

Debian Post-install

Sergio Alvariño salvari@gmail.com

julio-2016

Resumen

Instalación de Debian
Solo para referencia rápida y personal.

Introducción

Mi portátil es un ordenador Acer 5755G con las siguientes características:

- Core i5 2430M 2.4GHz
- NVIDIA Geforce GT 540M
- 8Gb RAM
- 750Gb HD

La gráfica es una Nvidia Optimus, es decir una tarjeta híbrida que funciona perfectamente en Ubuntu 14.04 usando Bumblebee.

Para hacer la actualización del sistema opté por desinstalar el dvd y montar en su lugar un disco SSD en un Caddie para Acer. La instalación fué muy fácil, y aunque el portátil arranca perfectamente de cualquiera de los dos discos opté por instalar el SSD en la bahía del HD original y pasar el HD al caddie.

Comentar los problemas con calentamiento en Ubuntu

Comentar la creación de usb bootable

Lo primero fue la instalación del Bumblebee

firmware-linux-nonfree Bumblebee-nvidia primus

Instalación

Gestión de paquetes

Habilitar backports

Aptitude

Synaptic

Instalado git desde aptitude

Instalado terminator

Documentos

Pandoc

Instalado el Pandoc descargando paquete *deb* desde la página web del Pandoc.

Descargamos las plantillas desde el repo ejecutando los siguientes comandos:

```
cd ~/.pandoc
git clone https://github.com/jgm/pandoc-templates templates
```

Vanilla LaTeX

El LaTeX de Debian está un poquillo anticuado, si se quiere usar una versión reciente hay que aplicar este truco.

```
cd ~
mkdir tmp
cd tmp
wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
tar xzf install-tl-unx.tar.gz
cd install-tl-xxxxxx
```

La parte xxxxxx varía en función del estado de la última versión de LaTeX disponible.

```
sudo ./install-tl
```

Una vez lanzada la instalación podemos desmarcar las opciones que instalan la documentación y las fuentes. Eso nos obligará a consultar la documentación *on line* pero ahorrará practicamente el 50% del espacio necesario. En mi caso sin *doc* ni *src* ocupa 2,3Gb

```
mkdir -p /opt
sudo ln -s /usr/local/texlive/2016/bin/* /opt/texbin
```

Por último para acabar la instalación añadimos **/opt/texbin** al *path*.

Falsificando paquetes

Ya tenemos el **texlive** instalado, ahora necesitamos que el gestor de paquetes sepa que ya lo tenemos instalado.

```
sudo apt-get install equivs --no-install-recommends
mkdir -p /tmp/tl-equivs && cd /tmp/tl-equivs
equivs-control texlive-local
```

Para hacerlo más fácil podemos descargarnos un fichero ya preparado, ejecutando:

```
wget http://www.tug.org/texlive/files/debian-equivs-2015-ex.txt
/bin/cp -f debian-equivs-2015-ex.txt texlive-local
```

Editamos la versión y

```
equivs-build texlive-local
sudo dpkg -i texlive-local_2015-1_all.deb
```

Todo listo, ahora podemos instalar cualquier paquete que dependa de texlive

Fuentes

Para dejar disponibles las fuentes opentype y truetype que vienen con texlive para el resto de aplicaciones:

```
sudo cp $(kpsewhich -var-value TEXMFSYSVAR)/fonts/conf/texlive-fontconfig.conf /etc/fonts/conf.d/09-texlive.conf
gksudo gedit /etc/fonts/conf.d/09-texlive.conf
```

Borramos la línea:

```
<dir>/usr/local/texlive/2016/texmf-dist/fonts/type1</dir>
```

Y ejecutamos:

```
sudo fc-cache -fsv
```

Actualizaciones

Para actualizar nuestro latex a la última versión de todos los paquetes:

```
sudo /opt/texbin/tlmgr update --self
sudo /opt/texbin/tlmgr update --all
```

También podemos lanzar el instalador gráfico con:

```
sudo /opt/texbin/tlmgr --gui
```

Para usar el instalador gráfico hay que instalar previamente:

```
sudo apt-get install perl-tk --no-install-recommends
```

Lanzador para el actualizador de texlive

```
mkdir -p ~/.local/share/applications
/bin/rm ~/.local/share/applications/tlmgr.desktop
cat > ~/.local/share/applications/tlmgr.desktop << EOF
[Desktop Entry]
Version=1.0
Name=TeX Live Manager
Comment=Manage TeX Live packages
GenericName=Package Manager
Exec=gksu -d -S -D "TeX Live Manager" '/opt/texbin/tlmgr -gui'
Terminal=false
Type=Application
Icon=system-software-update
EOF
```

Ojo que hay que dejar instalado el gksu:

```
sudo aptitude install gksu
```

Emacs

Instalado emacs desde aptitude emacs

Instalado chrome añadiendo fuentes a aptitude, hay que borrar el fichero que sobra. chrome

comentado cdrom en sources.list

Instalado keepass2

Instalado terminator

instalado gksu

instalado fish

configurado d-apt, instalados todos los programas incluidos

gimp ya estaba instalado, instalado el gimp data-extra

```
solventado deb multimedia sudo apt-get -y --allow-unauthenticated install -
reinstall deb-multimedia-keyring
```

```
sudo apt-get install libav-tools
```

```
sudo apt-get install faad gstreamer0.10-ffmpeg gstreamer0.10-x gstreamer0.10-
fluendo-mp3 gstreamer0.10-plugins-base gstreamer0.10-plugins-good gstreamer0.10-
plugins-bad gstreamer0.10-plugins-ugly ffmpeg lame twolame vorbis-tools
libquicktime2 libfaac0 libmp3lame0 libxine2-all-plugins libdvread4 libdvnav4
libmad0 sox libxvidcore4 libstdc++5
```

sudo apt-get install w64codecs

sudo apt-get install rar unrar zip unzip unace bzip2 lzop p7zip p7zip-full p7zip-rar

Desarrollo sw

sudo apt-get install build-essential checkinstall make automake cmake autoconf git git-core dpkg wget Instalado apt-d y todos los programas asociados

Open Java

apt-get install openjdk-7-jre icedtea-7-plugin

compresores

Disk manager apt-get install ntfs-3g disk-manager

apt-get -t jessie-backports install gnucash

apt-get install rsync grsync

Docker

apt-get install apt-transport-https ca-certificates apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D
edit docker.list with deb https://apt.dockerproject.org/repo debian-jessie main

apt-cache policy docker-engine -- comprobamos que todo está bien.

sudo apt-get install docker-engine -- da un error en makedev por udev activo

sudo service docker start

sudo docker run hello-world - todo bien

sudo gpasswd -a salvari docker

apt-cache policy inkscape apt-get -t jessie-backports install inkscape

apt-get install libreCAD

apt-get -t jessie-backports install freecad

Instalado calibre

Shells alternativos: zsh y fish

fish

Instalamos **fish** desde aptitude con:

```
sudo aptitude install fish
```

zsh

Igualmente instalamos **zsh**:

```
sudo aptitude install zsh
```

Arrancamos **zsh** desde un terminal:

```
/usr/bin/zsh
```

Vamos a usar antigen así que nos lo clonamos en `__~/apps/`

```
cd ~/apps
```

```
git clone https://github.com/zsh-users/antigen
```

Y editamos el fichero `~/.zshrc` para que contenga:

```
source ~/apps/antigen/antigen.zsh
```

```
# Load the oh-my-zsh's library.
```

```
antigen use oh-my-zsh
```

```
# Bundles from the default repo (robbyrussell's oh-my-zsh).
```

```
antigen bundle git
```

```
antigen bundle command-not-found
```

```
antigen bundle autojump
```

```
antigen bundle extract
```

```
# antigen bundle heroku
```

```
# antigen bundle pip
```

```
# antigen bundle lein
```

```
# Syntax highlighting bundle.
```

```
antigen bundle zsh-users/zsh-syntax-highlighting
```

```
# git
```

```
antigen bundle arialdomartini/oh-my-git
```

```
antigen theme arialdomartini/oh-my-git-themes oppa-lana-style
```

```
# autosuggestions
```

```
antigen bundle tarruda/zsh-autosuggestions
```

```
#antigen theme agnoster
```

```
# Tell antigen that you're done.
```

```
antigen apply
```

```
# append to path
path+=('/home/salvari/apps/julia/current/bin/')
# prepend
# path=('/home/salvari/bin/' $path)
# export PATH
```

Antigen ya se encarga de descargar todo lo que queramos utilizar en zsh.

Nos queda arreglar las fuentes para que funcione correctamente la linea de estado en los repos de git. Necesitamos una fuente *Awesome*

Cambiar las opciones de idioma

Cambiar lenguaje /etc/locale-gen sudo locale-gen

Cuentas online abiertas

- google
- pocket (plugin de chrome)

TODO

- emacs
- zsh
- cinelerra
- rewrap
- zotero
- playonlinux
- darktable
- rawtherapee
- krita
- mypaint

Inkscape <https://elizsarobhasa.makes.org/thimble/MTMwNDIzMjE5Mg==/3d-printing-from-a-2d-drawing> Instalar tb jessyink

tor openframeworks

chibios * http://wiki.chibios.org/dokuwiki/doku.php?id=chibios:community:setup:openocd_chibios
* <http://www.josho.org/blog/blog/2014/11/30/nucleo-gcc/> * <http://www.stevebate.net/chibios-rpi/GettingStarted.html>

Links

[Systemd](#) [Gnome shortcuts](#) [Gnome optimizaciones](#) [Instalación Debian](#) [zsh](#) [zsh](#)