# Structure

This document contains the proposed structure of the book. It is still work in progress.

* [Chapters](#chapters)
  + [Introduction by Steve Bourne](#introduction-by-steve-bourne)
  + [Section Zero - For the Newcomer](#section-zero---for-the-newcomer)
    - [Chapter 1 - Is this book for me?](#chapter-1---is-this-book-for-me)
    - [Chapter 2 - What is the Shell?](#chapter-2---what-is-the-shell)
    - [Chapter 3 - Opening the Shell](#chapter-3---opening-the-shell)
    - [Chapter 4 - Setting up your Shell](#chapter-4---setting-up-your-shell)
    - [Chapter 5 - The Absolute Basics](#chapter-5---the-absolute-basics)
    - [Interlude 1 - The Renaissance of the Shell](#X0f92883435f28a19825a36b2d3541b172276302)
  + [Section One - Transitioning from The GUI](#section-one---transitioning-from-the-gui)
    - [Chapter 6 - Moving Around](#chapter-6---moving-around)
    - [Chapter 7 - Organising Files](#chapter-7---organising-files)
    - [Chapter 8 - Creating Files, Opening Files, Previewing Files and Searching for Files](#X1ee94efa29f32a4ef9e3bdb4e3a3db8c9234e22)
    - [Chapter 9 - Become a Clipboard Gymnast](#chapter-9---become-a-clipboard-gymnast)
    - [Chapter 10 - Getting Help](#chapter-10---getting-help)
    - [Interlude 2 - The Unix Philosophy](#interlude-2---the-unix-philosophy)
  + [Section Two - Core Skills](#section-two---core-skills)
    - [Chapter 11 - Navigating the Command-line](#chapter-11---navigating-the-command-line)
    - [Chapter 12 - Job Control](#chapter-12---job-control)
    - [Chapter 13 - Thinking in Pipelines](#chapter-13---thinking-in-pipelines)
    - [Chapter 14 - The Subtleties of Shell Commands](#Xb32938a185ae26e6fe7ac714ae1cd837cff179a)
    - [Chapter 15 - The Secure Shell](#chapter-15---the-secure-shell)
    - [Interlude 3 - The Shell in Detail](#interlude-3---the-shell-in-detail)
  + [Section Three - Manipulating Text & Streams](#X8018443c56a52c398754fee7ca219f9a3bd426b)
    - [Chapter 16 - Slice and Dice with Cut and Rev](#X209ca9b8cdadc03e6b30892edc24f8cca447fa5)
    - [Chapter 17 - Getting to Grips with Grep](#chapter-17---getting-to-grips-with-grep)
    - [Chapter 18 - Advanced Text Manipulation with Sed and Awk](#Xd7100d2d47eaab42fbba59d74abcfead2ec252e)
    - [Chapter 19 - Clever Commands with Xargs and Quick Scripts](#Xf71bced5f29ca653e237efb4f4a1f4759572324)
    - [Chapter 20 - Bash Text Tricks - Regexes and String Manipulation](#X2bee1bebd70d04ed9adf04a755df606d86c9c12)
    - [Chapter 21 - The Missing Toolkit - jq, yq and xq](#X44ee6bdf15ea798748fe9a6a6659fd0682d1f9d)
    - [Interlude 4 - The Linux and Shell Family Tree](#X822d1c9b590158532d1efc7f3a9624ed6e51bc1)
  + [Section Four - Building Your Toolkit](#section-four---building-your-toolkit)
    - [Chapter 22 - Configuring the Shell](#chapter-22---configuring-the-shell)
    - [Chapter 23 - Customising your Command Prompt](#Xde638cba2c2c28e3f9978ec9286cd8ff052807d)
    - [Chapter 24 - Managing Your Dotfiles](#chapter-24---managing-your-dotfiles)
    - [Chapter 25 - Creating Shell Scripts and Understanding Shebangs](#X5bb57c30517257c3973f5bf1c5dfb7ea2169ef1)
    - [Chapter 26 - Gitting to Grips with Git](#chapter-26---gitting-to-grips-with-git)
    - [Interlude 5 - TBC](#interlude-5---tbc)
  + [Section Five - Shell Scripting](#section-five---shell-scripting)
    - [Chapter 27 - Looping over files and folders](#X1e0b10e96bc808c1e5ef14c84eb9d9061cb5e99)
    - [Chapter 28 - Basic Maths](#chapter-28---basic-maths)
    - [Chapter 29 - Understanding Heredocs](#chapter-29---understanding-heredocs)
    - [Chapter 30 - How to avoid scripting!](#chapter-30---how-to-avoid-scripting)
    - [Interlude Five - TBC](#interlude-five---tbc)
  + [Section Six - Advanced Techniques](#section-six---advanced-techniques)
    - [Chapter 31 - Managing Multiple Programming Languages with Make](#Xa132aee73f2123044a450a0948b6a2d9fcda7fb)
    - [Chapter 32 - The Power of Terminal Editors](#Xe26236ce1a21ff6086cfcb17743aeddb781064e)
    - [Chapter 33 - The Multiplexer](#chapter-33---the-multiplexer)
    - [Chapter 34 - Writing Tools Which Follow The Unix Philosophy](#X735e19feacaec694d7150bc27acafce7a6b5650)
    - [Interlude Six - The Future](#interlude-six---the-future)
* [Potential Future Chapters](#potential-future-chapters)
  + [Chapter - The Linux Filesystem, The Posix Standard and the GNU Project](#X6fbe02b9dcf2a31e0dc7123456f6b21aa69f5da)

## Chapters

### Introduction by Steve Bourne

For some, the text based interface to a computer might seem archaic in this day and age. However, the skills required to work with a computer shell might have never been in such high demand, or so relevant. In the introduction we cover just why the shell is more and more relevant for professionals and hobbyists, and why investing in your learning on this topic could be a really smart move.

### Section Zero - For the Newcomer

This section is for those who are completely new to this topic. In this section we’ll introduce just what the shell is, who this book is useful for, and what you can expect to learn.

We’ll also look at how to set your computer up so that you can follow along with the examples. We’ll finish by showing a few basic skills so that you can move around in the shell and get started with the rest of the book.

If you are already comfortable with running a shell, what bash is, and how to run basic commands like ls and cd, then you can comfortably skip this section and move onto Section One.

#### Chapter 1 - Is this book for me?

This book can be useful for a very wide group of people! First of all we’ll cover who might find this book useful, and whether it will be useful for you.

#### Chapter 2 - What is the Shell?

This chapter will briefly explain just what the shell is! It shows how a shell looks on Windows, MacOS and Linux, as well as how to open and run the shells which come with your computer. It also describes *why* you might use a shell in this day and age!

#### Chapter 3 - Opening the Shell

Now we’ll take a look at how to open the shell which comes with your computer, and run a simple command.

#### Chapter 4 - Setting up your Shell

In this book we are going to assume that you are using “Bash” - the “Bourne-Again Shell”. This is by far the most popular shell, and many online examples and documents assume that this is the shell you are using. Many alternative shells are also “Bash-like” and will be very familiar to Bash users. This is the shell you are most likely to use and come across in day-to-day work with a computer.

In this chapter we’ll make sure our system is able to run a recent version of Bash, regardless of whether we are using MacOS, Microsoft Windows, or Linux. There are a few different ways to configure your system, particularly if you are using Microsoft Windows. We’ll take a look at the options and choose the one which makes the most sense for you.

#### Chapter 5 - The Absolute Basics

Before we close this section, we’ll cover the absolutely basics of how to use a shell. We’ll look at the same soft of operations you might use when operating the computer from a graphical interface - how to move around the file system, copy and paste files, rename and move files and open files up. Once you’ve completed this chapter you’ll know the basics of how to use your shell to navigate your computer, and we are ready to get onto the meat of the book.

#### Interlude 1 - The Renaissance of the Shell

The “interludes” which end each section are completely optional! They don’t tech any specific skills, but instead give a little more flavour and background about the world of the shell, Linux and modern computing. In this first interlude we’ll look at just why the shell is experiencing something of a renaissance in the modern age of IT.

### Section One - Transitioning from The GUI

These are the key skills which everyone should know. Without them, you might struggle to perform certain tasks at all. Experienced users can probably skip this section, or just review the summary. But if you are new to the shell, this is the best place to start! This section focuses on helping you quickly get up to speed with how to perform the same kind of tasks you might have performed in a GUI (Graphical User Interface) with the shell.

#### Chapter 6 - Moving Around

Switching from a graphical user interface to the shell can take some getting used to. First we’ll take a look at how to navigate your system using the shell, and get information on files and folders in the system.

This section will touch on the pwd, cd, ls, pushd, popd and file commands, as well as introducing the concept of the “working directory” and “environment variables”.

#### Chapter 7 - Organising Files

Copying, moving, renaming and deleting files in a graphical user interface is normally fairly intuitive. Now we’ll learn how to perform the same operations in a shell. Once you can organise your files, you are well on your way to being able to use the shell more effectively for day to day tasks.

This chapter will introduce the cp, mv, rm, rmdir, and mkdir commands. It will also introduce ‘wildcards’ and ‘globs’.

#### Chapter 8 - Creating Files, Opening Files, Previewing Files and Searching for Files

Now that we know how to organise the files in our computer, we’ll take a look at how to create new files, preview the contents of files, open files, and particularly importantly, how to search for files.

This chapter will introduce the cat, less, touch, zip, unzip and find commands.

#### Chapter 9 - Become a Clipboard Gymnast

As you spend more time in the shell, you’ll find the clipboard particularly useful. In this chapter we’ll see how to access and manipulate the clipboard, as well as starting to see the basics of how ‘pipelines’ work, which is fundamental to using the shell.

This chapter will introduce the alias, pbcopy, pbpaste, xcopy, xpaste commands, as well as showing the basics of piping commands.

#### Chapter 10 - Getting Help

If you are new to the shell, then we’ve already covered a lot. It’s going to be critical to know how to remember all of these commands, and look up how they are used in the future, without interrupting your flow. In this chapter we’ll see how to get help on commands, as well as some particularly useful tricks to get the *right* information quickly, rather than a dense and complicated manual page.

This chapter will introduce the man, curl and help commands. We’ll also learn about the tldr, cht.sh and fuck tools, as well as how manual pages are structured in Linux.

#### Interlude 2 - The Unix Philosophy

We’ve already introduced a lot of commands. It turns out that the idea of small, focused commands which do one thing and one thing only is very fundamental to how Linux and Unix systems are designed. The idea of having a large number of simple tools, which can be composed together to perform complex operations is called “The Unix Philosophy”. Understanding this philosophy can help you understand why certain choices have been made with the shell and Linux tooling, and help you understand your system better.

### Section Two - Core Skills

There are a few techniques and skills which you’ll touch on time and time again. This section introduces some of the core skills and concepts which it is useful to understand when working with the shell.

#### Chapter 11 - Navigating the Command-line

If there’s one set of skills which will save you an enormous amount of time, it’s the skills needed to quickly navigate the command line. These skills are also immediately transferable to a whole set of other common tools. In this section we’ll see how to very quickly navigate the command-line with the keyboard, and where we can also use these techniques.

This chapter introduces the most commonly used keyboard shortcuts used in ‘readline’ based tools, like the shell and many programming tools.

#### Chapter 12 - Job Control

The shell contains a whole set of features for working with long running tasks. These features are called “Job Control”. Knowing how to move a task into the background, work on something else, move it back into the foreground, and generally control these jobs can be a real time-saver. Even if you don’t use these features regularly, understanding how job control works (and how to recover work you might have accidentally paused or put into the background) can be very helpful.

In this chapter we’ll introduce job control, and the bg, fg, and jobs commands.

#### Chapter 13 - Thinking in Pipelines

The shell allows us to chain sets of commands together. These ‘pipeline’ features allow us to take basic commands and perform increasingly complex and sophisticated operations.

In this chapter we’ll see how pipelines work, introduce the tee command, and learn about basic redirection.

#### Chapter 14 - The Subtleties of Shell Commands

Not everything we run in a shell works in the same way. Some commands run external programs, some commands are ‘built in’, and implemented by the shell itself. Understanding the different types of shell commands which exist can be very important when problem-solving or trying to work out why or how something works. In this chapter we’ll look at the subtle differences between different types of commands.

In this chapter we’ll see the which, whereis and related commands. We’ll also look at the differences between builtins, executables, aliases and functions.

#### Chapter 15 - The Secure Shell

One of the main reasons you might use a shell is to operate a machine remotely. In these cases it’s often the case that the *only* way to interact with these machines is via the shell, as they don’t have a graphical user interface installed. In this chapter we’ll look at how the secure shell, ssh works, and how to configure it for maximum convenience and efficiency.

#### Interlude 3 - The Shell in Detail

We’ve spent quite a bit of time with the shell now. For those who are interested, we’ll now look into the technicalities of what a shell is, how it differs from a terminal, and what a tty is.

### Section Three - Manipulating Text & Streams

A key part of how Linux and Unix systems work is that almost everything is represented as a text file in the system, and almost everything can be *configured* with a text file. This means that you may find yourself regularly manipulating text. There are almost too many options for how to do this! In this section we’ll look at some of the key techniques which can be used to work with text, and demonstrate this with practical examples.

#### Chapter 16 - Slice and Dice with Cut and Rev

One of the most simple and useful tools for working with text is the cut tool. In this chapter we’ll see how we can use it, in combination with rev to rapidly slice and dice text.

#### Chapter 17 - Getting to Grips with Grep

The grep tool is a real workhorse for shell users - once you’ve learned how to use it you will find yourself using it again and again. In this chapter we’ll see how you can use grep for common tasks, and how to use it in combination with other tools.

#### Chapter 18 - Advanced Text Manipulation with Sed and Awk

Sed - the ‘stream editor’ tool can be used to perform sophisticated manipulation of text. In many cases a small command involving sed can quickly solve problems. In this chapter we’ll look at some common ways to use sed, and when you might want to consider the awk tool.

#### Chapter 19 - Clever Commands with Xargs and Quick Scripts

With your grep, sed and cut skills ready to rock, you might find scenarios where you want to dynamically build new commands. In this chapter we’ll see how xargs can be used to build commands. We’ll also see some tricks for building commands dynamically, and writing them to a script file.

#### Chapter 20 - Bash Text Tricks - Regexes and String Manipulation

Bash has some built in capabilities which can be very handle when you are working with text. In this chapter we’ll take a look at the built in regular expression support in Bash, and also take a look at some nifty tricks for manipulating text in the shell.

#### Chapter 21 - The Missing Toolkit - jq, yq and xq

Sometimes you might find yourself battling with the shell text manipulation tools to manipulate JSON, YAML or XML. The jq, yq and xq tools can save you a lot of time, and might just be the missing tools you need to add to your toolkit.

In this chapter we’ll see how to install these tools and perform common tasks with them.

#### Interlude 4 - The Linux and Shell Family Tree

There are lots of different flavours of Linux and Unix, and lots of different shells. This interlude takes a quick look at the Linux and Unix family tree, and the different shells which have evolved over time.

### Section Four - Building Your Toolkit

As you work more with the shell, you will want to customise it and build tools and commands of your own. Before we look at shell scripting, we’ll take a look at the shell is configured, different ways a shell can run, and effective ways to manage your shell configuration.

#### Chapter 22 - Configuring the Shell

There are a number of different ways to configure your shell, and some options which can change how it operates. In this chapter we’ll take a look at the different configuration files for the shell and how they work, and how you can change your shell configuration. We’ll also see some of the shell options available which can change how the shell works.

#### Chapter 23 - Customising your Command Prompt

The shell command prompt can be configured to show you what you find most important. In this section we’ll see how the command prompt can be configured, and take a look at some of the advanced options available.

#### Chapter 24 - Managing Your Dotfiles

As you customise your shell and environment, it becomes more and more important to manage this customisation effectively and track changes to it. In this chapter we’ll see how to manage your configuration - and ‘dotfiles’ - as a GitHub repository. We’ll also see how this can be used to share ideas and look at some great examples.

#### Chapter 25 - Creating Shell Scripts and Understanding Shebangs

As you get more comfortable with the shell, you can bundle common commands into scripts. These scripts can use Bash features, or can be written with other languages. In this chapter we’ll see how to create simple shell scripts, scripts which use different languages, and see how ‘sourcing’ works.

#### Chapter 26 - Gitting to Grips with Git

If you are working with text, code, or other types of content, Git can be a powerful tool to manage changes and collaboration. People have sometimes found it hard to get to grips with in the shell, in this chapter we’ll look at the most operations for git, and how to use things like interactive commands to allow us to quickly and easily work with git repositories.

#### Interlude 5 - TBC

I haven’t got a solid idea for this interlude yet - I’m keeping track of ideas as I bulk out each chapter!

### Section Five - Shell Scripting

As we’ve seen, there’s a lot we can do with the shell. Creating shell scripts allows us to further automate and extend what we can do with these skills. In this section we’ll look at some of the key concepts which will help you write practical shell scripts for day to day use.

#### Chapter 27 - Looping over files and folders

One of the most common tasks we will do when scripting is operate functions over a set of files or folders. In this chapter we’ll look at basic loops, tests and how to operate on many files. We’ll also see how we can integrate commands like find into shell scripts.

#### Chapter 28 - Basic Maths

You can use the shell to perform arithmetic. Knowing how to do this can be quite convenient! In this chapter we’ll look at how basic arithmetic works in the shell and how to use it in scripts.

#### Chapter 29 - Understanding Heredocs

Heredocs are a feature of the shell you might have seen before, without necessarily understanding what is going on. In this chapter we’ll look at heredocs and how they can be used to manage files and build more advanced scripts.

#### Chapter 30 - How to avoid scripting!

The shell is powerful, but can be complex. There are times where jumping into a shell script can make a task more complex than it needs to be. In this chapter we’ll briefly look at some alternatives to shell scripting, from simple python and ruby scripts, to basic C programming. We’ll also get an understanding of how to compile programs we’ve downloaded, which can be a common requirement on Linux systems.

#### Interlude Five - TBC

To be confirmed.

### Section Six - Advanced Techniques

The more time you spend in the shell, the more effective you may well find yourself. If you are benefiting from your shell chops, this section contains suggestions for deeper topics to look into, which can take you even further.

#### Chapter 31 - Managing Multiple Programming Languages with Make

GNU Make can be a powerful tool for anyone who uses multiple programming languages, or wants to make it easier to allow people to use their projects. In this chapter we’ll look at some common patterns which we can use make for to make our lives easier when working with different programming languages and platforms.

#### Chapter 32 - The Power of Terminal Editors

If you are already in the shell, then you might find it convenient to edit code and files directly from the shell. Getting familiar with some of the capabilities of Vim or Emacs can open up a whole new world for how you work with text in the shell.

#### Chapter 33 - The Multiplexer

If you are regularly using a shell, then learning how to use a terminal multiplexer like screen or tmux can greatly improve your productivity. In this chapter we’ll see how terminal multiplexers work and brush the surface of what they can do.

#### Chapter 34 - Writing Tools Which Follow The Unix Philosophy

To finish the book we’ll look at how you can write tools which integrate seamlessly into the shell. We’ll see how tooling and patterns have evolved, and some suggestions for how to write tools which others can use easily. We’ll look at some of the similarities and patterns in very popular tools like aws, kubectl and gcp.

#### Interlude Six - The Future

The shell, in particular the Bourne-Again Shell has been popular for many years. But what does the future hold? With the advent of the Linux Subsystem for Windows, new shells like Nushell, and the latest version of MacOS switching from Bash to Z-Shell, we finish off by looking at some of the trends which might shape how we use shells in the future.