Software I Use

General Workflow

Due to recent uncertainty of the long-term lifespan of OS X as it currently exists, I've been trying to move to more platform agnostic tools. As a long time OS X user, I have some strong opinions about the behavior of many of the other GUI libraries/platforms that exist. This has left me drifting towards command-line software rather than traditional GUIs. This post summarizes the tools I use daily, and what has made the migration of my workflow to the command line a lot easier.

> As a note, all of the software I use that has a command line interface (using curses/urwid/termbox/whatever) has the ability to interact with it using both keyboard and mouse. This was a very important aspect of user interfaces to me, so that got prioritized. Additionally, I picked software based on the way I saw their open source and communities get handled. I always try to choose things that are open and welcoming of newcomers, as well as respectful of all people that interact with the maintainers and software.

Shell

- fish – I've been using the default shell that came with OS X for a long time, up until I discovered fish a couple of years ago. This has been such a pleasant experience to use, both on the command line and when reporting bugs and contributing back.
- secure-env – This is a tool I made to encrypt and decrypt secrets that get stored in environment variables. It uses a backing sqlite data store for the encrypted data, but the encryption/decryption method is up to the user. I use GPG keys for the encryption.
- battery-level – Small tool for reporting current battery level on OS X machines. This interfaces with IOKit to query for active batteries that are attached and allows you to query them individually or by asking for the default. I use this as part of my shell prompt to show me how much battery is left as I run Terminal.app in full screen and don't always have that info available at a glance. In addition, I do use FruitJuice to help me track battery health.

- space – Another small utility I wrote to help me remember more complex shell commands that i would have to run often. This allows me to setup rules based on the directory i'm in to execute various actions without having to remember the whole command from memory. (Basically like a make-file without having to store it in-directory).
- clipper – This is a small tool that allows for better handling of the clipboard between local and remote hosts.

> Many people use **screen** or **tmux** to manage multiple applications running at once, I typically use additional terminal tabs. Even when dealing with remote hosts I find this a much better experience than juggling multiple levels of windowing abstraction.

Email

- getmail – I use **getmail** to retrieve mail from my mail-server (FastMail) over IMAP, also used with with Office365 over IMAP to fetch email on my work computer. This is a pretty simple and sturdy tool that interfaces with the system keychain to get your credentials and fetch all of your email and download it to a directory in a particular mailbox format (I use MailDir).
- notmuch – Instead of organizing email into folders, I organize with a tagging system. This is almost identical to folders in practice but can be much more powerful when performing searches as individual messages can be associated with multiple tags depending on contents, sender, etc.
- alot – This is a holistic interface for interacting with **notmuch** as a fully-featured email client. All of my normal interactions (reading, composing, replying, searching, etc) with email happen inside of this client.
- msmtp – Used for sending email, it interfaces with the system keychain to retrieve login credentials for authentication then will send a message to a specific SMTP server. This sounds a bit complicated for a causual user, but can be easily configured to work automatically from within **alot** to send messages for you.
- extract_url – Makes reading emails with embedded links "clickable" for the command line, extracts and opens URLs in your default browser.

> As a note, I only interact with email via my laptop these days. I have removed the Mail app from my iOS devices so I don't even get notifications of new mail. This also allows me to create a good (and healthy!) separation of work email vs personal email based on which computer I am working from.

Calendar

- vdirsyncer – In a similar vein to how my email is configured, I have a separate tool for fetching and synchronizing my calendars from viewing and editing them. This is a tool for syncing and managing interactions with CalDAV and CardDAV servers (as well as local ones). Using this tool I have configured it to fetch and update my iCloud-based calendars as well as syncing a number of other calendars I subscribe to.
- khal – This is an interface for the calendars provided by **vdirsyncer**. It is quite powerful and one of the best calendaring pieces of software I have seen. While this is my primary driver for calendars on my computers, I use Fantastical 2 for iOS on my phone and iPad.

Chat & Social Media

- weechat – This is a modern IRC client that I use in conjunction with running my own IRC bouncer. The IRC bouncer stores the credentials for, and connects to IRC and Slack servers for me. When combined with a tool like BitlBee, I can connect to a multitude of networks and services all from within one client. I use a set of scripts to help manage and interact with weechat, which are listed below:
    - mass_hl_blocker.pl – Automatically hides messages that consist of highlighting the names of many people in a particular room.
    - colorize_nicks.py – Applies colors to each person's nickname or handle so that tracking conversations can become easier.
    - go.py – Quickly jump between buffers/views.
    - pybuffer.py – A development tool to debug code against the weechat plugin API.
    - sdm_settings.py – This is a custom script that manages my personal configuration settings.
    - title.py – Automatically updates the title of the terminal window to match what room you are currently in.
    - unwanted_msg.py – Screens input to prevent sending commands as messages.
    - urlbuf.py – Creates a buffer that keeps a record of all URLs that get sent to any channel you are in.
    - whowas_timeago.py – Allows you to perform lookups of seen handles that are not currently active.
- rainbowstream – This is by no means a replacement twitter client for Tweetbot, but it is nice for causually watching a twitter stream.

Editor

- micro – Text editor, this is pretty standard but has some nice features with mouse support and a good Lua-based plugin API. I use a number of plugins to augment the features it already provides.
    - natural-navigation – (Work in Progress) Aims to give cursor based navigation a bit more of a natural feel that would be seen when editing text on OS X.
    - solarized-light – Color theme based on the solarized-light colors.
    - comment – Add and remove comment blocks from code.
    - fzf – Integration with **fzf** for quickly jumping between files within a directory.
    - wc – Get word count statistics on the current file (useful for writing prose, rather than code).
    - snippets – Insert code and text snippets automatically.
    - scratch – Plugin to automatically create scratch files to work out of without cluttering up an existing directory.

Data & Organizing

- buku – This is a tool for managing bookmarks that allows you to add additional comments and tags. There are a lot of documents that I like to bookmark for future reference and cannot seem to be able to find when I need them so I am trying this tool out to see if it fills my needs.
- omni – Small tool that will allow you to create and send items to your OmniFocus inbox from the command line.
- goaccess – This is a powerful tool for sorting through log files. I use it to collect and analyze the statistics from my apache webserver logs.
- jrnl – Command line journaling application. I was using this to store data before I started to try **buku**, would like something that is a combination of both of these tools but haven't found anything quite like that yet. I still use this for storing data (that isn't web-based) for future reference.

Security

- GPG – I use GPG exclusively on the command line now, but I got started by using the tools and integrations from GPGTools. This is marvelous software and you should support it.
- Keybase – Primary interface for interacting with GPG, gives access all the features provided by Keybase.

Utilities

- w3m – A pager (like **more** or **less**), but works with web-pages. Useful for viewing offline web content (such as html-based emails).
- dtrx – General purpose utility that will use the right decompression algorithm to extract the contents of a compressed file. Good for when you want to open something and don't want to remember how the flags to every decompression tool works.
- youtube-dl – Download videos for offline viewing, sometimes buffering is a pain and I want to watch something uninterrupted.
- fzf – A fuzzy picker, given a set of inputs it will allow you to search and select items from that list.
- the_platinum_searcher – Fast and powerful tool for searching through a code-base.
- since – Like **tail** but with state, it lets you view the contents of a file based on whatever was added since the last time you **since**'d the file.
- path-extractor – Takes text input then filters and returns any unix paths found, this makes scripts a lot easier without an ugly amount of grep/sed/awk.
- hotkeyd – Small daemon that allows you to register global hotkeys to run AppleScripts.
- httpie – A *much* better **curl**, no contest when writing code against a RESTful API.
- gistit – Small tool to upload and create gists from the command line.
- diff-so-fancy – Make diffs more readable by highlighting the changed contents and not entire lines.
- pandoc – Convert documents between any known type, extremely handy general utiltity (especially when you get sent word docs).
- tig – Small command line interface for interacting with git, I primarily use this for reading git history or examining diffs. I try to keep all of my git command line usage extremely simple (to minimize the fact that git is extremely bad). For any complex git operations I use Tower.