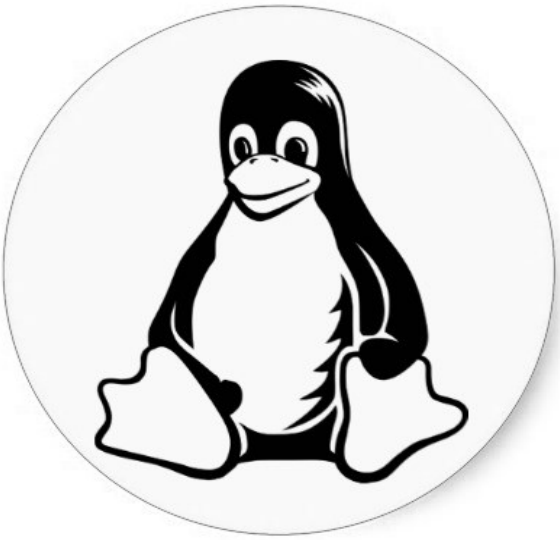
Universitat Oberta de Catalunya

LINUX DESDE CÓDIGO FUENTE

-Trabajo final de carrera-



Universitat Oberta de Catalunya
1er semestre 2013-2014
Alumno: Adrian Constantin Bungarzan

Índice de contenido

1. Introducción

No hay mejor manera para estudiar a fondo como funciona un Linux por dentro que compilándolo desde código fuente, tampoco hay mejor momento para ello que durante el trabajo de final de carrera donde uno lo da todo para conseguir el mejor resultado.

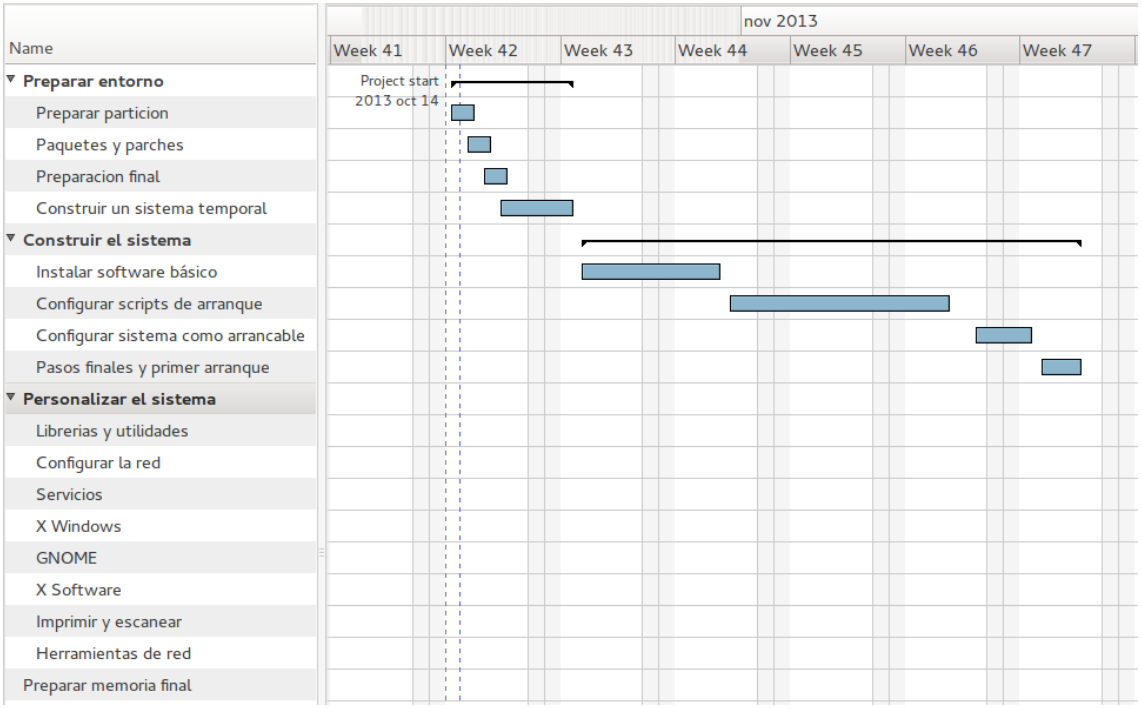
Siempre me ha gustado “trastear” con las distintas distribuciones Linux disponibles, soy ese tipo de personas a los que les gusta probar cosas nuevas y experimentar con lo desconocido. La primera instalación de una distribución Linux fue en el año 2006 cuando por vez primera he tocado este sistema operativo y le tocó a Mandrake ser el afortunado. Poco tiempo después lo siguieron otras distribuciones como RedHat, SuSE, Debian y algunas variantes basadas en Debian como Guadalinex o Ubuntu. Poco a poco este desconocido se ha hecho hueco en mi corazón y me ha hecho cambiar totalmente de aires en cuanto a sistemas operativos se refiere ya que era, como la mayoría, usuario Windows.

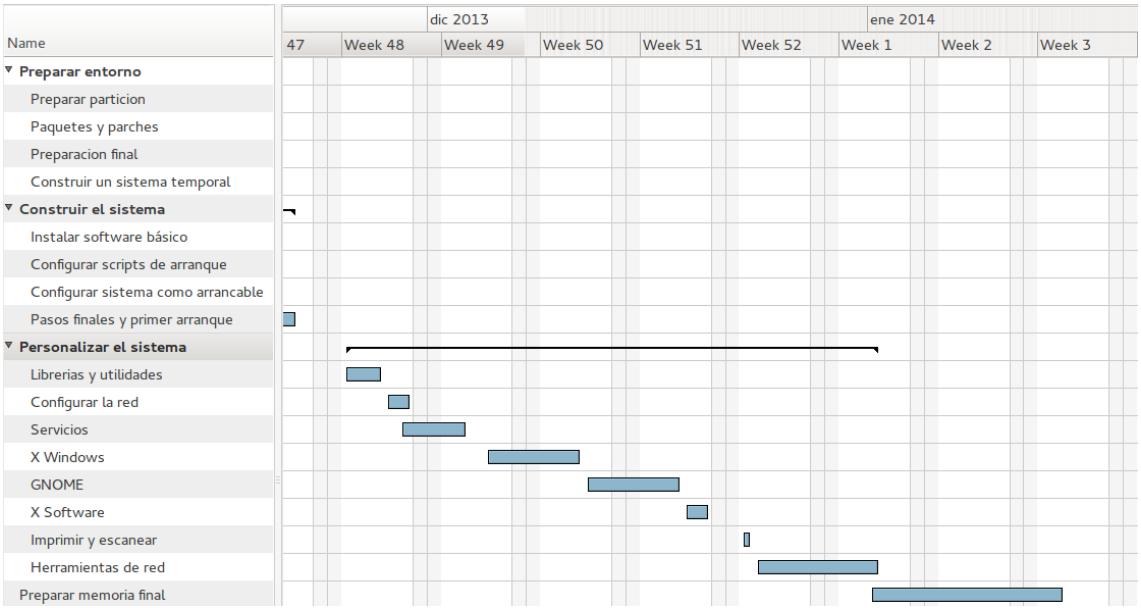
Han pasado siete años desde aquel momento y he aprendido muchas cosas. He conseguido dejar atrás el Windows sin tenerlo nisiquiera instalado en el disco duro de mi ordenador personal. A día de hoy no volvería a trabajar con un sistema Microsoft como sistema operativo principal.

Debido a este “sentimiento” que tengo por Linux he decidido enfocar el trabajo de final de carrera en como instalar un sistema mínimo desde código fuente, compilando el kernel y todos los paquetes y librerías necesarios para poder arrancar y trabajar con él.

2. Planificación inicial

En la PAC1 se ha entregado la siguiente planificación del proyecto pero a la finalización de este se ha demostrado que no se ha podido conseguir todos los puntos. Sin embargo los más importantes sí que se han podido incluir en el proyecto.





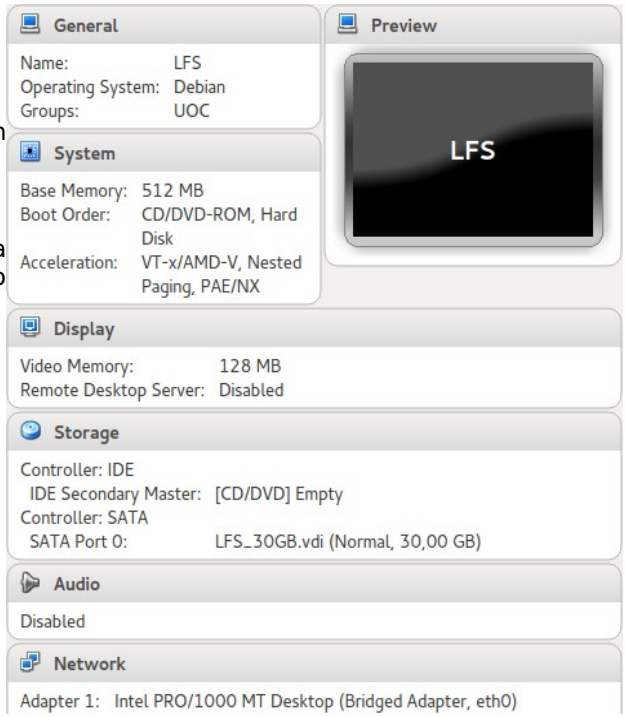
3. Entorno de trabajo

3.1 Configuración del sistema host

El sistema operativo que hace de host para el nuevo Linux es una distribución Debian 7.0.0 instalada en una máquina virtual sobre la plataforma VirtualBox. Esta máquina virtual tiene la siguiente configuración:

- 30 GB de disco duro
- 512 MB de memoria RAM
- 1 tarjeta de red
- 128 MB de memoria video sin acceleración 3D
- Sin soporte audio

En la siguiente captura se puede ver la configuración de la máquina virtual dentro de VirtualBox:



3.2 Requisitos mínimos

Esta máquina virtual tiene que cumplir una serie de requisitos de paquetes para asegurarse de que todo va a funcionar según lo previsto. Estos paquetes y sus versiones son los siguientes:

- Bash-3.2
- Binutils-2.17
- Bison-2.3
- Bzip2-1.0.4
- Coreutils-6.9
- Diffutils-2.8.1
- Findutils-4.2.31
- Gawk-4.0.1
- GCC-4.1.2
- Glibc-2.5.1
- Grep-2.5.1a
- Gzip-1.3.12
- Linux Kernel-2.6.32
- M4-1.4.10
- Make-3.81
- Patch-2.5.4
- Perl-5.8.8
- Sed-4.1.5
- Tar-1.18
- Texinfo-4.9
- Xz-5.0.0

Para verificar que el sistema host dispone de estas versiones mínimas hacemos uso del siguiente script *version-check.sh*:

```
#!/bin/bash
# Simple script to list version numbers of critical development tools

export LC_ALL=C
bash --version | head -n1 | cut -d" " -f2-4
echo "/bin/sh -> `readlink -f /bin/sh`"
echo -n "Binutils: "; ld --version | head -n1 | cut -d" " -f3-
bison --version | head -n1
if [ -e /usr/bin/yacc ];
  then echo "/usr/bin/yacc -> `readlink -f /usr/bin/yacc`";
  else echo "yacc not found"; fi

bzip2 --version 2>&1 < /dev/null | head -n1 | cut -d" " -f1,6-
echo -n "Coreutils: "; chown --version | head -n1 | cut -d")" -f2
diff --version | head -n1
find --version | head -n1
gawk --version | head -n1
if [ -e /usr/bin/awk ];
  then echo "/usr/bin/awk -> `readlink -f /usr/bin/awk`";
  else echo "awk not found"; fi

gcc --version | head -n1
g++ --version | head -n1
ldd --version | head -n1 | cut -d" " -f2-  # glibc version
grep --version | head -n1
gzip --version | head -n1
cat /proc/version
m4 --version | head -n1
make --version | head -n1
patch --version | head -n1
echo Perl `perl -V:version`
sed --version | head -n1
tar --version | head -n1
echo "Texinfo: `makeinfo --version | head -n1`"
xz --version | head -n1

echo 'main(){}' > dummy.c && g++ -o dummy dummy.c
if [ -x dummy ]
  then echo "g++ compilation OK";
  else echo "g++ compilation failed"; fi
rm -f dummy.c dummy
```

La ejecución de este script tiene la siguiente salida:

```
root@zenon:/mnt/lfs/sources# bash version-check.sh
bash, version 4.2.37(1)-release
/bin/sh -> /bin/dash
Binutils: (GNU Binutils for Debian) 2.22
bison (GNU Bison) 3.0
/usr/bin/yacc -> /usr/local/bin/bison
bzip2,  Version 1.0.6, 6-Sept-2010.
Coreutils:  8.13
diff (GNU diffutils) 3.2
find (GNU findutils) 4.4.2
GNU Awk 4.1.0, API: 1.0
/usr/bin/awk -> /usr/bin/mawk
gcc (Debian 4.7.2-5) 4.7.2
g++ (Debian 4.7.2-5) 4.7.2
(Debian EGLIBC 2.13-38) 2.13
grep (GNU grep) 2.12
gzip 1.5
Linux version 3.2.0-4-686-pae (debian-kernel@lists.debian.org) (gcc version 4.6.3 (De-
bian 4.6.3-15) ) #1 SMP Debian 3.2.41-2
m4 (GNU M4) 1.4.16
GNU Make 3.81
patch 2.6.1
Perl version='5.14.2';
GNU sed version 4.2.1
tar (GNU tar) 1.26
Texinfo: makeinfo (GNU texinfo) 4.13
xz (XZ Utils) 5.1.0alpha
g++ compilation OK
root@zenon:/mnt/lfs/sources#
```

El sistema host dispone de todos los paquetes necesarios para continuar con la instalación.

3.3 Particionado del disco

La máquina virtual tiene asignado un disco duro virtual de 30GB de los cuales 10GB están dedicados a la instalación del sistema host (en este caso es un Debian 7 32bits) y los 20GB restantes están destinados exclusivamente al nuevo Linux (de aquí en adelante LFS).

La disposición del espacio en el disco (/dev/sda) viene definida por las siguientes tabla:

Punto montaje	Partición	Tipo	Tamaño	Descripción
/	/dev/sda3	Ext4	10 GB	Partición raíz
/boot	/dev/sda2	Ext4	100 MB	Partición para el gestor de arranque (GRUB) y la imagen del kernel
/opt	/dev/sda4	Ext4	9,9 GB	Particón para la instalación de paquetes de gran tamaño.

Primero creamos una variable de entorno para que sea más comodo trabajar:

```
root@zenon:/home/lfs# export LFS=/mnt/lfs
```

Creamos los puntos de montaje y montamos las particiones:

```
root@zenon:/home/lfs# mkdir -pv $LFS
mkdir: created directory `/mnt/lfs'
root@zenon:/home/lfs# mount -v -t ext4 /dev/sda3 $LFS
/dev/sda3 on /mnt/lfs type ext4 (rw)
root@zenon:/home/lfs# mkdir -pv $LFS/boot
mkdir: created directory `/mnt/lfs/boot'
root@zenon:/home/lfs# mount -v -t ext4 /dev/sda2 $LFS/boot
/dev/sda2 on /mnt/lfs/boot type ext4 (rw)
root@zenon:/home/lfs# mkdir -pv $LFS/opt
mkdir: created directory `/mnt/lfs/opt'
root@zenon:/home/lfs# mount -v -t ext4 /dev/sda4 $LFS/opt
/dev/sda4 on /mnt/lfs/opt type ext4 (rw)
```

Finalmente comprobamos la nueva disposición del disco:

```
root@zenon:/home/lfs# df -h
Filesystem                                          Size  Used Avail Use% Mounted on
rootfs                                               12G  3.7G  7.0G  35% /
/dev/sda3                                           9.7G  150M  9.0G   2% /mnt/lfs
/dev/sda2                                            97M  5.6M   87M   7% /mnt/lfs/boot
/dev/sda4                                           8.7G  148M  8.1G   2% /mnt/lfs/opt
root@zenon:/home/lfs#
```

3.4 Paquetes y parches

Este punto incluye la lista de paquetes necesarios para construir un sistema Linux básico. Vamos a crear un directorio de trabajo `$LFS/sources` donde descargaremos todos estos paquetes y parches que luego tendremos que descomprimir y compilar. Utilizando este directorio los elementos necesarios estarán ubicados en la partición LFS y estarán disponibles durante todas las etapas del proceso.

Creamos el directorio:

```
root@zenon:/home/lfs# mkdir -v $LFS/sources
mkdir: created directory `/mnt/lfs/sources'
```

Configuramos permisos de escritura para este directorio y el *sticky bit* (con este bit activado solamente el propietario del directorio puede eliminar su contenido aunque el usuario tenga permisos de escritura):

```
root@zenon:/home/lfs# chmod -v a+wt $LFS/sources
mode of `/mnt/lfs/sources' changed from 0755 (rwxr-xr-x) to 1777 (rwxrwxrwt)
root@zenon:/home/lfs#
```

Descargamos todos los paquetes con la ayuda del fichero *wget-list* en la entrada del comando *wget*. El fichero *wget-list* contiene las direcciones de descarga de todos los paquetes necesarios:

```
http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.xz
http://ftp.gnu.org/gnu/automake/automake-1.14.tar.xz
http://ftp.gnu.org/gnu/bash/bash-4.2.tar.gz
http://alpha.gnu.org/gnu/bc/bc-1.06.95.tar.bz2
http://ftp.gnu.org/gnu/binutils/binutils-2.23.2.tar.bz2
http://ftp.gnu.org/gnu/bison/bison-3.0.tar.xz
http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz
http://sourceforge.net/projects/check/files/check/0.9.10/check-0.9.10.tar.gz
http://ftp.gnu.org/gnu/coreutils/coreutils-8.21.tar.xz
http://ftp.gnu.org/gnu/dejagnu/dejagnu-1.5.1.tar.gz
http://ftp.gnu.org/gnu/diffutils/diffutils-3.3.tar.xz
http://prdownloads.sourceforge.net/e2fsprogs/e2fsprogs-1.42.8.tar.gz
http://prdownloads.sourceforge.net/expect/expect5.45.tar.gz
ftp://ftp.astron.com/pub/file/file-5.14.tar.gz
http://ftp.gnu.org/gnu/findutils/findutils-4.4.2.tar.gz
http://prdownloads.sourceforge.net/flex/flex-2.5.37.tar.bz2
http://ftp.gnu.org/gnu/gawk/gawk-4.1.0.tar.xz
http://ftp.gnu.org/gnu/gcc/gcc-4.8.1/gcc-4.8.1.tar.bz2
http://ftp.gnu.org/gnu/gdbm/gdbm-1.10.tar.gz
http://ftp.gnu.org/gnu/gettext/gettext-0.18.3.tar.gz
http://ftp.gnu.org/gnu/glibc/glibc-2.18.tar.xz
ftp://ftp.gmplib.org/pub/gmp-5.1.2/gmp-5.1.2.tar.xz
http://ftp.gnu.org/gnu/grep/grep-2.14.tar.xz
http://ftp.gnu.org/gnu/groff/groff-1.22.2.tar.gz
http://ftp.gnu.org/gnu/grub/grub-2.00.tar.xz
http://ftp.gnu.org/gnu/gzip/gzip-1.6.tar.xz
http://anduin.linuxfromscratch.org/sources/LFS/lfs-packages/conglomeration//iana-etc/ia-
na-etc-2.30.tar.bz2
http://ftp.gnu.org/gnu/inetutils/inetutils-1.9.1.tar.gz
http://www.kernel.org/pub/linux/utils/net/iproute2/iproute2-3.10.0.tar.xz
http://ftp.altlinux.org/pub/people/legion/kbd/kbd-1.15.5.tar.gz
http://www.kernel.org/pub/linux/utils/kernel/kmod/kmod-14.tar.xz
http://www.greenwoodsoftware.com/less/less-458.tar.gz
http://www.linuxfromscratch.org/lfs/downloads/7.4/lfs-bootscripts-20130821.tar.bz2
http://download.savannah.gnu.org/releases/libpipeline/libpipeline-1.2.4.tar.gz
http://ftp.gnu.org/gnu/libtool/libtool-2.4.2.tar.gz
http://www.kernel.org/pub/linux/kernel/v3.x/linux-3.10.10.tar.xz
http://ftp.gnu.org/gnu/m4/m4-1.4.16.tar.bz2
http://ftp.gnu.org/gnu/make/make-3.82.tar.bz2
http://download.savannah.gnu.org/releases/man-db/man-db-2.6.5.tar.xz
http://www.kernel.org/pub/linux/docs/man-pages/man-pages-3.53.tar.xz
http://www.multiprecision.org/mpc/download/mpc-1.0.1.tar.gz
http://www.mpfr.org/mpfr-3.1.2/mpfr-3.1.2.tar.xz
http://ftp.gnu.org/gnu/ncurses/ncurses-5.9.tar.gz
http://ftp.gnu.org/gnu/patch/patch-2.7.1.tar.xz
http://www.cpan.org/src/5.0/perl-5.18.1.tar.bz2
http://pkgconfig.freedesktop.org/releases/pkg-config-0.28.tar.gz
http://sourceforge.net/projects/procps-ng/files/Production/procps-ng-3.3.8.tar.xz
http://prdownloads.sourceforge.net/psmisc/psmisc-22.20.tar.gz
http://ftp.gnu.org/gnu/readline/readline-6.2.tar.gz
http://ftp.gnu.org/gnu/sed/sed-4.2.2.tar.bz2
http://pkg-shadow.alioth.debian.org/releases/shadow-4.1.5.1.tar.bz2
http://www.infodrom.org/projects/sysklogd/download/sysklogd-1.5.tar.gz
http://download.savannah.gnu.org/releases/sysvinit/sysvinit-2.88dsf.tar.bz2
http://ftp.gnu.org/gnu/tar/tar-1.26.tar.bz2
http://prdownloads.sourceforge.net/tcl/tcl8.6.0-src.tar.gz
http://www.iana.org/time-zones/repository/releases/tzdata2013d.tar.gz
http://ftp.gnu.org/gnu/texinfo/texinfo-5.1.tar.xz
http://www.freedesktop.org/software/systemd/systemd-206.tar.xz
http://anduin.linuxfromscratch.org/sources/other/udev-lfs-206-1.tar.bz2
http://www.kernel.org/pub/linux/utils/util-linux/v2.23/util-linux-2.23.2.tar.xz
ftp://ftp.vim.org/pub/vim/unix/vim-7.4.tar.bz2
http://tukaani.org/xz/xz-5.0.5.tar.xz
http://www.zlib.net/zlib-1.2.8.tar.xz
http://www.linuxfromscratch.org/patches/lfs/7.4/automake-1.14-test-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/bash-4.2-fixes-12.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/bzip2-1.0.6-install_docs-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/coreutils-8.21-i18n-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/kbd-1.15.5-backspace-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/make-3.82-upstream_fixes-3.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/perl-5.18.1-libc-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/tar-1.26-manpage-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/readline-6.2-fixes-1.patch
http://www.linuxfromscratch.org/patches/lfs/7.4/texinfo-5.1-test-1.patch
```

```
root@zenon:/mnt/lfs/sources# wget -i wget-list -P $LFS/sources
.............
.............
```

```
.............
FINISHED --2013-10-17 21:30:25--
Total wall clock time: 7m 45s
Downloaded: 73 files, 315M in 6m 50s (786 KB/s)
root@zenon:/mnt/lfs/sources#
```

Antes de continuar verificamos que todos los paquetes y parches se han descargado correctamente con la ayuda del comando *md5sum* y del fichero *md5sums* que contiene el código hash de todos los ficheros:

```
root@zenon:/mnt/lfs/sources# md5sum -c md5sums
autoconf-2.69.tar.xz: OK
automake-1.14.tar.xz: OK
bash-4.2.tar.gz: OK
bc-1.06.95.tar.bz2: OK
binutils-2.23.2.tar.bz2: OK
bison-3.0.tar.xz: OK
bzip2-1.0.6.tar.gz: OK
check-0.9.10.tar.gz: OK
coreutils-8.21.tar.xz: OK
dejagnu-1.5.1.tar.gz: OK
diffutils-3.3.tar.xz: OK
e2fsprogs-1.42.8.tar.gz: OK
expect5.45.tar.gz: OK
file-5.14.tar.gz: OK
findutils-4.4.2.tar.gz: OK
flex-2.5.37.tar.bz2: OK
gawk-4.1.0.tar.xz: OK
gcc-4.8.1.tar.bz2: OK
gdbm-1.10.tar.gz: OK
gettext-0.18.3.tar.gz: OK
glibc-2.18.tar.xz: OK
gmp-5.1.2.tar.xz: OK
grep-2.14.tar.xz: OK
groff-1.22.2.tar.gz: OK
grub-2.00.tar.xz: OK
gzip-1.6.tar.xz: OK
iana-etc-2.30.tar.bz2: OK
inetutils-1.9.1.tar.gz: OK
iproute2-3.10.0.tar.xz: OK
kbd-1.15.5.tar.gz: OK
kmod-14.tar.xz: OK
less-458.tar.gz: OK
lfs-bootscripts-20130821.tar.bz2: OK
libpipeline-1.2.4.tar.gz: OK
libtool-2.4.2.tar.gz: OK
linux-3.10.10.tar.xz: OK
m4-1.4.16.tar.bz2: OK
make-3.82.tar.bz2: OK
man-db-2.6.5.tar.xz: OK
man-pages-3.53.tar.xz: OK
mpc-1.0.1.tar.gz: OK
mpfr-3.1.2.tar.xz: OK
ncurses-5.9.tar.gz: OK
patch-2.7.1.tar.xz: OK
perl-5.18.1.tar.bz2: OK
pkg-config-0.28.tar.gz: OK
procps-ng-3.3.8.tar.xz: OK
psmisc-22.20.tar.gz: OK
readline-6.2.tar.gz: OK
sed-4.2.2.tar.bz2: OK
shadow-4.1.5.1.tar.bz2: OK
sysklogd-1.5.tar.gz: OK
sysvinit-2.88dsf.tar.bz2: OK
tar-1.26.tar.bz2: OK
tcl8.6.0-src.tar.gz: OK
tzdata2013d.tar.gz: OK
texinfo-5.1.tar.xz: OK
systemd-206.tar.xz: OK
udev-lfs-206-1.tar.bz2: OK
util-linux-2.23.2.tar.xz: OK
vim-7.4.tar.bz2: OK
xz-5.0.5.tar.xz: OK
zlib-1.2.8.tar.xz: OK
automake-1.14-test-1.patch: OK
bash-4.2-fixes-12.patch: OK
bzip2-1.0.6-install_docs-1.patch: OK
coreutils-8.21-i18n-1.patch: OK
kbd-1.15.5-backspace-1.patch: OK
```

```
make-3.82-upstream_fixes-3.patch: OK
perl-5.18.1-libc-1.patch: OK
tar-1.26-manpage-1.patch: OK
readline-6.2-fixes-1.patch: OK
texinfo-5.1-test-1.patch: OK
root@zenon:/mnt/lfs/sources#
```

3.5 Preparación final

Crear el directorio $LFS/tools

Creamos el directorio `$LFS/tools` para mantener separados los programas compilados en el punto *Construir un sistema temporal* de los programas compilados en el punto *Instalar software básico del sistema.* Los programas compilados aquí son herramientas temporales y no van a ser parte del sistema final. Manteniendo estos programas separados en otro directorio, pueden ser descartados sin dificultad más tarde cuando ya no los necesitamos. Esto también evitan que estos programas acaben por error en los directorios del sistema host.

```
root@zenon:/mnt/lfs/sources# mkdir -v $LFS/tools
mkdir: created directory `/mnt/lfs/tools'
root@zenon:/mnt/lfs/sources#
```

El siguiente paso es crear un accesos simbólico a `/tools` del sistema host. Esto apuntará al nuevo directorio recién creado en la partición LFS:

```
root@zenon:/mnt/lfs/sources# ln -sv $LFS/tools /
`/tools' -> `/mnt/lfs/tools'
root@zenon:/mnt/lfs/sources#
```

El enlace creado hace que las herramientas a compilar hagan referencia siempre a `/tools`, es decir, el compilador, el ensamblador y el enlazador funcionarán tanto antes como después de hacer el *chroot*.

Crear el usuario LFS

Cuando trabajamos con el usuario *root* podemos cometer errores que pueden dañar o destruir el sistema. Por este motivo es recomandable compilar los paquetes como usuario sin privilegios. Crearemos el usuario *lfs* como miembro del nuevo grupo *lfs*:

```
root@zenon:/mnt/lfs/sources# groupadd lfs
root@zenon:/mnt/lfs/sources# useradd -s /bin/bash -g lfs -m -k /dev/null lfs
Not copying any file from skel directory into it.
root@zenon:/mnt/lfs/sources#
```

El significado de las opciones:

`-s /bin/bash`	Esto configura **bash** como el shell por defecto para el usuario *lfs*
`-g lfs`	Esta opción agrega el usuario *lfs* al grupo *lfs*
`-m`	Esto crear el directorio home para *lfs*
`-k /dev/null`	Este parámetro evita posibles copias de los ficheros del directorio skeleton (por defecto /etc/skel) cambiando la entrada por el dispositivo especial *null*
`lfs`	Es el nombre para el usuario y grupo creados

Para poder iniciar sesión con el usuario *lfs* tenemos que asignarle una contraseña:

```
root@zenon:/mnt/lfs/sources# passwd lfs
```

Concedemos al usuario *lfs* acceso total en los directorios $LFS/tools y $LFS/sources cambiando el propietario del directorio:

```
root@zenon:/mnt/lfs/sources# chown -v lfs $LFS/tools
changed ownership of `/mnt/lfs/tools' from root to lfs
root@zenon:/mnt/lfs/sources# chown -v lfs $LFS/sources
changed ownership of `/mnt/lfs/sources' from root to lfs
root@zenon:/mnt/lfs/sources#
```

Configurar el entorno

Para un mejor entorno de trabajo vamos a crear dos ficheros de arranque para la shell de bash. Uno de ellos es ~/.bash_profile y contiene esta línea:

```
exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash
```

Cuando estamos autenticados con usuario *lfs* la shell inicial es normalmente una del tipo *login* que lee los ficheros */etc/profile* y *.bash_profile*. El comando *exec env -i … /bin/bash* del fichero recién creado sustituye la shell que se está ejecutando por una nueva con un entorno completamente vacío, con la excepción de las variables *HOME*, *TERM* y *PS1*. Con este método nos aseguramos tener un entorno limpio.
La nueva instancia shell es del tipo *non-login* que no lee el fichero */etc/profile* o *.bash_profile*, pero en cambio lee el fichero *.bashrc*. A continuación creamos el fichero *.bashrc*:

```
set +h
umask 022
LFS=/mnt/lfs
LC_ALL=POSIX
LFS_TGT=$(uname -m)-lfs-linux-gnu
PATH=/tools/bin:$PATH
export LFS LC_ALL LFS_TGT PATH
```

El comando *set +h* desactiva la función hash de bash. Desactivando esta función la shell siempre buscará en el PATH cuando va a ejecutar un programa.
Configurando la máscara de creación de ficheros (umask) a 0222 nos aseguramos que los nuevos ficheros y directorios creados solo estarán disponibles con permisos de escritura para el propietario pero cualquier usuario tiene acceso de lectura y ejecución.
La variable *LFS* tiene el valor de la ruta del punto de montaje, */mnt/lfs*.
La variable *LC_ALL* controla la localización de ciertos programas haciendo que sus mensajes cumplan con las convenciones de un país específico. Configurando el valor de esta variable a *POSIX* o *C* (las dos son equivalentes) nos aseguramos de que todo va a funcionar según lo esperado cuando hayamos cambiado de entorno (*chroot*).
LFS_TGT es la variable encargada de configurar la descripción de la máquina que utilizaremos a la hora de compilar nuestro *toolchain* temporal.
Poniendo */tools/bin* al principio del *PATH*, todos los programas instalados en el apartado *Construir un sistema temporal,* van a estar disponibles para la shell inmediatamente después de la instalación. Esta opción junto con la desactivación de la función hash disminuye el riesgo de utilizar programas antiguos de la máquina host cuando los mismos programas están disponibles en el entorno de sistema temporal que vamos a crear a continuación.

Finalmente ejecutamos el siguiente comando para tener el entorno totalmente preparado para compilar las herramientas temporales:

```
lfs@zenon:~$ source ~/.bash_profile
```

3.6 Construir un sistema temporal

En este punto vamos a construir un sistema Linux mínimo que va a contener las herramientas necesarias para empezar a construir el sistema LFS final en el próximo apartado *Construir el sistema*.
La construcción de este sistema mínimo contiene dos etapas. La primera de ellas es preparar un nuevo *toolchain* (compilador, ensamblador, enlazador, librerías y unas cuantas utilidades de gran uso) totalmente independiente del sistema host. La segunda etapa utiliza estas herramientas para preparar otras herramientas esenciales.

Los ficheros compilados en este capitulo se van a instalar en el directorio $LFS/tools para mantenerlos separados de los ficheros instalados en el siguiente capítulo y los directorios del sistema host productivo.

Antes de continuar he extraido todos los paquetes del directorio $LFS/sources.

Binutils-2.23.2 – Paso 1

El paquete Binutils contiene un enlazador, un ensamblador y otras herramientas para manejar *object files*.

La documentación de Binutils recomienda compilarlo fuera del directorio origen en un directorio dedicado, para ello se ha creado el dictorio `$LFS/sources/binutils-build` y desde aquí se han invocado los comandos configure, make y make install:

```
../binutils-2.23.2/configure   \
    --prefix=/tools            \
    --with-sysroot=$LFS        \
    --with-lib-path=/tools/lib \
    --target=$LFS_TGT          \
    --disable-nls              \
    --disable-werror && make && make install
```

Tiempo: 2m26.073s

Opciones utilizadas:

`--prefix=/tools`	Esto le dice al script configure que prepare la instalación de Binutils en el directorio /tools
`--with-sysroot=$LFS`	Le dice al sistema que busque las librerías necesarias dentro de $LFS
`--with-lib-path=/tools/lib`	Esto especifica la ruta de la librería a ser utilizada por el linker
`--target=$LFS_TGT`	Esta opción le dice al script configure que utilice como descripción de la maquina el valor de la variable LFS_TGT para que sea diferente del nombre devuelto por el script config.guess
`--disable-nls`	Desactiva Native Language Support (NLS) que no es necesario para las herramientas temporales
`--disable-werror`	Evita parar la compilación en caso de mensajes de warning

GCC-4.8.1 – Paso 1

El paquete GCC contiene el set del compilador, que a su vez contiene los compiladores C y C++. Actualmente GCC requiere los paquetes GMP, MPFR y MPC así que antes de continuar se han descomprimido y renombrado los directorios a gmp, mpfr y mpc quitando la nomenclatura de la versión.

Vamos a crear un directorio dedicado de la misma manera que hemos hecho con Binutils. En este caso el directorio se llamará `gcc-build` y desde aquí invocamos los comandos:

```
../gcc-4.8.1/configure                                \
    --target=$LFS_TGT                                 \
    --prefix=/tools                                   \
    --with-sysroot=$LFS                               \
    --with-newlib                                     \
    --without-headers                                 \
    --with-local-prefix=/tools                        \
    --with-native-system-header-dir=/tools/include    \
    --disable-nls                                     \
    --disable-shared                                  \
    --disable-decimal-float                           \
    --disable-threads                                 \
    --disable-libatomic                               \
    --disable-libgomp                                 \
    --disable-libitm                                  \
    --disable-libmudflap                              \
    --disable-libquadmath                             \
    --disable-libsanitizer                            \
    --disable-libssp                                  \
    --disable-libstdc++-v3                            \
    --enable-languages=c,c++                          \
    --with-mpfr-include=$(pwd)/../gcc-4.8.1/mpfr/src \
    --with-mpfr-lib=$(pwd)/mpfr/src/.libs && make && make install
```

Tiempo: 16m30.641s

Significado de las opciones utilizadas:

`--with-newlib`	Teniendo en cuenta que la librería C todavía no está disponible nos aseguramos de que la constante inhibit_libc está definida cuando compilamos libgcc.
`--without-headers`	Evita que GCC busque las cabeceras estandares compatibles con el sistema, para nuestros propósitos estas no serán necesarias.
`--with-local-prefix=/tools`	Es la ubicación en el sistema donde GCC buscará los ficheros *include* instalados. Por defecto es /usr/local.
`--with-native-system-header-dir=/tools/include`	Por defecto GCC busca las cabeceras del sistema en /usr/include.
`--disable-shared`	Este switch fuerza GCC para enlazar las librerías internas de forma estática. Con esto evitamos posibles problemas con el sistema host.
`--disable-decimal-float`, `--disable-threads`, `--disable-libatomic`, `--disable-libgomp`, `--disable-libitm`, `--disable-libmudflap`, `--disable-libquadmath`, `--disable-libsanitizer`, `--disable-libssp`, `--disable-libstdc++-v3`	Estos switches se deshabilitan porque sino fallarán a la hora de preparar un compilador cruzado (cross-compiler) y no son necesarios para compilar el libc temporal.
`--enable-languages=c,c++`	Esta opción asegura que se van a compilar los compialdores C y C++ unicamente.

Linux-3.10.10 API Headers

Linux kernel necesita de un API (Application Programming Interface) para la librería C del sistema (en este caso es Glibc).

Verificamos y extraemos las cabeceras del kernel. Se pasan a un directorio intermedio y se copian en la ubicación necesaria porque la extracción elimina cualquier fichero del directorio destino:

```
make headers_check
make INSTALL_HDR_PATH=dest headers_install
cp -rv dest/include/* /tools/include
```

Tiempo: 0m12.421s

Glibc-2.18

Contiene la librería C principal. Esta librería proporciona las rutinas básicas para alocar memoria, buscar directorios, abrir y cerrar ficheros, leer y escribir ficheros, manipular cadenas de texto, igualar patrones, operaciones aritméticas y muchas otras.

La documentación de Glibc recomienda compilar Glibc en un directorio dedicado. Para ello se ha creado el directorio `$LFS/sources/glibc-build` y se ha ejecutado la siguiente instrucción:

```
../glibc-2.18/configure                             \
      --prefix=/tools                               \
      --host=$LFS_TGT                               \
      --build=$(../glibc-2.18/scripts/config.guess) \
      --enable-kernel=2.6.32                        \
      --with-headers=/tools/include                 \
      --libc_cv_forced_unwind=yes && make && make install
```

Tiempo: 19m30.122s

Significado de las opciones de configure:

`--host=$LFS_TGT, --build=$ (../glibc-2.18/scripts/config.guess)`	El efecto de combindar estos switches es que Glibc se autoconfigura para cross-compile usando el cross-linker y el cross-compiler ubicados en /tools
`--enable-kernel=2.6.32`	Prepara Glibc para compilar la librería con soporte para Linux kernels versión 2.6.32 y posteriores
`--with-headers=/tools/include`	Esto le indica a Glibc que se compile contra las cabeceras recién instaladas en el directorio de herramientas, para que sepa exactamente qué características tiene el núcleo y pueda optimizarse.
`--libc_cv_forced_unwind=yes`	Este switch se pasa para indicarle al script configure que está disponible el soporte para force-unwind

Libstdc++-4.8.1

Libstdc++ es la librería C++ estandar. Es necesaria para el correcto funcionamiento del compilador g++. Creamos un directorio dedicado para Libstdc++, entramos en el y ejecutamos la compilación:

```
../gcc-4.8.1/libstdc++-v3/configure     \
    --host=$LFS_TGT                      \
    --prefix=/tools                      \
    --disable-multilib                   \
    --disable-shared                     \
    --disable-nls                        \
    --disable-libstdcxx-threads          \
    --disable-libstdcxx-pch              \
    --with-gxx-include-dir=/tools/$LFS_TGT/include/c++/4.8.1 && make && make install
```

Tiempo: 0m51.979s

Significado de las opciones:

`--host=$LFS_TGT`	Indica que se utilizará el compilador recién creado en lugar del que hay por defecto /usr/bin
`--disable-libstdcxx-threads`	Todavía no hemos compilado los hiso de la librería C, por lo tanto la librería C++ no se podrá compilar
`--disable-libstdcxx-pch`	Esta opción evita la instalación de ficheros *include* precompilados no necesarios en este momento
`--with-gxx-include-dir=/tools/ $LFS_TGT/include/c++/4.8.1`	Informamos de forma explcícita la ubicación donde el compilador buscará los ficheros *include*

Binutils-2.23.2 – Paso 2

Preparamos, compilamos e instalamos:

```
CC=$LFS_TGT-gcc                \
AR=$LFS_TGT-ar                 \
RANLIB=$LFS_TGT-ranlib         \
../binutils-2.23.2/configure   \
    --prefix=/tools            \
    --disable-nls              \
    --with-lib-path=/tools/lib \
    --with-sysroot && make && make install
```

Tiempo: 2m24.244s

Significado de los parámetros:

`CC=$LFS_TGT-gcc AR=$LFS_TGT-ar RANLIB=$LFS_TGT-ranlib`	Teniendo en cuenta que es una instalación nativa de Binutils, configuramos estas variables para asegurarnos de que el sistema utilizará el nuestro compilador en lugar del compilador del sistema host.
`--with-lib-path=/tools/lib`	Esto le indica al script configure el path de búsqueda durante la compilación de Binutils, evita que busque en directorios del

	sistema host.
`--with-sysroot`	Esta opción configura el enlazador (linker) para que encuentre objetos compartidos que son necesarios por otros objetos compartidos explícitamente incluidos en la línea de comandos del enlazador. Sin esta opción algunos paquetes pueden no compilarse correctamente.

Ahora preparamos el enlazador para la fase de "reajuste" del siguiente punto (*GCC-4.8.1 – Paso 2*):

```
make -C ld clean
make -C ld LIB_PATH=/usr/lib:/lib
cp -v ld/ld-new /tools/bin
```

El parámetro *ld clean* elimina todos los ficheros compilados en el subdirectorio *ld* mientras que la variable *LIB_PATH* de la segunda compilación reconstruye todo el subdirectorio *ld*.

GCC-2.8.1 – Paso 2

```
CC=$LFS_TGT-gcc                                          \
CXX=$LFS_TGT-g++                                         \
AR=$LFS_TGT-ar                                           \
RANLIB=$LFS_TGT-ranlib                                   \
../gcc-4.8.1/configure                                   \
    --prefix=/tools                                      \
    --with-local-prefix=/tools                           \
    --with-native-system-header-dir=/tools/include       \
    --enable-clocale=gnu                                 \
    --enable-shared                                      \
    --enable-threads=posix                               \
    --enable-__cxa_atexit                                \
    --enable-languages=c,c++                             \
    --disable-libstdcxx-pch                              \
    --disable-multilib                                   \
    --disable-bootstrap                                  \
    --disable-libgomp                                    \
    --with-mpfr-include=$(pwd)/../gcc-4.8.1/mpfr/src \
    --with-mpfr-lib=$(pwd)/mpfr/src/.libs && make && make install
```

Tiempo: 20m55.159s

Significado de los parámetros:

`--enable-clocale=gnu`	Esta opcion asegura que se selecciona el modelo regional correcto para las librerías C++ bajo todas las circunstancias.
`--enble-threads=posix`	Activa el manejo de excepciones de C++ para código multi-hilo.
`--enable-__cxa_atexit`	Esta opción permite el uso de `__cxa_atexit` (en lugar de `atexit`) para registrar los destructores C++ para objetos locales y globales. Esta opción es esencial para el completo manejo de los destructores.
`--enable-languages=c,c++`	Esta opción asegura que se van a instalar los dos compiladores, C y C++.
`--disable-libstdcxx-pch`	No instala las cabeceras precompiladas para libstdc++. Ocupan mucho espacio y no son necesarias.
`--disable-bootstrap`	Para instalaciones nativas de GCC por defecto se hace una instalación *bootstrap*. Esto compila Gcc varias veces para asegurarse una correcta compilación.

Muchos programas y scripts ejecutan *cc* en lugar de *gcc*. Por esta razón vamos a crear un enlace simbólico:

```
ln -sv gcc /tools/bin/cc
```

Tcl-8.6.0

El paquete Tcl contiene el set *Tool Command Language*.

Preparamos, compilamos e instalamos:

```
cd unix
./configure –prefix=/tools && make && make install
```

Tiempo: 2m15.565s

Damos permiso de escritura sobre la librería instalada para que los símbolos de depuración se puedan eliminar más tarde:

```
chmod -v u+w /tools/lib/libtcl8.6.so
```

Instalamos las cabeceras de Tcl necesarias para el paquete *Expect*:

```
make install-private-headers
```

Expect-5.45

El paquete Expect contiene un programa para la realización de diálogos programados con otros programas interactivos.

Antes de nada forzamos el script configure para que use */bin/stty* en lugar de */usr/loca/bin/stty* que puede encontrarse en el sistema host:

```
./configure --prefix=/tools --with-tcl=/tools/lib \
  --with-tclinclude=/tools/include
```

Tiempo: 0m13.480s

Significado de las opciones:

`--with-tcl=/tools/lib`	Asegura que el script configure encuentra la instalación de Tcl en el sistema temporal en lugar del ya existente en el sistema host.
`--with-tclinclude=/tools/include`	Indica explícitamente donde encontrar las cabeceras internas de Tcl.

DejaGNU-1.5.1

Este paquete contiene un framework que se utiliza para testear otros programas. El programa instalado se llama *runtest* y es un script que localiza la shell **expect** correcta y luego ejecuta DejaGNU.

```
./configure --prefix=/tools && make install
```

Tiempo: 0m1.619s

Check-0.9.10

Es un framework de pruebas para C.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m16.813s

Ncurses-5.9

Contiene librerías que permiten la construcción de una interfaz para el usuario, para aplicaciones ejecutadas en un terminal.

```
./configure --prefix=/tools --with-shared \
    --without-debug --without-ada –enable-overwrite
```

Tiempo: 2m59.398s

Significado de los parámetros:

`--without-debug`	Esto asegura que Ncurses no instala el soporte para el compilador Ada
`--without-ada –enable-overwrite`	Esto le indica a Ncurses que instale sus ficheros de cabecera dentro de /tools/include en lugar de /tools/include/ncurses para asegurar que otros programas puedan encontrar las cabeceras de Ncurses satisfactoriamente.

Bash-4.2

El paquete Bash contiene *Bourne-Again SHell*.

```
./configure --prefix=/tools --without-bash-malloc && make && make install
```

Tiempo: 2m10.315s

La opción *--without-bash-malloc* desactiva el uso de la función de asignación de memoría (malloc)debido a que es conocido que causa fallos de segmentación (*segmentation faults)*. Al desactivar esta opción, Bash utilizará las funciones `malloc` de Glibc que son más estables.

Creamos un enlace simbólico para programas que utilizan **sh** para shell:

```
ln -sv bash /tools/bin/sh
```

Bzip2-1.0.6

Este paquete contiene programas para comprimir y descomprimir ficheros. Comprimir ficheros de texto con **bzip2** produce un porcentaje de compresión mucho mejor que con el tradicional **gzip**.

El paquete Bzip2 no contiene un script **configure**.

```
make && make PREFIX=/tools install
```

Tiempo: 0m10.221s

Coreutils-8.21

El paquete Coreutils contiene utilidades para mostrar y configurar las características básicas del sistema.

```
./configure --prefix=/tools --enable-install-program=hostname && make && make install
```

Tiempo: 4m30.412s

La opción *--enable-install-program* activa la instalación del binario **hostname**, por defecto esta opción está desactivada pero es necesaria para las pruebas del paquete Perl.

Diffutils-3.3

Este paquete contiene programas que muestran las diferencias entre ficheros o directorios.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m21.996s

File-5.14

El paquete File contiene una utilidad que determina el tipo de un fichero especificado.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m29.217s

Findutils-4.4.2

El paquete Findutils contiene programas para buscar ficheros. Estos programas se proporcionan para buscar recursivamente en un árbol de directorios y para crear, mantener y buscar una base de datos (muchas veces más rápido que la búsqueda recursiva pero poco fiable si la base de datos no ha sido actualizada reciéntemente).

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m13.599s

Gawk-4.1.0

El paquete Gawk contiene programas para manipular ficheros de texto.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m20.159s

Getttext-0.18.3

El paquete Gettext contiene utilidades para las opciones regionales y de localización. Estas permiten que los programas se puedan compilar con NLS (*Native Language Support*), activando la salida de mensajes en el idioma nativo del usuario.

Para nuestro set temporal de herramientas solamente necesitamos compilar e instalar un binario de Gettext. Preparamos y compilamos el paquete:

```
EMACS="no" ./configure --prefix=/tools --disable-shared
make -C gnulib-lib
make -C src msgfmt
```

Tiempo: 3m30.339s

La opción *–disable-shared* evita la instalación de las librerías compartidas de Gettext.

Ahora que ya está compilado solo falta instalar el binario:

```
cp -v src/msgfmt /tools/bin
```

Grep-2.14

El paquete Grep contiene programas para buscar dentro de ficheros.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m9.858s

Gzip-1.6

El paquete Gzip contiene programas para comprimir y descomprimir ficheros.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m48.057s

M4-1.4.16

El paquete M4 contiene un procesador de macros.

Antes de compilar reparamos una incompatibilidad entre este paquete y Glibc-2.18:

```
sed -i -e '/gets is a/d' lib/stdio.in.h
```

Instalamos el paquete:

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m59.299s

Make-3.82

El paquete Make contiene un programa para compilar paquetes.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m29.553s

Patch-2.7.1

El paquete Patch contiene un programa para modificar o crear ficheros aplicando un fichero *parche* normalmente creado con el programa **diff.**

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m1.124s

Perl-5.18.1

El paquete Perl contiene Practical Extraction and Report Language.

Primero aplicamos el siguiente parche para adaptar algunas rutas de la librería C:

```
patch -Np1 -i ../perl-5.18.1-libc-1.patch
```

Preparamos y compilamos:

```
sh Configure -des -Dprefix=/tools && make
```

Tiempo: 9m31.485s

Por el momento solo algunas herramientas y librerías son necesarias:

```
cp -v perl cpan/podlators/pod2man /tools/bin
mkdir -pv /tools/lib/perl5/5.18.1
cp -Rv lib/* /tools/lib/perl5/5.18.1
```

Sed-4.2.2

El paquete Sed contiene un editor de flujos de texto.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 0m55.878s

Tar-1.26

El paquete Tar contiene un programa para archivar ficheros.

Antes de compilar reparamos una incompatibilidad entre este paquete y Glibc-2.18:

```
sed -i -e '/gets is a/d' lib/stdio.in.h
```

Instalamos el paquete:

```
./configure --prefix=/tools && make && make install
```

Tiempo: 2m9.200s

Texinfo-5.1

El paquete Texinfo contiene programas para leer, escribir y convertir páginas de información.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m39.690s

Xz-5.0.5

El paquete Xz contiene programas para comprimir y descomprimir ficheros. Ofrece capacidades para trabajar con los formatos de comperesión *lzma* y el nuevo *xz*. Comprimir ficheros de texto con **xz** produce un mejor porcentaje de compresión que los tradicionales **gzip** o **bzip2**.

```
./configure --prefix=/tools && make && make install
```

Tiempo: 1m12.036s

3.6 Cambiar propietario de $LFS/tools

Actualmente el directorio $LFS/tools tiene configurado como propietario el usuario *lfs*, un usuario que solamente existe en el sistema host. Si este directorio se mantiene tal cual, los ficheros tienen como propietario un código de usuario (User ID) sin una cuenta correspondiente. Esto es peligroso porque una cuenta de usuario creada más tarde puede tener el mismo User ID y sería el propietario del directorio $LFS/tools y de todos los ficheros aquí dentro, exponiéndolos estos ficheros a una posible manipulación maliciosa.

Para evitar este problema cambiamos el propietario del directorio al usuario *root*:

```
chown -R root:root $LFS/tools
```

4. Construir el sistema

4.1 Introducción

En este capítulo empezamos la instalación del sistema final, eso quiere decir que cambiamos de raíz (*chroot*), hacemos unas cuantas preparaciones y luego empezamos a instalar los paquetes. La clave de entender que es lo que hace funcionar un sistema Linux es saber para que se utiliza cada paquete y porque es necesario.

Todos los paquetes se van a instalar secuencialmente y no en paralelo para evitar problemas con algunos paquetes que puedan tener referencias al directorio `/tools`.

4.2 Preparar el sistema de ficheros del kernel virtual

Varios sistemas de ficheros exportados por el kernel se utilizan para comunicar hacia y desde el kernel mismo. Estos sistemas de ficheros son virtuales en el sentido de que no se va a utilizar espacio en disco para ellos, su contenido reside en la memoria.

Creamos los directorios donde se montaran estos sistemas de archivos:

```
mkdir -v $LFS/{dev,proc,sys}
```

Cuando el kernel arranca el sistema necesita que existan unos nodos de dispositivos, concretamente los dispositivos *console* y *null*. Se tienen que crear en el disco duro así que estarán disponibles antes de que *udevd* arranque y adicionalmente cuando Linux se inicia con *init=/bin/bash*. Creamos los dispositivos:

```
mknod -m 600 $LFS/dev/console c 5 1
mknod -m 666 $LFS/dev/null c 1 3
```

El método recomendado para poblar el directorio */dev* con dispositivos es montando un sistema de ficheros virtual (como *tmpfs*) en el directorio */dev* y permitir que los dispositivos se puedan crear de forma dinámica en este sistema de ficheros tan pronto se detecten o se acceda a ellos. La creación de los dispositivos normalmente se hace durante el proceso de arranque por Udev. Teniendo en cuenta que el sistema no tiene todavía Udev y todavía no se ha arrancado, es necesario montar y pobla */dev* manualmente. Esto se consigue haciendo un montaje enlazado con el directorio */dev* del sistema host:

```
mount -v --bind /dev $LFS/dev
```

Ahora podemos montar los sistemas de ficheros virtuales restantes:

```
mount -vt devpts devpts $LFS/dev/pts -o gid=5,mode=620
mount -vt proc proc $LFS/proc
mount -vt sysfs sysfs $LFS/sys
```

Significado de las opciones de montaje para devpts:

`gid=5`	Esto nos asegura que todos los dispositivos del tipo *devpts* creados tienen como propietario el grupo con ID 5. Este es el ID que usaremos más adelante para el grupo `tty`. Utilizamos el ID del grupo en lugar del nombre porque el sistema host puede utilizar un ID diferente para el grupo `tty`.
`mode=0620`	Esto asegura que todos los dispotivos creados del tipo *devpts* tienen el modo 0620 (permiso de lectura y escritura para el usuario y permiso de escritura para el grupo)

4.3 Cambiar el entorno raíz (chroot)

Llegados en este punto tenemos todas las herramientas compiladas en el entorno temporal preparado en los pasos anteriores. Ahora es el momento de cambiar de entorno de trabajo, es decir, vamos a trabajar sobre el sistema final y vamos a comenzar la instalación propiamente dicha de nuestro Linux.

Hacemos uso del comando *chroot* para cambiar de raíz. Con este comando, aparte de hacer el cambio de raíz a */mnt/lfs*, tambíen limpiamos todas las variables de entorno (*env -i*) y configuramos solamente las siguientes: *HOME, TERM, PS1* y *PATH*.

```
root@zenon:/# chroot "$LFS" /tools/bin/env -i  \
HOME=/root                        \
TERM="$TERM"                      \
PS1='\u:\w\$ '                    \
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \
/tools/bin/bash --login +h
I have no name!:/#
```

Es muy importante agregar al final del PATH el directorio */tools/bin.* Eso quiere decir que tan pronto el programa se encuentre instalado en su versión final (en */usr/bin*) ya no se utilizará la versión instalada en el sistema temporal (*/tools/bin*).

El prompt ha cambiado a "I have no name!", eso es normal porque todavía no se ha creado el fichero */etc/passwd.* En cuanto bash encuentre este fichero buscará el usuario y devolverá el valor correcto.

4.4 Crear la estructura de directorios

Ya hemos cambiado de raíz, vamos a crear la estructura de directorios necesaria para que un sistema Linux básico pueda trabajar sin problemas. Esta estructura está basada en el estandard FHS (Filesystem Hierarchy Standard), no obstante esta estructura incluye muchos más directorios que para nuestro objetivo no son necesarios y por lo tanto no se van a crear.
Todos los directorios se crean con la máscara de permisos 755 (acceso total para root y de lectura y ejecución para los demás usuarios).

```
I have no name!:/# mkdir -pv /{bin,boot,etc/{opt,sysconfig},home,lib,mnt,opt,run}
mkdir: created directory '/bin'
mkdir: created directory '/etc'
mkdir: created directory '/etc/opt'
mkdir: created directory '/etc/sysconfig'
mkdir: created directory '/home'
mkdir: created directory '/lib'
mkdir: created directory '/mnt'

I have no name!:/# mkdir -pv /{media/{floppy,cdrom},sbin,srv,var}
mkdir: created directory '/media'
mkdir: created directory '/media/floppy'
mkdir: created directory '/media/cdrom'
mkdir: created directory '/sbin'
mkdir: created directory '/srv'
mkdir: created directory '/var'
I have no name!:/#
```

```
I have no name!:/# install -dv -m 0750 /root
install: creating directory '/root'
I have no name!:/#
```

En el directorio */root* solamente puede entrar el usuario root.

```
I have no name!:/# install -dv -m 1777 /tmp /var/tmp
install: creating directory '/tmp'
install: creating directory '/var/tmp'
I have no name!:/#
```

En estos dos directorios cualquier usuario puede entrar, crear y modificar ficheros pero no eliminar los ficheros de otros usuarios (esta última característica se activa con el "sticky bit", el bit de más peso de la mascara **1**777).

```
I have no name!:/# mkdir -pv /usr/{,local/}{bin,include,lib,sbin,src}
mkdir: created directory '/usr'
mkdir: created directory '/usr/bin'
mkdir: created directory '/usr/include'
mkdir: created directory '/usr/lib'
mkdir: created directory '/usr/sbin'
mkdir: created directory '/usr/src'
mkdir: created directory '/usr/local'
mkdir: created directory '/usr/local/bin'
mkdir: created directory '/usr/local/include'
mkdir: created directory '/usr/local/lib'
mkdir: created directory '/usr/local/sbin'
mkdir: created directory '/usr/local/src'

I have no name!:/# mkdir -pv /usr/{,local/}share/{doc,info,locale,man}
mkdir: created directory '/usr/share'
mkdir: created directory '/usr/share/doc'
```

```
mkdir: created directory '/usr/share/info'
mkdir: created directory '/usr/share/locale'
mkdir: created directory '/usr/share/man'
mkdir: created directory '/usr/local/share'
mkdir: created directory '/usr/local/share/doc'
mkdir: created directory '/usr/local/share/info'
mkdir: created directory '/usr/local/share/locale'
mkdir: created directory '/usr/local/share/man'

I have no name!:/# mkdir -v  /usr/{,local/}share/{misc,terminfo,zoneinfo}
mkdir: created directory '/usr/share/misc'
mkdir: created directory '/usr/share/terminfo'
mkdir: created directory '/usr/share/zoneinfo'
mkdir: created directory '/usr/local/share/misc'
mkdir: created directory '/usr/local/share/terminfo'
mkdir: created directory '/usr/local/share/zoneinfo'

I have no name!:/# mkdir -pv /usr/{,local/}share/man/man{1..8}
mkdir: created directory '/usr/share/man/man1'
mkdir: created directory '/usr/share/man/man2'
mkdir: created directory '/usr/share/man/man3'
mkdir: created directory '/usr/share/man/man4'
mkdir: created directory '/usr/share/man/man5'
mkdir: created directory '/usr/share/man/man6'
mkdir: created directory '/usr/share/man/man7'
mkdir: created directory '/usr/share/man/man8'
mkdir: created directory '/usr/local/share/man/man1'
mkdir: created directory '/usr/local/share/man/man2'
mkdir: created directory '/usr/local/share/man/man3'
mkdir: created directory '/usr/local/share/man/man4'
mkdir: created directory '/usr/local/share/man/man5'
mkdir: created directory '/usr/local/share/man/man6'
mkdir: created directory '/usr/local/share/man/man7'
mkdir: created directory '/usr/local/share/man/man8'
I have no name!:/#
```

```
I have no name!:/# for dir in /usr /usr/local
> do
> ln -sv share/{man,doc,info} $dir
> done
'/usr/man' -> 'share/man'
'/usr/doc' -> 'share/doc'
'/usr/info' -> 'share/info'
'/usr/local/man' -> 'share/man'
'/usr/local/doc' -> 'share/doc'
'/usr/local/info' -> 'share/info'
I have no name!:/#
```

Para cada directorio de */usr* y */usr/local* se crea un enlace simbólico en el directorio */usr/share* y */usr/local/share*, respectivamente.

```
I have no name!:/# mkdir -v /var/{log,mail,spool}
mkdir: created directory '/var/log'
mkdir: created directory '/var/mail'
mkdir: created directory '/var/spool'

I have no name!:/# ln -sv /run /var/run
'/var/run' -> '/run'

I have no name!:/# ln -sv /run/lock /var/lock
'/var/lock' -> '/run/lock'

I have no name!:/# mkdir -pv /var/{opt,cache,lib/{misc,locate},local}
mkdir: created directory '/var/opt'
mkdir: created directory '/var/cache'
mkdir: created directory '/var/lib'
mkdir: created directory '/var/lib/misc'
mkdir: created directory '/var/lib/locate'
mkdir: created directory '/var/local'
I have no name!:/#
```

Además de esta estructura de directorios es necesario que algunas aplicaciones y librerías estén disponibles para algunos programas que van a necesitarlos. Como todavía no hemos instalado nada en el actual entorno, vamos a crear unos enlaces simbólicos a estos programas dentro del entorno temporal utilizado antes de cambiar de *chroot*:

```
I have no name!:/# ln -sv /tools/bin/{bash,cat,echo,pwd,stty} /bin
```

```
'/bin/bash' -> '/tools/bin/bash'
'/bin/cat' -> '/tools/bin/cat'
'/bin/echo' -> '/tools/bin/echo'
'/bin/pwd' -> '/tools/bin/pwd'
'/bin/stty' -> '/tools/bin/stty'
I have no name!:/# ln -sv /tools/bin/perl /usr/bin
'/usr/bin/perl' -> '/tools/bin/perl'
I have no name!:/# ln -sv /tools/lib/libgcc_s.so{,.1} /usr/lib
'/usr/lib/libgcc_s.so' -> '/tools/lib/libgcc_s.so'
'/usr/lib/libgcc_s.so.1' -> '/tools/lib/libgcc_s.so.1'
I have no name!:/# ln -sv /tools/lib/libstdc++.so{,.6} /usr/lib
'/usr/lib/libstdc++.so' -> '/tools/lib/libstdc++.so'
'/usr/lib/libstdc++.so.6' -> '/tools/lib/libstdc++.so.6'
I have no name!:/# sed 's/tools/usr/' /tools/lib/libstdc++.la > /usr/lib/libstdc++.la
I have no name!:/# ln -sv bash /bin/sh
'/bin/sh' -> 'bash'
I have no name!:/#
```

El comando *sed* hace una copia del fichero */tools/lib/libstdc++.la* en */usr/lib/libstdc++.la* cambiando el valor de la variable libdir a */usr/lib* para que haga uso de la librería del entorno real.

A continuación vamos a crear los ficheros */etc/passwd* y */etc/groups* para que podamos hacer login con el usuario *root*.

El primero de ellos almacena la información de los usuarios y tendrá solamente los usuarios *root*, *bin* y *nobody*. De estos tres usuario *root* es el único que tiene un home, una shell y permite hacer login aunque todavía no le hemos establecido ninguna contraseña.

```
cat > /etc/passwd << "EOF"
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/dev/null:/bin/false
nobody:x:99:99:Unprivileged User:/dev/null:/bin/false
EOF
```

```
cat > /etc/group << "EOF"
root:x:0:
bin:x:1:
sys:x:2:
kmem:x:3:
tape:x:4:
tty:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
lp:x:9:
dialout:x:10:
audio:x:11:
video:x:12:
utmp:x:13:
usb:x:14:
cdrom:x:15:
mail:x:34:
nogroup:x:99:
EOF
```

Ahora que tenemos creados los dos ficheros con la información necesaria para los usuarios y grupos, podemos abir una nueva shell para verificar que todo está bien. Además utilizamos la opción *+h* para que la shell busque los programas en el PATH y no en su hash interno ya que nos interesa ejecutar las aplicaciones que a partir de ahora vamos a instalar y no las que bash tiene guardadas en hash.

```
I have no name!:/# exec /tools/bin/bash --login +h
root:/#
```

Algunos programas del sistema que graban información en ficheros log, los más conocidos y usados son *login*, *agetty* e *init*. Estos programas guardan la actividad de los usuarios en el sistema para saber quien, cuándo y como han intentado acceder. Los ficheros log son de mucha utilidad para la seguridad del sistema, gracias a estos ficheros se pueden detectar intrusos o mal uso del usuario. Los permisos para estos ficheros son muy estrictos y la información que almacenan no puede ser comprometida, por eso el usuario root es el único que tiene permiso de escritura:

```
root:/# touch /var/log/{btmp,lastlog,wtmp}
root:/# chgrp -v utmp /var/log/lastlog
changed group of '/var/log/lastlog' from root to utmp
root:/# chmod -v 664  /var/log/lastlog
mode of '/var/log/lastlog' changed from 0644 (rw-r--r--) to 0664 (rw-rw-r--)
root:/# chmod -v 600  /var/log/btmp
```

```
mode of '/var/log/btmp' changed from 0644 (rw-r--r--) to 0600 (rw-------)
root:/#
```

En el fichero */var/log/wtmp* se graban todos los registros de login y logout al sistema. En */var/log/lastlogin* se almacena el último login mientras que en */var/log/btmp* se graban todos los login fallidos.

4.5 Instalación de paquetes

Linux-3.10.10 API Headers

Este paquete pone a disposición de Glibc la API del kernel. Comprobamos y extraemos del *source* las cabeceras del kernel visibles para el usuario (*headers_check*). Con el segundo comando (*make headers_install*), aparte de hacer la instalación, hacemos una copia de las cabeceras en un directorio intermedio porque sino el proceso de extracción las elimina del directorio destino. A continuación buscamos y eliminamos algunos ficheros ocultos utilizados para los desarrolladores del kernel que no son necesarios. Por último, copiamos las cabeceras en el directorio `/usr/include`:

```
make headers_check
make INSTALL_HDR_PATH=dest headers_install
find dest/include \( -name .install -o -name ..install.cmd \) -delete
cp -rv dest/include/* /usr/include
```

Entre las cabeceras API instaladas podemos destacar las siguientes: ASM, DRM, MTD, RDMA, SCSI, sonido, video y Xen.

Man-pages-3.53

Este paquete contiene más de 1900 páginas con información acerca de funciones, ficheros de dispositivos y ficheros importantes de configuración. Para instalarlas solamente se tiene que ejecutar el siguiente comando:

```
make install
```

Glibc-2.18

Contiene la librería principal de C. Esta librería proporciona las rutinas básicas para alocar memoria, buscar directorios, abrir, cerrar, leer y escribir ficheros, manipular cadenas de texto, comparar patrones, operaciones aritméticas y muchas otras.

La documentación de Glibc recomienda compilar Glibc en un directorio dedicado. Para ello se ha creado el directorio `/sources/glibc-build` y se ha lanzado el script configure (el parámetro *libexecdir* configura el directorio de instalación de algunos ficheros auxiliares en la ruta `/usr/lib/glibc` en vez del directorio que viene por defecto, `/usr/libexec`) y el comando de compilación *make*:

```
../glibc-2.18/configure    \
    --prefix=/usr          \
    --disable-profile      \
    --enable-kernel=2.6.32 \
    --libexecdir=/usr/lib/glibc && make
```

Seguimos con la instalación:

```
# make install
..................
..................
if test -r /usr/include/gnu/stubs-32.h && cmp -s /sources/glibc-build/stubs.h /usr/in-
clude/gnu/stubs-32.h; \
then echo 'stubs.h unchanged'; \
else /tools/bin/install -c -m 644 /sources/glibc-build/stubs.h
/usr/include/gnu/stubs-32.h; fi
rm -f /sources/glibc-build/stubs.h
/sources/glibc-build/elf/sln /sources/glibc-build/elf/symlink.list
rm -f /sources/glibc-build/elf/symlink.list
test ! -x /sources/glibc-build/elf/ldconfig || LC_ALL=C LANGUAGE=C \
  /sources/glibc-build/elf/ldconfig  \
                           /lib /usr/lib
```

```
LD_SO=ld-linux.so.2 CC="gcc" /usr/bin/perl scripts/test-installation.pl /sources/glibc-
build/
Your new glibc installation seems to be ok.
make[1]: Leaving directory `/sources/glibc-2.18'
root:/sources/glibc-build#
```

El mensaje marcado en negrita confirma la correcta instalación de glibc.

Para que el sistema pueda responder con mensajes en varios idiomas se tienen que instalar los *locales* (ficheros de configuración regional). Con el siguiente comando se instalan todos los *locales* disponibles en el fichero *glibc-2.18/localedata/SUPPORTED*.

```
make localedata/install-locales
```

Si quisieramos instalar otros *locales* no incluidos en este fichero ejecutaríamos el comando *localedef*.

Glibc necesita el fichero */etc/nsswitch.conf*. Este contiene información que puede proceder de varias fuentes. Nos permite buscar cierto tipo de información administrativa (hosts, passwd, group, shadow, networks, etc.), especificando que fuentes queremos comprobar y en que orden se harán estas comprobaciones.

```
cat > /etc/nsswitch.conf << "EOF"
# Begin /etc/nsswitch.conf

passwd: files
group: files
shadow: files

hosts: files dns
networks: files

protocols: files
services: files
ethers: files
rpc: files

# End /etc/nsswitch.conf
EOF
```

También necesitaremos poder configurar la fecha y hora en función de la zona geográfica. Para ello instalamos los datos de *timezone*. Descomprimimos el paquete *tzdata2013d.tar.gz* y ejecutamos las siguientes instrucciones:

```
ZONEINFO=/usr/share/zoneinfo
mkdir -pv $ZONEINFO/{posix,right}

for tz in etcetera southamerica northamerica europe africa antarctica  \
          asia australasia backward pacificnew solar87 solar88 solar89 \
          systemv; do
    zic -L /dev/null   -d $ZONEINFO       -y "sh yearistype.sh" ${tz}
    zic -L /dev/null   -d $ZONEINFO/posix -y "sh yearistype.sh" ${tz}
    zic -L leapseconds -d $ZONEINFO/right -y "sh yearistype.sh" ${tz}
done

cp -v zone.tab iso3166.tab $ZONEINFO
zic -d $ZONEINFO -p Europe/Madrid
unset ZONEINFO
```

Para configurar la zona horaria ejecutamos el script *tzselect*. Primero seleccionamos el continente Europa:

```
root:/sources/glibc-build# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#? 8
```

A continuación seleccionamos el país España:

```
Please select a country.
 1) Aaland Islands    18) Greece            35) Norway
 2) Albania           19) Guernsey          36) Poland
 3) Andorra           20) Hungary           37) Portugal
 4) Austria           21) Ireland           38) Romania
 5) Belarus           22) Isle of Man       39) Russia
 6) Belgium           23) Italy             40) San Marino
 7) Bosnia & Herzegovina  24) Jersey        41) Serbia
 8) Britain (UK)      25) Latvia            42) Slovakia
 9) Bulgaria          26) Liechtenstein     43) Slovenia
10) Croatia           27) Lithuania         44) Spain
11) Czech Republic    28) Luxembourg        45) Sweden
12) Denmark           29) Macedonia         46) Switzerland
13) Estonia           30) Malta             47) Turkey
14) Finland           31) Moldova           48) Ukraine
15) France            32) Monaco            49) Vatican City
16) Germany           33) Montenegro
17) Gibraltar         34) Netherlands
#? 44
```

Seleccionamos península:

```
Please select one of the following time zone regions.
1) mainland
2) Ceuta & Melilla
3) Canary Islands
#? 1
```

Confirmamos los datos y guardamos la configuración:

```

The following information has been given:

      Spain
      mainland

Therefore TZ='Europe/Madrid' will be used.
Local time is now:  Sat Nov 16 16:00:12 CET 2013.
Universal Time is now:     Sat Nov 16 15:00:12 UTC 2013.
Is the above information OK?
1) Yes
2) No
#?
#? 1

You can make this change permanent for yourself by appending the line
      TZ='Europe/Madrid'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Europe/Madrid
root:/sources/glibc-build#
```

El fichero */etc/localtime* contiene la zona horaria que acabamos de configurar como Europe/Madrid:

```
cp -v /usr/share/zoneinfo/Europe/Madrid /etc/localtime
```

Verificamos la fecha y hora con el comando *date*:

```
root:/sources/glibc-build# date
Sat Nov 16 15:08:07 UTC 2013
root:/sources/glibc-build#
```

El cargador dinámico (*dynamic loader*) es el encargado de buscar las librerías dinámicas necesarias en tiempo de ejecución. Por defecto busca estas librerías en los directorios */lib* y */usr/lib*, pero es necesario añadir también los directorios */usr/local/lib* y */opt/lib*. El fichero que contiene los directorios de búsqueda es */etc/ld.so.conf* y lo tenemos que crear:

```
cat > /etc/ld.so.conf << "EOF"
# Begin /etc/ld.so.conf
/usr/local/lib
/opt/lib
```

```
# Add an include directory
include /etc/ld.so.conf.d/*.conf
EOF
```

Creamos el directorio */etc/ld.so.conf.d* para que el cargador dinámico también incluya los contenidos de los ficheros encontrados aquí.

```
mkdir -pv /etc/ld.so.conf.d
```

Ya tenemos instalada la librería de C pero falta configurar el *toolchain* para que los nuevos programas compilados utilicen estas librerías. Primero haremos una copia del *linker* que tenemos en el directorio */tools* y lo reemplazamos por el que hemos creado anteriormente:

```
mv -v /tools/bin/{ld,ld-old}
mv -v /tools/$(gcc -dumpmachine)/bin/{ld,ld-old}
mv -v /tools/bin/{ld-new,ld}
ln -sv /tools/bin/ld /tools/$(gcc -dumpmachine)/bin/ld
```

Solamente nos falta configurar GCC para que apunte al nuevo *dynamic linker*. Para hacerlo eliminamos todas las instancias que apuntan al directorio */tools* con la ayuda del comando *sed*:

```
gcc -dumpspecs | sed -e 's@/tools@@g'                   \
    -e '/\*startfile_prefix_spec:/{n;s@.*@/usr/lib/ @}' \
    -e '/\*cpp:/{n;s@$@ -isystem /usr/include@}' >      \
    `dirname $(gcc –print-libgcc-file-name)`/specs
```

Validamos la correcta instalación de Gcc:

1. Funciones básicas (*compile* y *link*):

```
root:/sources/glibc-build# echo 'main(){}' > dummy.c
root:/sources/glibc-build# cc dummy.c -v -Wl,--verbose &> dummy.log
root:/sources/glibc-build# readelf -l a.out | grep ': /lib'
      [Requesting program interpreter: /lib/ld-linux.so.2]
```

2. Ficheros *startfiles* correctos:

```
root:/sources/glibc-build# grep -o '/usr/lib.*/crt[1in].*succeeded' dummy.log
/usr/lib/crt1.o succeeded
/usr/lib/crti.o succeeded
/usr/lib/crtn.o succeeded
```

3. Ficheros de cabecera:

```
root:/sources/glibc-build# grep -B1 '^ /usr/include' dummy.log
#include <...> search starts here:
 /usr/include
```

4. Directorios de búsqueda del *linker:*

```
root:/sources/glibc-build# grep 'SEARCH.*/usr/lib' dummy.log |sed 's|; |\n|g'
SEARCH_DIR("/usr/lib")
SEARCH_DIR("/lib");
```

5. Librería C (libc):

```
root:/sources/glibc-build# grep "/lib.*/libc.so.6 " dummy.log
attempt to open /lib/libc.so.6 succeeded
```

6. *Dynamic linker*:

```
root:/sources/glibc-build# grep found dummy.log
found ld-linux.so.2 at /lib/ld-linux.so.2
```

Las pruebas han validado la correcta instalación de Gcc.

Zlib-1.2.8

El paquete Zlib contiene rutinas de compresión y descompresión utilizadas para algunos programas.

```
./configure --prefix=/usr && make && make install
```

File-5.14

El paquete File contiene una utilidad con la que se puede determinar el tipo de un fichero pasado como parámetro.

```
./configure --prefix=/usr && make && make install
```

Binutils-2.23.2

El paquete Binutils contiene un *linker,* ensamblador y otras herramientas para manipular ficheros objeto.

Creamos el directorio `../binutils-build` para compilar Binutils fuera del directorio que contiene los ficheros *source.* Desde el directorio creado preparamos y compilamos Binutils:

```
root:/sources/binutils-build# ../binutils-2.23.2/configure  \
      --prefix=/usr –enable-shared                           \
      make tooldir=/usr                                      \
      make check                                             \
      make install                                           \
      make tooldir=/usr install
```

Instalamos la cabecera *libiberty* necesaria para algunos paquetes:

```
cp -v ../binutils-2.23.2/include/libiberty.h /usr/include
```

GMP-5.1.2

El paquete GMP contiene librerías necesarias para las opreaciones aritméticas. Antes de preparar la compilación forzamos el script para 32 bits asignándo el valor *ABI=32*. Además, habilitamos el soporte para C++:

```
ABI=32 ./configure --prefix=/usr --enable-cxx && make
```

Antes de continuar con la instalación es muy recomendable comprobar que la compilación ha sido satisfactoria. Para ello ejecutamos el comando *make check* guardando la información de la salida en el fichero *gmp-check-log*:

```
make check 2>&1 | tee gmp-check-log
```

Con el siguiente comando nos aseguramos que todos los 185 tests han sido satisfactorios:

```
root:/sources/gmp-5.1.2# awk '/tests passed/{total+=$2} ; END{print total}' gmp-check-
log
185
root:/sources/gmp-5.1.2#
```

Finalmente, instalamos la documentación del paquete:

```
mkdir -v /usr/share/doc/gmp-5.1.2
cp    -v doc/{isa_abi_headache,configuration} doc/*.html \
         /usr/share/doc/gmp-5.1.2
```

MPFR-3.1.2

El paquete MPFR contiene funciones matemáticas de precisión múltiple.

```
./configure  --prefix=/usr        \
             --enable-thread-safe \
             --docdir=/usr/share/doc/mpfr-3.1.2 && make && make install
```

Finalmente instalamos la documentación:

```
make html
make install-html
```

MPC-1.0.1

El paquete MPC contiene una librería para operaciones aritméticas con números complejos de alta precisión y redondeo correcto del resultado.

```
./configure --prefix=/usr && make && make install
```

GCC-4.8.1

El paquete GCC contiene la colección del compilador GNU que incluye los compiladores C y C++.

Como en ocasiones anteriores, creamos un directorio independiente gcc-build para compilar el paquete. Preparamos la compilación ejecutando el script *configure*. El parámetro –disable-install-libiberty evita instalar la versión propia del GCC, ya hemos instalado esta librería con la versión de libiberty proporcionada por el paquete Binutils. También configuramos GCC para enlazar con la librería Zlib ya instalada en el sistema:

```
../gcc-4.8.1/configure --prefix=/usr              \
                       --libexecdir=/usr/lib      \
                       --enable-shared            \
                       --enable-threads=posix     \
                       --enable-__cxa_atexit      \
                       --enable-clocale=gnu       \
                       --enable-languages=c,c++   \
                       --disable-multilib         \
                       --disable-bootstrap        \
                       --disable-install-libiberty \
                       --with-system-zlib && make && make install
```

Sed-4.2.2

El paquete Sed contiene un editor de flujo de texto. Es muy útil para filtrar y transformar texto utilizando patrones de búsqueda así como sustituir cadenas de texto por otro texto. Es muy potente y puede trabajar con expresiones regulares.

Preparamos la compilación e instalamos el paquete y la documentación:

```
./configure --prefix=/usr --bindir=/bin --htmldir=/usr/share/doc/sed-4.2.2
make
make html
make install
make -C doc install-html
```

La opción *htmldir* especifica donde instalar la documentación.

Bzip2-1.0.6

El paquete Bzip2 contiene programas para comprimir y descomprimir ficheros.

Primero aplicamos el parche que nos permite instalar la documentación del paquete:

```
patch -Np1 -i ../bzip2-1.0.6-install_docs-1.patch
```

Generamos un fichero *Makefile* diferente para que Bzip2 cree la librería compartida *libbz2.so* y después instalamos el paquete utilizando este fichero *Makefile*:

```
make -f Makefile-libbz2_so
make
make PREFIX=/usr install
```

Por último instalamos el binario bzip2 en el directorio /bin y creamos los enlaces simbólicos necesarios:

```
cp -v bzip2-shared /bin/bzip2
cp -av libbz2.so* /lib
ln -sv ../../lib/libbz2.so.1.0 /usr/lib/libbz2.so
rm -v /usr/bin/{bunzip2,bzcat,bzip2}
ln -sv bzip2 /bin/bunzip2
ln -sv bzip2 /bin/bzcat
```

Pkg-config-0.28

El paquete pkg-config contiene una herramienta que se utiliza para indicar los ficheros *include* y las librerías a compilar durante la ejecución de los scripts *configure* y *make*. Al ejecutar el programa *pkg-config* devuelve la *meta* información de la librería o el paquete pasados como parámetro.

```
./configure --prefix=/usr         \
            --with-internal-glib  \
            --disable-host-tool   \
            --docdir=/usr/share/doc/pkg-config-0.28 && make && make install
```

El parámetro *with-internal-glib* permite utilizar la versión de Glib interna ya que no está instalada todavía en nuestro sistema. La opción *disable-host-tool* deshabilita la creación innecesaria de un enlace al programa pkg-config.

Ncurses-5.9

El paquete Ncurses contiene unas librerías que proveen una API que permite a los programadores escribir interfaces basadas en texto, también llamadas TUI (Text User Interface). Optimiza el refresco de pantalla, cosa que permite reducir la latencia experimentada cuando se usan en shells remotas.

```
./configure --prefix=/usr           \
            --mandir=/usr/share/man \
            --with-shared           \
            --without-debug         \
            --enable-pc-files       \
            --enable-widec && make && make install
```

Las opción *enable-widec* prepara la compilación de las librerías de caracteres largos (*libncursesw.so.5.9*) en lugar de las librerías normales (*libncurses.so.5.9*), mientras que la opción *enable-pc-files* activa la creación e instalación de los ficheros con extensión *.pc* para el programa *pkg-config*.

Movemos las librerías compartidas en el directorio donde deberían estar (*/usr*) y recreamos los enlaces simbólicos para que apunten a la nueva ubicación:

```
mv -v /usr/lib/libncursesw.so.5* /lib
ln -sfv ../../lib/libncursesw.so.5 /usr/lib/libncursesw.so
```

Finalmente instalamos la documentación:

```
mkdir -v       /usr/share/doc/ncurses-5.9
cp -v -R doc/* /usr/share/doc/ncurses-5.9
```

Shadow-4.1.5.1

El paquete Shadow contiene programas para manipular contraseñas de forma segura. Es un sistema de contraseñas mejorado respecto al estándar previo de almacenaje de contraseñas Unix y Linux. Lo que hace es borrar las claves secretas codificadas del fichero */etc/passwd* (que necesariamente es legible para todos) y las sitúa en el fichero */etc/shadow* el cual solo puede leer el usuario root.

Entre las ventajas de las contraseñas "Shadow" destacan el cifrado más robusto, la obtención de información acerca de la vigencia de las contraseñas, el poder controlar cuanto tiempo puede permanecer sin cambios una contraseña antes de que el usuario sea obligado a cambiarla y la capacidad de usar el fichero */etc/login.defs* para reforzar la política de seguridad, especialmente lo concerniente a la vigencia de contraseñas.

Desactivamos la instalación del programa groups y los manuales ya que el paquete Coreutils dispone de una versión mejorada:

```
sed -i 's/groups$(EXEEXT) //' src/Makefile.in
find man -name Makefile.in -exec sed -i 's/groups\.1 / /' {} \;
```

Cambiamos el método de encriptación *crypt* que instala por defecto por otro más seguro que utiliza la encriptación de contraseñas SHA-512, que tambíen permite contraseñas más largas de 8 caracteres. También cambiaremos la ubicación obsoleta de los buzones de los usuarios que Shadow utiliza por defecto, /var/spool/mail, por la que se utiliza actualmente, /var/mail:

```
sed -i -e 's@#ENCRYPT_METHOD DES@ENCRYPT_METHOD SHA512@' \
       -e 's@/var/spool/mail@/var/mail@' etc/login.defs
```

Una vez realizadas estas modificaciones ya podemos compilar e instalar Shadow:

```
./configure --sysconfdir=/etc && make && make install
```

Movemos el binario passwd en el directorio correcto:

```
mv -v /usr/bin/passwd /bin
```

Entre las posibilidades que ofrece el paquete Shadow destacamos la creación, modificación y eliminación de usuarios y grupos, configurar y cambiar sus contraseña y otras tareas de administración.

Activamos las contraseñas *shadow* para usuarios y grupos:

```
pwconf
grpconv
```

El programa useradd es el utilizado para agregar usuarios al sistema. Si se ejecuta sin parámetros creará un grupo que tendrá el mismo nombre que el usuario. Asignará el primer número disponible, empezando por 1000, para los nuevos UID y GID. El fichero */etc/default/useradd* contiene las configuraciones por defecto al ejecutar useradd.

Ahora que tenemos instalado el paquete Shadow, podemos cambiar la contraseña para el usuario root:

```
root:/sources/shadow-4.1.5.1# passwd root
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
passwd: password changed.
root:/sources/shadow-4.1.5.1#
```

Util-linux-2.23.2

El paquete Util-linux contiene varios programas útiles para manipular sistemas de ficheros, consolas, particiones y mensajes. Entre ellos los más destacados son: *dmesg, fdisk, fsck, kill, mkfs, more, mount, rename, renice, umount, whereis.*

Ejecutamos el script configure con los parámetros *disable-su, disable-sulogin* y *disable-login* para no instalar los programas *su, sulogin* y *login* que ya los ofrece el paquete Shadow.

```
./configure --disable-su --disable-sulogin --disable-login && make && make install
```

Psmisc-22.20

El paquete Psmisc contiene programas para mostrar información acerca de procesos que se encuentran en ejecución. Estos son: *fuser, killall, peekfd, prtstat, pstree.*

```
./configure --prefix=/usr && make && make install
```

Procps-ng-3.3.8

El paquete Procps-ng contiene programas para monitorizar procesos. Entre ellos se encuentran: *free, ps, top, uptime, vmstat, w, watch.*

Ejecutamos el script *configure* son los parámetros *disable-skill* y *disable-kill* para deshabilitar la instalación de los programas obsoletos *skill* y *snice* y también el programa *kill* ya instalado con el paquete *util-linux*:

```
./configure --prefix=/usr                          \
            --exec-prefix=                         \
            --libdir=/usr/lib                      \
            --docdir=/usr/share/doc/procps-ng-3.3.8 \
            --disable-static                       \
            --disable-skill                        \
            --disable-kill && make && make install
```

E2fsprogs-1.42.8

El paquete E2fsprogs contiene utilidades para manipular sistemas de ficheros de tipo *ext2*. También soporta los tipos *ext3* y *ext4*. Entre los programas que instala destacamos: *debugfs, dumpe2fs, fsck.ext2, fsck.ext3, fsck.ext4, lsattr, mke2fs, mkfs.ext2, mkfs.ext3, mkfs.ext4, resize2fs, tune2fs.*

La documetación de E2fsprogs recomienda la compilación de este paquete desde un subdirectorio del directorio *source*. Creamos el directorio *build* y desde allí ejecutamos el script *configure*. El parámetro *enable-elf-shlibs* crea unas librerías compartidas necesarias para algunos programas de este paquete. Desactivamos la instalación de los programas *libuuid, libblkid,* demonio *uuidd* y *fsck* porque ya los hemos instalado con el paquete Util-linux.

```
../configure --prefix=/usr         \
             --enable-elf-shlibs   \
             --disable-libblkid    \
             --disable-libuuid     \
             --disable-uuidd       \
             --disable-fsck && make && make install & make install-libs
```

Se han instalado también las cabeceras y librerías estáticas.

Coreutils-8.21

El paquete Coreutils contiene utilidades para mostrar y configurar características básicas del sistema. Algunas de estas utilidades son: *basename, cat, chgrp, chmod, chown, chroot, cp, cut, date, df, dir, du, echo, env, groups, head, id, ln, ls, mkdir, mv, nice, nohup, pwd, rm, rmdir, sleep, sort, tail, tee, touch, tr, uname, usrs, wc, who, whoami.*

Preparamos el script *configure* con la opción *enable-no-install-program* para evitar la instalación de los binarios *kill* y *uptime* que ya tenemos instalados. A continuación compilamos e instalamos el paquete:

```
FORCE_UNSAFE_CONFIGURE=1 ./configure \
            --prefix=/usr            \
            --libexecdir=/usr/lib    \
            --enable-no-install-program=kill,uptime && make && make install
```

Para cumplir con el estandar de FHS movemos los binarios de los programas en la ubicación correcta:

```
mv -v /usr/bin/{cat,chgrp,chmod,chown,cp,date,dd,df,echo} /bin
mv -v /usr/bin/{false,ln,ls,mkdir,mknod,mv,pwd,rm} /bin
mv -v /usr/bin/{rmdir,stty,sync,true,uname,test,[} /bin
mv -v /usr/bin/chroot /usr/sbin
mv -v /usr/share/man/man1/chroot.1 /usr/share/man/man8/chroot.8
sed -i s/\"1\"/\"8\"/1 /usr/share/man/man8/chroot.8
```

Iana-Etc-2.30

El paquete Iana-Etc proporciona datos para los servicio (*/etc/services*) y protocolos (*/etc/protocols*) de red.

```
make && make install
```

M4-1.4.16

El paquete M4 contiene un procesador de macros.

```
./configure --prefix=/usr && make && make install
```

Flex-2.5.37

El paquete Flex contiene una utilidad para generar programas que reconocen patrones de texto.

```
./configure --prefix=/usr \
            --docdir=/usr/share/doc/flex-2.5.37
make
make install
```

Creamos un enlace simbólico en el directorio /usr/lib para aquellos programas que buscan aquí la librería *lex*:

```
ln -sv libfl.a /usr/lib/libl.a
```

Bison-3.0

El paquete Bison contiene un generador de analizadores sintácticos. Instala dos programas: *bison*, que genera, a partir de una serie de reglas, un programa para analizar la estructura de ficheros de texto; *yacc*, un envoltorio para *bison*, destinado a los programas que todavía llaman a *yacc* en lugar de a *bison*. El programa *yacc* invoca a *bison* con la opción *-y*. Bison es un sustituto de Yacc (Yet Another Compiler Compiler, Otro Compilador de Compiladores).

```
./configure –prefix=/usr && make && make install
```

Grep-2.14

El paquete Grep contiene programas para buscar dentro de ficheros. Instala tres programas: *egrep*, muestra las líneas que coincidan con una expresión regular extendida; *fgrep*, muestra las líneas que coincidan con una lista de cadenas fijas; *grep*, muestra las líneas que coincidan con una expresión regular básica.

```
./configure --prefix=/usr --bindir=/bin && make && make install
```

Readline-6.2

El paquete Readline contiene un conjunto de librerías que ofrecen edición de la línea de comandos y capacidades de historial. Instala dos librearías: *libhistory*, proporciona una interfaz de usuario consistente para la rellamada de líneas de historial; *libreadline*, asiste en la consistencia de la interfaz de usuario entre programas discrecionales que necesitan suministrar una interfaz de línea de comandos.

```
./configure --prefix=/usr –libdir=/lib
make SHLIB_LIBS=-lncurses
make install
```

La opción *SHLIB_LIBS=-lncurses* fuerza a Readline a enlazarse contra la libería *libncurses*.

Movemos las librerías estáticas a la ubicación correcta:

```
mv -v /lib/lib{readline,history}.a /usr/lib
```

Ahora eliminamos los ficheros con extensión *.so* del directorio */lib* y los reenlazamos a */usr/lib*:

```
rm -v /lib/lib{readline,history}.so
ln -sfv ../../lib/libreadline.so.6 /usr/lib/libreadline.so
ln -sfv ../../lib/libhistory.so.6 /usr/lib/libhistory.so
```

Bash-4.2

El paquete Bash contiene la “Bourne-Again SHell”. Instala tres programas: *bash*, un intérprete de comandos ampliamente usado, realiza muchos tipos de expansiones y sustituciones en una línea de comandos dada antes de ejecutarla, lo que hace de este intérprete una herramienta poderosa; *bashbug*, un guión que ayuda al usuario en la composición y envío de informes de errores relacionados con *bash*, usando un formato estándar; *sh*, enlace simbólico al programa *bash*.

Aplicamos correcciones para varios fallos descubiertos dede la publicación inicial de esta versión:

```
patch -Np1 -i ../bash-4.2-fixes-12.patch
```

Preparamos la compilación con la opción *htmldir* para configurar el directorio donde instalar la documentación en formato HTML, y la opción *with-installed-readline* para utilizar la librería que ya tenemos instalada. A continuación compilamos e instalamos el paquete:

```
./configure --prefix=/usr                     \
            --bindir=/bin                     \
            --htmldir=/usr/share/doc/bash-4.2 \
```

```
            --without-bash-malloc             \
            --with-installed-readline && make && make install
```

Ejecutamos el *bash* recién instalado reemplazando el que estamos utilizando actualmente:

```
exec /bin/bash --login +h
```

Bc-1.06.95

El paquete Bc contiene un lenguage de procesamiento númerico con precisión arbitraria. Instala los programas *bc* (una calculadora en línea de comandos) y *dc* (una calculadora polaca inversa en línea de comandos).

Volvemos a utilizar la opción *with-readline* para utilizar la librería ya instalada:

```
./configure --prefix=/usr --with-readline && make && make install
```

Libtool-2.4.2

El paquete Libtool contiene el guión de GNU para soporte genérico de librerías. Oculta la complejidad del uso de librerías compartidas tras una interfaz consistente y portable. El programa *libtool* instalado proporciona servicios de soporte generalizados para la compilación de librerías mientras que *libtoolize* proporciona una forma estándar de añadir soporte para *libtool* a un paquete.

```
./configure --prefix=/usr && make && make install
```

GDBM-1.10

El paquete GDBM contiene *GNU Database Manager*. Es una base de datos en formato de fichero de disco que almacena claves y datos en ficheros separados. La información actual de cualquier registro almacenado se indexa con una clave única, lo que permite recuperarlo en menos tiempo de lo que se tardaría si fuera almacenado en un fichero de texto. Instala la librería *libgdbm* que contiene funciones para manipular la base de datos.

```
./configure --prefix=/usr && make && make install
```

Inetutils-1.9.1

El paquete Inetutils contiene programas para trabajo básico en red. Los programas que instala son: *ftp, hostname, ping, ping6, rcp, rexec, rlogin, rsh, talk, telnet, tftp, traceroute*.

```
./configure --prefix=/usr  \
    --libexecdir=/usr/sbin \
    --localstatedir=/var   \
    --disable-ifconfig     \
    --disable-logger       \
    --disable-syslogd      \
    --disable-whois        \
    --disable-servers && make && make install
```

Utilizamos el switch *disable-** para evitar la instalación de los programas *ifconfig, logger, syslogd, whois* y *servers* ya que se instalan desde otros paquetes por ser versiones mejoradas.

Perl-5.18.1

El paquete Perl contiene el Lenguaje Práctico de Extracción e Informe.

Creamos primero un fichero */etc/hosts* básico que es referenciado por uno de los ficheros de configuración de Perl:

```
echo "127.0.0.1 localhost $(hostname)" > /etc/hosts
```

Por defecto Perl utiliza una copia interna del módulo Zlib, ejecutamos este comando *sed* para evitar esto y utilizar la librería Zlib instalada:

```
ed -i -e "s|BUILD_ZLIB\s*= True|BUILD_ZLIB = False|"           \
      -e "s|INCLUDE\s*= ./zlib-src|INCLUDE    = /usr/include|" \
      -e "s|LIB\s*= ./zlib-src|LIB        = /usr/lib|"         \
   cpan/Compress-Raw-Zlib/config.in
```

Configuramos, compilamos e instalamos el paquete:

```
sh Configure -des -Dprefix=/usr                 \
                  -Dvendorprefix=/usr           \
                  -Dman1dir=/usr/share/man/man1 \
                  -Dman3dir=/usr/share/man/man3 \
                  -Dpager="/usr/bin/less -isR"  \
                  -Duseshrplib && make && make install
```

La opción *Dvendorprefix* asegura que *perl* indica donde deben instalar los paquetes sus módulos. *Dpager* corrige un error en el modo en que *perldoc* invoca al programa *less*. Las opciones *Dman1dir* y *Dman3dir* fuerza la instalación de las páginas de manual de Perl. Por último, el parámetro *Duseshrplib* compila una librería compartida necesaria para algunos módulos de Perl.

Autoconf-2.69

El paquete Autoconf contiene programas para generar guiones del intérprete de comandos que pueden configurar automáticamente el código fuente. Instala los programas: *autoconf, autoheader, autom4te, autoreconf, autoscan, autoupdate, ifnames.*

```
./configure --prefix=/usr && make && make install
```

Automake-1.14

El paquete Automake contiene programas para generar Makefiles que se utilizan con Autoconf. Instala los siguientes programas: *acinstall, aclocal, aclocal-1.14, automake, automake-1.14, compile, config.guess, config.sub, depcomp, install-sh, mdate-sh, missing, mkinstalldirs, py-compile, ylwrap.*

```
./configure --prefix=/usr --docdir=/usr/share/doc/automake-1.14 && make && make install
```

Diffutils-3.3

El paquete Diffutils contiene programas que muestran las diferencias entre ficheros o directorios. Instala los siguientes programas: *cmp, diff, diff3, and sdiff.*

```
./configure --prefix=/usr && make && make install
```

Gawk-4.1.0

El paquete Gawk contiene programas para manipular ficheros de texto. Instala los siguientes programas: *awk* (enlace a *gawk*), *gawk, gawk-4.1.0, igawk.*

```
./configure --prefix=/usr --libexecdir=/usr/lib && make && make install
```

Finalmente instalamos la documentación:

```
mkdir -v /usr/share/doc/gawk-4.1.0
cp    -v doc/{awkforai.txt,*.{eps,pdf,jpg}} /usr/share/doc/gawk-4.1.0
```

Findutils-4.4.2

El paquete Findutils contiene programas para encontrar ficheros. Se suministran estos programas para hacer búsquedas recursivas en un árbol de directorios, y para crear, mantener y consultar una base de datos (más rápida que la búsqueda recursiva, pero imprecisa si la base de datos no se ha actualizado recientemente). Instala estos programas: *bigram, code, find, frcode, locate, oldfind, updatedb, xargs.*

```
./configure --prefix=/usr                   \
            --libexecdir=/usr/lib/findutils \
            --localstatedir=/var/lib/locate && make && make install
```

El parámetro *localstatedir* cambia la ubicación de la base de datos de *locale* a /var/lib/locate para cumplir con el estándar FHS.

Gettext-0.18.3

El paquete Gettext contiene utilidades para la internacionalización y localización. Esto permite a los programas compilarse con Soporte de Lenguaje Nativo (NLS), lo que les permite mostrar mensajes en el idioma nativo del usuario.

```
./configure --prefix=/usr \
            --docdir=/usr/share/doc/gettext-0.18.3 && make && make install
```

Groff-1.22.2

El paquete Groff contiene programas para procesar y formatear texto.

```
./configure --prefix=/usr && make && make install
```

Xz-5.0.5

El paquete Xz contiene programas para comprimir y descomprimir ficheros. Comprimir ficheros de texto con *xz* ofrece un mayor porcentaje de compresión que los comandos *gzip* o *bzip2*.

```
./configure --prefix=/usr --libdir=/lib --docdir=/usr/share/doc/xz-5.0.5
make
make pkgconfigdir=/usr/lib/pkgconfig install
```

GRUB-2.00

El paquete GRUB contiene *Grand Unified Bootloader*. Es un gestor de arranque múltiple, desarrollado por el proyecto GNU que se usa comúnmente para iniciar uno, dos o más sistemas operativos instalados en un mismo equipo.

```
./configure --prefix=/usr          \
            --sysconfdir=/etc      \
            --disable-grub-emu-usb \
            --disable-efiemu       \
            --disable-werror && make && make install
```

Más adelante vamos a configurar el sistema para utilizar GRUB como gestor de arranque.

Less-458

El paquete Less contiene un visor de ficheros de texto.

Preparamos la compilación con el parámetro *sysconfdir* para decirle al programa instalado que busque dentro del directorio */etc* por ficheros de configuración. A continuación compilamos e instalamos el paquete:

```
./configure --prefix=/usr --sysconfdir=/etc && make && make install
```

Gzip-1.6

El paquete Gzip contiene programas para comprimir y descomprimir ficheros.

```
./configure --prefix=/usr --bindir=/bin && make && make install
```

IPRoute2-3.10.0

El paquete IPRoute2 contiene programas para el trabajo básico y avanzado en redes basadas en IPV4.

Este paquete no tiene el script *configure*, por lo tanto compilamos e instalamos con la opción *DESTDIR* en blanco para asegurarse que los binarios se instalan en el directorio correcto:

```
make DESTDIR=
make DESTDIR=              \
     MANDIR=/usr/share/man \
     DOCDIR=/usr/share/doc/iproute2-3.10.0 install
```

Kbd-1.15.5

El paquete Kbd contiene ficheros de mapas de teclado y utilidades para el teclado.

El comportamiento de las teclas Retroceso y Borrado no es homogeneo en los diferentes maspas de teclado del paquete Kbd. El siguiente parche corrige este problema para los mapas de teclado i386:

```
patch -Np1 -i ../kbd-1.15.5-backspace-1.patch
```

Preparamos, compilamos e instalamos el paquete (el parámetro *disable-vlock* evita la instalación de la utilidad *vlock* ya que un es necesaria):

```
./configure --prefix=/usr –disable-vlock && make && make install
```

Finalmente copiamos los ficheros de documentación:

```
mkdir -v       /usr/share/doc/kbd-1.15.5
cp -R -v doc/* /usr/share/doc/kbd-1.15.5
```

Kmod-14

El paquete Kmod contiene librerías y utilidades para cargar módulos de kernel. Entre los programas instalados destacamos: *lsmod, modprobe, modinfo*.

```
./configure --prefix=/usr       \
            --bindir=/bin       \
            --libdir=/lib       \
            --sysconfdir=/etc   \
            --disable-manpages  \
            --with-xz           \
            --with-zlib && make && make install
```

Las opciones *with-** habilita el manejo de módulo de kernel comprimidos.

Finalmente instalamos el paquete y creamos enlaces simbólicos para la compatibilidad con *Module-Init-Tools*, que representa el paquete que manejaba anteriormente los módulos de kernel en Linux:

```
make pkgconfigdir=/usr/lib/pkgconfig install

for target in depmod insmod modinfo modprobe rmmod; do
  ln -sv ../bin/kmod /sbin/$target
done

ln -sv kmod /bin/lsmod
```

Libpipeline-1.2.4

El paquete Libpipeline contiene una librería para manipular tuberías de subprocesos de forma flexible y fácil.

```
PKG_CONFIG_PATH=/tools/lib/pkgconfig ./configure --prefix=/usr && make && make install
```

Make-3.82

El paquete Make contiene un programa (*make*) para compilar paquetes.

Antes de compilar aplicamos un parche que soluciona varios problemas desde que se publicó la versión inicial:

```
patch -Np1 -i ../make-3.82-upstream_fixes-3.patch
```

Compilamos e instalamos el paquete:

```
./configure –prefix=/usr && make && make install
```

Man-DB-2.6.5

El paquete Man-DB contiene programas para encontrar y visualizar páginas de manual.

```
./configure --prefix=/usr                        \
            --libexecdir=/usr/lib                \
            --docdir=/usr/share/doc/man-db-2.6.5 \
            --sysconfdir=/etc                    \
            --disable-setuid                     \
            --with-browser=/usr/bin/lynx         \
            --with-vgrind=/usr/bin/vgrind        \
            --with-grap=/usr/bin/grap && make && make install
```

Patch-2.7.1

El paquete Patch contiene un programa para modificar o crear ficheros mediante la aplicación de un fichero “parche” creado normalmente con el programa *diff*.

```
./configure --prefix=/usr && make && make install
```

El programa *patch* instalado modifica ficheros acorde a un fichero “parche”. Un parche normalmente es un listado de diferencias creado con el programa *diff*. Aplicando estas diferencias sobre los ficheros originales, el programa *patch* crea las versiones parcheadas.

Sysklogd-1.5

El paquete Sysklogd contiene programas para registrar los mensajes del sistema, como aquellos generados por el núcleo cuando sucede algo inusual.

Compilamos e instalamos el paquete (Sysklogd no viene con el script *configure*):

```
make && make BINDIR=/sbin install
```

El paquete Sysklogd implementa dos demonios. El demonio *syslogd* es una versión mejorada del estándar *Berkeley utility program*. Este demonio es el responsable de proporcionar logs de mensajes recibidos desde aplicaciones del sistema local así como de sistemas remotos. El demonio *klogd* escucha los mensajes del kernel y es el responsable de priorizar y procesar mensajes del sistema operativo. Este demonio puede ejecutarse como cliente de *syslogd* o como un programa independiente.

El fichero de configuración de este paquete es */etc/syslog.conf*:

```
cat > /etc/syslog.conf << "EOF"
# Begin /etc/syslog.conf

auth,authpriv.* -/var/log/auth.log
*.*;auth,authpriv.none -/var/log/sys.log
```

```
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
*.emerg *

# End /etc/syslog.conf
EOF
```

Sysvinit-2.88dsf

El paquete Sysvinit contiene programas para controlar el arranque, ejecución y cierre del sistema.

Evitamos la instalación de los programas *wall, mountpoint* y *utmpdump* de este paquete porque tenemos otras versiones más actualizadas instaladas por el paquete Util-linux:

```
sed -i -e '/utmpdump/d' \
       -e '/mountpoint/d' src/Makefile
```

Compilamos e instalamos el paquete:

```
make -C src && make -C src install
```

Este paquete instala los siguientes programas: *bootlogd, fstab-decode, halt, init, killall5, last, lastb (*enlace a *last), mesg, pidof (*enlace a *killall5), poweroff (*enlace a *halt), reboot (*enlace a *halt), runlevel, shutdown, sulogin* y *telinit (*enlace a *init).*

Tar-1.26

El paquete Tar contiene un programa de archivado. Instala dos programas: *tar*, para crear, extraer ficheros, listar contenido de archivos, y *rmt*, para manipular remotamente unidades de cinta magnética.

Con el siguiente parche añadimos un programa que genera una página de manual para *tar*:

```
patch -Np1 -i ../tar-1.26-manpage-1.patch
```

Preparamos tar para la compilación, lo compilamos y luego ejecutamos la instalación:

```
./configure --prefix=/usr \
            --bindir=/bin \
            --libexecdir=/usr/sbin
make
make install
make -C doc install-html docdir=/usr/share/doc/tar-1.26
```

Finalmente generamos la página de manual y la copiamos en la ubicación correcta:

```
perl tarman > /usr/share/man/man1/tar.1
```

Texinfo-5.1

El paquete Texinfo contiene programas para leer, escribir y convertir páginas de información.

```
./configure --prefix=/usr && make && make install
```

El programa *info* instalado lee páginas info, que son similares a las páginas de manual, pero tienden a ser más profundos que una simple explicación de las opciones de un programa.

Udev-206

El paquete Udev contiene programas para la creación dinámica de nodos de dispositivos.

Compilamos el paquete:

```
make -f udev-lfs-206-1/Makefile.lfs
```

Instalamos el paquete:

```
make -f udev-lfs-206-1/Makefile.lfs install
```

Inicializamos la base de datos de hardware:

```
build/udevadm hwdb –update
```

Finalmente configuramos las reglas persistentes de red *udev*:

```
bash udev-lfs-206-1/init-net-rules.sh
```

El paquete Udev instala los directorio */etc/udev, /lib/udev, /lib/firmware, /usr/share/doc/udev* así como la librería *libudev.so* y los programas *accelerometer, ata_id, cdrom_id, collect, mtd_probe, scsi_id, v4l_id, udevadm* y *udevd*.

Vim-7.4

El paquete Vim contiene un poderoso editor de texto siendo el más popular a día de hoy en los sistemas basados en Unix.

Para empezar, cambiamos la ubicación por defecto del fichero de configuración *vimrc* al directorio /etc:

```
echo '#define SYS_VIMRC_FILE "/etc/vimrc"' >> src/feature.h
```

Instalamos el paquete tras compilarlo:

```
./configure --prefix=/usr && make && make install
```

Creamos el enlace simbólico *vi* y sus páginas de manual:

```
ln -sv vim /usr/bin/vi
for L in "" fr it pl ru; do
    ln -sv vim.1 /usr/share/man/$L/man1/vi.1
done
```

Por defecto, la documentación de Vim se instala en /usr/share/vim. El siguiente enlace permite que la documentación sea accesible mediante /usr/share/doc/vim-7.4, haciendolo consistente con la localización de la documentación del resto de paquetes:

```
ln -sv ../vim/vim74/doc /usr/share/doc/vim-7.4
```

4.6 Configuración del sistema

Configurar la red

Nuestro sistema solo dispone de una tarjeta de red que la llamaremos *eth0*. Creamos el fichero de configuración para el dispositivo *eth0*. A la hora de activar o desactivar una tarjeta de red, el sistema buscará el fichero de configuración para cada una de ellas en la ruta /etc/sysconfig. Aquí deberían encontrarse los ficheros de configuración con la nomencaltura *ifconfig.eth0* para el dispositivo *eth0*. Si tuvieramos más dispositivos, por ejemplo una tarjeta inalámbrica *wlan0*, el fichero de configuración debería tener el nombre *ifconfig.wlan0*. Nuestro fichero **`/etc/sysconfig/ifconfig.eth0`** contiene la información básica necesaria para funcionar:

```
ONBOOT=yes
IFACE=eth0
SERVICE=ipv4-static
IP=192.168.1.50
GATEWAY=192.168.1.1
PREFIX=24
BROADCAST=192.168.1.255
```

La variable *ONBOOT* tiene que tener el valor *yes* para que el dispositivo *eth0* se active al arrancar el sistema. La variable *IFACE* contiene el nombre del dispositivo (en este caso *eth0*). *SERVICE* indica el modo de configuración de la dirección IP. En este caso vamos a utilizar una dirección IP estática pero también se puede configurar por DHCP, en este caso el valor de esta variable sería *ipv4-dynamic*. La variable *IP* contiene la dirección IP que le asignamos a este dispositivo, *PREFIX* establece la máscara de la red y *BROADCAST* es la dirección broadcast de la red.

Para que el sistema pueda conectarse a Internet tiene que conseguir resolver los nombres de dominios. Para ellos creamos el fichero **/etc/resolv.conf** donde configuramos los servidores DNS (en este caso configuramos dos servidores DNS públicos de Google):

```
# Begin /etc/resolv.conf

nameserver 8.8.8.8
nameserver 8.8.4.4

# End /etc/resolv.conf
```

El fichero **/etc/hosts** es necesario si tenemos configurada alguna tarjeta de red. Aquí se configuran la IP, el FQDN y los alias que queramos asignar:

```
# Begin /etc/hosts

127.0.0.1 localhost
192.168.1.50 zenon.home.es zenon

# End /etc/hosts
```

Instalar los scripts de arranque (bootscripts)

El paquete LFS-Bootscripts-20130821 contiene los scripts de arranque/parada que el sistema necesita a la hora de arracar/parar el sistema operativo. Con el comando **make install** se crea la estructura de directorios /etc/rc.d, /etc/init.d, /etc/sysconfig, /lib/services y /lib/lsb y después se instalan los scripts en el directorio /etc/init.d: checkfs, cleanfs, console, functions, halt, ifdown, ifup, localnet, modules, mountfs, mountvirtfs, network, rc, reboot, sendsignals, setclock, ipv4-static, swap, sysctl, sysklogd, template, udev, udev_retry.

Se han creado enlaces simóblicos para los scripts en los direferens runlevels:

```
ln -sf ../init.d/mountvirtfs /etc/rc.d/rcS.d/S00mountvirtfs
ln -sf ../init.d/modules     /etc/rc.d/rcS.d/S05modules
ln -sf ../init.d/localnet    /etc/rc.d/rcS.d/S08localnet
ln -sf ../init.d/udev        /etc/rc.d/rcS.d/S10udev
ln -sf ../init.d/swap        /etc/rc.d/rcS.d/S20swap
ln -sf ../init.d/checkfs     /etc/rc.d/rcS.d/S30checkfs
ln -sf ../init.d/mountfs     /etc/rc.d/rcS.d/S40mountfs
ln -sf ../init.d/cleanfs     /etc/rc.d/rcS.d/S45cleanfs
ln -sf ../init.d/udev_retry  /etc/rc.d/rcS.d/S50udev_retry
ln -sf ../init.d/console     /etc/rc.d/rcS.d/S70console
ln -sf ../init.d/sysctl      /etc/rc.d/rcS.d/S90sysctl
ln -sf ../init.d/network     /etc/rc.d/rc0.d/K80network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc0.d/K90sysklogd
ln -sf ../init.d/sendsignals /etc/rc.d/rc0.d/S60sendsignals
ln -sf ../init.d/swap        /etc/rc.d/rc0.d/S65swap
ln -sf ../init.d/mountfs     /etc/rc.d/rc0.d/S70mountfs
ln -sf ../init.d/localnet    /etc/rc.d/rc0.d/S90localnet
ln -sf ../init.d/halt        /etc/rc.d/rc0.d/S99halt
ln -sf ../init.d/network     /etc/rc.d/rc1.d/K80network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc1.d/K90sysklogd
ln -sf ../init.d/network     /etc/rc.d/rc2.d/K80network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc2.d/K90sysklogd
ln -sf ../init.d/sysklogd    /etc/rc.d/rc3.d/S10sysklogd
ln -sf ../init.d/network     /etc/rc.d/rc3.d/S20network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc4.d/S10sysklogd
ln -sf ../init.d/network     /etc/rc.d/rc4.d/S20network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc5.d/S10sysklogd
ln -sf ../init.d/network     /etc/rc.d/rc5.d/S20network
ln -sf ../init.d/network     /etc/rc.d/rc6.d/K80network
ln -sf ../init.d/sysklogd    /etc/rc.d/rc6.d/K90sysklogd
ln -sf ../init.d/sendsignals /etc/rc.d/rc6.d/S60sendsignals
ln -sf ../init.d/swap        /etc/rc.d/rc6.d/S65swap
ln -sf ../init.d/mountfs     /etc/rc.d/rc6.d/S70mountfs
ln -sf ../init.d/localnet    /etc/rc.d/rc6.d/S90localnet
ln -sf ../init.d/reboot      /etc/rc.d/rc6.d/S99reboot
```

Los runlevels son diferentes estados del sistema y actualmente existen siete, que van numerados de 0 a 6. También existen tres runlevels especiales que se llaman *ondemand runlevels* y son A, B y C. Cada uno corresponde a las acciones que el sistema tiene que tomar en función del runlevel seleccionado. SysVinit (o simplemente init) es el encargado de gestionar este esquema de runlevels y ejectuar el script correspondiente en cada estado. El runlevel por defecto es el 3.

Para cambiar de runlevel se ejecuta el comando *init <runlevel>*, por ejemplo *init 6* es el equivalente al comando *reboot*.

Como ya hemos visto antes, existen varios directorios bajo /etc/rc.d que tienen el nombre rc?.d (donde ? representa el número del runlevel). Dentro de estos directorios existen varios enlaces simbólicos cuyos nombres empiezan por K o por S seguidos por dos números de 00 a 99. Estos números indican el orden en el que se tienen que ejecutar los scripts mientras que K y S representan parar (kill) o iniciar (start) el servicio. La excepción a los explicado es cuando se trata de los runlevels 0 y 6 en cuyo caso no se iniciará ningún servicio aunque tenga la S en el principio del nombre ya que no tiene sentido iniciar nada cuando estamos a punto de reiniciar o parar el sistema.

A continuación una breve descripción de cada uno de estos runlevels:

0: paro del sistema
1: modo un solo usuario
2: modo múltiples usuarios sin funciones de red
3: modo múltiples usuarios con funciones de red
4: reservado para personalización, de lo contrario lo mismo que 3
5: lo mismo que 4, normalmente utilizado para login en modo gráfico (KDE, GNOME, etc)
6: reinicio del sistema

Durante la inicialización del kernel el primer programa que se ejecuta es *init* en caso de no especificarse otro en la línea de comandos. Este programa lee el fichero de inicialización /etc/inittab.

El fichero /etc/inittab describe que procesos se ejecutan al arrancar el sistema o durante el modo de operación normal en funcion del runlevel. Cada entrada de este fichero tiene el formato *id:runlevels:acción:proceso*, donde *id* es una secuencía única de hasta 4 caracteres que identifica la entrada en *inittab*. El segundo campo, *runlevels*, es el listado de runlevels para cuales la acción especificada en el siguiente campo (*accion)* se tiene que tomar. Por último, el campo *proceso* indica que proceso se tiene que ejecutar.
Cuando se cambia de *runlevel* cualquier proceso que se encuentra en ejecución y no está especificado para el nuevo *runlevel,* se mata, primero con una señal SIGTERM luego con SIGKILL.

Las acciones válidas que se pueden tomar son:

- *respawn*
  - El proceso se reiniciará independientemente de cuando termine
- *wait*
  - El proceso se iniciará una sola vez cuando se entra en el runlevel especificado y *init* esperará su finalización
- *once*
  - El proceso se ejecutará una sola vez cuando se entra en el runlevel especificado
- *boot*
  - El proceso se ejecutará durante el arranque del sistema
- *bootwait*
  - El proceso se ejecutará durante el arranque del sistema, mientras *init* espera su finalización. El campo *runlevels* se ignora para esta acción.
- *off*
  - No hace nada
- *ondemand*
  - Solo aplica para los runlevels A, B o C. Cuando se ejecuta uno de estos runlevels se lanzará el proceso pero el sistema no cambiará de runlevel.
- *initdefault*
  - Esta acción indica en que runlevel se entrará después del arranque del sistema. Si no se indica ninguno *init* preguntará por un runlevel en consola. El campo *proceso* se ignora para esta acción.
- *sysinit*
  - El proceso se ejecutará durante el arranque del sistema. Se ejecutará antes de cualquier entrada tipo *boot* o *bootwait*. El campo *runlevels* se ignora para esta acción.
- *powerwait*
  - El proceso se ejecutará cuando la alimentación eléctrica se para. Normalmente un proceso que comunica con la fuente de alimentación informa a *init* de la parada. *Init* esperará la finalización del proceso antes de continuar.
- *powerfail*
  - Lo mismo que *powerwait* con la excepción de que *init* no esperará la finalización del proceso.
- *powerokwait*
  - Este proceso se ejecutará en cuanto *init* recibe la información de que la alimentación ha sido recuperada.

- *powerfailnow*
  - Este proceso se ejecutará cunado a *init* se le informa que la batería de la fuente externa está casi descargada y la alimentación se corta.
- *ctrlaltdel*
  - El proceso se ejecutará cuando *init* recibe una señal SIGINT. Esto significa que alguien en la consola del sistema ha presionado la combinación de teclas CTRL-ALT-DEL.
- *kbrequest*
  - El proceso se ejecutará cuando *init* recibe una señal del teclado conforme una combinación de teclas especial ha sido presionada en el tecaldo de la consola.

Nuestro fichero */etc/inittab* contiene las siguientes entradas:

```
# Begin /etc/inittab

id:3:initdefault:

si::sysinit:/etc/rc.d/init.d/rc S

l0:0:wait:/etc/rc.d/init.d/rc 0
l1:S1:wait:/etc/rc.d/init.d/rc 1
l2:2:wait:/etc/rc.d/init.d/rc 2
l3:3:wait:/etc/rc.d/init.d/rc 3
l4:4:wait:/etc/rc.d/init.d/rc 4
l5:5:wait:/etc/rc.d/init.d/rc 5
l6:6:wait:/etc/rc.d/init.d/rc 6

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

su:S016:once:/sbin/sulogin

1:2345:respawn:/sbin/agetty --noclear tty1 9600
2:2345:respawn:/sbin/agetty tty2 9600
3:2345:respawn:/sbin/agetty tty3 9600
4:2345:respawn:/sbin/agetty tty4 9600
5:2345:respawn:/sbin/agetty tty5 9600
6:2345:respawn:/sbin/agetty tty6 9600

# End /etc/inittab
```

La shell Bash lee unos ficheros de arranque para al hacer el login, en estos ficheros se configuran algunos ajustes como pueden ser de configuración regional. A continuación vamos a crear el fichero de configuración /etc/profile en el cual indicamos que *locales* se utilizan por defecto cuando un usuario hacer login. Hay que tener en cuenta que este fichero se ignora si en el home del usuario existe el fichero equivalente.

Listamos todos los locales instalados en el sistema pero filtrando el idioma español de España:

```
root:/# locale -a | grep es_ES
es_ES
es_ES.iso88591
es_ES.iso885915@euro
es_ES.utf8
es_ES@euro
root:/#
```

ISO 8859-15 es la parte 15 de ISO 8859, un estándar de codificación de caracteres definido por la Organización Internacional para la Estandarización. También es conocido como Latin-9, y de forma no oficial como Latin-0 pero no como Latin-15. Es similar a ISO 8859-1 pero sustituye algunos símbolos poco comunes por el símbolo del euro y algunos otros caracteres que faltaban. Codifica los caracteres con 8 bits y puede usarse para representar el alfabeto y otros caracteres importantes para almacenar textos en inglés, francés, alemán, español y portugués (entre otros idiomas de Europa occidental) en ordenadores.

Antes de crear el fichero */etc/profile* vamos a probar que está correctamente instalado nuestro locale:

```
root:/# LC_ALL=es_ES.iso885915@euro locale charmap
ISO-8859-15
root:/# LC_ALL=es_ES.iso885915@euro locale language
Spanish
```

```
root:/# LC_ALL=es_ES.iso885915@euro locale int_curr_symbol
EUR
root:/# LC_ALL=es_ES.iso885915@euro locale int_prefix
34
```

Ahora que hemos comproado que el locale que nos interesa está instalado correctamente ya podemos proceder con la creación del fichero /etc/profile cuyo contenido será el siguiente:

```
# Begin /etc/profile

export LANG=es_ES.iso885915@euro

# End /etc/profile
```

Configurar el sistema como arrancable

Para que nuestro sistema sea arrancable necesitamos tener el fichero */etc/fstab* correctamente creado, compilar un kernel para el nuevo sistema y luego instalar un gestor de arranque (GRUB) para que podamos escoger el sistema operativo en el arranque.

El fichero */etc/fstab* es necesario para montar de forma automática los filesystems disponibles en el sistema y también permite ejecutar una comprobación de integridad antes de montarlo:

```
# Begin /etc/fstab

# file-system montaje             tipo         opciones     dump   fsck
/dev/sda1    /            ext4         defaults            1      1
/dev/sda2    /boot        ext4         defaults            0      0
/dev/sda4    /opt         ext4         defaults            0      0
proc         /proc        proc         nosuid,noexec,nodev 0      0
sysfs        /sys         sysfs        nosuid,noexec,nodev 0      0
devpts       /dev/pts     devpts       gid=5,mode=620      0      0
tmpfs        /run         tmpfs        defaults            0      0
devtmpfs     /dev         devtmpfs     mode=0755,nosuid    0      0

# End /etc/fstab
```

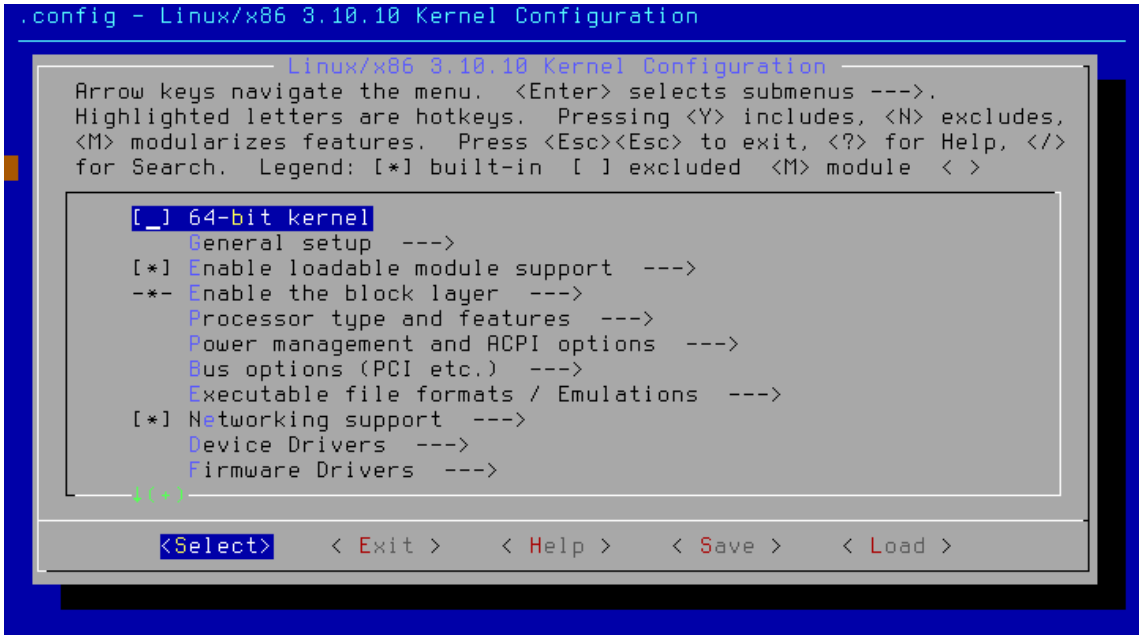
Instalación del kernel

A continuación instalaremos la versión 3.10.10 del kernel. Antes de seguir nos aseguramos de que será una instalación totalmente limpia mediante el comando **make mrproper**. Para compilar el kernel utilizamos el siguiente comando:

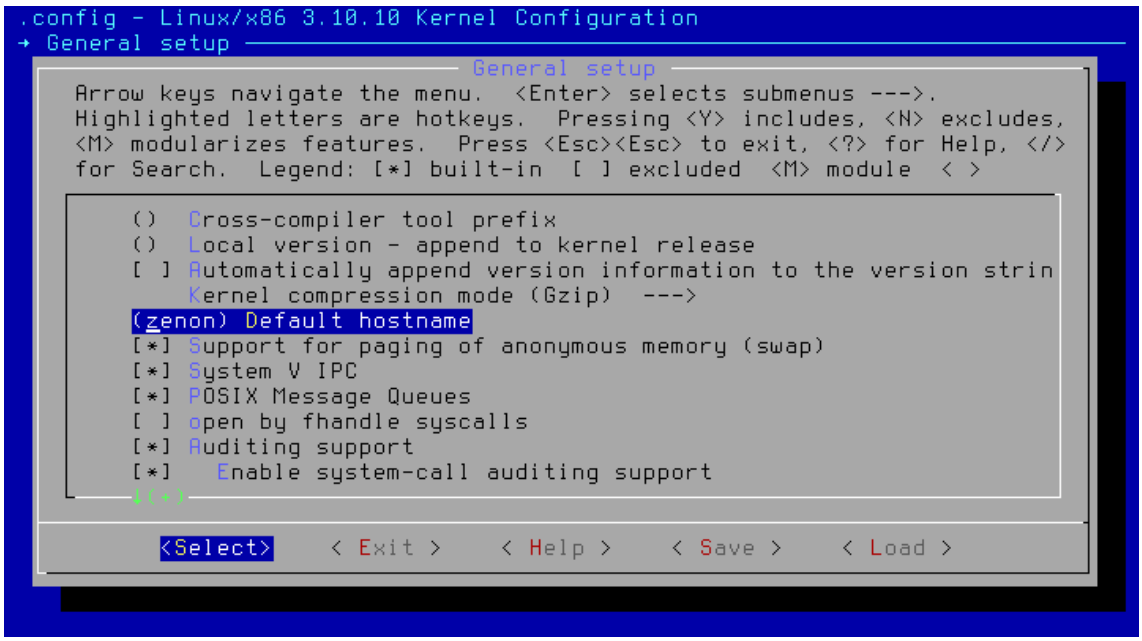
```
make LANG=es_ES.ISO-8859-15@euro LC_ALL= menuconfig
```

Las opciones modificadas son:

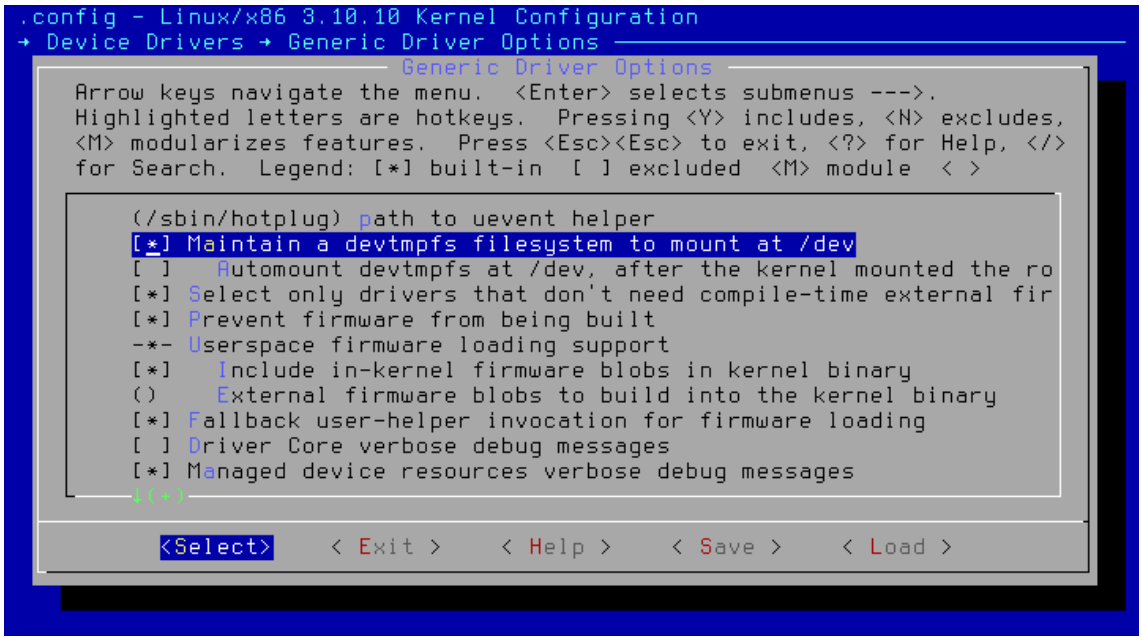
Desactivar: 64-bit kernel



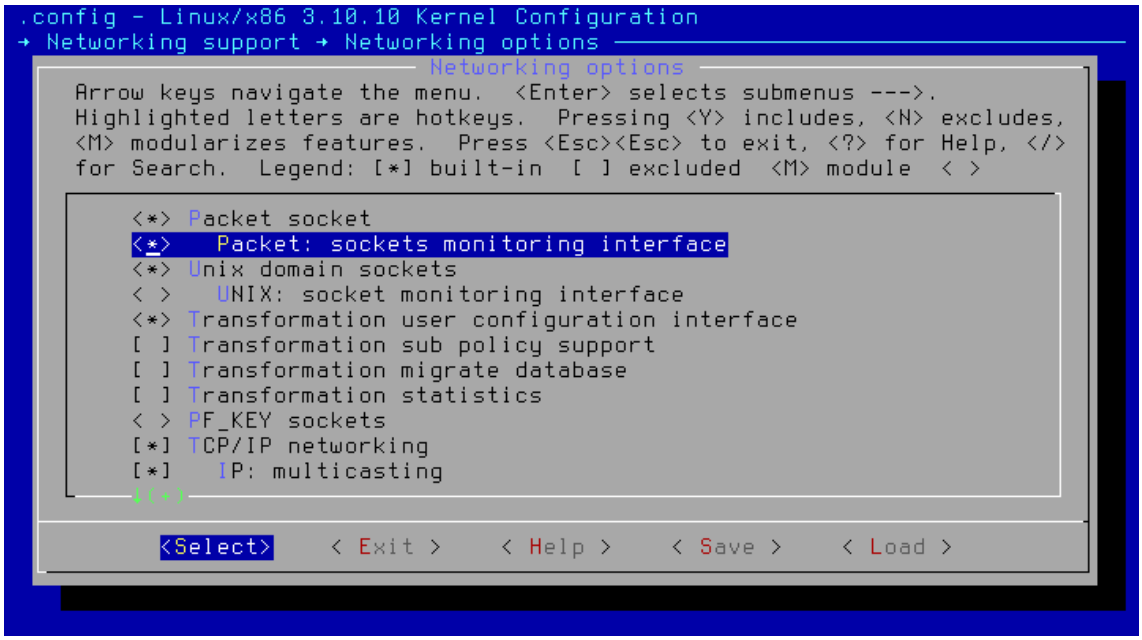
General setup → Default hostname: zenon



Activar: Device Drivers → Generic Drivers Options → Maintain a devtmpfs filesystem to mount at /dev



Activar: Networking support → Networking options → Packet: sockets monitoring interface



Despues de guardar los cambios en el fichero *.config* ya podemos compilar el kernel con el comando **make**. Una vez completada la compilación se generan unos ficheros muy importantes para que el sistema pueda arrancar. El primero de ellos es la imagen del kernel (bzImage), la copiamos en el directorio /boot:

```
cp -v arch/x86/boot/bzImage /boot/vmlinuz-3.10.10
```

El fichero de símbolos del kernel (System.map) permite ejecutar las funciones de API así como las direcciones de estructuras de datos del kernel durante su ejecución. También lo copiamos en /boot:

```
cp -v System.map /boot/System.map-3.10.10
```

Por seguridad hacemos una copia del fichero de configuración generado con el comando **make menuconfig**. Este fichero contiene toda la configuración del kernel recién compilado:

```
cp -v .config /boot/config-3.10.10
```

Instalar el gestor de arranque GRUB

GRUB (GNU GRand Unified Bootloader) es el gestor de arranque más utilizado en entornos Linux y permite arrancar varios sistemas operativos instalados en el mismo equipo. Actualmente soporta la mayoría de sistemas de archivos: ext2/ext3/ext4, reiserfs, xfs, ufs, vfat, ntfs, jfs, hfs, zfs, btrfs, iso9660.

GRUB utiliza la primera pista física del disco, esta parte del disco duro no es parte de ningún sistema de archivos. El directorio de instalación es /boot/grub y aquí es donde se guardan todos los ficheros de configuración. Nuestro sistema tiene una partición dedicada para /boot de 100MB.

Para instalar GRUB ejecutamos la siguiente instrucción, esta sobreescribirá el actual *boot loader* (MBR) en el primer disco duro:

```
grub-install /dev/sda
```

A cotinuación creamos el fichero de configuración necesario para poder arrancar el sistema:

```
# Begin /boot/grub/grub.cfg
set default=0
set timeout=5

insmod ext2
set root=(hd0,msdos2)

menuentry "GNU/Linux, Linux 3.10.10" {
        linux   /vmlinuz-3.10.10 root=/dev/sda3 ro
}
```

Pasos finales y primer arranque

Tenemos todos preparado para reinicar el sistema por primera vez. Antes de continuar se han revisado los siguientes ficheros:

- /etc/bashrc
- /etc/dircolors
- /etc/fstab
- /etc/hosts
- /etc/inputrc
- /etc/profile
- /etc/resolv.conf
- /etc/vimrc
- /root/.bash_profile
- /root/.bashrc
- /etc/sysconfig/network
- /etc/sysconfig/ifconfig.eth0

Salimos de del entorno *chroot* para poder desmontar correctamente los filesystems:

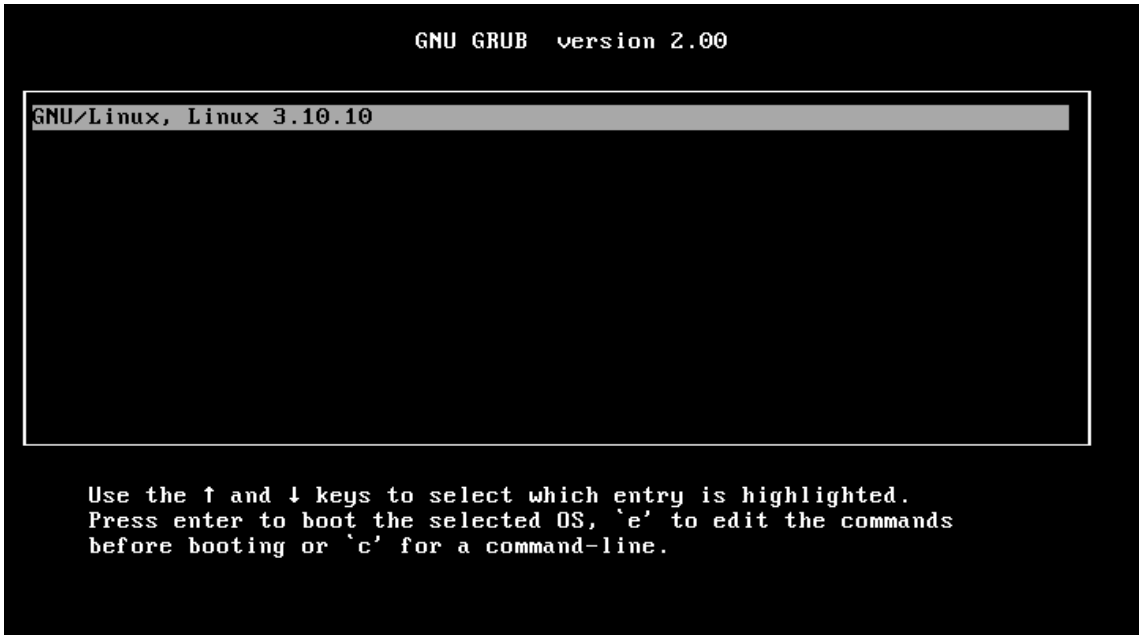
```
logout
umount -v $LFS/dev/pts

if [ -h $LFS/dev/shm ]; then
  link=$(readlink $LFS/dev/shm)
  umount -v $LFS/$link
  unset link
```

```
else
  umount -v $LFS/dev/shm
fi

umount -v $LFS/dev
umount -v $LFS/proc
umount -v $LFS/sys
umount -v $LFS/opt
umount -v $LFS/boot
umount -v $LFS
```

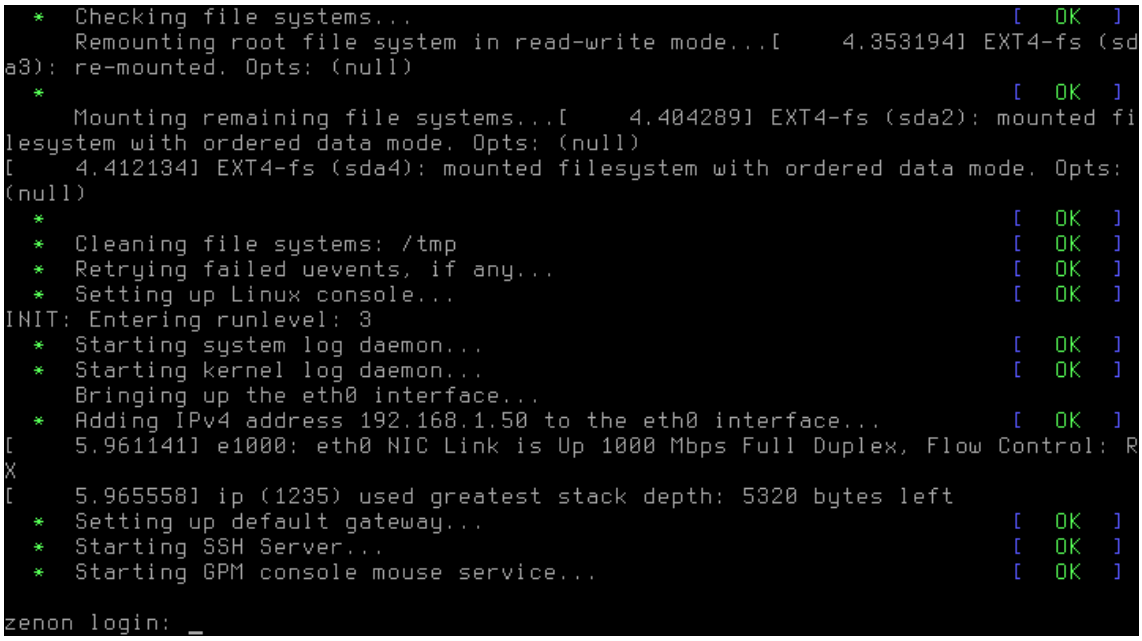
Reiniciamos el sistema:

```
shutdown -r now
```

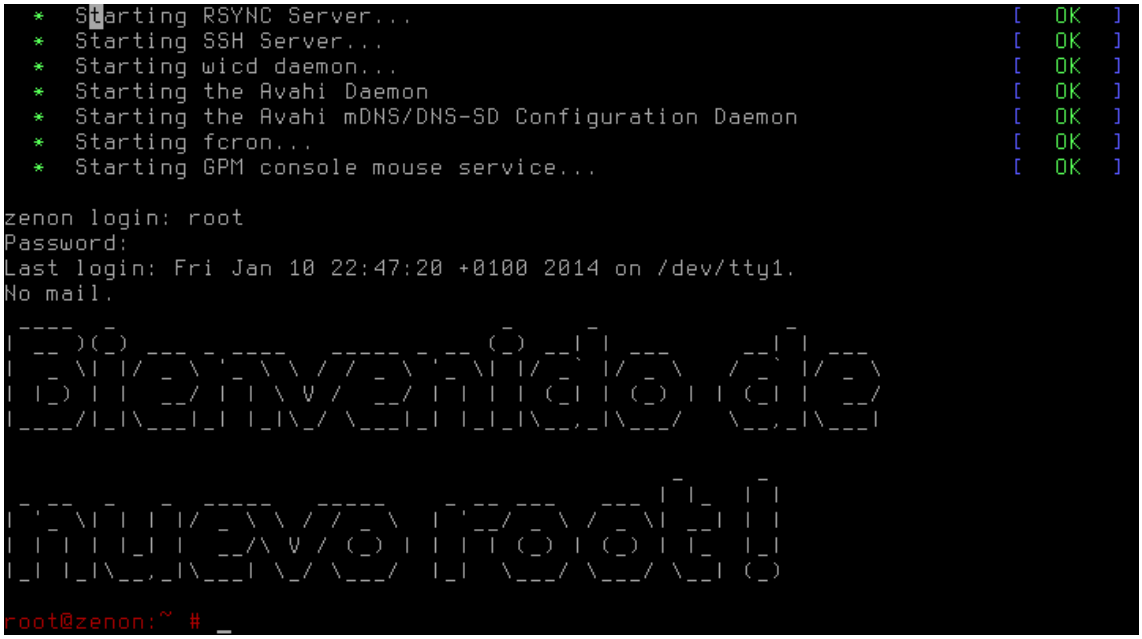
Nada más arrancar aparece el menú de GRUB con la entrada de nuestro sistema:



El sistema arranca sin ningún error y llega a pedirnos el usuario:



Iniciamos sesión con usuario *root*:



En este punto tenemos un sistema operativo Linux mínimo, tiene todo lo necesario para poder trabajar en modo consola con acceso a la red. A continuación instalaremos algunos programas y paquetes para personalizar nuestro Linux.

5. Personalizar el sistema

5.1 Librerías generales

Las librerías contienen código necesario para la ejecución de muchos programas. La ventaja de tener las librerías es que los programas no necesitan duplicar código y así evitar la creación de *bugs*, solamente llaman a funciones de las librerías instaladas en el sistema.

Existen muchísimas librerías pero nosotros vamos a instalar las que necesitamos para el buen funcionamiento de nustro Linux y las que son necesarias para la instalación de algunos programas específicos.

A continuación las librerías que se han instalado:

- dbus-1 => 1.6.14
- dconf-dbus-1 => 0.16.1
- dbus-glib-1 => 0.100.2
- dbus-python => 1.2.0
- expat => 2.1.0
- avahi-glib => 0.6.31
- glib-2.0 => 2.39.2
- dbus-glib-1 => 0.100.2
- gmime-2.6 => 2.6.17
- avahi-gobject => 0.6.31
- polkit-gobject-1 => 0.111
- gobject-introspection-no-export-1.0 => 1.36.0
- cairo-gobject => 1.12.16
- gobject-introspection-1.0 => 1.36.0
- gobject-2.0 => 2.39.2
- iso-codes => 3.46
- iso-codes => 3.46
- libcroco-0.6 => 0.6.8
- libdaemon => 0.14
- libdrm => 2.4.46
- libffi => 3.0.13
- libglade-2.0 => 2.6.4
- libtasn1 => 3.3
- libusb-1.0 => 1.0.9
- libusb-1.0 => 1.0.9
- libxslt => 1.1.28
- mtdev => 1.1.4
- nspr => 4.10.0
- popt => 1.16

5.2 Utilidades del sistema

Se han instalado los siguientes programas:

- GPM-1.20.7: contiene un servicio para el ratón que permite utilizarlo en consola y *xterm*
- acpid-2.0.19: es un demonio para gestionar eventos ACPI (Advanced Configuration and Power Interface) muy flexible y extensible
- Fcron-3.12: contiene un programador de tareas
- pciutils-3.2.0: contiene un set de programas para listar dispositivos PCI, revisar su estado y cambiar sus registros de configuración
- Sysstat-10.1.7: contiene utilidades para monitorizar el rendimiento del sistema y la actividad de uso. Sysstat contiene la utilidad *sar*, común en muchos sistemas Unix comerciales, y herramientas que permiten programar vía cron la recogida e histórico de rendiminato y la actividad
- UnRar-5.0.11: contiene una utilidad que permite extraer ficheros de archivos RAR
- usbutils-007: contiene una utilidad que se utiliza para mostrar información acerca de los buses USB disponibles en el sistema y los dispositivos conectados a ellos
- Which-2.20: contiene una utilidad que nos muestra el path completo de un determinado programa
- Zip-3.0: contiene utilidadez Zip utilizadas para comprimir ficheros en archivos ZIP
- D-Bus: es un bus de mensajes del sistema, una forma fácil de comunicación entre aplicaciones.

5.3 Herramientas de red y auditorías

Debido a que nuestro sistema se utilizará para auditorías de redes, se van a instalar varios programas dedicados a la monitorización de red y otras utilidades y librerías interesantes.

Se han instalado los siguientes paquetes:

- dhcpcd-6.0.5: es una implementación del cliente DHCP pero con muchas más funciones
- NcFTP-3.2.5: contiene una interfaz muy poderosa y flexible para el estándar FTP
- Net-tools-CVS_20101030: es una colección de programas para controlar el subsistema de red del Kernel de linux. Entre los programas instalados destacamos: arp, ifconfig, netstat, route.
- NFS-Utils-1.2.8: contiene las herramientas servidor y cliente necesarias para usar las abilidades NFS del kernel. El protocolo NFS permite compartir sistemas de ficheros a través de la red
- rsync-3.0.9: contiene la utilidad *rsync*. Permite sincronizar ficheros muy grandes a través de la red
- Wireless Tools-29: *Wireless Extension* es una API genérica en el kernel de Linux que le permite al controlador exponer la configuración y estadísticas específicas para Wireless LANs. Instala el programa *iwconfig*, entre otros.
- wpa_supplicant-2.0: WPA Supplicant es un cliente WPA (Wi-Fi Protected Access). Implementa la negociación de clave WPA con un WPA Authenticator y con una auntenticación EAP (Extensible Authentication Protocol) con un servidor de autenticación.
- Avahi-0.6.31: es un sistema que facilita el descubrimiento de servicios en una red local
- BIND Utilities-9.9.3-P2: es una colección de programas para el sitio cliente. Incluye los programas nslookup, dig y host.
- Nmap-6.40: es una utilidad para explorar redes y para auditorías de seguridad. Soporta escaneo de ping y rastreo TCP/IP.
- Traceroute-2.0.19: contiene un programa que se utiliza para mostrar las rutas de red que un paquete utiliza para llegar a un host destino.
- Whois-5.0.26: es una aplicación en el sitio cliente que hace una consulta para obtener información correspondiente a un nombre de dominio concreto. Instala los programas *whois* y *mkpasswd*.
- Wicd-1.7.2.4: es un gestor de res escrito en Python. Simplifica la configuración de la red detectando y conectado automáticamente a redes con cable y wireless. Incluye soporte para autenticación WPA y configuración DHCP.
- Snort-2.9.5.6: es un sistema de detección y prevención de intrusos en la red (IDS/IPS). Gracias a los beneficios de las firmas, su protocolo y la inspección basada en anomalias, es la tecnología IDS/IPS más utilizada a nivel mundial.
- Wireshark-1.10.2: es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos de forma únicamente hueca.
- Lynis-1.3.9: es una herramienta de auditoría muy poderosa para sistemas operativos basados en Unix. Escanea el sistema por información de seguridad, información general del sistema, información sobre software instalado, errores de configuración, problemas de seguridad, usuarios sin contraseñas, permisos de ficheros incorrectos, auditoría de cortafuegos y más.

Las herramientas más interesantes para mi son lynis y wireshark.

Para ejecutar lynis nos tenemos que cambiar al directorio donde está instalado (*/usr/local/lynis/)* y ejecutar el siguiente comando:

```
root@zenon:/usr/local/lynis # ./lynis -Q

[ Lynis 1.3.9 ]

################################################################################
 Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
 welcome to redistribute it under the terms of the GNU General Public License.
 See the LICENSE file for details about using this software.

 Copyright 2007-2014 - Michael Boelen, http://cisofy.com
 Enterprise support and plugins available via CISOfy - http://cisofy.com
################################################################################

[+] Initializing program
------------------------------------
  - Detecting OS...                                           [ DONE ]
```

```
  - Clearing log file (/var/log/lynis.log)...                 [ DONE ]

  ------------------------------------------------
  Program version:           1.3.9
  Operating system:          Linux
  Operating system name:     Linux
  Operating system version:  3.10.10
  Kernel version:            3.10.10
  Hardware platform:         i686
  Hostname:                  localhost
  Auditor:                   [Unknown]
  Profile:                   ./default.prf
  Log file:                  /var/log/lynis.log
  Report file:               /var/log/lynis-report.dat
  Report version:            1.0
  ------------------------------------------------
  - Checking profile file (./default.prf)...
  - Program update status...                                  [ NO UPDATE ]

[+] System Tools
------------------------------------
  - Scanning available tools...
  - Checking system binaries...
    - Checking /bin...                                        [ FOUND ]
    - Checking /sbin...                                       [ FOUND ]
    - Checking /usr/bin...                                    [ FOUND ]
-i used with no filenames on the command line, reading from STDIN.
    - Checking /usr/sbin...                                   [ FOUND ]
    - Checking /usr/local/bin...                              [ FOUND ]
    - Checking /usr/local/sbin...                             [ NOT FOUND ]
    - Checking /usr/local/libexec...                          [ NOT FOUND ]
    - Checking /usr/libexec...                                [ NOT FOUND ]
    - Checking /usr/sfw/bin...                                [ NOT FOUND ]
    - Checking /usr/sfw/sbin...                               [ NOT FOUND ]
    - Checking /usr/sfw/libexec...                            [ NOT FOUND ]
    - Checking /opt/sfw/bin...                                [ NOT FOUND ]
    - Checking /opt/sfw/sbin...                               [ NOT FOUND ]
    - Checking /opt/sfw/libexec...                            [ NOT FOUND ]
    - Checking /usr/xpg4/bin...                               [ NOT FOUND ]
    - Checking /usr/css/bin...                                [ NOT FOUND ]
    - Checking /usr/ucb...                                    [ NOT FOUND ]
    - Checking /usr/X11R6/bin...                              [ NOT FOUND ]

[+] Boot and services
------------------------------------
  - Checking boot loaders
    - Checking presence GRUB2...                              [ FOUND ]
    - Checking presence LILO...                               [ NOT FOUND ]
    - Checking boot loader SILO                               [ NOT FOUND ]
    - Checking boot loader YABOOT                             [ NOT FOUND ]
  - Check startup files (permissions)...                      [ OK ]

[+] Kernel
------------------------------------
  - Checking default run level...                             [ 3 ]
  - Checking CPU support (NX/PAE)
    CPU support: PAE and/or NoeXecute supported               [ FOUND ]
  - Checking kernel version and release                       [ DONE ]
  - Checking kernel type                                      [ DONE ]
  - Checking loaded kernel modules                            [ DONE ]
  - Checking Linux kernel configuration file...               [ FOUND ]
  - Checking core dumps configuration...                      [ DISABLED ]
    - Checking setuid core dumps configuration...             [ DEFAULT ]

[+] Memory and processes
------------------------------------
  - Checking /proc/meminfo...                                 [ FOUND ]
  - Searching for dead/zombie processes...                    [ OK ]
  - Searching for IO waiting processes...                     [ OK ]

[+] Users, Groups and Authentication
------------------------------------
  - Search administrator accounts...                          [ OK ]
  - Checking consistency of group files (grpck)...            [ OK ]
  - Checking non unique group ID's...                         [ OK ]
  - Checking non unique group names...                        [ OK ]
  - Checking password file consistency...                     [ OK ]
  - Query system users (non daemons)...                       [ DONE ]
```

```
  - Checking NIS+ authentication support                          [ NOT ENABLED ]
  - Checking NIS authentication support                           [ NOT ENABLED ]
  - Checking sudoers file                                         [ FOUND ]
    - Check sudoers file permissions                              [ OK ]
  - Checking PAM password strength tools                          [ OK ]
  - Checking PAM configuration file (pam.conf)                    [ NOT FOUND ]
  - Checking PAM configuration files (pam.d)                      [ FOUND ]
  - Checking PAM modules                                          [ FOUND ]
  - Checking accounts without expire date                         [ OK ]
  - Checking accounts without password                            [ OK ]
  - Checking user password aging                                  [ DISABLED ]
  - Checking Linux single user mode authentication                [ WARNING ]
  - Determining default umask
    - Checking umask (/etc/profile)                               [ UNKNOWN ]
    - Checking umask (/etc/login.defs)                            [ SUGGESTION ]
    - Checking umask (/etc/init.d/functions)                      [ SUGGESTION ]
    - Checking umask (/etc/init.d/rc)                             [ SUGGESTION ]
  - Checking LDAP authentication support                          [ NOT ENABLED ]

[+] Shells
------------------------------------

[+] File systems
------------------------------------
  - Checking mount points
    - Checking /home mount point...                               [ SUGGESTION ]
    - Checking /tmp mount point...                                [ SUGGESTION ]
  - Checking for old files in /tmp...                             [ OK ]
  - Checking /tmp sticky bit...                                   [ OK ]
  - ACL support root file system...                               [ ENABLED ]
  - Checking Locate database...                                   [ NOT FOUND ]

[+] Storage
------------------------------------
  - Checking usb-storage driver (modprobe config)...              [ NOT DISABLED ]
  - Checking firewire ohci driver (modprobe config)...            [ NOT DISABLED ]

[+] NFS
------------------------------------
  - Check running NFS daemon..                                    [ FOUND ]
    - Checking /etc/exports...                                    [ NOT FOUND ]
clnt_create: RPC: Unknown host
    - Checking NFS client access...                               [ OK ]

[+] Software: name services
------------------------------------
  - Checking default DNS search domain...                         [ NONE ]
  - Checking /etc/resolv.conf options...                          [ NONE ]
  - Searching DNS domain name...                                  [ UNKNOWN ]
  - Checking nscd status...                                       [ NOT FOUND ]
  - Checking BIND status...                                       [ NOT FOUND ]
  - Checking PowerDNS status...                                   [ NOT FOUND ]
  - Checking ypbind status...                                     [ NOT FOUND ]
  - Checking /etc/hosts
    - Checking /etc/hosts (duplicates)                            [ OK ]
    - Checking /etc/hosts (hostname)                              [ OK ]
    - Checking /etc/hosts (localhost)                             [ SUGGESTION ]

[+] Ports and packages
------------------------------------
  - Searching package managers...
  - Checking package audit tool...                                [ NONE ]

[+] Networking
------------------------------------
  - Checking configured nameservers...
    - Testing nameservers...
        Nameserver: 8.8.8.8...                                    [ OK ]
        Nameserver: 8.8.4.4...                                    [ OK ]
    - Minimal of 2 responsive nameservers...                      [ OK ]
  - Checking default gateway...                                   [ DONE ]
  - Getting listening ports (TCP/TCP)...                          [ DONE ]
      * Found 14 ports
  - Checking promiscuous interfaces...                            [ OK ]
  - Checking waiting connections...                               [ OK ]
  - Checking status DHCP client...                                [ NOT ACTIVE ]

[+] Printers and Spools
```

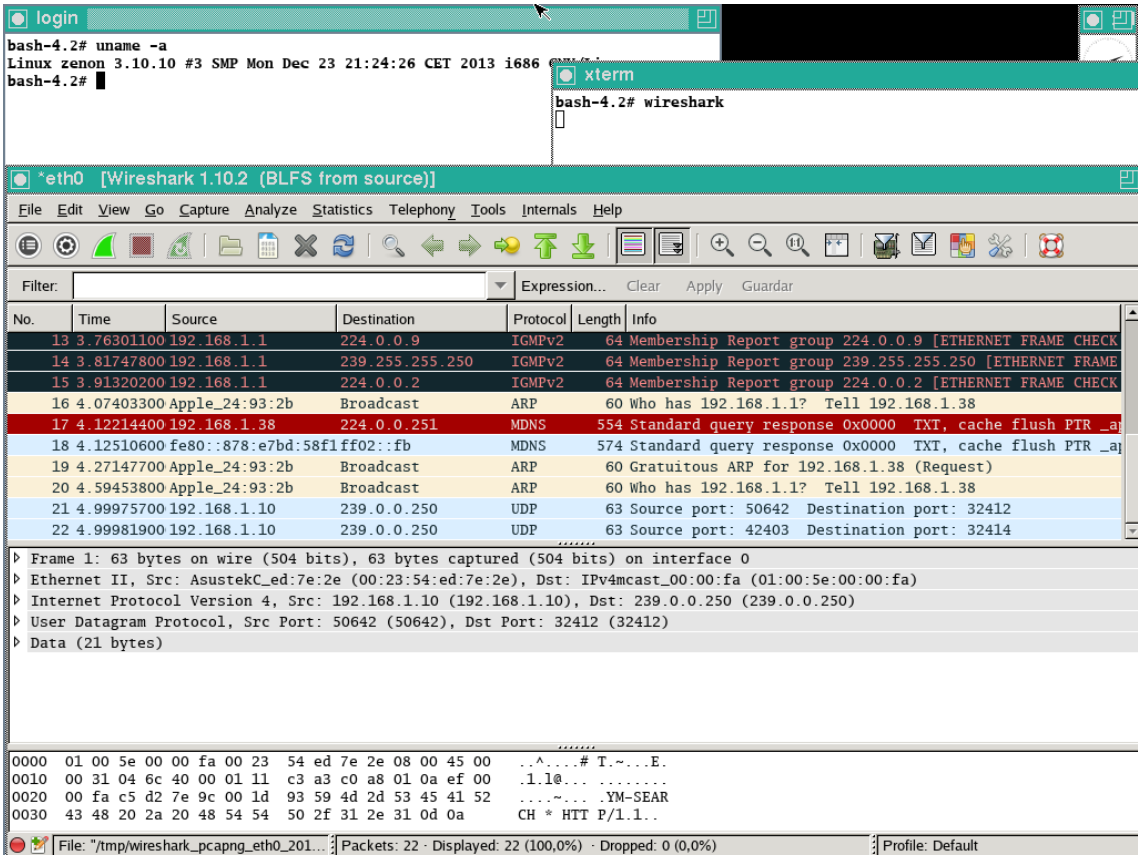
```
------------------------------------
  - Checking cups daemon...                                 [ NOT FOUND ]

[+] Software: e-mail and messaging
------------------------------------
  - Checking Exim status...                                 [ NOT FOUND ]
  - Checking Postfix status...                              [ NOT FOUND ]
  - Checking Qmail smtpd status...                          [ NOT FOUND ]

[+] Software: firewalls
------------------------------------
  - Checking iptables kernel module                         [ NOT FOUND ]
  - Checking iptables in config file                        [ FOUND ]
    Status pf                                               [ NOT FOUND ]
  - Checking host based firewall                            [ ACTIVE ]

[+] Software: webserver
------------------------------------
  - Checking Apache...                                      [ NOT FOUND ]
  - Checking nginx...                                       [ NOT FOUND ]

[+] SSH Support
------------------------------------
  - Checking running SSH daemon...                          [ FOUND ]
    - Searching SSH configuration...                        [ FOUND ]
    - Checking defined SSH options...                       [ DONE ]
    - SSH option: PermitRootLogin...                        [ DEFAULT ]
    - SSH option: Protocol...                               [ DEFAULT ]
    - SSH option: StrictModes...                            [ DEFAULT ]
    - SSH option: AllowUsers...                             [ NOT FOUND ]
    - SSH option: AllowGroups...                            [ NOT FOUND ]

[+] SNMP Support
------------------------------------
  - Checking running SNMP daemon...                         [ NOT FOUND ]

[+] Databases
------------------------------------
  - MySQL process status...                                 [ NOT FOUND ]
  - PostgreSQL processes status...                          [ NOT FOUND ]
  - Oracle processes status...                              [ NOT FOUND ]

[+] LDAP Services
------------------------------------
  - Checking OpenLDAP instance...                           [ NOT FOUND ]

[+] Software: PHP
------------------------------------
  - Checking PHP...                                         [ NOT FOUND ]

[+] Squid Support
------------------------------------
  - Checking running Squid daemon...                        [ NOT FOUND ]

[+] Logging and files
------------------------------------
  - Checking for a running log daemon...                    [ OK ]
    - Checking Syslog-NG status                             [ NOT FOUND ]
    - Checking Metalog status                               [ NOT FOUND ]
    - Checking RSyslog status                               [ NOT FOUND ]
    - Checking RFC 3195 daemon status                       [ NOT FOUND ]
    - Checking klogd                                        [ FOUND ]
    - Checking minilogd instances                           [ NONE ]
  - Checking logrotate presence                             [ WARNING ]
  - Checking remote logging                                 [ NOT ENABLED ]
  - Checking log directories (static list)                  [ DONE ]
  - Checking open log files                                 [ SKIPPED ]

[+] Insecure services
------------------------------------
  - Checking inetd status...                                [ NOT ACTIVE ]

[+] Banners and identification
------------------------------------
  - /etc/motd...                                            [ NOT FOUND ]
  - /etc/issue...                                           [ NOT FOUND ]
  - /etc/issue.net...                                       [ NOT FOUND ]
```

```
[+] Scheduled tasks
------------------------------------
  - Checking crontab/cronjob                                [ DONE ]
  - Checking atd status                                     [ NOT RUNNING ]

[+] Accounting
------------------------------------
  - Checking accounting information...                      [ NOT FOUND ]
  - Checking auditd                                         [ NOT FOUND ]

[+] Time and Synchronization
------------------------------------
  - Checking running NTP daemon (ntpd)...                   [ NOT FOUND ]
  - Checking running NTP daemon (timed)...                  [ NOT FOUND ]
  - Checking running NTP daemon (dntpd)...                  [ NOT FOUND ]
  - Checking NTP client in cron.d files...                  [ NOT FOUND ]
  - Checking for a running NTP daemon or client...          [ WARNING ]

[+] Cryptography
------------------------------------
  - Checking SSL certificate expiration...                  [ OK ]

[+] Virtualization
------------------------------------

[+] Security frameworks
------------------------------------
  - Checking presence AppArmor                              [ NOT FOUND ]
  - Checking presence SELinux                               [ NOT FOUND ]
  - Checking presence grsecurity                            [ NOT FOUND ]
  - Checking for implemented MAC framework                  [ NONE ]

[+] Software: file integrity
------------------------------------
  - Checking file integrity tools...
    - AFICK...                                              [ NOT FOUND ]
    - AIDE...                                               [ NOT FOUND ]
    - Osiris...                                             [ NOT FOUND ]
    - Samhain...                                            [ NOT FOUND ]
    - Tripwire...                                           [ NOT FOUND ]
    - OSSEC (syscheck)...                                   [ NOT FOUND ]
  - Checking presence integrity tool...                     [ NOT FOUND ]

[+] Software: Malware scanners
------------------------------------
  - Checking chkrootkit...                                  [ NOT FOUND ]
  - Checking Rootkit Hunter...                              [ NOT FOUND ]
  - Checking ClamAV scanner...                              [ NOT FOUND ]
  - Checking ClamAV daemon...                               [ NOT FOUND ]

[+] System Tools
------------------------------------
  - Starting file permissions check...
    /etc/lilo.conf                                          [ NOT FOUND ]
    /root/.ssh                                              [ OK ]

[+] Home directories
------------------------------------
  - Checking shell history files...                         [ OK ]

[+] Kernel Hardening
------------------------------------
  - Comparing sysctl key pairs with scan profile...
    - kernel.core_uses_pid (exp: 1)                         [ DIFFERENT ]
    - kernel.ctrl-alt-del (exp: 0)                          [ OK ]
    - kernel.sysrq (exp: 0)                                 [ DIFFERENT ]
    - net.ipv4.conf.all.accept_redirects (exp: 0)           [ DIFFERENT ]
    - net.ipv4.conf.all.accept_source_route (exp: 0)        [ OK ]
    - net.ipv4.conf.all.bootp_relay (exp: 0)                [ OK ]
    - net.ipv4.conf.all.forwarding (exp: 0)                 [ OK ]
    - net.ipv4.conf.all.log_martians (exp: 1)               [ DIFFERENT ]
    - net.ipv4.conf.all.mc_forwarding (exp: 0)              [ OK ]
    - net.ipv4.conf.all.proxy_arp (exp: 0)                  [ OK ]
    - net.ipv4.conf.all.rp_filter (exp: 1)                  [ DIFFERENT ]
    - net.ipv4.conf.all.send_redirects (exp: 0)             [ DIFFERENT ]
    - net.ipv4.conf.default.accept_redirects (exp: 0)       [ DIFFERENT ]
    - net.ipv4.conf.default.accept_source_route (exp: 0)    [ DIFFERENT ]
    - net.ipv4.conf.default.log_martians (exp: 1)           [ DIFFERENT ]
```

```
    - net.ipv4.icmp_echo_ignore_broadcasts (exp: 1)           [ OK ]
    - net.ipv4.icmp_ignore_bogus_error_responses (exp: 1)     [ OK ]
    - net.ipv4.tcp_syncookies (exp: 1)                        [ OK ]
    - net.ipv4.tcp_timestamps (exp: 0)                        [ DIFFERENT ]
    - net.ipv6.conf.all.accept_redirects (exp: 0)             [ DIFFERENT ]
    - net.ipv6.conf.all.accept_source_route (exp: 0)          [ OK ]
    - net.ipv6.conf.default.accept_redirects (exp: 0)         [ DIFFERENT ]
    - net.ipv6.conf.default.accept_source_route (exp: 0)      [ OK ]

[+] Hardening
------------------------------------
    - Installed compiler(s)...                                [ FOUND ]
    - Installed malware scanner...                            [ NOT FOUND ]

[+] Custom Tests
------------------------------------
  - Running custom tests...                                   [ SKIPPED ]

================================================================================

  -[ Lynis 1.3.9 Results ]-

  Tests performed: 143

  Warnings:
  ----------------------------
  - No password set for single mode [test:AUTH-9308]
  - No logrotate configuration has been found [test:LOGG-2146]
  - No running NTP daemon or available client found [test:TIME-3104]

  Suggestions:
  ----------------------------
  - Configure password aging limits to enforce password changing on a regular base
[test:AUTH-9286]
  - Set password for single user mode to minimize physical access attack surface
[test:AUTH-9308]
  - Default umask in /etc/login.defs could be more strict like 027 [test:AUTH-9328]
  - Default umask in /etc/init.d/rc could not be found and defaults usually to 022, whi-
ch could be more strict like 027 [test:AUTH-9328]
  - To decrease the impact of a full /home file system, place /home on a separated par-
tition [test:FILE-6310]
  - To decrease the impact of a full /tmp file system, place /tmp on a separated parti-
tion [test:FILE-6310]
  - The database required for 'locate' could not be found. Run 'updatedb' or 'locate.up-
datedb' to create this file. [test:FILE-6410]
  - Disable drivers like USB storage when not used, to prevent unauthorized storage or
data theft [test:STRG-1840]
  - Disable drivers like firewire storage when not used, to prevent unauthorized storage
or data theft [test:STRG-1846]
  - Split resolving between localhost and the hostname of the system [test:NAME-4406]
  - Install a package audit tool to determine vulnerable packages [test:PKGS-7398]
  - Check if files are properly rotated by a some tool instead of logrotate
[test:LOGG-2146]
  - Enable logging to an external logging host for archiving purposes and additional
protection [test:LOGG-2154]
  - Enable auditd to collect audit information [test:ACCT-9628]
  - Check if any NTP daemon is running or a NTP client gets executed daily, to prevent
big time differences and avoid problems with services like kerberos, authentication or
logging differences. [test:TIME-3104]
  - Install a file integrity tool [test:FINT-4350]
  - One or more sysctl values differ from the scan profile and could be tweaked
[test:KRNL-6000]
  - Harden the system by removing unneeded compilers. This can decrease the chance of
customized trojans, backdoors and rootkits to be compiled and installed [test:HRDN-7220]
  - Harden compilers and restrict access to world [test:HRDN-7222]
  - Harden the system by installing one or malware scanners to perform periodic file
system scans [test:HRDN-7230]
================================================================================
  Files:
  - Test and debug information      : /var/log/lynis.log
  - Report data                     : /var/log/lynis-report.dat
================================================================================
  Hardening index : [57]     [###########         ]

  Enterprise support and plugins available via CISOfy - http://cisofy.com
================================================================================
  Tip: Disable all tests which are not relevant or are too strict for the
       purpose of this particular machine. This will remove unwanted suggestions
```

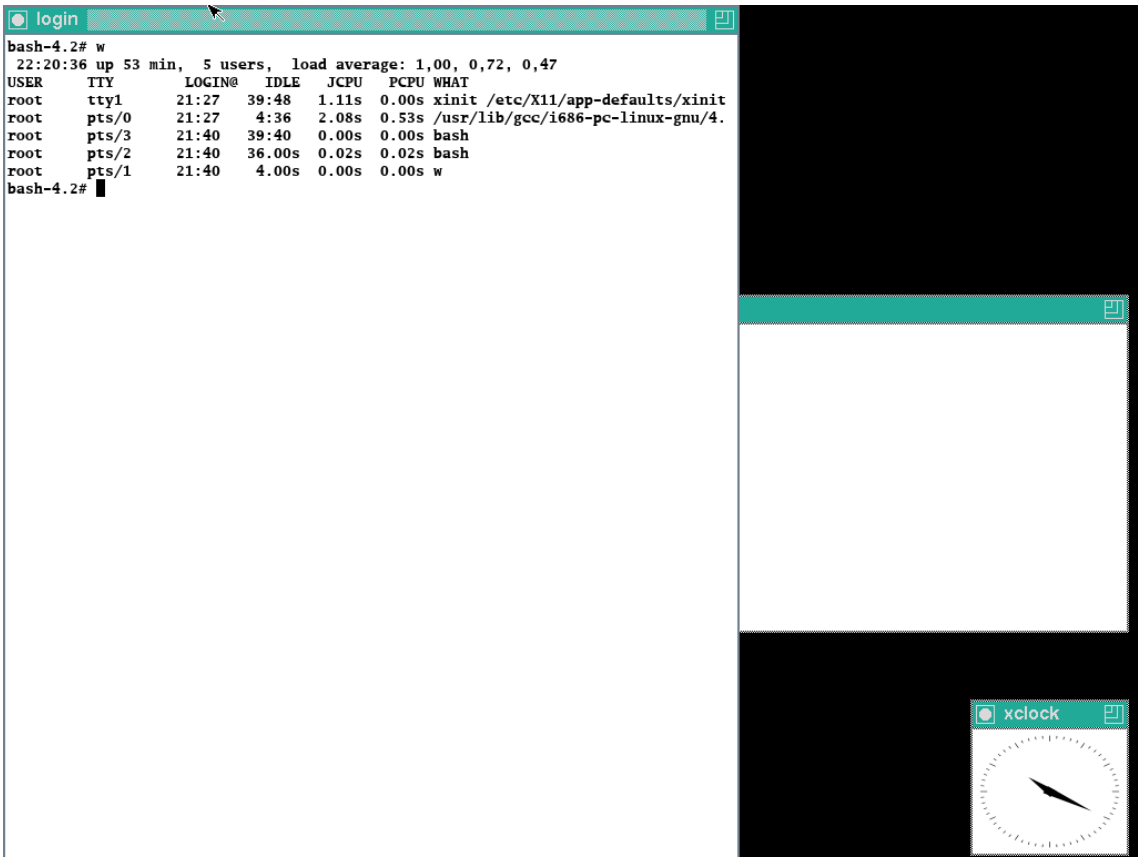
```
      and also boost the hardening index. Each test should be properly analyzed
      to see if the related risks can be accepted, before disabling the test.
================================================================================
  Lynis 1.3.9
  Copyright 2007-2014 - Michael Boelen, http://cisofy.com
================================================================================


root@zenon:/usr/local/lynis #
```

La instalación de Wireshark ha sido quizás la más complicada por los errores de compilación que me he encontrado. Al final he conseguido instalarlo, en la siguiente captura se puede ver la interfaz gráfica:



5.4 X Window System

El sistema tiene instalado el paquete *twm* que contiene un gestor de ventanas mínimo. Se han instalado los drivers Xorg VESA para así poder arrancar el entorno X con el comando *startx*. Así es como se ve el escritorio en X:



5.5 Revisión de paquetes instalados

Durante la instalación de todos los programas y utilidades se han tenido que instalar varias dependencias. En este punto final de la instalación de Linux este es un listado de todos los paquetes instalados:

```
root@zenon:/ # pkg-config --list-all
sqlite3                                 SQLite - SQL database engine
eog                                     Eye of GNOME - The GNOME Image Viewer
xcb-damage                              XCB Damage - XCB Damage Extension
xcb-xevie                               XCB Xevie - XCB Xevie Extension
blkid                                   blkid - Block device id library
pixman-1                                Pixman - The pixman library (version 1)
libpci                                  libpci - libpci
gmodule-export-2.0                      GModule - Dynamic module loader for GLib
libevent_pthreads                       libevent_pthreads - libevent_pthreads adds pthreads-
based threading support to libevent
xaw6                                    Xaw - X Athena Widgets Library, version 6
xaw7                                    Xaw - X Athena Widgets Library, version 7
cairo-svg                               cairo-svg - SVG surface backend for cairo graphics
library
fontenc                                 fontenc - The fontenc Library
libdrm_intel                            libdrm - Userspace interface to kernel DRM services
iso-codes                               iso-codes - ISO country, language, script and cu-
rrency codes and translations
nettle                                  Nettle - Nettle low-level cryptographic library (sy-
mmetric algorithms)
gnutls                                  GnuTLS - Transport Security Layer implementation for
the GNU system
```

```
libpcreposix                    libpcreposix - PCREPosix - Posix compatible interfa-
ce to libpcre
recordproto                     RecordProto - Record extension headers
atk-bridge-2.0                  atk-bridge-2.0 - ATK/D-Bus Bridge
libcroco-0.6                    libcroco - a CSS2 Parsing and manipulation Library
in C.
gail-3.0                        Gail - GNOME Accessibility Implementation Library
dri                             dri - Direct Rendering Infrastructure
dbus-1                          dbus - Free desktop message bus
xcb-renderutil                  XCB RenderUtil library - XCB RENDER-extension utili-
ties library
e2p                             e2p - Ext2fs userpace programs utility library
xcb-event                       XCB Event library - XCB event callback interface
gcr-ui-3                        gcr-ui-3 - GObject and GUI library for high level
crypto parsing and display
xcb-render                      XCB Render - XCB Render Extension
totem-plparser                  totem-plparser - Totem Playlist Parser library
glamor                          glamor - X.Org glamor common library.
gnome-icon-theme-symbolic       gnome-icon-theme-symbolic - A collection of symbolic
icons used as the basis for GNOME themes
cairo-png                       cairo-png - PNG functions for cairo graphics library
libprocps                       libprocps - Library to control and query process
state
gcr-3                           gcr-3 - GObject and GUI library for high level cryp-
to parsing and display
xft                             Xft - X FreeType library
pangoxft                        Pango Xft - Xft font support for Pango
xcursor                         Xcursor - X Cursor Library
libnl-3.0                       libnl - Convenience library for netlink sockets
xcmiscproto                     XCMiscProto - XCMisc extension headers
libxslt                         libxslt - XSLT library version 2.
gck-1                           gck-1 - GObject bindings for PKCS
glu                             glu - Mesa OpenGL Utility library
gtk+-x11-3.0                    GTK+ - GTK+ Graphical UI Library
xorg-server                     xorg-server - Modular X.Org X Server
xtrans                          XTrans - Abstract network code for X
libexif                         libexif - Library for easy access to EXIF data
gdk-pixbuf-2.0                  GdkPixbuf - Image loading and scaling
xf86vidmodeproto                XF86VidModeProto - XF86VidMode extension headers
gtksourceview-3.0               gtksourceview - GTK+ 3.0 Source Editing Widget
libdaemon                       libdaemon - a lightweight C library that eases the
writing of UNIX daemons
libisoburn-1                    libisoburn - Multi-session filesystem extension to
libisofs, libburn.
libburn-1                       libburn - Library to read/write optical discs
damageproto                     DamageProto - Damage extension headers
libvala-0.20                    Vala - The Vala compiler library
nspr                            NSPR - The Netscape Portable Runtime
xf86dgaproto                    XF86DGAProto - XF86DGA extension headers
expat                           expat - expat XML parser
videoproto                      VideoProto - Video extension headers
kbproto                         KBProto - KB extension headers
xcb-xvmc                        XCB XvMC - XCB XvMC Extension
xcb-atom                        XCB Atom library - XCB atom cache
libtirpc                        libtirpc - Transport Independent RPC Library
uuid                            uuid - Universally unique id library
xfont                           Xfont - X font Library
snort_output                    Snort - Snort dynamic output modules
cairo-xlib                      cairo-xlib - Xlib surface backend for cairo graphics
library
xatracker                       xatracker - Xorg Gallium3D acceleration library
fontutil                        FontUtil - Font utilities dirs
xcb-dri2                        XCB DRI2 - XCB DRI2 Extension
panel                           panelw - ncurses 5.9 add-on library
gconf-2.0                       gconf - GNOME Config System.
randrproto                      RandrProto - Randr extension headers
xcb-aux                         XCB Aux library - XCB convenient functions
avahi-gobject                   avahi-gobject - Avahi Multicast DNS Responder (GLib
GObject Support)
xcb-image                       XCB Image library - XCB image convenience library
xcb-icccm                       XCB ICCCM library - XCB ICCCM binding
polkit-gobject-1                polkit-gobject-1 - PolicyKit Authorization API
xrandr                          Xrandr - X RandR Library
gdk-2.0                         GDK - GTK+ Drawing Kit (x11 target)
avahi-glib                      avahi-glib - Avahi Multicast DNS Responder (GLib Su-
pport)
glamor-egl                      glamor-egl - X.Org glamor common library.
scrnsaverproto                  ScrnSaverProto - ScrnSaver extension headers
```

```
dri2proto                           DRI2Proto - DRI2 extension headers
libtasn1                            libtasn1 - Library for ASN.1 and DER manipulation
quota                               quota - Quota management library
pycairo                             Pycairo - Python bindings for cairo
xcomposite                          Xcomposite - X Composite Extension Library
sm                                  SM - X Session Management Library
gjs-1.0                             gjs-1.0 - JS bindings for GObjects
dconf                               dconf - dconf client library
ss                                  ss - Subsystem command parsing library
xxf86dga                            Xxf86dga - XFree86 Direct Graphics Access Extension
Library
cairo-ps                            cairo-ps - PostScript surface backend for cairo gra-
phics library
p11-kit-1                           p11-kit - Library and proxy module for properly
loading and sharing PKCS
fixesproto                          FixesProto - X Fixes extension headers
python                              Python - Python library
libcanberra                         libcanberra - Event Sound API
python2                             Python - Python library
xcb-dpms                            XCB DPMS - XCB DPMS Extension
gmodule-2.0                         GModule - Dynamic module loader for GLib
gdk-pixbuf-xlib-2.0                 GdkPixbuf Xlib - GdkPixbuf rendering for Xlib
gnome-icon-theme                    gnome-icon-theme - A collection of icons used as the
basis for GNOME themes
icon-naming-utils                   icon-naming-utils - Utilities for mapping legacy
GNOME and KDE icon names to the new Icon Naming Specification
libdrm_radeon                       libdrm_radeon - Userspace interface to kernel DRM
services for radeon
libdrm                              libdrm - Userspace interface to kernel DRM services
xcb-xinput                          XCB XInput - XCB XInput Extension (EXPERIMENTAL)
pango                               Pango - Internationalized text handling
xres                                XRes - X Resource Information Extension Library
lua                                 Lua - An Extensible Extension Language
xcb-xtest                           XCB XTEST - XCB XTEST Extension
pangoft2                            Pango FT2 and Pango Fc - Freetype 2.0 and fontconfig
font support for Pango
cairo-xlib-xcb                      cairo-xlib-xcb - Xlib/XCB functions for cairo gra-
phics library
atspi-2                             atspi - Accessibility Technology software library
xcb-proto                           XCB Proto - X protocol descriptions for XCB
fontsproto                          FontsProto - Fonts extension headers
gobject-introspection-no-export-1.0 gobject-introspection - GObject Introspection
gdk-3.0                             GDK - GTK+ Drawing Kit
compositeproto                      CompositeExt - Composite extension headers
mozjs-17.0                          SpiderMonkey 17.0.4esrpre - The Mozilla library for
JavaScript
panelw                              panelw - ncurses 5.9 add-on library
xvmc                                XvMC - The XvMC Library
xorg-evdev                          xorg-evdev - X.Org evdev input driver.
xineramaproto                       XineramaProto - Xinerama extension headers
xf86bigfontproto                    XF86BigFontProto - XF86BigFont extension headers
libgtop-2.0                         libgtop - Portable System Access Library
vapigen-0.20                        vapigen - Vala API Generator
glesv1_cm                           glesv1_cm - Mesa OpenGL ES 1.1 CM library
libpcre                             libpcre - PCRE - Perl compatible regular expressions
C library with 8 bit character support
libpcre16                           libpcre16 - PCRE - Perl compatible regular expres-
sions C library with 16 bit character support
libpipeline                         libpipeline - Pipeline manipulation library
cairo-xcb-shm                       cairo-xcb-shm - XCB/SHM functions for cairo graphics
library
pygobject-2.0                       PyGObject - Python bindings for GObject
cairo-pdf                           cairo-pdf - PDF surface backend for cairo graphics
library
libevent_openssl                    libevent_openssl - libevent_openssl adds openssl-ba-
sed TLS support to libevent
pangocairo                          Pango Cairo - Cairo rendering support for Pango
snort_preproc                       Snort - Snort dynamic preprocessors
gnome-keyring-1                     gnome-keyring - The GNOME keyring libraries
vg                                  vg - Mesa OpenVG 1.0 library
printproto                          PrintProto - Print extension headers
dmxproto                            DMXProto - DMX extension headers
gbm                                 gbm - Mesa gbm library
glproto                             GLProto - GL extension headers
x11-xcb                             X11 XCB - X Library XCB interface
gthread-2.0                         GThread - Thread support for GLib
libusb-1.0                          libusb-1.0 - C API for USB device access from Linux,
Mac OS X, OpenBSD, NetBSD and Windows userspace
```

```
xcb-glx                             XCB GLX - XCB GLX Extension
libssl                              OpenSSL - Secure Sockets Layer and cryptography li-
braries
xf86driproto                        XF86DRIProto - XF86DRI extension headers
libpcrecpp                          libpcrecpp - PCRECPP - C++ wrapper for PCRE
cairo-xcb                           cairo-xcb - XCB surface backend for cairo graphics
library
xcb-screensaver                     XCB Screensaver - XCB Screensaver Extension
menu                                menuw - ncurses 5.9 add-on library
xcb-composite                       XCB Composite - XCB Composite Extension
gl                                  gl - Mesa OpenGL library
libcanberra-gtk3                    libcanberra-gtk3 - Gtk3 Event Sound API
libcrypto                           OpenSSL-libcrypto - OpenSSL cryptography library
egl                                 egl - Mesa EGL library
libxml-2.0                          libXML - libXML library version2.
libdrm_nouveau                      libdrm_nouveau - Userspace interface to nouveau ker-
nel DRM services
alsa                                alsa - Advanced Linux Sound Architecture (ALSA) -
Library
xcb-xv                              XCB Xv - XCB Xv Extension
mount                               mount - mount library
xextproto                           XExtProto - XExt extension headers
cairo-gobject                       cairo-gobject - gobject functions for cairo graphics
library
gmodule-no-export-2.0               GModule - Dynamic module loader for GLib
yelp-xsl                            yelp-xsl - Yelp XSLT Stylesheets
dconf-dbus-1                        dconf-dbus-1 - dconf client library for libdbus-1
x11                                 X11 - X Library
xcb-res                             XCB Res - XCB X-Resource Extension
libnl-cli-3.0                       libnl-cli - Command Line Interface library for ne-
tlink sockets
bigreqsproto                        BigReqsProto - BigReqs extension headers
xtst                                Xtst - The Xtst Library
libpcre32                           libpcre32 - PCRE - Perl compatible regular expres-
sions C library with 32 bit character support
nss                                 NSS - Network Security Services
libevent                            libevent - libevent is an asynchronous notification
event loop library
libisofs-1                          libisofs - ISO9660 filesystem creation library
gcr-base-3                          gcr-base-3 - GObject library for high level crypto
parsing
gobject-introspection-1.0           gobject-introspection - GObject Introspection
xmu                                 Xmu - Xmu Library
gtk+-unix-print-2.0                 GTK+ - GTK+ Unix print support
vapigen                             vapigen - Vala API Generator
hogweed                             Hogweed - Nettle low-level cryptographic library
(public-key algorithms)
pciaccess                           pciaccess - Library providing generic access to the
PCI bus and devices.
gjs-internals-1.0                   gjs-internals-1.0 - Internal API for gjs (for modu-
les and embedders); uses mozjs
libexslt                            libexslt - EXSLT Extension library
avahi-ui-gtk3                       avahi-ui - Avahi Multicast DNS Responder (Common
GTK3 UI support)
libsoup-2.4                         libsoup - a glib-based HTTP library
resourceproto                       ResourceProto - Resource extension headers
xi                                  Xi - X Input Extension Library
vte-2.90                            vte - Vte terminal widget.
pygtk-2.0                           PyGTK - Python bindings for GTK+ and related libra-
ries
udev                                udev - udev
libkmod                             libkmod - Library to deal with kernel modules
xp                                  Xp - X Print Library
xbitmaps                            X bitmaps - Bitmaps that are shared between X appli-
cations
cairo-fc                            cairo-fc - Fontconfig font backend for cairo graphi-
cs library
xt                                  Xt - X Toolkit Library
libnl-nf-3.0                        libnl-nf - Netfilter Netlink Library
glib-2.0                            GLib - C Utility Library
gmime-2.6                           GMime - MIME parser and utility library
xv                                  Xv - The Xv Library
xorg-macros                         X.Org Macros - A set of autoconf project macros for
X.Org modules
mtdev                               mtdev - Multitouch Protocol Translation Library
libsecret-1                         libsecret-1 - GObject bindings for Secret Service
API
cairo                               cairo - Multi-platform 2D graphics library
```

```
cairo-ft                        cairo-ft - FreeType font backend for cairo graphics
library
libnl-route-3.0                 libnl-route - Netlink Routing Family Library
osmesa                          osmesa - Mesa Off-screen Rendering library
xcb-sync                        XCB Sync - XCB Sync Extension
gio-unix-2.0                    GIO unix specific APIs - unix specific headers for
glib I/O library
xcb-xprint                      XCB Xprint - XCB Xprint Extension
polkit-agent-1                  polkit-agent-1 - PolicyKit Authentication Agent API
vorbisfile                      vorbisfile - vorbisfile is a library that provides a
convenient high-level API for decoding and basic manipulation of all Vorbis I audio
streams
glesv2                          glesv2 - Mesa OpenGL ES 2.0 library
libpeas-gtk-1.0                 libpeas-gtk - libpeas-gtk, a GObject plugins library
(Gtk widgets)
xkbfile                         xkbfile - The xkbfile Library
ncurses++w                      ncurses++w - ncurses 5.9 add-on library
xfixes                          Xfixes - X Fixes  Library
vorbisenc                       vorbisenc - vorbisenc is a library that provides a
convenient API for setting up an encoding environment using libvorbis
avahi-ui                        avahi-ui - Avahi Multicast DNS Responder (Common
GTK2 UI support)
xcb-xfixes                      XCB XFixes - XCB XFixes Extension
xmuu                            Xmuu - Mini Xmu Library
gtk+-unix-print-3.0             GTK+ - GTK+ Unix print support
libpeas-1.0                     libpeas - libpeas, a GObject plugins library
xinerama                        Xinerama - The Xinerama Library
formw                           formw - ncurses 5.9 add-on library
xcb-xkb                         XCB XKB - XCB Keyboard Extension (EXPERIMENTAL)
libpng                          libpng - Loads and saves PNG files
usbutils                        usbutils - USB device database
xpm                             Xpm - X Pixmap Library
gdk-x11-2.0                     GDK - GTK+ Drawing Kit (x11 target)
gnome-desktop-3.0               gnome-desktop-3.0 - Utility library for loading
.desktop files
librsvg-2.0                     librsvg - library that renders svg files
xcb-keysyms                     XCB Keysyms library - XCB Keysyms
snort                           Snort - Snort dynamic plugins/detection/rules
gnome-video-effects             gnome-video-effects - A collection of GStreamer
effects to be used in different GNOME Modules
xcb-randr                       XCB RandR - XCB RandR Extension
libudev                         libudev - Library to access udev device information
freetype2                       FreeType 2 - A free, high-quality, and portable font
engine.
xext                            Xext - Misc X Extension Library
cairo-script                    cairo-script - script surface backend for cairo gra-
phics library
xcb-xf86dri                     XCB XFree86-DRI - XCB XFree86-DRI Extension
libtiff-4                       libtiff - Tag Image File Format (TIFF) library.
vorbis                          vorbis - vorbis is the primary Ogg Vorbis library
xau                             Xau - X authorization file management libary
harfbuzz                        harfbuzz - HarfBuzz text shaping library
xcb-record                      XCB Record - XCB Record Extension
xdamage                         Xdamage - X Damage  Library
ice                             ICE - X Inter Client Exchange Library
atk                             Atk - Accessibility Toolkit
openssl                         OpenSSL - Secure Sockets Layer and cryptography li-
braries and tools
libwnck-3.0                     libwnck - Window Navigator Construction Kit library
xkbcomp                         xkbcomp - XKB keymap compiler
xxf86vm                         Xxf86vm - XFree86 Video Mode Extension Library
gtk+-2.0                        GTK+ - GTK+ Graphical UI Library (x11 target)
dmx                             dmx - The dmx Library
ncursesw                        ncursesw - ncurses 5.9 library
python-2.7                      Python - Python library
libffi                          libffi - Library supporting Foreign Function Inter-
faces
popt                            popt - popt library.
libfs                           libFS - Library Interface to the X Font Server
libsecret-unstable              libsecret-unstable - GObject bindings for Secret
Service API (Unstable)
xcb-shape                       XCB Shape - XCB Shape Extension
xkeyboard-config                XKeyboardConfig - X Keyboard configuration data
totem-plparser-mini             totem-plparser-mini - Totem Playlist Parser library,
mini version
shared-mime-info                shared-mime-info - Freedesktop common MIME database
xrender                         Xrender - X Render Library
xproto                          Xproto - Xproto headers
```

```
inputproto                          InputProto - Input extension headers
libnl-genl-3.0                      libnl-genl - Generic Netlink Library
avahi-core                          avahi-core - Avahi Multicast DNS Responder (Embedda-
ble Stack)
xcb-util                            XCB Util Core library - XCB util core interface
xcb-shm                             XCB Shm - XCB Shm Extension
xscrnsaver                          XScrnSaver - The XScrnSaver Library
libpng16                            libpng - Loads and saves PNG files
xcb-ewmh                            XCB EWMH library - XCB EWMH binding
pangox                              Pango X - X Window System font support for Pango
xcb                                 XCB - X-protocol C Binding
dbus-glib-1                         dbus-glib - GLib integration for the free desktop
message bus
liblzma                             liblzma - General purpose data compression library
gdk-x11-3.0                         GDK - GTK+ Drawing Kit
ogg                                 ogg - ogg is a library for manipulating ogg bits-
treams
geoip                               geoip - A non-DNS IP-to-country resolver library.
portaudio-2.0                       PortAudio - Portable audio I/O
dbus-python                         dbus-python - Python bindings for D-Bus
ncurses                             ncursesw - ncurses 5.9 library
form                                formw - ncurses 5.9 add-on library
fontconfig                          Fontconfig - Font configuration and customization
library
cairo-xlib-xrender                  cairo-xlib-xrender - Xlib Xrender surface backend
for cairo graphics library
gtk+-3.0                            GTK+ - GTK+ Graphical UI Library
libsoup-gnome-2.4                   libsoup - a glib-based HTTP library
ext2fs                              ext2fs - Ext2fs library
avahi-client                        avahi-client - Avahi Multicast DNS Responder (Client
Support)
zlib                                zlib - zlib compression library
gsettings-desktop-schemas           gsettings-desktop-schemas - Shared GSettings schemas
for the desktop, including helper headers
com_err                             com_err - Common error description library
menuw                               menuw - ncurses 5.9 add-on library
gobject-2.0                         GObject - GLib Type, Object, Parameter and Signal
Library
gail                                Gail - GNOME Accessibility Implementation Library
gio-2.0                             GIO - glib I/O library
libglade-2.0                        Libglade - a library for dynamically loading GLADE
interface files
gtk+-x11-2.0                        GTK+ - GTK+ Graphical UI Library (x11 target)
xdmcp                               Xdmcp - X Display Manager Control Protocol library
xcb-xinerama                        XCB Xinerama - XCB Xinerama Extension
libkms                              libkms - Library that abstract aways the different
mm interface for kernel drivers
libcanberra-gtk                     libcanberra-gtk - Gtk Event Sound API
accountsservice                     Accounts Service - Client Library for communicating
with accounts service
renderproto                         RenderProto - Render extension headers
root@zenon:/ #
```

Entre los paquetes instalados se encuentran algunos que no son necesarios para el funcionamiento del sistema en modo consola, estos son los programas diseñados para funcionar en entorno GNOME. Uno de los objetivos iniciales era arrancar el entorno X en GNOME pero la compleja instalación desde código no me permite inclurlo en el proyecto. Sin embargo voy a instalar GNOME desde código una vez entregado el proyecto para enriquecer mis conocimientos y aprender como funciona. Voy a utilizar la herramienta JHBuild y las indicaciones de la página oficial de GNOME (http://www.gnome.org/gnome-3/source/).

6. Conclusiones

Antes de empezar el proyecto no tenía ni idea del tiempo que le iba a dedicar. En la planificación inicial he hecho unos cálculos aproximados para cada parte pero al terminarlo me he dado cuenta de que se necesita mucho más tiempo y dedicación para poder compilar e instalar todo lo propuesto. Sin embargo el resultado no está muy lejos de la idea inicial.

Después de dedicarle decenas de horas a este proyecto puedo decir que estoy satisfecho por el trabajo conseguido aunque me gustaría haber podido instalar más paquetes, sin embargo el tiempo no me lo ha permitido. Aun así he conseguido construir un Linux limpio con entorno X disponible aunque no he podido instalar el deseado GNOME. Han habido momentos de bloqueo pues en algunos paquetes faltaban dependencias, fallos de compilación o simplemente no encontraba según que librería y he tenido que consultar en Internet y mirar los foros. Muchos contratiempos han impedido conseguir un Linux más rico en cuanto a interfaz gráfica.

La compilación del Kernel necesita tiempo y más conocimientos para entender todas las opciones que ofrece. Con la ayuda de los foros he conseguido hacer funcionar varias funcionalidades, como por ejemplo el entorno X. He tenido que recompilarlo muchas veces, modificar el fichero de configuración de GRUB, reiniciar el sistema y comprobar que los cambios se han aplicado correctamente.

Este sistema operativo será la base de mi propio Linux que en el futuro cercano tendré instalado como sistema principal de mi PC. Me gusta tener el control sobre todo lo que se instala en el sistema. He instalado varias herramientas que permiten hacer auditorías de red, como por ejemplo wireshark, snort, nmap, bind o traceroute.

Aunque esta es la versión final del proyecto y el trabajo sobre el que se hará la valoración acaba aquí, siento que estoy en deuda conmigo y este trabajo se terminará. Por lo tanto me propongo como objetivo personal (y no académico) seguir con la instalación de GNOME para así poder convertirlo en un sistema operativo totalmente funcional en entorno gráfico.

He disfrutado muchísimo durante la preparación del proyecto. He pasado muchas horas delante de la pantalla alimentándome de nuevos conocimientos y ahora puedo decir que dispongo de lo necesario para poder instalar cualquier programa desde código fuente. Ha sido todo un reto y volvería a intentarlo con un proyecto de estas proporciones.

7. Referencias

www.linuxfromscratch.com
www.wikipedia.org
www.kernel.org
http://www.linuxforums.org/forum/
www.snort.org
http://www.wireshark.org/
http://www.yolinux.com/TUTORIALS/LinuxSecurityTools.html
http://www.figlet.org/
http://www.rootkit.nl/projects/lynis.html
http://www.tecmint.com/install-lynis-auditing-tool-in-rhel-centos-fedora/
http://www.gnome.org/gnome-3/source/