***Implementation Log***

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Work Done** | **Notes/thoughts** (Likely full of incoherent thoughts but need writing proof) | **Problems** |
| 18/11/2023  12pm – 5 | First 1 hr spent re-going over lectures and reading solutions to get to a point of understanding |  | Calling hexloop in finish causes a big mass of numbers already mentioned in the statement before it, I assume this is due to it reading data from an address and not actually changing it. This doesn’t then explain why the loop stops, I thought the loop should end when something returns 0.  Worked out issue, cons\_write\_hex is being tested hence why the results are same previously.  Using this I need to work out how to get to a sector input this into the loop and give the loop an end.  Various inputs loaded into week 3 solution, but these tests didn’t really help all that much |
| 19/11/2023  12 - 5 | Began work to create a loop that can display a string 16 times | Properly understanding the question asked now, perhaps if I start by hard listing locations e.g., 0x0000 then adding 1 to that I can then display whatever is in there in a loop. Then using the info from the examples given in week 3/4 I convert whatever I get to hex and even ascii.  Rereading assignment brief I need to look at week 2 lectures about reading from sector disk.  I have returned to editing and working with the week3 code. | Every time I try to get week2 stuff about reading from disk to work it keeps failing, as of now I am unsure why this is.  (Figure 1)  Keep getting this error whenever running qemu tried changing address I am writing to in bootasm2 from 0x9000 to 0xd000 and others, but same errors occur.  Things failed once again, and it seems I forgot to attempt to make the loop before uni closed. I have 30 minutes to trial that before returning home.  5pm finish after 4 hrs still where I was t=but understanding a little better.  For next time make sure to just try a loop |
| 20/11/2023  11 - 13 |  | After placing “movw (0x0000), %bx)” into the week 3 solution before using call cons\_write\_hex I get consistent outputs which leads me to assume it is reading from the disk and displaying what’s there in hex, however it seems bigger than what I assume it should look like | I still have no idea how get digit works or its purpose however I feel it is unnecessary for the module at this stage. As long as I implement hexloop I should be able to attain 50 percent |
| 21/11/2023  12 – 5 50 | A loop that displays 16 lines of 0 characters formatted like the spec suggests has been made, an understand of push and pop was used for this.  The loop now changed to increment, what were four zeros as a string, the hexadecimal value by 16 (Figure 2) |  | Creating a read sector function from the code given in bootasm.s. s and when I try use this it always results in a fail/error message I cannot work why this is or how I am to use this code. Reading the disk is still an issue that eludes me.  Still cannot figure out how to use/read the function to read from the disk. Need to speak to some people to figure how to reads, once read I should be able to convert the data it outputs, i.e., put it in the loop(s) I created and get the hex out |
| 22/11/2023 | Functionality for waiting for a key press added into the code, I should add in a message that tells the user to do this | For the longest time I was wondering why the disk reading affected my other “super\_disp\_loop” and it turns out I had misunderstood some things. The code Wayne created does something (I think it resets the stack and puts what its working with there and there for when I pop and pull things, I’m pulling the wrong number)  Break through moving 12(%si) int bx got con write hex to display something readable, hopefully this is the correct offset at the start of the sector | Not sure what is actually being read so the breakthrough might not be as broken through as I would have hoped |
| 22/11/2023 | Tried writing what’s shown in fig one as a way to read disk but seems understanding here is lacking, I attempted this as it seems I miss understand how to actually use the example given by Wayne |  |  |
| 23/11/2023  11 – 5 VM turned off to give head and eyes a reast need return soon |  | I believe I have been working with a fundamental misunderstanding b=of both what I’m reading and the task.  The offset is where I am reading from thusly when I read the value it will appear as a hex value, I then need to increment by 1 to get the next number doing this 16 times, the next line will display the offset 0010 and then I go up by one each time.  Moving (%si) into bx and then reading this gives me a hex value adding 1 to this and the reading once again shows the second half of the previous as the front of the next I think I have finally started reading what I want.  Figure 4 shows the results.  Now I need to put this in a loop.  Going down this line of thought only lead to inconsistencies past anything greater than adding 4 which leads me to believe I am not in fact reading the disk |  |
| 23/11/2023  later |  | Try using the offset calculations in the document Wayne provided? Shifting and adding numbers to get where you want? | A variety of issues and the loop seems to also break what I have done to some degree however the thing builds and something is read, the capability to loop is show and I have a key press (30%?) leave here and pick up slack in phase 2 of project |

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

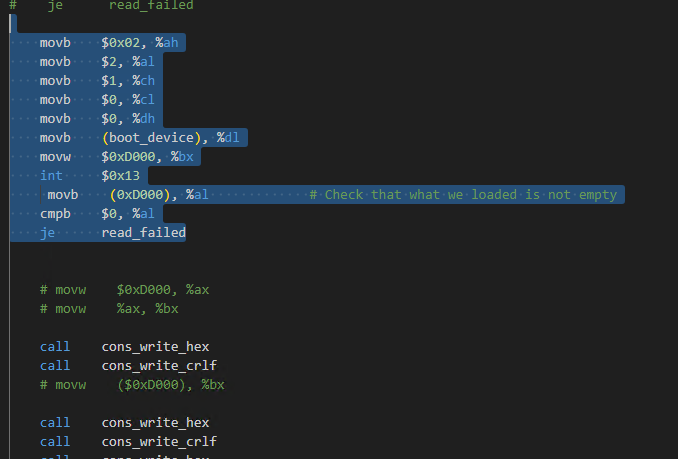


Figure 3

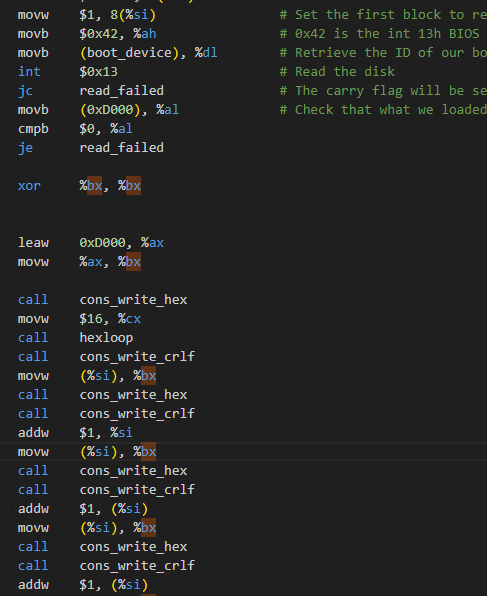


Figure shows that adding $1 to (%si) shifts the hex value read to the left.