

# Session 1: Concepts



P. S. Langeslag

18 October 2018

# Power User

Someone who uses advanced functionality

- ▶ Example: regular expressions

## Does a Power User Need Linux?

No;

but it helps to have a terminal and a range of POSIX-compliant tools.

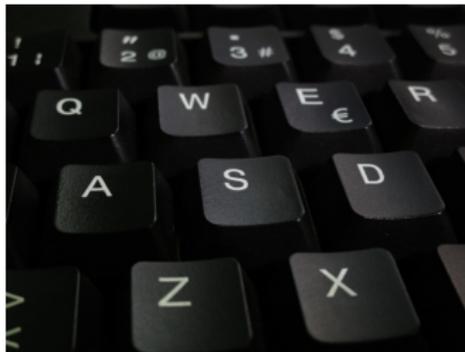
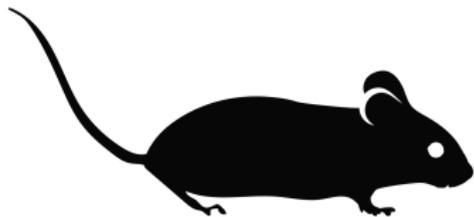
### POSIX

Portable Operating System Interface (1988–), based on UNIX.

A standard for

- ▶ System processes
- ▶ Shells
- ▶ Basic utilities available to users (`awk`, `sed`)

# Mouse vs Keyboard vs Touch (vs Voice)



# UNIX-Like Operating Systems

Table 1: UNIX-like operating systems

Initial Release	Product
1960s	MULTICS
1969	UNIX
1977	BSD
1984	Mac OS
1984	GNU
1987	MINIX
1991	Linux

# Linux or GNU/Linux?

- ▶ Operating system
  - ▶ Kernel
  - ▶ Userland

# The UNIX Philosophy

1. Everything is a file
2. Any tool should do one thing well

# The UNIX Filesystem

---

/	Top-level directory
bin	Essential executables
boot	Boot partition
dev	Hardware interface
etc	System-wide configuration files
home	User directories
lib	Library files
mnt	Storage
opt	Third-party software
proc	System resources
root	Root user home directory
sys	Hardware information
tmp	Temporary files
usr	Executables, libraries, documentation for user apps
var	Log files

---

# Pipeline

```
xrandr --current | grep -P '(?<!eDP1) connected' |  
awk '{print $1;}'
```

- ▶ stdin: standard input (keyboard input or forwarded output)
- ▶ stdout: standard output (display or forward stream)
- ▶ stderr: error output stream

# Free Software

- ▶ Free as in freedom of speech: allowed to modify and share code.
- ▶ Contrast free as in beer, i.e. free of monetary compensation. Cf. the definition of the Free Software Foundation:

“Free software” means software that respects users’ freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”. We sometimes call it “libre software,” borrowing the French or Spanish word for “free” as in freedom, to show we do not mean the software is gratis.<sup>1</sup>

---

<sup>1</sup>“What Is Free Software?” n.d., <https://www.gnu.org/philosophy/free-sw.html>.

# Types of “Programming” Language

1. Compiled languages
  - ▶ Fortran, BASIC, C, Java, Haskell
2. Interpreted languages (scripting languages)
  - ▶ bash, Python, Ruby, PHP, HTML, XML, CSS, SQL
1. General-purpose languages
  - ▶ Fortran, BASIC, C, Haskell, Python, Ruby
2. Domain-specific languages
  - ▶ bash, JavaScript, PHP, HTML, XML, CSS, SQL
1. Low-level languages
  - ▶ Fortran, C
2. High-level languages
  - ▶ BASIC, Haskell, Java, Python, Ruby, PHP

## Linus Torvalds on C Versus C++ (1/3)

On Wed, 5 Sep 2007, Dmitry Kakurin wrote:

When I first looked at Git source code two things struck me as odd: 1. Pure C as opposed to C++. No idea why. Please don't talk about portability, it's BS.

*YOU* are full of bullshit.

C++ is a horrible language. It's made more horrible by the fact that a lot of substandard programmers use it, to the point where it's much much easier to generate total and utter crap with it. Quite frankly, even if the choice of C were to do *nothing* but keep the C++ programmers out, that in itself would be a huge reason to use C.

In other words: the choice of C is the only sane choice. I know Miles Bader jokingly said "to piss you off", but it's actually true. I've come to the conclusion that any programmer that would prefer the project to be in C++ over C is likely a programmer that I really *would* prefer to piss off, so that he doesn't come and screw up any project I'm involved with.

## Linus Torvalds on C Versus C++ (2/3)

C++ leads to really really bad design choices. You invariably start using the “nice” library features of the language like STL and Boost and other total and utter crap, that may “help” you program, but causes:

- ▶ infinite amounts of pain when they don't work (and anybody who tells me that STL and especially Boost are stable and portable is just so full of BS that it's not even funny)
- ▶ inefficient abstracted programming models where two years down the road you notice that some abstraction wasn't very efficient, but now all your code depends on all the nice object models around it, and you cannot fix it without rewriting your app.

## Linus Torvalds on C Versus C++ (3/3)

In other words, the only way to do good, efficient, and system-level and portable C++ ends up to limit yourself to all the things that are basically available in C. And limiting your project to C means that people don't screw that up, and also means that you get a lot of programmers that do actually understand low-level issues and don't screw things up with any idiotic "object model" crap.

So I'm sorry, but for something like git, where efficiency was a primary objective, the "advantages" of C++ is just a huge mistake. The fact that we also piss off people who cannot see that is just a big additional advantage.

(6 Sep 2007, on the [gmane.comp.version-control.git](mailto:gmane.comp.version-control.git) newsgroup)

## Sidenote: Developer Culture and Inclusiveness



Figure: Linus Torvalds (still from “Aalto Talks with Linus Torvalds”)

Bright, Peter. “Linus Torvalds Apologizes for Years of Being a Jerk, Takes Time Off to Learn Empathy: And Linux Has Adopted a Real Code of Conduct to Replace Its Previous ‘Code of Conflict’.” *Ars Technica*, September 17, 2018. <https://arstechnica.com/gadgets/2018/09/linus-torvalds-apologizes-for-years-of-being-a-jerk-takes-time-off-to-learn-empathy/>

# Aims of the Course

To equip students with a first practical knowledge of terminal-based productivity tools.

- ▶ Markdown for text processing and presentation slides
- ▶ L<sup>A</sup>T<sub>E</sub>X for text processing and presentation slides
- ▶ bibl<sub>a</sub>tex for citation management
- ▶ The bash shell for file operations
- ▶ git for version control
- ▶ (tmux for multiplexing)

The greatest potential of this course is in self-study!

## Recommended Reading (1/2)

### bash

Vossen, J. P., and Carl Albing. *bash Cookbook*. 2nd ed. Sebastopol, CA: O'Reilly, 2017.

Robbins, Arnold. *bash: Pocket Reference*. Sebastopol, CA: O'Reilly, 2010.

Evans, Julie. *Bite Size Command Line*. Computer Wizard Industries, 2008. <https://gumroad.com/l/bite-size-command-line>.

### Vim

———, Elbert Hannah, and Linda Lamb. *Learning the vi and Vim Editors: Pocket Reference*. 7th ed. Sebastopol, CA: O'Reilly, 2008.

———. *vi and Vim Editors: Pocket Reference*. 2nd ed. Sebastopol, CA: O'Reilly, 2011.

## Recommended Reading (2/2)

### LaTeX

Kottwitz, Stefan. *LaTeX: Beginners Guide*. Birmingham: Packt, 2011.

———. *LaTeX Cookbook*. Birmingham: Packt, 2015.

“LaTeX.” <https://en.wikibooks.org/wiki/LaTeX>.

“Overleaf.” <https://www.overleaf.com>.

### Other

Stubblebine, Tony. *Regular Expression: Pocket Reference*. Sebastopol, CA: O'Reilly, 2009.

# Logging into the Course Terminal

1. You need to be on the campus network, physically or by VPN.
2. You'll need an ssh client:
  - ▶ Included in all Linux distributions, all versions of OS X;
  - ▶ On Windows, install [PuTTY](#).  
(stock ssh client trickier to set up for X11 forwarding)
3. To make use of graphical applications on Windows, you'll also need [Xming](#) or [Cygwin](#).
4. Log onto [langeslag.uni-goettingen.de](https://langeslag.uni-goettingen.de) on port 22 with the credentials I have sent you.

# PuTTY Settings (Basic)

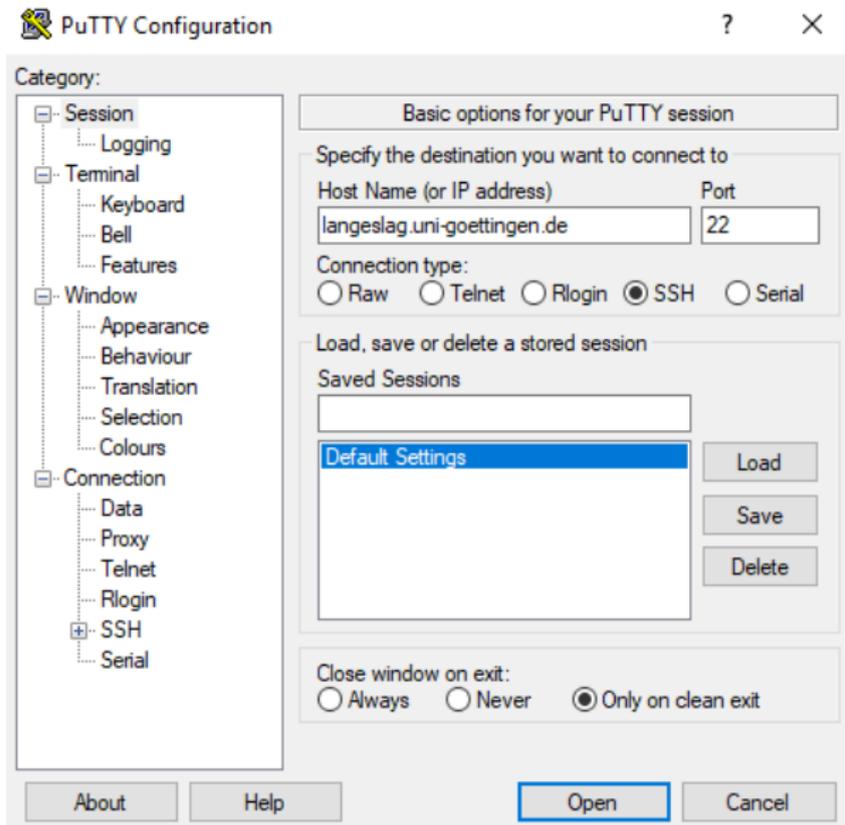


Figure: PuTTY: connection settings

# PuTTY Settings (X11 Forwarding)

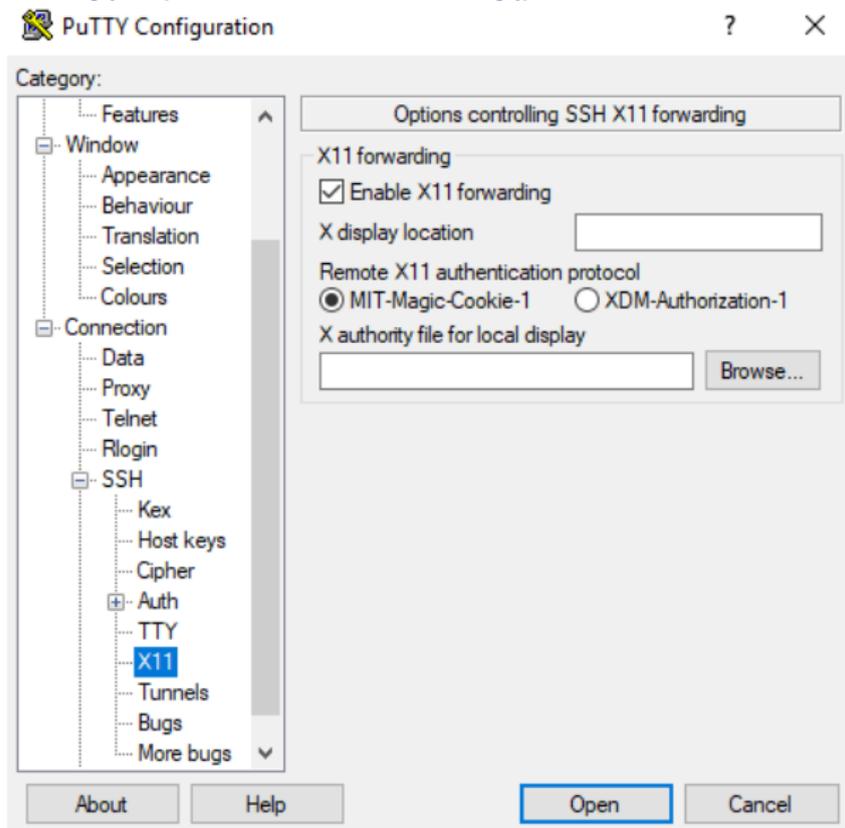


Figure: PuTTY: X11 forwarding

# PuTTY With Xming

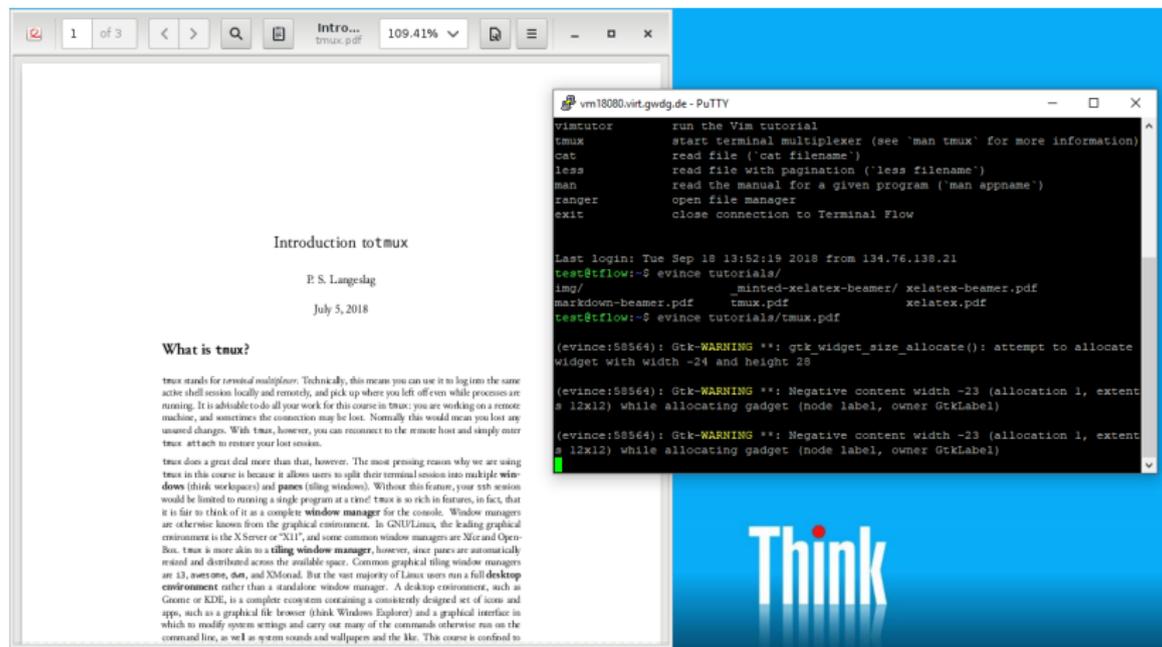


Figure: PuTTY with Xming

# PuTTY Settings (Appearance)

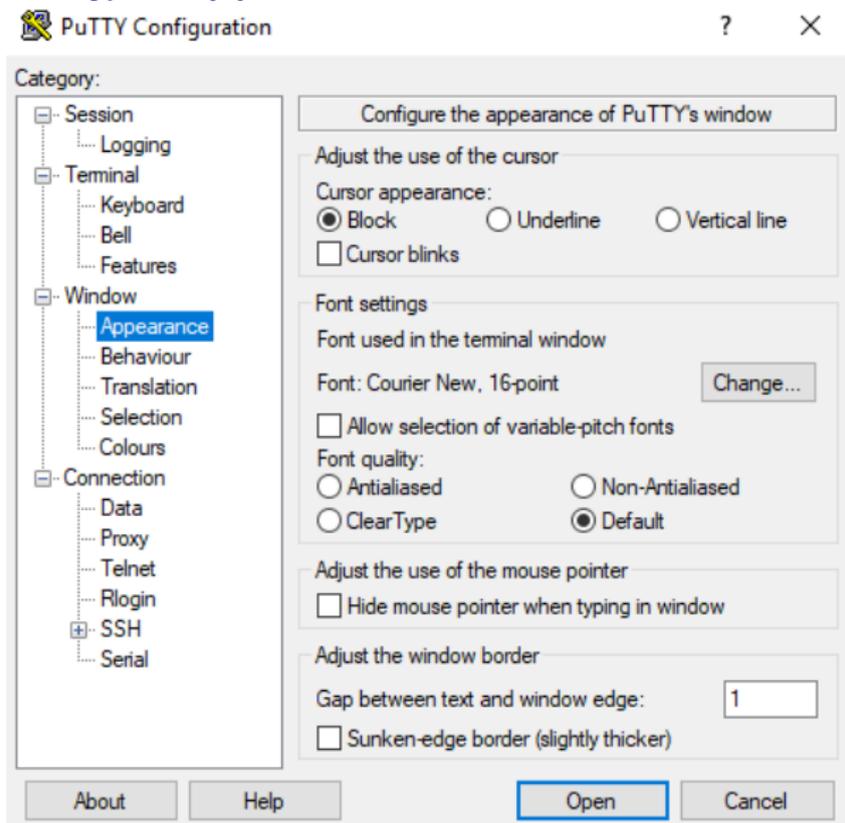


Figure: PuTTY: X11 forwarding

## ssh on Linux or Mac

```
$ ssh -Y username@langeslag.uni-goettingen.de
```

## If You Encounter X11 Forwarding Issues

1. Disable X11 forwarding (e.g. drop the `-Y` argument)
2. Modify your `~/.latexmkrc` as follows:

```
< $pdf_previewer = 'evince';  
> $pdf_previewer = 'less';
```

3. Create a file `~/.bashrc` with the following content:

```
export PDFVIEWER_texdoc="less"
```

Issues will remain because I haven't configured an explicit fallback option.

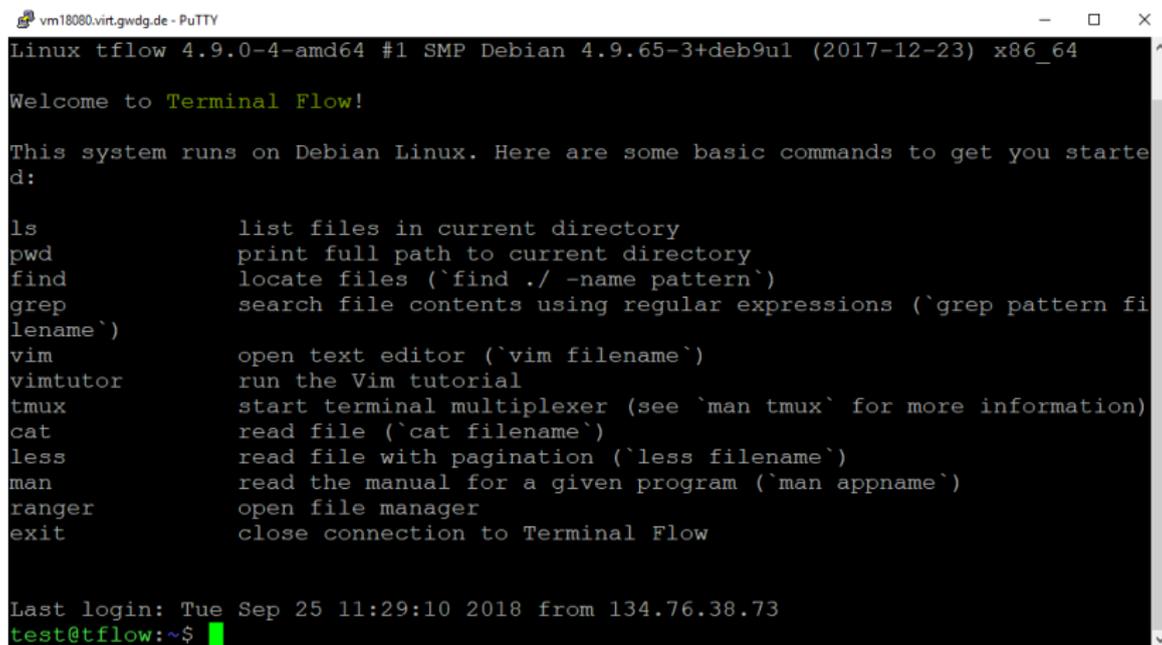
# Virtual Private Network (VPN)

See [https://info.gwdg.de/docs/doku.php?id=en:services:network\\_services:vpn:start](https://info.gwdg.de/docs/doku.php?id=en:services:network_services:vpn:start)

## Linux

```
sudo ip tuntap add mode tun tun0
sudo ip link set dev tun0 up
sudo openconnect 134.76.22.1
```

# The Shell in PuTTY

A screenshot of a PuTTY terminal window. The title bar at the top reads "vm18080.virt.gwdg.de - PuTTY". The terminal content shows the system information: "Linux tflow 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86\_64". Below this is a green "Welcome to Terminal Flow!" message. A block of text lists basic Linux commands and their descriptions. At the bottom, it shows the last login time and the current shell prompt "test@tflow:~\$".

```
vm18080.virt.gwdg.de - PuTTY
Linux tflow 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64

Welcome to Terminal Flow!

This system runs on Debian Linux. Here are some basic commands to get you started:

ls                list files in current directory
pwd               print full path to current directory
find              locate files (`find ./ -name pattern`)
grep              search file contents using regular expressions (`grep pattern filename`)
vim               open text editor (`vim filename`)
vimtutor          run the Vim tutorial
tmux              start terminal multiplexer (see `man tmux` for more information)
cat               read file (`cat filename`)
less              read file with pagination (`less filename`)
man               read the manual for a given program (`man appname`)
ranger            open file manager
exit              close connection to Terminal Flow

Last login: Tue Sep 25 11:29:10 2018 from 134.76.38.73
test@tflow:~$
```

Figure: The Terminal Flow welcome prompt (“MOTD”)

# Your Home Directory

---

/home/username/

experimental/

public\_html/

tutorials/

work/

.config/

.vim/

.vimrc\_background

.latexmkrc

.Xauthority

---

# Command Syntax

---

Program Name	Options	Arguments
true		
false		
pwd		
ls		
ls	-a	
date	+%A\ %d\ %B	
cd		~/experimental
ls	-lh	~/tutorials
rm	-rf	file1 dir1 dir2 file2

---

# In-Terminal Aid with Programs

- ▶ Tab completion
- ▶ `which`
- ▶ `man`

# Manual Sections

---

Section	Description
1	General commands
2	System calls
3	Library functions
4	Special files
5	File formats
6	Games and screensavers
7	Miscellaneous
8	System administration commands; daemons

---

## Navigating less

---

Official key	Action	Also permitted
j / ENTER	One line down	Cursor down
k	One line up	Cursor up
d	Half-screen down	
u	Half-screen back	
f, SPACE	Page down	PgDown
b	Page up	PgUp
/	Find	
q	Quit	

---

## References I

“Aalto Talk with Linus Torvalds [Full-Length].” Aalto Center for Entrepreneurship, June 14, 2012.

<https://www.youtube.com/watch?v=MShbP3OpASA>.

Bright, Peter. “Linus Torvalds Apologizes for Years of Being a Jerk, Takes Time Off to Learn Empathy.” *Ars Technica*, September 17, 2018.

<https://arstechnica.com/gadgets/2018/09/linus-torvalds-apologizes-for-years-of-being-a-jerk-takes-time-off-to-learn-empathy/>.

Evans, Julie. *Bite Size Command Line*. Computer Wizard Industries, 2018. <https://gumroad.com/l/bite-size-command-line>.

Kottwitz, Stefan. *LaTeX: Beginners Guide*. Birmingham: Packt, 2011.

———. *LaTeX Cookbook*. Birmingham: Packt, 2015.

“LaTeX,” n.d. <https://en.wikibooks.org/wiki/LaTeX>.

“Overleaf,” n.d. <https://www.overleaf.com>.

## References II

Robbins, Arnold. *bash: Pocket Reference*. Sebastopol, CA: O'Reilly, 2010.

———. *vi and Vim Editors: Pocket Reference*. 2nd ed. Sebastopol, CA: O'Reilly, 2011.

Robbins, Arnold, Elbert Hannah, and Linda Lamb. *Learning the vi and Vim Editors: Pocket Reference*. 7th ed. Sebastopol, CA: O'Reilly, 2008.

Stubblebine, Tony. *Regular Expression: Pocket Reference*. Sebastopol, CA: O'Reilly, 2009.

Vossen, J. P., and Carl Albing. *bash Cookbook*. 2nd ed. Sebastopol, CA: O'Reilly, 2017.

“What Is Free Software?” n.d.

<https://www.gnu.org/philosophy/free-sw.html>.