

# Roadmap Overview

## Preface

This roadmap is a personal project that I'm working on as a way to document my journey learning about the world of systems and programming. While I have done some self-studying and know a few things about systems, I'm nothing more than a jack of all trades with shallow knowledge. But all of the knowledge and experience I've gained over the past 3 years has led me to creating this roadmap that I plan to follow.

When I first started to self-study programming, I knew nothing about what was out there, I knew that I was interested in software engineering and robotics, but I hadn't the slightest idea of how to get there. Now I know the path I need to follow to get there, but first I wanted to pursue my interests in Systems, Networking and General Programming.

That's when I realized that this was the perfect time to master the fundamentals starting all the way from System Administration with Linux to covering Computer Networking then Cloud Technologies and finally diving right back into Computer Science.

I am sharing this roadmap of mine as a way to showcase my learning journey starting from Systems & Networking to Computer Engineering and I hope that others who read this may find something that they can take away for themselves.

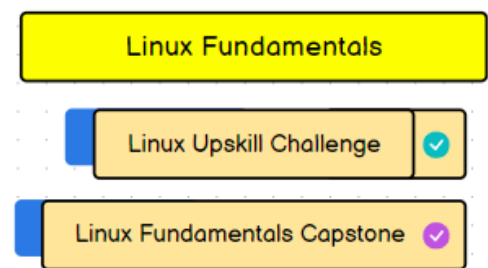
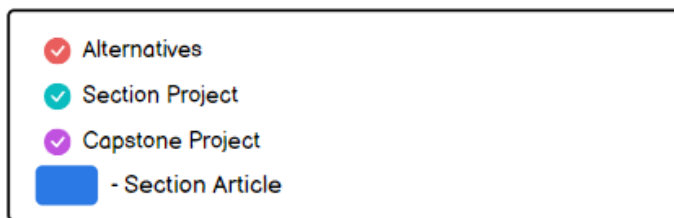
## The Main Topics

The roadmap, as of now, consists of 5 main topics with each major category containing their own sub-topic.

- **Systems & Networking Fundamentals**
- **Intro to Programming**
- **Containers & Orchestration**
- **Intro to Computer Science**
- **System Programming**

The intended flow of the roadmap is to go through each major category and work on every sub-topic then do the associated projects for each section. All of this culminates in a final capstone project that will put your understanding of the category to the absolute test.

## roadmap.sh explanation



The philosophy of this roadmap is taking the long-game approach to studying. A big mistake we often make in learning is that we will cover a subject just enough to scrape by. While you don't have to know everything about a subject to be able to do interesting things, when it comes to our foundations true understanding is essential.

So, to remedy this, we will cover each major topic in just enough depth to get a strong grasp of the basics. The **blue button** behind every sub-topic will link to the full article in the Lab-Book. Meanwhile the actual **sub-topic** will cover a brief summary of the course, video, or book we will be working through. It will also include relevant resources. While not pictured here, the **Alternatives** section will provide other courses/videos that work as a roughly equal substitute, but they may also both be taken.

The best way to truly cement one's knowledge on a topic is by doing, and so the **turquoise-colored checkmark** ☑ next to the sub-topic will be where we begin our section project work. Lastly, we arrive at the **Capstone Project** ☑, this is a section dedicated to putting all of the knowledge we have learned to the test. The Capstone Project will either be a single large project or a series of projects that puts everything we have learned together. Lastly while not pictured, there will be a **dark-blue checkmark** next to courses/books that are optional but may be worthwhile depending on your use-case.

## Personal Note

Throughout this journey I will do my best to be as objective as I can. However, I am still in the middle of my learning and I am only human. While technology itself is neutral, the way we use it, how we think about it, what we think is best moving forward and so on, are things that are human viewpoints. There are of course times when things are simply 0 or 1, but as you go through your own journey you will find that you will have to come up with your own definitions and reasons for doing things backed by personal preference or well researched analysis or simply that's the way things are done.

So, to carry on with that spirit I will place the **sparkling star icon** ✨ whenever certain notes or topics are things I am speaking my own mind on. I encourage you to come up with your own analysis or thoughts on those topics.

# The Roadmap

In **The Roadmap** below. All of the time estimates are based on studying the material for 2-3 hours per day.

### **Legend**

-  **Alternatives** | -  **Optional**

## **System & Networking Fundamentals**

A system can be defined as a set of parts working together, and a network as a group or system of interconnected things.

The aim of this path is to familiarize ourselves with Linux & Computer networking. We'll learn to both configure and administer our very own Linux Servers, write powerful scripts, and come away with a sys-admin's eye and toolset. We'll also gain a solid foundation in computer networking, uncovering the Blackbox that the internet often seems to be, learning about various protocols, topologies, and architectures.

At the end of this path, we'll be able to call ourselves a jr sysadmin and jr network-engineer.

## **Linux Fundamentals**

- **( Online Course ) Linux Upskill Challenge**
  - The Linux Upskill Challenge in my eyes is the perfect introduction to Linux.
  - It is a month-long course that will get you up-to speed with Linux and give you enough skills to start managing your own servers as well as give you the tools you need to set up services like WordPress or your own Wikis like this bookstack instance. It also has references to more in-depth material if you find that the base challenge is too easy.
  - **Estimated Time** : 21 days - 1 month
- **( Book - O'Reilly ) Learning the UNIX OS**
  - A short and sweet book on the Unix OS by O'Reilly. The book may be over 20 years old but there's still a lot of value to be gained.
  - This book will give you a good overview of the Unix OS and by extension Linux, as well as providing some interesting history lessons. After covering this book, I became a lot more comfortable working with Linux, and it has led to some great quality of life improvements.
  - **Estimated Time** : 7-10 days
- **( Online Course ) Linux Journey**
  - We will be wrapping up our Introduction to Linux with Linux Journey. While we should be pretty comfortable with Linux by now, there are still quite a few things left to cover.
  - Things like Networking, Processes Monitoring, The Kernel, and other important things to know. Once we finish going through this, we will be ready to start taking our first steps towards Linux System Administration.
  - **Estimated Time** : 7-14 days

- ( **Book - O'Reilly** ) **Learning the bash Shell**

- Now that we are familiar with Linux and have started managing our own servers and home-lab we're ready to begin taking full advantage of shell scripting. In essence shell scripting is about automating repetitive tasks: These can range from backing up files to monitoring logs & system resources, updating devices, and managing accounts.
- If done right, scripting can save you many hours in the long-run and you may start to create a collection of scripts that you'll end up using for years.
- **Estimated Time** : 15 days

- ( **Book [ ]** ) **UNIX and Linux System Administration Handbook**

- Now that we are comfortable with the Linux environment and have shell scripting under our toolbelt we are finally ready to take on the world of Linux System Administration.
- This book will serve as a guide to understanding how Linux System Administration works. The main things we will be covering are System Administration, Networking, Storage & Ops.
- **Estimated Time** : n/a

- ( **Book [ ]** ) **The Debian Administrator's Handbook**

- I'm a big fan of using Ubuntu & Ubuntu Server for my projects and homelab. The Debian Admins handbook works as a free alternative to the handbook mentioned above.
- **Estimated Time** : n/a

- **Linux Fundamentals Capstone Project**

- **WIP**
- **Estimated Time** : n/a

## Networking Fundamentals

- ( **Online Course - FreeCodeCamp** ) **Intro to Computer Networks**

- Intro to Computer Networks is less hands-on than the other previous courses. The reason for this is that the CCNA will cover a lot of new material and will require a lot of labbing throughout the course.
- We will be going through FreeCodeCamp's Computer Networking course as well as some articles that cover the basics of computer networking.
- **Estimated Time** : n/a

- ( **Books & Online Courses** ) **Cisco CCNA**

- We now begin the CCNA. For a lot of people this will be one of the more difficult things to get through, at least in regards to IT.
- This is where we will learn about routing and switching, wireless networking, and security. Which will be followed up by a small tour of what's possible with automation and programming.

- In summary, the CCNA is a true introduction to Computer Networking. However the world of Computer Networking is incredibly vast. If Intro to Computer Networks was a foreword then the CCNA can be seen as a prologue.
- **Estimated Time** : n/a
- **( O'Reilly ) Network Warrior**
  - Now that we've gone through the CCNA, we're ready to round out our foundational networking knowledge with O'Reilly's Network Warrior.
  - Network Warrior focuses on covering the other half that the CCNA overlooks. And that's the focus on the fundamentals with practical applications. Instead of going through hypothetical scenarios, we'll be taking a look at real networks you would find out in the field.
  - **Estimated Time** : n/a
- **( No Starch Press ) Practical Packet Analysis**
  - At this point we should have a strong understanding of how networking works. and have now started to unravel the veil that is computer networking.
  - But there's one major thing we're missing, we know how to troubleshoot, configure & design networks. But do we really understand what sort of data is flowing through our network?
  - When we connect to a site what sort of traffic is generated, who's generating it? What requests are being made & who's receiving them? It's finally time to learn what's really going on in our networks.
  - **Estimated Time** : n/a
- **( Helsinki MOOC ) Core 5G and Beyond**
  - Initially I was going to consider this portion optional but when you considering that over 55% of internet traffic comes from mobile devices, cellular networks are not something that can just be ignored.
  - This course from Helsinki gives a general overview of how mobile networks work, everything from LTE to 5G and beyond. This course will be mostly theoretical in nature but there will be some limited hands on labs.
  - **Estimated Time** : n/a
- **Networking Fundamentals Capstone Project**
  - **WIP**
  - **Estimated Time** : n/a

## Intro to Programming & Automation

- **Definition of Programming** - "Programming is the process of writing instructions for a computer to execute"
- **The Python Philosophy** - "Simple is better than complex"

This is by no means an introduction to computer science, that will be something to cover later. Instead, this path is all about easing us into computer programming. We'll be learning about the

absolute basics of programming and some automation as well. We'll also cover Databases learning about CRUD and Data-In/Data-Out, then finally rounding things out with version control. By the end of this path we'll have taken our first steps in programming. There's still a long way to go before we can call ourselves software developers much less engineers, but bit-by-bit we are getting there.

## Intro To Programming

- ( **Helsinki MOOC** ) **Python Programming**
  - "Simple is better than complex" - Python is an all-purpose programming language that focuses on being both simple and readable. It's a powerful tool used everywhere from Web Development to Automation and System Administration, to Data-Science & Scientific Computing the list goes on.
  - This simplicity and power are the reasons why we're taking our first steps into programming with Python.
  - The Helsinki MOOC will get us acquainted with the necessary basics, serving as a foundation for the future.
  - **Estimated Time** : n/a
- ( **No Starch Press** ) **Automate the Boring Stuff With Python**
  - Going through this book is entirely optional, but if you often find yourself doing repetitive tasks: whether it be at your job, doing personal projects or anything else, I would say it's a book that's worth your time.
  - **Estimated Time:** n/a
- ( **Books & Online Courses** ) **Intro to MySQL**
  - Learning how to effectively work with databases is important, much of software engineering involves CRUD and data-in/data-out operations. That's not even considering the importance of effectively collecting and reading data.
  - In this book and course, we'll learn how to structure data within a database and come away with enough knowledge to understand and work with data-oriented tools.
  - **Estimated Time:** n/a
- ( **Articles & Online Courses** ) **Git**
  - Git is a version control system. Version control tracks any changes made to a code base, so a version control system tracks changes to a file or set of files over time.
  - Since Version Control Systems keep track of every change made to a code base, if any mistake is made in a new version, we're able to roll back to a previous working version. This allows us to experiment with new changes and configurations without fear of disrupting a system entirely.
  - This makes git a powerful tool for managing codebases, scripts, configuration files and more.
  - **Estimated Time:** n/a
- **Intro to Programming Capstone Project**
  - **WIP**

# Containers & Orchestration

## Docker

- [Hypervisors](#)
- [Containers](#)
- [DevOps with Docker](#)
- [Docker Capstone Project](#)

## Kubernetes

- [Intro to Orchestration](#)
- [Kubernetes](#)
- [Kubernetes Capstone Project](#)

# Intro to Computer Science

## Programming

- [Harvard's CS50](#)
- [Programming Capstone Project](#)

## Hardware Fundamentals

- [Code: The Hidden Language of Computer Hardware and Software](#)
- [Computer Principles](#)
- [From Nand To Tetris](#)
- [C Programming](#)
- [Hardware Fundamentals Capstone Project](#)

# Computer Systems

## Systems

- [Data Structures in C](#)
- [UC Berkeley - CS61C Great Ideas in Computer Architecture](#)
- [Software Construction](#)
- [Operating Systems](#)
- [System Capstone Project](#)

## Linux Programming

- [How Linux Works](#)
- [Advanced Programming in Unix](#)
- [Linux Programming Interface](#)

- **Linux Programming Capstone Project**

---

Revision #92

Created 31 October 2023 09:53:11 by 01Blu3

Updated 9 January 2024 17:28:46 by 01Blu3