

ecoRaee – B4

T0.1.2.1 – Creación imagen comprobación hw

Identificador del documento:	T0.1.2.1
Fecha:	21/10/2013
Actividad	T0.1.2.1
Estado del documento:	Pendiente de revisión
Enlace del documento:	

Abstract:

En el presente documento se explica el procedimiento seguido para la personalización de un sistema live de Ubuntu, que incluye la ejecución de un script propio para la obtención del listado de los componentes hardware del equipo en que se ejecute. Dicho listado se guarda en un fichero xml y es copiado mediante ssh a un repositorio creado a tal efecto.

Datos de la entrega

	Nombre	Compañía / Actividad	Fecha	Firma
Autor	Roberto Rosende Dopazo	T0.1.2.1	21/10/2013	
Verificado por				
Revisado por				
Aprobado por				

Log del documento

Version	Fecha	Comentario	Autor
V1.0	21/10/2013	Versión inicial del documento	Roberto Rosende Dopazo

Registro de cambios del documento

Versión	Item	Motivo del cambio

Tabla de contenido

1.- Resumen de pasos seguidos para la elaboración del cd autoarrancable.	4
2.- Acceso chroot al nuevo sistema.....	4
3.- Personalizando el nuevo sistema.....	5
4.- Creación script personalizado.....	5
5.- Creación de la imagen iso autoarrancable.....	7

1.- Resumen de pasos seguidos para la elaboración del cd autoarrancable.

- Descargar imagen iso de Ubuntu 12.04.3 versión desktop.
- Montar imagen y accediendo mediante un entorno chroot aligerar paquetes instalados.
- Crear script que automatiza el proceso de lectura del listado de hardware y copia el resultado via ssh a un servidor dado.
- Crear imagen iso con el nuevo sistema creado en los pasos anteriores.

2.- Acceso chroot al nuevo sistema

Descargar la iso de Ubuntu de <http://www.ubuntu.com>

Montar la imagen iso en un directorio temporal:

```
$ sudo mount -o loop ubuntu-12.04.3-desktop-i386.iso ~/tmplive/mnt/  
mount: /dev/loop0 is write-protected, mounting read-only
```

Crear un directorio en el que se extraigan los ficheros de la iso excepto el sistema de ficheros squash

```
$ cd tmplive  
$ mkdir ~/livecd  
$ mkdir ~/livecd/cd  
$ rsync --exclude=/casper/filesystem.squashfs -a ~/livecd/ livecd/cd
```

Crear el sistema de ficheros squash para luego poder hacer chroot al sistema que estamos configurando:

```
$ mkdir ~/livecd/squashfs  
$ mkdir ~/livecd/custom  
$ sudo modprobe squashfs  
$ sudo mount -t squashfs -o loop /tmp/livecd/casper/filesystem.squashfs ~/livecd/squashfs/  
$ sudo cp -a ~/livecd/squashfs/* ~/livecd/custom
```

Hacemos chroot y montamos los sistemas de ficheros especiales: proc, sysfs y devpts:

```
$ sudo mount --bind /dev/ edit/dev  
$ sudo chroot edit  
# mount -t proc none /proc  
# mount -t sysfs none /sys  
# mount -t devpts none /dev/pts
```

Exportamos las variables home y lc_all para evitar problemas en el nuevo sistema:

```
# export HOME=/root
# export LC_ALL=C
```

Con esto tenemos acceso completo de root al nuevo sistema, configuramos la red si es necesario usando el comando dhclient para casos de dhcp o configurando de forma estática según nuestro entorno.

3.- Personalizando el nuevo sistema

Puesto que el sistema completo de ubuntu no es necesario para el proposito de este cd autoarrancable, personalizamos el entorno para desinstalar el software prescindible. Para ello hacemos uso del gestor de paquetes apt.

```
# apt-get remove --purge gnome-games*
# dpkg-query -W --showformat='${Package}\n'
# apt-get remove --purge gnome-games*
# apt-get remove --purge telepathy* rhythmbox* printer-driver-*
# apt-get autoremove
# dpkg-query -W --showformat='${Package}\n' | grep -v lib
# apt-get remove --purge totem* gstreamer* evince* bluez* cups* firefox*
# dpkg -l 'linux-*' | sed '/^ii/!d;/'"$(uname -r | sed "s/^(.*)-([0-9])\
+ \)/\1/")"/d;s/^[^ ]* [^ ]* \([^\ ]*\).*/\1/;/[0-9]/!d' | xargs sudo apt-get -y purge
# apt-get autoremove
# apt-get clean
# apt-get autoremove
```

Una vez que terminamos de desintalar lo que no nos interesa limpiamos posibles ficheros temporales y sobrantes del sistema.

```
# rm -rf /tmp/*
# rm -rf /etc/hosts
# rm -rf /etc/resolv.conf
```

4.- Creación script personalizado.

El script que ejecuta el comando lshw generando el fichero xml y copiandolo al servidor destinado como repositorio de estos ficheros es el siguiente:

```
root@localhost:~# cat checkHardware.sh
#!/bin/bash

##USER, SERVER and DIR are set to run scp
USER="lector"
```

```

SERVER="192.168.40.101"
DIR="/home/lector/lshw_xml_files/"
##FILENAME is created with product name get from lshw and time in Unix format
FILENAME="`lshw | grep -m1 product: | sed -e 's/[ ]\{1,\}product: //g'`_`date +%s`.xml"
echo "Nombre del fichero: "$FILENAME
lshw -xml > "`echo $FILENAME`"
scp "$FILENAME" "$USER@$SERVER:$DIR"

if [ $? -eq 0 ] ; then
    echo "Fichero copiado con EXITO.";
else
    echo "ERROR al copiar el fichero.";
fi

```

Para que el fichero funcione de forma automática sin pedir claves al hacer la copia del xml, debemo copiar la clave pública del sistema al equipo y usuario que hacen de repositorio de los ficheros xml.

```

root@localhost:~# ssh-copy-id lector@192.168.40.101
The authenticity of host '192.168.40.101 (192.168.40.101)' can't be established.
ECDSA key fingerprint is b1:25:51:65:98:f9:b7:c5:4c:cd:6e:65:36:fd:3b:34.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.40.101' (ECDSA) to the list of known hosts.
lector@192.168.40.101's password:
Now try logging into the machine, with "ssh 'lector@192.168.40.101'", and check in:

~/ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.

```

Comprobamos que podemos conectarnos sin password a dicho equipo.

```

root@localhost:~# ssh lector@192.168.40.101
Linux repolshw 3.2.0-4-amd64 #1 SMP Debian 3.2.51-1 x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lector@repolshw:~$

```

Comprobamos que el script funciona correctamente y que el fichero xml se copia en su destino:

```
root@localhost:~# ./checkHardware.sh
Nombre del fichero: HP Pavilion dv6500 Notebook PC (GX991LA#ABM)_1382349369.xml
HP Pavilion dv6500 Notebook PC (GX991LA#ABM)_1382349369.xml
100% 40KB 40.2KB/s 00:00
Fichero copiado con EXITO.
root@localhost:~# ssh lector@192.168.40.101 "ls -l /home/lector/lshw_xml_files/"
total 44
-rw-r--r-- 1 lector lector 41157 Oct 21 11:56 HP Pavilion dv6500 Notebook PC
(GX991LA#ABM)_1382349369.xml
```

Para que el script se ejecute nada más cargado el sistema y su configuración básica, añadimos una llamada al mismo al fichero rc.local. Con esto evitaremos tener que esperar incluso a que el sistema cargue de todo ya que veremos la salida del script por pantalla en el arranque, pudiendo desconectar el equipo en cuanto el fichero se haya copiado.

```
root@localhost:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/root/checkHardware.sh
exit 0
```

5.- Creación de la imagen iso autoarrancable

Llegados a este punto ya debemos desmontar los sistemas de fichero especiales, salir del entorno chroot y generar la imagen iso que podremos después volcar a un cd para su ejecución.

Desmontamos los sistemas de ficheros especiales que montamos en pasos anteriores:

```
# umount /proc || umount -lf /proc
# umount /sys || umount -lf /sys
# umount /dev/pts/
```

Salimos del entorno chroot

```
# exit
```

Antes de crear la imagen iso debemos volver a crear el fichero de manifiesto del sistema y regenerar el sistema de ficheros squash.

Para el fichero de manifiesto:

```
$ sudo chmod +w extract-cd/casper/filesystem.manifest
$ sudo chroot edit/ dpkg-query -W --showformat='${Package} ${Version}\n' > extract-cd/casper/filesystem.manifest
```

Para regenerar el sistema de fichero squash:

```
# rm extract-cd/casper/filesystem.squashfs
# mksquashfs edit extract-cd/casper/filesystem.squashfs
# printf $(sudo du -sx --block-size=1 edit | cut -f1) > extract-cd/casper/filesystem.size
```

Le damos el nombre que queramos a la imagen personalizada:

```
# cat extract-cd/README.diskdefines
#define DISKNAME Ubuntu 12.04.3 LTS "Hardware Checker v1.0" - Release i386
#define TYPE binary
#define TYPEbinary 1
#define ARCH i386
#define ARCHi386 1
#define DISKNUM 1
#define DISKNUM1 1
#define TOTALNUM 0
#define TOTALNUM0 1
```

Borramos la suma md5 y calculamos la correspondiente al nuevo sistema:

```
$ cd extract-cd
$ sudo rm md5sum.txt
$ find -type f -print0 | sudo xargs -0 md5sum | grep -v isolinux/boot.cat | sudo tee md5sum.txt
```

Por último creamos la imagen iso del sistema

```
# mkisofs -D -r -V "$IMAGE_NAME" -cache-inodes -J -l -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o ../ubuntu-12.04.3-desktop-i386-hardcheck_v1.0.iso .
```

Aunque la imagen está pensada para ser volcada a cd y ejecutada desde ahí, podemos pasarla a un pendrive usando un software tipo Unetbootin y también funcionará correctamente.