ACM CS Survival Guide

Krerkkiat Chusap

Jan 21, 2019

Contents:

This is a quick guide for computer science student. The guide will go over the basic concept in computer science, mostly regrading the tools, and tricks that can be used to develope a software.

This guide is created by Ohio University's Chapter of Association for Computing Machinery (ACM). We meet every Wednesday 7:30pm at ARC 315. We also host study session on every Thursday from 6pm to 8pm at ARC 315.

1 Introduction

The idea for this project began when many junior and senior level computer science students were seen using nano for large projects. Even though nano can be powerful and useful in some aspects, students should experiment with the tools they use when writing code to learn what works best for them. There are quite many editor out there that may facilitate the software development remotely greatly. For example, micro, and not to mention vim and emacs.

The idea for this guide then began to form on the idea of introducing the more powerful editors to first year computer science students. It has since grown into a full presentation - "CS Survival Guide"[1]. After suggestions from previous undergrads, the guide now has more information on the software, tools, and tricks that the graduates found useful or learned along the way.

We hope that you find this guide to be beneficial to you to as you begin your computer science major at Ohio University.

[1] https://docs.google.com/presentation/d/1iybf_B2hZ4G_OO1sBR0F6j50CJuYcPNC31zsP491o8o/edit?usp=sharing

2 School Servers

2.1 How many are there?

The school has multiple servers available for EECS students to connect to. These servers make it possible for you to work on homework in your dorms or at home without needing access to a Linux distro on your computer. That said, sometimes it is useful to go to the labs to avoid distractions.

There is a list of the servers available at this page[2]. Just click on the "EECS Machine and services status pages". Your browser will prompt you to sign in, but you can just leave both fields blank. When you "logged in", click on "Hosts" on the sidebar to get the list of all the servers.

This list contains some other servers as well. The most useful servers are prefixed with any of the following.

- `odd` These are the Ubuntu machines in the computer lab.
- `sp-` These are more of the Ubuntu machines in the computer lab.
- `p` These are the old Solaris machines. You will only rarely use these.
- `pu` They are Ubuntu machines that has the same setup as the lab workstations.
- `tesla2` This is a special Ubuntu machine with CUDA that can be used for machine learning.

Since all of these are with the EECS department the full address will be something similar to the following:

- `odd15.cs.ohio.edu`
- `sp-003.cs.ohio.edu`
- `p1.cs.ohio.edu`
- `pu1.cs.ohio.edu`
- `tesla2.cs.ohio.edu`

These servers all keep your home folder on a network filesystem, so all your files will be accessible on any of them (including the Solaris servers).

As a small note, there is a `tesla1` machine, however students do not have access to it.

[2] http://ace.cs.ohio.edu/

2.2 How to connect to one?

On Linux or MacOS you can open the terminal emulator, and type

```
$ ssh odd15.cs.ohio.edu
```

On Windows, the easiest way is to use puTTY. You can save your sessions with it to quickly connect to various servers as needed.

2.3 File Transferring

There are serveral options to transfer the file from your laptop/pc to the servers.

2.3.1 WinSCP

This is a file transferring tool that is part of the puTTY project. It will have access to your saved puTTY sessions, so you do not need to create new sessions for this tool.

2.3.2 FileZilla

A popular free to use file transferring tool.

2.3.3 Cyberduck

Cyberduck[3] is a file transferring tool that also has support for other cloud storage services.

2.3.4 `scp` command

This is a common command that should be avaiable on most Unix-like systems. This command can overwrite files, so be careful with it. More information can be found using `man scp` command.

The following example is to transfer file from local machine to the server.

```
$ scp ./project1.cc bobcat@odd15.cs.ohio.edu:/home/bobcat/project1.cc
```

The following example is to transfer a folder (recursively) from remote machine to the local machine.

```
$ scp -r bobcat@odd15.cs.ohio.edu:/home/bobcat/cs2400/project1/ ./project1/
```

2.4 Note on other servers

If you have access to servers other than the school servers, all of the aforementioned information is also applicable.

Just use the hostname or public IP address of that server instead of the school servers.

[3] https://cyberduck.io/

3 Basic Unix Commands

3.1 The `man` command

This `man` command allows you to read the manual pages for a command. The manual page contains the overview description of the command, its command line arguments, and some example.

For example to get the manual page for `ls` command you can run,

```
$ man ls
```

The manual pages can sometimes be hard to read, but they are available in every Linux or Unix-like machines. However, you can also refer to Devhint[4] which is a quick reference for tools, commands, and topics in computer science.

Many commands also have a help option that will give you some information. For example, with `ls` you can run

```
$ ls --help
```

The exact syntax will vary by command if available at all.

3.2 `Ctrl+c` vs `Ctrl+z`

`Ctrl+z` will suspend the program you are running (z like Zzz). On the other hand `Ctrl+c` will stop the program you are running (e.g. program that stuck in infinite loop).

If you use `Ctrl+z` a lot and don't kill or restart the processes, you will end up with a lot of processes just sitting there.

To list the program that in the background, `jobs` can be used. To restart a suspended process in the foreground, you can use the `fg` command. `bg` will restart a process in the background.

Starting the command with '&' at the end will start the job in the background from start. Be careful though since output will still show up on the shell, so be sure to do some proper redirection of the outputs.

[4] https://devhints.io

3.3 What about `Ctrl+s` and `Ctrl+q`?

If you accidentally hit `Ctrl+s`, what you will experience is the terminal is frozen from any other inputs that you are trying to type in. To resolve that hit `Ctrl+q` to unfreeze it.

For deeper detail and history, please see this stack exchange question[5].

3.4 `ps` and `kill` commands

This follow from the previous section where if you found yourself in the situation where many processes are suspended in the background.

3.5 Input and output redirection

This will be useful if the project or assignment has the input test cases, so you can just use the input redirection, so you do not have to type in the input every time.

3.6 Directory Listing

To view the contents of the directory you are in, run `ls`. This is the most basic form of the command.

`ls -l`, shows more details, such as permissions size and the last time the files were edited. This will be sorted by name.

```
$ ls -l
total 19472
-rw------- 1 krerkkiat krerkkiat  463097 Apr 19  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪1 (1).pptx
-rw------- 1 krerkkiat krerkkiat  463097 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪1.pptx
-rw------- 1 krerkkiat krerkkiat 1460689 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪2.pptx
-rw------- 1 krerkkiat krerkkiat 1271326 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪3.pptx
-rw------- 1 krerkkiat krerkkiat  984855 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪4.pptx
-rw------- 1 krerkkiat krerkkiat  882673 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪1 (1).pptx
-rw------- 1 krerkkiat krerkkiat  882673 Apr 12  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪1.pptx
-rw------- 1 krerkkiat krerkkiat  955386 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪2(1).pptx
-rw------- 1 krerkkiat krerkkiat  984164 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪3.pptx
drwx------ 2 krerkkiat krerkkiat    4096 Apr  5  2017 circuit2 exam
drwx------ 2 krerkkiat krerkkiat    4096 Apr  5  2017 Content
-rw------- 1 krerkkiat krerkkiat 3614295 Apr 23  2017 EE2114 Final Review.pptx
-rw------- 1 krerkkiat krerkkiat   22045 Jan 26  2017 EE2114 Syllabus - Spring 2017.
↪docx
```

(continues on next page)

[5] https://unix.stackexchange.com/questions/12107/how-to-unfreeze-after-accidentally-pressing-ctrl-s-in-a-terminal

(continued from previous page)

```
drwx------ 6 krerkkiat krerkkiat    4096 Mar 19  2017 Labs
-rw------- 1 krerkkiat krerkkiat 7074816 Mar 24  2017 Midterm2-Review.ppt
```

However, you can add `-rt` to it to sort the result by time and show the recent files on the bottom.

```
$ ls -lrt
total 19472
-rw------- 1 krerkkiat krerkkiat   22045 Jan 26  2017 EE2114 Syllabus - Spring 2017.
↪docx
drwx------ 6 krerkkiat krerkkiat    4096 Mar 19  2017 Labs
-rw------- 1 krerkkiat krerkkiat 7074816 Mar 24  2017 Midterm2-Review.ppt
drwx------ 2 krerkkiat krerkkiat    4096 Apr  5  2017 circuit2 exam
drwx------ 2 krerkkiat krerkkiat    4096 Apr  5  2017 Content
-rw------- 1 krerkkiat krerkkiat  463097 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪1.pptx
-rw------- 1 krerkkiat krerkkiat 1460689 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪2.pptx
-rw------- 1 krerkkiat krerkkiat 1271326 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪3.pptx
-rw------- 1 krerkkiat krerkkiat  984855 Apr 12  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪4.pptx
-rw------- 1 krerkkiat krerkkiat  882673 Apr 12  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪1.pptx
-rw------- 1 krerkkiat krerkkiat  984164 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪3.pptx
-rw------- 1 krerkkiat krerkkiat  955386 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪2(1).pptx
-rw------- 1 krerkkiat krerkkiat  882673 Apr 19  2017 Ch16_PPT_Fund_Elec_Circ_6e-Day␣
↪1 (1).pptx
-rw------- 1 krerkkiat krerkkiat  463097 Apr 19  2017 Ch15_PPT_Fund_Elec_Circ_6e-Day␣
↪1 (1).pptx
-rw------- 1 krerkkiat krerkkiat 3614295 Apr 23  2017 EE2114 Final Review.pptx
```

If you use `-a` it will show hidden files (these are files that start with a '.').

3.7 Downloading a file

`wget` or `curl` can be used to download a file to the remote server that you are connected to.

4

Editors

Here are some well known editors.

4.1 Command line

These will be come in handy if you are perform quick editing on the server. You will usually be told to use `nano`, but you also have the following option.

- **`nano`** A simple lightweight text editor.
- **`micro`** A simple lightweight text editor written in Go. It is focused on being intuitive. More information can be found at https://micro-editor.github.io/. `micro` is not installed by default on the school servers, but you can quickly download the prebuilt binary and run it as follow.

```
$ wget https://github.com/zyedidia/micro/releases/download/v1.4.1/micro-1.4.1-
↪linux64.tar.gz
$ tar -xf micro-1.4.1-linux64.tar.gz
$ ./micro-1.4.1/micro
```

- **`emacs`** This editor is a little more complex to use at first, but is significantly more powerful than the two above.
- **`vim`** Like `emacs` this editor is one of the more complicated to learn, but is also very powerful. If you want to get a quick start in customizing you Vim editor, Vim Bootstrap[6] is a really good place to start.

4.2 GUI

- **VS Code**[7] Microsoft's cross-platform text editor. It has a set of extensions[8] that allow users to remotely collaborate on the project.
- **Atom**[9] Popular cross-platform text editor. This editor also have functionality that is similar to the live collaboration of VS Code. More information can be found here[10].

[6] https://vim-bootstrap.com/

[7] https://code.visualstudio.com/

[8] https://visualstudio.microsoft.com/services/live-share/

[9] https://atom.io/

[10] https://teletype.atom.io/

- **Sublime Text**[11] Popular cross-platform text editor.
- **Notepad++**[12] Popular free text editor for Windows.
- **Geany**[13] It is the light-weight editor.
- **Gedit** This should be installed on the workstation.

All of these text editors (except for Notepad++) are available in Linux repositories.

4.3 Tips and Tricks

One quick note on line numbers. Most of editors will turn on line numbers by default for you. However Gedit does not, but you can turn them on via `Edit->Preferences`. On `vim` you can use the `:set number` command to show line numbers or `:set nonumber` to turn them off.

If you ever accidentally start `vim` you can quit without saving by typing `ESC:q!`.

If you ever accidentally start `emacs` you can quit without saving by typing `Ctrl-x Ctrl-c`

If you write code on Windows and then copy the code to a Linux system, you will find the '^M' character at the end of every line. To remove these characters, you can run `sed -i 's/\r//g filename'` in the directory with the file. To prevent this from happening, you can go into your editor's settings and change the line ending type from Windows to UNIX.

[11] https://www.sublimetext.com/
[12] https://notepad-plus-plus.org/
[13] https://www.geany.org/

5 Version Control System (Git)

There are various version control system, but this will only focus on Git[14].

5.1 Basic Workflow

5.2 Team Workflow

5.3 Useful Commands

5.4 GUI Frontend

[14] https://git-scm.com/

6 Running other OSes

Many of you have Windows or macOS installed, but will want access to Linux on your own personal systems for development. There are two major options, using a virtual machine, or dual-booting.

6.1 Virtual Machine

A virtual machine allows your computer to run a second operating system inside of your primary OS. Most virtual machine offerings will not perform as well as having Linux installed natively, but many people feel more comfortable doing this than dual booting. There are many options, so try out a few of them and see which one works best for you. Of the following four options, all but one will work on any host OS.

6.1.1 VirtualBox

A popular and free "general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use".

6.1.2 QEMU

A free as in speech emulator. It is capable of emulating multiple architecture types, including x86, x86_64, and MIPS. It does not come with a GUI, but there are some that are available for it.

6.1.3 Hyper-V

A hypervisor built into Windows Professional, Education, and Enterprise editions. It performs very well if you have access to it.

6.1.4 VMware

A free (but not open source) virtualizer for x86 hardware.

6.2 Dual Boot

Some of the previous graduates found that they preferred to having Linux run natively for better performance, or because virtualization just did not work well on their systems due to old hardware. Systems with a dual boot set up can boot more than one OS. When set up properly a Linux boot loader will appear when turning on the computer and allows the user to pick between Windows and Linux, or macOS and Linux at boot time.

Modern Linux distros such as Ubuntu and Fedora include installers capable of setting up your system to have more than one OS installed. They will automatically shrink your Windows partition, and create new partitions on your hard drive that will contain Linux. Try out a LiveUSB and make sure your hardware will work with Linux before you install Linux.

7

More resources

These are some more information.

- **Learn X in Y minutes**[15] is a pretty thorough introduction to the programming languages.
- **GeeksforGeeks**[16] is comprehensive guide on multiple topics in computer science. For example, programming languages like C or C++, Algorithms, Data structures.
- **Awesome list**[17] is a curated list of lists of awesome resources.

[15] https://learnxinyminutes.com/
[16] https://www.geeksforgeeks.org/
[17] https://github.com/sindresorhus/awesome

8 Contributors

This guide is started by Krerkkiat Chusap, so all the blame should go to him (especially if you found yourself maintaining this guide). However, all compliments should go to all the people that contribute to this guide either directly or indirectly, and whether their names are mentioned on this page or not.

I (Krerkkiat Chusap) would like to specially thank the following people.

- **Dr. Razvan Bunescu, Associate Professor, Advisor** for being an advisor for the chapter.
- **Nasseef Abukamail, Associate Lecturer** for the help in attracting audience for the presentation.
- **Samuel Merten, Graduate advisor** for fully support the idea and its execution.
- **Current (2018-2019) officers of OU ACM** Nathan Steadman (President), Aubree Dix (Vice President), Rory Dunne (Treasurer), Anirudh Sharma (Secretary).

  Without these officers, I would not have time to play around with this idea.
- **Austin Crabtree** for the first round of external (from myself) ideas and presentation suggestion.
- **Timothy Steinberger** for the second round of external ideas, and for heavily revise, and proof read the early version of the guide. Also, he add a lot of content into the guide at its early stage.
- **Colton Smith** for the rehearsal of the presentation. His friendly conversation also keep me active in the chapter.

Other contributors that are not listed here can be found at GitHub repository[18].

[18] https://github.com/krerkkiat/acm-cs-survival-guide/graphs/contributors

9 About

9.1 About This Guide

The creation of this guide is inspired by these documents[19], especially this one[20] and this one[21], that Mr. Tysko has for the new users of the EECS machines.

The guide was almost be written in LaTeX which may be really hard to maintain. The thank should go to Raymond Hettinger (@raymondh[22]) from saving the guide to be written in LaTeX with one of his talk[23] at PyCon 2018 (Cleveland).

9.2 About Us

We are Ohio University's Chapter of Association for Computing Machinery (ACM). We meet every Wednesday 7:30pm at ARC 315. We also host study session on every Thursday from 6pm to 8pm at ARC 315.

- Orgsync Page[24]
- Facebook Group[25]

[19] http://ace.cs.ohio.edu/
[20] http://ace.cs.ohio.edu/eecshelp.html
[21] http://ace.cs.ohio.edu/new_users/index.html
[22] https://twitter.com/raymondh
[23] https://youtu.be/T-TwcmT6Rcw
[24] https://orgsync.com/112927/chapter
[25] https://www.facebook.com/groups/128726397182128/