List of useful resources for students of computer-related majors

M. Yas. Davoodeh

November 25, 2020

Contents

1 Notes

Starred items are those which I am not personally comfortable with but I suggest them to starters, for ease of use or access.

All the softwares in this list are free! Free as in freedom. FOSS!

For installing applications named like `this` in Linux just give the name (`this` in this example) to your Package Manager:

For example to install `this` example in Ubuntu (and other Debian distros like Pop!-OS) run:

```
sudo apt install this
```

For installing `this` in Manjaro (and other Arch-based distros) run:

```
sudo pacman -Sy this
```

1.1 Terminology

- Noob: you.
- GUI: Graphical User Interface. Clicky clicky icons.
- CLI: Command Line Interface. Basically typing and letter focused interfaces.
- FOSS/Libre: Free (as in Freedom) and open-source software. Software that you can easily download, manipulate, redistribute without getting sued or spied on.
- Cross-platform: Software that can be on multiple platforms (Windows, Linux, Androids, etc).

2 Tools

2.1 Must haves

Even if you don't use Linux use the tools named here since your other (smarter) classmates are probably using these and will hate you if you make their day harder with using some unprofessional application or format.

There is no debate. It's doctor's orders.

- **Linux** as your host operating system

  Pros:

  - It can run Windows applications and games.
  - It's free (no money).
  - It's free (as in freedom) and open-source (FOSS).
  - It's fine tuned for students and professionals in computer field.
  - Most of academic tools are mainly for Linux.
  - Installing applications in it is usually as easy as pressing a button and it will be downloaded and installed automatically.
  - It's lightweight and everything is under YOUR control. You can delete, install, customize and basically do ANYTHING. It just takes knowledge.
    - Yes, as mentioned before, it includes running and installing Windows applications and games.
  - Support is excellent and it can be installed on older and less powerful machines.
  - A thriving community out there to support you online and offline.

    If you don't know Linux is independent from its visuals so any version (distribution or distro for short), can come with any looks! I found KDE Plasma desktop so easy to use and so similar to Windows. Manjaro and Ubuntu are 2 very easy to setup and maintain distros. I suggest Manjaro since you can install ANY applications in it with little effort. However if you are unlucky you may get that one-in-a-million case where installing the applications is not always as easy as in Ubuntu and you wish you had Manjaro. Therefore, I suggest installing Manjaro KDE or Kubuntu (KDE on Ubuntu) for starters.

- *Eclipse: A great starter IDE for most languages (Later replaced by your IDE of choice Emacs)
- Git (CLI version) (`git`): It's a must have. Sooner or later you will have to learn it. Do **NOT** install a GUI like the one GitHub suggests you unless you want to regret it, tell people shamefully that you don't know Git as you should and have your family and friends to leave you. Pro Git book is a great starting point. The Farsi translation can be found on my repository (not merged with the site site).

- *OnlyOffice or Libreoffice (`libreoffice` or `libreoffice-fresh`): Awesome FOSS office suits. OnlyOffice has a better support and user interface comparing to Libreoffice. **Even if you still use Microsoft Office get the habit of saving your files as `.odt` not `.doc` or `.docx`.** It will be easier to share them without breaking them and all applications support it. (Later your office suit will render useless and replaced by LaTeX, your TeX editor and other applications/Emacs plugins.)

- LaTeX: LaTeX (*latek* is used more often comparing to *lateks*) is a typesetting program. Basically will let you make any sort of paper or shape/graph without hassle or difficulties. LaTeX will make typing math *super* easy. A LaTeX export of this file is made in the same folder to demonstrate its clarity and ease of use. It literally took typing one word in Emacs to make that PDF. Much like Vim and Git due to its vast uses everywhere, you will, probably, eventually, have to learn it. Example of LaTeX math (rendered correctly in the PDF version):

$$f(x) = \sum_{i=0}^{\infty} \int_{b}^{a} \frac{1}{3} x_i^3$$

2.2 Advanced

- Vim: Vim is a minimal text editor. If learned correctly it enables you the power of gods in editing. You can edit as fast as you think. The learning curve is not the smoothest tho. Just like LaTeX and Git, Vim key's and workflow are implemented in countless applications (like Emacs and Eclipse). It's wiser to learn it faster before you have to.

- Emacs (DOOM Emacs): The only application you will need for doing anything professionally. There is much depth to it and takes a lot of time to explain. For now think of it as an IDE with a lot of plugins for everything you can think of. Since you learned Vim at this stage and you cannot accept anything inferior. You will want to use Doom Emacs which is a configuration of Doom for noobs in Emacs and Vimers. Emacs pretty much supports EVERY language, has excellent support and is itself programmable.

2.3 Misc

2.3.1 Personal choices

Most Linux newbies feel rather confused once they see the *countless* number of software available. Here are my personal choices

- Firefox (`firefox`): FOSS browser

- VLC (`vlc`) or MPV (`mpv`): FOSS media player. MPV is more minimal comparing to VLC.

- Aria2 (`aria2`) with or without Persepolis (`persepolis`): Simply the best CLI download manager. Persepolis is a modern, cross-platform FOSS GUI for Aria2

- *Kate (`kate`) and *Geany (`geany`): Best FOSS text editors for your pre-Vim/Emacs primitive life

- *Okular (`okular`) or Zathura: Okular is everything you want from a reader in one package. An overkill like other KDE applications. On the other hand Zathura is a sleek Vim-like reader. (Later noting features of Okular will render useless comparing to Org-mode's Org-noter mode in Emacs)

- *Octave (`octave`): For when you grew tired of the basic BC. FOSS programming language for doing your math homeworks. (Some prefer Python; Either ways both of them will later be replaced by Emacs Calc-mode)

- Tableplus (`tableplus`): FOSS GUI

- *JupyterLab (`jupyterlab`): A multi-language GUI literate-programming tool for having your code along your explanations just like a notebook. (Later replace by Org-mode in Emacs or Emacs' iPython-mode)

- GIMP (`gimp`): Easy to use, FOSS Image manipulation tool. It's lightweight and has plenty of tutorials.

- Zsh (`zsh`): An actual useful, convenient, normal shell featuring one of the best shell management frameworks, Oh My Zsh!

- Draw.io: Perfect easy, on-the-fly graph/figure generator. It's not a replacement for LaTeX, PlantUML or GIMP but super easy to use.

- Cmder: Helps you feel in ease using CLI in Windows till you get your Linux installed.

- DWM (or any tiling window manager): If you don't know what's a window manager it's a good sign that it's too soon to install one.

2.3.2 Suggested languages

There are plenty of languages out there. Thousands and thousands. Which one to learn? These are ones that you will need 100% no matter what you do.

All items in the list are here because you *will* use these languages sooner or later, they are just simply superior or they have countless children and dialects. It's true, other languages like Php or Java are super popular and professionally useful but there are paths in which an student does not necessarily need to know them. It's best to delay the effort until you are 100% sure you want/have to know them. Besides, once you know C, it's easy to learn most other languages in a week or two. With all being said, if you wait long enough, everything will be replaced by a version of JS, C, Python or Lisp. (it's me calling it, the end of 2020)

- C: Yes, I'm sorry. C. Big old daddy. Always the top language. Since most of the programs and other languages are written in C (like Python), you must know the basics at the very least.

- Python: Fun language (for the first 3 months). Has plenty of packages, community support and code pre-written.

- JavaScript: Unfortunately it's everywhere. You need it because you don't want to learn 900 languages to do different things with.

- Shell: The language of Linux. If your answer to any of the questions ahead is yes, you need to learn this: Want to write programs and don't 10000 hours experience in Java? Want to work remote connect to another system? Work with your computer without a monitor? Want everything automated? Want your desktop to look like /r/unixporn?

- Lisp: Essential for configuring your Emacs, suitable for writing fast code and driving yourself crazy. (It's basically the coke of the languages. Nothing else will ever hit harder. You WILL come back.)

- LaTeX: I promise you, you don't want to draw graphs or write math without LaTeX ever again. No need to mention managing bibliographies or Table of Contents. Software like Org-mode will help you not to learn LaTeX but for fine tuning and practical use it's necessary to do so.

- PlantUML: Save time and get extra credit from teachers. Although your graphs usually just add to confusion instead of taking it away, for some reason they are so credible in teachers opinion whether they help or not.

3 Sites to register in

- SoloLearn: A powerful cross-platform application for practicing and communicating

- GitHub: A place to share your codes (owned by Microsoft tho and if you are Iranian your account will shut down anytime soon so good luck).

- GitLab: Like GitHub (but already shut down Iranian accounts)

- YouTube: Google's video platform
- Medium: A lot of useful computer articles. You can use the account on sites like TowardsDataScience.
- Discord: Family friendly version of IRC. Join any server and ask noobish questions about anything
- Reddit: A hub for any community. Noob friendly and great for asking questions or finding resources.
- *Trello: Manage your life (Later replaced by your planner of choice Org-mode in Emacs)

4 Practices

4.1 Typing

If you type a source code with numbers and symbols in it (something like C++) in less than 100WPM then you need to practice typing. If your type speed in such text is less than your handwritten speed you should *seriously* consider practicing before entering the uni.

4.1.1 Websites

- TypingClub: Zero-to-hero typing course. Try starting from level 1 and focus on achieving "silver stars" in every level. I suggest not to go ahead if you cannot achieve silver stars.
- KeyBR: Best to go site for practicing your weaknesses and learning the hard way
- 10FastFingers.com: Time to test your skills. Also you can practice there to be one of those cheap typists who only learn typical words and struggle with numbers for their whole life.

4.1.2 Software

- Klavaro: Pick up every language or layout fast and easily with Klavaro. Maybe not the best for practicing, but for learning new layout outs it's the best. It's also a FOSS cross-platform application.
- RapidTyping: The middle ground. Perfect for beginners, extensive practices and countless layouts but not the best in any of the fields individually. Freeware - Windows only (can be run on Linux via Wine)

5 Youtube channels

5.1 Educational

These Youtube channels are serious, educational and have awesome quality.

- 3Blue1Brown: The best math channel with animations and clear explanation
- MIT OpenCourseWare: Have you watched your SICP today?
- sentdex: One man army of all tutorials
- TED-Ed: Lessons worth hearing
- Stefan Mischook: That one nice old teacher in school who gives real life tips on the major.
- Andy Balaam: Cool humble tutorials – Scheme course is top-notch
- Zach Star: I also thought I don't get math.
- Computerphile: More hardcore subjects on computer theory

- Numberphile: Answering questions you never thought existed
- The Linux Experiment: High quality serious news, discussions and educational videos on Linux, freedom and open-source
- DistroTube: Learn about Linux, freedom and productivity. Yes, they lied to you all along
- Luke Smith: Boomer makes you love freedom and get back to life then makes you hate it instantly.
- LearnLinuxTV: Dedicated to Linux and tools that will make your life easier

5.2 Random videos on topic

These educational channels are less structured and more broad or general. Awesome for taking ideas or finding answers to your obscure questions.

- Code Bullet: Get interested in AI with project on games and random applications nobody asked for.
- The Coding Train: Sober Code Bullet actually teaching stuff
- Jabrils: Does cool things with AI
- carykh: Random computer projects
- Siraj Raval: Indian Jabrils

5.3 Random randoms

Channels in this group are mostly “commercial” or not educational. The creators of this group just put up their own content (again, great for taking ideas or insights).

- TechLead: Fun sublimes for nerds
- The Verge: Commercial break for nerds. Talks about what phone not to buy next.
- LGR: Oldschool vibes about gaming and computers.
- DIY Perks: Thinker and create with computers (and other than computers)
- the Hacksmith: Yes, engineering can be cool

5.4 Game related

The channels below are directed toward Video Games and Games. Some of them are a bit casual but most of them are also technically valuable and educational.

- AI and Games: Technical talks about AI and Games
- Raycevick: Philosophical and in-depth talks about games and gaming
- Indigo Gaming: Raycevick with much more budget focusing on documentaries
- Game Maker’s Toolkit: Talks in depth about making games both technically and other ways.
- kliksphilip: kliksphilip talking about life, Youtube, being productive and philosophy even some juggling.
- 2kliksphilip: 2kliksphilip focuses more on gaming and computers in general.
- 3kliksphilip: 3kliksphilip is for VALVe employees to take notes on how to make their games better.
- CodeParade: A freelancer programmer/game-developer

- ThinMatrix: A freelancer game-developer talking about development and being productive
- Hopson: Random computer/game-development videos
- videogamedunkey: The ultimate game critic
- Girlfriend Reviews: Game reviews for starters

5.5 These are here just because I like them to be

Other channels that I follow. Not strictly educational in computers majors but still educational. All about CSGO, languages and technology. I particularly love CSGO because it gives you a one-in-a-kind in game developer console, has extensive mechanics and a lot of unique features. Awesome for learning how to play games, work in a team and experiment with software both at the same time.

- TheWarOwl: The guy who teached you how to play CSGO talks about life, competition, games and CSGO
- Langfocus: Your way to go if you are interested in languages or NLP
- Pro Guides CSGO Tips, Tricks and Guides
- SlavTek: Cheeki breeki tek channel of Boris
- BananaGaming: Learn how to config CSGO and get notified about updates.
- HOUNGOUNGAGNE: How to be humble and good at the same time (especially in CSGO)

6 Discord servers

- Doom Emacs: Noob friendly lounge for all sort of related (sometimes unrelated) topics

7 Books

7.1 Absolute noob level

- *The study of Language* by George Yule: Best all-in-one introduction book to (mostly natural) languages.

7.2 Beginner

- *An Introduction to Formal Languages and Automata* by Linz Peter: One in a kind practice book for your Language (with capital L) knowledge.
- The MIT SICP (*Structure and Interpretation of Computer Programs*) by Harold Abelson and Gerald Jay Sussman with Julie Sussman: An awesome free of charge, noob friendly tour on computer science and programs in Lisp. Video course is on MIT OpenCourseWare.
- *Artificial Intelligence: A Modern Approach* by Stuart Russell and Peter Norvig: A good starting point for theory of AI. At least it's fun in history chapter.
- *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* by Aurélien Géron: Suggested starting point of AI