Ministry of Education, Culture and Research of the Republic of Moldova
Technical University of Moldova
Department of Software and Automation Engineering

REPORT

Laboratory work no. 1
Printing Methods

Elaborated:
st. gr. FAF-213 Botnari Ciprian

Verified:
lect. univ. Rostislav Călin

Chişinău – 2023

Table of Contents

Topic: Printing Methods

Tasks

Create a program in assembler which will print text to the screen. It should respect the following conditions:

1. ALL possible methods should be used in order to print text.
    a. M1: Write character as TTY
    b. M2: Write character
    c. M3: Write character/attribute
    d. M4: Display character + attribute
    e. M5: Display character + attribute & update cursor
    f. M6: Display string
    g. M7: Display string & update cursor
    h. M8(optional): Print directly to video memory
2. Compiled program should be used in order to create a floppy image and it should be bootable. Use this image to boot the OS in a VirtualBox VM and the text which you intended to print should appear on the screen.
3. You can use any assembly compiler.
4. You should be able to modify the code, to recompile it and to boot the VM with new version of programs
5. In order to use documentation from TechHelp/XView DOS application, you can install DosBox.

Code

I made a build.sh script that does the following:

1. Compile the assembly code to a .com file
2. Copy the .com file to a .flp file
3. Resize the .flp file to 1.44MB (floppy disk)

Besides this, I created a build_all.sh to compile all the asm files at the same time.

Some important instructions:

- mov AH, 0Eh ; Teletype output(TTY)
- mov AH, 0Ah ; Function 0Ah - Write character
- mov AH, 09h ; Function 09h = Write Character and Attribute at Cursor Position
- mov AH, 02h ; Function 02h = Set Cursor Position
- mov AX, 1300h ; Function 13h, Write String

Results

Below are 2 out of 7 examples: 1 is for char and another one is for string.

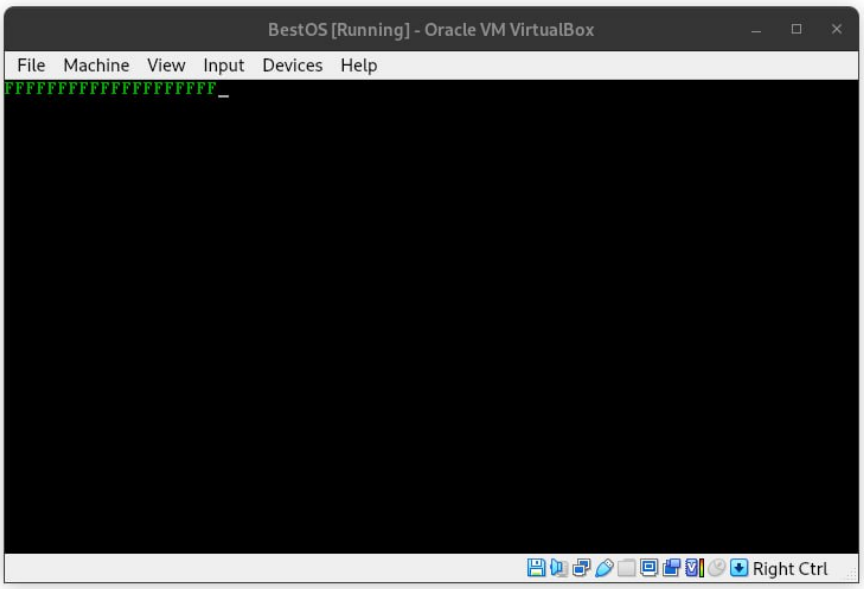

Figure 1. *Char + attribute, update cursor*

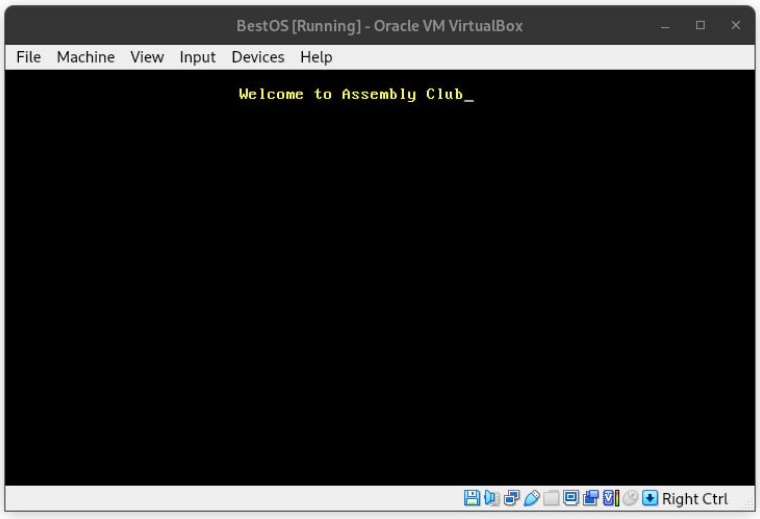

Figure 2. *String + attribute, update cursor*

Conclusion

In the end, this lab work was quite a learning experience for me in Assembly. I had to figure out seven different ways to show text on the screen, from the basic stuff to more complicated things like moving the cursor around and adding attributes to the text. The coolest part was when I got to compile my code into a floppy image and see it work in a Virtual Box – it was like magic when the text showed up on the screen.
Build scripts, like "build.sh" and "build_all.sh," made it a lot easier to test and change my code. It's nice to know I can always go back and update things. All in all, this experience really taught me a lot about low-level programming and how different text output methods are used in building an operating system.

Github

Sufferal/os (github.com)