Ubuntu 18.04 LTS for Human Beans

This page collects information to have Ubuntu 18.04 LTS (Xenial Xerus) in a laptop or desktop as usable as possible for Human Beans willing to use it for production.

1.1. Enable Ubuntu Partners

First enable "partners" repos, since we will use it later to install skype, at least.

### Contents of the updated /etc/apt/sources.list

```bash
## Uncomment the following two lines to add software from Canonical's 'partner' repository.  
## This software is not part of Ubuntu, but is offered by Canonical and the respective vendors as a service to Ubuntu users.  
deb http://archive.canonical.com/ubuntu bionic partner  
deb-src http://archive.canonical.com/ubuntu bionic partner
```

1.2. Extra repositories

Useful extra repositories

### Command on a console

```bash
xavi@penguinbookpro:~$ sudo su
add-apt-repository -y ppa:nilarimogard/webupd8
add-apt-repository -y ppa:danielrichter2007/grub-customizer
cd
add-apt-repository -y ppa:webupd8team/java
add-apt-repository -y ppa:bit-team/stable
add-apt-repository -y ppa:jtaylor/keepass
add-apt-repository -y 'deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/'  
  # main binary packages for R 3.5.x
add-apt-repository -y ppa:marutter/c2d4u3.5  
  # extra binary packages for R 3.5.x from the usual marruter repo
add-apt-repository -y ppa:atareao/telegram
add-apt-repository -y ppa:maarten-baert/simplescreenrecorder
cd
add-apt-repository -y ppa:yg-jensge/shotwell
add-apt-repository -y ppa:mkusb/ppa
add-apt-repository -y ppa:hamishmb/myppa
add-apt-repository -y ppa:utappia/stable
add-apt-repository -y ppa:gezakovacs/ppa  
  # for unetbootin
add-apt-repository -y ppa:kubuntu-ppa/backports  
  # upgrades to major versions of kde apps

cd
# Add the release PGP keys for synctHING repo:
apt install curl

curl -s https://synctHING.net/release-key.txt | apt-key add -
```
# Add the "release" channel to your APT sources for updated-twice-per-month syncthing daemon:
echo "deb https://apt.syncthing.net/ syncthing release" | tee
/etc/apt/sources.list.d/syncthing.list
# Next 3 lines are for syncthing-gtk
sh -c "echo 'deb http://download.opensuse.org/repositories/home:/kozec/xUbuntu_18.04/
/' > /etc/apt/sources.list.d/home:kozec.list"
wget -nv
https://download.opensuse.org/repositories/home:kozec/xUbuntu_18.04/Release.key -O
Release.key
apt-key add - < Release.key
# Add the key for the new repo for R 3.5.x from cloud.r-project.org
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E084DAB9 # marutter
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 089EBE08314DF160 #
ubuntugis-stable
# Google Earth repo
sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/earth/deb/ stable main" >>
/etc/apt/sources.list.d/google.list'
# Setup the Google Key with:
wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -

# Update packages list again, just in case
apt-get update
Press enter each time when requested.

After keys are imported, you can quit from the root session with "exit":

root@computer:/home/xavi# exit
exit
xavi@computer:/home/xavi#

1.3. Install with interaction

Install programs which require interaction, like a confirmation step, a new password or user interaction of some sort.

- mysql-server, for the webserver in localhost, etc.
- phpmyadmin, to manage mysql databases easily through a web browser
- apachetop, for monitoring apache in real time
- ttf-mscorefonts-installer, required by other packages (PlayOnLinux, maybe?)
- a few others like java, and network filesystem network packages
If you need a webserver & php/mysql:

Command on a console

```bash
sudo apt-get install -y mysql-server phpmysqladmin apachetop postfix tasksel
tasksel install lamp-server
sudo mysql_secure_installation
```

If you don't run the `mysql_secure_installation` command, mysql root will not have password set yet, and you won't be able to access mysql through phpmysqladmin yet.

But for some reason, Ubuntu 18.04 comes with a different way to authenticate the root mysql user, and you may end up with problems to login to mysql with the password you just indicated in the steps above. Therefore, if that is your case also, you may need to run these steps to setup mysql root password as usual in earlier Ubuntu LTS versions:


1.4. Install the rest in program sets

You can then install all the rest of the required and useful software for common tasks (including video editing, screencasting, etc). Note that the first package names until wcatalan are localization packages for our local language, and you can skip them or adapt for your language, etc.

Command on a console

```bash
```
If you need webserver and other typical programs intended for servers, you can install this other set too:

**Command on a console**

```
```

If you want to have locale packages such as Catalan, you might be interested in installing these extra packages:

**Command on a console**

```
sudo apt-get install -y  aspell-ca hunspell-ca
```

And for Spanish:

**Command on a console**

```
```

### 1.4.1. R

Per omissió, Ubuntu 18.04 ve amb R 3.4.x. Però en els passos previs d'aquestes instruccions, s’han afegit ja els repositoris per a que la instal·lació d’R es faci amb la versió 3.5.x, que porta algunes millores interessants, desenvolupades inicialment a la branca ALTREP\(^2\) d’R.
Per poder instal·lar adequadament paquets d'R des de dins de la consola d'R et caldran paquets extra de sistema. I t'aniran bé algunes eines habituals associades (git, svn, etc). Pots posar les dependències habituals amb:

**Command on a console**

```bash
sudo apt-get install -y r-recommended r-cran-xml libgraphviz-dev libcairo2-dev r-cran-cairodevice freeglut3 freeglut3-dev r-cran-rglpk r-cran-rgl r-cran-misc3d libx11-dev libixt-dev libcurl4-openssl-dev libxml2-dev r-cran-xml libgraphviz-dev libcairo2-dev bwidget tk-table libv8-dev r-cran-rjava libmpfr-dev libc6 libssl-dev texlive-latex-extra texlive-lang-spanish libx11-dev libxml2-dev libxml2:i386 libixt-dev r-cran-misc3d subversion git texmaker tk-dev unaccent xvfb libgdal-dev libproj-dev r-cran-rmysql libmagick++-dev r-cran-rcolorbrewer r-cran-doparallel libssh2-1-dev libbudunits2-dev libgdal-dev libgeos-dev libproj-dev libgsl-dev libprotobuf-dev protobuf-compiler libssl-dev libcairo2-dev libglpk-dev dos2unix pandoc r-cran-rmio r-cran-gstat r-cran-maps r-cran-mapdata r-cran-ncdf4 r-cran-sp r-cran-raster r-cran-geor r-cran-ggmap r-cran-leaflet r-cran-rjags r-cran-snow r-cran-ggplot2 r-cran-igraph r-cran-lme4 r-cran-devtools r-cran-roxygen2 r-cran-rjava r-cran-xlsx
```

Una llista de paquets habituals d'R que et pot interessat tenir instal·lats de partida (un cop inicies una sessió d'R) són:

**Command on an R console**

```r
```

Si estàs actualitzant des d'una versió d'R anterior, et poden ser útils aquestes instruccions de terminal de sistema:

**Command on a console**

```bash
sudo chmod 777 /usr/lib/R/site-library /usr/lib/R/site-library/* -R
sudo chmod 777 /usr/lib/R/library /usr/lib/R/library/* -R
sudo chmod 777 /usr/share/R/doc/html/* -R
```

I aquestes altres de consola d'R:

**Command on an R console**

```r
update.packages(ask = F, lib="/usr/lib/R/site-library", checkBuilt = TRUE)
```
Cal tenir present que per omissió RStudio demana la contrasenya de Gitlab/Github cada vegada que vols fer un [push] dels commits locals cap.

### 1.4.2. Ús de credentials git via RStudio

Per omissió RStudio demana la contrasenya de Github cada vegada que vols fer un [push] dels commits locals cap a un repositori remot (com Gitlab o altres). Per que no ho demani cada vegada, pots dir-li que emprí l’anell de claus del sistema per a recuperar les credentials git, amb les següents instruccions:

#### Instruccions en terminal

```
sudo apt-get install libgnome-keyring-dev
sudo make --directory=/usr/share/doc/git/contrib/credential/gnome-keyring
git config --global credential.helper /usr/share/doc/git/contrib/credential/gnome-keyring/git-credential-gnome-keyring
```

### 1.5. Boot-repair (in multi-boot environments)

Totes instructions will save you some headache in a multi-boot environment when one OS changes the boot menu and doesn't allow you boot in your preferred OS. You can boot from an Ubuntu live CS/DVD/USB, and install boot-repair program from its own repo, with:

```
sudo add-apt-repository ppa:yannubuntu/boot-repair
sudo apt-get update
sudo apt-get install -y boot-repair && boot-repair
```

More information:

Non installed yet:

- jitsi -> installed by hand from their website, v 2.8.x
- linux-firmware-nonfree
- gstreamer1.0-plugins-bad-multiverse -> gstreamer1.0-plugins-bad

Other useful programs installed by hand in a later stage:

- rstudio
- chrome
- jitsi
- gdiskdump
Seeds For Change - seeds4c.org

- phpstorm
- acroread
- canon scangear MG3000. See Canon PIXMA MG3000 Printer and Scanner
- "Master pdf editor"
  (http://code-industry.net/free-pdf-editor.php[^5])
- elastic search

You can test email sending in the command line with:

```sh
echo testing | mail -s test_subject youremail@example.com
```

### 1.5.1. Main changes noticed for human beans

**Good:**

- Bluetooth works by default with most devices
- Battery icon displays battery info for connected Bluetooth devices using batteries (mouse, keyboard, etc)
- Icon to edit microphone volume levels shows up automatically on the top bar when the microphone is in use

**Confusing:**

- "USB startup creator" doesn't let you choose a persistent partition file when creating bootable USB disks with Ubuntu distros.

### 1.5.2. Extra R packages to be installed

Usual R packages that are useful in many cases. You can install them in an R console in a terminal (Type "R" in the terminal to launch the R console), or through RStudio command line.

```
if (!require(devtools)) install.packages("devtools"); library(devtools)
# Check Availability of package manager "PacMan" (and install it, if missing)
# if you have R < 3.5
if (version$major==3 && version$minor < 5) {
  # Install the devtools si et cal
  if (!require("devtools")) install.packages("devtools"); require("devtools")
  if (!require("pacman")) {
    cat("R Version: ", paste0(version$major,".",version$minor), ". ")
    cat("Tens una versió de R anterior a 3.5, per tant, instal.lam la versió
```
1.5.3. Install new php versions and switch between them

Ubuntu 18.04 comes with php 7.2, and some php applications might still fail with php 7.2. Therefore, in some cases, it might be sensible to have both php 7.0 and php 5.x, so that you can choose which version to use for your needs at any time. Or even better, 7.1. You need to add a new repository to have other versions of php available, and you will require the equivalent branch for a patched apache2 that will work with the other php version repository.

You can do so with:

```
xavi@computer# sudo su
root@computer# add-apt-repository ppa:ondrej/php
root@computer# add-apt-repository ppa:ondrej/apache2
root@computer# apt-get update
root@computer# apt-get install php5.6 php5.6-mysql php-gettext php5.6-mbstring php-xdebug libapache2-mod-php5.6 php5.6-curl php5.6-gd php5.6-mcrypt php5.6-xml php5.6-xmrpc libphp5.6-embed php-memcache php5.6-intl php5.6-zip php5.6-sqlite3 php5.6-zip
root@computer# apt-get install php7.2 php7.2-mysql php-gettext php7.2-mbstring php-xdebug libapache2-mod-php7.2 php7.2-curl php7.2-gd php7.2-mcrypt php7.2-xml php-xmllib parser php7.2-xmrpc libphp7.2-embed php-memcache php7.2-intl php7.2-zip php7.2-sqlite3 php7.2-zip
```
Eventually Installing both php5.6 & php7.* resulted in a lot of complaining from apt and lots of conflicts. The first suggested resolution was to remove all the stock php5 packages so that PHP 5.6 could be installed - so I just accepted the first suggestion.

The config files are all in /etc/php/5.6 and /etc/php/7.* respectively - inside here is where you can configure which extensions are loaded, set the ini settings, and everything else for each version in isolation.

So to switch from php 5.6 to php 7.2 you need to do two things:

```
# For php in web apps
user@computer:/$ sudo a2dismod php5.6; sudo a2enmod php7.2; sudo service apache2 restart
# For php-cli in the command line
user@computer:/$ sudo ln -sfn /usr/bin/php7.2 /etc/alternatives/php
user@computer:/$ php -v
# PHP 7.2.x (...)
# Copyright (c) 1997-2016 The PHP Group
# Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
```

or from php7.2 to php5.6:

```
# For php in web apps
user@computer:/$ sudo a2dismod php7.2 ; sudo a2enmod php5.6 ; sudo service apache2 restart
# For php-cli in the command line
user@computer:/$ sudo ln -sfn /usr/bin/php5.6 /etc/alternatives/php
user@computer:/$ php -v
# PHP 5.6.21-1+donate.sury.org-xenial+2 (cli)
# Copyright (c) 1997-2016 The PHP Group
# Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
#    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
```

From the commandline, I have both php5.6 and php7.2 available as commands. I also still have a php command - look in /etc/alternatives to see that it symlinks to a particular version of PHP cli*. You can also quickly check which yours is using by running php -v.
Remember that php.ini and other files with settings are not in /etc/php5 anymore but in /etc/php/\texttt{version}/

# 1.6. PHP Debugger

You may use a nice (but closed source) IDE such as PHPStorm (which provides a version at no cost for FLOSS projects such as Tiki), or stick with pure FLOSS apps such as Aptana Studio IDE.

See:

- **Download**:  
- **Installation**:  
- **Setup XDebug for PHP applications**  
- **Add svn integration**  

# 1.7. Fix Shutter editor

Shutter seems to be missing some perl dependencies in repositories to have the edit feature installed by default. You can add packages by hand (they work fine) with:

```
Commands in a terminal
```

```bash
 cd ~/tmp
 wget https://launchpad.net/ubuntu/+archive/primary/+files/libgoocanvas-common_1.0.0-1_all.deb
 wget https://launchpad.net/ubuntu/+archive/primary/+files/libgoocanvas3_1.0.0-1_amd64.deb
 sudo dpkg -i libgoocanvas-common_1.0.0-1_all.deb
 sudo dpkg -i libgoocanvas3_1.0.0-1_amd64.deb
 sudo apt-get install -y libextutils-depends-perl libextutils-pkgconfig-perl
 sudo dpkg -i libgoo-canvas-perl_0.06-2ubuntu3_amd64.deb
 sudo apt-get install -f
```

Then close shutter, and shutter indicator icon in the task bar, and restart it again. You'll be able to
1.8. Video Edition

The first program to try can be **OpenShot** (multiplatform).

But if you need more power/features, you can use **KDEnLive** (GNU/Linux only, and maybe MacOSX but not Windows).

And if you want to use a newer version than the one that comes with your distribution (either *ubuntu 16.04, 18.04, etc., or even with the Kubuntu-ppa/Backports repo), you can install KDEnLive by means of the **Flatpack** system, which works nicely in my first tests.

```
sudo add-apt-repository ppa:alexlarsson/flatpak
sudo apt update
sudo apt install flatpak # install flatpack software
sudo apt install gnome-software-plugin-flatpak # Install the Software Flatpak plugin (optional)
flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo # -- Add the Flathub repository
flatpak install flathub org.kde.kdenlive # Install KdenLive through Flatpack - KDENLive version 18.12.0 by the time of this writing December 2018 ! on top of Kubuntu 16.04
flatpak run org.kde.kdenlive # Run KdenLive through Flatpack
```

1.9. Dowload your own full YouTube Playlists

You can easily download your own videos of youtube (or even your entire playslists) through the application **youtube-dl**. You can install **youtube-dl** from repositories, even if it won't be updated as frequently as needed sometimes, to get sync with youtube way of working. Therefore, you are encouraged to install through the command-line:

```
Commands in a terminal window

sudo apt update && sudo apt install youtube-dl # install programs from repos if you don't have them yet
sudo wget https://yt-dl.org/downloads/latest/youtube-dl -O /usr/local/bin/youtube-dl # get latest version from developers site
sudo chmod a+rx /usr/local/bin/youtube-dl # fix permissions of the downloaded file
sudo youtube-dl -U # update your previous youtube-dl version
```

You can download your own youtube playlist with:

```
Commands in a terminal window

youtube-dl -ci "https://www.youtube.com/playlist?list=YOURPLAYLISTID" -o "%(title)s-"
Replace **YOURPLAYLISTID** with your own YouTube playlist id (note it's not your youtube channel ID, but the youtube playlist id). You can get it from the url of a video when shown through it's own playlist. That url might look like:

- https://www.youtube.com/watch?v=guFSSJTyZI&list=PLM2ch5TKp0OdTMEqCOJQw6BwJ19VDcj4&index=2
- https://www.youtube.com/playlist?list=PLM2ch5TKp0OdTMEqCOJQw6BwJ19VDcj4

Therefore, the playlist id is that part after `list=` and before any ampersand (`&`) if any.

In this case: **PLM2ch5TKp0OdTMEqCOJQw6BwJ19VDcj4**

Therefore, this would be the instruction in the command line:

```
Commands in a terminal window
youtube-dl -ci
"https://www.youtube.com/playlist?list=PLM2ch5TKp0OdTMEqCOJQw6BwJ19VDcj4" -o "%(title)s-%(id)s.%(ext)s"
```

And if you need in mp4 format only (because your tv doesn't accept some webm or mkv codec files, or whatever other reason), you can add these extra params:

```
Commands in a terminal window
youtube-dl -ci -o "%(title)s-%(id)s.%(ext)s" -f 'bestvideo[ext=mp4]+bestaudio[ext=m4a]/mp4'
"https://www.youtube.com/playlist?list=PLM2ch5TKp0OdTMEqCOJQw6BwJ19VDcj4"
```

Derived from:

- https://unix.stackexchange.com/a/328536

Looking for a GUI?

There is a nice GUI called youtube-dlg, however, it doesn't seem to be updated as frequently as needed to keep it updated to latest youtube-dl changes (and therefore, with youtube changes).

You might try the GUI Download-DLG, in case it still works for you (if did for me months ago, but not currently, so I had to go to the command line to be able to fetch my own video playlist successfully).
#sudo add-apt-repository ppa:nilarimogard/webupd8 # needed once for youtube-dlg
sudo apt update && sudo apt install youtube-dlg # install programs from repos if you
don't have them yet
# refresh youtube-dl from the commands above so that you have the latest version
installed

Derived from:


Alias names for this page:
playlist | playlists | Youtube | YoutubeLists | YoutubeList | Youtube Lists | Youtube List

1.10. Make Customized Ubuntu Live USB

See [UKC: Ubuntu Customization Kit](https://www.makeuseof.com/tag/ubuntu-customization-kit-linux-operating-system/), which is still available in the default ubuntu repositories in 16.04 and 18.04 LTS.

If you want to attempt to make Live USB with persistent changes, you can try making the usb with
mksusb (see below: #Create_Linux_Live_USB_disks)

1.10.1. Example for a OMDbuntu Live USB disk

Choose Lubuntu 18.04 64 bit as a base (it doesn't seem to work as expected for Lubuntu 16.04 since
it doesn't seem to be able to find internet repos to get packages from).
Desktop environment: Other
Lang: ca
Please choose customization action
1) Run console application
2) Continue building
Your choice (1..2)> 1

Extra system packages:

```
Extra system packages installed in a terminal window inside UCK

add-apt-repository -y ppa:webupd8team/java
add-apt-repository -y ppa:marutter/rrutter
apt install -y gfxboot-theme-ubuntu squashfs-tools syslinux-utils libnss-winbind
winbind cifs-utils oracle-java8-installer nfs-common git git-cola mc htop parcellite
kupfer p7zip-full printer-driver-cups-pdf meld sendmail gigolo
apt install -y r-recommended r-cran-xml libgraphviz-dev libcairo2-dev r-cran-
```
Seeds For Change - seeds4c.org

cairodevice freeglut3 freeglut3-dev r-cran-rgl r-cran-rgl package r-cran-misc3d libx11-dev
libx-t-dev libcurl4-openssl-dev libxml2-dev r-cran-xml libgraphviz-dev libcairo2-dev
bwidget tk-table libv8-dev r-cran-rrjava libmpfr-dev libc6 libssl-dev texlive-latex-extra
texlive-lang-spanish libx11-dev libxml2-dev libx-t-dev r-cran-misc3d subversion
get textmaker tk-dev unaccent xvfb libgdal-dev libproj-dev libmagick++-dev r-cran-
rcolorbrewer r-cran-doparallel libssh2-1-dev libbudunits2-1-dev libgdal-dev libgeos-dev
libproj-dev libjq-dev libprotobuf-dev protobuf-compiler libssl-dev libcairo2-dev
libapparmor-dev g++

# apt install -y r-cran-mysql # no troba el paquet slax a debian9
R CMD javareconf
wget https://download1.rstudio.org/rstudio-xenial-1.1.463-amd64.deb
apt install -y ./rstudio-xenial-1.1.463-amd64.deb

Extra R packages

Extra R packages installed in a terminal window inside UCK

if (!require(devtools)) install.packages("devtools"); library(devtools)
# Check Availability of package manager "PacMan" (and install it, if missing)
# if you have R < 3.5
if (version$major==3 & & version$minor < 5) {
  # Instal.la devtools si et cal
  if (!require("devtools"); install.packages("devtools"); require("devtools")
    cat("R Version: ", paste0(version$major,".",version$minor), ", ")
    cat("Tens una versió de R anterior a 3.5, per tant, instal.lem la versió
    correspons de PacMan (0.4.1)"
  )
  # Instal.la la darrera versió de pacman (0.4.1) que anava abans de la que va només
  amb R 3.5+ (pacman v0.5)
  if (!require("pacman") install_version("pacman", version = "0.4.1");
    require("pacman")
} else {
  # if you have R 3.5 or higher
  if (!require("pacman") install.packages("pacman"); require("pacman")
}
pl_load("devtools", "plotly", "Nozzle.R1", "VennDiagram", "stringr", "xml2",
"parallel", "data.table", "doParallel", "xtable", "plyr", "dplyr", "reshape2",
"rjson", "d3heatmap", "htmlwidgets", "googleVis", "doMC", "knitr", "checkpoint",
"sem", "rmarkdown", "relimp", "aplpack", "addinslist", "tidyverse", "magick",
"webshot", "pacman", "officer", "flextable", "huxtable", "RODBC", "leaflet",
"gitlabr", "colorspace", "mapdeck", "rsvg", "png", "cartography", "magick", "rJava",
"rio")

Before exiting the "-+Run console application+-", you need to update (or replace) the customization
script in order to make it work with latest Ubuntu versions (16.04+; derived from here)
Create a file called customize_iso2 in your home directory for instance:
Add these contents inside:

```
#!/bin/bash

# NAME:
#   customize_iso -- customize iso image outside of root FS
#
# SYNOOPSIS:
#   customize_iso [remaster_home] [customization_scripts_dir]
#
# DESCRIPTION:
#   This procedure performs additional customization that needs to
#   happen outside of the image of the root file system:
#   - Configuration of the boot environment
#     - isolinux configuration
#     - gfxboot configuration (language/keyboard support!)
#     - propagation of a new kernel/initrd
#     - Simple bootmanager (directory "install" in image)
#   - Configuration of additional resources outside the root FS
#     - Add-Ons (like sample document/music/video files)
#     - More Add-Ons (like OSS for Win / Mac)
#
# NOTES:
#   Execution of the procedure may require Internet access to download
#   the source of the latest version gfxboot-theme-ubuntu.
```
If a local copy of gfxboot-theme-ubuntu is available in the remaster home, it will be used instead of a (possibly updated) download.

```
SCRIPT_DIR=`dirname "$0"`
REMASTER_HOME=${1:-~/tmp}
SCRIPT_DIR=${2-$REMASTER_HOME/customization-scripts}
ISO_REMASTER_DIR="$REMASTER_HOME/remaster-iso"
REMASTER_DIR="$REMASTER_HOME/remaster-root"
BOOT_LANG=`cat "$SCRIPT_DIR/livecd_locale"

function failure()
{
    echo "$@
    exit 1
}

function get_latest_kernel()
{
    set -- $(ls "$REMASTER_DIR"/boot/vmlinuz* 2>/dev/null |
        sed -e "s@$REMASTER_DIR/boot/@"
        tr --  '-.' '	' |
        sort --key=2 --key=3 --key=4 --key=5 --numeric-sort |
        tail -n1 )
    [ "$1" = "vmlinuz" ] && echo "$REMASTER_DIR/boot/$1-$2.$3.$4-$5-$6"
}

# Create a temporary directory to assemble the gfxboot stuff in
BUILD_DIR=`mktemp -d`
if [ -d $REMASTER_HOME/gfxboot-theme-ubuntu ]
then
    cp -r $REMASTER_HOME/gfxboot-theme-ubuntu "$BUILD_DIR" ||
    failure "Cannot copy gfxboot-theme-ubuntu to $BUILD_DIR"
pushd "$BUILD_DIR" >/dev/null ||
    failure "Cannot change directory to $BUILD_DIR"
else
    pushd "$BUILD_DIR" >/dev/null ||
    failure "Cannot change directory to $BUILD_DIR"

    #-----HS (1)-->
    DISTRO_CODENAME=`cd "$ISO_REMASTER_DIR"/dists && find . -maxdepth 1 -type d |
        grep '/' | cut -d '/' -f2` ||
    # failure "Unable to identify Ubuntu distro code name"
    DISTRO_CODENAME="$(lsb_release -c|awk '{print $2}')"||
    failure "Unable to identify Ubuntu distro code name"
    #-----HS (1)<--
    APT_SOURCES_TMP_DIR=`mktemp -d`
    wget -c
    #-----HS (2)-->
Seeds For Change - seeds4c.org

# GFXBOOT_THEME_UBUNTU_SOURCE_PACKAGE=http://archive.ubuntu.com/ubuntu/ubuntu/ubuntu/pool/main/g/gfxboot-theme-ubuntu/$(zgrep gz "$APT_SOURCES_TMP_DIR"/Sources.gz | grep gfxboot-theme-ubuntu | sed -n 1p | awk '{ print $3 }')
    #-----HS (2)<--
    wget $GFXBOOT_THEME_UBUNTU_SOURCE_PACKAGE ||
    failure "Unable to download gfxboot-theme-ubuntu source package from $GFXBOOT_THEME_UBUNTU_SOURCE_PACKAGE"
    #-----HS (3)-->
    #tar xzf *.tar.gz ||
    tar xzf *.tar.gz
dpkg -x *.deb . ||
    failure "Unable to extract gfxboot-theme-ubuntu source package"
    #-----HS (3)<--
fi

# Build the gfx boot theme
    #-----HS (4)<--
    cd usr/share
gfxboot-theme-ubuntu
    cd po
    ln -s pt.po pt_PT.po
cd ..
    ##if [ "$BOOT_LANG" = "pt_PT" ]; then
    ##    make DEFAULT_LANG="pt" || failure "Failed to build gfxboot theme"
    ##else
    make DEFAULT_LANG="$BOOT_LANG" || failure "Failed to build gfxboot theme"
    ##fi

# Fix list of languages
pushd boot >/dev/null

# Create regexp that matches all language packs on CD
langpack=""
for langpack in `cat "$SCRIPT_DIR/language_packs"`; do
    if [ -z "$LANGPACKS" ]; then
        LANGPACKS="$langpack"
    else
        LANGPACKS="$LANGPACKS|$langpack"
    fi
done

# Rewrite langlist
cat "$SCRIPT_DIR/livecd_locales" >langlist
popd >/dev/null
# Copy to isolinux in image directory

```bash
cp -af boot/* "$ISO_REMASTER_DIR/isolinux/" ||
    failure "Error while copying boot files to $ISO_REMASTER_DIR/isolinux"
```

```bash
popd >/dev/null
```

# Cleanup

```bash
[ "$BUILD_DIR" != "/" ] && rm -rf "$BUILD_DIR"
```

# Copy kernel and initrd, in case it was changed during installation

```bash
VMLINUZ=$(get_latest_kernel)
```

```bash
if [ "$VMLINUZ" != "" ]
then
    INITRD="$REMASTER_DIR"/boot/initrd.img-$(`basename $VMLINUZ` | cut -d'-' -f 2-)
    if [ -e "$VMLINUZ" -a -e "$INITRD" ]
    then
        echo "Updating kernel:
        echo "  kernel="$VMLINUZ"
        echo "  initrd="$INITRD"
        cp -f "$VMLINUZ" "$ISO_REMASTER_DIR/casper/vmlinuz"
        cp -f "$INITRD" "$ISO_REMASTER_DIR/casper/initrd.gz"
        else
            echo "Not updating kernel as initrd not present"
        fi
    fi
fi
```

# Misc fixup for Karmic

```bash
pushd "$ISO_REMASTER_DIR"/isolinux >/dev/null
```

# What used to be called "message" is now called "bootlogo"

```bash
if [ -f isolinux.cfg -a -n "grep "gfxboot bootlogo" isolinux.cfg 2>/dev/null"
then
    if [ -f message ]
    then
        echo "Using bootlogo instead of message"
        mv message bootlogo
    fi
fi
```

# What used to be a gzipped initrd now is a lzma compressed initrd

```bash
if [ -f text.cfg ] || [ -f txt.cfg ]
then
    # At least one of the .cfg file will be missing. Drop error message.
    lzused=`grep initrd.lz text.cfg txt.cfg 2>/dev/null`
    lzmacmd=`which lzma`
    if [ -n "$lzused" ]
    then
        if [ -n "$lzmacmd" ]
then
  if [ -f ../casper/initrd.gz ]
  then
    pushd "$ISO_REMASTER_DIR"/casper >/dev/null
    echo "Recompressing initrd from gzip to lzma"
    rm -f initrd.lz
    gunzip <initrd.gz | $lzmacmd >initrd.lz
    rm -f initrd.gz
    popd >/dev/null
  fi
else
  if [ -f ../casper/initrd.gz ]
  then
    echo "lzma command not installed"
    echo "Switching permanently to gzipped initrd"
    sed -i -e 's/initrd\.lz/initrd.gz/g' text.cfg txt.cfg
    rm -f ../casper/initrd.lz
  else
    : do nothing - no initrd.gz
  fi
fi
else
  : do nothing - initrd.lz not used
fi

Then overwrite the customize_iso in uck with your own one:

```
Command to be run on a terminal window out of UCK

  cp /home/xavi/customize_iso2 /home/xavi/tmp/customization-scripts/customize_iso
```

Then you can continue with step 2 in:

```
Please choose customization action
1) Run console application
2) Continue building
Your choice (1..2) 2
```
1.11. Create Linux Live USB disks

1.11.1. With mkusb

You can add an extra repository to include `mkusb dus` (Do USB Stuff), which will allow you to create boot USB disks with persistence of changes in a reliable way for recent Ubuntu distros such as 16.04 LTS and newer. In such distros the default traditional methods such as USB Disk Creator or UnetBootIn no longer work in all cases as they did with earlier Ubuntu versions.

Derived from: [https://help.ubuntu.com/community/mkusb](https://help.ubuntu.com/community/mkusb)

```
sudo add-apt-repository ppa:mkusb/ppa  # and press Enter
sudo apt-get update
sudo apt-get install mkusb mkusb-nox usb-pack-efi
```


But there are many other pre-built images, such as Lubuntu 16.04 32bits and 64 bits: [https://help.ubuntu.com/community/mkusb/persistent/lubuntu](https://help.ubuntu.com/community/mkusb/persistent/lubuntu)

1.11.2. With Slax

- Get a 16Gb USB disk
- Make two partitions
  - leave 4Mb of free space at the beginning of USB disk
  - 1. ~8.5Gb ext4 or ext3
  - 2. ~7.5Gb ntfs
- Uncompress slax.iso in your Linux disk (with `isomaster` for instance)
- Copy your slax folder to your USB ext4 partition
- Run as root the file in your USB ext4 partition: `sudo bash ./slax/boot/bootinst.sh` to make your USB bootable
- Shutdown your computer, and boot it again this time from your USB stick. Slax should boot
- Make your desired changes following guidelines on slax.org website
- Change keyboard by hand tweaking a file in the persistent changes folder
  - Edit this file `/root/.fluxbox/startup` and add there near the top a line such as `setxkbmap es`
- Reboot to check that your changes were kept as desired
- Clone your usbdisk (with partitions and all) with `dd`
  - Find your USB disk device name (sdc in my case). You can use command `sudo lsblk` or `sudo fdisk -l`
• make image with dd. Example:
```bash
sudo dd if=/dev/sdc of=/media/xavi/mollero_ext4/tmp/backup_slaxomdr.img bs=4M status=progress
```

• clone back over another usb with dd again (or theoretically with a dd GUI, but gdiskdump fails for me on Kubuntu 16.04)
To restore the image to another USB drive just invert the process. Just make sure that the new USB drive (/dev/sdd in this example) is as big or bigger than the original one
```bash
sudo dd if=/media/xavi/mollero_ext4/tmp/backup_slaxomdr.img of=/dev/sdd bs=4M status=progress
```

### 1.12. Mapes mentals

Una aplicació de KDE que sembla molt recomanada és **Semantik**. Respecte a les aplicacions multiplataforma, les més recomanades serien FreeMind, o Xmind.

#### 1.12.1. Semantik

Veure: [https://waf.io/semantik.html](https://waf.io/semantik.html)[23]

**Features:**

- Open-source with source code on GitHub and available for download here.
- Mind maps are either in tree or 2-dimension form.
- Create UML diagrams, flowcharts, technical documentation, etc.
- Generate “flat” documents such as reports and presentations.
- Supports LaTeX, OpenOffice, and HTML document formats.
- Export diagrams in PDF or PNG format.
- Command line tools for document exports.

**Essential commands with Semantik:**

- Double click to add a concept, type "Enter" to edit it
- Double click on a leaf or use the link mode to connect concepts
- Double click on a link to remove it
- Middle button + move to scroll on the map
- Scroll wheel for zooming
- When a concept is selected, press enter to start editing it
- Control+R to reorganize the map
- Control+H to center the zoom on selected objects
- When a map is ready, use Ctrl+G to generate a document
1.12.2. XMind

Hi ha un projecte en github d’un instal·lador en bash (https://github.com/mriza/XMind-Linux-Installer\(^{[24]}\), al que cal fer només una petita acció posterior per a que tot funcioni (al menys en el Kubuntu 18.04 que jo he provat i documentat ). Cal eliminar el contingent d’aquesta línia:

```bash
--add-modules=java.se.ee
```

de l’arxiu que genera a `/opt/xmind/XMind_amd64/XMind.ini`

Passos a seguir:

1. Cal tenir instal·lats un paquets de sistema base (que en teoria ja els tens si has seguit els passos anteriors):
   ```bash
   sudo apt-get install -y unzip default-jre libgtk2.0-0 libwebkitgtk-1.0-0 lame libc6 libglib2.0-0
   ```

2. Baixat el projecte de codi via git:
   ```bash
   Instrucció en terminal de Linux
   git clone https://github.com/mriza/XMind-Linux-Installer.git
   ```

3. Download XMind from http://www.xmind.net/download/linux\(^{[26]}\) (la versió 8 ja va bé)
4. Add execute permission to the installer script and run it,
   ```bash
   Instrucció en terminal de Linux
   chmod +x xmind-installer.sh; sudo ./xmind-installer.sh elteuusuariubuntu
   ```

Edita l’arxiu `/opt/xmind/XMind_amd64/XMind.ini` per eliminar aquesta línia:

```bash
--add-modules=java.se.ee
```

I ja hauria d’estar. Pots obrir XMind amb `Alt` + `Barra Espaiadora` i escriure XMind.

A mi m’ha funcionat amb Java 1.8.

1.13. WPS-Office

WPS Office is an office productivity suite, including Writer, Presentation and Spreadsheets. WPS
Office is a native snap package in Ubuntu. To install it in all currently supported versions of Ubuntu open the terminal and type:

```
# Option 1
sudo snap install wps-office-all-lang-no-internet # all interface languages & check
     spell tools and without internet connection to their servers

## Option 2
#sudo snap install wps-office-multilang # all interface languages and with internet
     connection to their servers

## Option 3
#sudo snap install wps-office # only english interface and with internet connection
     to their servers
```

The wps-office snap package will be updated automatically when updates are available.

This is a multilanguage snap package for wps-office. It contains all interface languages available for
the Windows version. This package is based on the wps-office-all-lang-no-internet package and the
official wps-office package. The official wps-office can be found at https://snapcraft.io/wps-office.

The package wps-office-all-lang-no-internet can be found at
https://snapcraft.io/wps-office-all-lang-no-internet or
package is shipped without internet plug. So there is no risk of a backdoor.

This package can be found at https://snapcraft.io/wps-office-multilang and

### 1.14. Element Secure Messenger

A new world of messaging and collaboration
No datamining - Own your conversations, choose where they live
No walled gardens - Talk to anyone, whatever app they are using
No eavesdropping - Be sure you’re talking to the right people, in private

Typical messaging apps own your conversations, mine your data and lock you into a walled garden.
Element is not a typical messaging app.
Element is unique. You own your data and decide where it lives. Element lets you talk securely to
anyone, regardless of their app.
Personal use - 20M+ users worldwide
Communities - Communities from fan clubs to Firefox
Business - 2M+ business users
Public sector - Trusted by French, US and German governments
https://element.io/get-started
sudo apt install -y wget apt-transport-https
echo "deb [signed-by=/usr/share/keyrings/riot-im-archive-keyring.gpg] https://packages.riot.im/debian/ default main" | sudo tee /etc/apt/sources.list.d/riot-im.list
sudo apt update
sudo apt install element-desktop

Pending to be reviewed

[12] https://www.youtube.com/watch?v=guFSSJTyZI&list=
[15] https://unix.stackexchange.com/a/328536
[18] https://askubuntu.com/a/699474
[22] https://help.ubuntu.com/community/mkusb/persistent/lubuntu
[23] https://waf.io/semantik.html
Seeds For Change - seeds4c.org

[32] https://element.io/get-started
[34] https://www.youtube.com/watch?v=l8xa-PkDhhk
[38] http://ardownload.adobe.com/pub/adobe/reader/unix/9.x/9.5.5/enu/AdbeRdr9.5.5-1_i386linux_enu.deb
[39] https://sourceforge.net/projects/portablesigner/
[40] http://www.sinadura.net/es/inicio
[42] https://www.elastic.co/guide/en/elasticsearch/reference/5.0/deb.html#deb-repo
[45] https://www.elastic.co/guide/en/kibana/5.0/deb.html
[47] https://www.howtoforge.com/tutorial/multiple-usb-writing-on-linux/
[48] https://itsfoss.com/multiple-linux-one-usb/
[50] https://github.com/mbusb/multibootusb/releases/download/v9.2.0/python3-multibootusb_9.2.0-1_all.deb
[52] https://humdi.net/vnstat/
[53] https://github.com/phw/peek