

Compilando MPlayer

Autor:

Death Master

Índice de contenidos

Índice de contenidos	2
Introducción	3
Un poco de historia	4
Calentando motores	5
Los codecs	11
Configuración (I)	17
Atando cabos sueltos	22
Configuración (II)	24
Compilación e instalación	29
Maquillando gmplayer	31
Distribución de este documento	34
Licencia	35

Introducción

“Si no se ve en MPlayer, eso no es un vídeo.”

*Alguien (lo siento, no recuerdo quién),
en el Hacklab wh2001 “Cielito Lindo”,
instantes antes de ver “Matrix Reloaded”.*

Aún recuerdo mis inicios con GNU/Linux -de hecho han transcurrido pocos años desde entonces-, cuando cada paso era un nuevo reto. Pelearse con el hardware, con las configuraciones, con los servicios, con el código fuente de los programas... todo ello reportaba (además de muchos dolores de cabeza, y algún que otro mosqueo) soltura con el sistema y sus componentes. Y así pasó el tiempo.

Echando la vista atrás, son muchas las distribuciones que he probado en este tiempo (hasta acabar con la que, hoy por hoy, es mi favorita: Debian), muchos también los programas, muchas las dudas consultadas en los foros, mucho el código fuente compilado...

Una de las cosas que más me ha gustado siempre del sistema GNU es la gran cantidad de programas que existen capaces de ofrecernos soluciones diferentes para el mismo problema: navegadores web, clientes de correo, editores de texto... entre todos ellos, cada usuario elige el que más se adapta a sus gustos y/o preferencias y hace de él su “favorito”.

Aunque en un principio pudiera parecer que la distribución de usuarios debería ser muy heterogénea, a la hora de la verdad suele centrarse en dos o tres programas, quedando el resto relegados a una posición minoritaria. Este hecho ha dado lugar a los muchos dilemas del software libre: KDE o Gnome, Vim o Emacs...

Existen excepciones, desde luego. Algunas en las que existe una mayor heterogeneidad en la distribución de usuarios (por ejemplo en los gestores de correo, clientes de mensajería instantánea...) y otras en las que el uso de un software es prácticamente mayoritario en detrimento del resto. En mi opinión, uno de los ejemplos más significativos de este último grupo es el software del que vamos a hablar: MPlayer.

Con MPlayer me inicié un poco más tarde que con el sistema GNU/Linux. En un principio me conformaba con los reproductores de vídeo que incluían las distribuciones con las que trabajaba (hay que reconocer que últimamente han mejorado enormemente, pero hace unos años...). Un poco más tarde, tuve mi primer encuentro con MPlayer, cuando oí hablar de él por primera vez e intenté compilarlo. Tuve problemas y no pude lograr que funcionara, por lo que me quedé con Xine (<http://xinehq.de/>), que sí pude compilar correctamente.

Un tiempo después, por fin, logré compilar y hacer funcionar correctamente una de las versiones previas de MPlayer 0.90 (no recuerdo cuál), y desde entonces es mi reproductor favorito.

Toda esta historia es la misma que la de cualquier usuario medio de Linux (quizá con ligeras diferencias), y en casi todos los casos hay un punto común bastante tedioso: los problemas para hacer que MPlayer funcione correctamente. MPlayer no suele incluirse como paquete preinstalado en las distribuciones (y cuando lo hace es con sus funcionalidades limitadas) y no suelen existir binarios precompilados (y cuando existen, están muy poco optimizados), por lo que su instalación pasa necesariamente por la compilación del código fuente.

Todo ese proceso, más aún si queremos optimizarlo al máximo, es el que genera todos los problemas a la hora de poder usar un software tan bueno como MPlayer. Por eso, vamos a ver paso a paso cómo llevarlo a cabo para que nos ahorremos todo el trámite problemático y pasemos de “oír hablar de” a “usar” MPlayer directamente. ;-)

Death Master

Un poco de historia

El proyecto de MPlayer nació el **22 de septiembre del año 2000** de la mano de **A'rpi**, con el nombre de **mpg12play v0.1-v0.3**; y la primera versión de MPlayer propiamente dicho fue la **v0.3** del 18 de noviembre del 2000.

Han pasado casi cinco años desde entonces y el proyecto ha madurado enormemente. Aunque aún no se ha alcanzado la primera versión final (**1.0**), sus versiones previas (**1.0preX**) llevan siendo desarrolladas y liberadas desde el 1 de septiembre de 2003. Para conocer más sobre el desarrollo histórico de las versiones de MPlayer: <http://www.mplayerhq.hu/DOCS/HTML-single/en/MPlayer.html#history>. También podemos visitar el enlace en castellano: <http://www.mplayerhq.hu/DOCS/HTML-single/es/MPlayer.html#history>.

Para saber más sobre el desarrollo del propio software, lo mejor es revisar el archivo de noticias de su actual sitio web: <http://www.mplayerhq.hu/homepage/design7/news-archive.html>. Podemos observar que MPlayer es un proyecto que cumple la máxima del dicho "despacito pero con buena letra": la liberación de sus versiones suele distanciarse mucho en el tiempo, pero cada una de ellas incorpora una **enorme cantidad de cambios, mejoras y nuevas implementaciones**. Y cuando digo enorme, es **ENORME**. Aquí hay un fichero changelog muy interesante que ilustra esa afirmación: <http://www.videohelp.com/tools?changelog=393>.

Esa enorme cantidad de mejoras con cada sucesiva versión han hecho que actualmente el proyecto posea una **lista de características impresionante**: <http://www.mplayerhq.hu/homepage/design7/info.html>. Todo ello ha hecho de MPlayer uno de los programas más populares en el mundo del software libre. Si miramos las estadísticas del portal **freshmeat** (<http://freshmeat.net/stats/>) podemos observar que MPlayer ocupa el **primer puesto en popularidad**, por encima incluso del propio kernel de Linux; y el séptimo puesto en las calificaciones de los usuarios (con una media de **9.12 puntos sobre 10**).

En todo este tiempo han pasado muchas cosas: **Nick Kurshev** (autor de **Vidix**) desarrolló un fork de MPlayer denominado **MplayerXP** (<http://mplayerxp.sf.net/>) de escasa repercusión y poco éxito; se comenzó el desarrollo de una segunda generación de MPlayer (**MplayerG2**) reescrita desde cero, y que lleva años abandonado (<http://mplayerhq.hu/~arpi/g2/>); el reproductor de vídeo doméstico Kiss **violó la licencia GPL** de MPlayer (<http://mplayerhq.hu/~gabucino/mp-kiss-gabu.rm>); el 26 de marzo de 2004 **A'rpi**, creador del proyecto, lo abandona; el 23 de abril, apenas un mes después, **Gabucino** abandona también el proyecto...

Muchas, en definitiva, son las cosas que han pasado desde ese lejano ya año 2000. Os recomiendo encarecidamente bucear en profundidad en la web del proyecto (<http://www.mplayerhq.hu/>), sin duda encontraréis muchas cosas interesantes.

En la actualidad, la última versión liberada de MPlayer, el **16 de abril de 2005**, es la **1.0pre7** apodada "**PatentCounter**" en honor a la inminente (en el momento de la liberación) votación en el Parlamento europeo sobre la directiva de patentabilidad de software (que, afortunadamente, fue rechazada). Sobre esta versión trabajaremos.

Calentando motores

Lo primero que debemos hacer siempre es descargar el paquete comprimido que contiene el código fuente de la aplicación a compilar. Como ya hemos dicho, vamos a trabajar con la versión **1.0pre7** de MPlayer, por lo que vamos a la página web del proyecto (<http://www.mplayerhq.hu/>) y visitamos la sección de **descargas** (<http://www.mplayerhq.hu/homepage/design7/dload.html>), dentro de la cual encontraremos un apartado llamado **"Source and Binaries"** (<http://www.mplayerhq.hu/homepage/design7/dload.html#source>).

En dicho apartado encontraremos los enlaces para descargar MPlayer, bien sea su **código fuente**, el de la última versión del **CVS**, o **binarios precompilados** específicos para distintos sistemas (Microsoft Windows, Linux Red Hat o Fedora, y Mac OS X).

Source and Binaries

Quick compilation and installation instructions are contained in the README, the installation section of the documentation has more complete information.

DESCRIPTION	HTTP	FTP
MPlayer v1.0pre7 source	[HU HU2 US CH]	[HU HU2 US]
daily CVS snapshot source	[HU HU2 US]	[HU HU2 US]
Windows releases	[HU HU2 US CH]	[HU HU2 US]
Red Hat / Fedora RPM packages	[US PL PL2]	[PL]
Mac OS X preview release	[HU HU2 US CH]	[HU HU2 US]

Si bien es perfectamente posible compilar MPlayer a partir de las fuentes del **CVS (Concurrent Versions System)**, no es muy recomendable: se trata de la **versión en desarrollo** y podemos encontrarnos con ciertos imprevistos. Por ello, seleccionamos uno cualquiera de los enlaces a **MPlayer v1.0pre7 source** y **descargamos** el fichero desde él.

```

master@menzoberranzan:~$ wget http://www1.mplayerhq.hu/MPlayer/releases/MPlayer-1.0pre7.tar.bz2
--14:03:36-- http://www1.mplayerhq.hu/MPlayer/releases/MPlayer-1.0pre7.tar.bz2
=> `MPlayer-1.0pre7.tar.bz2'
Resolving www1.mplayerhq.hu... 192.190.173.45
Connecting to www1.mplayerhq.hu[192.190.173.45]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6,868,877 [application/x-tar]

100%[=====>] 6,868,877
52.98K/s  ETA 00:00

14:05:44 (52.65 KB/s) - `MPlayer-1.0pre7.tar.bz2' saved [6868877/6868877]

```

```

master@menzoberranzan:~$

```

Una vez descargado, comprobamos que el fichero ha sido descargado correctamente. Para ello, vamos a comprobar el **hash MD5** y **SHA-1** del fichero.

```

master@menzoberranzan:~$ md5sum MPlayer-1.0pre7.tar.bz2
5fadd6957d3aab989cd760ff38fb8fdf MPlayer-1.0pre7.tar.bz2
master@menzoberranzan:~$ sha1sum MPlayer-1.0pre7.tar.bz2
df1e8d4f2f44d72c6f7989932f3b272e815ecb80 MPlayer-1.0pre7.tar.bz2
master@menzoberranzan:~$

```

Una vez **verificada la integridad** del fichero, lo descomprimimos:

```

master@menzoberranzan:~$ tar xjf MPlayer-1.0pre7.tar.bz2
master@menzoberranzan:~$

```

Tras lo cual se habrá creado un **directorio MPlayer-1.0pre7** que contiene -entre otros- el **código fuente** de la aplicación. El sistema de compilación de MPlayer es el estándar en Linux: un fichero de **autoconfiguración (configure)** que genera un makefile, el cual a su vez nos permite **compilar** el software automáticamente. Lo primero, pues, es echar un vistazo a las opciones de configuración de MPlayer:

master@menzoberranzan:~/MPlayer-1.0pre7\$./configure --help

Usage: ./configure [OPTIONS]...

Configuration:

-h, --help display this help and exit

Installation directories:

--prefix=DIR use this prefix for installing mplayer [/usr/local]
 --bindir=DIR use this prefix for installing mplayer binary
 [PREFIX/bin]
 --datadir=DIR use this prefix for installing machine independent
 data files (fonts, skins) [PREFIX/share/mplayer]
 --mandir=DIR use this prefix for installing manpages [PREFIX/man]
 --confdir=DIR use this prefix for installing configuration files
 [PREFIX/etc/mplayer]
 --libdir=DIR use this prefix for object code libraries [PREFIX/lib]

Optional features:

--disable-mencoder disable mencoder (a/v encoder) compilation [enable]
 --enable-gui enable gmpayer compilation (GTK 1.2 GUI) [disable]
 --enable-largefiles enable support for files > 2 GBytes [disable]
 --enable-linux-devfs set default devices to devfs ones [disable]
 --enable-termcap use termcap database for key codes [autodetect]
 --enable-termios use termios database for key codes [autodetect]
 --disable-iconv do not use iconv(3) function [autodetect]
 --disable-setlocale disable setlocale using in mplayer [autodetect]
 --disable-langinfo do not use langinfo [autodetect]
 --enable-lirc enable LIRC (remote control) support [autodetect]
 --enable-lirccd enable LIRCCD (LIRC client daemon) input [autodetect]
 --enable-joystick enable joystick support [disable]
 --disable-vm disable support X video mode extensions [autodetect]
 --disable-xf86keysym disable support for 'multimedia' keys [autodetect]
 --disable-tv disable TV Interface (tv/dvb grabbers) [enable]
 --disable-tv-v4l disable Video4Linux TV Interface support [autodetect]
 --disable-tv-v4l2 disable Video4Linux2 TV Interface support [autodetect]
 --disable-tv-bsdbt848 disable BSD BT848 Interface support [autodetect]
 --disable-edl disable EDL (edit decision list) support [enable]
 --disable-rtc disable RTC (/dev/rtc) on Linux [autodetect]
 --disable-network disable network support (for: http/mms/rtp) [enable]
 --enable-winsoc2 enable winsoc2 usage [autodetect]
 --enable-smb enable Samba (SMB) input support [autodetect]
 --enable-live enable LIVE.COM Streaming Media support [autodetect]
 --disable-dvdread Disable libdvdread support [autodetect]
 --disable-mpdvdkit Disable mpdvdkit/mpdvdkit2 support [autodetect]
 --disable-cdparanoia Disable cdparanoia support [autodetect]
 --disable-freetype Disable freetype2 font rendering support [autodetect]
 --disable-fontconfig Disable fontconfig font lookup support [autodetect]
 --disable-unrarlib Disable Unique RAR File Library [enabled]
 --enable-menu Enable OSD menu support (NOT DVD MENU) [disabled]
 --disable-sortsub Disable subtitles sorting [enabled]
 --enable-fribidi Enable using the FriBiDi libs [disabled]
 --disable-enca Disable using ENCA charset oracle library [autodetect]
 --disable-macosx Disable Mac OS X specific features [autodetect]
 --enable-macosx-finder-support Enable Mac OS X Finder invocation parameter parsing [disabled]
 --enable-macosx-bundle Enable Mac OS X bundle file locations [autodetect]
 --disable-inet6 Disable IPv6 support [autodetect]
 --disable-gethostbyname2 gethostbyname() function is not provided by the C
 library [autodetect]

--disable-ftp *Disable ftp support [enabled]*
 --disable-vstream *Disable tivo vstream client support [autodetect]*

Codecs:

--enable-gif *enable gif support [autodetect]*
 --enable-png *enable png input/output support [autodetect]*
 --enable-jpeg *enable jpeg input/output support [autodetect]*
 --enable-liblzo *enable external liblzo support [autodetect]*
 --disable-win32 *disable Win32 DLL support [autodetect]*
 --disable-dshow *disable Win32/DirectShow support [autodetect]*
 --disable-qtz *disable Quicktime codecs [autodetect]*
 --disable-xanim *disable XAnim DLL support [autodetect]*
 --disable-real *disable RealPlayer DLL support [autodetect]*
 --disable-xvid *disable XviD codec [autodetect]*
 --disable-x264 *disable H.264 encoder [autodetect]*
 --disable-divx4linux *disable DivX4linux/Divx5linux codec [autodetect]*
 --enable-opendivx *enable _old_ OpenDivx codec [disable]*
 --disable-libavcodec *disable libavcodec [autodetect]*
 --disable-libavformat *disable libavformat [autodetect]*
 --enable-libfame *enable libfame realtime encoder [autodetect]*
 --disable-internal-tremor *do not build internal OggVorbis support [enabled]*
 --enable-tremor-low *build with lower accuracy internal tremor [disabled]*
 --enable-external-tremor *build with external tremor [disabled]*
 --disable-vorbis *disable OggVorbis support entirely [autodetect]*
 --enable-theora *build with OggTheora support [autodetect]*
 --disable-internal-matroska *disable internal Matroska support [enabled]*
 --enable-external-faad *build with external FAAD2 (AAC) support [autodetect]*
 --disable-internal-faad *disable internal FAAD2 (AAC) support [autodetect]*
 --disable-ladspa *disable LADSPA plugin support [autodetect]*
 --disable-libdv *disable libdv 0.9.5 en/decoding support [autodetect]*
 --disable-mad *disable libmad (MPEG audio) support [autodetect]*
 --disable-toolame *disable Toolame (MPEG layer 2 audio) support in mencoder [autodetect]*
 --enable-xmms *build with XMMS inputplugin support [disabled]*
 --disable-mp3lib *disable builtin mp3lib [enabled]*
 --disable-liba52 *disable builtin liba52 [enabled]*
 --enable-libdts *enable libdts support [autodetect]*
 --disable-libmpeg2 *disable builtin libmpeg2 [enabled]*
 --disable-amr_nb *disable amr narrowband, floating point [autodetect]*
 --disable-amr_nb-fixed *disable amr narrowband, fixed point [autodetect]*
 --disable-amr_wb *disable amr wideband, floating point [autodetect]*

Video output:

--disable-vidix *disable VIDIX [enable on x86 *nix]*
 --enable-gl *build with OpenGL render support [autodetect]*
 --enable-dga[=n] *build with DGA [n in {1, 2}] support [autodetect]*
 --enable-vesa *build with VESA support [autodetect]*
 --enable-svga *build with SVGAlib support [autodetect]*
 --enable-sdl *build with SDL render support [autodetect]*
 --enable-aa *build with AALib render support [autodetect]*
 --enable-caca *build with CACA render support [autodetect]*
 --enable-ggi *build with GGI render support [autodetect]*
 --enable-directx *build with DirectX support [autodetect]*
 --enable-dxr2 *build with DXR2 render support [autodetect]*
 --enable-dxr3 *build with DXR3/H+ render support [autodetect]*
 --enable-dvb *build with support for output via DVB-Card [autodetect]*
 --enable-dvbhead *build with DVB support (HEAD version) [autodetect]*
 --enable-mga *build with mga_vid (for Matrox G200/G4x0/G550) support
 (check for /dev/mga_vid) [autodetect]*
 --enable-xmga *build with mga_vid X Window support*

(check for X & /dev/mga_vid) [autodetect]
 --enable-xv build with Xv render support for X 4.x [autodetect]
 --enable-xvmc build with XvMC acceleration for X 4.x [disable]
 --enable-vm build with XF86VidMode support for X11 [autodetect]
 --enable-xinerama build with Xinerama support for X11 [autodetect]
 --enable-x11 build with X11 render support [autodetect]
 --enable-fbdev build with FBDev render support [autodetect]
 --enable-mlib build with MLIB support (Solaris only) [autodetect]
 --enable-3dfx build with obsolete /dev/3dfx support [disable]
 --enable-tdxfb build with tdfxfb (Voodoo 3/banshee) support [disable]
 --enable-directfb build with DirectFB support [autodetect]
 --enable-zr build with ZR360[56]7/ZR36060 support [autodetect]
 --enable-bl build with Blinkenlights support [disable]
 --enable-tdfxvid build with tdfx_vid support [disable]
 --disable-tga disable targa output support [enable]
 --disable-pnm disable pnm output support [enable]
 --disable-md5sum disable md5sum output support [enable]

Audio output:

--disable-alsa disable ALSA sound support [autodetect]
 --disable-ossaudio disable OSS sound support [autodetect]
 --disable-arts disable aRts sound support [autodetect]
 --disable-esd disable esd sound support [autodetect]
 --disable-polyp disable Polypaudio sound support [autodetect]
 --disable-jack disable JACK sound support [autodetect]
 --disable-nas disable NAS sound support [autodetect]
 --disable-sgiaudio disable SGI sound support [autodetect]
 --disable-sunaudio disable Sun sound support [autodetect]
 --disable-win32waveout disable Windows waveout sound support [autodetect]
 --disable-select disable using select() on audio device [enable]

Miscellaneous options:

--enable-runtime-cpudetection Enable runtime CPU detection [disable]
 --cc=COMPILER use this C compiler to build MPlayer [gcc]
 --host-cc=COMPILER use this C compiler to build apps needed for the build process [gcc]
 --as=ASSEMBLER use this assembler to build MPlayer [as]
 --target=PLATFORM target platform (i386-linux, arm-linux, etc)
 --enable-static build a statically linked binary. Set further linking options with --enable-static="-lslang -lncurses"
 --charset convert the help messages to this charset
 --language=list a white space or comma separated list of languages for translated man pages, the first language is the primary and therefore used for translated messages and GUI (also the environment variable \$LINGUAS is honored) [en]
 (Available: bg cs de dk el en es fr hu it ja ko mk nl no pl ro ru sk sv tr uk pt_BR zh_CN zh_TW all)

--enable-shared-pp install & use shared postprocessing lib
 --with-install=PATH use a custom install program (useful if your OS uses a GNU-incompatible install utility by default and you want to use GNU version)
 --install-path=PATH the path to a custom install program this option is obsolete and will be removed soon, use --with-install instead.

Advanced options:

--enable-mmx build with MMX support [autodetect]
 --enable-mmx2 build with MMX2 support (PIII, Athlon) [autodetect]
 --enable-3dnow build with 3DNow! support [autodetect]


```

--enable-3dnowex    build with extended 3DNow! support [autodetect]
--enable-sse        build with SSE support [autodetect]
--enable-sse2       build with SSE2 support [autodetect]
--enable-shm        build with shm support [autodetect]
--enable-altivec    build with AltiVec support (PowerPC) [autodetect]
--disable-fastmemcpy    disable 3DNow!/SSE/MMX optimized memcpy() [enable]
--enable-big-endian Force byte order to big-endian [autodetect]
--enable-debug[=1-3] compile debugging information into mplayer [disable]
--enable-profile    compile profiling information into mplayer [disable]
--disable-sighandler    disable sighandler for crashes [enable]
--enable-crash-debug    enable automatic gdb attach on crash [disable]
--enable-i18n       _experimental_ gnu gettext() support [autodetect]
--enable-dynamic-plugins    Enable support for dynamic a/v plugins [disable]

```

Hazardous options a.k.a. "DO NOT REPORT ANY BUGS!"

```

--disable-gcc-checking    disable gcc version checking [enable]

```

Use these options if autodetection fails (Options marked with (*) accept multiple paths separated by ':'):

```

--with-extraincdir=DIR    extra headers (png, mad, sdl, ...) in DIR (*)
--with-extralibdir=DIR    extra library files (png, mad, sdl, ...) in DIR (*)
--with-x11incdir=DIR      X headers in DIR (*)
--with-x11libdir=DIR      X library files in DIR (*)
--with-dxr2incdir=DIR     DXR2 headers in DIR (*)
--with-dvbincdir=DIR     DVB headers in DIR (*)
--with-madlibdir=DIR      libmad (libmad shared library) in DIR (*)
--with-mlibdir=DIR        libmllib (MLIB support) in DIR (Solaris only)
--with-codecsdir=DIR     Binary codec files in DIR
--with-win32libdir=DIR    W*ndows DLL files in DIR
--with-xanimlibdir=DIR    XAnim DLL files in DIR
--with-reallibdir=DIR     RealPlayer DLL files in DIR
--with-xvidlibdir=DIR     libxvidcore (XviD) in DIR (*)
--with-xvidincdir=DIR     XviD header in DIR (*)
--with-x264libdir=DIR     libx264 in DIR
--with-x264incdir=DIR     x264 header in DIR
--with-dtslibdir=DIR     libdts library in DIR (*)
--with-dtsincdir=DIR     libdts header in DIR (*)
--with-livelibdir=DIR     LIVE.COM Streaming Media libraries in DIR
--with-toolamedir=DIR     path to Toolame library and include file
--with-xmmsplugindir=DIR  XMMS plugins in DIR
--with-xmmslibdir=DIR     libxmms.so.1 in DIR
--with-bio2jack=DIR       libbio2jack.a in DIR
--with-cdparanoiaincdir=DIR    cdparanoia headers in DIR (*)
--with-cdparanoialibdir=DIR    cdparanoia libraries (libcdda_*) in DIR (*)
--with-xvmlib=NAME        name of adapter-specific library (e.g. XvMCNVIDIA)
--with-termcaplib=NAME     name of library with termcap functionality
                          name should be given without leading "lib"
                          checks for "termcap" and "tinfo"

--with-freetype-config=PATH path to freetype-config
                          (e.g. /opt/bin/freetype-config)
--with-fribidi-config=PATH path to fribidi-config
                          (e.g. /opt/bin/fribidi-config)
--with-glib-config=PATH    path to glib*-config (e.g. /opt/bin/glib-config)
--with-gtk-config=PATH     path to gtk*-config (e.g. /opt/bin/gtk-config)
--with-sdl-config=PATH     path to sdl*-config (e.g. /opt/bin/sdl-config)
master@menzoberranzan:~/MPlayer-1.0pre7$

```

Como podemos comprobar, la cantidad de opciones es impresionante. En general, cada opción especifica el **argumento** con el que debe invocarse el script de configuración para modificarla, así como una breve **descripción** de la misma.

La mayoría de las opciones podremos ignorarlas perfectamente, especialmente aquellas en cuya descripción se especifica que el propio script se encargará de **verificar si deben ser activadas o no**. Dichas opciones están marcadas con el tag **[autodetect]**.

Otras opciones están **activadas [enable]** o **desactivadas [disable]** por defecto y deberemos especificar explícitamente a la hora de ejecutar el script si deseamos modificar su comportamiento.

Por último, existen ciertas opciones que deben ser especificadas, como por ejemplo una **ruta a un fichero o directorio**, o un **tipo de plataforma**. Estas opciones no suelen usarse (aunque, como más adelante veremos, nosotros sí necesitaremos de alguna).

Ahora, antes de comenzar a compilar el programa, es el momento de hablar de los **codecs**.

Los codecs

Si hablamos de contenidos multimedia, más aún en el caso de audio y vídeo, es imprescindible hablar de los **codecs**. La palabra **codec** es una abreviatura de **CO**dicador-**DE**Codificador (**CO**der-**DE**Coder) o de **CO**mpresor-**DE**Compresor (**CO**mpresor-**DE**Compresor), según la fuente que consultemos. Un codec describe la **especificación** de un método (lógico o físico) que realiza transformaciones en un **flujo de datos** o una **señal**.

Más coloquialmente, un codec es “lo que nos permite ver y oír nuestros ficheros de vídeo y audio”. Es imprescindible tener en nuestro sistema determinados codecs, y que el software encargado de reproducir los ficheros los reconozca, si deseamos poder reproducirlos correctamente.

Para saber más sobre los codecs, recomiendo echar un vistazo a sus artículos de la **Wikipedia** (en castellano e inglés respectivamente):

<http://es.wikipedia.org/wiki/C%C3%B3dec>

<http://en.wikipedia.org/wiki/Codec>

En lo que a MPlayer se refiere, desearemos disponer de **tantos codecs activados como nos sea posibles**, de forma que el programa reconozca la mayor cantidad de tipos de fichero posible. En la sección de descargas de la web del proyecto MPlayer encontramos un apartado llamado “**Codecs**” (<http://www.mplayerhq.hu/homepage/design7/dload.html#codecs>) a través del cual -pinchando en el enlace “other binary codec packages”- podemos acceder a la página de Codecs (<http://www1.mplayerhq.hu/homepage/design7/codecs.html>).

Como la arquitectura sobre la que vamos a compilar MPlayer es **GNU/Linux**, debemos bajar el pack con el título “**all**”.

```
master@menzoberranzan:~$ wget http://www1.mplayerhq.hu/MPlayer/releases/codecs/all-20050412.tar.bz2
```

```
--14:18:10-- http://www1.mplayerhq.hu/MPlayer/releases/codecs/all-20050412.tar.bz2  
=> `all-20050412.tar.bz2'
```

```
Resolving www1.mplayerhq.hu... 192.190.173.45  
Connecting to www1.mplayerhq.hu[192.190.173.45]:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 12,963,387 [application/x-tar]
```

```
100%[=====>] 12,963,387  
52.73K/s ETA 00:00
```

```
14:22:11 (52.69 KB/s) - `all-20050412.tar.bz2' saved [12963387/12963387]
```

```
master@menzoberranzan:~$
```

Comprobamos los **hashes**...

```
master@menzoberranzan:~$ md5sum all-20050412.tar.bz2  
291780f05ff1ef78287749eee1ff2404 all-20050412.tar.bz2  
master@menzoberranzan:~$ sha1sum all-20050412.tar.bz2  
39b949dacc015d8e3aa308c24c74a97598fe92c9 all-20050412.tar.bz2  
master@menzoberranzan:~$
```

Y pasamos a **descomprimir** el paquete y copiar todos los codecs a un directorio cualquiera (sería recomendable, para poder seguir el manual al pie de la letra, que utilizarais el mismo que yo).

```
master@menzoberranzan:~$ tar xjf all-20050412.tar.bz2
master@menzoberranzan:~$ su
Password:
menzoberranzan:/home/master# mkdir /usr/local/lib/codecs
menzoberranzan:/home/master# cp all-20050412/* /usr/local/lib/codecs/
menzoberranzan:/home/master# exit
exit
master@menzoberranzan:~$
```

Bien, ya tenemos el paquete **estándar** de codecs. Pero hay un codec que usaremos mucho y, por tanto, vamos a darle un trato especial: se trata de **XviD**.

XviD (<http://www.xvid.org/>) es un **codec MPEG-4 libre** (bajo **GNU GPL**) y gratuito. Como siempre, para más información recomiendo visitar la **Wikipedia**.

<http://es.wikipedia.org/wiki/XVID>

<http://en.wikipedia.org/wiki/Xvid>

Seguramente muchos de vosotros no conoceréis **XviD**, pero sí conoceréis otro codec similar llamado **DivX**. Y seguramente que muy pocos de vosotros sabréis que ambos tienen un antepasado común: **OpenDivX**. En 2001 **DivXNetworks** fundó el proyecto **OpenDivX**, un codec libre y gratuito basado en una variante de MPEG-4. Por varios motivos (intereses económicos principalmente), DivXNetworks decidió **cerrar su codec** en la versión 4, pasando a ser freeware su versión de decodificación y de pago la versión de codificación (pro). En ese punto, **XviD surgió como un fork** de la última versión libre de OpenDivX y siguió su desarrollo por su propio camino.

Aunque en un principio XviD no gozaba de mucha popularidad, cada vez ésta es mayor. Como ha ocurrido con muchas parejas de software cerrado-abierto "hermanas" (StarOffice y OpenOffice.org, Netscape y Mozilla...), al final el software propietario acaba nutriéndose del libre, y ahora DivX incorpora ciertas características más tarde que sus competidores libres. Un ejemplo lo tenemos en la última versión de DivX (la sexta) que incorpora varias capas de audio para soporte de **varios idiomas** en un único fichero, así como **soporte de subtítulos** y **menús**. Estas funcionalidades (excepto los menús) están implementadas desde hace bastante tiempo en el formato **OGM** (que utiliza XviD como contenedor de vídeo).

Bien, lo que vamos a hacer es **bajar el código fuente del codec XviD y compilarlo** nosotros mismos.

Lo primero que haremos es visitar la página oficial del proyecto **XviD** (<http://www.xvid.org/>) y bajar la última versión del codec.

```
master@menzoberranzan:~$ wget http://downloads.xvid.org/downloads/xvidcore-1.1.0-beta2.tar.bz2
--14:30:23-- http://downloads.xvid.org/downloads/xvidcore-1.1.0-beta2.tar.bz2
=> `xvidcore-1.1.0-beta2.tar.bz2'
Resolving downloads.xvid.org... 131.188.30.102
Connecting to downloads.xvid.org[131.188.30.102]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 594,159 [application/x-tar]

100%[=====>] 594,159
52.99K/s  ETA 00:00

14:30:35 (52.65 KB/s) - `xvidcore-1.1.0-beta2.tar.bz2' saved [594159/594159]

master@menzoberranzan:~$
```

Comprobamos los **hashes**...

```
master@menzoberranzan:~$ md5sum xvidcore-1.1.0-beta2.tar.bz2
1556584438f5b9fcd4c5e4ca829da602 xvidcore-1.1.0-beta2.tar.bz2
master@menzoberranzan:~$ sha1sum xvidcore-1.1.0-beta2.tar.bz2
07983d16ee205ec135d9e09d83287caf10c350b5 xvidcore-1.1.0-beta2.tar.bz2
master@menzoberranzan:~$
```

Y **descomprimos** el paquete.

```
master@menzoberranzan:~$ tar xjf xvidcore-1.1.0-beta2.tar.bz2
master@menzoberranzan:~$
```

Para **compilar XviD**, el proceso es muy similar al que deberemos seguir con MPlayer, aunque bastante más **simplificado** (por lo que nos puede servir como un buen entrenamiento). En primer lugar debemos **ejecutar el script de autoconfiguración**.

```
master@menzoberranzan:~$ cd xvidcore-1.1.0-beta2/build/generic/
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$ ./configure
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking whether to use default CFLAGS... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for a BSD-compatible install... /usr/bin/install -c
checking for ranlib... ranlib
checking for whether to use assembly code... yes
checking for architecture type... ia32
checking how to run the C preprocessor... gcc -E
checking for egrep... grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for int *... yes
checking size of int *... 4
checking whether byte ordering is bigendian... no
checking for build extensions... .so .a .o
checking for platform specific LDFLAGS/CFLAGS... ok
checking for yasm... no
checking for nasm... yes
checking for nasm patch version... 38
checking for asm object format... elf
checking stdio.h usability... yes
checking stdio.h presence... yes
checking for stdio.h... yes
checking signal.h usability... yes
```

```
checking signal.h presence... yes
checking for signal.h... yes
checking for stdlib.h... (cached) yes
checking for string.h... (cached) yes
checking assert.h usability... yes
checking assert.h presence... yes
checking for assert.h... yes
checking math.h usability... yes
checking math.h presence... yes
checking for math.h... yes
configure: creating ./config.status
config.status: creating platform.inc
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$
```

Si no hay ningún problema, vuestra salida por consola debería ser muy similar a la mía. En el caso de que vuestra máquina no dispusiera de alguno de los componentes necesarios para poder compilar XviD, el script de configuración **lanzaría un aviso** con el problema encontrado. Una pequeña visita a google o a cualquier foro os podrá ayudar a solventar ese problema, puesto que aquí no puedo tratar todos los que pudieran existir (que son muchos).

Ahora es el momento de iniciar la **compilación** propiamente dicha de XviD.

```
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$ make
D: =build
C: ./decoder.c
C: ./encoder.c
C: ./xvid.c
C: bitstream/bitstream.c
C: bitstream/cbp.c
C: bitstream/mbcoding.c
C: dct/fdct.c
C: dct/idct.c
C: dct/simple_idct.c
C: image/colorspace.c
C: image/image.c
C: image/interpolate8x8.c
C: image/font.c
C: image/postprocessing.c
C: image/qpel.c
C: image/reduced.c
C: motion/estimation_bvop.c
C: motion/estimation_common.c
C: motion/estimation_gmc.c
C: motion/estimation_pvop.c
C: motion/estimation_rd_based.c
C: motion/estimation_rd_based_bvop.c
C: motion/gmc.c
C: motion/motion_comp.c
C: motion/vop_type_decision.c
C: motion/sad.c
C: prediction/mbprediction.c
C: plugins/plugin_single.c
C: plugins/plugin_2pass1.c
C: plugins/plugin_2pass2.c
C: plugins/plugin_lumimasking.c
C: plugins/plugin_dump.c
C: plugins/plugin_psnr.c
C: quant/quant_h263.c
C: quant/quant_matrix.c
```

```
C: quant/quant_mpeg.c
C: utils/emms.c
C: utils/mbtransquant.c
C: utils/mem_align.c
C: utils/mem_transfer.c
C: utils/timer.c
A: bitstream/x86_asm/cbp_3dne.asm
A: bitstream/x86_asm/cbp_mmx.asm
A: bitstream/x86_asm/cbp_sse2.asm
A: dct/x86_asm/fdct_mmx_ffmpeg.asm
A: dct/x86_asm/fdct_mmx_skal.asm
A: dct/x86_asm/fdct_sse2_skal.asm
A: dct/x86_asm/idct_3dne.asm
A: dct/x86_asm/idct_mmx.asm
A: dct/x86_asm/idct_sse2_dmitry.asm
A: dct/x86_asm/simple_idct_mmx.asm
A: image/x86_asm/colospace_rgb_mmx.asm
A: image/x86_asm/colospace_yuv_mmx.asm
A: image/x86_asm/colospace_yuyv_mmx.asm
A: image/x86_asm/interpolate8x8_3dn.asm
A: image/x86_asm/interpolate8x8_3dne.asm
A: image/x86_asm/interpolate8x8_mmx.asm
A: image/x86_asm/interpolate8x8_xmm.asm
A: image/x86_asm/postprocessing_mmx.asm
A: image/x86_asm/postprocessing_sse2.asm
A: image/x86_asm/reduced_mmx.asm
A: image/x86_asm/qpel_mmx.asm
A: motion/x86_asm/sad_xmm.asm
A: motion/x86_asm/sad_sse2.asm
A: motion/x86_asm/sad_mmx.asm
A: motion/x86_asm/sad_3dne.asm
A: motion/x86_asm/sad_3dn.asm
A: quant/x86_asm/quantize_h263_mmx.asm
A: quant/x86_asm/quantize_h263_3dne.asm
A: quant/x86_asm/quantize_mpeg_xmm.asm
A: quant/x86_asm/quantize_mpeg_mmx.asm
A: utils/x86_asm/mem_transfer_mmx.asm
A: utils/x86_asm/mem_transfer_3dne.asm
A: utils/x86_asm/interlacing_mmx.asm
A: utils/x86_asm/cpuid.asm
L: libxvidcore.a
L: libxvidcore.so.4.1
```

XviD has been successfully built.

** Binaries are currently located in the '=build' directory
* To install them on your system, you can run '# make install'
as root.*

```
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$
```

Bien, si no ha habido ningún problema, veremos el mensaje que nos dice que **XviD ha sido compilado correctamente**. Ahora sólo nos resta **instalarlo** en el sistema.

```
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$ su
Password:
menzoberranzan:/home/master/xvidcore-1.1.0-beta2/build/generic# make install
D: /usr/local/lib
I: /usr/local/lib/libxvidcore.so.4.1
I: /usr/local/lib/libxvidcore.a
D: /usr/local/include
I: /usr/local/include/xvid.h
menzoberranzan:/home/master/xvidcore-1.1.0-beta2/build/generic# exit
exit
master@menzoberranzan:~/xvidcore-1.1.0-beta2/build/generic$
```

Ahora tenemos XviD compilado para nuestra máquina e instalado en nuestro sistema. Ya podemos empezar a tantear a MPlayer.

Configuración (I)

Ahora que ya tenemos en nuestro sistema los **principales codecs**, es el momento de realizar el primer intento de configuración de MPlayer, invocando el script de autoconfiguración. Los diversos **parámetros** con los que vamos a invocarlo son:

- `--enable-gui` - **Activa** la compilación de **gmpayer**, la interfaz gráfica de MPlayer basada en **GTK**.
- `--enable-largefiles` - **Activa** el soporte para **ficheros mayores de 2 GB**. Si disponéis de una tarjeta capturadora de televisión y grabáis habitualmente programas con ella, los ficheros generados serán en muchas ocasiones de gran tamaño.
- `--with-codecsdir=/usr/local/lib/codecs/` - Indicamos al script la ruta en la que están los **codecs**.
- `--with-win32libdir=/usr/local/lib/codecs/` - Indicamos al script la ruta en la que están las **librerías de Windows**.
- `--with-xvidlibdir=/usr/local/lib/` - Indicamos al script la ruta al fichero **libxvidcore.a**.
- `--with-xvidincdir=/usr/local/include/` - Indicamos al script la ruta al fichero **xvid.h**.

Otras opciones de compilación interesantes, pero que **no vamos a usar**, serían:

- `--disable-mencoder` - **Desactiva** el soporte de **mencoder** (una utilidad de MPlayer para codificación de vídeo). Esta opción puede disminuir en gran medida el tiempo de compilación, y si no vamos a utilizar mencoder, es una opción bastante recomendable.
- `--disable-tv` - Desactiva la **interfaz de televisión**. Igual que en el caso anterior, si no vamos a utilizarla, es muy recomendable desactivarla.
- `--disable-network` - Si vamos a instalar MPlayer en un equipo con acceso a la red y deseamos evitar **riesgos de seguridad innecesarios**, es conveniente desactivar el soporte de red.
- `--language=es` - Si deseamos disponer del manual (**manpage**) traducido al castellano, debemos incluir este parámetro. Yo, sinceramente, uso la mayoría del software en inglés, por lo que muchas veces ni me molestó en activar el soporte en castellano.
- `--cc=[ruta_al_compilador]` - Si deseamos utilizar un **compilador** distinto al estándar en nuestro sistema, debemos invocar este parámetro. Mucha gente prefiere utilizar la serie de **GCC 2.9.5** para compilar aplicaciones importantes (como el **kernel**), aunque yo en muy raras ocasiones he tenido problemas con otras versiones de este magnífico compilador.

Bien, ahora que sabemos cómo vamos a invocar al script de configuración, es el momento de lanzarse a la piscina.

```
master@menzoberranzan:~/MPlayer-1.0pre7$ ./configure --enable-gui --enable-largefiles --with-
codecsdir=/usr/local/lib/codecs/ --with-win32libdir=/usr/local/lib/codecs/ --with-xvidlibdir=/usr/local/lib/
--with-xvidincdir=/usr/local/include/
Detected operating system: Linux
Detected host architecture: i386
Checking for cc version ... 3.3.5, ok
Checking for host cc ... cc
Checking for CPU vendor ... GenuineIntel (15:2:4)
Checking for CPU type ... Intel(R) Pentium(R) 4 CPU 2.20GHz
Checking for GCC & CPU optimization abilities ... pentium4
Checking for kernel support of mmx ... yes
Checking for kernel support of mmx2 ... yes
Checking for kernel support of sse ... yes
Checking for kernel support of sse2 ... yes
Checking for mtrr support ... yes
Checking for assembler support of -pipe option ... yes
Checking for assembler (as 2.15) ... ok
Checking for Linux kernel version ... 2.6.7, ok
Checking for mplayer binary name ... mplayer
Checking for awk ... mawk
```

Checking for extra headers ... none
Checking for extra libs ... none
Checking for -lposix ... no
Checking for -lm ... yes
Checking for i18n ... yes
Checking for iconv ... yes
Checking for langinfo ... yes
Checking for language ... using en (man pages: en)
Checking for enable sighandler ... yes
Checking for runtime cpudetection ... no
Checking for restrict keyword ... __restrict
Checking for __builtin_expect ... yes
Checking for kstat ... no
Checking for posix4 ... no
Checking for lrintf ... yes
Checking for nanosleep ... yes
Checking for socklib ... yes (using -lnsl)
Checking for inet_pton() ... yes (using -lnsl)
Checking for inttypes.h (required) ... yes
Checking for int_fastXY_t in inttypes.h ... yes
Checking for word size ... 32
Checking for stddef.h ... yes
Checking for malloc.h ... yes
Checking for memalign() ... yes
Checking for alloca.h ... yes
Checking for mman.h ... yes
Checking for dynamic loader ... yes
Checking for dynamic a/v plugins support ... no
Checking for pthread ... yes (using -lpthread)
Checking for sys/soundcard.h ... yes
Checking for sys/dvdio.h ... no
Checking for sys/cdio.h ... no
Checking for linux/cdrom.h ... yes
Checking for dvd.h ... no
Checking for BSDI dvd.h ... no
Checking for HPUX SCSI header ... no
Checking for userspace SCSI headers (Solaris) ... no
Checking for termcap ... yes (using -ltermcap)
Checking for termios ... yes (using sys/termios.h)
Checking for shm ... yes
Checking for linux devfs ... no
Checking for scandir() ... yes
Checking for strsep() ... yes
Checking for strlcpy() ... no
Checking for strlcat() ... no
Checking for fseeko() ... yes
Checking for localtime_r() ... yes
Checking for vsscanf() ... yes
Checking for swab() ... yes
Checking for posix select() ... yes
Checking for gettimeofday() ... yes
Checking for glob() ... yes
Checking for sys/sysinfo.h ... yes
Checking for Mac OS X APIs ... no
Checking for Mac OS X Finder Support ... no
Checking for Mac OS X Bundle file locations ... no
Checking for Samba support (libsmbclient) ... no
Checking for 3dfx ... no
Checking for tdfxfb ... no

Checking for tdfxvid ... no
Checking for tga ... yes
Checking for DirectFB headers presence ... not found
Checking for DirectFB ... no
Checking for X11 headers presence ... yes (using /usr/X11R6/include)
Checking for X11 libs presence ... yes (using /usr/X11R6/lib)
Checking for X11 ... yes
Checking for DPMS ... yes (using Xdpms 4)
Checking for Xv ... yes
Checking for XvMC ... no
Checking for Xinerama ... no
Checking for Xxf86vm ... no
Checking for XF86keysym ... yes
Checking for DGA ... no
Checking for OpenGL ... yes
Checking for /dev/mga_vid ... no
Checking for xmga ... no
Checking for GGI ... no
Checking for AA ... no
Checking for CACA ... no
Checking for SVGAlib ... no
Checking for FBDev ... yes
Checking for DVB ... no (specify path to DVB/ost/include with --with-dvbinclidir=DIR)
Checking for DVB HEAD ... yes
Checking for PNG support ... yes
Checking for JPEG support ... no
Checking for PNM support ... yes
Checking for md5sum support ... yes
Checking for GIF support ... no
Checking for VESA support ... yes
Checking for SDL ... no
Checking for Windows waveout ... no
Checking for Directx ... no
Checking for NAS ... no
Checking for DXR2 ... no
Checking for DXR3/H+ ... no
Checking for libmp1e ... no
Checking for libfame ... no
Checking for OSS Audio ... yes
Checking for aRts ... no
Checking for Esound ... no
Checking for Polyp ... no
Checking for JACK ... no
Checking for ALSA audio ... yes (using alsa 1.0.x and alsa/asoundlib.h)
Checking for Sun audio ... no
Checking for Sun mediaLib ... no
Checking for SGI audio ... no
Checking for VCD support ... ok
Checking for DVD support (libmpdvdkit) ... yes
Checking for DVD support (libdvdread) ... disabled by libmpdvdkit2
Checking for cdparanoia ... no
Checking for freetype >= 2.0.9 ... no
Checking for fontconfig ... no
Checking for fribidi with charsets ... no
Checking for ENCA ... no
Checking for zlib ... yes
Checking for RTC ... yes
Checking for external liblzo support ... no
Checking for mad support ... no

Checking for Toolame ... no
Checking for OggVorbis support ... yes (internal Tremor)
Checking for OggTheora support (only the CVS version!) ... no
Checking for mp3lib support ... yes
Checking for liba52 support ... yes
Checking for libdts support ... no
Checking for libmpeg2 support ... yes
Checking for Matroska support ... yes
Checking for internal FAAD2 (AAC) support ... yes
Checking for external FAAD2 (AAC) support ... no
Checking for LADSPA plugin support ... no
Checking for Win32 codec DLL support ... yes (using /usr/local/lib/codecs/)
Checking for Win32 loader support ... yes
Checking for DirectShow ... yes
Checking for XAnim DLL ... yes (using /usr/local/lib/codecs/)
Checking for RealPlayer DLL ... yes (using /usr/local/lib/codecs/)
Checking for LIVE.COM Streaming Media libraries ... no
Checking for FFmpeg libavcodec (static) ... yes
Checking for FFmpeg libavformat (static) ... yes
Checking for amr narrowband ... no
Checking for amr narrowband, fixed point ... no
Checking for amr wideband ... no
Checking for libdv-0.9.5+ ... no
Checking for zr ... no
Checking for bl ... no
Checking for XviD ... yes
Checking for DivX4 compatibility in XviD ... no
Checking for x264 ... no
Checking for DivX4linux/DivX5linux/OpenDivX decore ... no
Checking for libmp3lame (for mencoder) ... no
Checking for DivX4linux encode (for mencoder) ... no
Checking for mencoder ... yes
Checking for fastmemcpy ... yes
Checking for UniquE RAR File Library ... yes
Checking for TV interface ... yes
Checking for EDL support ... yes
Checking for *BSD BrookTree 848 TV interface ... no
Checking for Video 4 Linux TV interface ... no
Checking for Video 4 Linux 2 TV interface ... no
Checking for audio select() ... yes
Checking for network ... yes
Checking for ftp ... yes
Checking for vstream client ... no
Checking for byte order ... little-endian
Checking for shared postprocess lib ... no
Checking for OSD menu ... no
Checking for QTX codecs ... yes
Checking for Subtitles sorting ... yes
Checking for XMMS inputplugin support ... no
Checking for inet6 ... yes
Checking for gethostbyname2 ... yes
Checking for GUI ... yes
Checking for XShape extension ... yes
Checking for GTK version ... 1.2.10 (using gtk-config)
Checking for glib version ... 1.2.10 (using glib-config)
Creating Gui/config.mak
Checking for automatic gdb attach ... no
Checking for compiler support for -fno-PIC ... yes
Checking for ftello() ... yes

```
Checking for VIDIX ... yes
Checking for joystick ... no
Checking for lirc ... no
Checking for lircc ... no
Creating config.mak
Creating config.h
Creating libvo/config.mak
Creating libao2/config.mak
Creating libaf/config.mak
```

Config files successfully generated by ./configure !

```
Install prefix: /usr/local
Data directory: /usr/local/share/mplayer
Config direct.: /usr/local/etc/mplayer
```

```
Byte order: little-endian
Optimizing for: pentium4 mmx mmx2 sse sse2 mtrr
```

```
Languages:
Messages/GUI: en
Manual pages: en
```

Enabled optional drivers:

```
Input: ftp network edl tv matroska mpdvdkit2 vcd dvb
Codecs: qtx xvid libavcodec real xanim dshow/dmo win32 faad2(internal) libmpeg2 liba52 mp3lib
tremor(internal)
```

```
Audio output: alsa oss mpegpes(dvb)
```

```
Video output: xvixidix cvidix vesa md5sum pnm png mpegpes(dvb) fbdev opengl xv x11 xover tga
```

```
Audio filters:
```

Disabled optional drivers:

```
Input: vstream tv-v4l2 tv-v4l tv-bsdbt848 live.com cdda dvdread smb
```

```
Codecs: opendivx x264 libdv amr_wb amr_nb libdts libtheora toolame libmad liblzo gif
```

```
Audio output: sgi sun jack polyp esd arts dxr2 nas dsound win32 sdl macosx
```

```
Video output: winvidix bl zr zr2 dxr3 dxr2 directx sdl gif89a jpeg svga caca aa ggi ximga mga dga
```

```
xvmc directfb tdfx_vid tdfxfb 3dfx quartz
```

```
Audio filters: ladspa
```

'config.h' and 'config.mak' contain your configuration options.

Note: If you alter these files (for instance CFLAGS) MPlayer may no longer compile *** DO NOT REPORT BUGS if you tweak these files ***

'make' will now compile MPlayer and 'make install' will install it.

Note: On non-Linux systems you might need to use 'gmake' instead of 'make'.

Please check mtrr settings at /proc/mtrr (see DOCS/HTML/en/video.html#mtrr)

Check configure.log if you wonder why an autodetection failed (check whether the development headers/packages are installed).

If you suspect a bug, please read DOCS/HTML/en/bugreports.html.

```
master@menzoberranzan:~/MPlayer-1.0pre7$
```

Se ha configurado correctamente, y ahora mismo **podríamos compilar** el software sin problemas... pero no lo haremos. Vamos a **afinar aún más** la compilación.

Atando cabos sueltos

Si echamos un ojo a la parte final de la salida por pantalla de la configuración, veremos **qué drivers han sido o no activados** para la compilación.

```

Enabled optional drivers:
Input: ftp network edl tv matroska mpdvdkit2 vcd dvb
Codecs: qtx xvid libavcodec real xanim dshow/dmo win32 faad2(internal) libmpeg2 liba52 mp3lib
tremor(internal)
Audio output: alsa oss mpegpes(dvb)
Video output: xvixidix cvidix vesa md5sum pnm png mpegpes(dvb) fbdev opengl xv x11 xover tga
Audio filters:
Disabled optional drivers:
Input: vstream tv-v4l2 tv-v4l tv-bsdbt848 live.com cdda dvdread smb
Codecs: opendivx x264 libdv amr_wb amr_nb libdts libtheora toolame libmad liblzo gif
Audio output: sgi sun jack polyp esd arts dvr2 nas dsound win32 sdl macosx
Video output: winvidix bl zr zr2 dvr3 dvr2 directx sdl gif89a jpeg svga caca aa ggi xmga mga dga
xvmc directfb tdfx_vid tdfxfb 3dfx quartz
Audio filters: ladspa

```

Como podemos comprobar, la mayoría de los desactivados no nos interesan demasiado... pero **algunos de ellos sí**. E incluso puede que a alguno de vosotros le interesen todos o casi todos. Es el momento de afinar y añadir soporte para aquellas cosas que no tengan el soporte activado y deseemos que lo tengan.

Llegados a este punto, la cantidad de drivers o codecs a activar depende completamente de las preferencias de cada usuario. Incluso únicamente con los codecs estándar que ya hemos activado, sería **más que suficiente** para la práctica totalidad del uso que vayamos a darle a MPlayer.

En mi caso he activado unos cuantos más **a modo de ejemplo**, utilizando la utilidad APT de la distribución Debian.

```

menzoberranzan:/home/master# apt-get install libsvga1-dev libjpeg62-dev giflib3g giflib3g-dev
libmpeg-dev libmpeg2-4 libmpeg2-4-dev libmpeg3-1 libmpeg3-dev libtiff2 libtiff-dev toolame libtheora0
libdts-dev libdv4 libdv4-dev libdvdnav4 libdvdnav-dev avifile-divx-plugin
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
libavifile-0.7c102
Suggested packages:
avifile-player avifile-utils avifile-mad-plugin avifile-mjpeg-plugin avifile-vorbis-plugin
avifile-win32-plugin avifile-xvid-plugin libmpeg-doc libfreetype6-dev freetype1-tools
Recommended packages:
libdv-bin
The following NEW packages will be installed:
avifile-divx-plugin giflib3g giflib3g-dev libavifile-0.7c102 libdts-dev libdv4 libdv4-dev
libdvdnav-dev libdvdnav4 libjpeg62-dev libmpeg-dev libmpeg2-4 libmpeg2-4-dev libmpeg3-1
libmpeg3-dev libsvga1-dev libtheora0 libtiff-dev libtiff2 toolame
0 upgraded, 20 newly installed, 0 to remove and 591 not upgraded.
Need to get 4242kB of archives.
After unpacking 11.8MB of additional disk space will be used.
Do you want to continue? [Y/n] y
[...]
menzoberranzan:/home/master#

```

Como ya he dicho, estos paquetes instalados **son únicamente un ejemplo** para que veamos la diferencia entre la primera configuración y la segunda. Algunos de los drivers para los que no he dado soporte son **altamente recomendables** de instalar.

Dado que para poder seguir el desarrollo del manual de la forma más natural posible necesitaba un equipo que no tuviera MPlayer correctamente compilado e instalado, elegí uno que prácticamente no uso para ver vídeos (**menzoberranzan**) y que disponía de una versión bastante más antigua de MPlayer compilada, así como soporte para **muy pocos** drivers. A modo de ejemplo, tenía instalado y configurado el soporte **OSD** antiguo mediante **fuentes truetype**, mientras que lo más recomendable (y lo que tengo configurado en el equipo que utilizo habitualmente para ver vídeos **-blingdenstone-**) es utilizar **freetype**. Adicionalmente, la existencia de paquetes rotos en la estructura de Debian SID en el momento de realizar el manual, ha hecho que deje en el tintero soportes tan importantes como **dvdread**, **libtheora**, **toolame**, **sdl**, **caca**, **aa**...

Por todo ello, esta parte no debe ser tomada al pie de la letra, sino únicamente como una guía para entender cómo activar el soporte de un driver que no lo tiene en vuestro propio sistema. Además, he utilizado el sistema de paquetes de **APT de Debian**, por lo que otras distribuciones (como **SuSE**, **Fedora**, **Mandriva**...) tendrán que utilizar métodos diferentes.

Algunos de los soportes que considero altamente recomendables son: **libdvdcss**, **libdvdread**, **libvorbis**, **libogg**, **toolame**...

Configuración (II)

Una vez activado el soporte para todos los drivers que deseamos utilizar, es el momento de realizar el **segundo intento de configuración** de MPlayer.

```
master@menzoberranzan:~/MPlayer-1.0pre7$ ./configure --enable-gui --enable-largefiles --with-
codecsdir=/usr/local/lib/codecs/ --with-win32libdir=/usr/local/lib/codecs/ --with-xvidlibdir=/usr/local/lib/
--with-xvidincdir=/usr/local/include/
Detected operating system: Linux
Detected host architecture: i386
Checking for cc version ... 3.3.5, ok
Checking for host cc ... cc
Checking for CPU vendor ... GenuineIntel (15:2:4)
Checking for CPU type ... Intel(R) Pentium(R) 4 CPU 2.20GHz
Checking for GCC & CPU optimization abilities ... pentium4
Checking for kernel support of mmx ... yes
Checking for kernel support of mmx2 ... yes
Checking for kernel support of sse ... yes
Checking for kernel support of sse2 ... yes
Checking for mtrr support ... yes
Checking for assembler support of -pipe option ... yes
Checking for assembler (as 2.15) ... ok
Checking for Linux kernel version ... 2.6.7, ok
Checking for mplayer binary name ... mplayer
Checking for awk ... mawk
Checking for extra headers ... none
Checking for extra libs ... none
Checking for -lposix ... no
Checking for -lm ... yes
Checking for i18n ... yes
Checking for iconv ... yes
Checking for langinfo ... yes
Checking for language ... using en (man pages: en)
Checking for enable sighandler ... yes
Checking for runtime cpudetection ... no
Checking for restrict keyword ... __restrict
Checking for __builtin_expect ... yes
Checking for kstat ... no
Checking for posix4 ... no
Checking for lrintf ... yes
Checking for nanosleep ... yes
Checking for socklib ... yes (using -lnsl)
Checking for inet_pton() ... yes (using -lnsl)
Checking for inttypes.h (required) ... yes
Checking for int_fastXY_t in inttypes.h ... yes
Checking for word size ... 32
Checking for stddef.h ... yes
Checking for malloc.h ... yes
Checking for memalign() ... yes
Checking for alloca.h ... yes
Checking for mman.h ... yes
Checking for dynamic loader ... yes
Checking for dynamic a/v plugins support ... no
Checking for pthread ... yes (using -lpthread)
Checking for sys/soundcard.h ... yes
Checking for sys/dvdio.h ... no
Checking for sys/cdio.h ... no
```


Checking for linux/cdrom.h ... yes
Checking for dvd.h ... no
Checking for BSDI dvd.h ... no
Checking for HPUX SCSI header ... no
Checking for userspace SCSI headers (Solaris) ... no
Checking for termcap ... yes (using -ltermcap)
Checking for termios ... yes (using sys/termios.h)
Checking for shm ... yes
Checking for linux devfs ... no
Checking for scandir() ... yes
Checking for strsep() ... yes
Checking for strlcpy() ... no
Checking for strlcat() ... no
Checking for fseeko() ... yes
Checking for localtime_r() ... yes
Checking for vsscanf() ... yes
Checking for swab() ... yes
Checking for posix select() ... yes
Checking for gettimeofday() ... yes
Checking for glob() ... yes
Checking for sys/sysinfo.h ... yes
Checking for Mac OS X APIs ... no
Checking for Mac OS X Finder Support ... no
Checking for Mac OS X Bundle file locations ... no
Checking for Samba support (libsmbclient) ... no
Checking for 3dfx ... no
Checking for tdfxfb ... no
Checking for tdfxvid ... no
Checking for tga ... yes
Checking for DirectFB headers presence ... not found
Checking for DirectFB ... no
Checking for X11 headers presence ... yes (using /usr/X11R6/include)
Checking for X11 libs presence ... yes (using /usr/X11R6/lib)
Checking for X11 ... yes
Checking for DPMS ... yes (using Xdpms 4)
Checking for Xv ... yes
Checking for XvMC ... no
Checking for Xinerama ... no
Checking for Xxf86vm ... no
Checking for XF86keysym ... yes
Checking for DGA ... no
Checking for OpenGL ... yes
Checking for /dev/mga_vid ... no
Checking for xmga ... no
Checking for GGI ... no
Checking for AA ... no
Checking for CACA ... no
Checking for SVGAlib ... yes
Checking for FBDev ... yes
Checking for DVB ... no (specify path to DVB/ost/include with --with-dvbinclidir=DIR)
Checking for DVB HEAD ... yes
Checking for PNG support ... yes
Checking for JPEG support ... yes
Checking for PNM support ... yes
Checking for md5sum support ... yes
Checking for GIF support ... yes (old version, some encoding functions disabled)
Checking for broken giflib workaround ... enabled
Checking for VESA support ... yes
Checking for SDL ... no

Checking for Windows waveout ... no
Checking for Directx ... no
Checking for NAS ... no
Checking for DXR2 ... no
Checking for DXR3/H+ ... no
Checking for libmp1e ... no
Checking for libfame ... no
Checking for OSS Audio ... yes
Checking for aRts ... no
Checking for Esound ... no
Checking for Polyp ... no
Checking for JACK ... no
Checking for ALSA audio ... yes (using alsa 1.0.x and alsa/asoundlib.h)
Checking for Sun audio ... no
Checking for Sun mediaLib ... no
Checking for SGI audio ... no
Checking for VCD support ... ok
Checking for DVD support (libmpdvdkit) ... yes
Checking for DVD support (libdvdread) ... disabled by libmpdvdkit2
Checking for cdparanoia ... no
Checking for freetype >= 2.0.9 ... no
Checking for fontconfig ... no
Checking for fribidi with charsets ... no
Checking for ENCA ... no
Checking for zlib ... yes
Checking for RTC ... yes
Checking for external liblzo support ... no
Checking for mad support ... no
Checking for Toolame ... no
Checking for OggVorbis support ... yes (internal Tremor)
Checking for OggTheora support (only the CVS version!) ... no
Checking for mp3lib support ... yes
Checking for liba52 support ... yes
Checking for libdts support ... yes
Checking for libmpeg2 support ... yes
Checking for Matroska support ... yes
Checking for internal FAAD2 (AAC) support ... yes
Checking for external FAAD2 (AAC) support ... no
Checking for LADSPA plugin support ... no
Checking for Win32 codec DLL support ... yes (using /usr/local/lib/codecs/)
Checking for Win32 loader support ... yes
Checking for DirectShow ... yes
Checking for XAnim DLL ... yes (using /usr/local/lib/codecs/)
Checking for RealPlayer DLL ... yes (using /usr/local/lib/codecs/)
Checking for LIVE.COM Streaming Media libraries ... no
Checking for FFmpeg libavcodec (static) ... yes
Checking for FFmpeg libavformat (static) ... yes
Checking for amr narrowband ... no
Checking for amr narrowband, fixed point ... no
Checking for amr wideband ... no
Checking for libdv-0.9.5+ ... yes
Checking for zr ... no
Checking for bl ... no
Checking for XviD ... yes
Checking for DivX4 compatibility in XviD ... no
Checking for x264 ... no
Checking for DivX4linux/DivX5linux/OpenDivX decore ... no
Checking for libmp3lame (for mencoder) ... no
Checking for DivX4linux encore (for mencoder) ... no

```
Checking for mencoder ... yes
Checking for fastmemcpy ... yes
Checking for Unique RAR File Library ... yes
Checking for TV interface ... yes
Checking for EDL support ... yes
Checking for *BSD BrookTree 848 TV interface ... no
Checking for Video 4 Linux TV interface ... no
Checking for Video 4 Linux 2 TV interface ... no
Checking for audio select() ... yes
Checking for network ... yes
Checking for ftp ... yes
Checking for vstream client ... no
Checking for byte order ... little-endian
Checking for shared postprocess lib ... no
Checking for OSD menu ... no
Checking for QTX codecs ... yes
Checking for Subtitles sorting ... yes
Checking for XMMS inputplugin support ... no
Checking for inet6 ... yes
Checking for gethostbyname2 ... yes
Checking for GUI ... yes
Checking for XShape extension ... yes
Checking for GTK version ... 1.2.10 (using gtk-config)
Checking for glib version ... 1.2.10 (using glib-config)
Creating Gui/config.mak
Checking for automatic gdb attach ... no
Checking for compiler support for -fno-PIC ... yes
Checking for ftello() ... yes
Checking for VIDIX ... yes
Checking for joystick ... no
Checking for lirc ... no
Checking for lircc ... no
Creating config.mak
Creating config.h
Creating libvo/config.mak
Creating libao2/config.mak
Creating libaf/config.mak
```

Config files successfully generated by ./configure !

```
Install prefix: /usr/local
Data directory: /usr/local/share/mplayer
Config direct.: /usr/local/etc/mplayer
```

```
Byte order: little-endian
Optimizing for: pentium4 mmx mmx2 sse sse2 mtrr
```

```
Languages:
Messages/GUI: en
Manual pages: en
```

```
Enabled optional drivers:
Input: ftp network edl tv matroska mpdvdkit2 vcd dvb
Codecs: qtx xvid libdv libavcodec real xanim dshow/dmo win32 faad2(internal) libmpeg2 libdts
liba52 mp3lib tremor(internal) gif
Audio output: alsa oss mpegpes(dvb)
Video output: xvixidix cvidix vesa gif89a md5sum pnm jpeg png mpegpes(dvb) fbdev svga opengl xv
x11 xover tga
Audio filters:
```

Disabled optional drivers:
Input: vstream tv-v4l2 tv-v4l tv-bsdbt848 live.com cdda dvdread smb
Codecs: opendivx x264 amr_wb amr_nb libtheora toolame libmad liblzo
Audio output: sgi sun jack polyp esd arts dxr2 nas dsound win32 sdl macosx
Video output: winvidix bl zr zr2 dxr3 dxr2 directx sdl caca aa ggi xmga mga dga xvmc directfb
tdfx_vid tdfxfb 3dfx quartz
Audio filters: ladspa

'config.h' and 'config.mak' contain your configuration options.
*Note: If you alter these files (for instance CFLAGS) MPlayer may no longer compile *** DO NOT REPORT BUGS if you tweak these files ****

'make' will now compile MPlayer and 'make install' will install it.
Note: On non-Linux systems you might need to use 'gmake' instead of 'make'.

Please check mtrr settings at /proc/mtrr (see DOCS/HTML/en/video.html#mtrr)

Check configure.log if you wonder why an autodetection failed (check whether the development headers/packages are installed).

If you suspect a bug, please read DOCS/HTML/en/bugreports.html.

master@menzoberranzan:~/MPlayer-1.0pre7\$

Como vemos, la lista de drivers activados y desactivados **ha cambiado**. Gracias a ello, ahora nuestro **MPlayer podrá gestionar una mayor cantidad de formatos** de lo que normalmente podría. Si deseáramos añadir soporte para nuevos drivers, sólo habría que repetir los pasos del punto anterior: **identificar el driver** opcional deseado, **buscarlo** (bien mediante algún software de gestión de paquetes de nuestra distribución, bien mediante Internet) e **instalarlo en nuestro sistema**.

Ha llegado el momento de la compilación propiamente dicha.

Compilación e instalación

Para **compilar MPlayer** una vez configurado, simplemente tenemos que invocar el **comando make**, que se encargará de leer el **fichero makefile** y compilar el software según las reglas definidas en él. Todo el proceso se realiza de forma **automática**.

```
master@menzoberranzan:~/MPlayer-1.0pre7$ make
./version.sh `cc -dumpversion`
make distclean
make[1]: Entering directory `/home/master/MPlayer-1.0pre7'
```

[...]

```
libinput.a postproc/libswscale.a osdep/libosdep.a -Llibmpdvdkit2 -lmpdvdkit libavcodec/libavcodec.a
libavformat/libavformat.a -ldv -L/usr/local/lib/ -lxxvidcore -lm -ldts -lm -lpng -lz -ljpeg -lasound
-ldl -lpthread -ltermcap -lnsl -lungif libfaad2/libfaad2.a mp3lib/libMP3.a liba52/liba52.a
libmpeg2/libmpeg2.a tremor/libvorbisidec.a -lpthread -ldl -rdynamic -lm
master@menzoberranzan:~/MPlayer-1.0pre7$
```

Dado que la salida por pantalla es **enorme** (e innecesaria si no existe ningún error de compilación), no la he pegado completa. Si **no recibimos ningún error** antes de finalizar, la compilación ha sido correcta y MPlayer se encuentra listo para funcionar. Es el momento de **instalarlo** en nuestro sistema.

```
master@menzoberranzan:~/MPlayer-1.0pre7$ su
Password:
menzoberranzan:/home/master/MPlayer-1.0pre7# make install
make[1]: Entering directory `/home/master/MPlayer-1.0pre7/libdha'
mkdir -p /usr/local/lib
install -m 755 -s -p libdha.so.1.0 /usr/local/lib/libdha.so.1.0
rm -f /usr/local/lib/libdha.so
ln -sf libdha.so.1.0 /usr/local/lib/libdha.so.1
ldconfig
ldconfig: /usr/X11R6/lib/libXvMCNVIDIA.so.1.0.5336 is not an ELF file - it has the wrong magic bytes
at the start.
```

```
make[1]: Leaving directory `/home/master/MPlayer-1.0pre7/libdha'
make[1]: Entering directory `/home/master/MPlayer-1.0pre7/vidix'
make[2]: Entering directory `/home/master/MPlayer-1.0pre7/vidix/drivers'
mkdir -p /usr/local/lib/mplayer/vidix
install -m 755 -s -p *.so /usr/local/lib/mplayer/vidix
make[2]: Leaving directory `/home/master/MPlayer-1.0pre7/vidix/drivers'
make[1]: Leaving directory `/home/master/MPlayer-1.0pre7/vidix'
if test ! -d /usr/local/bin ; then mkdir -p /usr/local/bin ; fi
install -m 755 -s mplayer /usr/local/bin/mplayer
ln -sf mplayer /usr/local/bin/gmplayer
if test ! -d /usr/local/man/man1 ; then mkdir -p /usr/local/man/man1 ; fi
for i in en; do \
    if test "$i" = en ; then \
        install -c -m 644 DOCS/man/en/mplayer.1 /usr/local/man/man1/mplayer.1 ; \
    else \
        mkdir -p /usr/local/man/$i/man1 ; \
        install -c -m 644 DOCS/man/$i/mplayer.1 /usr/local/man/$i/man1/mplayer.1 ; \
    fi ; \
done
install -m 755 -s mencoder /usr/local/bin/mencoder
for i in en; do \
    if test "$i" = en ; then \
```

```

        ln -sf mplayer.1 /usr/local/man/man1/mencoder.1 ; \
    else \
        ln -sf mplayer.1 /usr/local/man/$i/man1/mencoder.1 ; \
    fi ; \
done
*** Download skin(s) at http://www.mplayerhq.hu/homepage/dload.html
*** for GUI, and extract to /usr/local/share/mplayer/Skin/
install -m 644 Gui/mplayer/pixmaps/mplayer-desktop.xpm /usr/local/share/pixmaps/mplayer-
desktop.xpm
install -m 644 etc/mplayer.desktop /usr/local/share/applications/mplayer.desktop
menzoberranzan:/home/master/MPlayer-1.0pre7#

```

Si tampoco obtenemos ningún error en la instalación (mucho cuidado, el comando **make install** debe invocarse como **root**), ahora **MPlayer está listo e integrado** en nuestro sistema. Vamos a comprobarlo.

```

master@menzoberranzan:~$ mplayer -v
MPlayer 1.0pre7-3.3.5 (C) 2000-2005 MPlayer Team
CPU: Intel Pentium 4/Xeon/Celeron Foster (Family: 8, Stepping: 4)
Detected cache-line size is 64 bytes
CPUflags: MMX: 1 MMX2: 1 3DNow: 0 3DNow2: 0 SSE: 1 SSE2: 1
Compiled for x86 CPU with extensions: MMX MMX2 SSE SSE2

```

Usage: mplayer [options] [url|path/]filename

Basic options: (complete list in the man page)

```

-vo <drv[:dev]> select video output driver & device ('-vo help' for a list)
-ao <drv[:dev]> select audio output driver & device ('-ao help' for a list)
vcd://<trackno> play (S)VCD (Super Video CD) track (raw device, no mount)
dvd://<titleno> play DVD title from device instead of plain file
-alang/-slang select DVD audio/subtitle language (by 2-char country code)
-ss <timepos> seek to given (seconds or hh:mm:ss) position
-nosound do not play sound
-fs fullscreen playback (or -vm, -zoom, details in the man page)
-x <x> -y <y> set display resolution (for use with -vm or -zoom)
-sub <file> specify subtitle file to use (also see -subfps, -subdelay)
-playlist <file> specify playlist file
-vid x -aid y select video (x) and audio (y) stream to play
-fps x -srate y change video (x fps) and audio (y Hz) rate
-pp <quality> enable postprocessing filter (details in the man page)
-framedrop enable frame dropping (for slow machines)

```

Basic keys: (complete list in the man page, also check input.conf)

```

<- or -> seek backward/forward 10 seconds
up or down seek backward/forward 1 minute
pgup or pgdown seek backward/forward 10 minutes
< or > step backward/forward in playlist
p or SPACE pause movie (press any key to continue)
q or ESC stop playing and quit program
+ or - adjust audio delay by +/- 0.1 second
o cycle OSD mode: none / seekbar / seekbar + timer
* or / increase or decrease PCM volume
z or x adjust subtitle delay by +/- 0.1 second
r or t adjust subtitle position up/down, also see -vf expand

```

*** SEE THE MAN PAGE FOR DETAILS, FURTHER (ADVANCED) OPTIONS AND KEYS ***

```

master@menzoberranzan:~$

```

Maquillando gmplayer

Ahora que ya tenemos compilado MPlayer, necesitamos instalar al menos un **skin** (piel) para **gmplayer** si queremos poder usarlo. En la misma sección de descargas hay un apartado llamado "**Skins**" (<http://www.mplayerhq.hu/homepage/design7/dload.html#skins>) en el que podemos encontrar una buena colección de ellos. En Internet podréis encontrar más, pero para empezar tenemos de sobra.

En primer paso es **descargar el paquete** comprimido con el skin.

```
master@menzoberranzan:~$ wget http://www1.mplayerhq.hu/MPlayer/Skin/neutron-1.5.tar.bz2
--15:41:15-- http://www1.mplayerhq.hu/MPlayer/Skin/neutron-1.5.tar.bz2
=> `neutron-1.5.tar.bz2'
Resolving www1.mplayerhq.hu... 192.190.173.45
Connecting to www1.mplayerhq.hu[192.190.173.45]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 369,346 [application/x-tar]

100%[=====>] 369,346
53.16K/s  ETA 00:00

15:41:22 (52.81 KB/s) - `neutron-1.5.tar.bz2' saved [369346/369346]

master@menzoberranzan:~$
```

Comprobamos -como siempre- los **hashes**...

```
master@menzoberranzan:~$ md5sum neutron-1.5.tar.bz2
c1b100fd07cc915562a6a25e1ea92e1b  neutron-1.5.tar.bz2
master@menzoberranzan:~$ sha1sum neutron-1.5.tar.bz2
10b3e7173d9eefb3bbd68b87bdf4c2d5ed186609  neutron-1.5.tar.bz2
master@menzoberranzan:~$
```

Descomprimos el paquete...

```
master@menzoberranzan:~$ tar xjf neutron-1.5.tar.bz2
master@menzoberranzan:~$
```

Y **movemos el directorio** creado a la ruta donde deben estar los skins (indicada al final de la salida por consola del comando **make install**).

```
master@menzoberranzan:~$ su
Password:
menzoberranzan:/home/master# mv neutron /usr/local/share/mplayer/Skin/Neutron
menzoberranzan:/home/master# exit
exit
master@menzoberranzan:~$
```

Ya tenemos instalado nuestro primer skin: **Neutron**. Debo reconocer que es mi favorito, y el que he usado durante años... aunque últimamente estoy usando otros (en las **versiones viejas de MPlayer** sólo había soporte para un **único skin**). Vamos a instalar un par de skins más para tener donde elegir.

```
master@menzoberranzan:~$ wget http://www1.mplayerhq.hu/MPlayer/Skin/Corelian-1.1.tar.bz2
--15:43:35-- http://www1.mplayerhq.hu/MPlayer/Skin/Corelian-1.1.tar.bz2
=> `Corelian-1.1.tar.bz2'
Resolving www1.mplayerhq.hu... 192.190.173.45
Connecting to www1.mplayerhq.hu[192.190.173.45]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 284,521 [application/x-tar]
```

```
100%[=====] 284,521
53.25K/s ETA 00:00
```

```
15:43:41 (52.87 KB/s) - `Corelian-1.1.tar.bz2' saved [284521/284521]
```

```
master@menzoberranzan:~$ md5sum Corelian-1.1.tar.bz2
efb2c71db53b3c65ce7861af813aa641 Corelian-1.1.tar.bz2
master@menzoberranzan:~$ sha1sum Corelian-1.1.tar.bz2
92b0c29413f4d118befe2a16989baa243fb23a28 Corelian-1.1.tar.bz2
master@menzoberranzan:~$ tar xjf Corelian-1.1.tar.bz2
master@menzoberranzan:~$ su
```

```
Password:
```

```
menzoberranzan:/home/master# mv Corelian /usr/local/share/mplayer/Skin/Corelian
```

```
menzoberranzan:/home/master# exit
```

```
exit
```

```
master@menzoberranzan:~$ wget http://www1.mplayerhq.hu/MPlayer/Skin/QuickSilver-1.0.tar.bz2
--15:45:07-- http://www1.mplayerhq.hu/MPlayer/Skin/QuickSilver-1.0.tar.bz2
=> `QuickSilver-1.0.tar.bz2'
```

```
Resolving www1.mplayerhq.hu... 192.190.173.45
Connecting to www1.mplayerhq.hu[192.190.173.45]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 201,735 [application/x-tar]
```

```
100%[=====] 201,735
53.19K/s ETA 00:00
```

```
15:45:11 (52.81 KB/s) - `QuickSilver-1.0.tar.bz2' saved [201735/201735]
```

```
master@menzoberranzan:~$ md5sum QuickSilver-1.0.tar.bz2
a1c6810821b83bdf38395c32ee3762d3 QuickSilver-1.0.tar.bz2
master@menzoberranzan:~$ sha1sum QuickSilver-1.0.tar.bz2
bb8c38a379818a11b865b2d3ff8858bae8def319 QuickSilver-1.0.tar.bz2
master@menzoberranzan:~$ tar xjf QuickSilver-1.0.tar.bz2
master@menzoberranzan:~$ su
```

```
Password:
```

```
menzoberranzan:/home/master# mv QuickSilver /usr/local/share/mplayer/Skin/QuickSilver
```

```
menzoberranzan:/home/master# exit
```

```
exit
```

```
master@menzoberranzan:~$
```

Ya hemos instalado también los skins **Corelian** y **QuickSilver**. No he desglosado paso a paso el proceso por ser **idéntico** al primer ejemplo que hemos dado. Para añadir más skins -tantos como queráis-, únicamente debemos **repetir el proceso** con cualquier otro paquete.

Dado que gmplayer buscará su skin por defecto en el **directorio default**, vamos a realizar un **enlace simbólico** a uno cualquiera de los skins. Luego, podremos configurar nuestras preferencias como queramos y **serán recordadas para cada usuario** del sistema. Yo he elegido Neutron, por ser mi favorito.


```
menzoberranzan:/home/master# cd /usr/local/share/mplayer/Skin/
menzoberranzan:/usr/local/share/mplayer/Skin# ls
Corelian Neutron QuickSilver
menzoberranzan:/usr/local/share/mplayer/Skin# ln -s Neutron default
menzoberranzan:/usr/local/share/mplayer/Skin#
```

Ahora, únicamente debemos ejecutar el comando **gmplayer** para abrir la interfaz gráfica de MPlayer y poder empezar a usar este magnífico software.



Enhorabuena, ya has compilado MPlayer. El uso de este software es **muy sencillo** y creo que no es necesario dar detalles (además que el presente documento pretende ser un manual de **compilación**, no de uso :-P). Para conocer los **atajos de teclado** (que siempre vienen bien) recomiendo echar un vistazo a las **manpages de MPlayer**: <http://www.mplayerhq.hu/DOCS/man/es/mplayer.1.html>.

A disfrutar de tus vídeos.

Distribución de este documento

Este documento se distribuye en formato PDF realizado bajo OpenOffice.org 1.1.5.

Ficheros a distribuir:

Nombre: "MPlayer.pdf"

Descripción: Documento principal.

Nombre: "MPlayer.pdf.sig"

Descripción: Firma digital PGP del fichero "MPlayer.pdf" realizada por el autor.

Nombre: "hash.txt"

Descripción: Contiene las cadenas hash MD5 y SHA-1 de los ficheros "MPlayer.pdf" y "MPlayer.pdf.sig".

Datos adicionales:

Las cadenas de hash MD5 y SHA-1 pueden resultar útiles para comprobar la integridad del fichero descargado, pero no son garantía de la inalterabilidad del documento, pues éste puede haber sido alterado junto a las cadenas de hash.

Para comprobar la completa autenticidad e inalterabilidad del fichero, es necesario utilizar el sistema OpenPGP para validar el fichero .sig (MIME/PGP) de firma. Cualquier modificación no autorizada del documento hará que la firma del mismo no sea válida, y ésta es imposible de falsificar.

Death Master

Autenticación:

-----BEGIN PGP MESSAGE-----

```
qANQR1DBwUwDJoT5ygJgu7ABD/9ZQFPLdzoZ/Tp38lVrGpieCUnUtdxvDrzzNUTr
Gh7NCCfaxLYGfl6+faZCM8fHvKXqcyCVzk3WvK7l0k6CcSs0YxYgzqWQd4LNCyS
jpsXp3PFOPxIsZSDagUfgufY1Vfq3cG7dTDV8JZAs1pH5rSoOwqn2VR4bkyJiBxQ
aCq/jJ4zHycCxBT0SqX5SM9ZyMKOcf4iipvgP/4Iepnixlo1u3lc0i+8m5yEoC11
PRGJw/U19JN8SRyH20xACl7kS91xeKrJOB1soEsRUI9V56a9XERdj8Ch/wgdNk0J
l3PwmMBIVCAgAdbauJwzbZfpqGryOWdJOAQ7ylqJP9TQ1uKzDwU5vakFggQGiyFi
NfMpZuqb87mt9UOdRCwoiF18HuYrqNpfO+EAAlTofB1gdii3KXHPFIhzYw+RVJ21
WzRYzskGiHeiOxOSoz+Plxyy+Uqt7QQ8dhcHk5PxfKk65TZxOD/X7lchaJWBdlp5
B0vl0d/w0R8uUuS99Y++BU7j/DYvVqJACJ+1Pjl+6MB95Qzg/YBjePHLDVvQu1xK
OU32VEIPwvEb4pYt+1Vi512VO3PIA9IY1OifgGHL+uGgGJ13MQ/2WlGxJtYf71b
YCRmSAOXHlqkBWxk3iLDzpi5+QIN+6oyB5LZQq+dBqQMTxys1ZESTQ82JmTjfw1Q
jHAWBdLCdAHAdl+7k4mDrsUSB7vF534l0AkM8QJHL0BavSc905XrMG09q22aU4zT
R7KLoTQFhpUXhjG/Y1NrLi6u0YT48LD/nBZoBk+J6xTqoNp+hLz+wObyMSU4G/iY
OwJLCUI77cUlnB4P75UGq9GC7ewwHykxhVIRcncZfdPe9OBmQFzdXkSCzGlXq4mX
PmQTNmJqUbjqACESCu+9ioeaKfelmCgJGbu6SJKDfouRKIkxDkMOJSW81s6wlGgS
VzOGK2ejGZBa1AzAnvNjzQYiQW573WQUHwInOh0v/oCFJQkqOMt+Q9dJtaA2QYPb
OA2gPrZLhvmD4JO2ieX3OOQnP7P91Wb1ToQPfLQSR5xWc/T6WGLiGAA+YxO716u2
9hQOeNp6QNxQTYjmiZ5mncX1Nia6QryWUE9KsCXJ9zkaOR8AsHPV4iEbtHdph1fk
A4Z8u/O/hCytZVTfueXuTQaF4kL8F2e+d9r+njoykh+CbVqZtwjtE3vdfNVHwDC
g9xBffUgu9LSenq6Mn44dGwFJupFVZ+o+Wx9MiEA0LcAQWgD9mg+uZYITXxi20wB
5d3pBtO8M7fc0y/e7x9oDdpbi1Ldb/QoLQZHTF42lfGwSHESCriVudOm7IMWXdz
ym/F2puOdXkaYq1jcltA5YBc7.Jr4fTkcWMxWRyCl65lgS5buyqmy4XWcQb2divq1
piMcbUi4t2zcO3APLv6VtrMb6vwyG2bM7SA8/pr1xbX068g4EGZf29O4I39znUtp
nQB66QGJ31mhUa4qRgeXb/a8xrmg7JVNcoWzRhX3u4iZCIWfYRa4dl0l3LqUvDom
6A8kZsflh0xingwyE3dq0pdxvmdDrWr9RRF+TVAKI2rUZk0/mB4WFyWcKOMY0aA
mnd52bdNfEYmeWx+UyqK+Syd2FDjy9XRK+P6j/oCFVLOxpE4TMTNCl/AlfLfb24v
ZRUCHyBpMLx/S507/qUvF+USuDvPxbhUtlg3rhvWvPoO7dMHx+N5YK/9wbJWgdX+T
Aytlo4ZEvkfZaCq4/RUMrquLuS17InLingOG5WnEtPp219KzZolsFoLHoVnHS4Rq
hLLrkEU63dDFUXc=
=qx4M
```

-----END PGP MESSAGE-----

Licencia

Compilando MPlayer – <http://www.death-master.tk/> – Versión 1.0

Este documento ha sido liberado por su autor bajo la licencia GNU Free Documentation License (GFDL), y su utilización, copia o reproducción queda sujeta a los términos de la citada licencia, que puede ser consultada en el siguiente sitio web:

- **GNU Free Documentation License:** <http://www.gnu.org/copyleft/fdl.html>

GFDL Version 1.2, November 2002

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright (c) 2005 Death Master

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being "Distribución de este documento" and "Licencia", no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Cualquier copia, modificación, distribución o utilización en general de este documento debe respetar la autoría original del mismo, correspondiente a **Death Master**.

Compiling MPlayer – <http://www.death-master.tk/> – Version 1.0

This document has been freed by its author under the license GNU Free Documentation License (GFDL), and its use, copy or reproduction is subject to the terms of the mentioned license that can be consulted in the following website:

- **GNU Free Documentation License:** <http://www.gnu.org/copyleft/fdl.html>

GFDL Version 1.2, November 2002

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright (c) 2005 Death Master

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being "Distribución de este documento" and "Licencia", no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Any copy, modification, distribution or general purpose use of this document should respect the original responsibility of it, corresponding to **Death Master**.



* End Of File *