

PDIH - UGR

Seminario 1 - Ángel Gómez Ferrer

Setup básico:

Primero vamos a la página principal de DOSBox y lo descargamos:



Click here to download **DOSBox 0.74-3** for your Operating System, or to support us with a donation:

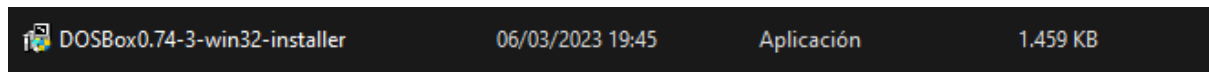
[Download Now](#) [Make A Donation](#) [DOSBox Tutorial](#) [DOSBox Manual](#)

Looking for a different version, a frontend or a translation? Here's a full list of files:

DOSBox (specific versions)		
Mac OS X	0.74-3-3	dmg (Universal)
Windows	0.74-3	Win32 installer
Fedora	0.74-3	rpm
Source	0.74-3	Source
Debian	0.74-3	deb
FreeBSD package	0.74-2	ports
RISC OS	0.74-2	zip
Gentoo Linux	0.74	portage
OS/2	0.74	exe (OS2)
Solaris 10 - sparc	0.73	pkg
BeOS	0.63	binary (x86)
Old dosbox versions	0.50-0.74-2	source + binary(x86)

Frontends		
Dosboxfront	4.1	Frontend for OS/2
DosBlaster	3.0	Frontend in .NET
Dapplegrey	2.3	Frontend for Mac OS X(Universal)
ProGammaX	2.2	Frontend in .Net
D.O.G	1.53	Frontend for windows
DOSShell	1.4	Frontend for windows

Corremos el ejecutable:



Tras esto tendremos instalado DOSBox.

Ejercicio 1:

Ejecutamos uno de los juegos proporcionados por el profesor, en mi caso el tetris, para ello he ubicado la carpeta JUEGOS en el directorio montado y accedido a ella a través de la consola:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

To adjust the emulated CPU speed, use ctrl-F11 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 http://www.dosbox.com

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

Z:\>mount C /Users/angel/DOSBOX
Drive C is mounted as local directory /Users/angel/DOSBOX\

Z:\>keyb sp
Keyboard layout sp loaded for codepage 437

Z:\>path c:\bc\bin

Z:\>c:

C:\>cd JUEGOS

C:\JUEGOS>cd TETRIS

C:\JUEGOS\TETRIS>tetris
```

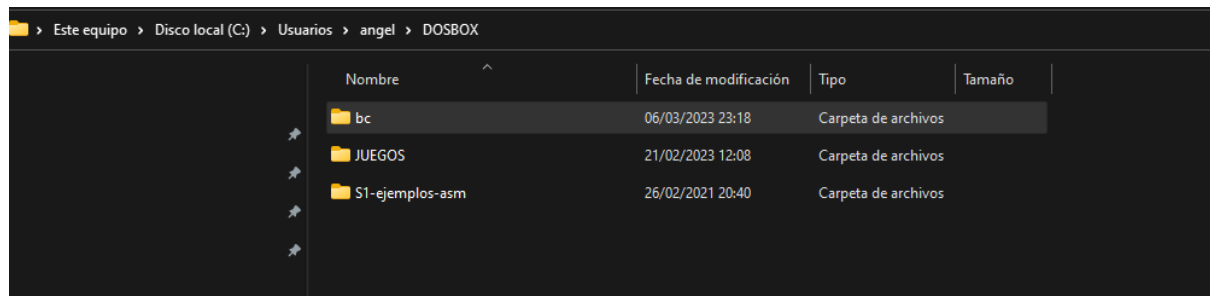
Y lo ejecutamos:



Ejercicio 2:

(Aclarar que estos pasos los hice antes que el ejercicio 1)

Para configurar simplemente tras haber creado los directorios adecuados e importado bc:



Incluimos las siguientes líneas en mi caso en la ubicación:

C:\Users\angel\AppData\Local\DOSBox

declararemos en path donde estará el ejecutable BC.EXE

```
240
241 [ipx]
242 # ipx: Enable ipx over UDP/IP emulation.
243
244 ipx=false
245
246 [autoexec]
247 # Lines in this section will be run at startup.
248 # You can put your MOUNT lines here.
249 mount C /Users/angel/DOSBOX
250 keyb sp
251 path c:\bc\bin
252
253 c:
254
255
256
```

Para la configuración de la pantalla y el motor gráfico opengl:

```
24 #      usescancodes: Avoid usage of symkeys
25
26 fullscreen=false
27 fulldouble=false
28 fullresolution=original
29 windowresolution=1024x768
30 output=opengl
31 autolock=true
32 sensitivity=100
33 waitonerror=true
34 priority=higher,normal
35 mapperfile=mapper-0.74-3.map
36 usescancodes=true
37
```

Ejercicio 3:

Creemos el archivo HOLA.ASM con el código ensamblador necesario:

```
HOLA.ASM
1  pila segment stack 'stack'
2      dw 100h dup (?)
3  pila ends
4  datos segment 'data'
5      msg db 'Hola Mundo$'
6  datos ends
7  codigo segment 'code'
8      assume cs:codigo, ds:datos, ss:pila
9  main PROC
10     mov ax,datos
11     mov ds,ax
12     mov dx,OFFSET msg
13     mov ah,9
14     int 21h
15     mov ax,4C00h
16     int 21h
17 main ENDP
18 codigo ends
19 END main
20
```

Lo incluimos en el directorio previamente montado para DOSBOX y podremos visualizarlo:

```
C:\>type hola.asm
pila segment stack 'stack'
dw 100h dup (?)
pila ends
datos segment 'data'
msg db 'Hola Mundo$'
datos ends
codigo segment 'code'
assume cs:codigo, ds:datos, ss:pila
main PROC
mov ax,datos
mov ds,ax
mov dx,OFFSET msg
mov ah,9
int 21h
mov ax,4C00h
int 21h
main ENDP
codigo ends
END main
C:\>
```

Creamos el ejecutable utilizando el script c.bat:

```
C:\>C.BAT hola
Turbo Assembler  Version 3.1  Copyright (c) 1988, 1992 Borland International

Assembling file:    hola.asm
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   472k

Turbo Link  Version 5.1 Copyright (c) 1992 Borland International
C:\>dir
Directory of C:\.
.                <DIR>                07-03-2023   0:30
..               <DIR>                06-03-2023  22:34
BC               <DIR>                06-03-2023  23:18
JUEGOS           <DIR>                21-02-2023  12:08
S1-EJE~1         <DIR>                26-02-2021  20:40
HOLA             ASM                  299 07-03-2023   0:26
HOLA             EXE                  1,519 07-03-2023   0:30
HOLA             MAP                   232 07-03-2023   0:30
HOLA             OBJ                   423 07-03-2023   0:30
    4 File(s)                2,473 Bytes.
    5 Dir(s)                 262,111,744 Bytes free.

C:\>
```

y probamos a ejecutar el ejecutable generado:

```
C:\>hola.exe
Hola Mundo
C:\>_
```

Actualizamos el código para que ejecute en bucle 7 veces: el Hola Mundo

```
1  pila segment stack 'stack'
2  dw 100h dup (?)
3  pila ends
4  datos segment 'data'
5  msg db 'Hola Mundo $'
6  datos ends
7  codigo segment 'code'
8  assume cs:codigo, ds:datos, ss:pila
9  main PROC
10     mov ax,datos
11     mov ds,ax
12     mov cx,1
13     ; inicializamos contador
14
15     bucle:
16     mov dx,OFFSET msg
17     mov ah,9
18     int 21h
19     ; incrementamos contador
20     inc cx
21     cmp cx,7
22     ; comparamos el contador con 7 (7 veces)
23     jne bucle
24     ; repetimos (not equal)
25
26     mov ah,9
27     int 21h
28     mov ax,4C00h ; terminar y salir al S.O.
29     int 21h
30     main ENDP
31     codigo ends
32
33     END main
34
```

Ejecutamos el nuevo ejecutable con el bucle:

```
C:\>C.BAT hola
Turbo Assembler  Version 3.1  Copyright (c) 1988, 1992 Borland International

Assembling file:   hola.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  472k

Turbo Link  Version 5.1 Copyright (c) 1992 Borland International
C:\>hola.exe
Hola Mundo Hola Mundo Hola Mundo Hola Mundo Hola Mundo Hola Mundo Hola Mundo
C:\>_
```