

Advanced `bash` scripting

Michael F. Herbst

24. – 28. August 2015

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 means that writing `bash` scripts is conceptually very simple — essentially like typing commands. When it comes to more involved tasks and more powerful scripts, however, some knowledge of the underlying operating system is certainly required. After all `bash` scripting is all about properly combining the available programs in a clever way.

This idea structures the whole course: In the first part we will 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`.

Participants are encouraged to bring examples of problems they had in their everyday workflow to the course for discussion.

Learning targets and 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 the 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

This course assumes some familiarity with a UNIX-like operating system like GNU/Linux and the `bash` shell. I.e. 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`, . . .

Whilst it is not assumed that you have any knowledge of programming or any experience in `bash` scripting, it is, however, highly recommended that at least either is the case.