Contents

[04/11/2023 – 4 hours 2](#_Toc150368076)

[06/11/2023 – 3h 2](#_Toc150368077)

# 04/11/2023 – 4 hours

Today I started off by cloning the starting template for the assignment and getting qemu working on my personal PC. Once done, I then began attempting to use the int 16 bios interrupt. This is the interrupt used to retrieve a keyboard input from a user. At first I was having issues with this as I didn’t fully understand the documentation for the interrupt table. Figure 1 shows the documentation for get keystroke, in order to use this, I had to pass in the value `00` into the AH register. When trying this I got an error which stated that there was a junk “h” after the expression, this is because I had already prefixed the `00` with 0x which denotes that it is a hexadecimal value, so removing the H resolved the problems.

A close-up of a computer code

Description automatically generated

Figure – interrupt documentation for keyboard input

# 06/11/2023 – 3 hours

Figure 2 shows me being able to type the number of the sector to read, this is currently not being used but I am creating the functionality for it. The next stage is to also display the sector number in the message “Contents of sector”, to do this I will be temporarily storing the contents of al inside of the stack, this will allow me to reuse it after the other console logs have been ran.

A screenshot of a computer

Description automatically generated

Figure 2 - keyboard input working

When attempting to use the pushw instruction I ran into an error **“Error: unsupported instruction `push'”,** this is because I was using the %al register which is not supported with this instruction. After reading the lecture slides for week 3 it stated that you can only push the whole register onto the stack, once I changed it to use %ax I no longer received the error.

**Another issue which I have ran into is with the order that I was pushing and popping. I was using the pushw instruction after I call cons\_write\_crlf so the value in %al was being overwritten by the special characters. To fix the problem I had to use the pushw instruction before I called cons\_write\_crlf in order to save the correct value (the keyboard input)**

**A screen shot of a computer

Description automatically generated**

Figure 3 - bug showing the wrong value being popped onto the register

**Figure 4 displays the code used to output the contents of figure 5.**

**A screenshot of a computer program

Description automatically generated**

Figure 4 - The final code for keyboard input and contents of sector message

Figure 5 displays the final output for the code above. The next stage is to start reading the contents off of the disk.

A black screen with white text

Description automatically generated

Figure 5 - final output of keyboard input label