Linux Tapas
A Series of Small Dishes to try for Linux

by Aaron Grothe

Introduction

Tapas?

Dave can't spell Potpourri without a spell checker and I'm not sure if it is the right word to use either. So I decided to go with Tapas.

Tapas is an appetizer or snack in Spanish cuisine. A bunch of small dishes are served. So this is a series of Linux Tapas.

Introduction (Continued)

If you have questions/comments please feel free to ask them anytime. You don't have to hold them until the end of the talk.

If there are other resources similar to these that you think might be useful to people please let the group know.

Hopefully this will be an interactive and productive session.

Gotty

Gotty allows you to share a terminal as a web application

Gotty isn't available in many distributions, natively packaged

So the typical install is to use the go packaging tools (need golang and git)

$ go get github.com/yudai/gotty

Gotty

Lets run a simple command

~/go/bin/gotty top # run top in a window

Lets fire up a web browser and hit the results

Gotty.

By default the session is read only

To change this use a "-w" - really potentially dangerous
Can add a "-r" to add a random value to the end of the URL
Can use the "-t" option to turn on tls requires generating a key in advance

Each session gets its own process. So to share a terminal it'd probably be easiest to use tmux on top of it.

Mostly use it myself to monitor things on a system that I might want to know.

Cheat

Cheat is a tool that shows additional information on the command line.

Cheat is available in the copr repos for Fedora, RHEL and CentOS

For Debian/Ubuntu and others you'll have to install it from the pip3 repos.

$ pip3 install cheat

Cheat

You also may have to set the CHEAT_PATH as well

$ export CHEAT_PATH=/usr/local/lib/python3.7/dist-packages/usr/share/cheat

Then a simple

$ cheat zfs # pulls up a simple cheatsheet for the command

Cheat.

You can also create your own cheatsheets as well by setting the CHEAT_USER_DIR and putting the documentation in there. It is currently in plain text, though markdown support has been requested.

An alternative to consider is to use curl with cheat.sh

$ curl cheat.sh/zfs

Cheat.sh and the contents of the cheat repo aren't in sync so it might be worthwhile to hit both of them.

Aha

Aha is available in most repos so you should be able to install it via the standard system utilities

apt-get install aha # ubuntu/debian
yum install aha # fedora,centos
arch install from aur repo

Aha converts terminal sequences to html codes

Aha.

Simple example

$ man lynx | aha > lynx.html

Creates an html friendly version of the lynx man page. Can also be used for things such as top, or anything else that does formatting.

jo

Jo is a utility that transforms input into json strings or tables. It is becoming a json world we're all just living in

Jo is available in most repos so again it is installable through distro

```
# apt-get install jo # ubuntu/debian
# yum install jo # fedora,centos (epel)
# install from aur repo
```

jo.

So lets take a simple example of a directory listing and turn it into a json object

$ jo -p -a *

Returns an array of objects

$ jo -p a=b c=d # will print a simple object with two fields

Useful if you need to programmtically want to create a json object simply.

Nativefier

Nativefier turns a website or webapp into a desktop application with an icon.

Nativefier wraps your app in a small electron app. Can be helpful for children/parents etc to make sure they can hit the sites that they want to.

Nativefier is a nodejs package so you use npm to install it

$ npm install nativefier

Nativefier

Nativefier is cross-platform so you can use it on windows/macosx. You can create packages for each.

We'll generate a "native" app to hit the OLUG website

$ nativefier http://www.olug.org

This creates a folder that contains all the needed files.

Lets go ahead and fire it up.

Nativefier.

A gentleman has made a complete electron app containing Windows 95 (with Doom) available.

This isn't anywhere near as amazing.

You can put in information such as basic auths and so on to provide seamless access to a site.

Currently you need several files in the directory. You can install them on the system or use another packaging system such as AppImage to turn the resulting package into a single app.

Cool-Retro-Term

Cool-Retro-Term is a nostalagic attempt to recreate the old style terminals you used to have back in the 70s and early 80s.

CRT is available from the OpenSuseBuild System, it is available in Arch repos and so on. For this example though we're going to use the AppImage version of it.

$ wget https://github.com/Swordfish90/cool-retro-term/releases/download/continuous/Cool_Retro_Term-6e4d5cf-x86_64.AppImage
$ chmod +x Cool*
$ ./Cool*

Cool-Retro-Term.

AppImage isn't the most efficient way to run the app but works very nicely for this demo. You can also download the sourcecode and build your own version as well if you'd prefer.

People have created custom fonts/color schemes for it as well to recreate an old environment.

Enjoy playing with it, but I get a headache after using it for a couple of minutes.

UP - The Ultimate Plumber

UP is a tool that allows you to dynamically create pipes

For this we're just going to download the executable, chmod +x it, and run it

$ wget https://github.com/akavel/up/releases/download/v0.3.2/up
$ chmod +x up

Might want to copy it to a more useful location such as /usr/local/bin or /usr/bin.

UP - The Ultimate Plumber

Lets fire it up

$ ./up

Error: up requires some data piped on standard input ….

So lets give it some data

$ ls /etc | ./up

Starts building up a dynamic run

UP - The Ultimate Plumber.

Lets add a couple of things to it

When satisfied hit Ctrl-X and it will write the command to a file named up1.sh, subsequent files will be up2.sh and so on

Lets take a look at the ./up1.sh we generated

Note: you still need to pipe input to it, but if you're playing with some complex pipelines, this can be very useful

Entr - a filewatcher

entr is a tool that allows you to watch files dynamically

Entr is available in most repos so you can install it via regular tools

```
# apt-get install entr # debian/ubuntu
# pacman -Sy entr # arch
# yum install entr # fedora/centos
```

Entr - a Filewatcher

We'll make a quick directory named hello and play around with a hello world program in it

```
$ mkdir hello
$ cd hello
$ vi a.c makefile
$ make
```

Lets run it and see the results. Yeah!!!

Entr.

Lets run an entr job to do an automated make every time one of the files changes

$ find ./ | entr -s 'make | head -n 20'

Now lets edit hello.c a bit

It is built again automatically and we'll run it now

Entr can do a lot of stuff, but I really haven't started to use it to much yet. Guess I'm too much of a cron kind of guy

Neofetch

Neofetch is a simple tool get system information

Neofetch is available in most repos so you can install it via regular tools

```
# apt-get install neofetch # debian/ubuntu
# pacman -Sy neofetch # arch
# yum install neofetch # fedora/centos
```

Neofetch.

We'll run it now and see the results

You can modify what information is displayed.

Setconf

Setconf is a simple package that is designed for modifying configuration files

Setconf is a python3 package so we'll use pip3 to install it. This time we'll just install it for this user

$ pip3 install setconf --user

Setconf

Lets modify the makefile from hello

```
CC=gcc
a.out: makefile a.c
    $(CC) a.c
```

$ make will compile with gcc

Lets change that to g++

Setconf CC g++

Setconf.

Setconf will modify lines in the following formats

CC=abc
Autostart := off
Build: true

Can you do this with sed? Sure. Setconf is designed to do one small task and do it well.

Tomb - simple encrypted filesystem

Tomb is a package for creating simple encrypted filesystems you can transport between systems. A bit like Truecrypt, Versacrypt, etc

apt-get install tomb # for Debian/Ubuntu

dnf copr enable blainester/tomb # for Fedora
dnf install tomb # for Fedora

pacman -Sy tomb # for Arch

Tomb - simple encrypted filesystem

We'll now create a simple 100mb system

```
$ tomb dig -s 100 olug.tomb
$ tomb forge olug.tomb.key
$ tomb lock olug.tomb -k olug.tomb.key
```

Now open it

```
$ tomb open olug.tomb -k olug.tomb.key
```

Now close it

```
$ tomb close
```

Tomb - simple encrypted filesystem

Running this on Debian Sid led to this not working :-(

This goes to show you that when you're working on the edge sometimes you'll have issues.

FD or FD-Find

FD is a find replacement for find

It is available for most systems

```
# apt-get install fd-find # note name change for package
# dnf install fd # for fedora or centos
# pacman -Sy fd
```

FD or FD-Find.

Why?

It can be a bit faster than find
It does things like color coding which are kind of nice

It has a built in version of the parallel command execution

E.g. compress all isos in a directory

$ fdfind -e iso -x bzip2

Bash Insulter

Well if you're like me and I know I am. You usually type a command wrong at least 10 times a day on average.

If you're tired of seeing the dreaded

"bash: maek: command not found"

You have a couple of options

Put in aliases for all 24 potential spellings of make
Or punish yourself with bash-insulter

Bash Insulter

Well if you're like me and I know I am. You usually type a command wrong at least 10 times a day on average.

If you're tired of seeing the dreaded

"bash: maek: command not found"

You have a couple of options

Put in aliases for all 24 potential spellings of make
Or punish yourself with bash-insulter

Bash Insulter

There are several versions of bash-insulter out there.

Bash insulter isn't available in many repos. So we'll grab a copy of it from the git repo

Clone repo to local machine

$ git clone https://github.com/hkbakke/bash-insulter.git bash-insulter

Copy to /etc

sudo cp bash-insulter/src/bash.command-not-found /etc/

Bash Insulter

Modify global bash.bashrc file

sudo vi /etc/bash.bashrc

And add the following lines to it

```
if [ -f /etc/bash.command-not-found ]; then
    . /etc/bash.command-not-found
fi
```

Bash Insulter.

Make file executable

sudo chmod +x bash-insulter/src/bash.command-not-found

Source the file in or logout/login

$ source /etc/bash.bashrc

Now it is time to give it a spin

$ lsdk

Resources

1. Copr in Lxer.com
2. Linux FU in hackaday
3. freshcode.club

Summary

There are a lot of tools out there.

You can learn a lot just by looking at freshcode.club, hackaday.com, packages.debian.org

A couple of them about looking into more are vimb (vim browser) and bashdb (bash debugger). Both of them are potentially useful to me.

Might be doing a part two of this where I try and get a couple of the tools into the repos for various distributions :-)

Q & A

Any Questions?

Thanks for listening.

Links

Gotty - https://github.com/yudai/gotty

Cheat - https://github.com/cheat/cheat

Aha - https://github.com/theZiz/aha

jo - https://github.com/jpmens/jo

nativefier - https://github.com/jiahaog/nativefier

Links.

cool-retro-term - https://github.com/Swordfish90/cool-retro-term

up - https://github.com/akavel/up

entr - http://eradman.com/entrproject/

neofetch - https://github.com/dylanaraps/neofetch

setconf - https://setconf.roboticoverlords.org/

fd - https://github.com/sharkdp/fd

Links

Bash-insulter: https://github.com/hkbakke/bash-insulter