

# Interfaces

- Audio Devices
- Disk Management
- Wireless
- Bluetooth
- System Sensors

# Audio Devices

When using various Linux Distributions, you may (or may not) run into some issues with audio devices. See some of the configs, logs, and commands below for helpful output in troubleshooting these issues.

## GUI Tools / Applications

If looking for a GUI Tool to select or view output / input audio devices, check out `pavucontrol` -

```
sudo pacman -Syu pavucontrol
pavucontrol
```

will install and open the application, which provides a simple interface for selecting audio devices, and even provides application-level audio control, which enables you to easily specify the devices for individual applications instead of forcing a system-wide audio setting for all running apps.

## Commands

:)

## Sound Card / Devices

Search for all connected audio cards, and output the result.

```
aplay -L | grep :CARD
```

List all connected PCI devices (Sound cards are a PCI device)

```
lspci
```

## Audible Sound Test

The command below will send static to each speaker connected to the device, sequentially, one at a time. Running this will continually test all speakers on a loop, until the user exits with `CTRL+C`.

```
speaker-test -D default:PCH -c 8
```

The output from the above test will look similar to the below, depending on your system and devices.

The `-D` argument specifies the audio device you want to test. This is useful when not entirely sure which device is valid, you can test quickly with this cmd and make changes later in `alsamixer` or another config tool with the results of your findings.

The `-c` argument specifies the number of audio channels you want to test, for my setup I only have a front left and right speaker, so 2 will suffice. If I had a surround sound with Left / Right speakers in the back and an additional center speaker, we would test over 5 channels.

```
[kapper@kapper-pc ~]$ speaker-test -D default:PCH -c 2
```

```
speaker-test 1.1.9
```

```
Playback device is default:PCH
```

```
Stream parameters are 48000Hz, S16_LE, 2 channels
```

```
Using 16 octaves of pink noise
```

```
Rate set to 48000Hz (requested 48000Hz)
```

```
Buffer size range from 2048 to 16384
```

```
Period size range from 1024 to 1024
```

```
Using max buffer size 16384
```

```
Periods = 4
```

```
was set period_size = 1024
```

```
was set buffer_size = 16384
```

```
0 - Front Left
```

```
1 - Front Right
```

```
Time per period = 5.648263
```

```
0 - Front Left
```

```
1 - Front Right
```

```
Time per period = 5.973649
```

```
0 - Front Left
```

```
^CWrite error: -4,Interrupted system call
```

```
xrun_recovery failed: -4,Interrupted system call
```

```
Transfer failed: Interrupted system call
```

## Sound Mixer / Settings

To open alsa mixer, run the below and use the `F6` key to ensure the proper device is selected. This tool can also be used to change volume levels, be careful messing with settings you are unfamiliar with, you could easily blow a speaker. At the least, connect a cheaper pair.

```
alsamixer
```

To check device audio settings / levels via CMD -

`amixer` to list devices and settings

`amixer sset Master unmute` to mute the Master device. Master can be changed to any valid device name given from the output of `amixer`

Also, see [Advanced Linux Sound Architecture](#) for more information on various documented issues encountered.

# Disk Management

Show all disks, usage, and format type

```
sudo df -T -h
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
udev	devtmpfs	930M	4.0K	930M	1%	/dev
tmpfs	tmpfs	191M	1.5M	190M	1%	/run
/dev/sda1	fuseblk	29G	25G	4.5G	85%	/isodevice
/dev/loop0	iso9660	1.6G	1.6G	0	100%	/cdrom
/dev/loop1	squashfs	1.5G	1.5G	0	100%	/rofs
/cow	overlay	22G	17G	4.1G	81%	/
tmpfs	tmpfs	954M	5.2M	949M	1%	/dev/shm
tmpfs	tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	tmpfs	954M	0	954M	0%	/sys/fs/cgroup
tmpfs	tmpfs	954M	56K	954M	1%	/tmp
tmpfs	tmpfs	191M	8.0K	191M	1%	/run/user/999
tmpfs	tmpfs	191M	20K	191M	1%	/run/user/70000
google-drive-ocamlfuse	fuse.google-drive-ocamlfuse	15G	9.1G	6.0G	61%	/home/kapper/gdrive

Check /var directories for disk usage, sort and limit results to 10

```
sudo du -ah /var | sort -nr | head -n 10
```

```
924K  /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_disco-updates_universe_i18n_Translation-en
924K  /var/cache/apparmor/26b63962.0/usr.lib.libreoffice.program soffice.bin
912K  /var/lib/texmf/web2c/pdftex/pdflatex.fmt
912K  /var/lib/texmf/web2c/pdftex/latex.fmt
888K  /var/cache/apt/archives/libgtkmm-3.0-1v5_3.24.0-2_amd64.deb
852K  /var/lib/apt/lists/security.ubuntu.com_ubuntu_dists_disco-security_main_i18n_Translation-en
828K  /var/lib/app-info/icons/ubuntu-disco-multiverse
824K  /var/lib/dpkg/info/linux-headers-5.0.0-38-generic.md5sums
817K  /var/lib/dpkg/info/linux-headers-5.0.0-13-generic.md5sums
804K  /var/log/syslog.3.gz
```

Scan this disk for usage, sort the results by directories > 1.0GB, show largest 5 results

```
sudo du -xh / | grep '^S*[0-9\\.]+G' | sort -rn | head -n 5
```

```
19G    /
9.8G   /home/kapper
9.8G   /home
6.7G   /usr
5.7G   /home/kapper/.cache
```

Show the largest 5 files on the system (greater-than 100MB), using block size of 1MB.

```
sudo find / -xdev -type f -size +100M -exec ls -l --block-size=M {} \; | sort -nk 5 -r | head -n 5
```

```
-rw-r--r-- 1 kapper kapper 174M Nov 30 06:13 /home/kapper/.local/share/JetBrains/Toolbox/apps/CLion/ch-
0/213.5744.254/lib/platform-impl.jar
-rw-r--r-- 1 kapper kapper 174M Nov 30 03:14 /home/kapper/.local/share/JetBrains/Toolbox/apps/PyCharm-P/ch-
0/213.5744.248/lib/platform-impl.jar
-rw-r--r-- 1 kapper kapper 174M Nov 27 05:41 /home/kapper/.local/share/JetBrains/Toolbox/apps/WebStorm/ch-
0/213.5744.224/lib/platform-impl.jar
-rw-r--r-- 1 kapper kapper 174M Nov 23 11:21 /home/kapper/.local/share/JetBrains/Toolbox/apps/datagrip/ch-
0/213.5744.178/lib/platform-impl.jar
-rw-r--r-- 1 kapper kapper 174M Dec  1 08:26 /home/kapper/.local/share/JetBrains/Toolbox/apps/Goland/ch-
0/213.5744.269/lib/platform-impl.jar
```

Show the 10 files consuming the most data on this system

```
sudo find / -printf '%s %p\n' | sort -nr | head -10
```

```
140737477885952 /proc/kcore
24244125696 /isodevice/casper-rw
3296907264 /media/lubuntu/37aba99c-8b85-4ddc-92eb-6f50251041e8/encrypted.block
1890263040 /media/lubuntu/37aba99c-8b85-4ddc-92eb-
6f50251041e8/home/.shadow/362074638d2508061facd43743c9f08ff66866b8/mount/ASTjdxE5eNzc0F3IkW870B
/4HICtmqSNBNCh4oi+U7116prbiG
1657700352 /isodevice/lubuntu-19.04-desktop-amd64.iso
1589342208 /cdrom/casper/filesystem.squashfs
471728128 /media/lubuntu/37aba99c-8b85-4ddc-92eb-
6f50251041e8/home/.shadow/362074638d2508061facd43743c9f08ff66866b8/mount/ASTjdxE5eNzc0F3IkW870B
/MK3nIrbwT+ZzY8n1fczEqB/Yqt2q,akFt0Uj7WDNJaHdNA0GbKtnhX7kza2zeVnGMI/E1hEyvd6NZ1+JT6hgXl2zA/JekS
pYOtLWQPv0kgorVJuFbCcIG/LRriMSOWaunntY7RsNuiUC/x+gYqqpRFtEjXG+JzztvwIQ4LDeq82QY
```

```
268435456 /sys/devices/pci0000:00/0000:00:02.0/resource2_wc
268435456 /sys/devices/pci0000:00/0000:00:02.0/resource2
137797651 /home/kapper/.local/share/Steam/ubuntu12_64/libcef.so
```

Print information on all connected block devices

```
sudo lsblk

sda      8:0    0 931.5G  0 disk
├─sda1   8:1    0  128M  0 part
├─sda2   8:2    0 925.5G  0 part
└─sda3   8:3    0   5.9G  0 part
```

Print the UUIDs of all connected block devices, along with some other hardware information

```
sudo blkid

/dev/sdb2: UUID="436b3ae3-4301-4b8a-80d3-fdf52c7d7059" TYPE="swap" PARTUUID="590670f6-3b89-41b5-b474-fcd6c048628d"
```

Print information on partitions on all connected block devices

```
sudo parted -l

Model: HDD A12345678-B3210 (scsi)
Disk /dev/sdb: 2000GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number Start   End     Size    File system  Name  Flags
 1    1049kB 1000MB  999MB   fat32              boot, esp
 2    1000MB 17.0GB 16.0GB  linux-swap(v1)
 3    17.0GB 117GB  100GB   ext4
 4    117GB  217GB  100GB   ext4
 5    217GB  427GB  210GB   ext4
```

Print information given a specific block device (partitions)

```
sudo tune2fs -l /dev/sdb3
```

tune2fs 1.45.4 (23-Sep-2019)

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: fagbraetd325t9-6gafdee7-4d2344agdd-93d2-6f4safsa5d6

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem features: has\_journal ext\_attr resize\_inode dir\_index filetype needs\_recovery extent 64bit flex\_bg sparse\_super large\_file huge\_file dir\_nlink extra\_isize metadata\_csum

Filesystem flags: signed\_directory\_hash

Default mount options: user\_xattr acl

Filesystem state: clean

Errors behavior: Continue

Filesystem OS type: Linux

Filesystem created: Mon Oct 14 12:34:48 2019

Last mount time: Thu Oct 24 12:27:30 2019

Last write time: Thu Oct 24 12:27:30 2019

Mount count: 50

Maximum mount count: -1

Lifetime writes: 50 GB

Default directory hash: half\_md4

Directory Hash Seed: 4e7gds499c3-8532e0-452356c-432890c-d0fds43e2be81ee

Journal backup: inode blocks

Checksum type: crc32c

Checksum: 0xb054235dk

# Wireless

I didn't end up having luck with `iw`, but I'm sure it is very useful. It seems I just wasn't able to interactively enter a password, so in the end I couldn't connect to WiFi. Worth looking at `iw` though.

```
sudo iw dev wlp0s20f3 scan
sudo iw dev wlp0s20f3 scan | grep SSI
sudo iw dev
sudo iw list
sudo iw wlp0s20f3 connect "Reed WIFI-2G"
```

See examples in `man nmcli-examples`. A lot of good information between this page and the `SEE ALSO` section at the bottom.

## Network configurations

```
tree /etc/NetworkManager/
.
├── conf.d
│   └── default-wifi-powersave-on.conf
├── dispatcher.d
│   ├── 01-ifupdown
│   ├── 99tlp-rdw-nm
│   ├── no-wait.d
│   ├── pre-down.d
│   └── pre-up.d
├── dnsmasq.d
├── dnsmasq-shared.d
├── NetworkManager.conf
└── system-connections
    ├── Mi Casa.nmconnection
    ├── FAKE WIFI-2G.nmconnection
    └── FAKE WIFI-5G.nmconnection
```

8 directories, 7 files

**Terminal NetworkManager UI** made using curses library can be installed and ran with the following commands

```
sudo apt install network-manager
nmtui
```

**Gnome NetworkManager GUI** for editing wireless and bluetooth connections using a GUI application build for Gnome desktops

```
sudo apt install network-manager-gnome
nm-connection-editor
```

**Wifi can be toggled** with `wifi on` and `wifi off`

```
wifi on
```

```
wifi = on
```

```
rfkill
```

```
ID TYPE    DEVICE    SOFT    HARD
0 wlan     phy0      unblocked unblocked
1 bluetooth hci0      unblocked unblocked
```

**Connecting to WiFi**

```
nmcli device wifi list
```

IN-USE	BSSID	SSID	MODE	CHAN	RATE	SIGNAL	BARS	SECURITY
*	40:B8:9A:D7:EC:AF	FAKE WIFI-2G	Infra	1	195 Mbit/s	100	<div></div>	WPA2
	40:B8:9A:D7:EC:B0	FAKE WIFI-5G	Infra	149	405 Mbit/s	94	<div></div>	WPA2
	FA:8F:CA:95:43:9B	Living Room	Infra	6	65 Mbit/s	75	<div></div>	--
	FA:8F:CA:82:9D:D4	Family Room TV.b	Infra	6	65 Mbit/s	57	<div></div>	--
	14:ED:BB:1F:44:6D	Hi	Infra	8	130 Mbit/s	57	<div></div>	WPA2
	14:ED:BB:1F:44:76	ATT9eu7M6L	Infra	149	540 Mbit/s	44	<div></div>	WPA2
	4C:ED:FB:AD:D8:08	Fluffymarshmallow	Infra	1	540 Mbit/s	30	<div></div>	WPA2
	70:77:81:DE:43:59	WIFIDE4355	Infra	1	195 Mbit/s	24	<div></div>	WPA2
	70:5A:9E:6C:D4:29	TC8717T23	Infra	6	195 Mbit/s	19	<div></div>	WPA2
	A8:A7:95:E8:68:82	Wildflower-2G	Infra	1	195 Mbit/s	14	<div></div>	WPA2
	CC:2D:21:57:E0:71	Rudy	Infra	6	130 Mbit/s	14	<div></div>	WPA1 WPA2
	CE:A5:11:3C:E4:C2	Orbi_setup	Infra	9	130 Mbit/s	14	<div></div>	--
	A8:6B:AD:EB:B4:56	Gypsy-2	Infra	6	195 Mbit/s	12	<div></div>	WPA1 WPA2
	CE:A5:11:3C:EF:8E	Orbi_setup	Infra	9	130 Mbit/s	12	<div></div>	--

Now bring up a connection with the access point we want, and pass the `--ask` flag to enter a password for authentication.

```
nmcli c up "FAKE WIFI-2G" --ask
```

Passwords or encryption keys are required to access the wireless network 'FAKE WIFI-2G'.

Password (802-11-wireless-security.psk): ••••••••••••••••••••

Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/9)

## Disable transmission devices with `rfctl`

```
rfkill list
```

0: phy0: Wireless LAN

Soft blocked: no

Hard blocked: no

1: hci0: Bluetooth

Soft blocked: yes

Hard blocked: no

## Block WiFi

```
rfkill block wlan
```

## Block Bluetooth

```
rfkill block bluetooth
```

# Bluetooth

You can use `bluetoothctl` and `bluetooth` to control bluetooth devices

To check the status of bluetooth

```
rfkill
```

ID TYPE	DEVICE	SOFT	HARD
0 wlan	phy0	unblocked	unblocked
1 bluetooth	hci0	blocked	unblocked

To turn bluetooth on (replace on with off to turn bluetooth off)

```
bluetooth on
```

```
bluetooth = on
```

```
kapper@xps:~/dot$ rfkill
```

ID TYPE	DEVICE	SOFT	HARD
0 wlan	phy0	unblocked	unblocked
1 bluetooth	hci0	unblocked	unblocked

```
rfkill
```

ID TYPE	DEVICE	SOFT	HARD
0 wlan	phy0	unblocked	unblocked
1 bluetooth	hci0	unblocked	unblocked

To scan and connect to devices, run `bluetoothctl` to enter a bluetooth shell

```
bluetoothctl
```

```
Agent registered
```

```
[bluetooth]#
```

Now, we can start a scan with `scan on`

```
[bluetooth]# scan on
Discovery started
[CHG] Controller AC:74:B1:85:27:98 Discovering: yes
[NEW] Device 6A:0C:07:6A:09:EC Inspire HR
[NEW] Device 48:FE:3D:EB:C8:C3 48-FE-3D-EB-C8-C3
[NEW] Device EB:28:A2:3E:99:3F One
```

After scanning for some time, type `devices` to see the devices discovered in a list. While doing this, we can stop the scan so our output isn't messed with.

```
[bluetooth]# scan off
Discovery stopped
[CHG] Controller AC:74:B1:85:27:98 Discovering: no
[CHG] Device 6B:98:C9:C1:86:6C RSSI is nil
[CHG] Device 59:A5:50:BA:7E:4E RSSI is nil
[CHG] Device 66:05:2D:A4:AF:D2 RSSI is nil
[CHG] Device 50:32:37:84:CB:D4 TxPower is nil
[CHG] Device 50:32:37:84:CB:D4 RSSI is nil
[CHG] Device 03:0D:0F:0F:E9:51 RSSI is nil
[CHG] Device 6A:81:34:01:76:C0 RSSI is nil
[CHG] Device EB:28:A2:3E:99:3F TxPower is nil
[CHG] Device EB:28:A2:3E:99:3F RSSI is nil
[CHG] Device 48:FE:3D:EB:C8:C3 RSSI is nil
[CHG] Device 6A:0C:07:6A:09:EC RSSI is nil
```

```
[bluetooth]# devices
Device 50:32:37:84:CB:D4 50-32-37-84-CB-D4
Device 90:DD:5D:98:3A:E7 90-DD-5D-98-3A-E7
Device F9:EB:78:07:17:4B Dell Keybd KB7221W
Device 28:11:A5:34:08:2C Dumbo
Device 34:82:C5:F8:04:F3 Sam
Device E6:4E:7A:3F:FD:E7 Dell Mouse MS5320W
Device F9:EB:78:08:17:4B Dell Keybd KB7221W
Device E6:4E:7A:57:FD:E7 Dell Mouse MS5320W
Device F9:EB:78:04:17:4B Dell Keybd
Device 6A:0C:07:6A:09:EC Inspire HR
Device 48:FE:3D:EB:C8:C3 48-FE-3D-EB-C8-C3
Device EB:28:A2:3E:99:3F One
Device 6A:81:34:01:76:C0 Family Room TV
```

Now, if we want to pair, simply type `pair` followed by the ID for the device

```
[bluetooth]# pair F9:07:78:DA:17:4B
Attempting to pair with F9:07:78:DA:17:4B
[CHG] Device F9:07:78:DA:17:4B Connected: yes
[agent] Passkey: 221692
[NEW] Primary Service (Handle 0x4461)
/org/bluez/hci0/dev_F9_07_78_DA_17_4B/service000a
00001801-0000-1000-8000-00805f9b34fb
Generic Attribute Profile
[NEW] Primary Service (Handle 0x4461)
/org/bluez/hci0/dev_F9_07_78_DA_17_4B/service000b
0000180a-0000-1000-8000-00805f9b34fb
Device Information
[NEW] Characteristic (Handle 0x4461)
/org/bluez/hci0/dev_F9_07_78_DA_17_4B/service000b/char000c
00002a29-0000-1000-8000-00805f9b34fb
Manufacturer Name String
[NEW] Characteristic (Handle 0x4461)
/org/bluez/hci0/dev_F9_07_78_DA_17_4B/service000b/char000e
00002a50-0000-1000-8000-00805f9b34fb
PnP ID
[CHG] Device F9:07:78:DA:17:4B UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device F9:07:78:DA:17:4B UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device F9:07:78:DA:17:4B UUIDs: 0000180a-0000-1000-8000-00805f9b34fb
[CHG] Device F9:07:78:DA:17:4B UUIDs: 0000180f-0000-1000-8000-00805f9b34fb
[CHG] Device F9:07:78:DA:17:4B UUIDs: 00001812-0000-1000-8000-00805f9b34fb
[CHG] Device F9:07:78:DA:17:4B ServicesResolved: yes
[CHG] Device F9:07:78:DA:17:4B Paired: yes
Pairing successful
[CHG] Device F9:07:78:DA:17:4B Name: Dell Keybd KB7221W
[CHG] Device F9:07:78:DA:17:4B Alias: Dell Keybd KB7221W
[CHG] Device F9:07:78:DA:17:4B Modalias: usb:v413Cp2511d0001
[Dell Keybd ]#
```

This device just happens to be a keyboard, so I'm asked to type the pascode `221692` on the keyboard, then press enter. Once I do this, the pair is completed and the devices are paired.

Next time you enable bluetooth with `bluetooth on`, and then you turn on this keyboard, the devices will automatically attempt to connect.

# System Sensors

Your system likely has many sensors built in for displaying useful information on internal hardware status. For example, the commands below will help in finding the path to system temperature sensors.

```
user@host ~ $:sensors -f
coretemp-isa-0000
Adapter: ISA adapter
Package id 0: +80.6°F (high = +176.0°F, crit = +212.0°F)
Core 0:      +73.4°F (high = +176.0°F, crit = +212.0°F)
Core 1:      +73.4°F (high = +176.0°F, crit = +212.0°F)
Core 2:      +69.8°F (high = +176.0°F, crit = +212.0°F)
Core 3:      +68.0°F (high = +176.0°F, crit = +212.0°F)

acpitz-acpi-0
Adapter: ACPI interface
temp1:      +82.0°F (crit = +221.0°F)
temp2:      +85.6°F (crit = +221.0°F)

nouveau-pci-0100
Adapter: PCI adapter
GPU core:   +0.97 V (min = +0.60 V, max = +1.27 V)
fan1:       691 RPM
temp1:      +89.6°F (high = +203.0°F, hyst = +37.4°F)
              (crit = +221.0°F, hyst = +41.0°F)
              (emerg = +275.0°F, hyst = +41.0°F)
power1:     36.13 W (crit = 275.00 mW)

asus-isa-0000
Adapter: ISA adapter
cpu_fan:    0 RPM
```

We can see that the CPU and GPU temperature sensors are known to our system as `coretemp-isa-0000` and `nouveau-pci-0100`, respectively. Run the command below to list the system path to all connected temperature devices by name, and cross-check these two outputs to gather the needed information for your sensors.

```
user@host ~ $:for i in /sys/class/hwmon/hwmon*/temp*_input; do echo "$(<$(dirname $i)/name): $(cat  
${i%_*}_label 2>/dev/n  
ull || echo $(basename ${i%_*})) $(readlink -f $i)"; done
```

```
acpitz: temp1 /sys/devices/virtual/thermal/thermal_zone0/hwmon0/temp1_input  
acpitz: temp2 /sys/devices/virtual/thermal/thermal_zone0/hwmon0/temp2_input  
coretemp: Package id 0 /sys/devices/platform/coretemp.0/hwmon/hwmon2/temp1_input  
coretemp: Core 0 /sys/devices/platform/coretemp.0/hwmon/hwmon2/temp2_input  
coretemp: Core 1 /sys/devices/platform/coretemp.0/hwmon/hwmon2/temp3_input  
coretemp: Core 2 /sys/devices/platform/coretemp.0/hwmon/hwmon2/temp4_input  
coretemp: Core 3 /sys/devices/platform/coretemp.0/hwmon/hwmon2/temp5_input  
nouveau: temp1 /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0/hwmon/hwmon3/temp1_input
```

## Displays

When attempting to manage displays, whether its the orientation or enabling / disabling, look to the man pages for `xrandr`. See the commands below for some examples.

```
# Output information on displays
```

```
xrandr
```

```
# List the output names for displays
```

```
xrandr --output
```

```
# Move DP-2 to the right of HDMI-1
```

```
xrandr --output DP-2 --right-of HDMI-1
```

```
`
```

```
#### Timezone
```

```
To see date / time, run `date`
```

To adjust local TZ settings, run `tzselect`. Pay attention to the final output of this tool as it will explain how to make your change permanent. For me, I had to add the following to the end of my `~/.profile` :

```
```bash
```

```
TZ='America/New_York'; export TZ
```

## Memory

Some useful commands to find information on memory usage -

```
# Output various memory details
cat /proc/meminfo

# Can be used with grep, awk, etc for more specific output..

# ex) Show MiB of memory available
grep -w MemAvailable: /proc/meminfo | awk '{print $2 / 1024 "MiB"}'
```

## Input Devices

Run the following to get information on input devices attached to the machine -

```
# In the output shown below, my keyboard is AT Translated Set 2 keyboard
xinput list

# Example output:
[ Virtual core pointer                id=2    [master pointer (3)]
 | ↳ Virtual core XTEST pointer        id=4    [slave  pointer (2)]
 | ↳ Elan Touchpad                     id=10   [slave  pointer (2)]
[ Virtual core keyboard              id=3    [master keyboard (2)]
 | ↳ Virtual core XTEST keyboard        id=5    [slave  keyboard (3)]
 | ↳ Power Button                      id=6    [slave  keyboard (3)]
 | ↳ Power Button                      id=7    [slave  keyboard (3)]
 | ↳ Sleep Button                      id=8    [slave  keyboard (3)]
 | ↳ TOSHIBA Web Camera - HD: TOSHIB   id=9    [slave  keyboard (3)]
 | ↳ AT Translated Set 2 keyboard       id=11   [slave  keyboard (3)]

# Test the device..
xinput test "AT Translated Set 2 keyboard"

# Example output:
key release 36
key press  40
dkey release 40
key press  50
key release 50

# The output above shows me pressing / releasing keys in real time.

# Exit with CTRL-C
```

## Power Supplies / AC Adapters

```
# List power supplies, AC adapters -
ls -l /sys/class/power_supply/

# Example output...

lrwxrwxrwx 1 root root 0 Mar 23 23:02 AC ->
```

```
.././devices/LNXSYSTEM:00/LNXSYBUS:00/PNP0A08:00/device:00/PNP0C09:00/ACPI0003:00/power_supply/AC
lrwxrwxrwx 1 root root 0 Mar 23 23:02 BAT0 ->
.././devices/LNXSYSTEM:00/LNXSYBUS:00/PNP0A08:00/device:00/PNP0C09:00/PNP0C0A:00/power_supply/BAT0
# Above, my battery is seen as BAT0, my AC port for charging is AC
```