

Advanced **bash** scripting

Michael F. Herbst

6th – 10th November 2017

Course description and content

The **bash** shell is the default shell in almost all major UNIX and Linux distributions, which makes learning about the **bash** scripting language pretty much unavoidable if one is working on a UNIX-like operating system. On the other hand this also implies that writing **bash** scripts is conceptually very simple — essentially like making structured notes of the commands one would need to type in the shell anyway.

When it comes to more involved tasks and more powerful scripts, however, taking a deeper look at the underlying operating system is typically required. After all **bash** scripting is all about properly combining the available programs on the UNIX operating system in a clever way as we will see.

In the first part of the course we will hence revisit some basic concepts of a UNIX-like operating system and review the set of UNIX coreutils one needs for everyday scripting. Afterwards we will talk about the **bash** shell and its core language features, including

- control statements (**if**, **for**, **while**, ...)
- file or user input/output
- **bash** functions
- features simplifying code reuse and script structure

The final part will be concerned with the extraction of information (from files ...) using so-called regular expressions and programs like **awk**, **sed** or **grep**.

Learning objectives

After the course you will be able to

- apply and utilise the UNIX philosophy in the context of scripting
- identify the structure of a **bash** script
- enumerate the core concepts of the **bash** scripting language
- structure a script in a way such that code is reusable in other scripts
- extract information from a file using regular expressions and standard UNIX tools

- name advantages and disadvantages of tools like `awk`, `sed` or `grep`, `cut` ..., and give examples for situations in which one is more suitable than the others.

Prerequisites

Familiarity with a UNIX-like operating system like GNU/Linux and the `bash` shell is assumed. For example you should be able to

- navigate through your files from the terminal.
- create or delete files or folders from the terminal.
- run programs from the terminal (like some “one-liners”).
- edit files using a common graphical (or command-line) text editor like `gedit`, `leafpad`, `vim`, `nano`, ...

It is not assumed, but highly recommended, that you have have some previous experiences with programming or scripting in a UNIX-like operating system.

Laptops for the exercises

Since part of the course will take place in seminar rooms without computers, you will need to bring your own laptops to the course to work on the exercises.

Pretty much any standard Linux or UNIX distribution will have the required software in their repositories. This explicitly includes Debian, Ubuntu, Linux Mint, Arch, Fedora on the Linux side and FreeBSD and MacOS X on the UNIX side. We will install what is needed during the course in case it is not yet on your machine by default anyway.

For Windows computers I recommend using a virtual machine with Linux Mint. Please let me know if you plan to bring along a Windows machine, such that I can prepare some images in advance.