Table of Contents

Linux Scripts

Over the years, I have build up several useful scripts to use on Linux systems. I'll try to document them here!

- backups
  - Borg
  - Dump and Backup a MySQL Database over SSH
  - Parity and Par2
  - TAR a system over SSH
- display
  - How to Enable Tearfree Mode (and what is it?)
- ffmpeg
  - Add (burn in) countdown timer to video
  - Change Video Gain (Brightness)
  - Convert 60i video to 60p using yadif (60 interlaced field per second video to 60 progressive frame per second video)
  - Convert 60p video to 60i (60 frame per second progressive video to 60 field per second interlaced)
  - Convert still image to a video
  - Volume Adjustment and Dynamic Range Compression using dynaudnorm and compand
  - Compress all FLAC (or MP3) files in subfolders to OPUS
  - Extract embedded 608 closed captions to subtitle file
  - Fix color space issues when converting from jpeg
  - Force a fixed frame rate
  - Interpolate video to higher framerate
  - Better, faster screen capture with kmsgrab
  - Merge/concatenate multiple video/audio files into one
  - Overlay Logo Over Video With Transparency
  - Scale video using black bars
  - Overlay videos side by side
  - Split a video file to multiple parts
  - Stream desktop over Icecast
  - Visualize music into notes
- files
  - Cannot unmount USB drive
  - Checksum entire folders
  - Compress a folder to tar.xz
  - Convert .bin and .cue files into a .iso
  - Create multi-session DVD and burn files
  - Deduplicate Files on BTRFS
  - Evenly Split Directories Into Chunks
  - Grep through PDF files
  - Merge multiple PDF's together
  - Mount SMB and NFS shares for a NAS with the fstab

  - OCR a PDF file
- i3 config
  - Audio Player Play/Pause/Skip/Back Controls
  - Sharp font for i3bar
  - Blue Light Filter using Redshift
  - End All Use of Program in i3
  - Lock Computer/System
  - Map Wacom (or other) Graphics Tablet to One Display
  - Disable Mouse Acceleration in i3
  - Notifications in i3
  - Volume Control Using i3
- media
  - Add Reverb to an Audio File Using SOX
  - Batch-convert files to JXL
  - Change Sample Rate of Pipewire
  - GStreamer VAAPI with Full Range Color
- rsync
  - Use Rsync to do a full one-way sync
- sway
  - Dim Backlight on a Laptop with Sway after a Timeout
  - How To Enable Freesync and 10 Bit Colour Support
  - How To Map a Wacom To A Single Output Using Sway
- wayland
  - Blue Light Filter on Wayland
  - How To Forward Wayland Applications over SSH
  - Run games in HDR
- Backups and Data
- LaTeX
- Desktop Experience
- Useful Programs
- Working with Websites

Any other unsorted stuff will be here!

Misc commands

Tearfree mode

Tearfree Mode

Run all executable in a folder

Sometimes it is useful to run all executables in a folder. Rather then looping, if you have debianutils or busybox, you can use the dedicated run-parts command.

Note the slightly different syntax between the debian and busybox ones. busybox can't do –report or -v to print the name of all files as they are run.

```
run-parts .
```

Where the . is the folder to run executables from.

Make sure you test what will be run as follows

```
run-parts --test
```

IIRC, you can also pass parameters to the files being run. This may be useful for running test executables.

Boot Linux to serial port

For many people, this would fall within the Why? section. For the 1 homelab sysadmin reading it, this may be useful.

Essentially, Linux by default boots to /dev/tty0, which is generally the main monitor and console. However, you can have it print all the systemd messages elsewhere, such as a serial port. This is useful if you have a motherboard with IPMI that supports serial port redirection but has broken KVM (*cough* supermicro x9 boards needing some java version for kvm *cough*).

This can be beter then running sshd or x2go on startup, because if you have a special boot process (such as full disk encryption), then you need to boot to be able to start those services.

To do this, just add the console to the kernel boot parameters:

```
console=ttyS1,9600
```

The first parameter is the console name. Serial ports start with /dev/ttyS0, then S1, so on. Like COM ports on Windows. The 2nd parameter is the baud rate, ie: speed. 9600 is quite slow, and any lower would take too long to boot, however this is a common speed that's almost guaranteed to be supported.

If you don't know where to add this, see you GRUB or REFIND or config of your bootloader. It's the line that includes

```
initrd=/initramfs-linux-zen.img
```

Mount Seagate external drive without UAS

Normally, you can't run smartctl on a Seagate external drive since they use UAS (USB attached SCSI), which might make it a bit faster for planar drives, but smartctl doesn't support reading SMART data over UAS.

First, unmount and unplug the drive if it is in use

Then, remove the usb-storage and uas modules

```
sudo modprobe -r usb-storage
sudo modprobe -r uas
```

(or alternatively, use rmmod)

```
sudo rmmod usb-storage
sudo rmmod uas
```

Next, before you plug the drive in (which will auto-load uas), you need to load the usb-storage module with the quirk to disable uas for your Seagate drive.

To do this, get the vendor and product id by doing lsusb (the xxxx:yyyy stuff)

Finally, reload the module with the quirk as follows

```
sudo modprobe usb-storage quirks=0bc2:331a:u
```

Where the xxxx:yyyy is the vendor and product id for your drive, and the :u tells it to not use uas.

Map Wacom tablet to one display

Moved to Map Wacom (or other) Graphics Tablet to One Display

Why?

Things that make you question why this was implemented.

Play music from motherboard speaker

Have you wanted to relive the experience of listening to music through a tin can or the audiophile grade (lol) 70v ceiling speakers installed at your old school? Well, the kernel has got the solution for you!

Some custom computers and most prebuilt computers have a small onboard speaker or buzzer on the motherboard. This is usually very handy for debugging as it can produce a range of tones to help troubleshoot problems. On Linux, the bell input and some escape characters normally trigger the buzzer. I also use the beep command to signal notifications with a range of tones. It's like the old RGB indicator on phones, where you can tell what program just fired a notification from the sound (like you could based on the colour of the LED).

```
beep
```

Of course, people have already compiled a range of melodies for this, and you can find it here https://github.com/ShaneMcC/beeps

But, what if that's not enough, and you want to use the buzzer as a speaker? By default the kernel loads the pcspkr module for using the speaker/buzzer. There is however another module snd-pcsp that can mount the speaker as an ALSA output, allowing PulseAudio to use it as a sound output.

```
sudo rmmod pcspkr
sudo modprobe snd-pcsp
```

Finally, bring up pavucontrol and enable the mono output. You mau need to unmute it first from alsamixer as well. Then, you can simply set a program like MPD to use it as an output and enjoy your music in what may be usable quality if your MB has a speaker, and horrifying quality that sounds like a dying fire alarm if your MB has a buzzer. (tip: change the base frequency to be above the range where you can hear it).

todo: attach a recording of the sound quality

Have fun!

From:
https://wiki.tonytascioglu.com/ - **Tony Tascioglu Wiki**

Permanent link:
https://wiki.tonytascioglu.com/scripts

Last update: **2022-06-03 20:39**