

Week 1 - Basic Configuration

Hongzheng Chen

Nov 15, 2019

This seminar is partly based on the seminar about tools in applied mathematics held in PKU¹.

1 Text Editors

Comparisons among different text editors can be found here. Based on the Stack Overflow's 2018 Developer Survey, the top 5 code editors are listed below:

1. Visual Studio Code: It is very young (1st version in mid-2015) but must be the text editor in the new era with many advanced text editor techniques. Open source and managed by Microsoft means VS Code has more resources and attracts more developers to devote to it, which leads to one of the most popular projects on Github. Support to keybindings of other text editors also makes more users from different societies. Moreover, highlighting and auto completion of some minor languages (e.g. Verilog, Prolog) gains its popularity. The biggest disadvantage may be its speed due to the Electron-based backend. New features can be found in this Chinese blog.
2. Sublime Text: It is light-weight and extremely fast. Many language support can also be found in Preference - Package Control, and it is easy to configure the build system of different languages. The keybinding is very user-friendly. Though it also has large community, it is close-source which somehow hinders its development.
3. Atom: Best known as its hackability and seamless Github integration.
4. Vim: For experts who prefer keyboard and shortcuts over UI. I think it is very hard to use. But you have to get used to it if you want to operate on a Linux server.
5. Notepad++: Lightweight and basic for Windows, but somehow old-fashion.

An editor suitable for you is the best. So do not hesitate, pick one, and just have a try.

2 Regular Expression (Regex)

Many text editor have support to regular expression, which is very commonly used in searching and replacing. Also, some programming languages have standard library of regular expression, including C++ 11's regex and Python 3's re, which we will cover in the following courses.

Some tutorials on regex include:

¹ToolsSeminar, <https://github.com/pppppass/ToolsSeminar>

- Regex 30 min blitz
- Learn Regex: The Easy Way

The best way to learn regex is practicing on some cases, you can make online test on [Regex 101](#). Write your example, and try to get out the pattern.

3 Linux Environment

It is highly suggested to use a Unix-like environment even if you use a Windows system. Reasons include:

- Some packages either do not work well or are hard to configure in Windows, especially large deep learning packages TensorFlow and PyTorch. On the contrary, there are thorough tutorials on configuration in Ubuntu (and maybe macOS).
- Tool chains are broken up in Windows. For example, Git, Make, Python, gdb, perf are initially installed in Linux (most of which will be covered in the following courses), while they must be manually installed in Windows and sometimes meet lots of dependency issues.
- Many computing packages are developed on Linux with little support to Windows. For example, it is very troublesome to configure large deep learning packages like TensorFlow and PyTorch on Windows, and you may meet many strange problems which even cannot find solutions on Internet. However, if you use Linux or macOS, you can easily install them and find solutions to some issues on Github or Stack Overflow.
- Getting accustomed to shells and terminals benefits because you may login to remote servers at some point. Some examinations and competitions may also only provide Linux environment.

There are several ways to have a Linux machine I recommend.

- **Install the Windows Subsystem for Linux (WSL).** It is effortless and fully compatible with your Windows computer. Installation Guide can be found on the official site. WSL2 will be released in 2020, you can install the preview version following this guide. To configure WSL in VS Code, instructions can be found [here](#).
- **Install a virtual machine.** It is also commonly used and will be very useful when you develop your own operating system (core course in the 2nd grade).

For a virtual machine, VirtualBox is recommended, which is a free virtual machine program by Oracle. Note that VMWare is another famous virtual machine software,

but there are license issues. Additionally, VirtualBox provides better support for Linux systems, while VMWare is better for Windows systems or heavy tasks.

Downloads of VirtualBox can be found [this page](#). Documentation can be found [here](#), where a [.pdf User Manual](#) is provided.

For Linux distributions, Ubuntu for desktops is a ready-to-use operating system for new-comers to Linux. There are also easy-to-understand instructions during installation on the website. Ubuntu 18.04 LTS is the most-recently stable one.

- **Buy a cloud server.** Many cloud server providers have education discount, which give students a first try to do some works on Linux. Some famous foreign cloud server providers include Amazon's AWS, Microsoft's Azure, and Google's Google Cloud. In China, Alibaba's Aliyun and Tencent's TencentCloud are two of the most popular cloud providers.

Tencent Cloud provides student discount of 10 y/m, details can be found [here](#). Aliyun also provides a discount of 9.5 y/m, products can be found [here](#). Many foreign cloud providers may have free accounts for the first year. If you have your own server, you can make some fantastic things like building your personal webpage (WordPress), configuring your own git server for collaboration, and having your own net disk (Seafile).

As an alternative, we may provide you accounts of the server in our lab after you get familiar with the Linux environment.

For Linux and Mac OS users, nothing is needed but just opening a terminal.

A Chinese introduction to Linux command lines is shown [here](#). NIH also provides a course Introduction to Linux to tell basic GNU and Linux concepts. The first 32 pages is adequate for a start in Linux.

SSH, Vim and Make, which are three important tools, are always included in Unix-like environments. If not, you may install it through your package manager (`apt-get` in Ubuntu and `yum` in CentOS).

To login a remote server from a client, you can use some SSH login software, like XShell and PuTTY. To enable remote login on the server, a SSH key is needed to be generated. This is described in Xuefeng Liao's Git tutorial in the section Remote repositories, as well as in *Git official tutorial* in section 4.3 *Git on the Server — Generating Your SSH Public Key*. If you want to permit remote logins into your system, a SSH service is also required, which is described [here](#).

SSH is also useful to transfer the message generated by WSL to your Windows system. Combining with Xming, you can run graphical programs on WSL. Instructions can be found [here](#).

A famous post in Stack Overflow describes how to exit the Vim editor, which puzzles thousands of sophisticated programmers. Complete introduction to Vim are given by this post and this website.

Make is an automatic build tool, which turns complicated build commands into a simple command `make`. Tutorials on Make can be easily found on the Internet, among which Yifeng Ruan's *Tutorial on Make* and *Make (software)* on Wikipedia give useful information. But you need not get through them too quickly, we will detailedly cover Make in the later courses.

4 Git & Github

Xuefeng Liao's Git Tutorial is recommended for beginners in Git and GitHub.

Git officially offers the book Pro Git, which is also a great tutorial for Git and GitHub. Note that `.pdf` format and Chinese version are also provided. This tutorial covers more topics than Xuefeng Liao's tutorial, and therefore first 5 sections are enough.

GitHub itself provides a training kit for Git and GitHub, in which one may refer to On Demand Train and Learning Path. The corresponding Git repository is this one.

Further information can be found in Git's official document, where a cheat sheet of Git commands are provided.

GitHub itself provides a brief introduction to GitHub. Note that GitHub Guides covers more topics which you may find helpful at some point.

Further information about `.gitignore` can be found in Xuefeng Liao's Git Tutorial and Git's reference. GitHub provides a repository for several `.gitignore` templates.

Difference among Licenses can be found here. About commit messages and change log, you can find templates in this website. To write a good `README.md` document, you can look through this template. The usage of Markdown will be covered in next course.

Basically, if you have installed Git on your computer, VS Code may detect that. Otherwise, you can look through this page for help.

5 Code Style

Please refer to Google Style Guide for further information, but you need not strictly conform it. What you need to remember is to format your code in a same way. For convenience, you can directly use the formatting function in the text editor.

6 Assignments

6.1 Before Your Work

Please make the prerequisite first.

1. Install a text editor suitable to you.
2. Have a Linux machine no matter using which approach.
3. Register a Github account.

All the materials and slides can be obtained by

```
$ git clone --recursive https://github.com/chhzh123/ToolsSeminar-CS.git
```

6.2 Your Work

Make sure you have cloned the repository, and hack the question below.

Question 1 (Username). A website constrains its user name as follows,

- Only lower case, upper case, underline, and hyphen are allowed.
- Only 5 ~ 20 characters are allowed.
- The first character should be lower case.

Now you have a list of user names, try to find **the number of valid user names**. No methods or tools are constrained, but be clever and efficient.

To obtain the list of names, follow the command below in the `ToolsSeminar-CS` folder. (Linux environment is needed.)

```
$ cd Assignments/BasicConfiguration
$ make
```

Then a file called `name.txt` is generated. You should use this file as the problem input. The input guarantees each name (valid/invalid) begins with `(#)`, where `#` is a number.

* This work is a naive attempt of data cleaning, which is an important step in big data processing. Most of the viable data in real life are complex strings, so you need to structure and filter out the useful data first, after which you can do the analysis. Actually the data obtained from Internet and from your experiments are much bigger and more irregular than this example. But since this is the first assignment, only toy example is given here (((

6.3 After Your Work

After finishing your work,

1. Create your personal public repository to store your assignment. (No README, no license for this time.)
2. Upload your repository to Github.
3. Commit your first work to the repository. (Only a text file briefly describing how you do or a piece of code should be uploaded.)

Remember:

- Structurally manage your repository. For example, you can create a folder each week under your repo, and store the corresponding assignment in the folder.
- Do not upload unnecessary files including the generated file and your intermediate files. You should write `.gitignore` file!
- Do not forget to write commit messages. Briefly state what you have done in this commit.
- Basically, you CANNOT remove the history of git, so be careful when you commit.