UNIX/Linux for beginners

11 Aug 2022

Making sense of data

What's the point?

- To give you a very broad overview
- To flood you with concepts & tools
- To give you TIME to TINKER

CAVEAT: everything rather Leo-centric

Not on the menu

- Sysadmin stuff
- Shell scripting
- Software development toolchains

UNIX

- UNIX: A family of computer operating systems
  - multi-user
  - interactive
  - portable
  - modular design

The UNIX philosophy: Do One Thing And Do It Well

- unified, hierarchical file system
- simple, composable tools, run as processes
- inter-process communication using pipes and (text) streams
- command-line interpreter + scripting

```
$ curl "https://en.wikipedia.org/wiki/Pipeline_(Unix)"
  | sed 's/[^a-zA-Z ]/ /g'
  | tr 'A-Z ' 'a-z\n'
  | grep '[a-z]'
  | sort -u
  | comm -23 - <(sort /usr/share/dict/words)
  | less
```

UNFORTUNATELY: the monoliths won…

…OR DID THEY…..?

Linux Versions and Distributions

- Unix: System V, BSD, AIX, HP-UX, SunOS/Solaris, Linux
- Linux: Ubuntu, Debian, CentOS, Red Hat, NixOS
- Kernel + software packages
- Graphical / Windowing environment: KDE, GDM, X
- Command-line: shell, terminal, console, prompt

- Mac OS is BSD-based
- On Windows, WSL can run anything you want
- So can Docker containers

Loggin on

- Console: graphical or command-line
- Remote connection: `telnet`, `ssh`
- User name and password
- Home directory, environment, shell
- Superuser: *root*
- Top of the file system: also root (a.k.a. `/`)

Different shells:

- minimal: `sh` / `dash`
- maximal: `bash`
- hipper than hip: `zsh`

Finding your way around

- Changing directories:
  - absolute: `cd /usr/share/doc`
  - relative: `cd ./somewhere/else`
  - `cd ~`
  - `cd $HOME`
  - `cd`
- Look at contents of a file:
  - `ls`, `cat`, `more`/`less`

- *current directory*: . (`pwd`)
- *parent directory* .. (`cd ..`)
- *previous directory*: - (`cd –`)

Finding commands

- Located in directories such as
  - `/bin`
  - `/usr/bin`
  - `/usr/local/bin`
  - `/usr/sbin`
- See your execution search path:
  - `echo $PATH`

- Your current directory should *never* be in your `$PATH`!
- Just use: `./command-name`

Shell basics

- Command have options and positional arguments:
  - `ls –alrt ––color /usr/bin /usr/sbin`
- Streams: stdin, stdout, stderr, and pipes
  - `cat foo | wc > results.txt`
- Shell globbing and variable expansion:
  - `cat *.py ?.cc ??.java *.*`
  - `rm –rf * $HOME /`

- Shell globbing syntax is similar to, but not the same as *regular expressions*
- Single quotes or backslash-escapes will prevent expansion and globbing

Environment basics

- Environment variables:
  - manipulate with: `env`, `export`, `unset`
  - tweak permanently in files like `.profile`, `.bashrc`, `.zshrc`, etc.
- Aliases:
  - `alias dir="ls -lashF -color`
  - `alias rm="rm -I"`

- Navigate through your command history:
  - `history`
- Use up/down arrow keys, `ctrl-A`, `ctrl-E`, `ctrl-R`

Other interesting shell tools

- `df –h`
- `du –sh *`
- `sudo –i` (has to be set up to work for you)
- `uptime`
- `w`
- `reset`

- Advanced, but very, very useful:
  - `for` loops
  - stream redirection
  - `tee`

File tools

- finding: `ls`, `find` / `xargs`, `tree`
- manipulating: `cp`, `mv`, `rm`, `mkdir`/`rmdir`, `touch`
- integrity: `md5sum sha256sum`
- compressing/archiving:
  - tar, zip (.zip), gzip (.gz), bzip2 (.bz2), compress (.Z)
- permissions: `chown`, `chmod`

- Advanced:
  - symbolic links: `ln`
  - bulk remote transfer: `rsync`

Text tools (filters)

- finding: `grep`
- processing: pipes, `cat`, `head`/`tail`, `sort`, `uniq`
- changing: `sed`
- comparing: `diff`

- Advanced:
  - `awk`

Process tools

- monitoring: `ps`, `top`, `htop`, `pstree`, `time`
- manipulation: `kill`, `ctrl-Z`, `bg`, `fg`
- automated execution: `crontab`
- finding users: `w`, `last`

- WHAT IS KEEPING MY DISK FROM UNMOUNTING??
  - `lsof directoryname`

Networking tools

- Remote connections:
  - old school: **`telnet`**, netcat (**`nc`**)
  - new school: **`ssh`**
- File transfer:
  - old school: **`ftp`**
  - new school: **`scp`**, **`sftp`**
  - best school: **`rsync`**

- *My urgent recommendation*:
  - consider using *ssh keys* instead of passwords

More networking tools

- local interfaces: `ifconfig`, `ip`
- web connectivity: `wget`, `curl –I`
- dns: `dig`, `nslookup`, `host`
- `ping`, `traceroute`

- Advanced, but very, very useful:
  - `screen`, `tmux`

Superuser tools

- can typically only be run as root (`sudo –i`)
- disks: `mount`, `umount`, `/etc/fstab`
- machine: `reboot`, `halt`

More obscure, still very useful

- `jq` – command-line JSON processor
- `rsync` - batch copying files
- `inxi` – what's my hardware?
- `lspci` – no really, what's my hardware?
- `w3m` – console web browser
- *ImageMagick* – `convert`, `identify`, `mogrify` images

- These tools are typically not standard but need to be installed:
  - Debian/Ubuntu: `sudo apt install inxi`
  - CentOS: `yum install w3m`

Modern versions of old commands

- `bat` instead of `cat`
- `httpie` instead of `wget` / `curl`
- `htop` instead of `top`
- `fd` instead of `find`
- `tldr` instead of `man`

Resources

- How do I…?
  - Google
  - Stack Overflow
- Learn stuff
  - Wikipedia
  - Github
  - `$ man bash` (and others…)

- My recommendation:
  - Keep a log file / cheat sheet