



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Todo lo que siempre quiso saber sobre Linux en el iBook... y no se atrevían a responder (25919 lectures)

Per **Ricardo Galli Granada**, [gallir](http://mnm.uib.es/gallir/) (<http://mnm.uib.es/gallir/>)

Creado el 06/09/2002 01:48 modificado el 08/09/2002 13:01

*Dado el éxito que están teniendo los iBooks de Apple entre la gente de Bulma (y Linux en general), y que la típica pregunta en las kedadas es ¿como tienes configuradas las X? ¿como haces funcionar los botones especiales de Apple? ¿que kernel usas?. Como soy un incauto, prometí a un bulmero darle la configuración de todo lo importante (eran horas muy avanzadas y no había bebido nada :-), me decidí a dejarlo escrito en un artículo (**basado y probado sobre una Debian Sid**).*

Respuestas rápidas: ¿vale la pena comprar un iBook para Linux? SI. ¿Como de rápido es?. Yo tengo un G3 a 500MHz, y lo `_percibo_` bastante más rápido que un P3 a 500 MHz.



ACTUALIZACIÓN: salieron las XFree versión 4.2.1 con soporte acelerado usando el driver "ati" , y **también disponible para Debian Sid** (versión pre de Branden). Ya actualicé la [página de configuración de las X](#)⁽¹⁾.

Índice

1. [Introducción](#)⁽²⁾
2. [Configuración de las X](#)⁽¹⁾
3. [Configuración del kernel](#)⁽³⁾
4. [Configuración para emulación de 3 botones y las modutils](#)⁽⁴⁾
5. [Control de energía](#)⁽⁵⁾
6. [Los botones de Apple](#)⁽⁶⁾

Introducción

Lo dicho, que intentaré dejar más o menos claro el tema de la configuración del kernel, paquetes que ayudan a mejorar el funcionamiento (sobre todo de batería) y como hacer funcionar los botones principales. Yo tengo instalada Debian, que para mí es la recomendada. He probado la Yellow Dog en un par de iBooks, pero no van demasiada finas, sobre todo con los iBooks nuevos con tarjetas Radeon.

DISCLAIMERS:

- Aunque yo tengo uno con la tarjeta ATI Rage, he hecho funcionar a un par de iBooks con la Radeon y el XFree 4.2. Las configuración del kernel debería funcionar, en las X sólo cambiar la línea indicada.
- Si no sabéis como instalar una Debian, podéis mirar en [esta página](#)⁽⁷⁾, yo no lo explico.

Comencemos por lo básico, seguramente el kernel ya lo tenéis funcionando, así que comenzamos por las X....



/etc/X11/XF86Config-4

Recomiendo, sobre todo si tenéis una iBook con Radeon, que instaléis el XFree 4.2. Para tenerlo en Debian hasta que aparezca en los paquetes oficiales podéis usar los *pre* de Branden. Para ello tenéis que agregar la línea

```
deb http://people.debian.org/~branden/ sid/$ (ARCH) /
```

a vuestro `/etc/apt/sources.list`.

IMPORTANTE: acabo de [leer en xfree.org](http://www.xfree.org)⁽⁸⁾ (8/9/2002) que en las X versión 4.2.1 ya soporta a casi todos los modelos de ATI con aceleración 3D usando sólo el driver "ati". La versión 4.2.1 está disponible para Debian en las páginas de Branden.

```
# /etc/X11/XF86Config-4
Section "Files"
    FontPath        "unix/:7100"                # local font server
    # if the local font server has problems, we can fall back on these
    FontPath        "/usr/lib/X11/fonts/misc"
    FontPath        "/usr/lib/X11/fonts/cyrillic"
    # Poned siempre las :unscaled y 100dpi antes que las otras
    FontPath        "/usr/lib/X11/fonts/100dpi:unscaled"
    FontPath        "/usr/lib/X11/fonts/75dpi:unscaled"
    FontPath        "/usr/lib/X11/fonts/Type1"
    FontPath        "/usr/lib/X11/fonts/Speedo"
    FontPath        "/usr/lib/X11/fonts/100dpi"
    FontPath        "/usr/lib/X11/fonts/75dpi"
EndSection

Section "Module"
    Load            "GLcore"
    Load            "bitmap"
    Load            "dbe"
    Load            "ddc"
    Load            "dri"
    Load            "extmod"
    Load            "freetype"
    Load            "glx"
    Load            "int10"
    Load            "pex5"
    Load            "record"
    Load            "speedo"
    Load            "type1"
    Load            "vbe"
    Load            "xie"
EndSection

Section "InputDevice"
    Identifier      "Generic Keyboard"
    Driver          "keyboard"
    Option          "CoreKeyboard"
    Option          "XkbRules"        "xfree86"
    Option          "XkbModel"       "macintosh"
    Option          "XkbLayout"     "es"
EndSection

Section "InputDevice"
    Identifier      "Configured Mouse"
    Driver          "mouse"
    Option          "CorePointer"
    Option          "Device"          "/dev/input/mice"
    Option          "Protocol"        "ImPS/2"
    Option          "Emulate3Buttons" "true"
    Option          "ZAxisMapping"    "4 5"
EndSection

Section "Device"
    Identifier      "ATI"
    Driver          "ati"
    # Si teneis una radeon, descomentad la siguiente línea
    # y comentad la anterior
```



```
# pero si tenéis la 4.2.1 no hace falta, funciona con "ati"
# Driver "radeon"
# BusID "PCI:0:16:0"
# Option "UseFBDev" "true"
EndSection

Section "Monitor"
# Identifier "LCD"
# HorizSync 28-49
# VertRefresh 43-72
# Option "DPMS"
EndSection

Section "Screen"
# Identifier "Default Screen"
# Device "ATI"
# Monitor "LCD"
# DefaultDepth 16
SubSection "Display"
# Depth 15
# Modes "1024x768"
EndSubSection
SubSection "Display"
# Depth 16
# Modes "1024x768"
EndSubSection
SubSection "Display"
# Depth 24
# Modes "1024x768"
EndSubSection
EndSection

Section "ServerLayout"
# Identifier "Default Layout"
# Screen "Default Screen"
# InputDevice "Generic Keyboard"
# InputDevice "Configured Mouse"
EndSection

Section "DRI"
# Mode 0666
EndSection
```

/etc/X11/XftConfig

A continuación el fichero XftConfig, lo importante, por si usáis anti-aliasing es la línea `match edit rgba=rgb;` que es para hacer antialiasing con "subpixels".

```
# /etc/X11/XftConfig
dir "/usr/X11R6/lib/X11/fonts/Type1"
dir "/usr/X11R6/lib/X11/fonts/TrueType"

match edit rgba=rgb;
#
# alias 'fixed' for 'mono'
#
match any family == "fixed"          edit family += "mono";

#
# Check users config file
#
includeif      "~/xftconfig"

#
# Alias between XLFD families and font file family name, prefer local
# fonts
#
match any family == "charter"        edit family += "bitstream charter";
match any family == "bitstream charter" edit family += "charter";
```



Configuración del kernel

El kernel más actualizado es la versión [parcheada de Benjamin Herrenschmidt](#)⁽⁹⁾. En la página está indicado como tener actualizado el kernel con el rsync. Básicamente hay que hacer:

```
mkdir /usr/src/benh_kernel
rsync -avz rsync.penguinppc.org::linux-2.4-benh /usr/src/benh_kernel
```

Aunque es muy probable que os funcione tal como está la configuración de Benjamin, os recomiendo que **luego de hacer el rsync modifiquéis o reemplacéis el .config que deja en /usr/src/benh_kernel y por el que pongo a continuación** y hacer un `make oldconfig` para controlar las nuevas versiones que puedan aparecer. Luego ya podéis compilar el kernel tranquilamente con el `make-kpkg` si tenéis Debian (si no es así, mirad primero como se compila un kernel si no sabéis). Esta configuración del kernel ya está preparada **para hacer funcionar el Airport y un ratón USB**.

IMPORTANTE: Si tenéis una Radeon descomentad la línea del Radeon (`CONFIG_FB_RADEON=y`) y comentad la de ATY128 (`CONFIG_FB_ATY128=y`), a mí me daba problemas de símbolos si ponía a ambas como módulo. Éstas líneas están marcadas en rojo.

/usr/src/benh_kernel/.config

```
# /usr/src/benh_kernel/.config
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_HAVE_DEC_LOCK=y

#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y

#
# Loadable module support
#
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y

#
# Platform support
#
CONFIG_PPC=y
CONFIG_PPC32=y
CONFIG_6xx=y
CONFIG_PPC_STD_MMU=y
CONFIG_ALL_PPC=y
#
# General setup
#
CONFIG_PCI=y
CONFIG_NET=y
CONFIG_SYSCTL=y
CONFIG_SYSVIPC=y
CONFIG_KCORE_ELF=y
CONFIG_BINFMT_ELF=y
CONFIG_KERNEL_ELF=y
CONFIG_PCI_NAMES=y
CONFIG_HOTPLUG=y

#
# Parallel port support
#
CONFIG_PPC_RTC=y
CONFIG_PROC_DEVICETREE=y
```



```
CONFIG_PPC_RTAS=y
CONFIG_BOOTX_TEXT=y

# Block devices
#
CONFIG_BLK_DEV_LOOP=y

#
# Networking options
#
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
CONFIG_FILTER=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_SYN_COOKIES=y

#
# ATA/IDE/MFM/RLL support
#
CONFIG_IDE=y

#
# IDE, ATA and ATAPI Block devices
#
CONFIG_BLK_DEV_IDE=y

CONFIG_BLK_DEV_IDEDISK=y
CONFIG_BLK_DEV_IDECD=y

#
# IDE chipset support/bugfixes
#
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_IDEPCI_SHARE_IRQ=y
CONFIG_BLK_DEV_IDEDMA_PCI=y
CONFIG_IDEDMA_PCI_AUTO=y
CONFIG_BLK_DEV_IDEDMA=y
CONFIG_BLK_DEV_ADMA=y
CONFIG_BLK_DEV_IDE_PMAC=y
CONFIG_BLK_DEV_IDEDMA_PMAC=y
CONFIG_BLK_DEV_IDEDMA_PMAC_AUTO=y
CONFIG_BLK_DEV_IDEDMA=y
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_IDEDMA_AUTO=y
CONFIG_BLK_DEV_IDE_MODES=y

#
# Network device support
#
CONFIG_NETDEVICES=y

#
# ARCnet devices
#
CONFIG_TUN=m

#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
CONFIG_BMAC=m
CONFIG_GMAC=m

#
# Wireless LAN (non-hamradio)
#
CONFIG_NET_RADIO=y
CONFIG_HERMES=m
CONFIG_APPLE_AIRPORT=m
```



```
CONFIG_NET_WIRELESS=y

#
# Console drivers
#
# CONFIG_VGA_CONSOLE is not set

#
# Frame-buffer support
#
CONFIG_FB=y
CONFIG_DUMMY_CONSOLE=y
CONFIG_FB_OF=y

# CONFIG_FB_RADEON=y
CONFIG_FB_ATY128=y

CONFIG_FBCON_CFB8=y
CONFIG_FBCON_CFB16=y
CONFIG_FBCON_CFB24=y
CONFIG_FBCON_CFB32=y
CONFIG_FBCON_FONTS=y
CONFIG_FONT_8x16=y
CONFIG_FONT_SUN8x16=y
CONFIG_FONT_SUN12x22=y

#
# Input core support
#
CONFIG_INPUT=y
CONFIG_INPUT_KEYBDEV=y
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
CONFIG_INPUT_EVDEV=y

#
# Macintosh device drivers
#
CONFIG_ADB_PMU=y
CONFIG_PMAC_PBOOK=y
CONFIG_PM=y
CONFIG_PMAC_APM_EMU=y
CONFIG_PMAC_BACKLIGHT=y
CONFIG_ADB=y
CONFIG_INPUT_ADBHID=y
CONFIG_MAC_EMUMOUSEBTN=y
CONFIG_MAC_HID=y

#
# Character devices
#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_SERIAL=m
CONFIG_UNIX98_PTYS=y
CONFIG_UNIX98_PTY_COUNT=256

#
# I2C support
#
CONFIG_I2C=y
CONFIG_I2C_KEYWEST=y
CONFIG_I2C_CHARDEV=y
CONFIG_I2C_PROC=y

#
# Mice
#
CONFIG_BUSMOUSE=y
```



```
CONFIG_MOUSE=y
CONFIG_PSMOUSE=y

#
# Ftape, the floppy tape device driver
#
CONFIG_AGP=y
CONFIG_AGP_UNINORTH=y
CONFIG_DRM=y

#
# DRM 4.1 drivers
#
CONFIG_DRM_NEW=y
CONFIG_DRM_R128=m
# CONFIG_DRM_RADEON=y

#
# File systems
#
CONFIG_AUTOFS4_FS=y
CONFIG_HFS_FS=m
CONFIG_EXT3_FS=y
CONFIG_JBD=y
CONFIG_TMPFS=y
CONFIG_RAMFS=y
CONFIG_ISO9660_FS=y
CONFIG_PROC_FS=y
CONFIG_DEVPTS_FS=y
CONFIG_EXT2_FS=y
CONFIG_UDF_FS=m

#
# Network File Systems
#
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_NFSD=y
CONFIG_NFSD_V3=y
CONFIG_SUNRPC=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
CONFIG_SMB_FS=m

#
# Partition Types
#
CONFIG_PARTITION_ADVANCED=y
CONFIG_MAC_PARTITION=y
CONFIG_MSDOS_PARTITION=y
CONFIG_SMB_NLS=y
CONFIG_NLS=y

#
# Native Language Support
#
CONFIG_NLS_DEFAULT="iso8859-15"
CONFIG_NLS_CODEPAGE_850=m
CONFIG_NLS_ISO8859_1=m
CONFIG_NLS_ISO8859_15=y
CONFIG_NLS_UTF8=m

#
# Sound
#
CONFIG_SOUND=y
CONFIG_DMASOUND_PMAC=y
CONFIG_DMASOUND=y
CONFIG_I2C=y
```



```

CONFIG_I2C_KEYWEST=y

#
# USB support
#
CONFIG_USB=y

#
# Miscellaneous USB options
#
CONFIG_USB_DEVICEFS=y
#
# USB Host Controller Drivers
#
CONFIG_USB_OHCI=y

CONFIG_USB_PRINTER=m

#
# USB Human Interface Devices (HID)
#
CONFIG_USB_HID=y
CONFIG_USB_HIDINPUT=y

#
# USB Serial Converter support
#
CONFIG_USB_SERIAL=m
CONFIG_USB_SERIAL_VISOR=m
CONFIG_USB_SERIAL_IPAQ=m

#
# Kernel hacking
#
CONFIG_MAGIC_SYSRQ=y

```

Configuración para emulación de 3 botones y las modutils

Ahora pondremos en el `sysctl.conf` las instrucciones para poder emular los tres botones del ratón. De la forma en que lo indico aquí, el **botón del medio será el F11**, y el **botón derecho será el botón que hay cerca del *touchpad*, a la derecha, entre la manzanita y la flecha izquierda**, esa que parece un v invertida con una rayita horizontal (o una k acostada :-).

NOTA: como veremos más adelante, se usará la F12 para expulsar el CD, como hace el Mac OS X.

/etc/sysctl.conf

```

# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information.
#
# Con esto simlaremos los tres botones
# del ratón
dev/mac_hid/mouse_button3_keycode = 96
dev/mac_hid/mouse_button2_keycode = 87
dev/mac_hid/mouse_button_emulation = 1

```

Aunque muchas de las líneas de los ficheros indicados a continuación ya están incluidas por la instalación, los pongo para que comparéis. Luego hay que generar el `modules.conf`, **así que no os olvidéis de ejecutar el comando `update-modules` luego de modificar los ficheros.**



/etc/modutils/network

```
alias eth1 airport
probeall airport hermes orinoco
alias eth0 gmac
```

/etc/modutils/sound

```
alias sound-slot-0 dmasound_pmac
alias sound-slot-1 dmasound_pmac
alias sound-service-0-0 dmasound_pmac
alias sound-service-0-3 dmasound_pmac
alias sound-service-1-0 dmasound_pmac
```

/etc/modutils/apm

```
alias char-major-10-134      apm
alias /dev/apm_bios          /dev/misc/apm_bios
alias /dev/misc/apm_bios    apm
```

Control de energía

pmud

El control de energía para un portátil es vital, por suerte en el iBook funciona perfectamente, inclusive para el *sleep* al cerrar la pantalla o como veremos más adelante, para regular la intensidad de la luz trasera de la pantalla.

El paquete imprescindible para el control de energía es el **pmud**, así que lo primero que habrá que hacer es el **apt-get install pmud**. Este paquete hace varias cosas, como habilitar o deshabilitar dispositivos cuando cambia el estado de la batería o la alimentación y de poner y despertar al iBook del *sleep*.

Para rebajar el consumo de batería casi al máximo, lo primero que hay que hacer es no poner el sistema de ficheros Ext3 o ReiserFS, sino el Ext2. Tanto el Ext3 como el ReiserFS escriben frecuentemente a disco, haciendo que el disco nunca se ponga en reposo.

Además yo modifiqué los parámetros del pmud para forzar a que el disco pare antes y cambiar la potencia de la luz de la pantalla dependiendo si está alimentado por batería o enchufado a la red. **Para ello basta con generar el fichero [/etc/power/pwrctl-local](#)** con las siguientes líneas:

/etc/power/pwrctl-local

```
# pwrctl-local de ricardo galli
#!/bin/bash
logger=/usr/bin/logger

function do_warn()
{
    local msg="Low battery, system will go down..."

    (
        /usr/X11R6/bin/xmessage -center -timeout 15 "$msg" || \
        echo "$msg" | /usr/bin/wall
    ) &
}

function pwrctl_Core99()
{
    case "$1" in
        minimum)
            # min power, set disk to spin down after 1 minute
            [ -f /sbin/hdparm ] && hdparm -p -S 12 /dev/hda
            case "$2" in
```



```

        ac)
            fblevel 12
        ;;
        *)
            fblevel 8
        ;;
    esac
;;
medium)
    # medium power, set disk to spin down after 2.5 minutes
    [ -f /sbin/hdparm ] && hdparm -p -S 20 /dev/hda
    case "$2" in
        ac)
            fblevel 15
        ;;
        *)
            fblevel 8
        ;;
    esac
;;
maximum)
    case "$2" in
        ac)
            # on mains, do not spin down
            [ -f /sbin/hdparm ] && hdparm -p -S 50 /dev/hda
            fblevel 15
        ;;
        *)
            # on battery, set disk to spin down after 5 minutes
            [ -f /sbin/hdparm ] && hdparm -p -S 25 /dev/hda
            fblevel 8
        ;;
    esac
;;
warning)
    do_warn
;;
lid-closed)
    fblevel off
;;
lid-opened)
;;
sleep)
;;
wakeup)
    $logger -p daemon.debug -t pwrctl "$0: Wake-up $1 $2"
    case "$2" in
        ac)
            fblevel 15
        ;;
        *)
            fblevel 8
        ;;
    esac
;;
    *)
        $logger -p daemon.error -t pwrctl "$0: invalid arg $1"
    ;;
    esac
}

pwrctl_Core99 $1 $2
exit 1

```

xpmumon

Para poder monitorizar el estado de la batería, ver el consumo o carga actual y el tiempo restante de batería, lo mejor para mí es el programa **xpmumon**, que está en el paquete del mismo nombre.



xpmumon

Hacer funcionar los botones de Apple

Control de la luz de la pantalla

El kernel ya reconoce directamente a los botones F1 y F2 que cambian el brillo de la luz de la pantalla, así que no hace falta nada para ello.

Control de volumen y expulsión del CD

Encontré que el paquete **pbbuttosd** funciona muy bien (es parecido al `ikeysd`) y encarga de interpretar a los botones de control de volumen, *mute* y al F12 para la expulsión del CD.

powerprefs

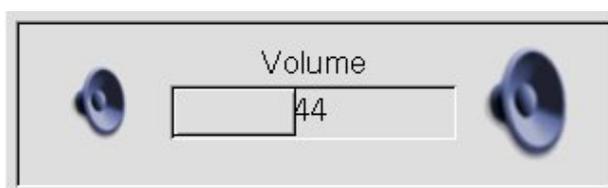
Opcionalmente, si no queréis modificar a mano el fichero de configuración del `pbbuttosd` (`/etc/pbbuttosd.conf`), podéis instalar el paquete `powerprefs` que os permite configurarlo gráficamente.



powerprefs

gtkpbbuttons

Además, aunque a mi no me gusta, podéis instalar el paquete **gtkpbbuttons** para que cuando presionéis los botones de control de volumen o brillo de pantalla os salgan unos gráficos en la pantalla, que aunque no tan bonitos como los del Mac OS X, están razonablemente bien.



gtkpbbuttons



Bueno, nada más por ahora, creo que con esto ya podéis dejar a vuestro portátil en muy buen estado. Y si hay errores y *typos*, perdonad, es muy tarde y estoy cansado para hacer una corrección exhaustiva. Mañana será...

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=1481&nIdPage=2>
 2. <http://bulma.net/body.phtml?nIdNoticia=1481>
 3. <http://bulma.net/body.phtml?nIdNoticia=1481&nIdPage=3>
 4. <http://bulma.net/body.phtml?nIdNoticia=1481&nIdPage=4>
 5. <http://bulma.net/body.phtml?nIdNoticia=1481&nIdPage=5>
 6. <http://bulma.net/body.phtml?nIdNoticia=1481&nIdPage=6>
 7. <http://people.debian.org/~branden/ibook.html>
 8. <http://www.xfree.org/current/Status6.html#6>
 9. <http://penguinppc.org/~benh/>
-

E-mail del autor: gallir_ARROBA_uib.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1481>