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

Gestión de paquetes

Instalamos *aptitude* y *synaptic*

```
sudo apt-get install aptitude
sudo apt-get install synaptic
```

Cambiamos las opciones de *aptitude* para que **no instale** los paquetes recomendados.

Quitamos el cdrom de los sources.list

Editamos el fichero */etc/apt/sources.list* y comentamos las lineas del cdrom.

Habilitamos los backports y multimedia

Backports:

```
sudo cat > /etc/apt/sources.list.d/backports.list << EOF
# backports
deb http://ftp.debian.org/debian/ jessie-backports main contrib non-free
EOF
```

Multimedia:

```
sudo cat >> /etc/apt/sources.list.d/multimedia.list << EOF
# multimedia
deb http://www.deb-multimedia.org/ jessie main non-free
EOF
```

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

Y actualizamos

```
sudo aptitude update
```

Instalación de varios paquetes sueltos

Instalado git desde aptitude

```
sudo aptitude install git
```

Configuración básica de **git**

```
git config --global user.name "Sergio Alvariño"
git config --global user.email "salvari@gmail.com"
git config --global core.editor emacs
git config --global color.ui true
git config --global credential.helper cache
git config --global credential.helper 'cache --timeout=7200'
git config --global push.default simple
```

```
git config --global alias.sla 'log --oneline --decorate --graph --all'
git config --global alias.car 'commit --amend --no-edit'
git config --global alias.unstage reset
git config --global alias.st status
git config --global alias.last  'log -1 HEAD'
git config --global alias.ca 'commit -a'
```

Instalado terminator

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

Instalado keepass2

instalado gksu

Diskmanager:

```
sudo apt-get install ntfs-3g disk-manager
```

Gnucash:

```
sudo apt-get -t jessie-backports install gnucash
```

Herramientas *sync*:

```
sudo apt-get install rsync grsync
```

Menu Libre: Un editor de menús para Gnome

```
sudo apt-get install menulibre
```

Tor

Bajado el comprimido desde la web y descomprimido en *~/apps* copiado el fichero desktop a *~/.local/share/applications*

Codecs

```
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 libdvdread4 \
libdvdnav4 libmad0 sox libxvidcore4 libstdc++5
```

```
sudo apt-get install w64codecs
```

Compresores et al

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

Gráficos

Inkscape

```
apt-cache policy inkscape
apt-get -t jessie-backports install inkscape
aptitude install ink-generator
```

Librecad

Instalado desde repos con aptitude

```
apt-get install librecad
```

```
apt-get -t jessie-backports install freecad
```

Gimp

Gimp ya estaba instalado, adicionalmente instalado el gimp data-extra

```
sudo aptitude install gimp-plugin-registry gimp-texturize gimp-data-extras gimp-gap
```

Fotografía

Rawtherapee y Darktable:

```
sudo aptitude install icc-profiles icc-profiles-free
sudo aptitude install rawtherapee darktable
```

Música

Clementine, decibel, audacity, soundconverter

```
sudo aptitude install clementine gstreamer0.10-plugins-bad
sudo aptitude install decibel-audio-player audacity soundconverter
```

```
sudo aptitude install recordmydesktop gtk-recordmydesktop
sudo aptitude install handbrake handbrake-cli handbrake-gtk
```

Documentos

Calibre

Ejecutamos lo que manda la página web:

```
sudo -v && wget -nv -O- https://raw.githubusercontent.com/kovidgoyal/calibre/master/setup/linu
| sudo python -c "import sys; main=lambda:sys.stderr.write('Download failed\n'); exec(sys.stdi
```

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/con
gksudo gedit /etc/fonts/conf.d/09-texlive.conf
```

Borramos la linea:

```
<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 (aunque debería estar de antes si sigues este doc)

```
sudo aptitude install gksu
```

Emacs

Instalado emacs desde los repos:

```
sudo aptitude install emacs
```

Instalamos los paquetes *markdown-mode*, *mardown-plus* y *pandoc-mode* desde el menú de gestión de paquetes de **emacs**.

También instalamos *d-mde* y *flymake-d*. Hay una sección de configuración en el fichero *.emacs*.

Configuramos el fichero *.emacs* definimos algunas preferencias, algunas funciones útiles y añadimos orígenes extra de paquetes.

```
(custom-set-variables
 ;; custom-set-variables was added by Custom.
 ;; If you edit it by hand, you could mess it up, so be careful.
 ;; Your init file should contain only one such instance.
 ;; If there is more than one, they won't work right.
 '(column-number-mode t)
 '(show-paren-mode t))
(custom-set-faces
 ;; custom-set-faces was added by Custom.
 ;; If you edit it by hand, you could mess it up, so be careful.
 ;; Your init file should contain only one such instance.
 ;; If there is more than one, they won't work right.
```

```
 '(default ((t (:family "Mensch" :foundry "bitstream" :slant normal :weight normal :height 128 :
;;-------------------------------------------------------------
;; Some settings
(setq inhibit-startup-message t) ; Eliminate FSF startup msg
(setq frame-title-format "%b")   ; Put filename in titlebar
;(setq visible-bell t)            ; Flash instead of beep
(set-scroll-bar-mode 'right)     ; Scrollbar placement
(show-paren-mode t)          ; Blinking cursor shows matching parentheses
(setq column-number-mode t)  ; Show column number of current cursor location
(mouse-wheel-mode t)          ; wheel-mouse support

(setq fill-column 78)
(setq auto-fill-mode t)          ; Set line width to 78 columns...

(setq-default indent-tabs-mode nil)     ; Insert spaces instead of tabs
(global-set-key "\r" 'newline-and-indent) ; turn autoindenting on
;(set-default 'truncate-lines t)        ; Truncate lines for all buffers
(require 'iso-transl)

;;-------------------------------------------------------------
;; Some useful key definitions
(define-key global-map [M-S-down-mouse-3] 'imenu)
(global-set-key [C-tab] 'hippie-expand)                    ; expand
(global-set-key [C-kp-subtract] 'undo)                     ; [Undo]
(global-set-key [C-kp-multiply] 'goto-line)              ; goto line
(global-set-key [C-kp-add] 'toggle-truncate-lines)       ; goto line
(global-set-key [C-kp-divide] 'delete-trailing-whitespace) ; delete trailing whitespace
(global-set-key [C-kp-decimal] 'completion-at-point)     ; complete at point
(global-set-key [C-M-prior] 'next-buffer)              ; next-buffer
(global-set-key [C-M-next] 'previous-buffer)           ; previous-buffer

;;-------------------------------------------------------------
;; Set encoding
(prefer-coding-system 'utf-8)
(setq coding-system-for-read 'utf-8)
(setq coding-system-for-write 'utf-8)

;;-------------------------------------------------------------
;; Maximum colors
(cond ((fboundp 'global-font-lock-mode)  ; Turn on font-lock (syntax highlighting)
     (global-font-lock-mode t)           ; in all modes that support it
       (setq font-lock-maximum-decoration t))) ; Maximum colors

;;-------------------------------------------------------------
;; Use % to match various kinds of brackets...
;; See: http://www.lifl.fr/~hodique/uploads/Perso/patches.el
```

```
(global-set-key "%" 'match-paren)             ; % key match parents
(defun match-paren (arg)
  "Go to the matching paren if on a paren; otherwise insert %."
  (interactive "p")
  (let ((prev-char (char-to-string (preceding-char)))
        (next-char (char-to-string (following-char))))
    (cond ((string-match "[[{(<]" next-char) (forward-sexp 1))
          ((string-match "[\]})>]" prev-char) (backward-sexp 1))
          (t (self-insert-command (or arg 1))))))

;;-------------------------------------------------------------
;; The wonderful bubble-buffer
(defvar LIMIT 1)
(defvar time 0)
(defvar mylist nil)

(defun time-now ()
   (car (cdr (current-time))))

(defun bubble-buffer ()
   (interactive)
   (if (or (> (- (time-now) time) LIMIT) (null mylist))
       (progn (setq mylist (copy-alist (buffer-list)))
          (delq (get-buffer " *Minibuf-0*") mylist)
          (delq (get-buffer " *Minibuf-1*") mylist)))
   (bury-buffer (car mylist))
   (setq mylist (cdr mylist))
   (setq newtop (car mylist))
   (switch-to-buffer (car mylist))
   (setq rest (cdr (copy-alist mylist)))
   (while rest
     (bury-buffer (car rest))
     (setq rest (cdr rest)))
   (setq time (time-now)))

(global-set-key [f8] 'bubble-buffer)    ; win-tab switch the buffer

(defun geosoft-kill-buffer ()
   ;; Kill default buffer without the extra emacs questions
   (interactive)
   (kill-buffer (buffer-name))
   (set-name))
(global-set-key [C-delete] 'geosoft-kill-buffer)

;;-----------------------------------------------------------------------
```

```
;; MELPA and others
(when (>= emacs-major-version 24)
  (require 'package)
  (package-initialize)
 (add-to-list 'package-archives '("melpa" . "http://melpa.org/packages/") t)
 (add-to-list 'package-archives '("gnu" . "http://elpa.gnu.org/packages/") t)
 (add-to-list 'package-archives '("marmalade" . "https://marmalade-repo.org/packages/") t)
  )

;;------------------------------------------------------------------------
;; flymake installed from package

(require 'flymake)
(global-set-key (kbd "C-c d") 'flymake-display-err-menu-for-current-line)
(global-set-key (kbd "C-c n") 'flymake-goto-next-error)
(global-set-key (kbd "C-c p") 'flymake-goto-prev-error)

;; Activate flymake for D
(add-hook 'd-mode-hook 'flymake-d-load)
```

Scribus

Instalado con aptitude

```
sudo aptitude install scribus
```

Desarrollo sw

Paquetes esenciales

```
sudo apt-get install build-essential checkinstall make automake cmake autoconf git git-core dpk
```

Open Java

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

D-apt e instalación de programas

configurado d-apt, instalados todos los programas incluidos

```
sudo wget http://master.dl.sourceforge.net/project/d-apt/files/d-apt.list -O /etc/apt/sources
sudo apt-get update && sudo apt-get -y --allow-unauthenticated install --reinstall d-apt-keyrin
```

Instalamos todos los programas asociados.

Arduino y Processing

Bajamos los paquetes de las respectivas páginas web, descomprimimimos en *~/apps/* y creamos los desktop file con **Menulibre**

Openframeworks

Bajamos el paquete comprimido de la página web del proyecto.

Descomprimimos en *~/apps*

Bajamos al directorio de la aplicación y ejecutamos:

```
sudo  scripts/linux/debian/install_dependencies.sh
sudo  scripts/linux/debian/install_codecs.sh

cd scripts/linux
./compileOF.sh -j2

cd OF/examples/graphics/polygonExample
make
make Run

cd OF/scripts/linux
./compilePG.sh
```

Va a instalar un montón de dependencias, hay que tomarlo con calma.

Al final también va a añadir una linea al fichero *~/.profile*

```
export PG_OF_PATH=/home/salvari/apps/of/of_v0.9.3_linux64_release
```

Docker

```
apt-get install apt-transport-https ca-certificates
apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C07
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
```

Shells alternativos: zsh y fish

Los dos son muy interesantes. He usado zsh casi un año, ahora voy a probar **fish**.

fish

Instalamos **fish** desde aptitude con:

```
sudo aptitude install fish
```

Instalamos oh-my-fish

```
curl -L https://github.com/oh-my-fish/oh-my-fish/raw/master/bin/install > install
fish install
rm install

chsh -s `which fish`
```

zsh

Igualmente instalamos **zsh**:

```
sudo aptitude install 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*

Instalación de fuentes adicionales

Nos bajamos unas cuantas fuentes que soporten los iconos *Awesome*.

```
cd ~/tmp
git clone https://github.com/abertsch/Menlo-for-Powerline
git clone https://github.com/powerline/fonts

mkdir ~/.fonts
cp someFontFile ~/.fonts/
fc-cache -vf ~/.fonts/
```

Cambiar las opciones de idioma

Ejecutamos:

```
sudo dpkg-reconfigure locales
```

Y después solo tenemos que cambiar la selección del idioma en la configuración de Gnome.

Nos pedirá rearrancar Gnome y renombrará todos los directorios de sistema.

Reprap

Sl1c3r

Descargamos el paquete binario desde la página web.

- Cambiar permisos en directorio */lib/vrt/*
- Instalado *lib-canberra-module* desde aptitude
- Es necesario instalar *freeglut*

OpenScad

Instalado desde aptitude.

Printrun

Descargamos desde github

```
sudo apt-get install python-serial python-wxgtk2.8 python-pyglet python-numpy \
cython python-libxml2 python-gobject python-dbus python-psutil python-cairosvg git

python setup.py build_ext --inplace
```

Python

Instalado python-pip y python-virtualenv desde aptitude.

Cuentas online abiertas

- google

- pocket (plugin de chrome)

TODO

- cinelerra
- zotero
- playonlinux
- darktable
- rawtherapee
- krita
- mypaint

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

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

rclone [https://syncthing.net/]

vmware

sudo aptitude install chromium

Links

Systemd Gnome shortcuts Gnome optimizaciones Instalación Debian zsh zsh

https://www.roaringpenguin.com/products/remind http://taskwarrior.org/