

# PERSONAL UBUNTU SET-UP

Frédéric GALLIANO

Université Paris-Saclay, Université Paris Cité, CEA, CNRS, AIM, 91191, Gif-sur-Yvette, France

February 23, 2024

## Contents

<b>1</b>	<b>SYSTEM SET-UP</b>	<b>2</b>
1.1	Accented Characters . . . . .	2
1.2	Customizing the Touch Pad . . . . .	2
1.3	Terminal . . . . .	2
1.4	Windows . . . . .	3
1.5	Drivers Nvidia . . . . .	3
1.6	Background Image . . . . .	3
1.7	GPG Keys . . . . .	3
1.8	OwnCloud . . . . .	3
1.9	Eduroam . . . . .	3
1.10	Forticlient . . . . .	4
1.11	Disks . . . . .	4
1.11.1	Partition of the primary disk . . . . .	4
1.11.2	Formatting and mounting the secondary disk . . . . .	4
1.11.3	Crypting partitions . . . . .	4
1.11.4	Putting everything together . . . . .	4
1.11.5	Automatically mount a crypted partition . . . . .	4
1.12	Swap . . . . .	5
1.13	Sleep issue on CEA laptops . . . . .	5
1.14	Printing with CUPS . . . . .	5
1.15	Sync Google calendars . . . . .	5
1.16	Date format . . . . .	5
1.17	BIOS Update & Firmware . . . . .	5
<b>2</b>	<b>SOFTWARE</b>	<b>5</b>
2.1	Web Browser . . . . .	5
2.1.1	Firefox . . . . .	5
2.1.2	Brave . . . . .	5
2.1.3	Chrome . . . . .	6
2.2	Emacs . . . . .	6
2.2.1	Packages installed with MELPA (in <code>~/.emacs.d/elpa/</code> ) . . . . .	6
2.2.2	Packages installed by hand (in <code>~/.emacs.d/lisp/</code> ) . . . . .	7
2.2.3	Dictionnaires for spell-checking . . . . .	7
2.2.4	For emails . . . . .	7
2.3	L <sup>A</sup> T <sub>E</sub> X et MusiX <sub>T</sub> E <sub>X</sub> . . . . .	8
2.3.1	Extra fonts . . . . .	8
2.4	Git . . . . .	8
2.5	Fortran . . . . .	8
2.5.1	HDF5 . . . . .	8
2.6	Python . . . . .	9
2.6.1	Standard . . . . .	9
2.6.2	Anaconda . . . . .	9

2.6.3	Nerd-dictation	9
2.6.4	Extra fonts	9
2.7	IDL	10
2.8	PDF	10
2.8.1	Readers	10
2.8.2	Have hyperlinks open with Brave	10
2.8.3	PDF Presenter Console	10
2.8.4	Converting HTML to PDF	10
2.8.5	Universal converter	10
2.8.6	PDFArranger	11
2.9	Zoom	11
2.10	Dropbox	11
2.11	Images	11
2.11.1	ImageMagick	11
2.11.2	PNG compression	11
2.11.3	Webp conversion	11
2.11.4	FITS files	11
2.11.5	Diaporama	11
2.11.6	Drawing	11
2.11.7	Webcam	11
2.12	Music	11
2.13	Miscellaneous	12
<b>3</b>	<b>SETTING FILES TO IMPORT</b>	<b>12</b>
3.1	Bashrc	12
3.2	Emacs	12
3.3	Rest	12
3.4	Release Upgrade at CEA	12
<b>4</b>	<b>TROUBLE SHOOTING</b>	<b>12</b>
4.1	Snap	12

## 1 SYSTEM SET-UP

- `sudo apt update`
- `sudo apt install xfce4`
- `sudo apt install lightdm`

### 1.1 Accented Characters

1. `sudo apt install gnome-tweaks`
2. Applications → Paramètres → Gestionnaire de paramètres → Clavier → Layout → compose key → Right Ctrl
3. Reboot

### 1.2 Customizing the Touch Pad

To avoid clicking on everything each time one uses the touch pad:

1. Settings manager → mouse and touchpad → touchpad
2. Unselect "Tap touchpad to click"
3. Unselect "Reverse scroll direction"

### 1.3 Terminal

I make Tilix my default terminal:

1. `sudo apt install tilix`
2. `exo-preferred-applications` → utilities → choose tilix
3. C-M-t → open tilix
4. Click on the preference icon:

- a) Global → Automatically copy text to clipboard when selecting
- b) Appearance → window style → disable CSD / hide toolbar
- c) Appearance → Theme variant → Dark
- d) Appearance → Default session name → Terminal
- e) Appearance → Terminal title style → None
- f) Appearance → Application title → *hostname* :{directory} [columnsx{rows}]
- g) Default → General → Terminal size → 100 60
- h) Default → General → Custom font : Monospace 10
- i) Default → Command → Run command as a login shell

## 1.4 Windows

Applications → Window manager

1. Style → Greybird-dark-accessibility
2. Keyboard → maximize window Alt-Fn-F10
3. Keyboard → move window Alt-Fn-F7
4. Keyboard → resize window Alt-Fn-F8
5. Keyboard → show desktop → Alt-F9
6. Keyboard → Move window to left workspace → Shift-Ctrl-Alt-Left
7. Keyboard → Move window to right workspace → Shift-Ctrl-Alt-Right

## 1.5 Drivers Nvidia

1. `sudo apt install nvidia-driver-455`
2. Reboot

## 1.6 Background Image

Applications → Desktop → select an image

## 1.7 GPG Keys

`sudo apt install gnupg2`

## 1.8 OwnCloud

1. `sudo apt install gnome-keyring`
2. `sudo apt install python3-keyring`
3. Follow the instructions on: <https://download.owncloud.com/desktop/ownCloud/stable/latest/linux/download/> and make sure that the owncloud files in `/etc/apt/sources.list.d/` and in `/etc/apt/trusted.gpg.d/` have `-r--r--r--` permissions:
 

```
sudo chmod go+r /etc/apt/sources.list.d/owncloud.list
sudo chmod go+r /etc/apt/trusted.list.d/owncloud.gpg
```
4. Launch `/opt/ownCloud/ownCloud/bin/owncloud`
5. Connection server: `https://mycore.core-cloud.net`
6. User: `xxxxxxxxxxxxxxxx@cea.fr`
7. Password: specific.

## 1.9 Eduroam

1. Launch `~/ownCloud/Ordinateurs/Eduroam/eduroam-linux-eduroam.py`
2. Configure the WIFI:
  - Sécurité:** WPA et WPA2 d'entreprise
  - Authentification:** TLS avec tunnel (TTLS)
  - Identité masquée:** `anonymous@eduroam.cnrs.fr`
  - Certificat CA:** `~/cat_installer/ca.pem`
  - Authetification interne:** PAP
  - Nom d'utilisateur:** `ae78hm@eduroam.cnrs.fr`
  - Mot de passe:** secret sentence (see in `~/ownCloud/Ordinateurs/Eduroam/eduroam.org`)

## 1.10 Forticlient

1. `sudo apt remove forticlientsslvpn-cea`
2. `sudo dpkg -i ~/ownCloud/AIM_Dap/forticlientsslvpn-cea_4.4.2342_amd64.deb`
3. Accept the license
4. Close the graphical interface without filling the form
5. Launch the VPN with the command `fortissslvpn_cli --server mobietendu-ea.cea.fr:443 --vpnuser xxxxxxxx@extra.cea.fr` (alias `vpncea`)
6. Password is the extra password
7. C-c to quit

## 1.11 Disks

### 1.11.1 Partition of the primary disk

1. `sudo fdisk -l`
2. `sudo lvresize -r -L 80G /dev/mapper/crypt-root`
3. `sudo lvresize -r -L 10G /dev/mapper/crypt-var`
4. `sudo lvresize -r -L 800G /dev/mapper/crypt-home`

### 1.11.2 Formatting and mounting the secondary disk

1. `sudo fdisk /dev/nvme1n1`
  - g
  - w
2. `sudo fdisk /dev/nvme1n1`
  - n
  - default sectors
  - w

### 1.11.3 Crypting partitions

1. Create the crypted partition: `sudo cryptsetup --cipher aes-xts-plain --key-size 512 --hash sha512 -v luksFormat /dev/nvme1n1`
2. Type a password sentence

### 1.11.4 Putting everything together

**Open the crypted partition:** `sudo cryptsetup -v luksOpen /dev/nvme1n1 Disk2.`

**Type:** password sentence

**Create a filesystem:** `sudo mkfs -t ext4 -L LuksPartition /dev/mapper/Disk2.`

**Mount the partition** `sudo mount /dev/mapper/Disk2 /mnt/data`

**Create a backup** `sudo cryptsetup -v luksHeaderBackup /dev/nvme1n1 --header-backup-file LuksHead`

### 1.11.5 Automatically mount a crypted partition

1. Create a random key
  - a) Create the directory `/etc/luks-keys` if it does not exist
  - b) `sudo chmod go-rwx /etc/luks-keys`
  - c) `sudo dd if=/dev/urandom of=/etc/luks-keys/Disk2_key bs=512 count=8`
2. Add the random key
  - a) `sudo cryptsetup -v luksAddKey /dev/nvme1n1 /etc/luks-keys/Disk2_key`
  - b) Change the rights of `Disk2_key` to have `-r-----`
  - c) Test the key: `sudo cryptsetup -v luksOpen /dev/nvme1n1 Disk2 --key-file=/etc/luks-keys/Disk2_key`
  - d) Close the partition: `sudo cryptsetup -v luksClose Disk2`
3. Automatically open the volume
 

**Print the UUID volume:** `sudo cryptsetup luksDump /dev/nvme1n1 | grep "UUID"`

**Edit `/etc/crypttab`:** add the line `Disk2 UUID=d98a244a-1aac-4ab7-965d-936aa0c61eeb /etc/luks-keys/Disk2_key luks`

**Check everything is OK:** `sudo cryptdisks_start Disk2.`
4. Automatically mount the disk Add in `/etc/fstab`: `/dev/mapper/Disk2 /mnt/data ext4 defaults 0 2`

## 1.12 Swap

- `sudo swapoff -a`
- `sudo dd if=/dev/zero of=/swapfile bs=1G count=10`
- `sudo chmod 0600 /swapfile`
- `sudo mkswap /swapfile`
- `sudo swapon /swapfile`
- `grep Swap /proc/meminfo`
- Add `/swapfile swap swap sw 00` at the end of `/etc/fstab`

## 1.13 Sleep issue on CEA laptops

1. `sudo apt install pm-utils`
2. Use `pm-suspend` in place of `systemctl suspend`

## 1.14 Printing with CUPS

1. `sudo vi ~/.cups/client.conf` (*a priori*, the file does not exist)
2. Write in the file: `servername saprint.extra.cea.fr`

## 1.15 Sync Google calendars

1. `crontab -e` (without `sudo`)
2. Add the line `0 0,12 * * * ~/ownCloud/Organisation/sync_gcal.sh`

## 1.16 Date format

1. `sudo vi /etc/default/locale`
2. Edit:

```
LANG="=fr_FR.UTF-8="
LANGUAGE="=fr_FR:en_US="
LC_ALL="=fr_FR.UTF-8="
```

1. Reboot
2. When restarting, the system asks if we want to keep the directory names (Desktop/, Pictures/, etc.) in English ⇒ answer yes

## 1.17 BIOS Update & Firmware

1. `sudo service fwupd start`
2. `sudo fwupdmgr refresh`
3. `sudo fwupdmgr update`

# 2 SOFTWARE

## 2.1 Web Browser

### 2.1.1 Firefox

- Installed by default
- Sync the keywords and bookmarks

### 2.1.2 Brave

- `sudo apt install apt-transport-https curl`
- `sudo curl -sS https://brave-browser-apt-release.s3.brave.com/brave-core.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/brave-browser-release.gpg`
- `echo "deb [signed-by=/etc/apt/trusted.gpg.d/brave-browser-release.gpg arch=amd64] https://brave-browser-apt-release.s3.brave.com/ stable main"|sudo tee /etc/apt/sources.`
- Make sure that the brave files in `/etc/apt/sources.list.d/` and in `/etc/apt/trusted.gpg.d/` have `-r--r--r--` permissions:

```
sudo chmod go+r /etc/apt/sources.list.d/brave-browser-release.list
sudo chmod go+r /etc/apt/trusted.list.d/brave-browser-release.gpg
```

- `sudo apt update`
- `sudo apt install brave-browser`
- **Sync the keywords and bookmarks**
- **Make Brave the default browser:** `xdg-settings set default-web-browser brave-browser.desktop`

### 2.1.3 Chrome

- `sudo sh -c 'echo "deb [arch=amd64] https://dl.google.com/linux/chrome/deb/ stable main" > /etc/apt/sources.list.d/google-chrome.list'`
- `wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -`
- `sudo apt update`
- `sudo apt install google-chrome-stable`

## 2.2 Emacs

- `sudo apt install emacs` works well, but some packages downloaded with `melpa` might be more recent and cause bugs. For the February 2024 installation I made, `apt` was giving me `emacs 27.1`, while I needed version 28 to make `auctex` work. I thus installed `emacs` from the repo:
- `sudo apt remove --autoremove emacs emacs-common`
- `sudo add-apt-repository ppa:kellek/emacs`
- `sudo apt update`
- `sudo apt emacs28`
- **Remove everything within `~/.emacs.d/`, except `lisp` and `init.el`.**

### 2.2.1 Packages installed with MELPA (in `~/.emacs.d/elpa/`)

Install with `M-x list-packages`.

- `auctex`
- `auctex-latexmk`
- `auctex-lua`
- `calfw`
- `calfw-ical`
- `calfw-org`
- `elpy`
- `fortpy`
- `guide-key`
- `helm-bibtex`
- `htmlize`
- `hydra`
- `neotree`
- `org`
- `org-bullets`
- `org-super-agenda`
- `pdf-tools`
- `rainbow-mode`
- `symon`
- `use-package`
- `guess-language`
- `gnus` (built-in)
- `recentf` (built-in)
- `saveplace` (not found anymore)
- `paren` (not found anymore)
- `printing` (built-in)
- `org-msg`
- `smtplib` (not found anymore)
- `dired-subtree`
- `dired-rainbow`
- `rainbow-delimiters`
- `php-mode`

- `tablist` (dependency)
- `let-alist` (built-in)
- `gnu-elpa-keyring-update`
- `highlight-indent-guides`

### 2.2.2 Packages installed by hand (in `~/ .emacs.d/lisp/`)

- `template`
- `org-msg`
- `org-faces`
- `grammlecte-cache`

For each `.el` in the directory, do `M-x byte-compile <file>`. If the file `.el` is not available, download it from GitHub.

### 2.2.3 Dictionaries for spell-checking

1. `sudo apt install ispell`.
2. `hunspell -D` → lists the dictionaries already installed and their location (`/usr/share/hunspell/`).
3. Download dictionaries from the [LibreOffice extension repository](#).
4. Extract the `.aff` and `.dic` files from the `.oxt` archives:

```
unzip lo-oo-ressources-linguistiques-fr-v5-7.oxt -d dict-fr
sudo cp dict-fr/*/*.aff dict-fr/*/*.dic /usr/share/hunspell/
rm -r dict-fr
```

5. Since we use the default local variables `fr_FR`, our main French dictionary should be renamed:

```
sudo cp /usr/share/hunspell/fr-classique.aff /usr/share/hunspell/fr_FR.aff
sudo cp /usr/share/hunspell/fr-classique.dic /usr/share/hunspell/fr_FR.dic
```

### 2.2.4 For emails

- `sudo apt instal isync` (for `mbsync`, the software retrieving the mails from the server)
- `sudo apt install altermime` (with default settings in the dialog)
- `sudo apt install html2text`

#### 1. Mu (powerful email search tool)

##### a) New mu install (works in 22.04)

- `sudo apt install xapian-tool`
- `sudo apt install maildir-tools`
- `sudo apt install mu4e`

##### b) Old mu install

- Prior to that, you need to install `meson` and other libraries (look at the Github page). However, do not do that using the `--user` option in `pip3`. Do it as `sudo`, otherwise it won't work when you will do `sudo make install`.
- `sudo apt install libssl-dev`
- Install `cmake` from the repo (not using `apt`; the version is not high enough)
- `sudo apt install guile-3.0`
- `sudo apt install graphviz`
- `sudo apt install doxygen`
- `sudo apt install help2man`
- `sudo apt install python3-docutils`
- `sudo apt install libmagic-dev`
- `sudo apt install bison`
- `sudo apt install libcld2`
- Download `xapian-core` from Github and install it.
- Download tarball from <https://github.com/djcb/mu>
- Follow the instruction to manually install `mu`

##### c) Setting-up mu

- `mkdir ~/mail`
- `mkdir ~/mail/CNRS`
- `mu init --maildir=~/mail --my-address=xxxxxxxxx.xxxxxxxxxx@cea.fr --my-address=xxxxxxxxx`
- `mu index`

It can be used on the command line to search emails, e.g. `mu find coucou`. To list the contacts: `mu cfind`.

#### 2. Create a certificate

- a) `mkdir -p ~/.cert`
- b) `openssl s_client -connect imap.cnrs.fr:993 -showcerts 2>&1 < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' | sed -ne '1,/-END CERTIFICATE-/p' > ~/.cert/imap.cnrs.fr.pem`

### 3. Configuration of mbsync

- a) Create the `~/.mbsyncrc` file (always the same)
- b) Create a crypted password file with `gpg2`:
  - i. Create `~/tmp` file with:
 

```
machine imap.cnrs.fr login xxxxxxxx.xxxxxxxx@ods.services password XXX
machine smtp.cnrs.fr login xxxxxxxx.xxxxxxxx@ods.services password XXX
machine imap.extra.cea.fr login xxxxxxxx password XXX
machine mx.extra.cea.fr login xxxxxxxx password XXX
machine imap.sfr.fr login xxxxxxxx.xxxxxxxx@neuf.fr password XXX
machine smtp.sfr.fr login xxxxxxxx.xxxxxxxx@neuf.fr password XXX
```
  - ii. `gpg2 --output .authinfo.gpg --symmetric tmp`
  - iii. `rm ~/tmp`
- c) It can be used on the command line, without going through emacs: `mbsync -aV` ⇒ sync all mailboxes. This is what emacs spawns.

### 4. Trouble shooting If the certificate changes, run:

```
openssl s_client -connect imap.cnrs.fr:993 -showcerts 2>&1 < /dev/null | sed -ne '/-BEG
```

## 2.3 L<sup>A</sup>T<sub>E</sub>X et MusiX<sub>T</sub>E<sub>X</sub>

```
sudo apt install texlive-full
```

### 2.3.1 Extra fonts

- `sudo apt install texlive-latex-extra fonts-arabeyes fonts-kacst`
- `sudo apt install fontforge`
- `sudo apt install cabextract`
- `wget https://gist.github.com/maxwelleite/10774746/raw/ttf-vista-fonts-installer.sh -q -O - | sudo bash`
- `fc-list | grep Calibri`

### 2.4 Git

```
sudo apt install git-all
```

### 2.5 Fortran

```
sudo apt install gfortran
```

#### 2.5.1 HDF5

##### 1. Required libraries

- a) `sudo apt-get install debhelper`
- b) `sudo apt-get install sharutils`
- c) `sudo apt-get install chrpath`
- d) `sudo apt-get install php-cli`
- e) `sudo apt-get install libaec-dev`

##### 2. Install from source code

- a) Download the archive from: <https://www.hdfgroup.org/downloads/hdf5/source-code/>
- b) `cd ~/Downloads`
- c) `gunzip < hdf5-X.Y.Z.tar.gz | tar xf -`
- d) `cd hdf5-X.Y.Z`
- e) `sudo ./configure --prefix=/usr/local/hdf5 --enable-fortran`
- f) `sudo make`
- g) `sudo make check`
- h) `sudo make install`
- i) `sudo make check-install`
- j) `rm -r hdf5-XXX*`



## 2.6 Python

### 2.6.1 Standard

1. `sudo apt install ipython3`
2. `sudo apt install python3-pip`
3. Modules (without sudo):
  - a) `pip install numpy`
  - b) `pip install scipy`
  - c) `apt install tk-dev (for matplotlib)`
  - d) `pip install matplotlib`
  - e) `pip install astropy`
  - f) `pip install h5py`
  - g) `pip install drawSvg`
  - h) `pip install csv`
  - i) `pip install PyPDF2`
  - j) `pip install datetime`
  - k) `pip install pandas`
  - l) `pip install reproject`
  - m) `pip install photutils`
  - n) `pip install scikit-learn`

### 2.6.2 Anaconda

1. Download anaconda from <https://www.anaconda.com/products/individual>
2. `sha256 Anaconda-2021.11-Linux-x86_64.sh`
3. `bash Anaconda-2021.11-Linux-x86_64.sh`
4. Don't care about the PYTHONPATH warning
5. Once the installation is finished, a paragraph has been added in `~/.bashrc`:
  - a) Add `conda config --set auto_activate_base false` after it
  - b) If it is not there, add the call to `bashrc.sh (aliases)` after that.
6. To install a package: `conda install matplotlib`
7. To turn off the annoying prompt: `conda deactivate`

### 2.6.3 Nerd-dictation

- `pip3 install vosk`
- `sudo apt install xdotool`
- `git clone https://github.com/ideasman42/nerd-dictation.git`
- `cd nerd-dictation`
- `wget https://alphacephei.com/kaldi/models/vosk-model-small-en-us-0.15.zip`
- `unzip vosk-model-small-en-us-0.15.zip`
- `mv vosk-model-small-en-us-0.15 model`
- `mv ~/nerd-dictation ~/bin/`
- `mkdir -p ~/.config/nerd-dictation`
- `mv ./model ~/.config/nerd-dictation`
- Then also download larger models from <https://alphacephei.com/vosk/models>

### 2.6.4 Extra fonts

- `sudo apt install font-manager`
- `sudo apt install fonts-comic-neue`
- `sudo apt install fonts-humor-sans`
- `sudo apt install ttf-mscorefonts-installer`
- Download the xkcd fonts (<https://github.com/ipython/xkcd-font/tree/master>)
- Install them:
 

```
sudo mkdir /usr/share/fonts/truetype/xkcd
sudo cp ~/Downloads/xkcd.otf /usr/share/fonts/truetype/xkcd/
sudo cp ~/Downloads/xkcd-script.ttf /usr/share/fonts/truetype/xkcd/
sudo cp ~/Downloads/xkcd-Regular.otf /usr/share/fonts/truetype/xkcd/
sudo fc-cache -f -v
```
- Eventually clean the matplotlib cache: `rm -r ~/.cache/matplotlib`

## 2.7 IDL

1. `cd $idlib;`
2. `tar -xzvf idl70pluslicense.tar.gz`
3. `sudo mkdir /usr/local/itt`
4. `sudo chmod a+rx /usr/local/itt`
5. `cd /usr/local/itt`
6. `sudo tar -xzvf $idlib/idl70linux.x86.tar.gz`
7. `sudo ./install (no DICOM)`
8. Copy the directory license in `/usr/local/itt/license`: `sudo cp -r $idlib/license/ /usr/local/itt/`
9. Install the following 32 bit librarires:
  - `sudo apt install libxmu6:i386`
  - `sudo apt install libxpm4:i386`
  - `sudo apt install libgtk2.0-0:i386`
  - `wget http://archive.ubuntu.com/ubuntu/pool/main/g/glibc/multiarch-support_2.27-3ubuntu1.2_amd64.deb`  
(not available anymore, instead `sudo apt install binutils-multiarch`)
  - `sudo dpkg -i multiarch-support_2.27-3ubuntu1.2_amd64.deb`
  - `sudo dpkg -i $idlib/libxp6_1.0.2-1ubuntu1_amd64.deb`
  - `sudo apt install libxft2:i386`
  - `sudo apt install libstdc++5:i386`
  - `sudo apt install libc6-i386`
  - `sudo apt install libxtst6:i386`
  - `sudo apt install lib32z1`
10. To install libxp6, you need to install multiarch-support:
  - `wget http://archive.ubuntu.com/ubuntu/pool/main/g/glibc/multiarch-support_2.27-3ubuntu1.2_amd64.deb`
  - `sudo apt install ./multiarch-support_2.27-3ubuntu1_amd64.deb`
11. To use the 64 bit version, install libxp6 (Debian package included in the directory \$idlib): `sudo dpkg -i libxp6_1.0.2-1ubuntu1_amd64.deb`
12. Add in the `.bashrc`:
  - only if `LD_LIBRARY_PATH` is empty: `LD_LIBRARY_PATH=/usr/local/lib`
  - then: `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/itt/idl/bin/bin.linux.x86_64`
  - and: `export LD_LIBRARY_PATH`
13. Open IDL launching `/usr/local/itt/idl/bin.linux.x86_64/idl`

## 2.8 PDF

### 2.8.1 Readers

1. `sudo apt install okular`
2. `sudo apt install qpdfview`
3. `sudo apt install qpdf`
4. `sudo apt install pdfsam`
1. XPDF
  - `sudo apt install qt5-default`
  - Download and compile xpdf from the repo

### 2.8.2 Have hyperlinks open with Brave

1. `sudo kcmshell5 filetypes`
2. Select text > html
3. Put Brave on top of the list

### 2.8.3 PDF Presenter Console

```
sudo apt install pdfpc
```

### 2.8.4 Converting HTML to PDF

```
sudo apt install wkhtmltopdf
```

### 2.8.5 Universal converter

```
sudo apt install pandoc
```

## 2.8.6 PDFArranger

To rearrange the pages of a PDF:

1. `sudo add-apt-repository ppa:linuxuprising/apps`
2. `sudo apt update`
3. `sudo apt install pdfarranger`

## 2.9 Zoom

1. Download the package from <https://zoom.us/download?os=linux>
2. `sudo dpkg -i zoom_amd64.deb`
3. Launch Zoom
4. Identification: SSO
5. Domaine: u-paris.zoom.us
6. We are then redirected toward the authentication page from the University

## 2.10 Dropbox

- `cd ~ && wget -O - "https://www.dropbox.com/download?plat=lnx.x86_64" | tar xzf -`
- `~/dropbox-dist/dropboxd`

## 2.11 Images

### 2.11.1 ImageMagick

It is installed by default (instruction `convert` on the command line). However, it does not work on PS and PDF, by default. To fix this issue, edit `/etc/ImageMagick-6/policy.xml` and replace the rights `none` by `read|write` for PS, PDF, and EPS.

### 2.11.2 PNG compression

```
sudo apt install pngquant
```

### 2.11.3 Webp conversion

```
sudo apt install webp
```

### 2.11.4 FITS files

```
sudo apt install saods9
```

### 2.11.5 Diaporama

```
sudo apt install gwenview
```

### 2.11.6 Drawing

```
sudo apt install inkscape
```

### 2.11.7 Webcam

```
sudo apt update cheese
```

## 2.12 Music

- `sudo apt install audacity`
- `sudo apt install gtick`
- `sudo apt install musescore3`
- `sudo apt install lilypond`
- `sudo apt install evince`
- `sudo apt install timidity`
- `sudo apt install hydrogen`
- `sudo apt install jack`
- `sudo apt install timidity timidity-interfaces-extra`

## 2.13 Miscellaneous

- `sudo apt install gcal`
  - `sudo apt install vlc`
  - `sudo apt install ffmpeg`
  - `sudo apt install arp-scan`
  - `sudo apt install nmap`
- 

## 3 SETTING FILES TO IMPORT

### 3.1 Bashrc

Once the `~/ownCloud/` directory is synchronised, add in `~/.bashrc`:

```
export bashalias=$HOME/ownCloud/Settings/bashrc.sh
if [ -f $bashalias ]; then
    . $bashalias
fi
```

### 3.2 Emacs

- Import `~/emacsd/` (mainly `init.el` and `lisp/`)
- Import `~/saves/`
- Import `~/templates/`

### 3.3 Rest

- Import `.bashrc`
- Import `.mbsyncrc`
- Import `.vimrc`

### 3.4 Release Upgrade at CEA

1. `sudo apt update`
  2. `sudo systemctl stop puppet`
  3. `sudo systemctl disable puppet`
  4. `sudo rm -r /etc/apt/sources.list.d/`
  5. `sudo do-release upgrade`
  6. Accept the default variables at each prompt.
  7. Once it is done reboot and ask the computer people to check everything
  8. `sudo systemctl start puppet`
  9. `sudo systemctl enable puppet`
- 

## 4 TROUBLE SHOOTING

### 4.1 Snap

In general avoid snap. It is always better to install the deb from the repository if there is a problem with apt.

---