



## ÍNDICE

PAG 2	Introducción
PAG 3	Qué es y qué no es DOSBox
PAG 4	Instalando DOSBox
PAG 7	Nociones básicas de MS-DOS
PAG 14	Uso de DOSBox
PAG 22	Teclas básicas
PAG 23	Acelerando la emulación
PAG 26	El archivo <i>dosbox.conf</i>
PAG 30	Juegos en modo protegido; usando DOS32A.
PAG 32	Dudas frecuentes.
PAG 36	<i>Front-ends</i> : D-Fend.
PAG 42	Notas sobre este texto.

# INTRODUCCIÓN

Mucha gente me ha escrito, sumida en grados diversos de frustración, preguntando “¿Cómo coño se usa DOSBox?”. Y una cosa es cierta; para el usuario típico de Windows, que se ha iniciado en los ordenadores con versiones diversas de este sistema operativo -y no ha tocado otros-, el asunto puede parecer al principio muy complicado. Imaginémoslo...

Nuestro conejillo de Indias, que únicamente ha conocido Win98 o WinXP, acaba de instalar DOSBox. Lo arranca, y... ¡Horror! ¡Una consola de comandos! ¿Y los iconos? ¿Y los menús desplegados de la barra de arriba? ¿Y...? En resumen, el pobre hombre no entiende nada; cierra DOSBox espantado y nunca ya lo vuelve a tocar. Aunque al final, si se decide por hacer la pregunta que comentaba arriba, podemos decir que ha superado el primer paso.

El otro tipo de mensajes que suelo leer vendrían a ser del estilo de “¡DOSBox es una mierda!”, “¡DOSBox es muy lento!”, “¡Esto no tiene sonido!”... Estas quejas suelen venir la mayoría de las veces de usuarios novatos, que no le han dedicado al emulador las suficientes “horas de vuelo”. Así que, resumiendo, DOSBox no es ninguna porquería; lo que pasa es que hay que saber cómo usarlo.

Y por ello me he decidido a escribir este pequeño tutorial. De hecho, llevaba tiempo dándole vueltas a la idea... Ahora, mi intención no es hacer un compendio enciclopédico sobre todo lo referente a DOSBox -cosa que está fuera de mis posibilidades-, sino una guía “de uso”; algo que oriente a los que lo emplean por primera vez, para que puedan aprender las nociones básicas y sacarle un rendimiento razonable.

Por lo demás, mi idea es enseñar cómo usar el DOSBox tal cual, sin el empleo de *front-ends* como puedan ser Boxer, D-Fend... Sencillamente, aprendiendo a utilizar DOSBox podréis defenderos con cualquiera de ellos, o bien no usarlos para nada si no lo deseáis -ésta ha sido mi opción-. De todos modos, hacia el final del manual hablaré un poco por encima de D-Fend, que es el que mejor conozco.

Espero que el manual os sirva de ayuda, y podáis ejecutar tranquilamente juegos como *Titus the Fox* en vuestros Pentium 4... Aunque con un poco de experiencia, se puede conseguir mucho más.

# QUÉ ES Y QUÉ NO ES DOSBOX

Antes que nada, no estaría mal hacerse una idea de lo que es DOSBox, lo que puede y no puede hacer... Aunque me imagino que si estás leyendo este tutorial ya sabrás más o menos de qué va el asunto.

DOSBox es un emulador de MS-DOS; esto es, un programa que permite ejecutar aplicaciones de DOS en otros sistemas operativos. Por ejemplo, podríamos trabajar con viejas versiones de WordPerfect en Windows XP, o jugar a Wolfenstein 3D en nuestro Linux... Así que por si acaso no tiréis a la basura todos esos polvorientos disquetes que guardáis en el armario 😊.

DOSBox funciona en gran variedad de sistemas operativos: Windows XP, Linux, Mac OS X, BeOS, OS/2... Es capaz de emular el modo protegido -introducido en los procesadores 286-; diferentes modos gráficos -Tandy, Hercules, CGA, EGA, VGA, VESA-; memoria extendida (XMS) o expandida (EMS); y las tarjetas de sonido más utilizadas en juegos y programas antiguos -SoundBlaster, Adlib, Gravis Ultrasound...-.

Bueno, si no sabes qué es todo esto, no importa demasiado; son detalles prescindibles para el usuario típico de DOSBox. Aunque para quién haya usado ordenadores con MS-DOS -supongo que una gran parte de la gente que ahora emplea DOSBox-, debería resultarle familiar.

¿Y qué NO es DOSBox?

Dicho rápidamente: una solución milagrosa.

Puesto que DOSBox se trata de un emulador, lo que ejecutemos con él no tendrá -ni de lejos- el mismo rendimiento que si pudiésemos correrlo de forma nativa en nuestro sistema operativo. Por ejemplo, si alguien pretende jugar a Blood con DOSBox en un Pentium II a 350 Mhz, que se lo vaya quitando de la cabeza; quizás la emulación funcionase, pero a un ritmo tan lento que la haría inútil para cualquier cosa.

Por lo que yo mismo he ido viendo, en una máquina de 1 – 1,5 Ghz DOSBox equivaldría a un 486 del montón (25-33 Mhz). Ahora, con ordenadores más modernos, con procesadores más potentes y más memoria RAM, supongo que se podrán alcanzar mejores resultados. También es de imaginar que en futuras versiones -en el momento de escribir esto, DOSBox va por la 0.63- se mejorará la velocidad de la emulación.

Digo todo esto para que cada cuál se haga unas expectativas realistas de lo que puede ofrecer el emulador, y nadie se rasgue las vestiduras si DOSBox ejecuta el Doom en su viejo armatoste a la velocidad de una babosa con reuma.

# INSTALANDO DOSBOX

Bueno, normalmente esto no es difícil, pero como siempre hay quién se pierde, lo comentaré aunque sea un poco por encima...

Evidentemente, lo primero sería visitar la página web de DOSBox (<http://dosbox.sourceforge.net>), y descargarnos el emulador -en su versión más actual posible, a no ser que tengamos algo especial en mente-. En la sección "*downloads*" lo encontraremos disponible para diversos sistemas operativos; evidentemente, escogeremos el que más nos interese.

## **Windows:**

En el caso de Windows, instalar DOSBox resulta bastante fácil, puesto que esta versión viene con un instalador propio. Simplemente especificamos la carpeta dónde queramos colocarlo -por ejemplo, *C:\Dosbox-*, y a funcionar... En un suspiro tenemos el programa descomprimido y la entrada correspondiente en el menú de inicio.

## **Linux:**

Para Linux hay algunos instaladores disponibles para diferentes distribuciones. Aunque si lo deseamos, podemos bajarnos el código fuente y compilarlo nosotros mismos. Bueno, ésta ha sido mi opción; teóricamente se consigue así un mejor rendimiento en nuestra máquina.

Aquí doy unas indicaciones para quién tenga ganas de hacerlo:

El primer paso es evidente: descargarnos el código fuente de la página oficial del DOSBox (ver arriba), y guardarlo dónde deseemos. Luego lo descomprimimos, cosa que podemos hacer desde una consola de comandos; accedemos a la carpeta en la que hayamos almacenado el archivo *.tar.gz*, y tecleamos:

```
$ tar xzf dosbox-0.63.tar.gz
```

En este ejemplo, el archivo con el código fuente es *dosbox-0.63.tar.gz*, aunque obviamente podríamos estar trabajando con otras versiones...

Es posible que nuestra versión del *tar* no soporte la opción *-z*; en tal caso -raro, supongo-, deberíamos usar primero *gunzip*, y luego descomprimir con *tar* el archivo *.tar* resultante. De todas formas, cualquier distribución de Linux cuenta con utilidades que permiten hacer esto cómodamente desde el entorno gráfico, como *Ark*, etc.

El caso es que, lo hagamos de la manera que lo hagamos, nos encontraremos una nueva carpeta (en este ejemplo llamada *dosbox-0.63*) que contendrá el código fuente listo para compilar.

Ahora sólo tendríamos que acceder a esta carpeta recién creada, y proceder a la compilación del modo acostumbrado:

```
$ ./configure  
$ make
```

En principio no es necesario nada más. Quién quiera echar un vistazo a las diversas opciones de compilación, debería teclear antes que nada el comando `./configure --help`; pero ya digo que normalmente no hace falta preocuparse de esto.

Después de ejecutar el `make` ya tendremos listos los binarios (ejecutables) del DOSBox. Podemos dejarlos en esa carpeta, moverlos a cualquier otra perteneciente a nuestro usuario, o instalarlos en el sistema. En este último caso, accederíamos a la cuenta del administrador, y teclearíamos:

```
# make install
```

Con ello instalaríamos DOSBox en el directorio `/usr/local/bin/dosbox`, con lo que estaría disponible para todos los usuarios de la máquina.

Por último, esto de las compilaciones se sabe cómo empieza pero no cómo acaba..., por el tema de las librerías requeridas en cada caso. Para compilar DOSBox, en principio necesitaríamos las librerías SDL (`SDL`, `SDL_mixer`, `SDL_net`, y probablemente `SDL_dev`, `SDL_mixer_dev` y `SDL_net_dev`), disponibles en <http://www.libsdl.org>, y tal vez alguna más... Bueno, en el archivo `INSTALL`, que viene con el código fuente, suele haber información detallada sobre esto; os recomiendo que lo leáis antes de poneros manos a la obra.

## **Mac OS X:**

Una lástima que aquí no pueda hablar de primera mano -el único Mac OS que conozco, así por encima, es el 8.1-... Pero bueno, el proceso parece bastante sencillo.

La versión de DOSBox para Mac OS X la podemos encontrar en <http://www.holwegner.com/software/#DOSBox>, en un archivo comprimido que contiene los binarios correspondientes (o sea, el programa listo para usar). Por cierto, sólo funcionará en sistemas Mac OS X 10.3 o superiores; no es compatible con versiones más antiguas.

Bueno, descargamos ese archivo (en el momento de escribir esto, `DOSBox-0.63-1.sit`), y lo descomprimos. Ésto último puede haberse hecho automáticamente tras la descarga; si no es el caso, bastará un doble clic sobre el archivo ya descargado en el escritorio.

Y con esto tendríamos una carpeta en nuestro escritorio con el DOSBox descomprimido, que podríamos ubicar dónde mejor nos pareciese. Por

ejemplo, en la carpeta de aplicaciones, accesible desde el icono del disco duro...

Y ya lo tenemos. Para utilizar el emulador, sólo tenemos que entrar en la carpeta del DOSBox y arrancar el ejecutable.



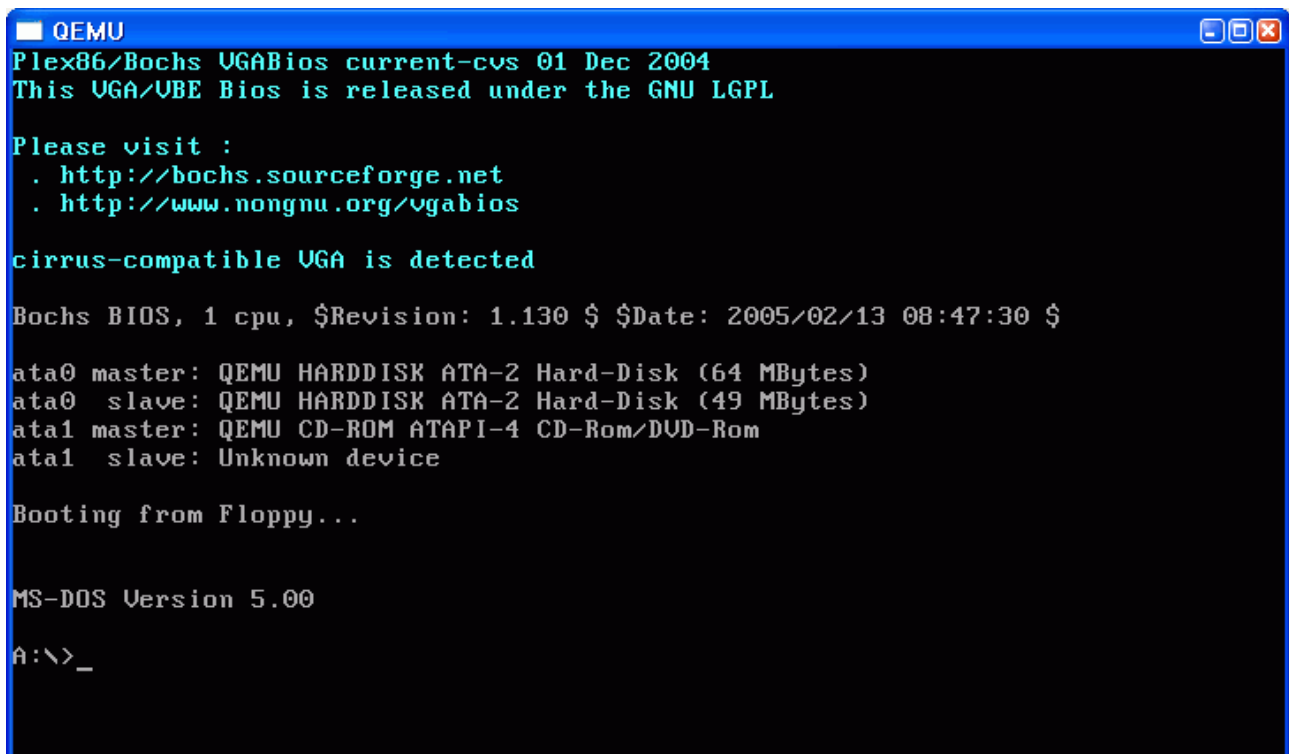
Aquí lo vemos en una bonita captura de pantalla -sacada de un hilo del foro de DOSBox, si no recuerdo mal-. El archivo comprimido que nos hemos descargado antes (*DOSBox-0.63-1.sit*) ya no nos sirve de nada y podemos eliminarlo tranquilamente.

## NOCIONES BÁSICAS DE MS-DOS

DOSBox es un emulador de MS-DOS, y para poder utilizarlo con un poco de soltura necesitaremos conocer al menos el funcionamiento básico de este sistema operativo. Evidentemente quiénes trabajaron en su día con MS-DOS ya tendrán esto bien sabido, y dudo que esta sección les sirva de mucho; pero para muchos usuarios actuales de Windows sí puede resultar interesante echarle una ojeada.

MS-DOS -y por lo tanto, DOSBox- es un sistema operativo que no utiliza entorno gráfico; para entendernos, no hay escritorio, ni iconos de colores, ni menús, ni nada similar. De este modo, para abrir carpetas, ejecutar programas y demás -o sea, para usar el ordenador- no emplearemos el ratón ni haremos doble clic en ningún sitio; todo lo que queramos que haga nuestra máquina lo iremos especificando mediante comandos escritos con el teclado.

Así, si iniciásemos un sistema MS-DOS nos encontraríamos con una simple pantalla de texto. Algo como esto (en este caso se trata de un PC emulado por QEMU, ejecutando la versión 5.00 de MS-DOS):



```
QEMU
Plex86/Bochs UGABios current-cvs 01 Dec 2004
This UGA/VBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

cirrus-compatible VGA is detected

Bochs BIOS, 1 cpu, $Revision: 1.130 $ $Date: 2005/02/13 08:47:30 $

ata0 master: QEMU HARDDISK ATA-2 Hard-Disk (64 MBytes)
ata0 slave: QEMU HARDDISK ATA-2 Hard-Disk (49 MBytes)
ata1 master: QEMU CD-ROM ATAPI-4 CD-Rom/DVD-Rom
ata1 slave: Unknown device

Booting from Floppy...

MS-DOS Version 5.00

A:\>_
```

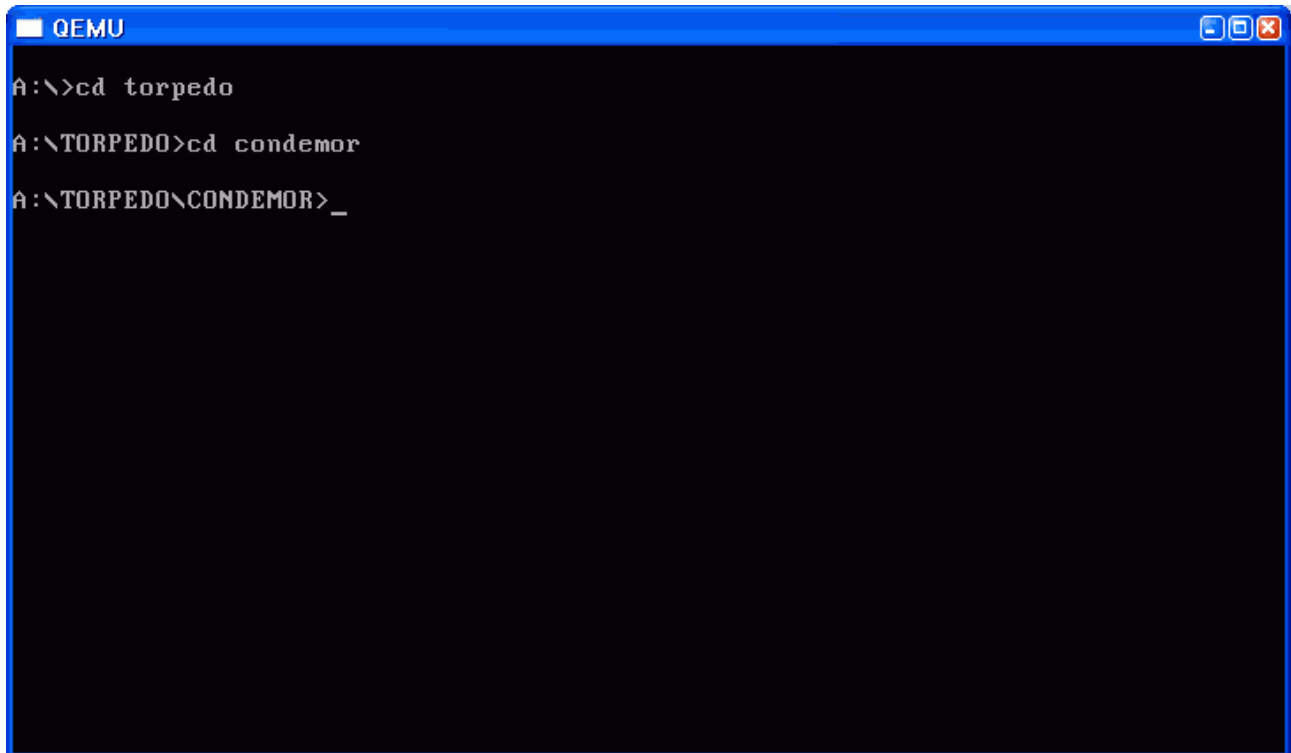
De aquí, lo que más nos interesa ahora es la última línea que aparece en la ventana (A:\>). Éste es el llamado *prompt*, que nos indica la unidad y la carpeta en la que estamos trabajando actualmente. Así, en la imagen de arriba acabamos de iniciar el sistema desde un disquette de arranque, por lo que la unidad en que nos encontramos será la A:.

Si ahora accediésemos a una carpeta determinada (una llamada



"torpedo" existente en nuestro disquette de arranque, por ejemplo), el *prompt* cambiaría para reflejar nuestra nueva posición; en lugar de "A:\>", tendríamos "A:\TORPEDO>". Resumiendo: la primera letra seguida de los dos puntos, indica la unidad en que nos encontramos (A:, C:, D:, etc); y lo que sigue a partir de la barra inclinada, la carpeta.

A continuación del *prompt* aparece el cursor con el que iremos escribiéndole al ordenador las órdenes que deseemos, una por una. Bueno, nada como verlo en una imagen:



```
QEMU
A:\>cd torpedo
A:\TORPEDO>cd condemor
A:\TORPEDO\CONDEMOR>_
```

Y esto nos lleva al primer comando de MS-DOS que deberíamos aprender: **cd** (que viene de *change directory*). Nos servirá para movernos entre las carpetas de una misma unidad. El uso viene a ser éste:

**CD [nombre carpeta]**

Resulta muy sencillo, como puede verse en la captura de pantalla. Como en casi todos los comandos de MS-DOS, podemos usar direcciones relativas o absolutas:

```
A:\torpedo> cd condemor
A:\torpedo> cd \condemor
```

En el primer ejemplo, entraríamos en la carpeta "condemor" existente dentro del presente directorio -en este caso sería A:\TORPEDO\CONDEMOR-;



en el segundo, iríamos a parar a `A:\CONDEMOR`, con independencia de dónde estuviésemos situados en el momento de escribir la orden.

Claro que también podemos ascender por el árbol de directorios hacia la carpeta raíz. Resulta tan sencillo como esto:

```
A:\TORPEDO\CONDEMOR> cd ..  
A:\TORPEDO> cd ..  
A:\>
```

Los dos puntos seguidos en la orden **cd** indican que deseamos subir un nivel en el árbol de carpetas. Por supuesto, también podemos sustituirlos por una dirección absoluta:

```
A:\TORPEDO\CONDEMOR> cd \  
A:\>
```

Bien, ahora ya sabemos como movernos entre las carpetas... Aunque no sólo de la unidad A: vive el hombre. Lo normal es que tengamos que trabajar con varias unidades... Pero tranquilos, ir de una a otra es tan fácil como teclear la letra de la unidad correspondiente:

```
A:\> c:  
C:\>
```

Muy bien, pero, ¿y los archivos? ¿Cómo vemos los archivos presentes en cada carpeta, para saber cuál ejecutar?. Bueno, pues para esto tenemos el comando **dir**, cuyo uso tampoco entraña mayores complicaciones:

**dir [archivo o carpeta] [opciones]**

**Dir** nos muestra los archivos contenidos en la carpeta que especifiquemos -o bien los del directorio actual, sino especificamos ninguna-. Es importante señalar que en MS-DOS los nombres de fichero sólo pueden tener como máximo ocho letras (los nombres largos típicos de Windows aparecerán truncados como `xxxxxx~1.xxx`).

Así, por ejemplo:

```
C:\> dir  
...
```

Nos mostraría una lista de todos los archivos -salvo los ocultos- presentes en el directorio en el que nos encontramos, en este caso `C:\`. También podemos ver el contenido de una carpeta concreta, especificando su

nombre o su dirección absoluta:

Por ejemplo:

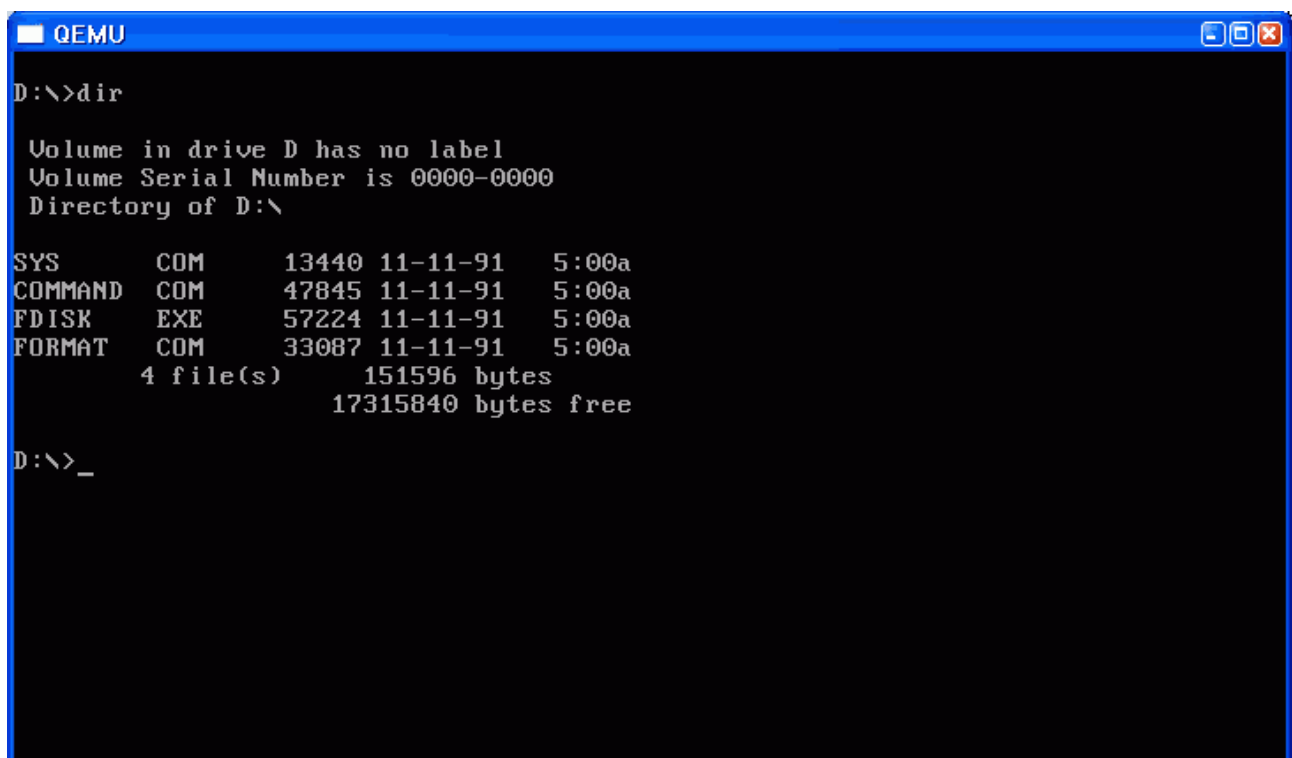
**C:\> dir c:\fistro**

... <-- Muestra el contenido de C:\FISTRO

**C:\> dir fistro**

... <-- Muestra el directorio C:\FISTRO, o el archivo FISTRO., si existe.

Aquí podemos ver este comando en acción:



```
QEMU
D:\>dir

Volume in drive D has no label
Volume Serial Number is 0000-0000
Directory of D:\

SYS      COM      13440 11-11-91   5:00a
COMMAND  COM      47845 11-11-91   5:00a
FDISK    EXE      57224 11-11-91   5:00a
FORMAT   COM      33087 11-11-91   5:00a
         4 file(s)      151596 bytes
                        17315840 bytes free

D:\>_
```

**Dir** nos permite usar caracteres especiales para especificar qué archivos queremos ver listados. Éstos son el asterisco (\*) y el interrogante (?), y cada uno tiene un modo de actuar. Veámoslo:

**C:\> dir ?aa.exe** <-- Muestra todos los archivos con extensión EXE; sus nombres tendrán tres letras; la segunda y la tercera serán "a", y la primera un carácter cualquiera. Así, en la lista podríamos ver archivos como *aaa.exe*, *faa.exe*, *3aa.exe*, *\_aa.exe*, etc.

**C:\> dir ?????.bat** <-- Muestra todos los archivos con extensión BAT cuyos nombres tengan cuatro letras: *jarl.bat*, *1234.bat*, *gh9\_.bat*, etc.

Así, vemos que el interrogante es un comodín que sustituye a cualquier

carácter concreto -el primero del nombre del archivo, el último de su extensión, etc-. El asterisco, en cambio, reemplaza a cualquier número de ellos. Así, tendremos:

**C:\> dir \*.bat** <-- Muestra todos los archivos que tengan extensión BAT, sea cual sea su nombre.

**C:\> dir chiquito.\*** <-- Muestra todos los archivos llamados "CHQUITO", tengan la extensión que tengan.

**C:\> dir ab\*.exe** <-- Muestra todos los archivos cuyos nombres comiencen por "ab", y tengan extensión EXE. Por ejemplo, *abajo.exe*, *abr.exe*, *abfistro.exe*, etc.

Ambos comodines pueden combinarse en un mismo comando; de ese modo, podríamos hacer búsquedas como *dir ?o\*.com*, *dir \*.b??*, y similares. Pero muy raro será que tengamos que hilar tan fino.

Por lo demás, me quedaría decir que el comando **dir** admite varias opciones, que permiten modificar su funcionamiento. Las principales serían estas:

**C:\> dir \*.exe /p** <-- En muchas ocasiones podemos encontrarnos con listas que no caben en la pantalla, y fluyen rápidamente sin permitirnos ver nada. Pero si utilizamos la opción */p*, el ordenador hará una pausa cada vez que la lista de archivos llene la pantalla, permitiéndonos así examinarlos con tranquilidad.

**C:\> dir \*.exe /w** <-- La opción */w* muestra una lista resumida de los archivos de la carpeta, aprovechando al máximo el espacio de la pantalla. En vez de colocar cada archivo en una línea y listar sus detalles (fecha y hora de creación, etc), los sitúa distribuidos en columnas, y mostrando solamente sus nombres y extensiones.

**C:\> dir \*.exe /s** <-- Por último, especificando la opción */s* le indicamos al ordenador que muestre también el contenido de todos los subdirectorios que se incluyan en la lista. Por ejemplo, con *dir a:\condemor /s* veríamos todos los archivos guardados dentro del directorio A:\CONDEMOR, así como los de los subdirectorios que estuviesen dentro de él.

Estas opciones también pueden combinarse entre sí, para hacer búsquedas más flexibles. Así, podríamos escribir comandos del estilo de *dir \*.bat /s /p*, y otros similares.

En la imagen de la siguiente página vemos el efecto de la opción */w*:

```
QEMU
C:\>dir *.a?? /w

Volume in drive C has no label
Volume Serial Number is 0000-0000
Directory of C:\

TILES000.ART    TILES001.ART    TILES002.ART    TILES003.ART    TILES004.ART
TILES005.ART    TILES006.ART    TILES007.ART    TILES008.ART    TILES009.ART
TILES010.ART
               11 file(s)    31985048 bytes
                   26597376 bytes free

C:\>_
```

Bien, y ahora, ¿cómo ejecutar archivos? Nada más sencillo.

En MS-DOS, en principio, los archivos ejecutables tendrán la extensión COM o EXE (binarios), o bien BAT (que son simplemente *scripts* compuestos por órdenes de MS-DOS, utilizados para arrancar programas con opciones personalizadas, abreviar secuencias de comandos, etc).

Para ejecutar un archivo, simplemente tenemos que escribir su nombre. La extensión no es imprescindible, a no ser que existan varios ficheros con el mismo nombre y distintas extensiones:

**C:\> lucas.exe** <-- Ejecutaríamos el archivo "lucas.exe".

**C:\> lucas** <-- Ejecutaríamos "lucas.exe", "lucas.com" o "lucas.bat"; cualquiera de los tres que se halle en la presente carpeta.

Con estos conocimientos tan básicos ya podríamos desenvolvemos, mejor o peor, en DOSBox. No obstante, comentaré otros comandos que también pueden resultarnos útiles:

**md [nombre carpeta]**

Este comando nos servirá para crear carpetas. Su uso viene a ser similar al del ya visto **cd**. Y como éste, admite direcciones relativas o absolutas. Esto es fácil de ver con un par de ejemplos:

**A:\CONDEMOR> md grijando**

Con este comando crearemos la carpeta A:\CONDEMOR\GRIJANDO.

**A:\CONDEMOR> md \grijando**

Y así crearíamos la carpeta A:\GRIJANDO.

Muy similar a **md** es la orden **rd**. Sólo que ésta tiene la función contraria. No crea carpetas, sino que las borra... Ahora, **rd** sólo eliminará las carpetas que estén vacías; si el directorio que deseamos borrar contiene algún archivo, simplemente nos aparecerá un error, y el comando no hará nada.

Para borrar archivos tenemos la instrucción **del**. Su uso no es muy diferente al de **dir**, visto más arriba. Y al igual que ésta, también admite el uso de comodines como el interrogante y el asterisco:

**C:\> del a\*.\*** <-- Borra todos los archivos que comiencen por "a".

**C:\> del \jarl\\*.com** <-- Borra todos los archivos con extensión COM presentes en el directorio C:\JARL.

Por lo demás, no uséis **del** en DOSBox o en el intérprete de comandos de Windows confiando en que los ficheros borrados irán a la papelera de reciclaje. Al contrario, lo que eliminéis con **del**, eliminado está...

También podemos copiar archivos de una localización a otra; para ello existe el comando **copy**, que viene a funcionar del siguiente modo:

**copy (origen) (destino)**

Así, tendríamos:

**A:\> copy \*.\* c:** <-- Copia todos los archivos de A:\ a C:\.

**A:\> copy trueno.txt c:\nopuido** <-- Copia el archivo "trueno.txt" a la carpeta C:\NOPUIDO.

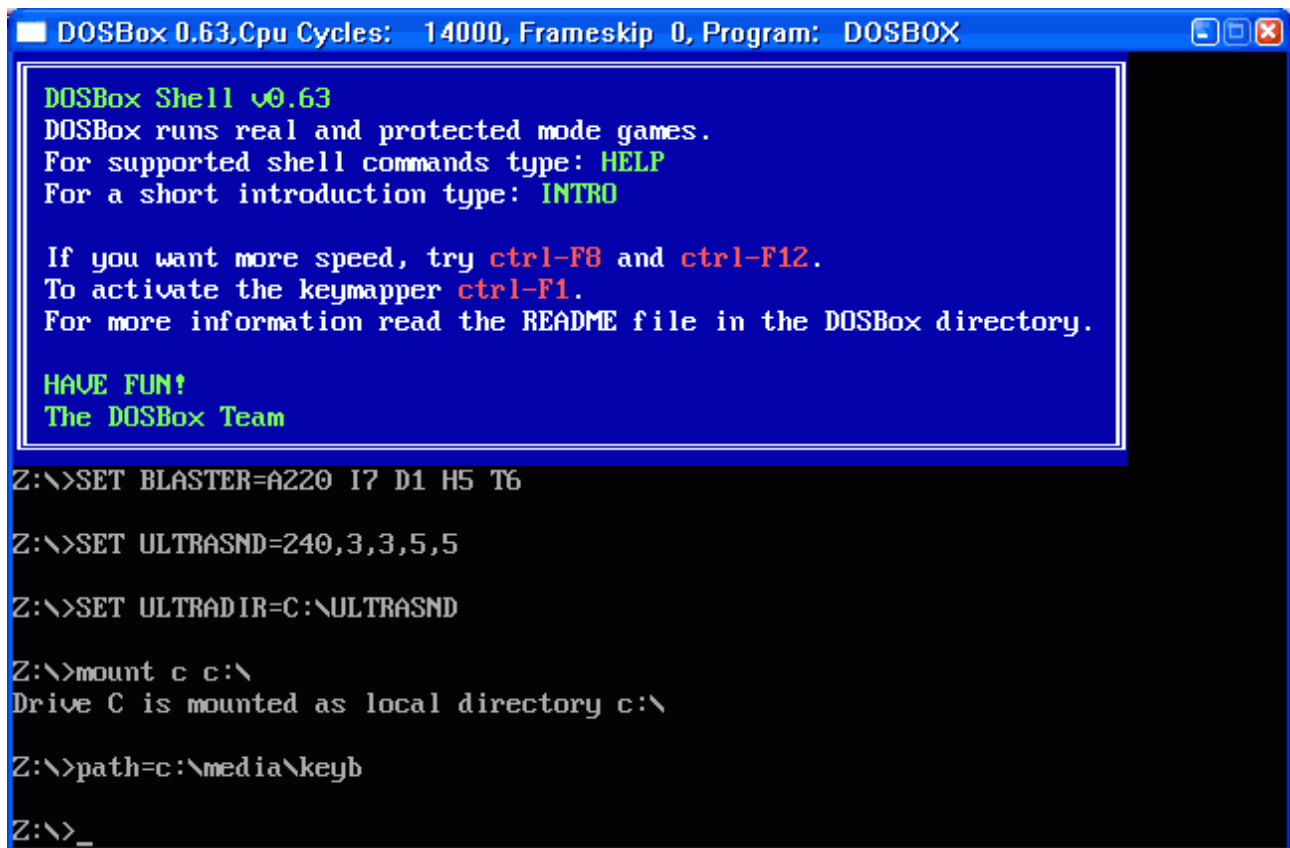
Y por último, quizás en algún momento necesitemos usar **ren**, el comando utilizado para renombrar archivos. La sintaxis es la misma que la empleada en **copy**:

**C:\> ren dortor.txt grijando.txt** <-- Cambia el nombre del archivo "dortor.txt" a "grijando.txt".

**C:\> ren a\*.\* b\*.\*** <-- Cambia el nombre de todos los ficheros que comiencen por "a", sustituyéndola por una "b". Por ejemplo, "argh.com" pasaría a ser "brgh.com", etc.

## USO DE DOSBOX

Bueno, después de nuestra pequeña lección de MS-DOS, vamos con el emulador sin más dilación... Así que accedemos a la carpeta dónde lo tenemos instalado y lo iniciamos, o bien lo arrancamos con el correspondiente acceso directo. Veremos que aparecerá rápidamente una ventana similar a ésta:



```
DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX

DOSBox Shell v0.63
DOSBox runs real and protected mode games.
For supported shell commands type: HELP
For a short introduction type: INTRO

If you want more speed, try ctrl-F8 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>SET ULTRASND=240,3,3,5,5

Z:\>SET ULTRADIR=C:\ULTRASND

Z:\>mount c c:\
Drive C is mounted as local directory c:\

Z:\>path=c:\media\keyb

Z:\>_
```

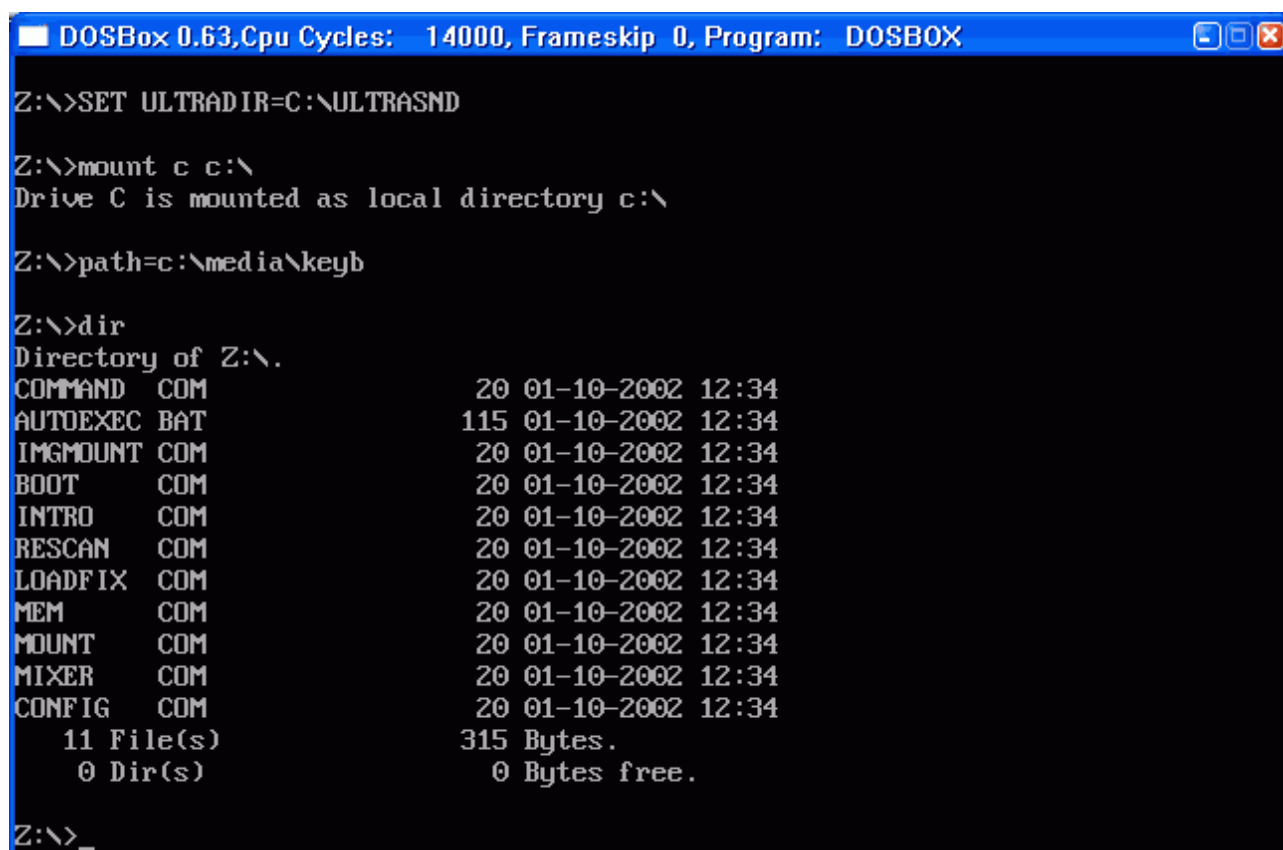
Bien, en el borde de arriba de la ventana tenemos algunos datos de interés. Los "CPU cycles", por ejemplo, nos indican la cantidad de instrucciones que estamos emulando; algo así como la velocidad a la que está funcionando DOSBox. "Frameskip" indica si hay algún salto de *frames* activado; lo normal es que tenga un valor de cero (sin salto de *frames*, el emulador muestra todas las imágenes que produce, sin omitir ninguna), aunque en algunas ocasiones querremos cambiarlo. Por último, "Program" indica simplemente el programa que estamos ejecutando en DOSBox.

Y ya en el interior de la ventana vemos la acostumbrada línea de comandos de MS-DOS, que a estas alturas no debería sorprender a nadie (de lo contrario, echadle una ojeada al capítulo anterior). El recuadro azul con letras blancas que vemos en la mitad superior de la ventana simplemente muestra unas pequeñas ayudas, gentileza de los desarrolladores de DOSBox.

Lo primero que nos llamará la atención será el hecho de comenzar a trabajar en una unidad llamada Z:. En realidad, ésta es una unidad ficticia que

crea el emulador al iniciarse. En ella se encuentran las utilidades básicas que incluye DOSBox por defecto, y resultarán imprescindibles para el manejo del emulador -bueno, realmente no todas, pero sí algunas-.

Con el comando **dir** podemos verlas todas:



The screenshot shows a DOSBox window titled "DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX". The command prompt shows the following sequence of commands and output:

```
Z:\>SET ULTRADIR=C:\ULTRASND
Z:\>mount c c:\
Drive C is mounted as local directory c:\
Z:\>path=c:\media\keyb
Z:\>dir
Directory of Z:\.
COMMAND  COM                20 01-10-2002 12:34
AUTOEXEC BAT             115 01-10-2002 12:34
IMGDMUNT COM                20 01-10-2002 12:34
BOOT      COM                20 01-10-2002 12:34
INTRO     COM                20 01-10-2002 12:34
RESCAN    COM                20 01-10-2002 12:34
LOADFIX   COM                20 01-10-2002 12:34
MEM        COM                20 01-10-2002 12:34
DMUNT      COM                20 01-10-2002 12:34
MIXER      COM                20 01-10-2002 12:34
CONFIG     COM                20 01-10-2002 12:34
    11 File(s)                315 Bytes.
    0 Dir(s)                   0 Bytes free.
Z:\>_
```

Bueno, COMMAND.COM y AUTOEXEC.BAT no nos interesan; forman parte del inicio de DOSBox -y de cualquier sistema operativo MS-DOS-, y no tendremos que preocuparnos de estos archivos para nada.

Ahora vamos a suponer que deseamos jugar una partida a Wolfenstein 3D, un videojuego que tenemos guardado en la carpeta `C:\JUEGOS\WOLF3D` de nuestro disco duro. Sin embargo, en DOSBox nos encontramos en la unidad `Z:`, y si intentamos pasar a `C:`, nos aparece una especie de error. ¿Qué pasa aquí?

Muy sencillo; DOSBox no utiliza directamente las unidades físicas de nuestro sistema operativo. Así que tenemos que "montarlas" antes de hacer nada -un concepto muy familiar para cualquier usuario de Linux-. Para eso tenemos la primera utilidad básica de DOSBox que veremos: **mount**.

**mount "letra de la unidad emulada" "directorio o unidad fisica" [-t tipo] [-aspi] [-ioct] [-used numero] [-size tamaño] [-label etiqueta] [-freesize tamaño\_en\_Mb]**



Bueno, ante todo no os asustéis 😊. Parece complicado, pero no lo es. **Mount** permite de hecho muchas opciones, pero en circunstancias normales apenas usaremos dos o tres. Las únicas que tendremos que especificar obligatoriamente son la "letra de la unidad emulada" y el "directorio o unidad física" que queremos que lea DOSBox.

Lo mejor, verlo en ejemplos:

```
Z:\> mount c c:\
```

Creamos en el emulador una unidad C:, cuyo contenido será nuestro disco duro C: y todos sus directorios.

```
Z:\> mount c c:\juegos
```

Básicamente lo mismo que en el ejemplo anterior... Creamos una unidad C:, en la que encontraremos el contenido de la carpeta C:\JUEGOS de nuestro disco duro...

Así, para jugar la partida a Wolfenstein 3D que hemos mencionado arriba, necesitaríamos escribir lo siguiente:

```
Z:\> mount c c:\juegos
```

```
Drive C is mounted as local directory C:\JUEGOS\
```

(Ahora ya tenemos una unidad C: en DOSBox).

```
Z:\> c: <-- Accedemos a la unidad C: recién creada.
```

```
C:\> cd wolf3d <-- Entramos a la carpeta del Wolfenstein 3D.
```

```
C:\WOLF3D> wolf3d.exe <-- Y lo ejecutamos.
```

Por supuesto, **mount** también trabaja con otros sistemas de archivos, aparte del clásico de Windows. Por ejemplo, en un sistema Linux:

```
Z:\> mount d /home/macario/juegos
```

(Creamos en DOSBox la unidad D:, cuyo contenido será el del directorio /home/macario/juegos de nuestro ordenador, y todos sus subdirectorios).

Con **mount** podemos también montar unidades de CD-ROM, o de disquette. Para ello tenemos la opción **-t**, que indica el tipo de unidad que queremos utilizar:

```
Z:\> mount d e:\ -t cdrom
```

(Montamos nuestra unidad de CD-ROM E: como D: en DOSBox).

```
Z:\> mount a a:\ -t floppy
```

(Creamos en DOSBox una unidad de disquette A:, que contendrá los archivos de la unidad A: de nuestro ordenador).

Las otras opciones de **mount** muy rara vez las utilizaremos, pero las comentaré aunque sea someramente:

#### **-size tamaño**

Especifica el tamaño de la unidad que queremos montar. No es una opción fácil, porque deberemos introducirlo no en *megabytes*, sino como los parámetros de disco que vemos en cualquier BIOS: cabezales, sectores y cilindros.

```
Z:\> mount c d:\ -size 4025,127,16513,1700
```

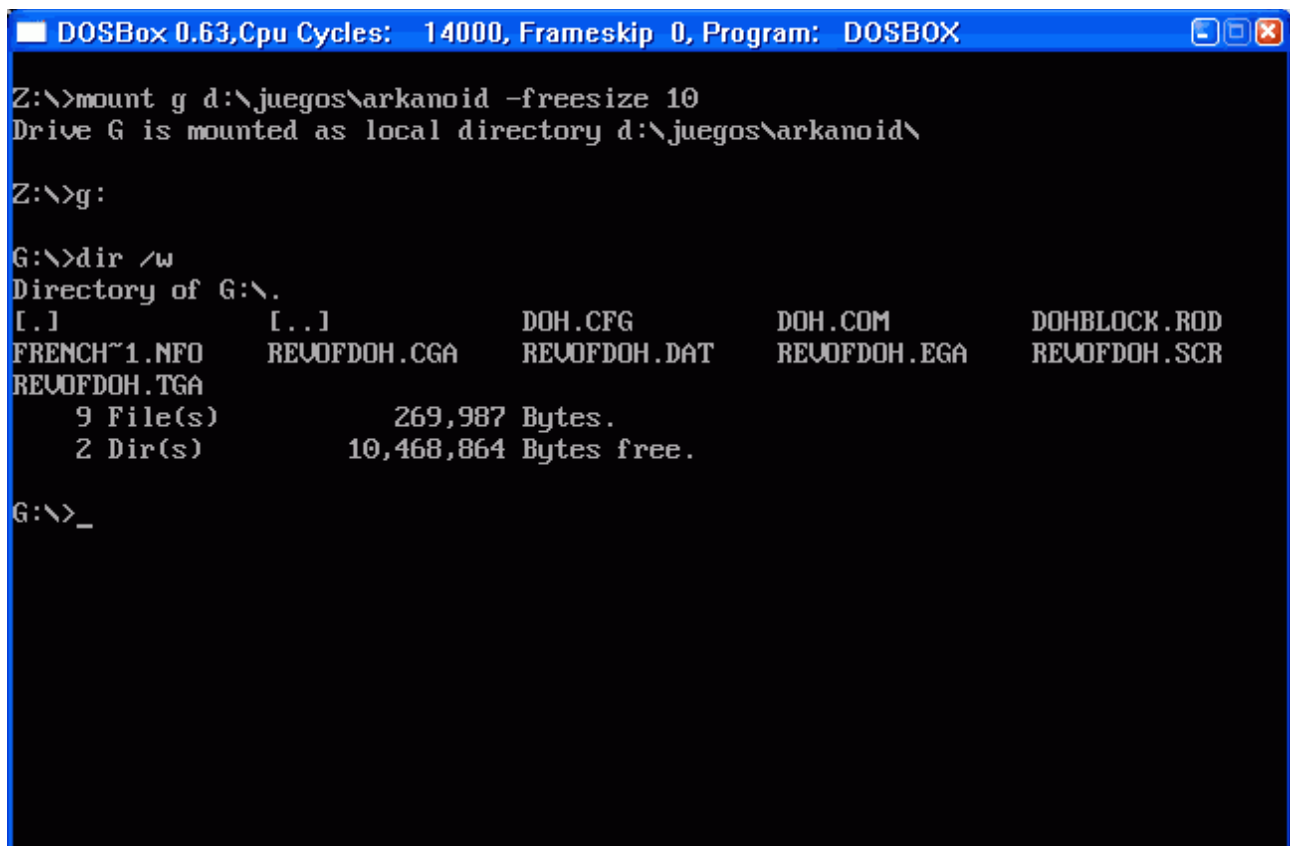
(Montamos una unidad C: de 870 Mb).

#### **-freesize tamaño\_en\_mb**

Tiene la misma función que **-size**, pero un uso más sencillo, puesto que ahora podemos introducir el tamaño de la unidad emulada en *megabytes*:

```
Z:\> mount c d:\ -freesize 870
```

(Igual que arriba; montamos una unidad C: de 870 Mb).

A screenshot of a DOSBox 0.63 window. The title bar reads "DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX". The command prompt shows the command "Z:\>mount g d:\juegos\arkanoid -freesize 10" and the response "Drive G is mounted as local directory d:\juegos\arkanoid\". Below this, the prompt "Z:\>g:" is shown. Then, the command "G:\>dir /w" is entered, resulting in a directory listing for "G:\\". The listing shows files like "FRENCH~1.NFO", "REVOFDOH.TGA", "DOH.CFG", "REVOFDOH.DAT", "DOH.COM", "REVOFDOH.EGA", "DOHBLOCK.ROD", and "REVOFDOH.SCR". It also shows statistics: "9 File(s) 269,987 Bytes." and "2 Dir(s) 10,468,864 Bytes free." The prompt "G:\>\_" is at the bottom.

```
DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX
Z:\>mount g d:\juegos\arkanoid -freesize 10
Drive G is mounted as local directory d:\juegos\arkanoid\
Z:\>g:
G:\>dir /w
Directory of G:\.
[.]          [...]      DOH.CFG          DOH.COM          DOHBLOCK.ROD
FRENCH~1.NFO REVOFDOH.CGA  REVOFDOH.DAT    REVOFDOH.EGA    REVOFDOH.SCR
REVOFDOH.TGA
          9 File(s)          269,987 Bytes.
          2 Dir(s)          10,468,864 Bytes free.
G:\>_
```

#### **-label etiqueta**

Especifica una etiqueta para la unidad que vamos a montar; vamos, como asignarle un nombre. Esto puede ser útil para arrancar ciertas utilidades que necesitan una unidad de CD con un nombre determinado.

```
Z:\> mount d f:\ -t cdrom -label tekwar
```

### **-aspi**

Utiliza los controladores ASPI. Sólo nos resultará útil al montar una unidad de CD-ROM en ciertos sistemas Windows.

```
Z:\> mount e e:\ -t cdrom -aspi
```

### **-ioctl**

Utiliza los comandos IOCTL para montar una unidad de CD-ROM. Válido solamente en sistemas Windows que lo soporten (Windows 2000/NT/XP).

```
Z:\> mount e e:\ -t cdrom -ioctl
```

### **-usecd numero**

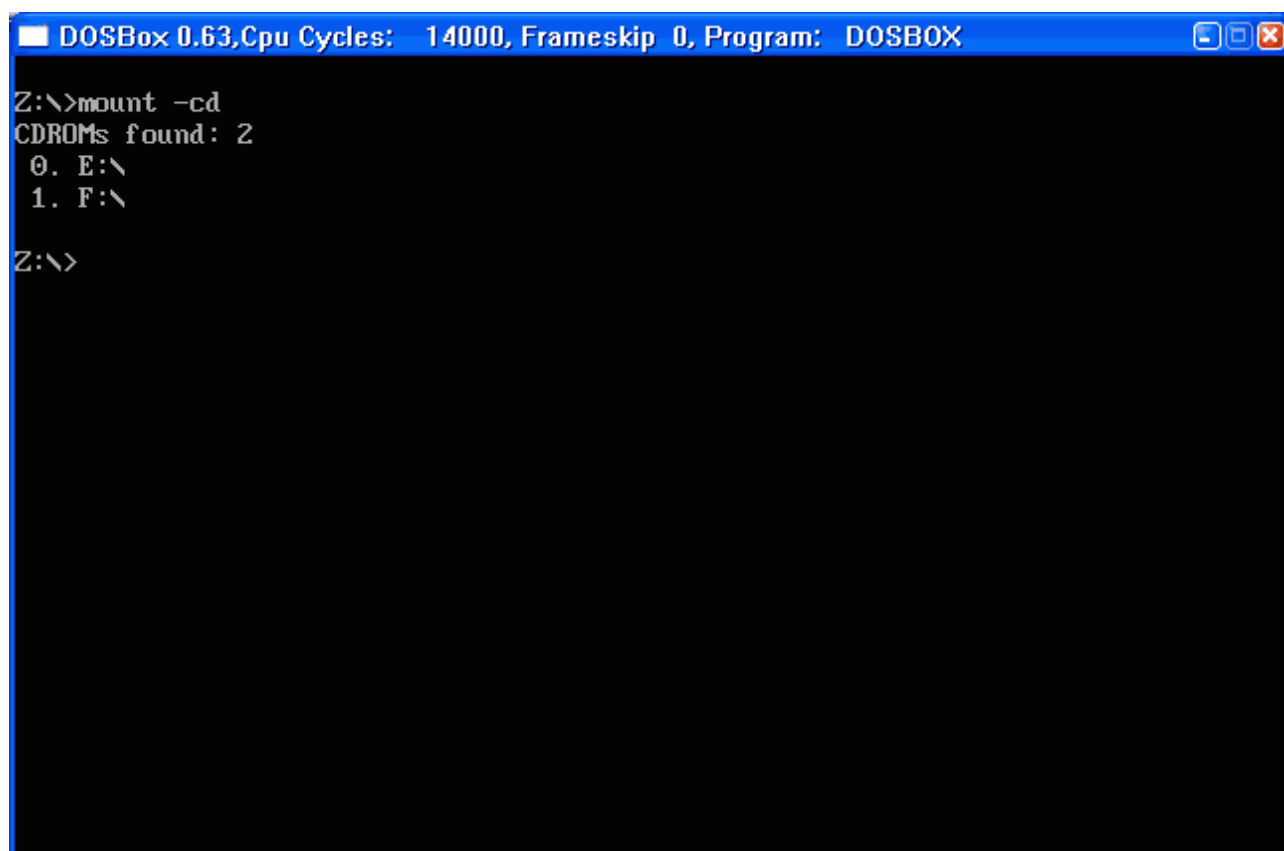
Usa el soporte SDL para emular la unidad de CD-ROM. Las unidades de CD-ROM de nuestro ordenador y sus números correspondientes podemos verlos con la opción *-cd*.

```
Z:\> mount d e:\ -t cdrom -usecd 0
```

### **-cd**

Muestra todas las unidades detectadas de CD, y sus números de identificación para su uso con la opción *-usecd*.

```
Z:\> mount -cd
```



```
DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX
Z:\>mount -cd
CDROMs found: 2
0. E:\
1. F:\
Z:\>
```

**-u**

Eliminamos la unidad que deseemos de las que tenemos montadas. Obviamente, la única que no podremos desmontar será la Z:...

```
Z:\> mount c c:\prog
```

```
Drive C is mounted as local directory C:\PROG\
```

(Creamos la unidad C:).

```
Z:\> mount -u c
```

```
Drive C has succesfully been removed.
```

(Y la eliminamos).

Bueno, si sabemos utilizar la utilidad **mount** y dominamos el funcionamiento básico de MS-DOS, ya tenemos la gran mayoría del trabajo hecho, y podremos usar DOSBox para ejecutar juegos y programas sin grandes complicaciones. Lo que quedaría por comentar serían refinamientos para sacarle el máximo partido al emulador.

Por ejemplo, valdría la pena mencionar las restantes utilidades de DOSBox, si bien para el usuario de a pie no tendrán mucho uso. Información detallada sobre la utilización de cada una de ellas podemos encontrarla en el archivo *README* que viene con el emulador.

**MEM:** Muestra la cantidad de memoria (convencional, extendida y expandida) que queda libre en la emulación. Una versión simplificada de la clásica utilidad de MS-DOS.

**CONFIG [-writeconf] [-writelang]:** Guarda la presente configuración del DOSBox en un archivo. Por supuesto, este fichero estará localizado en alguna unidad física del ordenador, y no en las emuladas con **mount**.

El archivo de configuración -que usualmente se llama *dosbox.conf*- controla diversas opciones de funcionamiento de DOSBox: la cantidad de memoria emulada, el modo de renderización gráfica -OpenGL, DirectDraw, etc-, y muchas más.

Si este archivo no existe, el emulador se inicia con las opciones por defecto. Pero con el comando **config** podemos crearlo, para modificarlo después cómodamente y configurar DOSBox a nuestro gusto. Por ejemplo:

```
Z:\> config -writeconf dosbox.conf
```

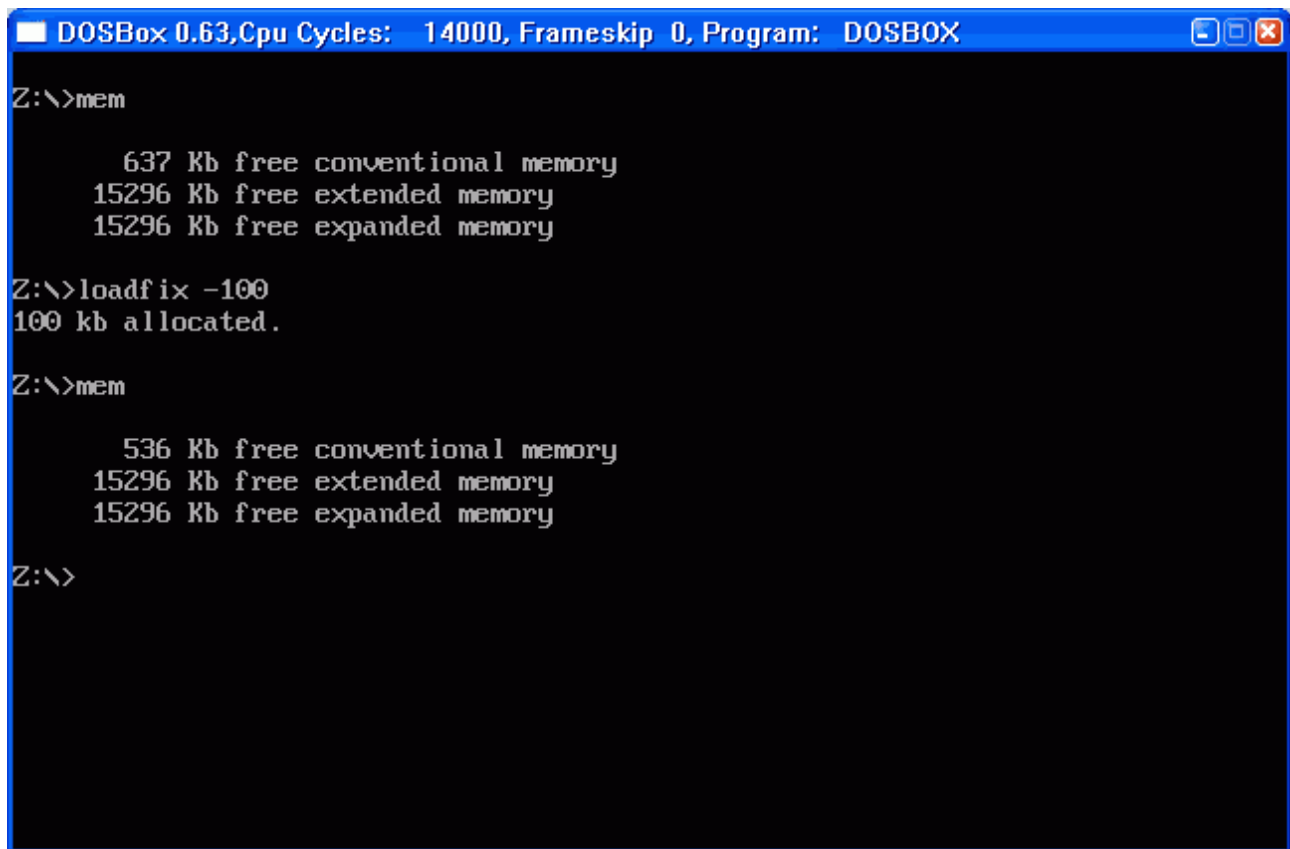
Crea el fichero de configuración *dosbox.conf*.

**LOADFIX [-memoria] [programa a ejecutar]**

**LOADFIX -f:** Puede suceder que, a la hora de ejecutar algunos programas realmente antiguos, nos encontremos con que tienen demasiada memoria convencional disponible. **Loadfix** se encarga de remediar esto; por así decir, "engulle" porciones de memoria, reduciendo de este modo la

cantidad que queda libre.

**-memoria:** La cantidad de memoria, en KB, que haremos desaparecer. Si no especificamos nada, será de 64 KB.



```
DOSBox 0.63,Cpu Cycles: 14000, Frameskip 0, Program: DOSBOX
Z:\>mem

        637 Kb free conventional memory
        15296 Kb free extended memory
        15296 Kb free expanded memory

Z:\>loadfix -100
100 kb allocated.

Z:\>mem

        536 Kb free conventional memory
        15296 Kb free extended memory
        15296 Kb free expanded memory

Z:\>
```

**-programa a ejecutar:** También es opcional. Simplemente podemos arrancar cualquier archivo ejecutable directamente con **loadfix**.

Por ejemplo:

**Z:\> loadfix -100 d:\narkanoid\doh.exe**

**-f:** Con esta opción liberamos toda la memoria que hayamos consumido con anteriores usos de **loadfix**.

**RESCAN:** Hace que DOSBox vuelva a leer el contenido de todas las unidades emuladas. Útil si, por ejemplo, hemos modificado alguna de ellas desde fuera del emulador -añadiendo o eliminando algún fichero, etc-.

**MIXER:** Para controlar el sonido emulado. Podemos ver el volumen de cada uno de los canales tecleando simplemente **mixer**. Ahora, también nos permite hacer cambios del siguiente modo:

**mixer canal izquierda:derecha**

El canal puede ser uno de los siguientes: MASTER, DISNEY, SB, GUS, SPKR, FM. Y los valores "izquierda:derecha" hacen referencia a un porcentaje de volumen (o sea, un número del uno al cien).

**IMGMOUNT:** Con esta utilidad podremos montar imágenes de disco y CD-ROM para usarlas en DOSBox. Los usuarios de máquinas virtuales como Bochs o QEMU estarán familiarizados con la idea; una imagen de disco es un archivo (por ejemplo, *c.img*, *linux.img*, etc) que contiene un disco duro emulado. Esto ya es un tanto complicado, y no voy a tratarlo aquí; en el archivo *README* que viene con DOSBox encontraréis una explicación detallada.

De todos modos, no necesitaremos usar imágenes de disco en DOSBox, a no ser que tengamos algo muy especial que hacer.

**BOOT:** Sirve para arrancar imágenes de disco (bien de disquettes o de rígidos) con otros sistemas operativos en DOSBox; por ejemplo, una imagen de un disquette de arranque de MS-DOS.

Para ver información detallada sobre su uso, leed el archivo *README*.

Por último, me quedaría comentar que DOSBox también permite conectar por internet, emulando el viejo protocolo IPX en nuestras conexiones UDP/IP. Esto debe ser activado en el archivo de configuración -por defecto suele venir inactivo-.

Luego, ya dentro del emulador, contamos con el comando IPXNET para trabajar con nuestra conexión. Yo no tengo conocimientos de primera mano sobre este aspecto concreto de DOSBox, así que no comentaré nada aquí. De todos modos, esto viene bastante bien explicado en el archivo *README*; aunque para sacarle provecho serán necesarios al menos las nociones más básicas sobre redes (qué es una IP, un *ping*, un puerto, un cliente, un servidor..., esas cosas).

Si alguna vez tengo ocasión de hacer pruebas prácticas con esto, ya actualizaría este manual con una sección al respecto. Mientras tanto, no tengo nada más que añadir...

## TECLAS BÁSICAS

Hay varias combinaciones de teclas que, mientras ejecutamos DOSBox, nos permitirán modificar ciertas opciones o llevar a cabo diversas tareas. Conocerlas es prácticamente imprescindible, aunque hayan algunas más importantes que otras:

- ◆ ALT + INTRO: Cambiamos entre el modo de ejecución en pantalla completa y el de ventana.
- ◆ CTRL + F1: Inicia las opciones de configuración de las teclas.
- ◆ CTRL + F4: Recarga el contenido de todas las unidades montadas. También sirve para cambiar entre imágenes de disco.
- ◆ CTRL + F5: Guarda una captura de pantalla (en formato PNG).
- ◆ CTRL + F6: Comienza / termina la grabación de los sonidos emitidos por el emulador en un archivo WAV.
- ◆ CTRL + ALT + F7: Comienza / termina la grabación de comandos OPL.
- ◆ CTRL + ALT + F8: Comienza / termina la grabación de comandos MIDI.
- ◆ CTRL + F7: Disminuye el salto de frames.
- ◆ CTRL + F8: Aumenta el salto de frames.
- ◆ CTRL + F9: Cierra DOSBox.
- ◆ CTRL + F10: Captura o libera el puntero del ratón.
- ◆ CTRL + F11: Reduce la velocidad de la emulación (disminuyendo el número de ciclos).
- ◆ CTRL + F12: Aumenta la velocidad de la emulación.

Éstas son las combinaciones de teclas que utiliza el emulador por defecto; sin embargo, podemos cambiarlas desde las opciones que aparecen al pulsar CTRL + F1.

Las capturas de pantalla y archivos de sonido que grabemos con DOSBox se almacenarán dentro del subdirectorio "*capture*". Y cuidado, porque si éste no existe DOSBox simplemente no guardará nuestras capturas.



## ACELERANDO LA EMULACIÓN

Bien, llegados a este punto, es de suponer que ya controlamos los aspectos básicos de DOSBox, y seguramente ya habremos hecho algunas pruebas con él... En este caso, probablemente -a no ser que hayamos ejecutado *software* realmente viejo- nos hayamos encontrado con un funcionamiento demasiado lento; sobre todo, si nuestro ordenador tiene ya un poco de "veteranía", o si estamos intentando correr algún juego exigente, como pudiera ser por ejemplo Doom. Esto, claro, no es deseable.

De hecho, muchos huyen de DOSBox cuando ven que sus títulos favoritos son emulados a la velocidad de una babosa renqueante, sin percatarse de todas esas opciones que pueden configurarse para conseguir mejores resultados. Y esto es precisamente lo que trataré aquí; consejos para optimizar el funcionamiento del emulador, y sacarle el máximo partido.

Empezaremos por las cosas básicas; esas que deberían resultar obvias, pero en las que no repara mucha gente...

Lo primero sería, antes de poner en marcha el DOSBox, cerrar todas las aplicaciones innecesarias, aunque estén ejecutándose en segundo plano. Tan sólo deberíamos dejar las más imprescindibles -a no ser que nuestro ordenador vaya sobrado, claro-. No hay que ser muy despierto para percatarse de que si utilizamos DOSBox mientras descargamos películas con el Emule, Firefox tiene diez pestañas abiertas y estamos escaneando un disco, los resultados no van a ser los mejores. En Linux no estaría de más desactivar escritorios virtuales; al menos, si vamos bastante apurados.

También es muy importante ejecutar el emulador en *pantalla completa* (podemos cambiar entre ventana y pantalla completa pulsando ALT + INTRO). Así podemos ganar en ocasiones más del doble de velocidad; mostrar el escritorio y al mismo tiempo la ventana de DOSBox consume muchos recursos del ordenador de un modo innecesario.

Y ahora vamos con el interesante tema de los "ciclos".

Como habréis visto en el apartado anterior, en DOSBox tenemos dos combinaciones de teclas (CTRL + F11 y CTRL + F12) para cambiar el número de ciclos en la emulación. O sea, para modificar la velocidad a la que está trabajando DOSBox. Este número lo veremos, cuando estemos en modo ventana, en su barra superior.

Pulsando CTRL + F12 aumentamos el número de ciclos, lo cuál influye decisivamente en el rendimiento. La configuración por defecto suelen ser 2000 ciclos; pero se nos pueden quedar muy cortos cuando ejecutemos *software* pensado para máquinas 386 o 486, así que a menudo tendremos que incrementarlos.

No hay un límite estricto al número de ciclos que podemos especificar en el emulador. Sin embargo, nuestra CPU sí lo tiene. Quiero decir, que aumentando los ciclos incrementamos también los recursos del ordenador que

está utilizando DOSBox; y estos sí que se agotan... Pasados ciertos valores -que dependen en todo caso de la potencia de nuestra máquina-, hacer crecer el número de ciclos ya no nos servirá de nada. Aun más; resulta contraproducente, y suele enlentecer el emulador. Así que en vez de pulsar CTRL + F12 a lo loco, deberíamos controlar los recursos del sistema que estamos empleando, para no pasarnos.



Para acelerar las cosas un poco más, podemos trastear también un poco con el salto de *frames*; para ello tenemos las combinaciones de teclas CTRL + F7 y CTRL + F8.

¿Y qué es un *frame*? Pues cada una de las imágenes generadas por la CPU que son mostradas en nuestro monitor; ni más ni menos. Esto funciona un poco como la técnica de los dibujos animados. Por ejemplo, una animación de una pelotita cruzando nuestra pantalla está elaborada en realidad con un número *x* de *frames* -imágenes fijas-, que al sucederse rápidamente crean una ilusión de movimiento.

En DOSBox, el salto de *frames* por defecto suele ser de 0; esto es, todas las imágenes generadas por el programa que estemos ejecutando serán reproducidas por el emulador. Sin embargo, no siempre podremos permitirnoslo, puesto que esto requiere bastantes recursos. Así que en ocasiones cambiar este valor puede sernos de ayuda.

Por ejemplo, especificando un salto de *frames* de 1, DOSBox mostrará una imagen e ignorará otra. En la práctica, supone reducir el número de *frames* reproducidos a la mitad, lo cual aumentará significativamente la

velocidad de la emulación.

Ahora, esto tiene su parte negativa. Saltos de *frames* demasiado grandes pueden restarle mucha fluidez al juego que estemos ejecutando, haciéndolo funcionar a golpes, con lo que estaríamos perdiendo por una parte lo que ganamos por otra. Por lo que yo mismo he ido viendo, saltos de *frames* mayores de 4-5 ya comienzan a resultar problemáticos en muchos juegos.

Todavía hay más cosas que podemos hacer para acelerar DOSBox.

Por ejemplo, hay ciertos aspectos del sonido que deberíamos tener en cuenta. La emulación de Gravis Ultrasound es algo más rápida que la de SoundBlaster, así que deberíamos escoger esta tarjeta de sonido al configurar los juegos que queramos usar -vamos, si la soportan, cosa que no se cumple en muchos-. Así mismo, para la música MIDI resulta preferible General Midi, en lugar de Adlib o SoundBlaster.

Otra solución es bajar la calidad del sonido de todos los dispositivos de audio emulados. Por defecto suele estar a 22050 Hz, pero podemos dejarlo en 11025 Hz. Para esto deberemos editar el archivo de configuración del DOSBox, el *dosbox.conf* (ver siguiente capítulo).

En casos más extremos, podemos simplemente desactivar el sonido de ese juego que tanto se resiste a funcionar con fluidez... Esto reduce los recursos utilizados por DOSBox y mejora un poco el rendimiento.

También tenemos la posibilidad de cambiar el "cpu core", de "normal" (su valor por defecto) a "dynamic"; esto se hace editando el archivo *dosbox.conf*, cosa que explico en el siguiente capítulo. El modo "dynamic" nos proporciona una velocidad mayor, pero como contrapartida resulta más inestable. Si os decidís a usarlo, grabad frecuentemente vuestras partidas para evitar disgustos.

Por lo demás, el "dynamic core" sólo funcionará en procesadores x86 (vamos, los PCs típicos). Así que quién quiera ejecutar DOSBox en otro tipo de máquina (un Mac, por ejemplo), mejor que se olvide de esta opción.

Por último, los juegos que funcionan en el modo protegido de MS-DOS (son fáciles de reconocer, puesto que para su arranque utilizan el archivo *DOS4GW.EXE*) exigen bastantes recursos y pueden darnos problemas. Con ellos es preferible usar *DOS32A*; lo explico en el capítulo correspondiente, más adelante.

## EL ARCHIVO DOSBOX.CONF

DOSBox cuenta con un archivo de configuración *-dosbox.conf*, situado en la misma carpeta que el ejecutable- que podemos modificar para arreglar a nuestro gusto el funcionamiento del emulador. Cualquier editor de texto -Bloc de Notas, Emacs, etc- nos servirá para hacerlo.

Puede suceder -aunque no es lo normal en la mayoría de versiones de DOSBox- que este archivo no esté presente. En tal caso deberíamos crearlo, arrancando DOSBox y ejecutando la utilidad **Config**. En el capítulo de "Uso de DOSBox" está explicado este detalle.

Bueno, hemos abierto el archivo *dosbox.conf* y estamos listos para modificarlo... Comentaré las opciones más importantes, aunque suelen venir explicadas dentro del mismo fichero, con comentarios en inglés.

**fullscreen= [false, true]**

Especifica si DOSBox arranca en un principio en una ventana, o en el modo de pantalla completa. El valor por defecto es *false* (se inicia en una ventana), pero podemos cambiarlo a *true* si queremos iniciar el emulador directamente en pantalla completa.

**output= [surface, overlay, ddraw, opengl, openglnb]**

Aquí establecemos el modo gráfico en el que trabajará DOSBox. Podemos elegir los siguientes valores: *surface*, *overlay*, *opengl*, *openglnb*, *ddraw*. Siempre podemos ir probándolos para ver cuál nos da mejores resultados; aunque la mayoría de la gente suele dejar el valor por defecto, *surface*. No obstante, *ddraw* también debería funcionar bien en sistemas Windows; así mismo, valdría la pena hacer pruebas con el valor *opengl* si contamos con una buena tarjeta gráfica -y especialmente si es de Nvidia-.

**memsize= [valor]**

Especifica los *megabytes* de memoria RAM que emulará DOSBox. El valor por defecto suele ser de 16, aunque podemos escoger otro que se adapte mejor a nuestras necesidades o a las posibilidades de nuestro ordenador.

**machine= [vga, cga, tandy, hercules]**

El tipo de tarjeta gráfica que emulará DOSBox. La opción por defecto es *vga*, que servirá para la mayoría de los viejos títulos de MS-DOS. Juegos programados para EGA o CGA deberían funcionar también con este valor. Sin embargo, para emular *software* que sólo admita Tandy o Hercules -que es muy poco, en realidad-, tendremos que usar las opciones correspondientes.

Por cierto, con el valor *vga* también podremos usar los modos en alta resolución VESA -empleados por algunos juegos de mitad de los noventa, como Duke Nukem 3D, etc-..., vamos, si nuestra CPU puede con ello.

### **captures= [carpeta]**

Especifica el subdirectorio en el que DOSBox almacenará las capturas de pantalla, grabaciones de sonido..., que vayamos haciendo. Por defecto es la carpeta "captures".

### **frameskip= [valor]**

Asignamos el salto de *frames* con el que funcionará DOSBox por defecto. El valor usual es cero, pero podemos usar cualquier otro que se adapte a nuestras necesidades. Para más información sobre los saltos de *frames*, echad una ojeada al capítulo "[Acelerando la emulación](#)".

### **core= [normal, full, simple, dynamic]**

Núcleo de CPU usado en la emulación. El valor por defecto es *normal*. Ahora, si queremos acelerar un poco el funcionamiento de DOSBox, podemos cambiarlo a *dynamic*. Sin embargo, este modo resulta más inestable que el normal (ver el capítulo "[Acelerando la emulación](#)").

### **cycles= [valor]**

El número de ciclos (instrucciones emuladas por milisegundo) con el que trabajará DOSBox al iniciarse. Suelen ser 2000, pero podemos cambiar este valor a otro más acorde con lo que necesitamos. Así nos ahorramos ajustar el número de ciclos con CTRL + F12 y CTRL + F11 cada vez que arranquemos el emulador.

En el capítulo "[Acelerando la emulación](#)" encontraréis un comentario más amplio sobre cómo ajustar los ciclos.

### **cycleup= [valor]**

### **cycledown= [valor]**

*Cycleup* determina el incremento del número de ciclos que obtendremos al pulsar CTRL + F12; por ejemplo, 500 o 1000 podrían ser buenos valores. *Cycledown* especifica la reducción de ciclos que conseguiremos con cada pulsación de CTRL + F11.

En ambas opciones, valores por debajo de 100 dejan de hacer referencia a una cantidad fija de ciclos, y son interpretados como porcentajes del número total.

### **rate= [valor]**

Especifica la calidad del sonido que reproducirá DOSBox. El valor por defecto es 22050 (22050 Hz), pero podemos bajarlo a 11025 si queremos acelerar un poco la emulación.

Por último, la calidad que asignemos al sonido de las tarjetas emuladas (SoundBlaster, Gravis Ultrasound, etc) no debería superar este valor, que viene a ser una especie de control general. De lo contrario, la emulación de estas tarjetas puede brindar unos resultados bastante pobres.

```
mpu401= [true, false]
gus= [true, false]
pcspeaker= [true, false]
disney= [true, false]
```

Con estas líneas activamos o desactivamos la emulación de diversos dispositivos de sonido; MPU401, Gravis Ultrasound (*gus*) y Disney Sound Source (*disney*). La opción *pcspeaker* hace referencia al altavoz interno del ordenador. Lo normal -y más recomendable- es que todas ellas estén activadas con el valor "true".

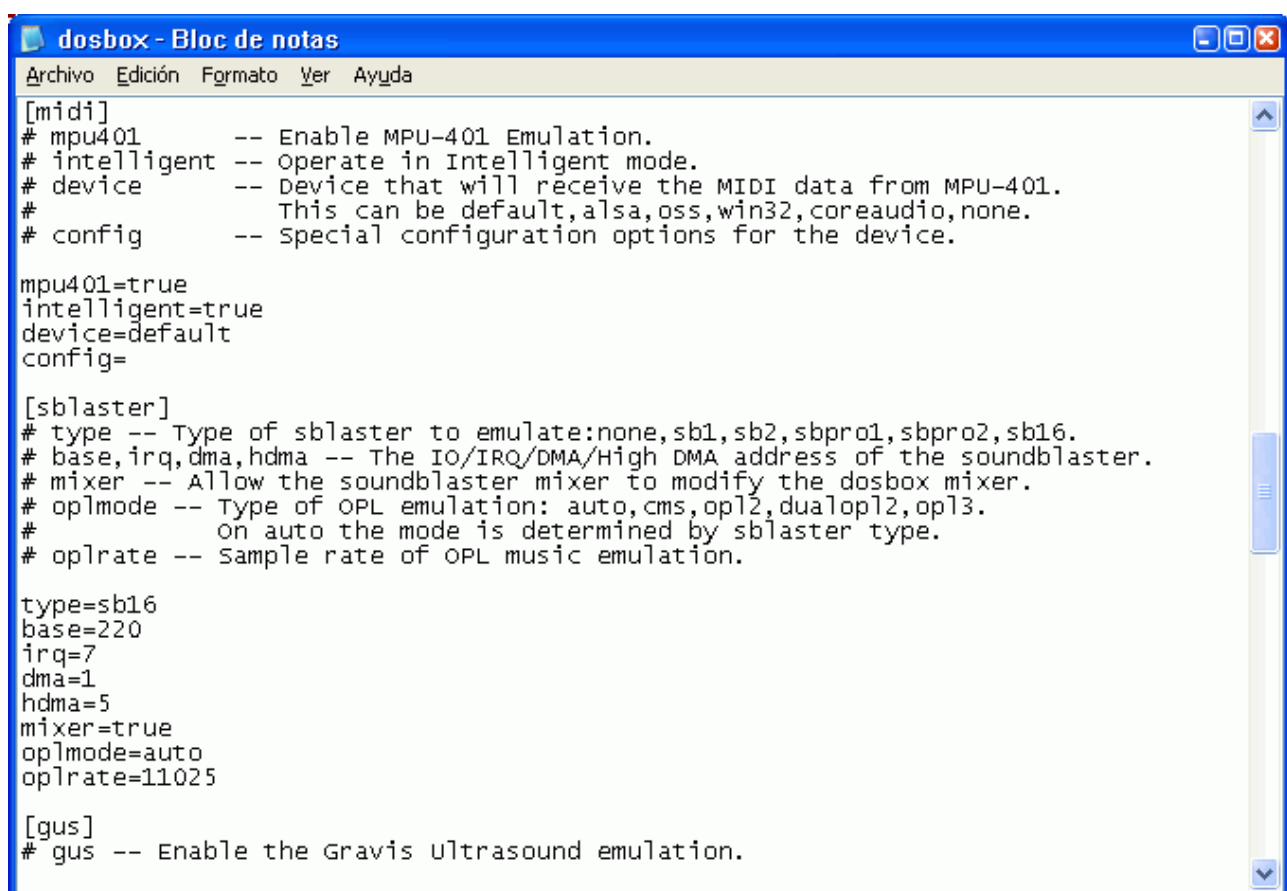
En las opciones de configuración de cada tarjeta podremos modificar la calidad del sonido emulado -recomendable no ponerla más alta que la del mezclador general de DOSBox-, así como sus valores de *hardware*.

Éstos últimos en realidad no deberíamos tocarlos sin tener muy claro lo que estamos haciendo. Los valores por defecto de las dos tarjetas que los emplean son los siguientes:

*Base=220, irq=7, dma=1, hdma=5* (SoundBlaster).

*Base=240, irq1=5, irq2=5, dma1=3, dma2=3* (Gravis Ultrasound).

Aquí tenemos parte del archivo *dosbox.conf*; como podréis ver, su sintaxis es bastante sencilla:



```
[midi]
# mpu401      -- Enable MPU-401 Emulation.
# intelligent -- Operate in Intelligent mode.
# device      -- Device that will receive the MIDI data from MPU-401.
#             This can be default,alsa,oss,win32,coreaudio,none.
# config      -- Special configuration options for the device.

mpu401=true
intelligent=true
device=default
config=

[sblaster]
# type -- Type of sbaster to emulate:none,sb1,sb2,sbpro1,sbpro2,sb16.
# base,irq,dma,hdma -- The IO/IRQ/DMA/High DMA address of the soundblaster.
# mixer -- Allow the soundblaster mixer to modify the dosbox mixer.
# oplmode -- Type of OPL emulation: auto,cms,opl2,dualopl2,opl3.
#          On auto the mode is determined by sbaster type.
# oplrate -- Sample rate of OPL music emulation.

type=sb16
base=220
irq=7
dma=1
hdma=5
mixer=true
oplmode=auto
oplrates=11025

[gus]
# gus -- Enable the Gravis Ultrasound emulation.
```

```
xms= [true, false]
ems= [true, false]
```

Activan o desactivan la emulación de memoria extendida (XMS) y expandida (EMS). Por defecto, ambas estarán en funcionamiento -valores *true*-. No es algo que nos deba preocupar mucho. Hay juegos que requieren memoria expandida para funcionar, y otros que no la permiten; pero ambos casos son muy minoritarios.

Y ahora pasamos a la última parte del archivo de configuración de DOSBox, y una de las más interesantes. Al final del texto probablemente encontraremos dos líneas que dicen así:

```
[autoexec]
# Lines in this section will be run at startup.
```

Bien, quiénes hayan usado MS-DOS se acordarán del archivo *AUTOEXEC.BAT* -que de hecho, también está presente en Windows 95/98/Me-. Un fichero que ejecutaba diversos comandos al iniciar el sistema...

Pues bien, la etiqueta *autoexec* de DOSBox tiene el mismo cometido.

Por ejemplo, imaginemos que tenemos por costumbre, al poner en marcha el emulador, montar la unidad *C:* de nuestro disco duro. Así, cada vez que iniciásemos una sesión con DOSBox, tendríamos que escribir un comando **Mount:**

```
Z:\> mount c c:\
```

Pero en realidad no sería necesario repetir esta orden siempre que arrancásemos el emulador. Lo mejor sería escribirla bajo la etiqueta *autoexec* del *dosbox.conf*:

```
[autoexec]
# Lines in this section will be run at startup.
mount c c:\
```

Y así, el comando se ejecutaría al iniciarse DOSBox, de manera que tendríamos la unidad *C:* montada automáticamente. Esto es algo muy práctico y cómodo para los usuarios habituales de DOSBox.

Por ejemplo, éste es mi *autoexec*:

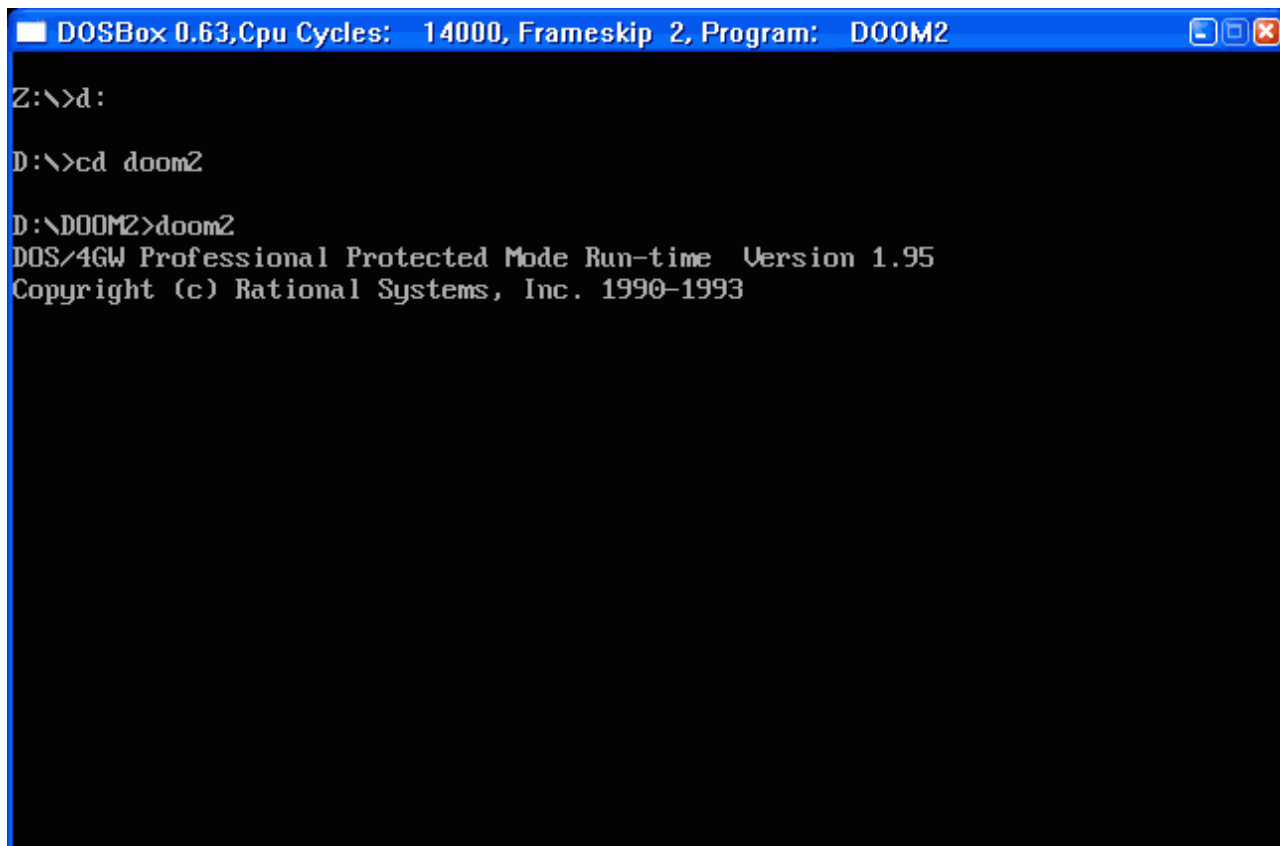
```
[autoexec]
# Lines in this section will be run at startup.
mount c c:\
path=c:\media\keyb;z:\
keyb sp
```



## JUEGOS EN MODO PROTEGIDO: DOS32A

A mediados de los noventa, algunos juegos y programas comenzaron a utilizar el modo protegido de MS-DOS. Éstos son fáciles de reconocer, puesto que todos ellos emplean el fichero *DOS4GW.EXE*, un antiguo extensor de MS-DOS. Por ejemplo, tenemos títulos tan famosos como Doom, Syndicate, etc.

En esta captura de pantalla, que he tomado mientras se ejecutaba el Doom 2, podemos ver como se inicia el *DOS4GW* antes de proceder a arrancar el juego:



```
DOSBox 0.63,Cpu Cycles: 14000, Frameskip 2, Program: D00M2
Z:\>d:
D:\>cd doom2
D:\DOOM2>doom2
DOS/4GW Professional Protected Mode Run-time Version 1.95
Copyright (c) Rational Systems, Inc. 1990-1993
```

La cuestión es que los juegos que emplean el modo protegido pueden resultar problemáticos en DOSBox. Requieren muchos recursos, muy a menudo la emulación no es demasiado rápida, y en algunas ocasiones -raras, también hay que decirlo; al menos con las últimas versiones de DOSBox- se vuelven inestables.

Para estos inconvenientes hay una solución: usar *DOS32A*.

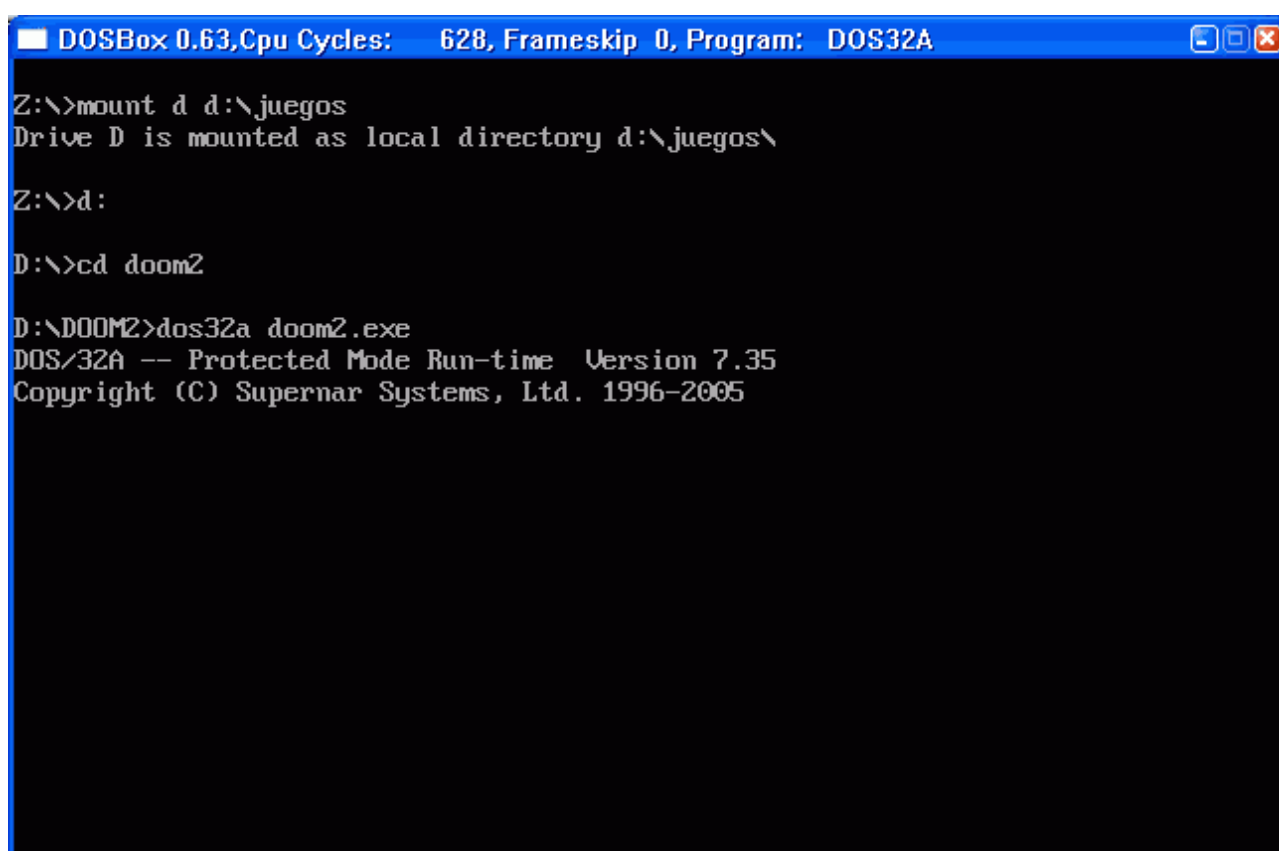
*DOS32A*, al igual que *DOS4GW*, se trata de un extensor de MS-DOS. Ahora, resulta mucho más moderno -de hecho, es un proyecto vivo y en desarrollo-. Se distribuye gratuitamente, bajo licencia GPL si no recuerdo mal. Y sobre todo, ofrece un mejor rendimiento que el viejo *DOS4GW*; mayor velocidad de ejecución, y más estabilidad.

*DOS32A* podemos usarlo en DOSBox o en cualquier sistema MS-DOS (ya

sea una máquina emulada con QEMU, VMWare, Bochs..., o un PC real); en cualquiera de estos casos merece la pena.

Lo primero que deberíamos hacer es descargarlo de la página web del proyecto: <http://dos32a.sourceforge.net/>. Luego descomprimos el fichero que nos hayamos descargado dónde nos apetezca. Posiblemente, éste contendrá varios archivos ejecutables (EXE); pero el que nos interesa es el *DOS32A.EXE*.

Siguiendo con el ejemplo de antes, vamos suponer que queremos utilizar *DOS32A* para jugar a Doom 2. El asunto es muy sencillo. Primero copiamos el archivo *DOS32A.EXE* en la carpeta del juego. Y luego, entramos en DOSBox y arrancamos Doom 2 del siguiente modo:

A screenshot of a DOSBox window. The title bar reads "DOSBox 0.63,Cpu Cycles: 628, Frameskip 0, Program: DOS32A". The command prompt shows the following sequence of commands and output:

```
Z:\>mount d d:\juegos
Drive D is mounted as local directory d:\juegos\

Z:\>d:

D:\>cd doom2

D:\DOOM2>dos32a doom2.exe
DOS/32A -- Protected Mode Run-time Version 7.35
Copyright (C) Supernar Systems, Ltd. 1996-2005
```

Y ya lo tenemos funcionando.

Así que el uso de *DOS32A* es el siguiente:

**dos32a (archivo .exe a ejecutar)**

Lo más cómodo es tener una copia del *DOS32A* en la carpeta de cada juego que queramos ejecutar de este modo. El programa es muy pequeño -menos de 50k-, y no vale la pena andar cortando y pegando.

## DUDAS FRECUENTES

Pues eso. Aquí trato de explicar algunas cuestiones que dan dolores de cabeza a muchos usuarios de DOSBox. Algunas han sido tratadas a lo largo de este manual, y otras no.

### ***La unidad de CD-ROM no funciona en DOSBox:***

Antes que nada deberíamos asegurarnos de que la estamos montando de la forma correcta (con la opción `-t cdrom` en el comando **mount**; ver el capítulo de "Uso de DOSBox").

Si aún así sigue sin funcionar adecuadamente, podemos montarlo con emulación a bajo nivel. Por ejemplo:

- ◆ `Z:\> mount d f:\ -t cdrom -usecd 0` (Con soporte SDL).
- ◆ `Z:\> mount d f:\ -t cdrom -usecd 0 -ioctl` (Con soporte IOCTL; en sistemas Windows 2000/NT/XP).
- ◆ `Z:\> mount d f:\ -t cdrom -usecd 0 -aspi` (En Windows 98, con los controladores ASPI instalados).

Tenéis más información en la parte de "Uso de DOSBox".

### ***El ratón no funciona bien:***

Normalmente, DOSBox detecta cuando un juego emulado está utilizando el ratón. Así, si clicamos en la ventana del emulador, el puntero del ratón queda "atrapado" en ella, y podremos usarlo para jugar del modo habitual. Sólo que en ciertas ocasiones esto puede no funcionar, y será necesario que capturemos el ratón pulsando CTRL + F10.

### ***El sonido funciona entrecortado; ¿por qué?***

En primer lugar, deberíamos ejecutar DOSBox en pantalla completa.

Si aun así persiste el problema, se debe a que estamos usando demasiados recursos de nuestra CPU. En tal caso deberíamos bajar el número de ciclos o aumentar el salto de *frames*. También hay que recordar que la emulación de Gravis Ultrasound y MPU401 o General Midi tiene un mejor rendimiento que la de SoundBlaster; si el juego o programa que estemos usando nos da la oportunidad, deberíamos utilizar estas tarjetas.

Otra opción sería incrementar el valor de *prebuffer* en el archivo de configuración de DOSBox -usualmente, *dosbox.conf*-. Esta opción está situada en la etiqueta *mixer*.

Ver los capítulos "Acelerando la emulación" y "El archivo *dosbox.conf*" para una información más general sobre estos asuntos.

## ***¿Qué tarjetas y dispositivos de sonido puede emular DOSBox?***

- ◆ El altavoz interno del PC.
- ◆ Creative CMS/Gameblaster:  
La primera tarjeta de sonido desarrollada por Creative Labs. La configuración por defecto la coloca en el puerto 0x220; esto puede traer conflictos si activamos al mismo tiempo la emulación de Adlib (puesto que ambas usaban en mismo puerto).
- ◆ Tandy.
- ◆ Adlib.
- ◆ SoundBlaster 16, SoundBlaster Pro I / II, SoundBlaster I / II:  
Con la configuración por defecto, DOSBox emula una SoundBlaster 16 con sonido estéreo de 16 bits. Esto podemos cambiarlo editando el archivo *dosbox.conf*.
- ◆ Disney Soundsource.
- ◆ Gravis Ultrasound.
- ◆ MPU401:  
Una interfaz de sonido que funcionará con las tarjetas emuladas General Midi y MT-32.

## ***¿Cómo cambiar el tipo de sistema que está emulando DOSBox?***

Para esto debemos editar el archivo de configuración del emulador, el *dosbox.conf*. En él veremos opciones para modificar la cantidad de memoria RAM que queremos disponible en la emulación, el tipo de tarjeta gráfica -VGA, CGA, etc-, y muchos otros detalles de interés.

Leed el capítulo "[El archivo dosbox.conf](#)" para más información.

## ***El juego o programa no puede leer su CD-ROM:***

Evidentemente, tenemos que asegurarnos de usar la opción *-t cdrom* con el comando **mount**. También podríamos emplear la opción *-label*, asignándole a la unidad emulada la misma etiqueta de volumen que tiene el CD que queremos leer.

Otra posibilidad sería montar la unidad de CD con emulación a bajo nivel. Sobre esto podéis ver el capítulo "[Uso de DOSBox](#)", o la primera pregunta de esta misma sección.

## ***¡Hey! ¿Qué pasa con el teclado? ¿Cómo escribir ciertos signos?***

DOSBox utiliza por defecto una configuración de teclas inglesa. Con nuestros teclados españoles no tendremos problemas para escribir caracteres alfanuméricos -o sea, letras y números-, pero sí con diversos signos de uso común como corchetes, asteriscos, paréntesis, etc. Ocurre que éstos están situados en teclas diferentes.

Para ahorraros una fatigosa búsqueda, aquí hago una pequeña lista con los símbolos más frecuentes y su localización en la configuración de teclas de DOSBox:

CARÁCTER	TECLAS EN DOSBOX
* (Asterisco)	MAYUSC + 8
( (Paréntesis abierto)	MAYUSC + 9
) (Paréntesis cerrado)	MAYUSC + 0
- (Guión)	' (Apóstrofe – Cerrar interrogante)
_ (Subrayado)	MAYUSC + ' (Cerrar interrogante)
= (Igual)	i (Exclamación)
+ (Más)	MAYUSC + i
; (Punto y coma)	Ñ
: (Dos puntos)	MASYUC + Ñ
/ (Barra de división)	- (a la izquierda de MAYUSC)
\ (Barra invertida)	<, Ç
(Barra vertical)	MAYUSC + Ç

Por cierto, y al contrario de lo que sucede en MS-DOS, en DOSBox las barras (/ y \) son intercambiables a la hora de especificar rutas de directorios.

Claro que podemos cambiar la configuración del teclado con CTRL + F1 mientras trabajamos con el emulador. O bien instalarnos el *KEYB.COM* de MS-DOS o FreeDOS -preferiblemente éste último-.

### ***¿Cómo compilar el código fuente de DOSBox en Linux?***

Más arriba, en el capítulo "*Instalando DOSBox*" he dejado unas sencillas indicaciones. De todos modos, en el archivo *INSTALL* -que viene con cualquier versión de DOSBox- encontraremos también instrucciones, y una pequeña lista con las librerías necesarias para la compilación.

### ***DOSBox funciona muy lento:***

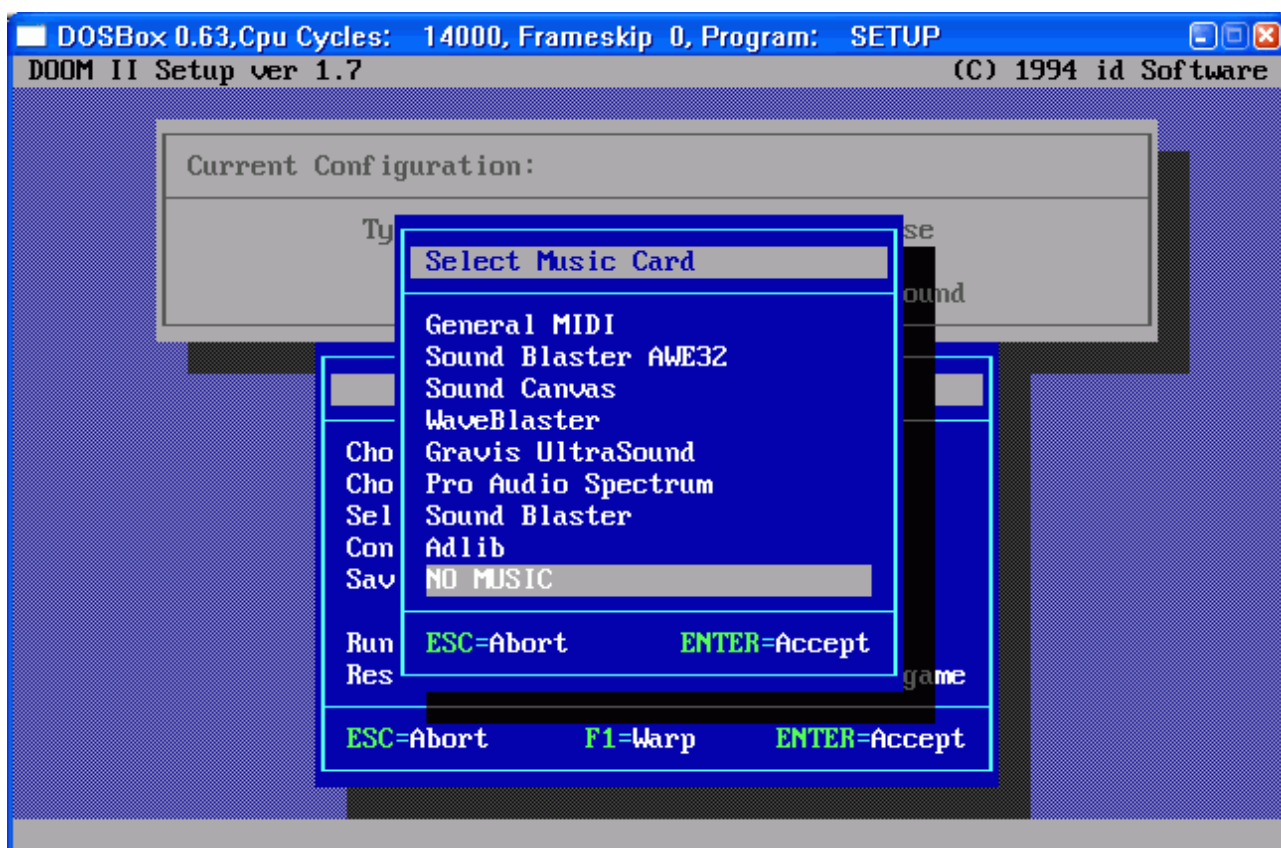
Hay diversos modos de aumentar el rendimiento del emulador y sacarle el máximo partido; sobre esto he hablado largo y tendido en el capítulo "*Acelerando la emulación*".

Por lo demás, no esperéis milagros de DOSBox. Es un emulador, y lo que ejecutemos con él nunca irá tan rápido como una aplicación nativa. Con una CPU de 1 – 1,5 Ghz podríamos emular el equivalente a un 486 de gama baja; con un Pentium II a 300-400 Mhz, podemos conseguir un 286 potente o un 386. A mejor máquina, claro, mejores resultados.

## ¿Cómo configuro una tarjeta de sonido en un juego?

Bueno, muchos juegos simplemente las detectaban automáticamente, sin necesidad que nosotros tuviéramos que tocar nada; por ejemplo, era el caso de Titus the Fox, y unas cuantas decenas más.

Ahora, en otros casos tendremos que configurar el sonido nosotros mismos. Para ello, normalmente hay que ejecutar alguna utilidad de configuración que venga con el juego -suelen llamarse *SETUP* o *INSTALL*-. Por ejemplo, aquí tenemos el *SETUP* del Doom 2:



Aquí escogeremos la tarjeta de sonido que más nos convenga, tanto para reproducir los efectos digitalizados como las melodías MIDI. Ahora, las dificultades vienen a la hora de configurar la tarjeta en cuestión; el ordenador nos preguntará por el puerto (*address*, *port* o *base address*), la interrupción (*IRQ*), y otras cuestiones como el canal *DMA*, etc.

Dado que se trata de tarjetas emuladas, estos valores están determinados por DOSBox. Podemos verlos editando el archivo *dosbox.conf*, y cambiarlos si lo deseamos -cosa que normalmente no necesitaremos hacer-.

Por si las dudas, los valores por defecto son estos:

SoundBlaster: Puerto 220, IRQ 7, DMA 1, HDMA 5.

Gravis Ultrasound: Puerto 240, IRQ 5, DMA 3.

General Midi: Puerto 330.

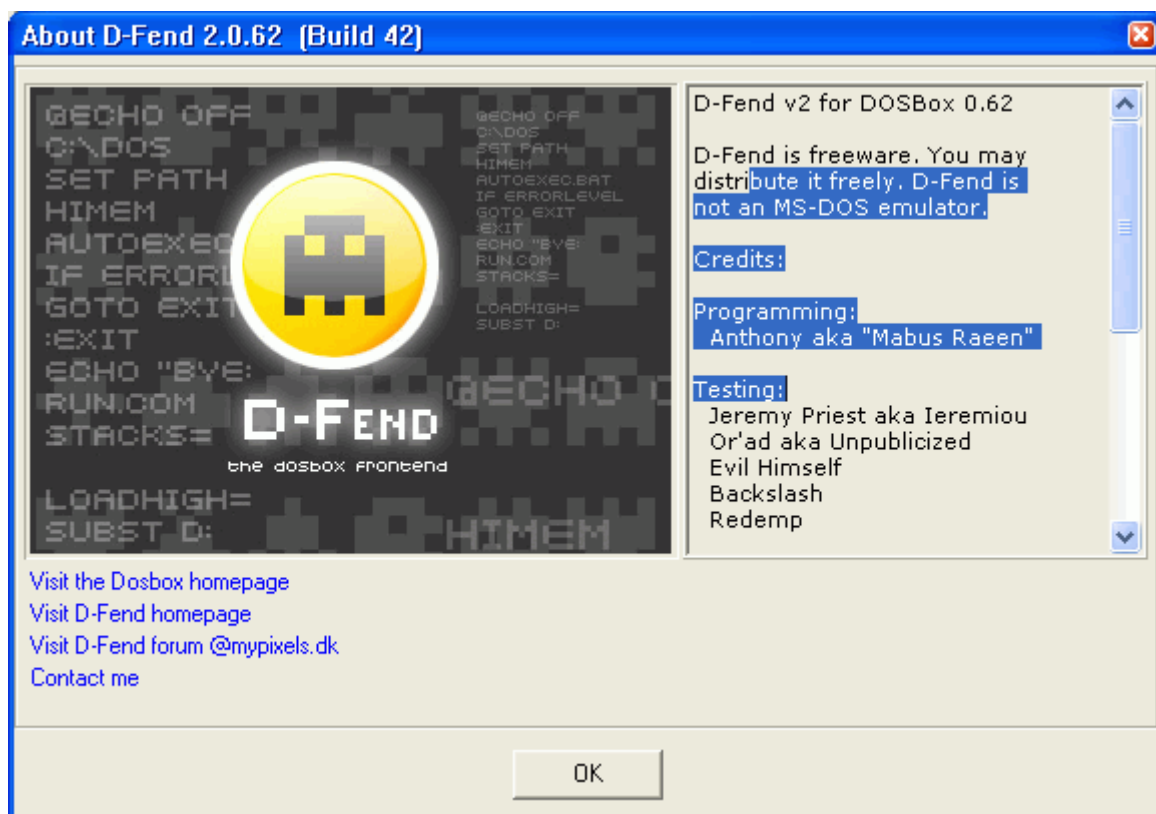
## FRONT-ENDS: D-FEND

El nombre de este capítulo intrigará a más de uno; ¿qué carajo es un *front-end*? ¿Algún animal amazónico? ¿Una herramienta de bricolaje? Pues no; estas palabrejas inglesas hacen referencia, entre otras cosas, a ciertas utilidades que pueden resultar de interés para algunos usuarios de DOSBox.

Los *front-ends* para DOSBox son simplemente pequeños programas que proporcionan una interfaz gráfica para trabajar con él. Vamos, un medio para usar el emulador desde cómodas ventanas de escritorio y sin escribir un comando. Los hay para Windows, para Mac, para Linux, y posiblemente para algún otro sistema.

En principio son más sencillos que emplear DOSBox tal cual, sin añadidos. Y digo "en principio", porque para manejar cualquier *front-end* necesitaremos conocer al menos las nociones básicas del emulador. Usarlos depende más que nada de las preferencias de cada cuál. Ahora, a mí nunca me han parecido un gran invento; para quién ya tiene unos conocimientos básicos, no hay nada tan flexible como utilizar directamente DOSBox.

Bueno, aquí hablaré sobre D-Fend, que es el *front-end* que mejor conozco. De todas formas, supongo que los restantes tendrán un uso bastante parecido.

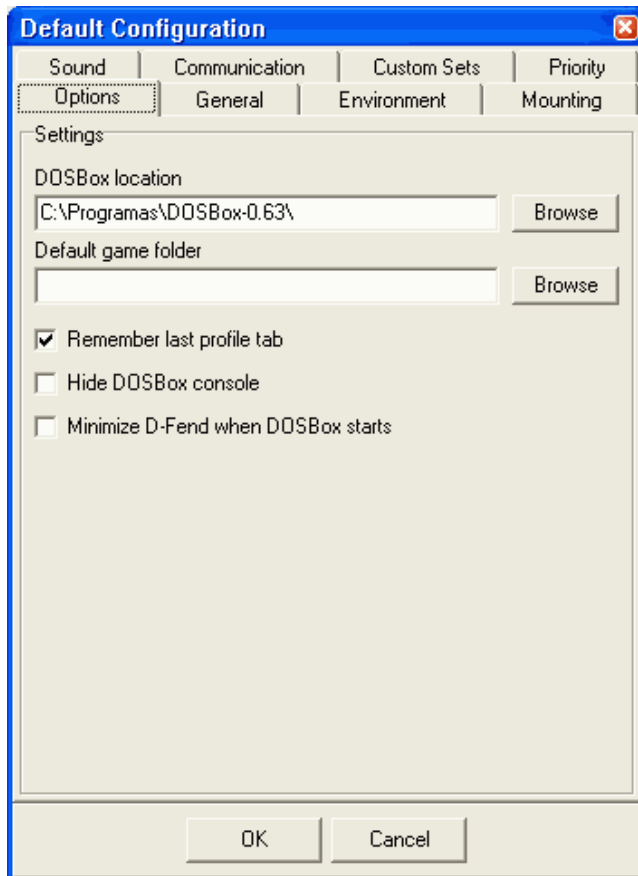


Bueno, instalarlo y configurarlo no ofrece mayor complicación. Podemos descargarlo de su página oficial (<http://members.home.nl/mabus>), aunque tal vez en la web de DOSBox (<http://dosbox.sourceforge.net>) también lo tengan.



Una vez instalado en el disco duro deberemos configurarlo. Lo primero -y más básico- sería especificar la carpeta en la que tenemos el DOSBox. Es posible que, la primera vez que ejecutemos D-Fend, él mismo lo pregunte. De lo contrario, accederemos a las opciones generales de configuración (*configure defaults*), y seleccionaremos la pestaña "Options" en el cuadro de diálogo que nos aparecerá.

Por ejemplo, aquí lo tenemos:



Claro, que una vez aquí podemos aprovechar para cambiar otros detalles, y optimizar un poco la ejecución de DOSBox. En realidad, estaríamos haciendo lo mismo que al editar el fichero de configuración del emulador, el *dosbox.conf*. Si queréis explicaciones detalladas sobre todo lo que podemos modificar aquí, echadle un vistazo al capítulo correspondiente ("El [archivo dosbox.conf](#)").

Bien, en principio sería buena idea abrir la pestaña "General", y marcar allí la opción "Start Fullscreen", para que lo que ejecutemos en DOSBox se haga en pantalla completa -más rápido que una ventana de escritorio-. "Render Method" hace exactamente lo mismo que el valor "output=" del archivo *dosbox.conf*; especificar el modo gráfico en el que trabajará el emulador.

"Environment" también es importante. Aquí podremos elegir cuánta memoria RAM queremos emular, activar o desactivar la emulación de XMS y EMS, y otros detalles significativos.

"*Mounting*" nos servirá para especificar unidades que se montarán automáticamente al iniciarse DOSBox. Es lo mismo que agregar un comando **mount** en el apartado *autoexec* del archivo *dosbox.conf*.

En el apartado "*Sound*", lógicamente, configuraremos todo lo referente a las tarjetas de sonido emuladas. Lo mejor sería dejar los valores que vienen por defecto. Aunque, por ejemplo, si queremos ganar un poco de velocidad al ejecutar DOSBox, podríamos bajar a 11025 Hz el "*Sample Rate*" general, y los valores correspondientes de las tarjetas SoundBlaster y Gravis Ultrasound (ver "*Acelerando la Emulación*" y "*El archivo dosbox.conf*").

La pestaña "*Communication*" controla todo lo referente a conexiones en red. Como nunca he usado DOSBox para esto, poco puedo decir aquí sobre ello. Quizás en un futuro...

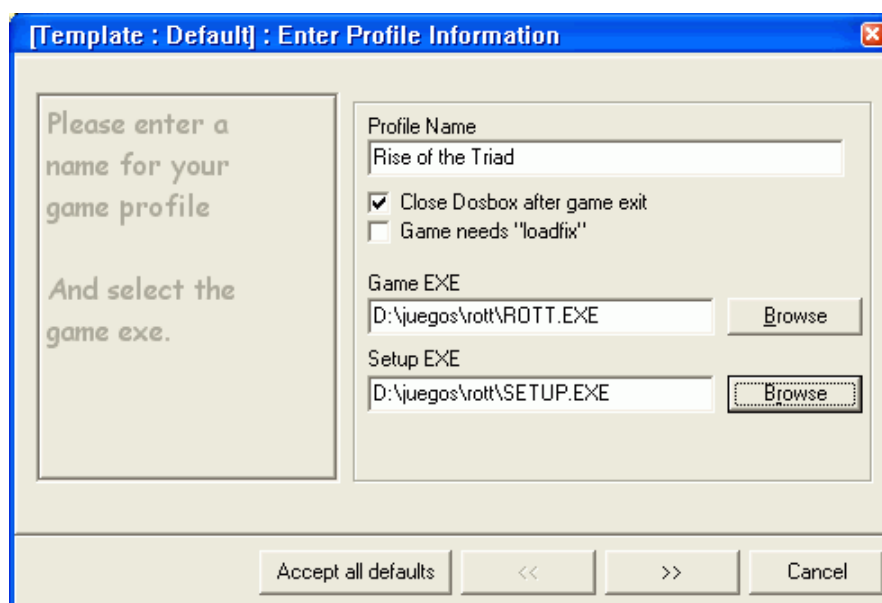
Por último, en "*Custom sets*" podremos introducir valores del archivo *dosbox.conf* que no estén presentes en los menús de D-Fend. Y en la pestaña "*Priority*", elegir la prioridad que tendrá DOSBox entre las demás tareas de nuestro sistema.

Una vez tengamos el D-Fend configurado a nuestro gusto, querremos ejecutar algo con él. Este programa funciona mediante perfiles; cada juego o utilidad que queramos emular en DOSBox tendrá sus propias opciones de ejecución almacenadas en un perfil. Y para ponerlo en marcha, simplemente haremos doble clic con el ratón en el perfil correspondiente.

Ahora, estos perfiles tendremos que ir creándolos nosotros; la cosa resulta sencilla, siempre y cuando conozcamos lo básico del uso de DOSBox. Y para ver cómo se hace, nada como un ejemplo; vamos a elaborar un perfil para ejecutar el juego Rise of the Triad.

Así que empecemos. Lo primero es abrir el "*profile wizard*". Depende de la versión de D-Fend que estemos usando, lo encontraremos en un icono bajo la barra de menús, o dentro del menú "*Profile*".

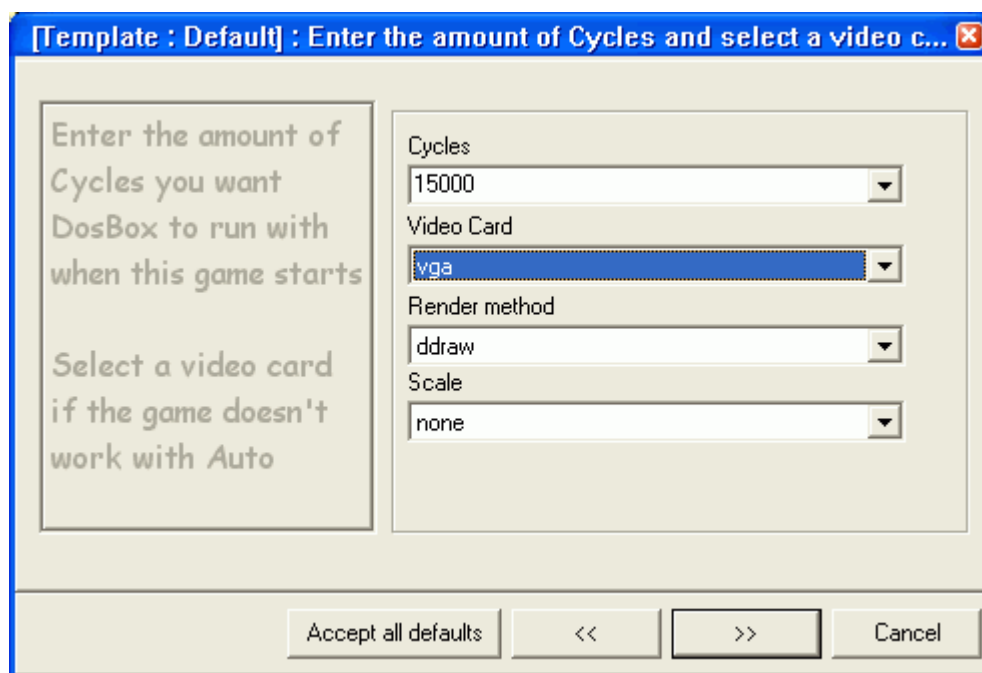
Nos encontraremos con un cuadro de diálogo como éste:



Esto no merecería mayores comentarios, pues las opciones se explican por sí mismas. En "Profile Name" escribiremos el nombre que tendrá el perfil que vamos a crear. "Game EXE" es precisamente eso; el ejecutable del juego o programa que queremos emular en DOSBox. "Setup EXE", por su parte, se refiere a la utilidad de configuración del juego; si no hay ninguna, lo dejaremos en blanco.

Hemos marcado la casilla "Close Dosbox after game exit". De este modo, cuando terminemos nuestra partida del Rise of the Triad y salgamos del juego, el DOSBox se cerrará también.

Una vez rellenado esto, pasamos a la siguiente ventana:

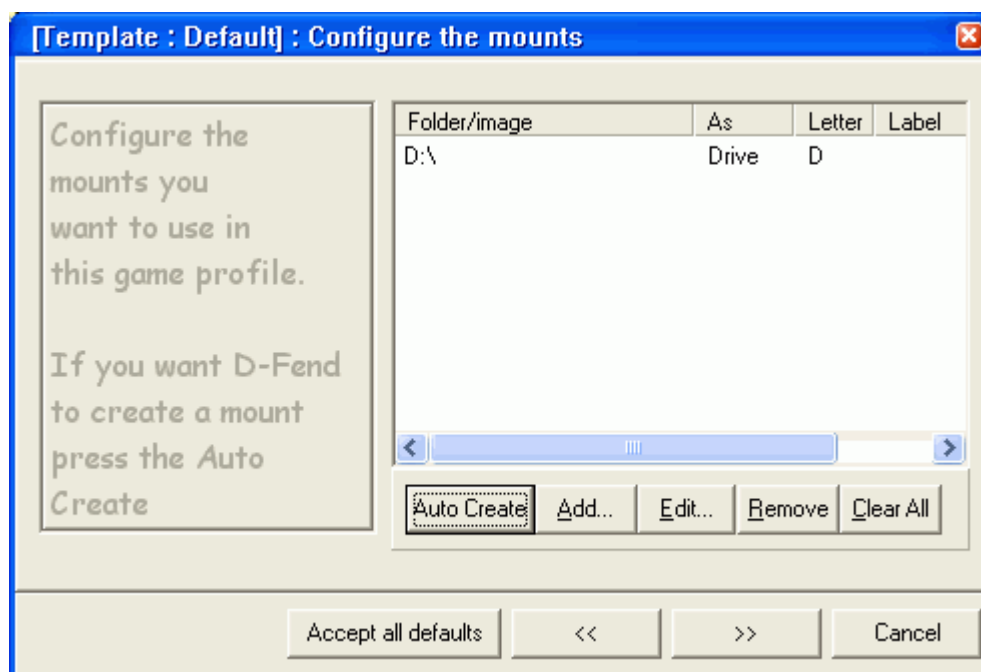


Aquí elegiremos la velocidad de la emulación (los ciclos; ver el capítulo "Acelerando la Emulación"), el tipo de tarjeta gráfica que emularemos (VGA, CGA, Tandy, Hercules) y el modo de renderización gráfica que usará DOSBox (*surface, opengl, ddraw...* Ver "El Archivo *dosbox.conf*").

En principio, para la gran mayoría de juegos y programas que ejecutemos en DOSBox deberíamos usar VGA como "Video Card". Ahora, si queremos usar alguna aplicación diseñada para funcionar con Tandy o Hercules -que son muy pocas; casi todas admitían además el modo gráfico CGA, que las tarjetas VGA podían reproducir sin problemas-, tendremos que seleccionar el valor correspondiente.

Bien, le damos a "siguiente", y aparecerá en pantalla el tercer cuadro de diálogo. Aquí tendremos que configurar las unidades que resultará necesario montar para ejecutar el juego. De todas formas, normalmente no hace falta calentarse mucho la cabeza; hay un botón llamado "Auto Create" que realiza todo el trabajo por nosotros, partiendo de los datos que hemos rellenado en el primer paso.

Aquí podemos verlo:



A continuación nos las veremos con la cuarta ventana, cuya función es la de configurar el funcionamiento de la tarjeta emulada SoundBlaster. Los valores que vienen por defecto deberían servirnos perfectamente, y no vale la pena que nos preocupemos por ellos. En todo caso, podríamos cambiar el primer valor -"Type"- si en lugar de una SoundBlaster 16 queremos emular otro modelo.

El quinto cuadro de diálogo está dedicado a la configuración de la Gravis Ultrasound; el sexto a la emulación MIDI, y el séptimo a la del altavoz interno del PC. Ninguno de ellos presenta mayores complicaciones, y podemos dejar tranquilamente sus valores por defecto.

En todos estos dispositivos de sonido deberíamos tener en cuenta sus "rates" -que por defecto están configurados en 22050 Hz-. Este valor no debería ser mayor que el que pusimos de "Sample rate" al configurar D-Fend (menú "configure defaults", pestaña "Sound"). De lo contrario, es posible que el sonido emulado no funcione correctamente, u obtengamos una calidad bastante mala.

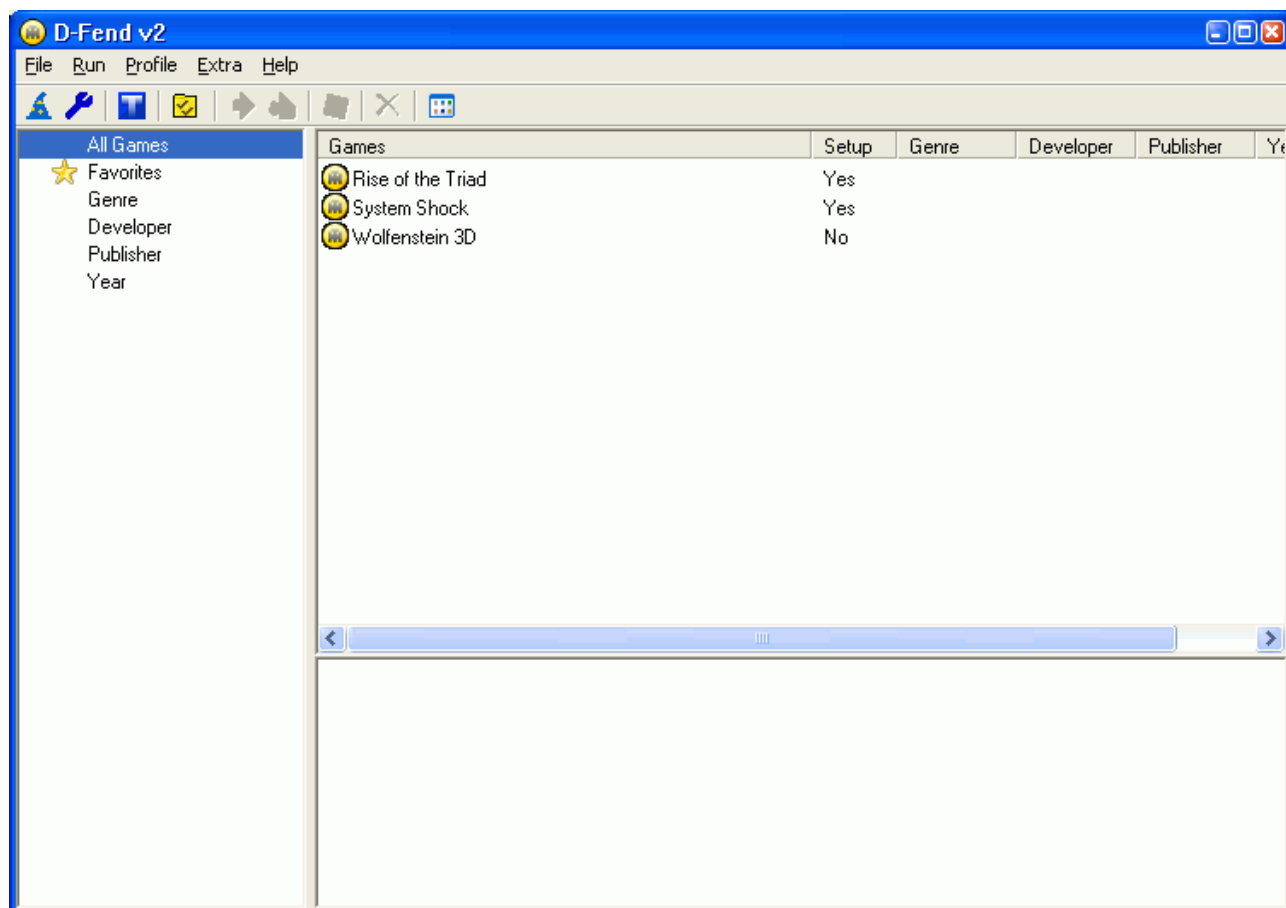
Y ya estamos terminando.

En el octavo -y último- paso tendremos que especificar cuánta memoria RAM emulará DOSBox, y si habilitaremos memoria extendida (XMS) y expandida (EMS).

Comentaré que a la hora de rellenar estos cuadros de diálogo existe un botón llamado "Accept all defaults". Lo podemos usar si, en lugar de ajustar opciones propias para un perfil, deseamos utilizar la configuración por defecto de D-Fend (que editamos después de instalarlo, en los "configure defaults").

Así que ya tenemos nuestro perfil acabado y listo para usar. Lo veremos en la ventana principal de D-Fend... Y de ahora en adelante, para jugar al Rise of the Triad emulándolo con DOSBox, nos bastaría con un doble clic de ratón.

Por supuesto, podemos editar los perfiles ya creados si queremos cambiar alguna opción que se nos haya pasado por alto, etc.



Bien, y con esto podríamos dar por terminada esta pequeña sección sobre D-Fend; apenas quedarían por comentar algunos detalles poco importantes. Una vez sepamos cómo crear un perfil -cosa muy fácil si conocemos el funcionamiento de DOSBox-, el programa dejará de tener secretos para nosotros.

Supongo que el uso de otros *front-ends* -Boxer, etc-, no cambiará demasiado respecto a lo que he explicado aquí. Yo me he centrado en D-Fend porque es de los más utilizados, y el único que he visto funcionar.

## NOTAS SOBRE ESTE TEXTO

Y bueno, hasta aquí el tutorial de DOSBox...

No he pretendido hacer una guía exhaustiva, entre otras cosas porque me faltan conocimientos para ello. Mi idea original era, más bien, redactar un pequeño manual de uso, con las nociones básicas para defenderse con el emulador y solucionar los problemas más comunes. Al final, no obstante, me he extendido más de lo que pensaba, y he comentado aspectos que no entraban en mi plan original.

Las fuentes de este tutorial son variadas. En primer lugar, y lo más importante, mi propia experiencia. Ahora, también me he basado en la documentación oficial de DOSBox (el archivo *README*), del que he tomado bastante información. Sobre algunos detalles me he informado en los foros de la web oficial de DOSBox (<http://dosbox.sourceforge.net>), y en un par de pequeñas guías que encontré en Internet.

Por lo demás, podéis modificar, distribuir o colgar en la web este texto a vuestro antojo y sin ningún permiso por mi parte. Si véis algún detalle que debe ser corregido, o creéis necesario ampliar el manual con alguna sección nueva, podéis hacerlo tranquilamente. En tales casos me gustaría que me enviaseis las copias modificadas por e-mail ([eyedecul@gmail.com](mailto:eyedecul@gmail.com)), más que nada por tener una versión actual del texto; pero bueno, tampoco es algo imprescindible.

En fin, esto vendría a ser como una licencia GPL... ☺

Y nada más.

Espero que el manual os resulte útil, y os ayude a reencontraros con muchas viejas glorias del videojuego. Al menos, con esa intención está redactado.

*Un saludo,  
Eye del Cul.*

A 23 de agosto 2005; texto escrito con OpenOffice...