

SEMINARIO 1. BAJO NIVEL

Alberto Socas Mendoza – Periféricos y dispositivos de Interfaz Humana

1. Instalar el software de DOSBox y ejecutar aplicaciones de MS-DOS

Como observamos en la primera imagen [Ilustración 1] hemos instalado DOSBox correctamente.

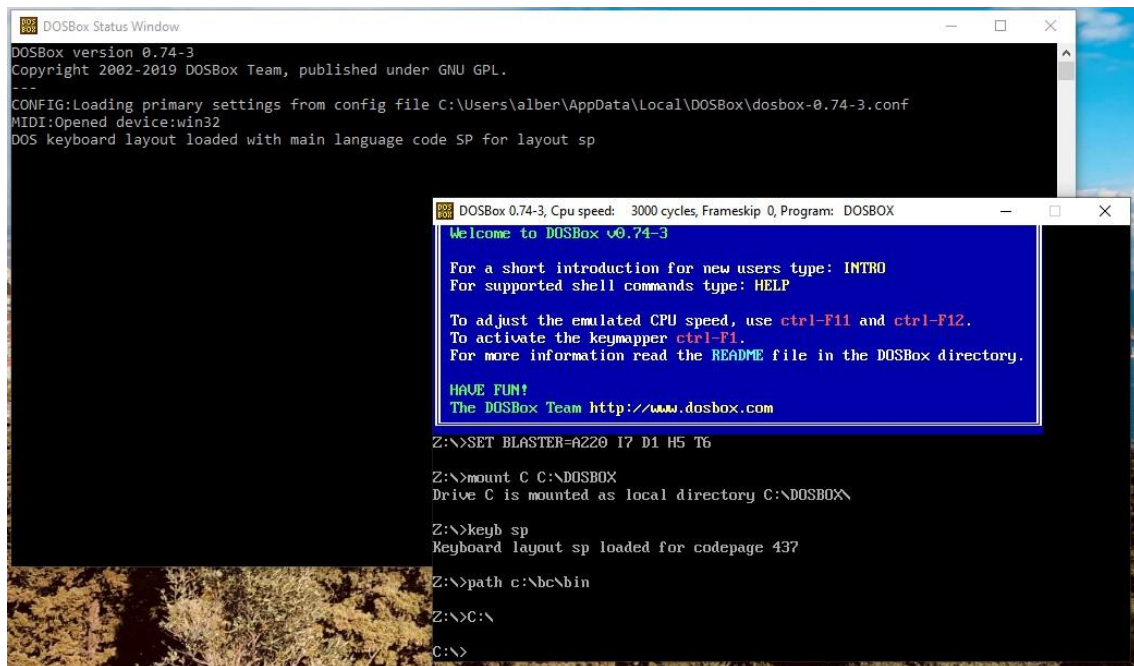


Ilustración 1

El siguiente paso es ejecutar un juego, por ejemplo, ejecutaremos VBALL [Ilustración 2]



Ilustración 2

El resultado es el siguiente [Ilustración 3]. En el que podremos jugar al Arcade Volleyball.



Ilustración 3

2. Configurar el inicio de DOSBox para que monte en la unidad C: el directorio donde se encuentra el entorno de programación Borland C. Añadir a la variable “PATH” de inicio el directorio “bin” donde se encuentra el ejecutable BC.EXE.

Como nos pide el ejercicio, configuramos el inicio de DOSBOX, montamos en C: el directorio C:\DOSBOX, que es donde se encuentra el entorno de programación Borland C [Ilustración 4]. Además, añadimos el teclado en español [keyb sp] y la variable PATH con su correspondiente directorio [Ilustración 5].

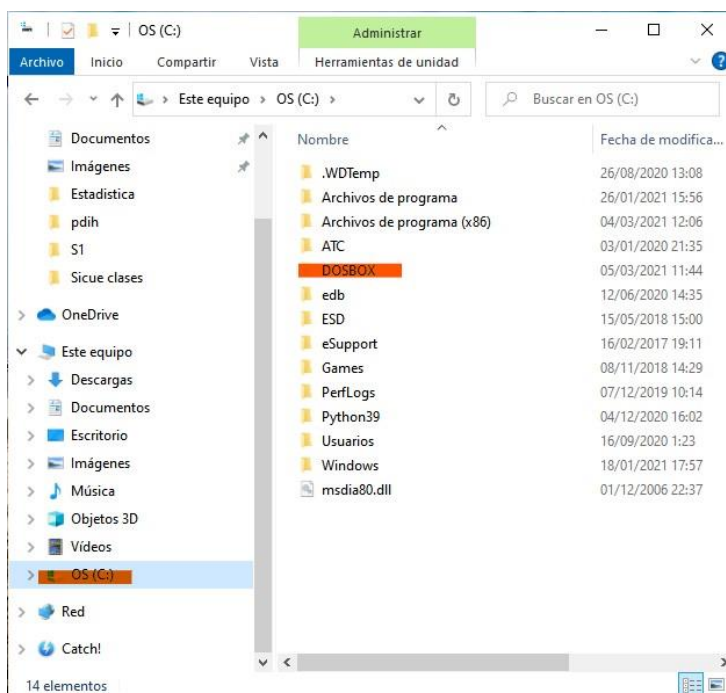


Ilustración 4

```

245
246 [autoexec]
247 # Lines in this section will be run at startup.
248 # You can put your MOUNT lines here.
249
250 mount C C:\DOSBOX
251 keyb sp
252
253 path c:\bc\bin
254
255 C:\

```

Ilustración 5

3. Crear el ejemplo “Hola mundo” en ensamblador, compilarlo y comprobar su funcionamiento. A continuación, incluir un bucle para mostrar el mensaje 7 veces.

Creemos el código que usaremos en este caso hola.asm, en el que el programa ejecutará en pantalla “Hola mundo” [Ilustración 6].



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program
C:\S1-EJE~1>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

```

Ilustración 6

A continuación, compilamos y enlazamos con los programas “tasm.exe” y “tlink.exe”, por simplicidad hemos creado un script “c.bat” que recibe el nombre del programa a compilar [Ilustración 7].

```

C:\S1-EJE~1>type c.bat
@echo off

REM Espera el nombre de un fichero con codigo ensamblador
REM Solo necesita el nombre, sin extension

if "%1"=="" goto :error_parametros
if NOT EXIST "%1.asm" goto :error_parametros

REM si existe el fichero, lo compilamos, enlazamos y depuramos:
c:\BC\BIN\TASM /zi %1.asm
c:\BC\BIN\TLINK /v %1.obj

goto :fin

:error_parametros
    echo ERROR: Necesito el nombre de un fichero con codigo ensamblador
    echo AVISO: Solo necesito el nombre del fichero, sin extension.

:fin

```

Ilustración 7

Le pasamos el nombre del fichero como nos pide, en este caso “c.bat hola” y se nos crea un nuevo archivo .exe, que ejecutaremos con el código [Ilustración 8]. Y finalmente tenemos nuestro programa ejecutado.

```

C:\S1-EJE~1>hola.exe
hola mundo

```

Ilustración 8

Para continuar con la segunda parte de este ejercicio debemos crear un bucle para que nuestro resultado se repita 7 veces. Añadimos a hola.asm ese pequeño código [Ilustración 9].

```

mov cx, 0
bucle:
    mov dx, OFFSET msg
    mov ah, 9
    int 21h

    inc cx
    cmp cx, 6
    jne bucle

```

Ilustración 9

Volvemos a nuestra máquina DOSBox, y ejecutamos de nuevo “type hola.asm” como nos muestra la Ilustración 6, pero en este caso nuestro código será distinto. Seguidamente compilamos y enlazamos con “type c.bat” [Ilustración 7]. Y le damos el nombre del fichero

como nos pide “c.bat hola” [Ilustración 10]. Se nos crea un fichero .exe, que ejecutaremos con la orden [Ilustración 11] y finalmente nos muestra el resultado esperado.

```
C:\S1-EJE~1>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
```

Ilustración 10

```
C:\S1-EJE~1>hola.exe
hola mundohola mundohola mundohola mundohola mundohola mundohola mundo
```

Ilustración 11