

Potpourri

Table of Contents

- [Keyboard remapping](#)
- [Daemons](#)
- [FUSE](#)
- [Backups](#)
- [APIs](#)
- [Common command-line flags/patterns](#)

Table of Contents (cont.)

- [Window managers](#)
- [VPNs](#)
- [Markdown](#)
- [Hammerspoon \(desktop automation on macOS\)](#)
- [Booting + Live USBs](#)
- [Docker, Vagrant, VMs, Cloud, OpenStack](#)
- [Notebook programming](#)
- [GitHub](#)

Keyboard remapping

- A key part of a programmer's productivity
- **Remapping keys:**
 - Caps Lock -> Ctrl or Escape
 - PrtSc -> Play/Pause music
 - Swap Ctrl and Meta (Windows/Command) key

Keyboard remapping (cont.)

- **Map keys to commands:**
 - Open terminal/browser
 - Insert specific text (email, ID number)
 - Sleep computer/displays
- **Complex modifications:**
 - Key sequences (e.g., shift five times for Caps Lock)
 - Tap vs hold remapping (e.g., Caps Lock -> Esc/Ctrl)
 - Keyboard or software specific remaps

Daemons

- Background processes: **daemons**
- Examples: `sshd` for SSH requests
- Linux: `systemd` for managing daemons
- `systemctl` to `enable`, `disable`, `start`, `stop`, `restart`, `status`
- **Example daemon configuration:**

```
## /etc/systemd/system/myapp.service
[Unit]
Description=My Custom App
After=network.target

[Service]
User=foo
Group=foo
WorkingDirectory=/home/foo/projects/mydaemon
ExecStart=/usr/bin/local/python3.7 app.py
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

- For scheduled tasks: `cron`

FUSE

- **Filesystem in User Space (FUSE)**
- Allows filesystems implementation by user programs
- **Examples:**
 - `sshfs` : Remote files via SSH
 - `rc1one` : Mount cloud storage like Dropbox, GDrive
 - `gocryptfs` : Encrypted overlay filesystem
 - `kbfs` : Distributed filesystem with encryption
 - `borgbackup` : Mount backups

Backups

- Data not backed up = at risk of being lost
- **Good backup basics:**
 - Off-site backups
 - Synchronization \neq backups (e.g., Dropbox/GDrive)
 - RAID \neq backups
- **Good backup features:**
 - Versioning
 - Deduplication
 - Security
- **Verify backups regularly!**

APIs

- Online services offer "APIs" for programmatic data access
- **Examples:**
 - US government weather forecasts via API
- **Common API traits:**
 - Structured URLs
 - Authentication often via tokens (e.g., OAuth)
- Tools like `jq` to handle JSON responses
- **IFTTT** for chaining API events

Common command-line flags/patterns

- `--help` for usage instructions
- Dry run and interactive flags for safety
- `--version` or `-V` to check the program version
- `--verbose` or `-v` for detailed output
- `-` for standard input/output
- `--` to stop processing flags (e.g., `rm -- -r`)

Window managers

- **Floating window managers:** Overlapping windows (e.g., Windows, macOS)
- **Tiling window managers:** Non-overlapping, arranged as tiles
 - Windows arranged according to a layout
 - Keyboard navigation and resizing
 - No mouse needed!

VPNs

- **VPN:** Changes your ISP as far as the internet is concerned
- **Considerations:**
 - Shift of trust from ISP to VPN provider
 - Encrypted traffic to VPN server
 - HTTPS already encrypts sensitive data
- **Risks:**
 - VPN misconfiguration
 - Malicious VPN providers
- **Alternatives:** MIT VPN, [WireGuard](#)

Markdown

- Lightweight markup language
- **Syntax:**
 - `*italics*`, `**bold**`
 - `## Heading`, `### Subheading`
 - `-` for bullet lists, `1.` for numbered lists
 - ``code`` for inline code, triple-backticks for code blocks
 - `[link text](url)` for hyperlinks
- Used everywhere, including this lecture's notes

Hammerspoon (desktop automation on macOS)

- **Hammerspoon:** Lua scripts for macOS automation
- **Examples:**
 - Hotkeys for window management
 - Menu bar buttons for layouts
 - WiFi network actions
- Extensive library for system interaction

Resources

- [Getting Started with Hammerspoon](#)
- [Sample configurations](#)
- [Anish's Hammerspoon config](#)

Booting + Live USBs

- BIOS/UEFI: Initial system setup
- Live USBs: OS on a USB flash drive
 - Useful for system recovery or testing
- Tools like [UNetbootin](#) to create Live USBs

Docker, Vagrant, VMs, Cloud, OpenStack

- **Virtual machines:** Emulate complete systems
- **Vagrant:** Machine configurations as code
- **Docker:** Container-based isolation
- **Cloud services:** AWS, Google Cloud, Azure, DigitalOcean
- **CSAIL OpenStack:** Free VMs for MIT CSAIL members

Notebook programming

- Interactive/exploratory development
- Popular environments:
 - [Jupyter](#) for Python and other languages
 - [Wolfram Mathematica](#) for math-oriented programming

GitHub

- Platform for open-source development
- **Contribution methods:**
 - Creating issues
 - Contributing code via pull requests
- Fork repositories, create branches, make changes, and create pull requests
- Engage with project maintainers and contribute to the community