Linux Mastery: Master the Linux Command Line

Mastering the Linux Terminal (S3)

Open and Close Terminal

- Keyboard- Shortcut: Ctrl + Alt + t (open), Crtl + d (close)

Our First Commands

- cal → calendar
- history → prev command list
    - !<number>
    - history -c; history –w → clear history
- clear shortcut: ctrl + l

Terminals, Commands and Shells

- shell interprets the commands(text) from the terminal

Understanding Command structure

- structure: <commandName> <options> <inputs>
    - commandName:
        - program you want to run in shell path
        - shell paths: echo $PATH
        - program path: which <commandName>
    - inputs:
        - not always required or optional
    - options:
        - can be chained together (eg. -abc)
        - long form with -- (eg. -h or --help)
            - can't be chained together
        - options can have inputs too
- commands are case sensitive

Using Linux Manual - Structure

- manual split into 8 sections

User Commands	Commands that can be run from the shell by a normal user (typically no administrative privileges are needed)
System Calls	Programming functions used to make calls to the Linux kernel
C Library Functions	Programming functions that provide interfaces to specific programming libraries.
Devices and Special Files	File system nodes that represent hardware devices or software devices.

File Formats and Conventions	The structure and format of file types or specific configuration files.
Games	Games available on the system
Miscellaneous	Overviews of miscellaneous topics such as protocols, filesystems and so on.
System administration tools and Daemons	Commands that require root or other administrative privileges to use

Using Linux Manual - Man Pages

- man command: man [options] input
    - –k option gives you a list of manpages the searchterm is contained
        - Access: man <section> <searchTerm>
- Synopsis Symbols:

[THING]	THING is optional.
<THING>	THING is mandatory (required)
THING …	THING can be repeated (limitlessly)
THING1 | THING2	Use THING1 OR THING2. Not Both.
THING	THING [Notice the Italics] Replace THING with whatever is appropriate.

Using Linux Manual - Putting it all together

- Not always manpage available
    - help <commandName>

Command Input and Output

- standard input(0): input for commands as data stream
- command arguments: can't be piped or redirected
- standard output(1): by default appears on terminal screen
- standard error(2): appears in terminal when error happens

Redirection - Standard Output

- redirect output with >
    - overwrites the file
    - append output to file with >>

Redirection - Standard Input + Standard Error

- redirect standard error with: 2>
    - append: 2>>
- both: <command> > <file1> 2> <file2>
    - append: <command> >> <file1> 2>> <file2>
- input: <
- you can also redirect to different terminals (tty to get terminal location)
- https://www.gnu.org/software/bash/manual/html_node/Redirections.html

Piping – Fundamentals

- Piping with: |
    - <command1> [options] [inputs] | <command2> [options] [inputs] | …
    - Second command uses output of first command
    - redirection is processed before pipe in default

Piping - Tee Command

- tee - read from standard input and write to standard output and files
- tee [OPTION]... [FILE]...
    - <command1> [options] [inputs] | tee <filename> | <command2> [options] [inputs]

Piping - Xargs Command

- xargs - build and execute command lines from standard input
    - <command1> [options] [inputs] | xargs <command2> [options] [inputs]
    - Command2 inputs are used before xargs inputs

Aliases

- Specify chain of commands, pipes, etc. under a single alias
- Have to be stored in .bash_aliases in home folder
- Structure: alias <aliasName>='<commands, pipes and co.>'

Mastering the Linux File System (S4)

Structure of the Linux File System

- Tree structure
- See cheat sheet for standard paths

Navigating the File System

- pwd → current path
- ls → use man page if you don't know it
- cd → use man page if you don't know it

File Extensions in Linux

- file command to see file type
- file extensions don't matter in Linux OS, header is important
    - Linux programs will still struggle with wrong extensions

Wildcards

- http://tldp.org/LDP/GNU-Linux-Tools-Summary/html/x11655.htm

Creating Files and Folders

- touch → see man page
    - brace expansion to create multiple files (in multiple folders too)
        - touch {a,b,c}_{1..3}/file{1..100} → creates 100 files in every folder
- mkdir → see man page
    - –p entire path gets created
    - brace expansion to create multiple folders:
        - mkdir {a,b,c}_{1..3} → creates 9 folders

Deleting Files and Folders

- rm → see man page
    - also works with wildcards
    - –r to delete recursive
    - –f to force
    - –i interactive
    - Brace expansion also works
- rmdir → see manpage (good to remove empty directories)

Copying Files and Folders

- cp → see manpage
    - –r to copy recursive (folders)

Moving + Renaming Files and Folders

- mv → see manpage

Editing Files using Nano

- https://www.nano-editor.org/dist/latest/cheatsheet.html
- nano settings under /etc/nanorc

Locate Command

- get path of searched term using a db
  - wildcards etc. are possible
  - –i → case insensitive
  - –e → checks if files exist
- locate –S → locate db info
  - Updated once a day or with sudo updatedb

Find Command

- For a lot of search tasks without use of db(always up to date/slower)
- find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]
  - –maxdepth <number> → only searches amount of levels down
    - Should be put first
  - –type <type> → search for files(f), directories(d), …
  - –name "<pattern>" → search for a name pattern
    - -iname for case insensitive
  - –size +|-<size> → search files with size bigger/smaller than size
  - –exec <command> → execute command on every find
    - Use –ok instead -exec for user confirmation
    - E.g.: sudo find Schreibtisch/ -size +100k -size -5M -type f -exec cp {} ~/copy_here \;

Viewing Files

- cat → read files
- tac → reads file in reversed lines
  - also works with mp3 (nice to know)
- rev → reversed content of the lines
- less → read file/input scrollable (pipe standard input in it)
- head –n <number> → read first n lines of file/input
- tail –n <number> → read last n lines of file/input
- wc → word count
  - –l → line count

Sorting Data

- sort → sorts input
  - –r → reversed
  - –n → for numbers
  - –h → for human readable data
  - –M → sort by month
  - –u → unique results
  - –k <columnNumber>[sort options] → sort by column number

Searching File Content

- grep → see man page
  - –i → case insensitive
  - –a → count
  - –v → invert search(find lines who don't contain search)

    - Good in combination(pipe) to filter output of other commands like ls, man

File Archiving and Compressing

- tar → see man page
    - –c → create
    - –v → verbose, for output
    - –f → for files
    - –t → test, to see what's inside the file
    - –x → extract files
    - –z → use gzip for compression
    - -j → use bzip2 for compression
- gzip → see manpage
    - gunzip → see manpage
- bzip2 → takes more time than gzip, but gets mostly more compression
    - better for bigger files
    - bunzip2
- zip → no need to create a tar before, not as compressed but works for windows too
    - unzip

Mastering Task Automation (S5)

Creating Bash Scripts

- BASH = born again shell
- chmod +x <file>→ to make file executable
- add PATH to .bashrc to add script paths to terminal commands

Scheduled Automation Using Cron

- https://crontab.guru/
- crontab –e → to edit crontab

Mastering Open Source Software (S6)

The GNU Project

- freedom to run programs as you wish
- freedom to study how the program works and change it → open source
- freedom to redistribute
- freedom to redistribute copies
- GNU public license
- Linux Kernel under GNU licence
- www.gnu.org

Compiling Software from Source Code

- Sudo apt-get install gcc → for c compiler
  - configure → to configure gcc for the system
- sudo apt-get install make → for make
  - make → compile the files with cmake
  - sudo make install to install compiled code

The Software Repositories

- https://help.ubuntu.com/community/Repositories/Ubuntu
- https://packages.ubuntu.com/
- uname –m → to get architecture

The Apt Cache

- apt-cache search <searchterm> → to find packages in cache
  - cache under /var/lib/apt/lists

Updating the Cache and Upgrading Software

- sudo apt-get update → update apt cache
- sudo apt-get uprade → update installed packages

Installing New Software

- apt-get install <package>[=<version>] → install a package

Downloading Source Code

- https://help.ubuntu.com/community/Repositories/CommandLine
- /etc/apt/sources.list → sources are stored here
  - Uncomment sources here to be able to download them
- sudo apt-get dpkg-dev → to install sourcecode
- sudo apt-get source <package> → install sourcecode

Uninstalling Software

- sudo apt-get remove <package> → not full removal
- sudo apt-get purge <package> → also removes config files
- sudo apt-get autoremove → removes unused dependencies
- sudo apt-get clean → removes archived packed packages
- sudo apt-get autoclean → removes unused archived packages