

Periféricos y Dispositivos de Interfaz Humana



**UNIVERSIDAD
DE GRANADA**

CURSO 2024 - 2025

Seminario 2. Acceso a dispositivos de E/S en C

**GABRIEL VICO ARBOLEDAS
RAÚL RODRÍGUEZ RODRÍGUEZ**

Índice

1. Configurar el inicio de DOSBox.....	3
2. Programa en C que use la función de cambio de modo de vídeo.....	4
3. Programa en C que use la función de lectura de caracteres.....	5

1. Configurar el inicio de DOSBox

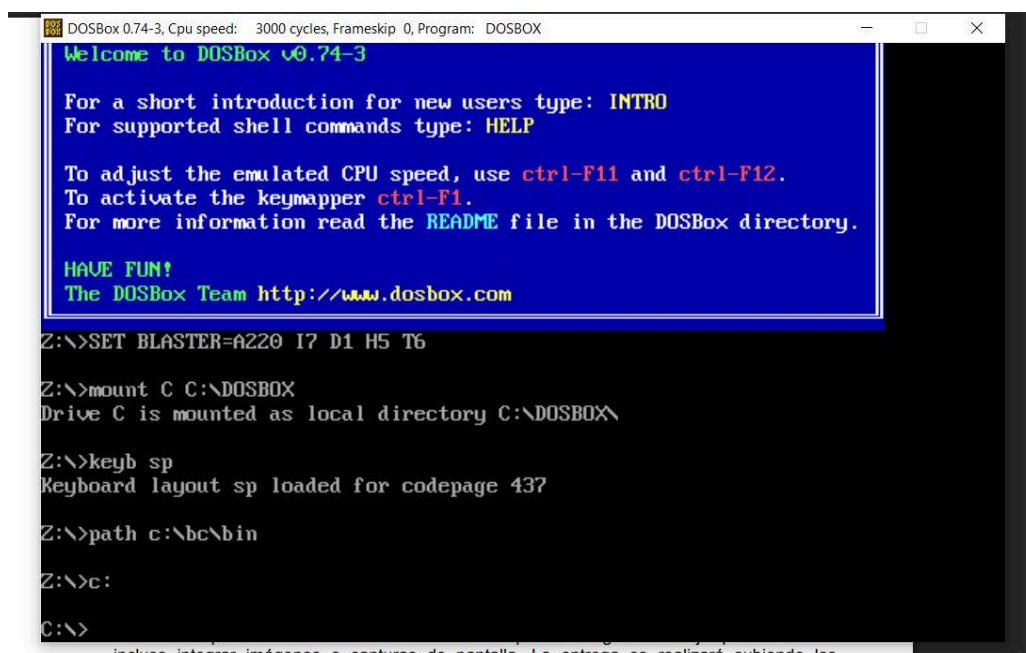
```
[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.
```

```
mount C C:\DOSBOX
keyb sp
path c:\bc\bin
```

```
C:
```

```
[sdl]
# fullscreen: Start dosbox directly in fullscreen. (Press ALT-Enter to go back)
# fulldouble: Use double buffering in fullscreen. It can reduce screen flickering, but it can also result in a slow DOSBox.
# fullresolution: What resolution to use for fullscreen: original, desktop or fixed size (e.g. 1024x768).
#                 Using your monitor's native resolution (desktop) with aspect=true might give the best results.
#                 If you end up with small window on a large screen, try an output different from surface.
#                 On Windows 10 with display scaling (Scale and layout) set to a value above 100%, it is recommended
#                 to use a lower full/windowresolution, in order to avoid window size problems.
# windowresolution: Scale the window to this size IF the output device supports hardware scaling.
#                   (output=surface does not!)
# output: What video system to use for output.
#          Possible values: surface, overlay, opengl, openglfb, ddraw.
# autolock: Mouse will automatically lock, if you click on the screen. (Press CTRL-F10 to unlock)
# sensitivity: Mouse sensitivity.
# waitonerror: Wait before closing the console if dosbox has an error.
# priority: Priority levels for dosbox. Second entry behind the comma is for when dosbox is not focused/minimized.
#           pause is only valid for the second entry.
#           Possible values: lowest, lower, normal, higher, highest, pause.
# mapperfile: File used to load/save the key/event mappings from. Resetmapper only works with the default value.
# usescancodes: Avoid usage of symkeys, might not work on all operating systems.

fullscreen=false
fulldouble=false
fullresolution=original
windowresolution=1024x768
output=opengl
autolock=true
sensitivity=100
waitonerror=true
priority=higher,normal
mapperfile=mapper-0.74-3.map
usescancodes=true
```



incluso integrar imágenes o capturas de pantalla. La entrega se realizará subiendo los

2. Programa en C que use la función de cambio de modo de vídeo

```
C ej2.c 9+ x
C: > DOSBOX > S2ej > C ej2.c > main()
1  #include <dos.h>
2  #define BYTE unsigned char
3
4  void video_mode (BYTE modo){
5      union REGS inregs, outregs;
6      inregs.h.ah = 0x00;
7      inregs.h.al = modo;
8      int86(0x10,&inregs, &outregs);
9      return;
10 }
11
12 void pause(){
13     union REGS inregs, outregs;
14     inregs.h.ah = 1;
15     int86(0x21, &inregs, &outregs);
16 }
17
18 int main(){
19     video_mode(3); // Texto 80x25-16Cotones
20     printf("\nPulsa una tecla: ");
21
22     pause();
23
24     video_mode(4); // Grafico 320x200 4Colores
25     printf("\nPulsa una tecla para finalizar");
26
27     pause();
28     video_mode(3); // Texto 80x25-16Cotones
29     return 0;
30 }
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EJ2

Pulsa una tecla:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EJ2

Pulsa una tecla para finalizar

3. Programa en C que use la función de lectura de caracteres

```
C ej2.c 9+ C ej3.c 5 X
C: > DOSBOX > S2ej > C ej3.c > main()
1  #include <dos.h>
2  #include <stdio.h>
3
4  #define BYTE unsigned char
5
6  char read_char();
7  void write_char(char c);
8
9  char read_char() {
10     union REGS inregs, outregs;
11     inregs.h.ah = 1;
12     int86(0x21, &inregs, &outregs);
13     return outregs.h.al;
14 }
15 void write_char(char c) {
16     union REGS inregs, outregs;
17     inregs.h.ah = 2;
18     inregs.h.dl = c;
19     int86(0x21, &inregs, &outregs);
20 }
21
22 int main() {
23     char aux;
24     printf("\n Introducir un caracter: ");
25     aux = read_char();
26
27     printf("\n Caracter introducido: ");
28     write_char(aux);
29
30     return 0;
31 }
32
33
```

```
C:\S2EJ>ej3.exe

Introducir un caracter: q
Caracter introducido: q
C:\S2EJ>ej3.exe

Introducir un caracter: w
Caracter introducido: w
C:\S2EJ>
```