

The CLI; Command Line Interface

WHY?

- Speed; see Fitt's Law and HCI
- Wider range of devices; I don't want to be less productive on an unfamiliar computer
- Better remote access; RDP (and particularly VNC) will perform poorly on low bandwidth long before SSH

APPROACHES

- GUIs...
 - are build for a specific set of use cases
- Terminal Applications...
 - are components that can be chained together to solve a much wider range of use cases

APPLICATIONS

- Linux
 - Gnome Terminal
 - Kitty
- Windows
 - Windows Terminal
 - Putty
- Android: Juice SSH

SHELLS

"shells" are interpreters, not unlike `php` or `python`.

- emphasise interactive use
- trivialise the execution of binary executables

Popular varieties are...

- Windows: `pwsh`
- macOS: `zsh`
- Linux: `bash`

bash SCRIPTS

- A "script" is just a text file with a series of commands
- Those commands can be the execution of installed commands, or
- they can be keywords build into the shell
- Those commands are usually more useful when chained together

ALIASES

- Quick substitutions
- eg. `nl` does not number blank lines by default...

```
jdownie@smitty ~$ head -n 4 jdinst
#!/bin/bash

. ~/bin/lib.sh
. /etc/os-release
jdownie@smitty ~$ alias nl
alias nl='nl -ba'
jdownie@smitty ~$ head -n 4 jdinst | nl
 1  #!/bin/bash
 2
 3  . ~/bin/lib.sh
 4  . /etc/os-release
jdownie@smitty ~$ unalias nl
jdownie@smitty ~$ head -n 4 jdinst | nl
 1  #!/bin/bash

 2  . ~/bin/lib.sh
 3  . /etc/os-release
jdownie@smitty ~$
```

SOURCING

```
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # pwd
/tmp/demo
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # ls -l
total 4
-rw-rw-r-- 1 jdownie jdownie 24 Sep  5 13:33 setMonth
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # bat setMonth
```

	File: setMonth
1	M=`date +%b`
2	export M

```
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # echo $M

jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # source setMonth
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ # echo $M
Sep
jdownie@smitty ㉿ master /tmp/demo
❏ ❹ #
```


AUTOMATIC SOURCING

- `~/.bash_profile` on login
- `~/.bashrc` on non-login

PIPING

```
➤ 🔍 jdownie@smitty ~ main ! ~/bin
PKG=$1
if [ "$PKG" = "brew" ]; then
elif [ "$PKG" = "bw" ]; then
jdownie@smitty ~ main ! ~/bin
➤ 🔍 grep PKG jdinst | head -n 3 | grep "then$"
if [ "$PKG" = "brew" ]; then
elif [ "$PKG" = "bw" ]; then
jdownie@smitty ~ main ! ~/bin
➤ 🔍 grep PKG jdinst | head -n 3 | grep "then$" | cut -d '"' -f 4
brew
bw
jdownie@smitty ~ main ! ~/bin
➤ 🔍 grep PKG jdinst | head -n 3 | grep "then$" | cut -d '"' -f 4 | tr '\n' ' '
brew bw jdownie@smitty ~ main ! ~/bin
➤ 🔍 grep PKG jdinst | grep "then$" | cut -d '"' -f 4 | tr '\n' ' '
brew bw fonts git glow gogh keys kubect1 neovim node pwsh rtm rust starship viu z zoxide Waybar rofi stow syncthing yay A
uamarine Hyprland docker jdownie@smitty ~ main ! ~/bin
```

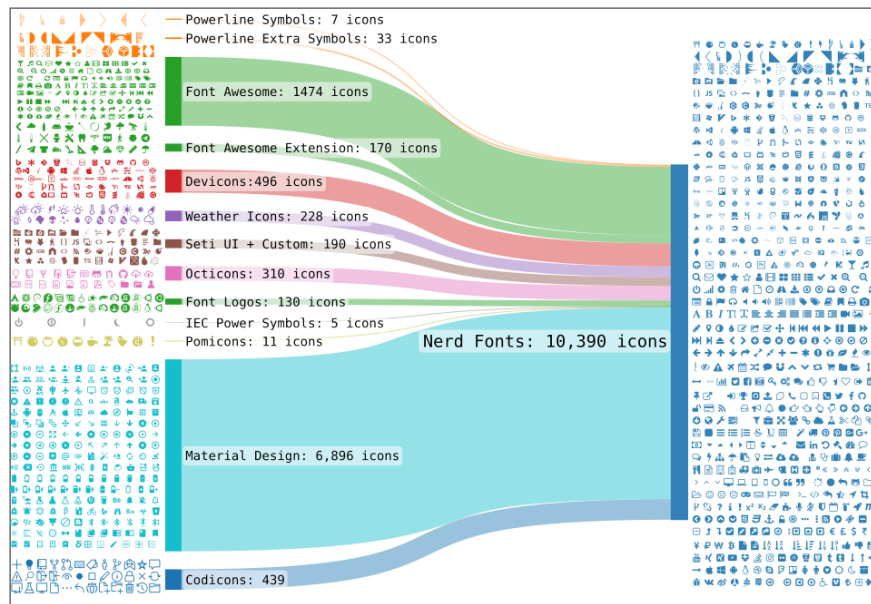
COSMETIC IMPROVEMENTS

STARSHIP

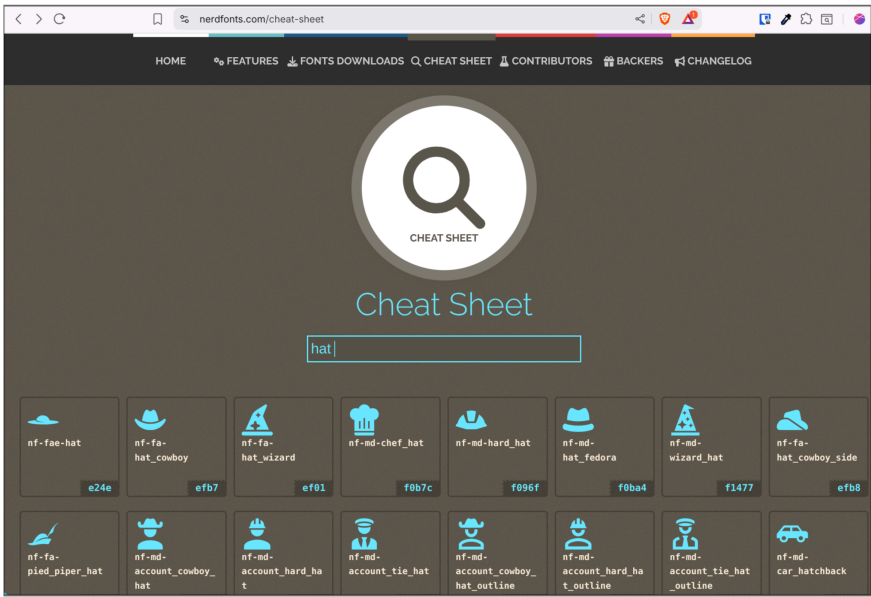
```
jdownie@smitty ~/dev/ConAI $ cd  
jdownie@smitty ~ $ |
```

```
jdownie@smitty 019 ~/dev/ConAI  
cd  
jdownie@smitty ~
```

NERD FONTS



NERD FONT CHEAT SHEET



NO NERD FONT?

```
jdownie@smitty  ~ /bin
┌─┐ ┌─┐ ┌─┐
└─┘ └─┘ └─┘
echo $STARSHIP_CONFIG
/home/jdownie/cfg/starship.toml
jdownie@smitty  ~ /bin
┌─┐ ┌─┐ ┌─┐
└─┘ └─┘ └─┘
sst
jdownie@smitty  ~ /bin
C U B ● echo $STARSHIP_CONFIG
/home/jdownie/cfg/starshipNoNerd.toml
jdownie@smitty  ~ /bin
C U B ● ls -l ~/cfg/starship*
-rw-rw-r-- 1.6k jdownie ubuntu  4 Sep 14:30 -- * /home/jdownie/cfg/starship.toml
-rw-rw-r-- 1.6k jdownie ubuntu 10 Sep 06:56 -- * /home/jdownie/cfg/starshipNoNerd.toml
jdownie@smitty  ~ /bin
C U B ●
```

SSH

- ...does lots of stuff!
- The defacto standard for securely connecting to remote computers.
- Both hosts and users have "keys" that attend to identification
- Host keys help to ensure that you're not connecting to a "man in the middle" host

BASIC SYNTAX

- `ssh 192.168.1.20`
- `ssh tdurden@192.168.1.20`
- `ssh tdurden@paperst.net.au`
- `ssh -p 2022 tdurden@paperst.net.au`

SSH CONFIG

- `~/.ssh/config` allows you to configure per host defaults...

```
Host h0k0 h0k1 h0k2 h0k3 h1k0 h1k1 h1k2 h1k3 h2k0 h2k1 h2k2 h2k3
  User ubuntu
  HostName %h.downie.net.au
  StrictHostKeyChecking no
  UserKnownHostsFile /dev/null
  # LogLevel QUIET
```

USER KEYS

- A user's keys are typically stored in the user's `.ssh` folder.
- (typically) the "public" key has a `.pub` extension, and the "private" key has no extension

```
jdownie@fry ~  
❯ cd .ssh/  
jdownie@fry ~/.ssh  
❯ ls -l  
lrwxrwxrwx@ - jdownie ubuntu 4 Jun 06:35 config -> ../cfg/ssh/config  
-rw-----@ 399 jdownie ubuntu 16 Feb 19:23 id_ed25519  
-rw-r--r--@ 95 jdownie ubuntu 16 Feb 19:23 id_ed25519.pub  
-rw-----@ 2.6k jdownie ubuntu 15 Jan 14:12 id_rsa  
-rw-r--r--@ 553 jdownie ubuntu 15 Jan 14:12 id_rsa.pub  
-rw-----@ 16k jdownie ubuntu 17 Sep 12:06 known_hosts  
-rw-----@ 15k jdownie ubuntu 17 Sep 12:06 known_hosts.old
```

PASSWORDLESS

- The `~/.ssh/authorized_keys` file on a remote host is a list of public keys to permit access
- The permission of that file needs to be `600` (`rw- - - - -`)
- `ssh-copy-id` will install your public key on a remote host for you

```
jdownie@smitty ~  
• • • whoami  
jdownie  
jdownie@smitty ~  
• • • hostname  
smitty  
jdownie@smitty ~  
• • • ssh-copy-id roberto  
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jdownie/.ssh/id_ed25519.pub"  
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
jdownie@roberto.downie.net.au's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'roberto'"  
and check to make sure that only the key(s) you wanted were added.  
  
jdownie@smitty ~  
• • • ssh roberto whoami  
jdownie  
jdownie@smitty ~  
• • • ssh roberto hostname  
roberto  
jdownie@smitty ~  
• • • |
```

COMMAND ALTERNATIVES

- There are lots of alternatives to commands that you might assume are "fixed" parts of Linux, eg...
 - `ls` -> `exa`
 - `zsh`: Enhanced `cd`
 - `batcat`: Enhanced `cat`

ls → exa

```
jdownie@smitty /tmp/demo
ls -l
total 20
-rw-r--r-- 1 jdownie jdownie 1031 Sep  5 12:37 file1.cs
-rw-r--r-- 1 jdownie jdownie 1031 Sep  5 12:37 file1.css
-rw-r--r-- 1 jdownie jdownie 1031 Sep  5 12:37 file1.html
-rw-r--r-- 1 jdownie jdownie 1031 Sep  5 12:37 file1.js
-rw-r--r-- 1 jdownie jdownie 1031 Sep  5 12:37 file1.sh
jdownie@smitty /tmp/demo
```

```
jdownie@smitty ~
alias ls
alias ls='exa --icons --group --git'
```

```
jdownie@smitty master ! /tmp/demo
ls -l
-rw-r--r-- 1.0k jdownie jdownie  5 Sep 12:37 -- 📄 file1.cs
-rw-r--r-- 1.0k jdownie jdownie  5 Sep 12:41 -M 📄 file1.css
-rw-r--r-- 1.0k jdownie jdownie  5 Sep 12:37 -- 📄 file1.html
-rw-r--r-- 1.0k jdownie jdownie  5 Sep 12:37 -- 📄 file1.js
-rw-r--r-- 1.0k jdownie jdownie  5 Sep 12:37 -- 📄 file1.sh
```

z.sh

- <https://github.com/rupa/z/>

```
jdownie@smitty ~  
❯ cd dev  
jdownie@smitty ~/dev  
❯ cd Orchestration  
jdownie@smitty ⚡ main ! ~/dev/Orchestration  
❯ cd deploy  
jdownie@smitty ⚡ main ! ~/dev/Orchestration/deploy  
❯ cd k3s  
jdownie@smitty ⚡ main ! ~/dev/Orchestration/deploy/k3s  
❯ cd  
jdownie@smitty ~  
❯ z k3s  
jdownie@smitty ⚡ main ! ~/dev/Orchestration/deploy/k3s  
❯
```


batcat

```
jdownie@smitty ~ main ! ~/bin
alias bat
alias bat='/usr/bin/batcat'
jdownie@smitty ~ main ! ~/bin
head brave | bat
```

	STDIN
1	#!/usr/bin/env python3
2	
3	import os, sys, json, pprint
4	
5	pp = pprint.PrettyPrinter(indent = 2)
6	
7	def find(start_dir, filename = None, dirname = None):
8	ret = None
9	for (root, dirs, files) in os.walk(start_dir):
10	if ret == None and filename != None and filename in files:

jdownie@smitty:-

x + - v

Every 2.0s: ls -l

total 20
-rw-r--r-- 1 jdownie jdownie 1031 Sep 5 12:37 file1.cs
-rw-r--r-- 1 jdownie jdownie 1033 Sep 5 12:41 file1.css
-rw-r--r-- 1 jdownie jdownie 1031 Sep 5 12:37 file1.html
-rw-r--r-- 1 jdownie jdownie 1031 Sep 5 12:37 file1.js
-rw-r--r-- 1 jdownie jdownie 1031 Sep 5 12:37 file1.sh

smitty: 12:44:54

in 0.033s (0)

LY CRUSHER SERVICE.docx'
conai-doc-1 | '/app/docx/samples/CF000 Fine Coal Centrifuge 8Wk SERV.docx' -> '/in/CF000
e Coal Centrifuge 8Wk SERV.docx'
conai-doc-1 | '/app/docx/samples/CR000 3 WEEKLY LUBRICATION FILTRATION LUBE.docx' -> '/in/
000 3 WEEKLY LUBRICATION FILTRATION LUBE.docx'
conai-doc-1 | Handing off from entrypoint to Flask...
conai-doc-1 | Starting threads...
conai-doc-1 | Starting uploadingThread...
conai-doc-1 | I should be trying to upload a document...
conai-doc-1 | Starting convertingThread...
conai-doc-1 | I should be trying to convert a document...
conai-doc-1 | None
conai-doc-1 | - docx /in/SC500 3 WEEKLY SCREEN MECH OFFLINE SERV.docx 1130196 True
conai-doc-1 | - docx /in/CR153 45 WEEKLY CRUSHER SERVICE.docx 679752 True
conai-doc-1 | Stopping threads...
conai-doc-1 | - docx /in/CF000 Fine Coal Centrifuge 8Wk SERV.docx 7725724 True
conai-doc-1 | - docx /in/CR000 3 WEEKLY LUBRICATION FILTRATION LUBE.docx 356680 True
conai-doc-1 | - pdf /in/HardPDF.pdf 194388 True
conai-doc-1 | That's enough convertingThread!
conai-doc-1 | That's enough uploadingThread!
conai-doc-1 exited with code 0
jdownie@smitty ↵ 019 ~/dev/conAI
A A A
Enable Watch

1 FROM docker.io/library/alpine:latest

2

3 RUN apk update && apk add python3 bash sudo postgresql-dev gcc python3-dev musl-dev

4 RUN addgroup -g 1000 conai

5 RUN adduser -D -u 1000 -G conai -h /home/conai conai

6 RUN echo "conai ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/010_nopasswd

7 RUN mkdir -v /app/secrets && chown -Rfv 1000:1000 /app/secrets

8 USER conai

9 WORKDIR /home/conai

10 RUN python3 -m venv /home/conai/.venv

11 COPY doc/requirements.txt requirements.txt

12 RUN . .venv/bin/activate && pip install --no-cache-dir -r requirements.txt

13 RUN echo ". /home/conai/.venv/bin/activate" >> /home/conai/.bashrc

14 WORKDIR /app

15 ENTRYPOINT ["bash", "/app/entrypoint"]

~

~

~

~

Dockerfile.doc

1,1

h vpn i docker 2 nvim

05 Sep 12:44:

distrobox

- You can (maybe should) "live" in a container environment.
- The underlying "bare metal" distro is important...
 - it's kernel is how you access your hardware (including your GPU)
 - Applications (particularly GUI apps) will perform better on bare metal
- You can make a mess of your container environment without compromising the stability of your bare metal installation

```
jdownie@smitty ~
jdownie@smitty ~
echo $PRETTY_NAME
Debian GNU/Linux 13 (trixie)
jdownie@smitty ~
apt --version
apt 3.0.3 (amd64)
jdownie@smitty ~

jdownie@smitty ~
echo $PRETTY_NAME
Fedora Linux 44 (Container Image Prerelease)
jdownie@smitty ~
yum --version | head -n 4
dnf5 version 5.2.17.0
dnf5 plugin API version 2.0
libdnf5 version 5.2.17.0
libdnf5 plugin API version 2.2
jdownie@smitty ~

jdownie@smitty ~
echo $PRETTY_NAME
Ubuntu 24.04.3 LTS
jdownie@smitty ~
apt --version
apt 2.8.3 (amd64)
jdownie@smitty ~

jdownie@smitty ~
echo $PRETTY_NAME
Arch Linux
jdownie@smitty ~
pacman -V

.--.
/  .-. ' .-. ' .-. ' .-.
\  '-. ' '-. ' '-. ' '-.
 '-.'

Pacman v7.0.0 - libalpm v15.0.0
Copyright (C) 2006-2024 Pacman Development Team
Copyright (C) 2002-2006 Judd Vinet

This program may be freely redistributed under
the terms of the GNU General Public License.

jdownie@smitty ~
pacman -V
```

CREATE

```
jdownie@smitty ~  
# distrobox create --name opensuse --image opensuse/tumbleweed  
Image opensuse/tumbleweed not found.  
Do you want to pull the image now? [Y/n]: y  
Resolved "opensuse/tumbleweed" as an alias (/etc/containers/registries.conf.d/shortnames.conf)  
Trying to pull registry.opensuse.org/opensuse/tumbleweed:latest...  
Getting image source signatures  
Copying blob 219e36c53a62 done |  
Copying config 347a6d54f3 done |  
Writing manifest to image destination  
347a6d54f38bffa6e8a1a93f4c8c98414216892cecd63a1ddf1834fa702903  
Creating 'opensuse' using image opensuse/tumbleweed [ OK ]  
Distrobox 'opensuse' successfully created.  
To enter, run:  
  
distrobox enter opensuse
```

distrobox

```
jdownie@smitty ~ $ echo $PRETTY_NAME
openSUSE Tumbleweed
jdownie@smitty ~ $ zypper --version
zypper 1.14.93
jdownie@smitty ~ $
```

DEFAULT OVER ssh

```
Host smitty
  HostName smitty.downie.net.au
  User jdownie
  RemoteCommand distrobox enter -n default -- bash -l
  RequestTTY yes
```

```
Logout
james.downie@COSOL-L227 ~
ssh smitty
Starting container
```

```
jdownie@smitty ~
#
```

BYPASSING DEFAULT

```
james.downie@COS0L-L227 ~  
ssh smitty.downie.net.au
```



DISTRO PACKAGES

```
jdownie@smitty ~ $ pacman -Q | wc -l
181
jdownie@smitty ~ $ cd bin/pkg/
jdownie@smitty ~/bin/pkg $ ls
arch                debian-11  debian-13  flatpak      pop-22.04  rpm-ostree  ubuntu-22.04  ubuntu-24.04  windows.ps
configuration.nix  debian-12  fedora-42  install.sh   rhel-10.0  ubuntu-20.04  ubuntu-23.10  windows.json
jdownie@smitty ~/bin/pkg $ ./install.sh
```

```
(10/10) Updating the info directory file...
Generating locales...
  en_AU.UTF-8... done
Generation complete.
jdownie@smitty ~/bin/pkg $ pacman -Q | wc -l
248
jdownie@smitty ~/bin/pkg $
```

HELPER SCRIPTS

~/bin/jdinst

- This is where i "bottle" my processes for installing/building apps
- Some of these (eg. neovim, Waybar, Hyprland) will compile from a cloned git repository
- Distrobox is useful for building because i can use the recommended build stack
brew bw fonts git glow gogh keys kubect! neovim node pwsh rtm rust starship viu z
zoxide Waybar rofi stow syncthing yay Aquamarine Hyprland docker

neovim UPDATE

```
nvim --version
NVIM v0.12.0-dev-1223+g2debe2f30a
Build type: RelWithDebInfo
LuaJIT 2.1.1753364724
Run "nvim -V1 -v" for more info
```

```
jdownie@smitty ~
❯ jdinstr neovim
nvim already exists
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
xclip is already the newest version (0.13-3).
xsel is already the newest version (1.2.1-1).
0 to upgrade, 0 to newly install, 0 to remove and 0 not to upgrade.
~/bld/neovim ~
mkdir -p ".deps"
/usr/bin/cmake -S /home/jdownie/bld/neovim//cmake.deps -B ".deps" -G "Unix Makefiles"
-- Found GNU Make at /usr/bin/gmake
-- CMAKE_BUILD_TYPE=RelWithDebInfo
-- Configuring done (0.3s)
```

```
nvim --version
NVIM v0.12.0-dev-1294+geb7c12d3e3
Build type: RelWithDebInfo
LuaJIT 2.1.1753364724
Run "nvim -V1 -v" for more info
jdownie@smitty ~
```

git:bin

- For my scripts, on all of my hosts
 - `dbox` for creating/entering Distrobox environments
 - `pkg/install.sh` for installing packages in Distrobox and bare metal environments
 - `jdinst` for installing other stuff
 - `.bashrc` and `.bash_profile` for setting aliases

git:cfg

- For my configurations, on all of my hosts
 - `starship.toml` for Starship
 - `repo.yaml` for my repositories
 - `shortcuts/` for my `.desktop` shortcuts
 - Lots of `.local` and `.config` files that i use `stow` to symlink to

EOF