

INSTALACIÓN DE DOSBOX

En mi caso he realizado la instalación en Linux, la instalación es bastante sencilla. Lo primero es descargar el código fuente de la página principal <https://www.dosbox.com/>. Una vez tengamos la carpeta descomprimida ejecutamos `./configure` para preparar la instalación y `make` para acabar de instalar Dosbox, debería quedar instalado en el directorio `/usr/local/bin`

```
hermeit@hermeit-VirtualBox:~/Downloads$ tar xzf dosbox-0.74-3.tar.gz
hermeit@hermeit-VirtualBox:~/Downloads$ ls
dosbox-0.74-3  dosbox-0.74-3.tar.gz  instalador-DOSBOX.zip
hermeit@hermeit-VirtualBox:~/Downloads$ ./configure
bash: ./configure: No such file or directory
hermeit@hermeit-VirtualBox:~/Downloads$ cd dosbox-0.74-3/
hermeit@hermeit-VirtualBox:~/Downloads/dosbox-0.74-3$ ./configure
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... no
checking whether make supports nested variables... no
checking whether make sets $(MAKE)... (cached) no
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: in `/home/hermeit/Downloads/dosbox-0.74-3':
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details
hermeit@hermeit-VirtualBox:~/Downloads/dosbox-0.74-3$ make
```

```
hermeit@hermeit-VirtualBox:/usr/local/bin$ ls
dosbox
hermeit@hermeit-VirtualBox:/usr/local/bin$
```

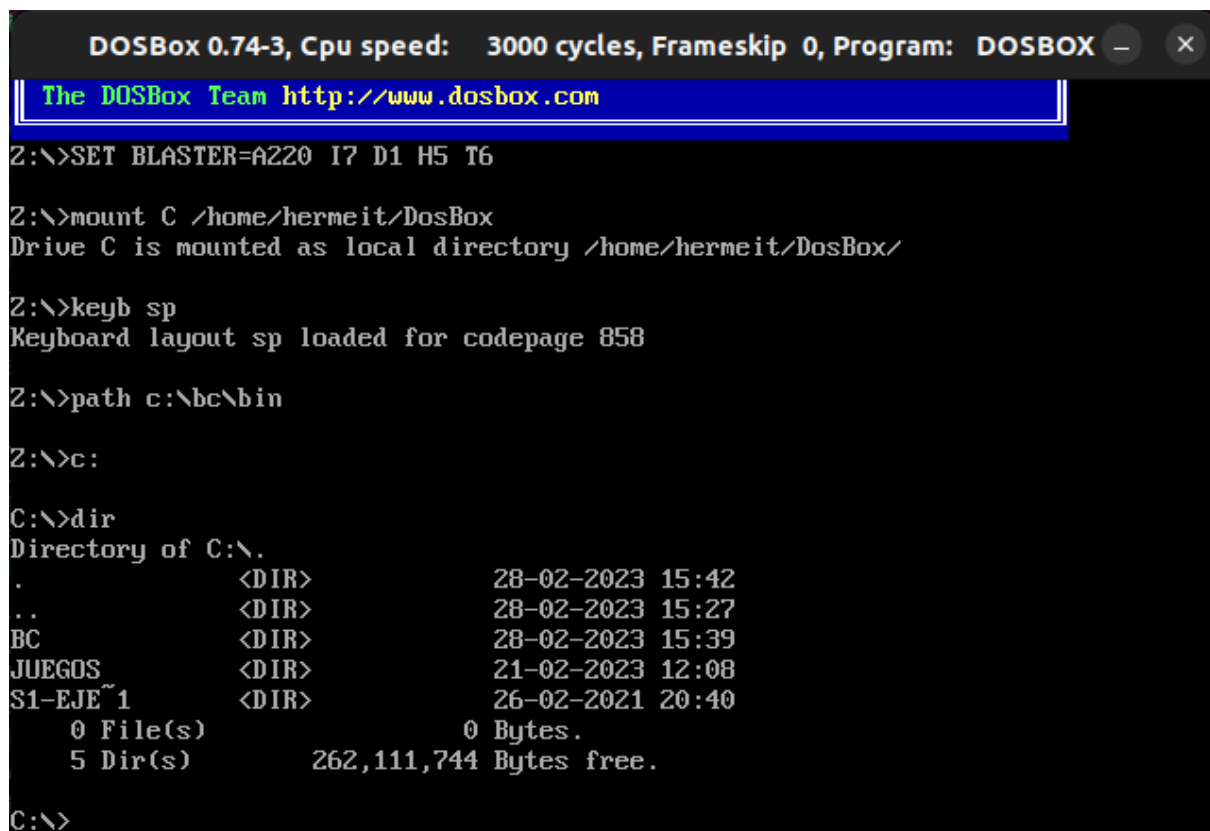
Una vez instalado solo tenemos que realizar una pequeña configuración antes de iniciarlo, para ello editaremos el archivo `dosbox-0.74-3.conf` y añadiremos al final del archivo las siguientes líneas:

```
[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.

mount C /home/hermeit/DosBox
keyb sp
path c:\bc\bin

c:
```

Lo que hacen estas líneas es montar la ruta donde tendremos nuestros archivos del Dosbox, de esta forma se montará automáticamente al iniciar la aplicación. La línea keyb sp lo que hace es cambiar la distribución del teclado al español, para así trabajar con mayor comodidad. Al ejecutar nuestro dosbox debería quedar así



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount C /home/hermeit/DosBox
Drive C is mounted as local directory /home/hermeit/DosBox/

Z:\>keyb sp
Keyboard layout sp loaded for codepage 858

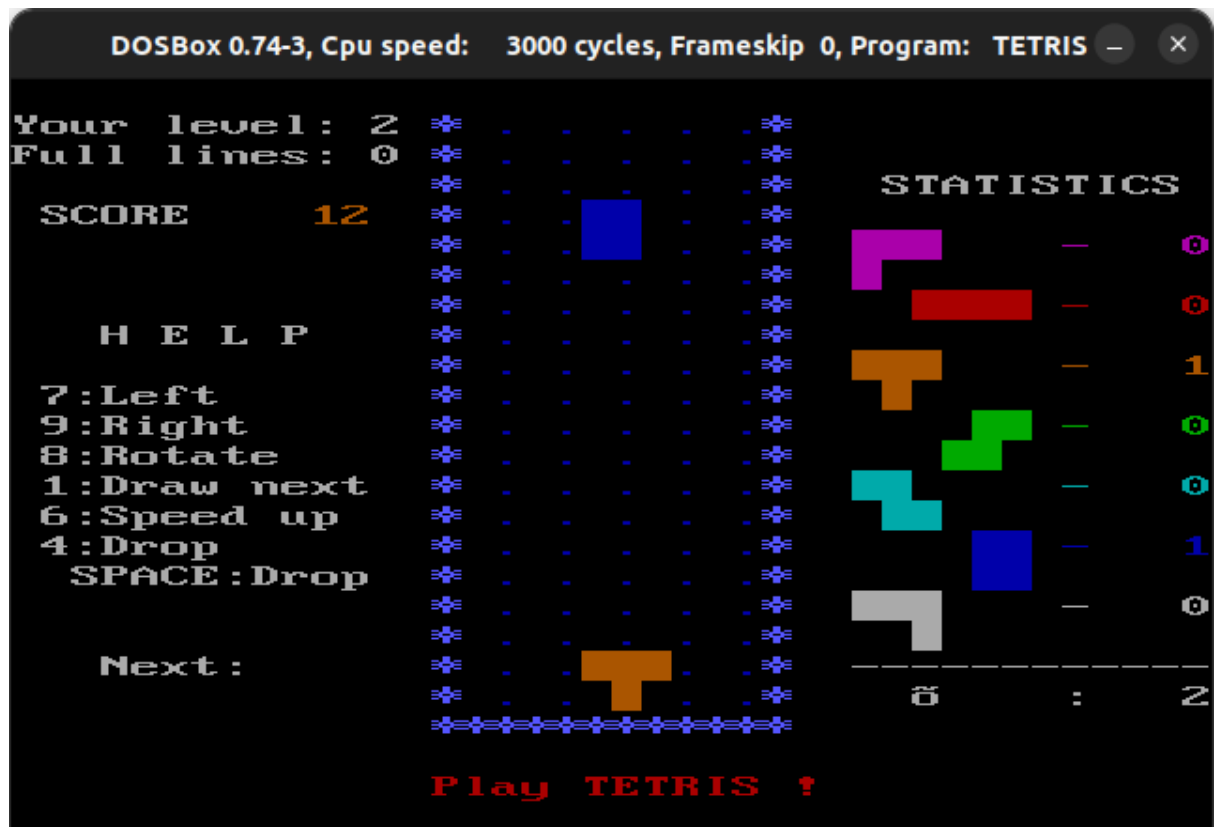
Z:\>path c:\bc\bin

Z:\>c:

C:\>dir
Directory of C:\.
.                <DIR>                28-02-2023  15:42
..               <DIR>                28-02-2023  15:27
BC               <DIR>                28-02-2023  15:39
JUEGOS          <DIR>                21-02-2023  12:08
S1-EJE~1        <DIR>                26-02-2021  20:40
    0 File(s)                0 Bytes.
    5 Dir(s)                262,111,744 Bytes free.

C:\>
```

Para jugar a un juego solo tenemos que movernos a su carpeta correspondiente y ejecutarlo, yo por ejemplo he ejecutado el tetris



EJECUCIÓN DE CÓDIGO ENSAMBLADOR EN MSDOS

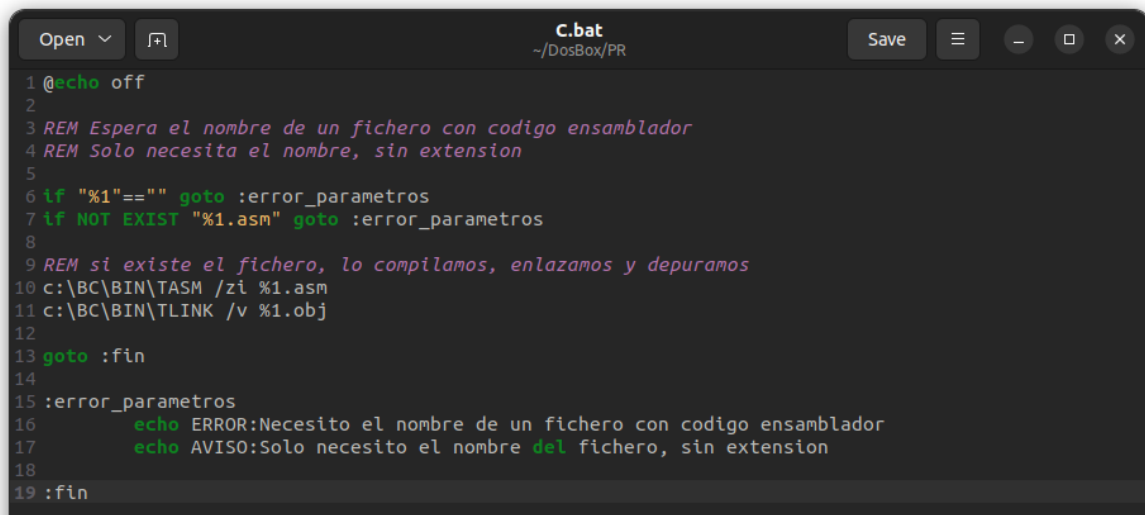
Vamos a crear un programa simple en ensamblador que muestre la cadena de texto “hola” para ello vamos a crear un archivo con el código en ensamblador, lo llamaremos hola.asm

```

Open  hola.asm  Save  -  +  x
~/DosBox
1 pila segment stack 'stack'
2     dw 100h dup (?)
3 pila ends
4
5 datos segment 'data'
6     msg db 'hola$'
7 datos ends
8
9 codigo segment 'code'
10    assume cs:codigo, ds:datos, ss:pila
11    main PROC
12        mov ax,datos
13        mov ds,ax
14
15        mov dx,OFFSET msg
16        mov ah,9
17        int 21h
18
19        mov ax,4C00h
20        int 21h
21
22    main ENDP
23 codigo ends
24
25 END main

```

Para futuras compilaciones, crearemos un script para compilar el código con las siguientes líneas, de esta forma solo tendremos que pasarle el nombre del archivo que queremos compilar



```
1 @echo off
2
3 REM Espera el nombre de un fichero con codigo ensamblador
4 REM Solo necesita el nombre, sin extension
5
6 if "%1"==" " goto :error_parametros
7 if NOT EXIST "%1.asm" goto :error_parametros
8
9 REM si existe el fichero, lo compilamos, enlazamos y depuramos
10 c:\BC\BIN\TASM /zi %1.asm
11 c:\BC\BIN\TLINK /v %1.obj
12
13 goto :fin
14
15 :error_parametros
16     echo ERROR:Necesito el nombre de un fichero con codigo ensamblador
17     echo AVISO:Solo necesito el nombre del fichero, sin extension
18
19 :fin
```

Ahora solo tendremos que ejecutar el archivo .exe creado mediante este script, de forma que nuestro código se ejecutará así

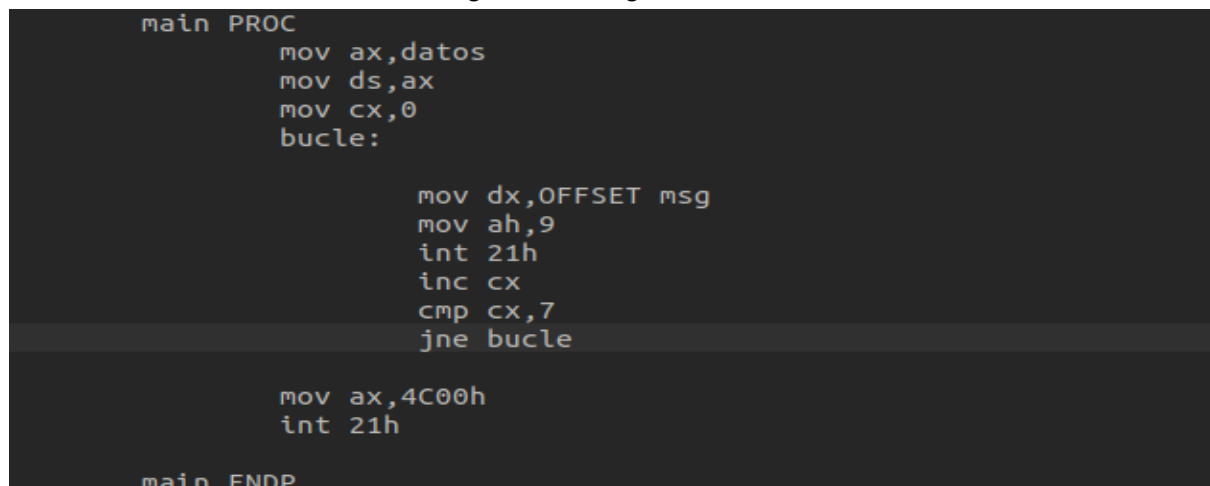


```
C:\PR>C.BAT hola
Turbo Assembler Version 3.1 Copyright (c) 1988, 1992 Borland International

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:\PR>HOLA.EXE
hola
C:\PR>
```

Vamos ahora a modificar el código para incluir un bucle que muestre el mensaje un número de 7 veces, la modificación del código será la siguiente



```
main PROC
    mov ax,datos
    mov ds,ax
    mov cx,0
    bucle:

        mov dx,OFFSET msg
        mov ah,9
        int 21h
        inc cx
        cmp cx,7
        jne bucle

    mov ax,4C00h
    int 21h

main ENDP
```

El resultado de la ejecución es el siguiente:

```
(C:\PR>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:\PR>HOLA.EXE
hola hola hola hola hola hola hola
C:\PR>
```