Stanojevic Temple)

Email: marija.stanojevic@temple.edu

Lab info

- Lab attendance is obligatory and noted every time
- ASK QUESTIONS - there is not stupid question
- Lab is time for assignments, quizzes and questions
- I'll cover common mistakes and some more examples in labs
- Cheating is not tolerated!
- For any assignment I might ask you to explain me your code in person, during the lab or office hours.
- Labs will be interactive. You'll write code in front of the whole class on voluntary basis. Mistakes are celebrated, they help us learn!
- If you have issue with Quiz/Assignment grade, you can complain within 7 days from the moment grade is posted.

Software

- There are many IDEs for C, most used are (install one):
 - NetBeans
 - Eclipse
 - CodeLite IDE
- What is GCC?
- GCC (GNU Compiler Collection)
 - collection of compilers for C/C++/Fortran/Ada/Go/... originally written for free GNU OS
- How to install latest version of GCC (Linux/Mac): <https://gcc.gnu.org/install/>
- GCC manual: <https://gcc.gnu.org/onlinedocs/gcc-8.2.0/gcc.pdf>
- How to check current version of GCC?
 - gcc --version
- Online C compiler: https://www.onlinegdb.com/online_c_compiler

Short history of C

- Developed by Dennis Ritchie and Ken Thompson in 1973 from B language
- First standard in 1989 (ANSI C) and 1990 (ISO) - same => C89
- Standard update in 1995 => C95
- Standard update in 1999 (ISO) => C99
- Standard update in 2011 => C11
- Standard update in 2017/18 => C18

Differences from Java

- How many of you had a Java course?
- How many of you had C course?
- Java vs C compiling process:
 - Java compiles code into bytecode that is read by JVM, bytecode doesn't depend on OS
 - C/C++ compiles directly into object and machine language, so it depends on OS
 - C/C++ code that works on one machine might not work with another
 - All variables have to be defined on top of the code
- C is lower language, closer to the assembler and hardware, faster, smaller
- Stuff that Java does automatically, you have to do in C by yourself
 - No garbage collector, you have to delete stuff from memory by yourself
 - Can't manipulate whole array, sets or lists at once, just item by item
 - You have to assign memory to data yourself
 - Data and its address are accessed in different ways
- C/C++ is basis of many things: Python, Python packages, JVM, Linux

Linux overview (1)

- **Open CMD on Windows (terminal in Linux/MacOS) and login to server**
- **ls -la** - lists files and folders in your current folder (including hidden)
- **pwd** - shows path to the current folder
- **cd folderName** - enters into folder named folderName
- **cd ../..** - goes into grandparent of current folder
- **nano fileName** - opens textual files; if it doesn't exist, it is created
- CTRL + X closes file opened with nano
- **mkdir & rmdir** - makes/removes directory
- **rm -r folderName** - deletes file/folder name with all children

Linux overview (2)

- `touch new.txt` - creates file with name new.txt
- `man & --help` - if you need help with commands
- `cp file.txt newfile.txt` - copies file.txt into newfile.txt
- `mv file.txt newfile.txt` - renames/moves file.txt into newfile.txt
- `locate file.txt` - finds file.txt
- `sudo command` - does the command as administrator
- `zip, unzip` - zips and unzips data
- Browse more commands: <http://man.openbsd.org>
- **Transfer data to/from server:**
 - From windows: use Putty, WinSCP (you need to install those)
 - From linux/mac: use Filezilla program (you need to install Filezilla)

Hello MyName!

```
// hello.c
#include <stdio.h>
int main() {
    char name[50];
    scanf("%s", name);
    printf("Hello %s! \n", name);
    return 0;
}
// function main that returns type int
// string is an array of chars
// scans your name from input
// \n is new line
// we finished program and everything is fine
```

- **gcc hello.c** => produces a.out; or **gcc hello -o hello.out hello.c** => produces hello.out
- **./a.out** or **./hello.out** => writes "Hello MyName!"
- **%c** - character, **%s** - string; **%d** - integer; **%f** - float/double; **%e** - scientific notation; **%o** - octal number; **%u** - unsigned integer; **%x** - unsigned hexadecimal **%p** - pointer (address); **%1.3f** - 1 is minimal number of positions that is used to write number and 3 is number of decimal places

- **Windows users have more complicated installation procedures for C:**
 - Install GCC called MinGW from [here](#).
 - After installation set-up Environment Variable Path to add path to the MinGW. Full tutorial for installation is [here](#).
 - To use Eclipse for C, [install in this order](#) (not recommended).
 - To use NetBeans for C, make sure you've installed MinGW, Java and set up Environment Variable Path before you install NetBeans for C. Download Netbeans for C/C++ from [here](#).
- If you are using any other operating system, you already have GCC, just install the editor (Eclipse, NetBeans, CodeBlock,...)