

Linux Input Method Frameworks and Engines

Manabu Sumioka

2020 年 11 月 15 日

目次

| | | |
|------|---|----|
| 1 | ユーザーと入力メソッド (IM) | 2 |
| 1.1 | 入力には, フレームワークとエンジンの 2 つが関わっている | 4 |
| 1.2 | Input method frameworks, フレームワーク | 4 |
| 1.3 | Input method engines, エンジン | 4 |
| 2 | Debian の input method frameworks と engines 2020/09/15 | 5 |
| 2.1 | IBus (Intelligent Input Bus) | 5 |
| 2.2 | Fcitx (Free Chinese Input Toy for X) | 5 |
| 2.3 | uim (Universal Input Method) | 5 |
| 2.4 | anthy | 6 |
| 2.5 | mozc | 6 |
| 2.6 | kkc | 6 |
| 2.7 | skk | 7 |
| 3 | InputMethod 群 | 7 |
| 3.1 | Input Method Frameworks | 7 |
| 3.2 | Input Method Engines for ibus | 7 |
| 3.3 | Gnome and ibus | 8 |
| 3.4 | SCIM - Smart Common Input Method | 8 |
| 3.5 | インストールした IME を動作させる方法 | 8 |
| 3.6 | Gnome と non-ibus | 8 |
| 3.7 | im-config | 9 |
| 付録 A | ユーザー群 | 9 |
| A.1 | LMDE 4 (debbie) の例 | 9 |
| 付録 B | システム情報 | 10 |
| B.1 | 現在利用中のシステム情報 (System Information) | 10 |
| B.2 | センサー | 10 |

| | | |
|------|---|----|
| B.3 | CPU 情報 | 11 |
| B.4 | ハードウェア情報 | 11 |
| B.5 | inxi | 11 |
| B.6 | IRC | 15 |
| 付録 C | PulseAudio スピーカーとヘッドフォンの切り替え方法 | 15 |
| 付録 D | WiFi などネットワークの不具合を調べるコマンド群 | 16 |
| D.1 | sudo system status networking | 16 |
| 付録 E | Linux mint マウスのかわりにキーボードで操作するやり方 | 17 |
| E.1 | 良く使うキーボードショートカット一覧 | 17 |
| E.2 | ワークスペース作業台 | 18 |
| 付録 F | おまけ | 18 |
| F.1 | 干支 | 18 |

目次

| | | |
|---|--|----|
| 1 | US Keyboard | 3 |
| 2 | Input-method-framework: UIM | 4 |
| 3 | IBus daemon | 5 |
| 4 | uim-skk on Ubuntu 20.04 | 6 |
| 5 | Mozc Settings | 6 |
| 6 | uim-pref-gtk | 7 |
| 7 | SCIM - GUI Setup Utility | 8 |
| 8 | Display Layout | 11 |
| 9 | スピーカーとヘッドフォンの音量 - 内蔵スピーカーを選択している状態 | 16 |

表目次

1 ユーザーと入力メソッド (IM)

Galois 群は、...数学的概念であって、方程式を解くための道具とみなしてはいけない。時代によって考え方が変わって来て昔の考え方にこだわるべきではないということである。要するに数学はきめられた伝統的な段階をふんできっちりやらなければならないものではない。

志村五郎『数学をいかに使うか』

1.0.1 ユーザーとコンピューター入力

コンピューターは、人間が文書をこしらえて、それを自分のために記録したりときには誰かと議論するための資料とするために使われたり、世間に公開するために利用される。文書入力に利用する主体は人間だけど、利用される側のコンピューターも文字入力を受け付けるための影の軍団が動いている。もちろん現代は、文字から音楽、映像のための利用が文字利用よりも多いだろうが文字がなくなることはおそらくない。無くなる時は文明が減びたときだろう。

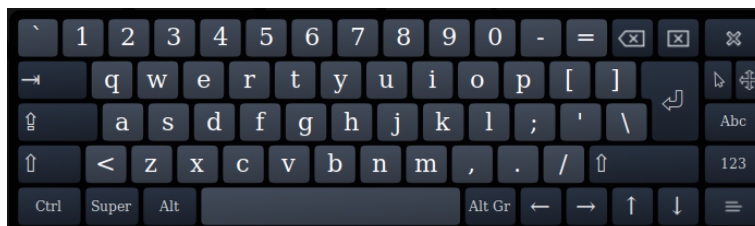


図1 US Keyboard

ところで「コンピューター対人間」のように、この2つは対立したものであるような見方が世間にはあるが、人間も文字入力(あるいは音声入力)を通してコンピューターの内側でプロセスを動かしているのである。そのため、コンピューターの内部に入るため、1ユーザーとしての身分(アカウント)をもらって内部で活動することになる。自分が所有するコンピューター(パソコンやスマホ等)であれば、コンピューター内部に入るための身分、アカウントを自由につくることができる。ただ、自分は王であるとか女王であるとか、あるいは貴族であるとか法にしばられない自由人であるとか好き勝手な身分、アカウントを拵えても、コンピューターの内部に入れば、1ユーザーにすぎない*1。

現実世界

或る個人

生物(約200種, 30兆個の細胞)

コンピューター世界

或るユーザー(複数かもしれない)

有限のプロセス群とファイル群

1ユーザーは、次のような情報をコンピューター内部で持つ。

ユーザー ID $\{x \in \mathbb{Z} \mid 0, 1, 2, 3, \dots, 1000, 1001, \dots\}$

グループ ID $\{x \in \mathbb{Z} \mid 0, 1, 2, 3, \dots, 1000, 1001, \dots\}$

ホームディレクトリー ユーザーにその権限を与えられたディレクトリー

デフォルトシェル OSとのインターフェース

Linuxでのひとつの例をA.1に示した。

ちなみに一ユーザーとしてコンピューター内部での自分の立場は、`id` コマンドで知ることができる。

```
$ id
```

```
uid=1000(manabu) gid=1000(manabu) groups=1000(manabu),4(adm),20(dialout),21(fax),24(cdrom),25(floppy),26
```

*1 ここでは単純化してコンピューターと呼んでいるが、LinuxあるいはFreeBSDのようなオペレーティング・システム(OS)がコンピューター内部を管理支配するユーザー群の一員として登録される。

本人らしきものの後には、adm, dialout, fax, ..., bluetooth, scanner まで幾つもでてくるのは、ながめていると何故なのかなんとなくわかる。当面それでいい。

1.1 入力には、フレームワークとエンジンの2つが関わっている

混同しやすくわかりにくいのは、input method framework (IM フレームワーク) と input method engine (IM エンジン) のサービスと、それがユーザーにどんな役目を果たしているかだ。ただサービスとか役目とか言ってしまうと混乱するばかりだから具体的例をだす。

なんかえらそうなことを書いているが、フレームワークとエンジンの関係分野についてはコードを書いたことはないし、読んだこともない。そのことをお断りしておきます。

1990年代のはじめころに、Xの動くマシンを手元に置き、あとはそれからずっと BSD, Linux ユーザーになった者の見解の一つとってください。

1.2 Input method frameworks, フレームワーク

デーモン (daemon) である*2。ユーザーからの入力を目的のアプリケーションへ渡すサービスを担当するデーモン。IMF デーモンは、IM エンジンとアプリケーションとのあいだを仲介する。入力された文字群を解釈した結果は、デーモンから、アプリケーションによってユーザに表示される。フレームワークによっては、アプリケーションが入力を完全に無視することもある。

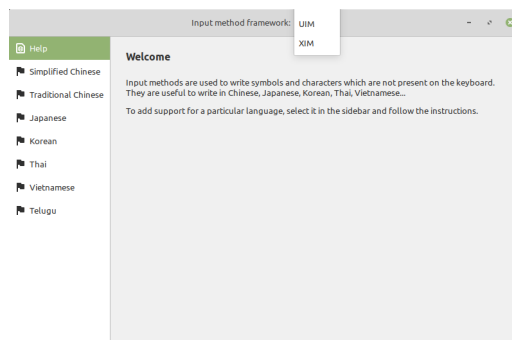


図2 Input-method-framework: UIM

1.3 Input method engines, エンジン

プログラム (program) である。ユーザーが入力した文字群を IM フレームワークを経由して受け取り、解析し、処理した文字列を IM フレームワークに出力し、ユーザーとアプリケーションの反応により完結する。

⇒ ⇒ IM ⇒ IM

*2 そもそも daemon のことを理解していただけるユーザーであってほしいです。そういえば、1990年代のはじめに文学出身の翻訳者と仕事をしていてダエモンっていったい何よ?と聞かれまともに答えてないままでした。デーモン (demon) の古いつづりでもあってギリシア語起源ですが、いまは検索すれば適当な答えがでます。

2 Debian の input method frameworks と engines 2020/09/15

IM/IMF ibus fcitx uim

anthy

kkc ×

mozc

skk

2.1 IBus (Intelligent Input Bus)

Chinese developer, Huang Peng.

私的には、自宅で NetBSD を使っていたころからお世話になっていた。

いまも、NetBSD は、リモートにあるサーバーを利用してます。ドキュメントは、昔からかわらず emacs + skk. このごろ週刊誌でよく終活特集があって老人になってあわてて引っ越ししたり、マンションを買って引っ越すのは止めなさいと警告がある。それに従っているわけではないが、サーバーは中年のころからのもので利用を変更するつもりはまったくありません。もう自前で動かしているわけではありませんから、もしもサービスが停止すれば終わりです。それでいいと思っています。

常用のマシンでは、IBus のデーモンは動かしていませんから図のようになります。



図3 IBus daemon

昔はたいへんお世話になりました。ただし、IBus に関しては、いまこれを書いているローカルの Linux ノートでは使っていません。

2.2 Fcitx (Free Chinese Input Toy for X)

First designed by Yuking.

どうもいまの Linux では、これもよく使われているみたいです。自分も一時期試していたが、いまはやめてしまった。

2.3 uim (Universal Input Method)

Japanese developers.

2020-05-01 最新

すこし不便なことはあるが、ローカルマシンでは uim-skk を使用中。

ただし、ドキュメントは、emacs + ddskk. これは、melpa でアップデートが便利です。

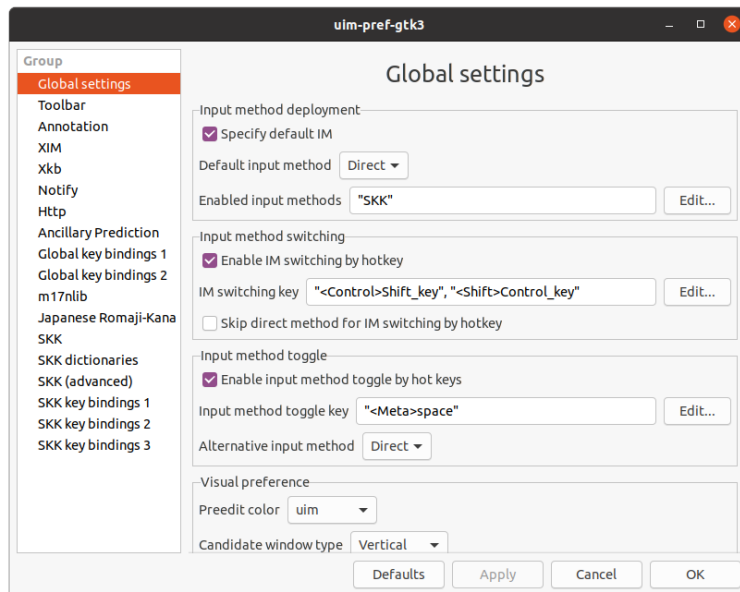


図4 uim-skk on Ubuntu 20.04

2.4 anthy

2015-06-23 から停滞していたが、2017-08-24 から更新が再開。
2019-08-25 テスト中が最新。

2.5 mozc

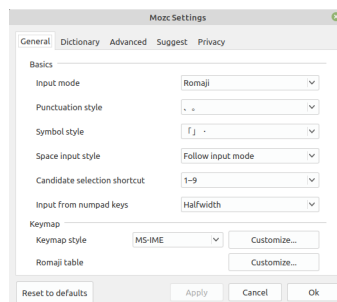


図5 Mozc Settings

2020-07-18 最新

2.6 kkc

2019-11-17 が最新

2.7 skk

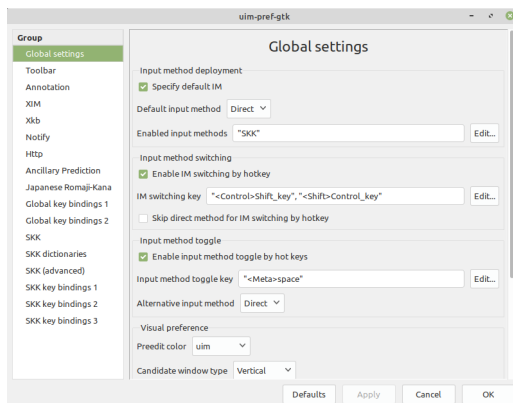


図6 uim-pref-gtk

3 InputMethod 群

[debian wiki - InputMethodBuster](#)を参考にした。

3.1 Input Method Frameworks

現在ポピュラーなのは、4つの input method(IM) フレームワークである。

- ibus Gnome のデフォルト
- uim 日本でポピュラー (emacs とフレンドリー)
- fcitx 中国でポピュラー
- scim KDE*³好み (?)

これらのうちから *im-config* パッケージによって、IM をひとつだけ選択することができる。
Debian のデフォルトデスクトップ Gnome は、特別な理由がないかぎり *ibus* をすすめる。

3.2 Input Method Engines for ibus

- ibus-pinyin Chinese Pinyin input(cn)
- ibus-libpinyin Chinese Pinyin input(cn)
- ibus-zhuyin Chinese zhuyin input(tw)
- ibus-libzhuyin Chinese zhuyin input(tw)
- ibus-cangjie Chinese cangjie input(tw)
- ibun-rime Chinese rime engine with configurable multiple schema(cn,tw)

*³ [Kubuntu](#) は、KDE ソフトウェアをデフォルトにした Ubuntu の派生版

ibus-mozc Japanese multi-phrase conversion
ibus-anthy Japanese multi-phrase conversion
ibus-skk Japanese simple conversion
ibus-kkc Japanese simple multi-phrase conversion
ibus-hangul Korean hangul input
ibus-unikey Vietnamese input
ibus-libthai Thai input

3.3 Gnome and ibus

最新の Gnome(3.30 for Buster) は *ibus* を組み込んである。

3.4 SCIM - Smart Common Input Method

Input Method 設定のためのツールは、アプリ検索で「input method」などとすれば、複数の入力方法アプリが見つかる。

図 7は SCIM の GUI 設定ツールである。

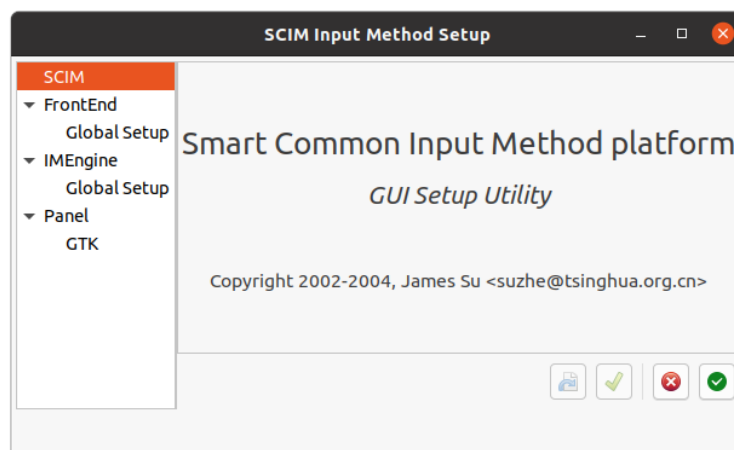


図7 SCIM - GUI Setup Utility

3.5 インストールした IME を動作させる方法

IME をインストールしただけでは動作しない。次の操作が必要。

Settings - Region and Language - Input Sources

3.6 Gnome と non-ibus

個人的には、uim を採用している。

参考リンク [uim in Buster](#)

3.7 im-config

現在の Input Method を調べる.

```
$ im-config
```

たぶんよくわからない.

そこで, `.xinputrc` の中をタイプして見る.

```
$ cat .xinputrc
```

すると, 次のように 3 行のテキストが出力される. 例では, `uim` を使用している.

```
# im-config(8) generated on ...
run_im uim
# im-config signature: a1234 ...
```

このファイルは, `im-config` によって生成されたものということがわかるようになっている.

したがって, これを人間が手で直接書き直すことができない.

`im-config` によって修正するしかない.

```
$ man im-config
```

付録 A ユーザー群

A.1 LMDE 4 (debbie) の例

| USER | User ID | Group ID | HomeDirectory | DefaultShell |
|--------|---------|----------|----------------|-------------------|
| root | 0 | 0 | /root | /bin/bash |
| daemon | 1 | 1 | /usr/sbin | /usr/sbin/nologin |
| bin | 2 | 2 | /bin | /usr/sbin/nologin |
| sys | 3 | 3 | /dev | /usr/sbin/nologin |
| sync | 4 | 65534 | /bin | /bin/sync |
| games | 5 | 60 | /usr/games | /usr/sbin/nologin |
| man | 6 | 12 | /var/cache/man | /usr/sbin/nologin |
| lp | 7 | 7 | /var/spool/lpd | /usr/sbin/nologin |
| mail | 8 | 8 | /var/mail | /usr/sbin/nologin |
| ... | | | | |

付録 B システム情報

Linux Mint の例

B.1 現在利用中のシステム情報 (System Information)

Menu - Administration - System Reports - System Information

クラッシュ、システムトラブルのときの診断に利用するための「Upload」ボタンが右下にある。

<https://termbin.com/...>

にアップロードされる。

B.2 センサー

```
$ sensors
```

```
pch_skylake-virtual-0
```

```
Adapter: Virtual device
```

```
temp1:          +46.0°C
```

```
acpitz-acpi-0
```

```
Adapter: ACPI interface
```

```
temp1:          +25.0°C (crit = +107.0°C)
```

```
coretemp-isa-0000
```

```
Adapter: ISA adapter
```

```
Package id 0:  +38.0°C (high = +100.0°C, crit = +100.0°C)
```

```
Core 0:         +34.0°C (high = +100.0°C, crit = +100.0°C)
```

```
Core 1:         +34.0°C (high = +100.0°C, crit = +100.0°C)
```

```
Core 2:         +30.0°C (high = +100.0°C, crit = +100.0°C)
```

```
Core 3:         +32.0°C (high = +100.0°C, crit = +100.0°C)
```

```
nouveau-pci-0100
```

```
Adapter: PCI adapter
```

```
GPU core:      +0.60 V (min = +0.60 V, max = +1.20 V)
```

```
temp1:         -0.0°C (high = +95.0°C, hyst = +3.0°C)
```

```
(crit = +105.0°C, hyst = +5.0°C)
```

```
(emerg = +135.0°C, hyst = +5.0°C)
```

B.3 CPU 情報

```
$ cat /proc/cpuinfo
```

```
$ lscpu
```

```
$ cpuid
```

B.4 ハードウェア情報

```
$ hardinfo
```



図8 Display Layout

B.5 inxi

Inxi は、たいていほとんどの Linux で使える。BSD でも動作する。
コマンドは、次のようにしてインストールできる。

```
$ sudo apt install inxi
```

inxi は、引数なしで実行すると、CPU、Kernel、Memory、Storage などの情報を表示し、最後に inxi のバージョンを示す。

```
$ inxi
```

```
CPU: Quad Core Intel Core i5-6300HQ (-MCP-) speed/min/max: 900/800/3200 MHz
Kernel: 4.19.0-10-amd64 x86_64 Up: 9h 42m Mem: 3845.4/32020.0 MiB (12.0%)
Storage: 961.33 GiB (6.3% used) Procs: 256 Shell: bash 5.0.3 inxi: 3.0.32
\end{listlisting}
```

次に recommends オプションをつけて実行してみる。

```
\begin{quote}
  \$ inxi --recommends
\end{quote}
```

```
\begin{lstlisting}
inxi will now begin checking for the programs it needs to operate.
```

```
Check inxi --help or the man page (man inxi) to see what options are
available.
```

Test: core tools:

```
Perl version: 5.028001
Current shell: bash 5.0.3
Default shell: bash
sh links to: /bin/dash
```

Test: required system directories:

```
/proc: ..... Present
/sys: ..... Present
```

All required system directories are present

Test: recommended system programs:

Note: IPMI sensors are generally only found on servers. To access that data, you only need one of the ipmi items.

```

blockdev: --admin -p/-P (filesystem blocksize)..... Present
dig: -i wlan IP ..... Present
dmidecode: -M if no sys machine data; -m..... Present
fdisk: -D partition scheme (fallback)..... Present
file: -o unmounted file system (if no lsblk)..... Present
hddtemp: -Dx show hdd temp..... Present
ifconfig: -i ip LAN (deprecated)..... Present
ip: -i ip LAN..... Present
ipmitool: -s IPMI sensors (servers)..... Present
ipmi-sensors: -s IPMI sensors (servers)..... Present
lsblk: -o unmounted file system (best option)..... Present
lsusb: -A usb audio; -N usb networking; --usb (optional)..... Present
modinfo: Ax; -Nx module version ..... Present
runlevel: -I fallback to Perl..... Present
sensors: -s sensors output ..... Present
strings: -I sysvinit version ..... Present
sudo: -Dx hddtemp-user; -o file-user ..... Present
tree: --debugger 20,21 /sys tree ..... Present
upower: -sx attached device battery info ..... Present
uptime: -I uptime ..... Present

```

All recommended system programs are present

Test: recommended display information programs:

```

glxinfo: -G glx info ..... Present
wmctrl: -S active window manager (fallback)..... Present
xdpyinfo: -G multi screen resolution ..... Present
xprop: -S desktop data ..... Present
xrandr: -G single screen resolution ..... Present

```

All recommended display information programs are present

Test: recommended downloader programs (You only need one of these):

Perl HTTP::Tiny is the default downloader tool if IO::Socket::SSL is present.
See --help --alt 40-44 options for how to override default downloader(s) in

case of issues.

If dig is installed, it is the default for WAN IP data. Strongly recommended.
Dig is fast and accurate.

```
curl: -i (if no dig); -w,-W; -U..... Present
dig: -i wlan IP ..... Present
wget: -i (if no dig); -w,-W; -U..... Present
```

All recommended downloader programs are present

Test: recommended Perl modules (Optional):

None of these are strictly required, but if you have them all, you can eliminate some recommended non Perl programs from the install.

HTTP::Tiny and IO::Socket::SSL must both be present to use as a downloader option. For json export Cpanel::JSON::XS is preferred over JSON::XS.

```
HTTP::Tiny: -U; -w,-W; -i (if dig not installed)..... Present
IO::Socket::SSL: -U; -w,-W; -i (if dig not installed)..... Present
Time::HiRes: -C cpu sleep (not required); --debug timers..... Present
Cpanel::JSON::XS: --output json - required for export..... Present
JSON::XS: --output json - required for export (legacy)..... Present
XML::Dumper: --output xml - Crude and raw..... Present
Net::FTP: --debug 21,22..... Present
```

All recommended Perl modules are present

Test: recommended directories:

```
/dev: -l,-u,-o,-p,-P,-D disk partition data..... Present
/dev/disk/by-id: -D serial numbers..... Present
/dev/disk/by-label: -l,-o,-p,-P partition labels..... Missing
/dev/disk/by-path: -D extra data..... Present
/dev/disk/by-uuid: -u,-o,-p,-P partition uuid..... Present
/sys/class/dmi/id: -M system, motherboard, bios..... Present
```

The following recommended directories are missing:

Directory: /dev/disk/by-label

Test: recommended files:

Note that not all of these are used by every system, so if one is missing it's usually not a big deal.

```
/etc/lsb-release: -S distro version data (older version)..... Present
/etc/os-release: -S distro version data (newer version)..... Present
/proc/asound/cards: -A sound card data..... Present
/proc/asound/version: -A ALSA data..... Present
/proc/cpuinfo: -C cpu data..... Present
/proc/mdstat: -R mdraid data (if you use dm-raid)..... Missing
/proc/meminfo: -I,-tm, -m memory data..... Present
/proc/modules: -G module data (sometimes)..... Present
/proc/mounts: -P,-p partition advanced data..... Present
/proc/scsi/scsi: -D Advanced hard disk data (used rarely)..... Missing
/var/log/Xorg.0.log: -G graphics driver load status..... Present
```

The following recommended files are missing:

File: /proc/mdstat

File: /proc/scsi/scsi

Ok, all done with the checks. Have a nice day.

B.6 IRC

Menu - Internet - HexChat

LinuxMint/#linuxmint-help につながる.

ここで相談のとき, 上のシステム情報へのリンクを使う.

付録 C PulseAudio スピーカーとヘッドフォンの切り替え方法

メニュー *Win* キーからサウンドの中のパルスオーディオ機能を選択することによって, サウンド出力先を切り替えることができる.

図9は, 内蔵スピーカーとヘッドフォンの例.

1. Win
2. Sound and Video
3. Pulse Audio Volume Control

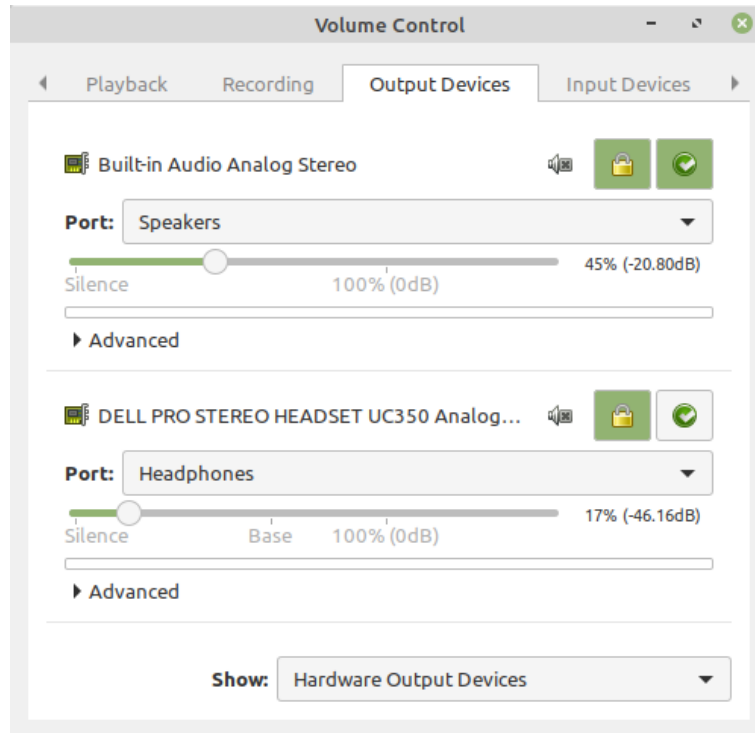


図9 スピーカーとヘッドフォンの音量 - 内蔵スピーカーを選択している状態

付録 D WiFi などネットワークの不具合を調べるコマンド群

D.1 sudo system status networking

Linux ノート PC など、しばらく放置してスリープ状態になったあと、再ログインしたあと WiFi 接続が切れていることに気づくことがある。たいていは再起動することになるのだが、ネットワークリスタートで接続できることもある。これは、ネットワーク関連のハードウェアと Linux カーネルの間で起きた問題であまり簡単なことではなく調べるとむずかしいことが多い。そのため、たいていはリブートしてしまう。

その前にできることと言えば、気休めかもしれないが、`systemctl` コマンドで状況を確認することだ。

```
$ sudo systemctl status networking
```

ドキュメントとして `man interfaces(5)` に詳細が載っているが、専門知識が必要。

付録 E Linux mint マウスのかわりにキーボードで操作するやり方

Linux mint 上でアプリケーションを動かすと窓が開く。複数のアプリケーションを動かせば複数の窓が開く。その窓の位置などは窓を管理するシステムによって制御されている。

さて、窓から窓へと移るには、*Alt + Tab*をつかう。これは、マイクロソフトの Windows でも同じ。*Alt + F4*で窓を閉じることができるのも、Windows と linux mint でかわりない。

ただ、アプリ窓がせまくてもっと広げたいときの操作、*Alt + F10*は Linux mint で動くが、Windows ではだめようだ。このように少しずつウィンドウ管理システムによって指の動かし方が違うため、ひとつのシステムに慣れると、それに取り込まれてしまいやすい傾向がある。

車の運転もそうだ。右ハンドルと左ハンドルは実際に乗ってみればわかるが同じように運転できる。右手でのミッション操作も左手での操作もどこかで脳内変換があるのかすぐ慣れる。しかし、ワイパーと方向指示器が右手、左手で逆になってるのは、ワイパーと方向指示器のような別アプリが逆手になっているため、一度自分の脳内で意識しないと間違えることもあったりする。

Linux を使う人は Microsoft Windows も使っている人がほとんどだと思う。

キーボードショートカットは、このワイパーと方向指示器のようなもので指の動かしかなどで覚えてしまわないと、なかなか使いこなせないものだけど、慣れると大変に便利である。しかし、Windows と Linux の違いはどうも私的には慣れないため片方は、マウス中心になってしまう。

筆者は、Linux は、キーボード派である。理由は、もともとコンピューターに慣れたのがまだ米国製のメインフレームそれから DEC のミニコン用 OS それから UNIX の順にだったせいだ。

ワークスペース、アプリ、窓の操作については、次の一覧にまとめた。

E.1 良く使うキーボードショートカット一覧

- *Win* アプリケーションを起動するためのメニュー
- *Win + D* デスクトップ
- *Win + E* ファイルマネージャー
- *Ctrl + Alt + T* 新しい Terminal を開く
- *Alt + F2* コマンド実行の指示
- *Ctrl + Alt + ← →* ワークスペースを移動する、手・指の大きさにもよるが、右手の人差し指と薬指が ← と →。左手小指が *Ctrl* 人差し指が *Alt* キーなど。左手薬指は、左シフトのためにおくなど。自分の好みで手指を配置。
- *Ctrl + Alt + Shift + ← →* アプリ窓を右/左ワークスペースに移す。左手は、指 3 本を同時につかうことになる。
- *Alt + Tab* 順に開いたアプリ窓をまわる。最後のアプリまで行くと最初にもどる。逆にまわるには *Shift* をくわえる。
- *Alt + F4* アプリ窓を閉じる
- *Alt + F10* アプリを全画面と非全画面でトグルする
- *Alt + F7* 窓サイズをかえる
- *Alt + SPACE* 窓メニュー表示

- *Alt + PrtScr* 窓のスクリーンショット

この他にもいろいろあり検索すれば、操作はでてる。

筆者がよく利用するのは、4つあるデスクトップを切り替えながら使うやり方だ。

E.2 ワークスペース作業台

Linux mint では、4つのワークスペースを作業に使うことができる。

Ctrl+Alt+↑ でスクリーンに4つのワークスペース一覧を表示する。

E.2.1 ワークスペース配列

Workspace1 Workspace2

Workspace3 Workspace4

4つの作業台(ワークスペース)を切り替えるには、*Ctrl + Alt* を押しながら、左右・上下の矢印キーを押す。

E.2.2 矢印キーでワークスペース間を移動

Workspace1 Workspace2 Workspace3 Workspace4

→ ←→ ←→ ←

付録 F おまけ

F.1 干支

干支(えと)は、中国由来だがそのことを知らない人々が日本ではいる。

とくに干支の「干」は、古代中国の順位を示す。

甲乙丙丁戊己庚辛壬癸(こうおつへいていぼきこうしんじんき)

十二支は、天を十二の動物にあてた中国の暦が由来になっている。

十二の動物は日本でもよく知られているが、中国人が解説した Chinese Zodiac も参考にして、ちょっと違いがあるかどうか見てみよう。

Chinese Zodiac