

Imaging

1 Introducción

El objeto de este proyecto es tomar directorios completos y producir imágenes arrancables. Una imagen es un fichero que se puede copiar, bit a bit, en un dispositivo, ya sea un CD (imagen tipo CD), ya sea un disco duro, una memoria SD, o una llave de memoria USB (imagen tipo HD). Contiene particiones, dentro de ellas sistemas de ficheros, y dentro de ellos directorios y ficheros. Los arrancables contienen un MBR (*Master Boot Record*), un sistema de arranque, en este caso SysLinux, y los ficheros y directorios necesarios para hacer funcionar una computadora.

Imaging mantiene, a partir de su directorio raíz `Imaging`, los subdirectorios:

- `input`, que contiene los *scripts*,
- `downloads`, que mantiene los paquetes descargados,
- `CDroot`, en donde se crea la raíz,
- `output`, donde se almacenan las imágenes generadas, y
- `doc`, en donde se mantiene la documentación (este documento).

2 Ficheros.

2.1 Imaginador

El principal. El primer parámetro, `$1`, sirve para indicar el tipo de imagen que se quiere generar.

Nuestro imaginador puede producir los siguientes tipos de imágenes:

- `iso` Imagen tipo CD. La típica, y que está de acuerdo a la norma ISO 9660, pero con extensiones para Windows y Linux. La norma original sólo acepta nombres 8.3.
- `hiso` Imagen tipo CD hibridada, para hacerla compatible con discos duros. No todas las BIOS pueden arrancarla, pero para las que lo hacen, como la de mi AA1, es la imagen ideal, ya que vale sea cual sea el dispositivo de arranque al que se grabe.
- `sys` Imagen tipo HD. La partición 1 contiene un sistema de ficheros `vfat`.
- `zip` Imagen tipo HD. La partición 4 contiene un sistema de ficheros `vfat`. Esta disposición es la única que funciona en algunos sistemas antiguos, como mi eBox.
- `ext` Imagen tipo HD. La partición 1 contiene un sistema de ficheros `ext2`.

Después del primer parámetro pueden indicarse cuantos directorios raíz se quieran. Estos directorios raíz, llamados convencionalmente `CDroot`, han de tener dos subdirectorios: `boot` con los ficheros de configuración de Syslinux, y otro con su propio nombre para el resto de los ficheros.

2.2 Comprobaciones iniciales

Para empezar se fijan las versiones, se define una función de uso general y se comprueban un par de asuntos.

```
../input/imaging.sh
```

```
1 #!/bin/bash
2
3 SLV="3.86"
4 #SLV="3.82"
5 #SLV="3.72" ## Doesn't work
6 #SLV="3.63" ## Works
7 #SLV="3.71" ## Works
8 SYSLINUX="syslinux-$SLV"
9 MTV="4.00"
10 #MTV="2.11"
11 MEMTEST="memtest86+-$MTV"
12
13 boldecho() {
14   echo -e "\033[1m$\033[0m"
15 }
16
17 if test $(whoami) != 'root'
18 then
19   echo "Sorry! You are $(whoami), and only the root can run this script"
20   exit 1
21 fi
22
23 if test "$1" == ""
24 then
```

```
25 boldecho "Usage: $0 fs [CDroot ...]"
26 echo "makes a bootable disk file image"
27 echo "formatted with one big fs partition"
28 echo "where fs = iso | hiso | sys | zip | ext"
29 echo "  iso is iso9660 (CD)"
30 echo "  hiso is an hybridated iso"
31 echo "  sys is vfat (Windows)"
32 echo "  zip is vfat as zip (partition 4)"
33 echo "  ext is ext2 (Linux)"
34 echo "  no is vfat but no partitions (as a floppy) NOT IMPLEMENTED"
35 echo "the created file will be called SD.iso, or SD.img"
36 exit 2
37 else
38   FS="$1"
39   shift
40 fi
41
42 ###
```

2.3 Directorios

Se gestionan los directorios

../input/imaging.sh

```
43 #DIRS
44
45 # ¿Usó $SDIR?
46 SDIR=$(dirname $(readlink -f $0))
47
48 if [ -d /mnt/SATA-Projects/Imaging ]
49 then
50   CDIR="/mnt/SATA-Projects"
51   RDIR="/mnt/SATA-Projects/Imaging"
52 else
53   if [ ! -d /tmp/Imaging ] ; then mkdir /tmp/Imaging ; fi
54   CDIR="/tmp"
55   RDIR="/tmp/Imaging"
56 fi
57
58 cd ${RDIR}
59 if test ! -d downloads ; then mkdir downloads ; fi
60 if test ! -d output ; then mkdir output ; fi
61 if test ! -d working ; then mkdir working ; fi
62 if test -d CDroot ; then
63   rm -r CDroot
64   mkdir -p CDroot/boot
65 else
66   mkdir -p CDroot/boot
67 fi
68
69 ###
```

2.4 Descargas

Se descargan los paquetes necesarios: Syslinux y Memtest.

```
../input/imaging.sh
```

```
70 #WGET
71 boldecho "$SYSLINUX"
72 cd ${RDIR}/downloads
73 if test ! -f $SYSLINUX.tar.bz2
74 then
75     echo "    Getting Syslinux"
76     wget http://www.kernel.org/pub/linux/utils/boot/syslinux/$SYSLINUX.tar.bz2
77     cd ${RDIR}/working
78     echo "    Extracting Syslinux"
79     tar xjf ${RDIR}/downloads/$SYSLINUX.tar.bz2
80     chown -R root:root $SYSLINUX # 1026:1026
81 fi
82
83 boldecho "$MEMTEST"
84 cd ${RDIR}/downloads
85 if test ! -f "$MEMTEST.bin.gz"
86 then
87     echo "    Getting Memtest"
88     wget http://www.memtest.org/download/$MTV/$MEMTEST.bin.gz
89     cd ${RDIR}/working
90     echo "    Extracting Memtest"
91     gzip -cd "${RDIR}/downloads/$MEMTEST.bin.gz" > memtest
92 fi
93
94 ###
```

2.5 Configuración

Se crean los ficheros de configuración del arranque.

```
../input/imaging.sh
```

```
95 #CONF
96 boldecho "Writing configuration file"
97
98 ##BEGIN CONFIG##
99
100 #BEGIN HEADER#
101 #Don't move this part
102 cat > /tmp/syslinux.cfg << "EOF"
103 DEFAULT menu.c32
104 PROMPT 0
105 menu title Welcome to Linux!
106 EOF
107
108 for D in $@ ; do
109     cd ${CDIR}
110     cd $D/boot
111     for F in *.hdr ; do
112         cat >> /tmp/syslinux.cfg << EOF
113         INCLUDE $F
114         EOF
115     done
116 done
117
```

```
118 #END HEADER#
119
120 #BEGIN HD BOOT OPTION#
121 #If this bugs you, get rid of it.
122 cat >> /tmp/syslinux.cfg << "EOF"
123 menu separator
124 label local
125 menu label Boot from ^hard drive
126 localboot 0x80
127 menu separator
128 EOF
129 #END HD BOOT OPTION#
130
131 # HERE YOU CAN ADD YOUR OWN ENTRIES
132 for D in $@ ; do
133 cd ${CDIR}
134 cd $D/boot
135 for F in *.cfg ; do
136 cat >> /tmp/syslinux.cfg << EOF
137 INCLUDE $F
138 EOF
139 done
140 done
141
142 #BEGIN MEMTEST ENTRY#
143 cat >> /tmp/syslinux.cfg << "EOF"
144 menu separator
145 label memtest
146 menu label ^Memtest86+
147 kernel /boot/memtest
148 EOF
149 #END MEMTEST ENTRY#
150
151 #BEGIN REBOOT ENTRY#
152 cat >> /tmp/syslinux.cfg << "EOF"
153 menu separator
154 label reboot
155 menu label ^reboot
156 com32 reboot.c32
157 EOF
158 #END REBOOT ENTRY#
159
160 ##END CONFIG##
161
162 ###
```

2.6 ISO

Se crean las imágenes para CD, tipo iso.

```
../input/imaging.sh
```

```
163 #ISO
164 cd ${RDIR}
165 if test ${FS} == "iso" -o ${FS} == "hiso"
166 then
```

```

167 boldecho "Creating ISO image"
168 cd ${RDIR}/working
169 mkdir ${RDIR}/CDroot/boot/isolinux/
170 cp ${SYS LINUX}/core/isolinux.bin ${RDIR}/CDroot/boot/isolinux/
171 cp ${SYS LINUX}/com32/modules/reboot.c32 ${RDIR}/CDroot/boot/isolinux/
172 cp ${SYS LINUX}/com32/menu/menu.c32 ${RDIR}/CDroot/boot/isolinux/
173 cp memtest ${RDIR}/CDroot/boot/
174 cd ${CDIR}
175 for D in $@ ; do
176   cp -a $D/boot/* ${RDIR}/CDroot/boot/isolinux/
177 done
178 cp /tmp/syslinux.cfg ${RDIR}/CDroot/boot/isolinux/isolinux.cfg
179 cd ${CDIR}
180 genisoimage -r -J -l -V "ISOboot" -input-charset iso8859-1 \
181   -b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat \
182   -no-emul-boot -boot-load-size 4 -boot-info-table \
183   -o ${RDIR}/output/SD.iso ${RDIR}/CDroot $@
184 echo "ISO image generated: ${RDIR}/output/SD.iso"
185 if test ${FS} == "hiso"
186 then
187   echo "Hybridizing the iso"
188   ${RDIR}/working/${SYS LINUX}/utils/isohybrid ${RDIR}/output/SD.iso
189 fi
190 exit
191 fi
192
193 ###

```

2.7 IMG

Se crean las imágenes para Disco duro o memoria USB, tipo img.

../input/imaging.sh

```

194 #IMG
195 FSCode="c" ; if test ${FS} == "ext" ; then FSCode="83" ; fi
196 PART="1" ; if test ${FS} == "zip" ; then PART="4" ; fi
197
198 boldecho "Creating HD image"
199 cat > /tmp/fdisk.cmd << EOF
200 o
201 n
202 p
203 $PART
204
205
206 t
207 $FSCode
208 a
209 $PART
210 p
211 w
212 EOF
213
214
215 cd ${RDIR}/output

```

```
216 if test -f SD.img ; then rm SD.img ; fi
217 dd if=/dev/zero of=SD.img bs=512 count=503808
218 losetup /dev/loop0 SD.img
219 fdisk -b 512 -H 16 -S 32 -C 984 /dev/loop0 < /tmp/fdisk.cmd
220 dd if=${RDIR}/working/$SYSLINUX/mbr/mbr.bin of=/dev/loop0
221 modprobe dm-mod
222 kpartx -av /dev/loop0
223
224 if test ${FS} == "ext" ; then
225   mkfs.ext2 -L Linuxboot /dev/mapper/loop0p1
226   mount /dev/mapper/loop0p1 /mnt/loop/
227   mkdir -p /mnt/loop/boot/extlinux/
228   cd ${CDIR}
229   for D in $@ ; do
230     cp -a $D/* /mnt/loop
231     cp -a $D/boot/* /mnt/loop/boot/extlinux/
232   done
233   cd ${RDIR}/working
234   cp $SYSLINUX/com32/menu/menu.c32 /mnt/loop/boot/extlinux/
235   cp $SYSLINUX/com32/modules/reboot.c32 /mnt/loop/boot/extlinux/
236   cp memtest /mnt/loop/boot/
237   cp /tmp/syslinux.cfg /mnt/loop/boot/extlinux/extlinux.conf
238   extlinux/extlinux -i /mnt/loop/boot/extlinux/
239   umount /mnt/loop
240   kpartx -d /dev/loop0
241   losetup -d /dev/loop0
242 else
243   mkfs.msdos -v -n DOSboot /dev/mapper/loop0p$PART
244   mount /dev/mapper/loop0p$PART /mnt/loop/
245   mkdir -p /mnt/loop/boot/syslinux/
246   cd ${CDIR}
247   for D in $@ ; do
248     cp -a $D/* /mnt/loop
249     cp -a $D/boot/* /mnt/loop/boot/syslinux/
250   done
251   cd ${RDIR}/working
252   cp $SYSLINUX/com32/menu/menu.c32 /mnt/loop/boot/syslinux/
253   cp $SYSLINUX/com32/modules/reboot.c32 /mnt/loop/boot/syslinux/
254   cp memtest /mnt/loop/boot/
255   cp /tmp/syslinux.cfg /mnt/loop/boot/syslinux/
256   umount /mnt/loop
257   kpartx -d /dev/loop0
258   losetup -d /dev/loop0
259   cd ${RDIR}
260   working/$SYSLINUX/linux/syslinux -o 16384 output/SD.img
261 fi
262
263 echo "SD image generated: ${RDIR}/output/SD.img"
264 exit
265 ###
```

2.8 Grabador

Sirve para recordar cómo grabar una tarjeta SD con un fichero imagen. Incluye medidas para evitar algunos accidentes.

../input/burn.sh

```
1 #!/bin/bash
2
3 if test $(whoami) != 'root'
4 then
5   echo "Sorry! You are $(whoami), and only the root can run this script"
6   exit 1
7 fi
8
9 boldecho() {
10  echo -e "\033[1m$\033[0m"
11 }
12
13 if test "$1" == ""
14 then
15   DEV="/dev/sdb"
16 else
17   DEV="$1"
18 fi
19
20 if test $(mount | grep -c "$DEV") != 0
21 then
22   boldecho "mount | grep \"$DEV\""
23   mount | grep "$DEV"
24   echo
25   echo "This would destroy all data in $DEV."
26   echo "But I will not do that for you, sorry."
27   echo "If you know what you are doing, you can try:"
28   boldecho "dd if=../output/SD.img of=$DEV bs=4096 conv=notrunc,noerror"
29   echo "on your own responsibility."
30   echo "You were severely advised!"
31   exit 2
32 fi
33
34 boldecho "dd if=../output/SD.img of=$DEV bs=4096 conv=notrunc,noerror"
35 dd if=../output/SD.img of=$DEV bs=4096 conv=notrunc,noerror
36 sync
37
```

2.9 Emuladores

Sirven para recordar cómo llamar a `qemu` y arrancar la imagen. Usan la extensión del fichero para determinar el tipo de imagen, `img` o `iso`, utilizada.

../input/run.sh

```
1 #!/bin/bash
2
3 if [ "$1" == "" ]
4 then
5   IMG="SD.img"
6 else
7   IMG="$1"
8 fi
9
10
```



```

11 if [ ! -f $IMG ]
12 then
13   if [ -d /mnt/SATA-Projects/Imaging ]
14   then
15     RDIR="/mnt/SATA-Projects/Imaging/output"
16   else
17     RDIR="/tmp/Imaging/output"
18   fi
19   if [ -f $RDIR/$IMG ]
20   then
21     IMG="$RDIR/$IMG"
22   else
23     echo "File $IMG not found!"
24     exit 1
25   fi
26 fi
27
28
29 BN=$(basename $IMG)
30
31 EXT=${BN##*.}
32 if [ "$EXT" == "$BN" ] ; then EXT="" ; fi
33
34 #echo Basename: "$BN"
35 #echo Extension: "$EXT"
36 #exit
37
38 if [ "$EXT" == "iso" ]
39 then
40   echo "qemu -cdrom $IMG -boot d"
41   qemu -cdrom $IMG -boot d
42 else
43   if [ "$EXT" == "img" ]
44   then
45     echo "qemu -hda $IMG -boot c"
46     qemu -hda $IMG -boot c
47   else
48     echo "ERROR! I don't know how to treat this kind of image: \"$EXT\"."
49   fi
50 fi
51

```

2.10 Versiones

Para crear una versión mínima, que contenga únicamente SciTeX, una vez que se ha ejecutado `imaging.sh` y el `create.sh` de SciTeX, hay que ejecutar `nameSciTeX.sh`.

../input/makeSciTeX.sh

```

1 #!/bin/bash
2
3 if test $(whoami) != 'root'
4 then
5   echo "Sorry! You are $(whoami), and only the root can run this script"
6   exit 1
7 fi

```

```

8
9 SCITEX="SCITEX8"
10 KERNEL="vmlinuz-linux-2.6.33.2-AA1"
11
12 CDIR=$(dirname $(readlink -f $0))
13
14 SF="/mnt/SATA-Projects/SciTeX/CDroot/SciTeX/$SCITEX.gz"
15 if [ ! -f $SF ] ; then
16   echo "File $SF not found!"
17   exit 1
18 fi
19
20 KF="/mnt/SATA-Projects/AA1/CDroot/boot/$KERNEL"
21 if [ ! -f $KF ] ; then
22   echo "File $KF not found!"
23   exit 2
24 fi
25
26 cd /mnt/SATA-Projects/SciTeX/CDroot
27 mv SciTeX/$SCITEX.gz $SCITEX.gz
28 rm SciTeX/*
29 mv $SCITEX.gz SciTeX/$SCITEX.gz
30 cp $KF SciTeX/vmlinuz
31 rm boot/*
32 cat > boot/SciTeX.cfg << EOF
33 label $SCITEX
34   menu label $SCITEX
35     kernel /SciTeX/vmlinuz
36     append initrd=/SciTeX/$SCITEX.gz root=LABEL=Linuxboot
37 EOF
38
39 cd /mnt/SATA-Projects
40 $CDIR/imaging.sh hiso SciTeX/CDroot
41
42 cd /mnt/SATA-Projects/Imaging/output
43 if [ -f SD.iso ]
44 then
45   echo "Renaming SD.iso -> SciTeX.iso"
46   mv SD.iso SciTeX.iso
47 fi
48

```

Para crear una versión con FreeDOS, una vez que se ha ejecutado `imaging.sh` y el `make.sh` de DOS, hay que ejecutar `makeDOS.sh`.

../input/makeDOS.sh

```

1 #!/bin/bash
2
3 if test $(whoami) != 'root'
4 then
5   echo "Sorry! You are $(whoami), and only the root can run this script"
6   exit 1
7 fi
8
9 DOS="freedos.img"
10

```

```
11 SYSLINUX="syslinux-3.86"
12 KERNEL="memdisk"
13
14 CDIR=$(dirname $(readlink -f $0))
15
16 DF="/mnt/SATA-Projects/DOS/CDroot/DOS/$DOS"
17 if [ ! -f $SF ] ; then
18   echo "File $SF not found!"
19   exit 1
20 fi
21
22 KF="/mnt/SATA-Projects/Imaging/working/$SYSLINUX/$KERNEL/$KERNEL"
23 if [ ! -f $KF ] ; then
24   echo "File $KF not found!"
25   exit 2
26 fi
27
28 cd /mnt/SATA-Projects/DOS/CDroot
29 mv DOS/$DOS $DOS
30 rm DOS/*
31 mv $DOS DOS/$DOS
32 cp $KF DOS/$KERNEL
33 if [ -d boot ]
34 then
35   rm boot/*
36 else
37   mkdir boot
38 fi
39 cat > boot/DOS.cfg << EOF
40 label DOS
41   menu label ^FreeDOS
42     kernel /DOS/$KERNEL
43     append initrd=/DOS/$DOS
44 EOF
45
46 cd /mnt/SATA-Projects
47 $CDIR/imaging.sh zip DOS/CDroot
48
49 cd /mnt/SATA-Projects/Imaging/output
50 if [ -f SD.img ]
51 then
52   echo "Renaming SD.img -> FreeDOS.img"
53   mv SD.img FreeDOS.img
54 fi
55
```

Y, para crear una versión completa con todo.

../input/makeRAMon.sh

```
1 #!/bin/bash
2
3 if test $(whoami) != 'root'
4 then
5   echo "Sorry! You are $(whoami), and only the root can run this script"
6   exit 1
7 fi
```

```
8
9 ../../RAMon/input/ramon5.4.sh
10 ./dir2booting.1.sh hiso ../../RAMon/CDroot \
11     ../../DOS/CDroot ../../SciTeX/CDroot
12
```