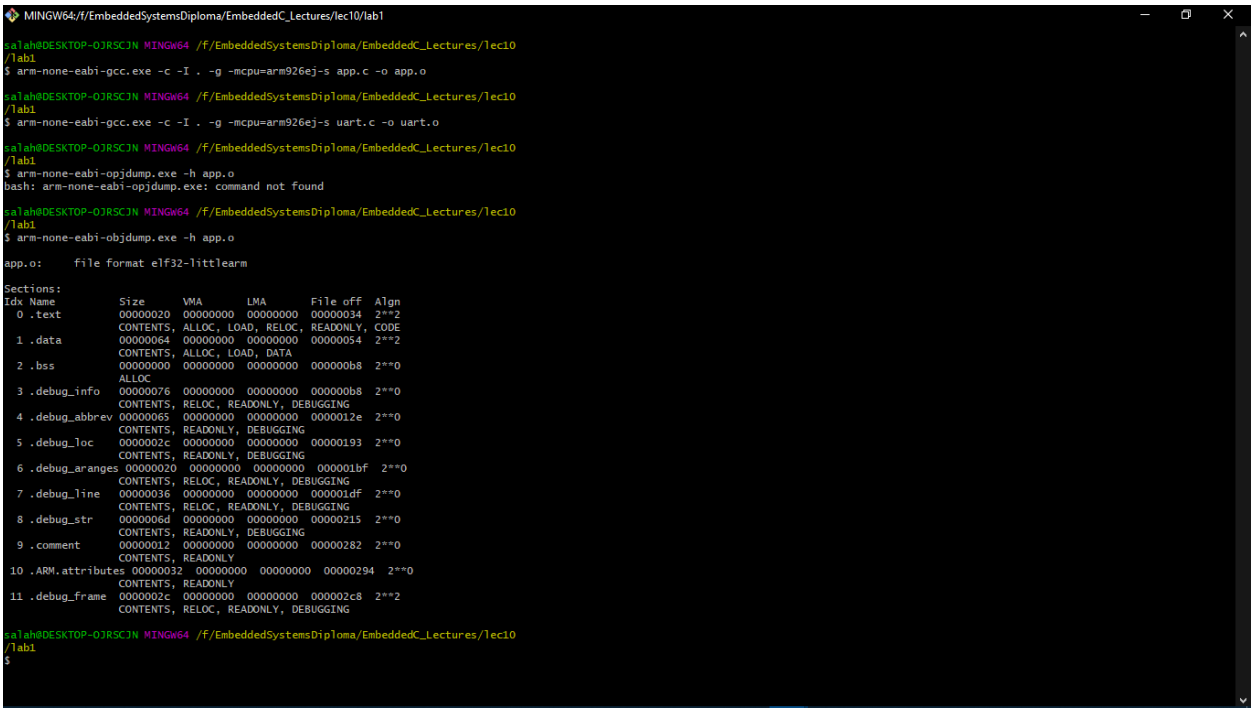
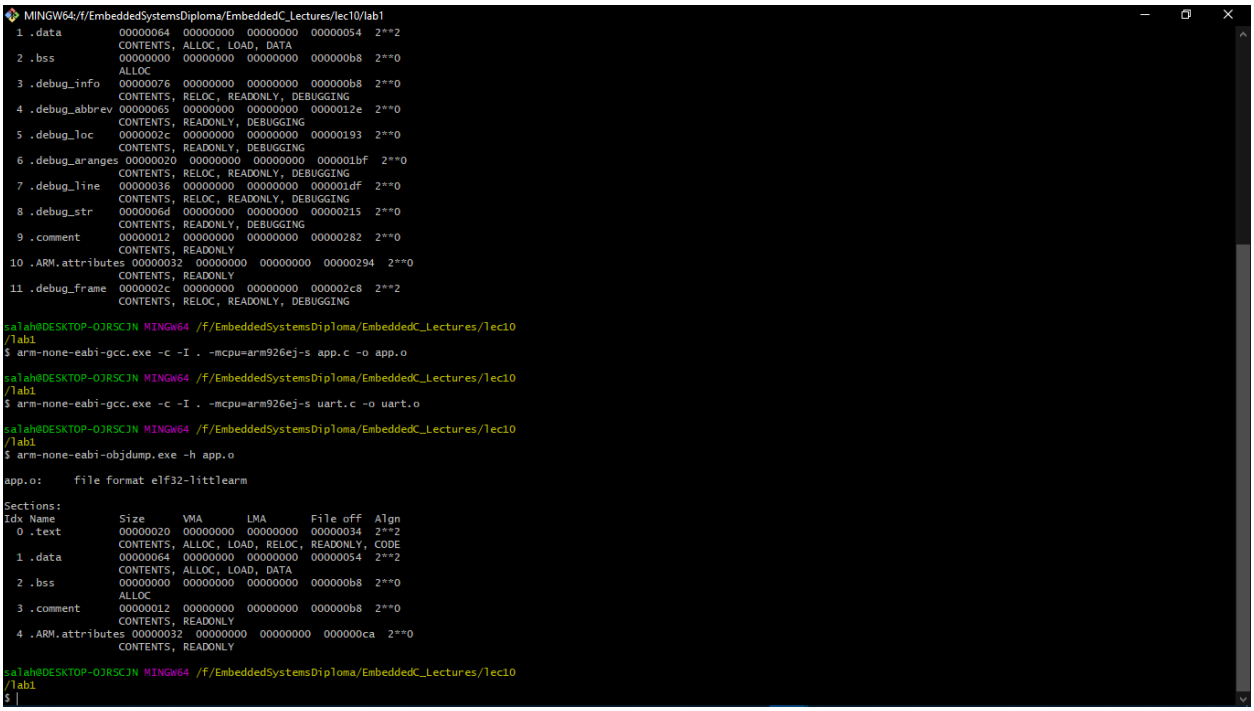
LAB ONE (Second embedded C Lecture)

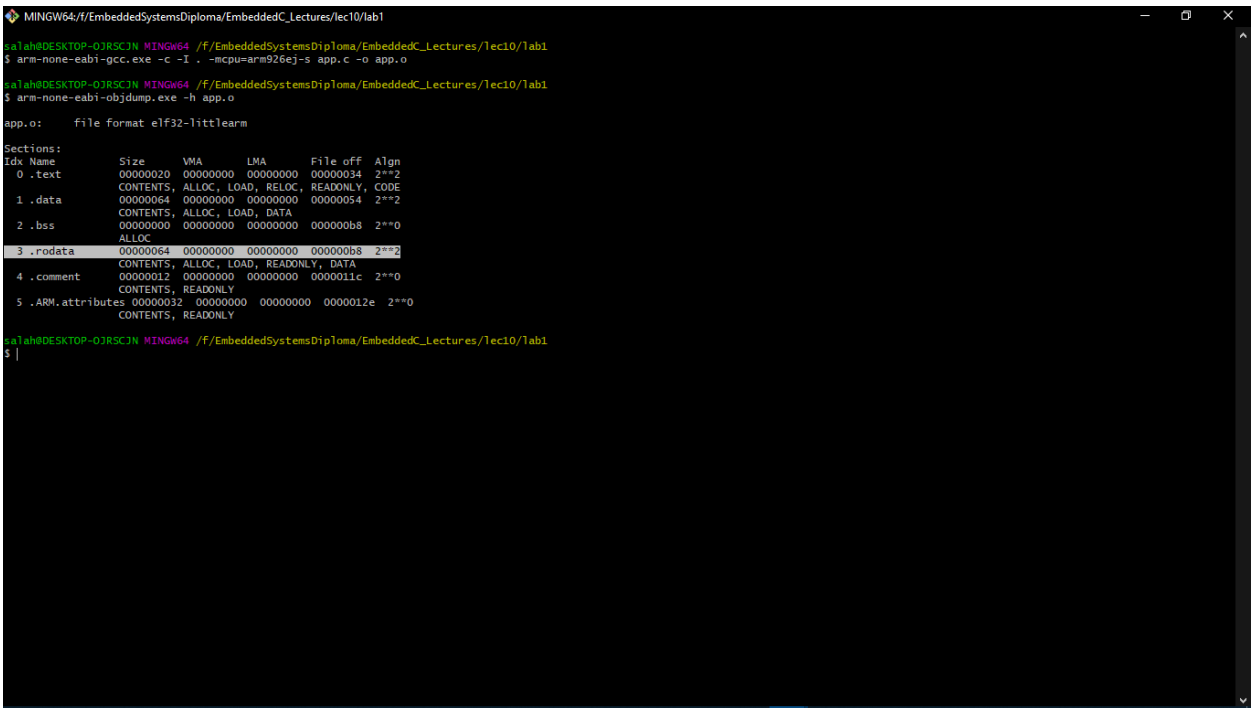
Using obj-dump utility (-h) to show sections headers with debugging information.



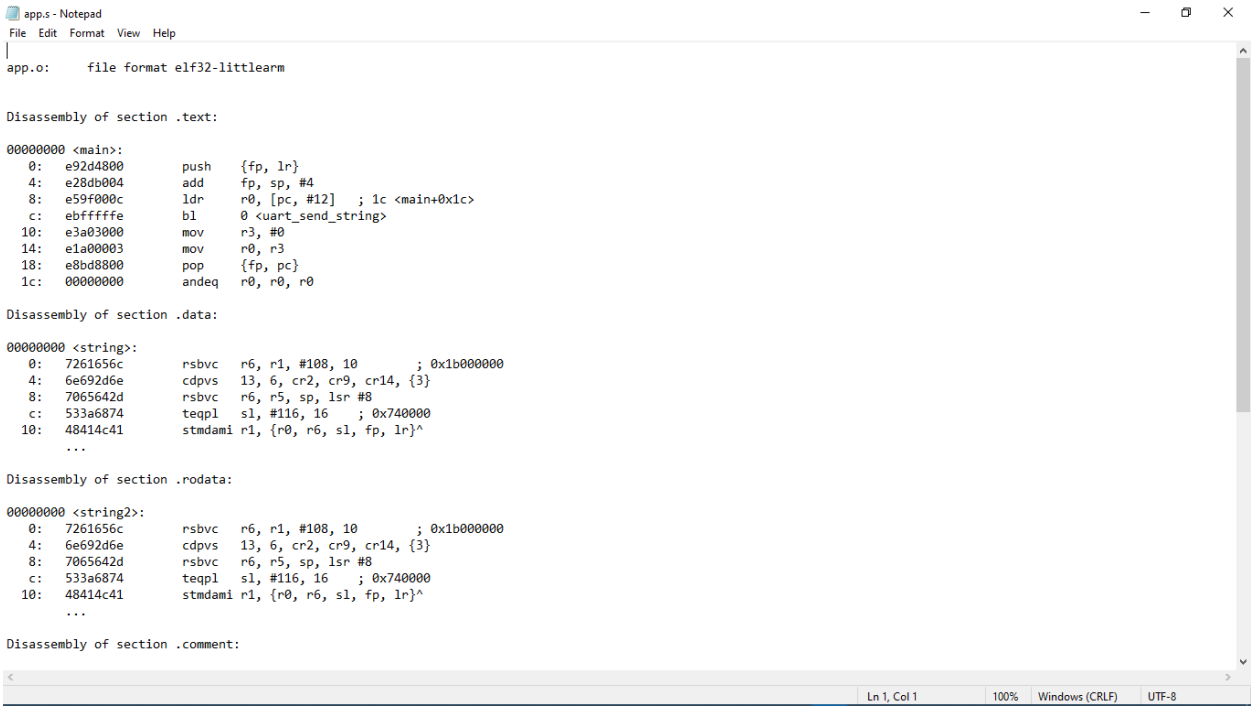
Using obj-dump utility (-h) to show sections headers without debugging information.



Using obj-dump utility (-h) to show sections headers after adding constant global data.



Using obj-dump utility (-D) to show disassembly of app.o file.



Using obj-dump utility (-D) to show disassembly of uart.o file.

uart.s - Notepad

File Edit Format View Help

```
uart.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <uart_send_string>:
   0:   e52db004        push    {fp}            ; (str fp, [sp, #-4]!)
   4:   e28db000        add     fp, sp, #0
   8:   e24dd00c        sub     sp, sp, #12
   c:   e50b0008        str     r0, [fp, #-8]
  10:   ea000006        b       30 <uart_send_string+0x30>
  14:   e59f3030        ldr     r3, [pc, #48]   ; 4c <uart_send_string+0x4c>
  18:   e51b2008        ldr     r2, [fp, #-8]
  1c:   e5d22000        ldrb    r2, [r2]
  20:   e5832000        str     r2, [r3]
  24:   e51b3008        ldr     r3, [fp, #-8]
  28:   e2833001        add     r3, r3, #1
  2c:   e50b3008        str     r3, [fp, #-8]
  30:   e51b3008        ldr     r3, [fp, #-8]
  34:   e5d33000        ldrb    r3, [r3]
  38:   e3530000        cmp     r3, #0
  3c:   1afffff4        bne     14 <uart_send_string+0x14>
  40:   e28bd000        add     sp, fp, #0
  44:   e8bd0800        ldmfd   sp!, {fp}
  48:   e12fff1e        bx      lr
  4c:   101f1000        andsne  r1, pc, r0

Disassembly of section .comment:

00000000 <.comment>:
   0:   43434700        movtmi  r4, #14080      ; 0x3700
   4:   4728203a                        ; <UNDEFINED> instruction: 0x4728203a
   8:   2029554e        eorcs   r5, r9, lr, asr #10
   c:   2e372e34        mrccs   14, 1, r2, cr7, cr4, {1}
  10:   Address 0x00000010 is out of bounds.
```

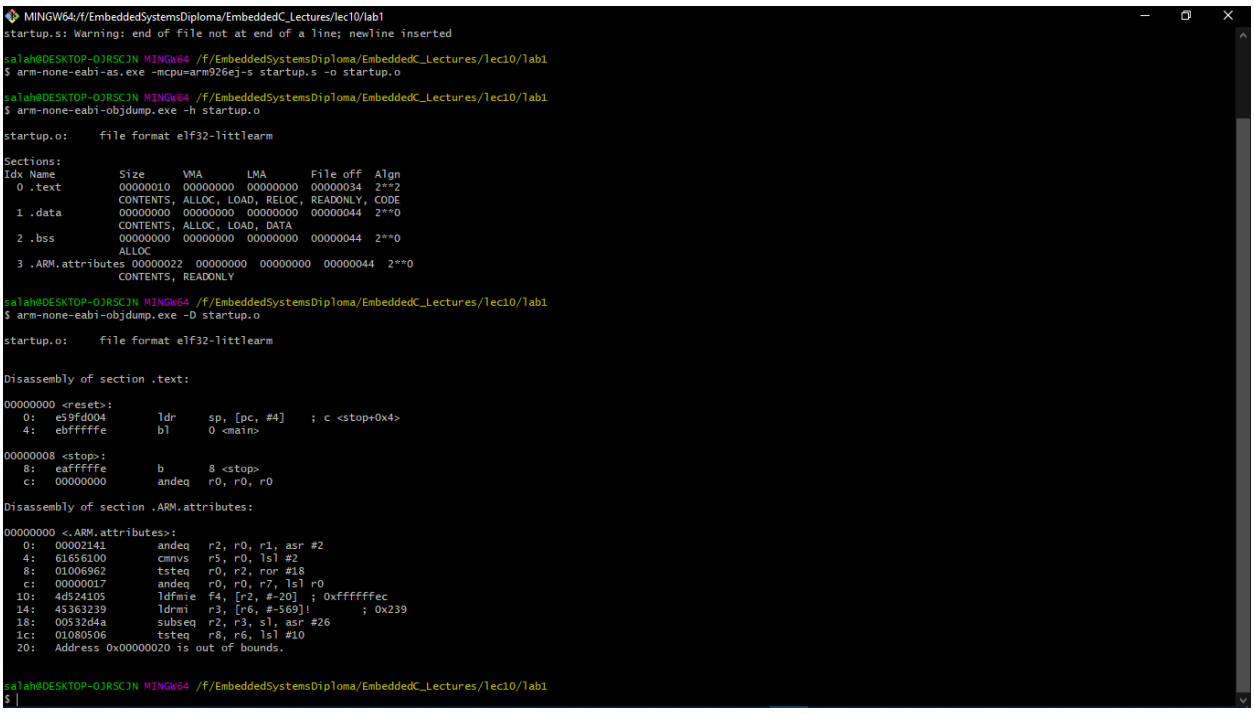
Ln 7, Col 29 | 100% | Windows (CRLF) | UTF-8

Using obj-dump utility (-s) to show all sections information of app.o file.

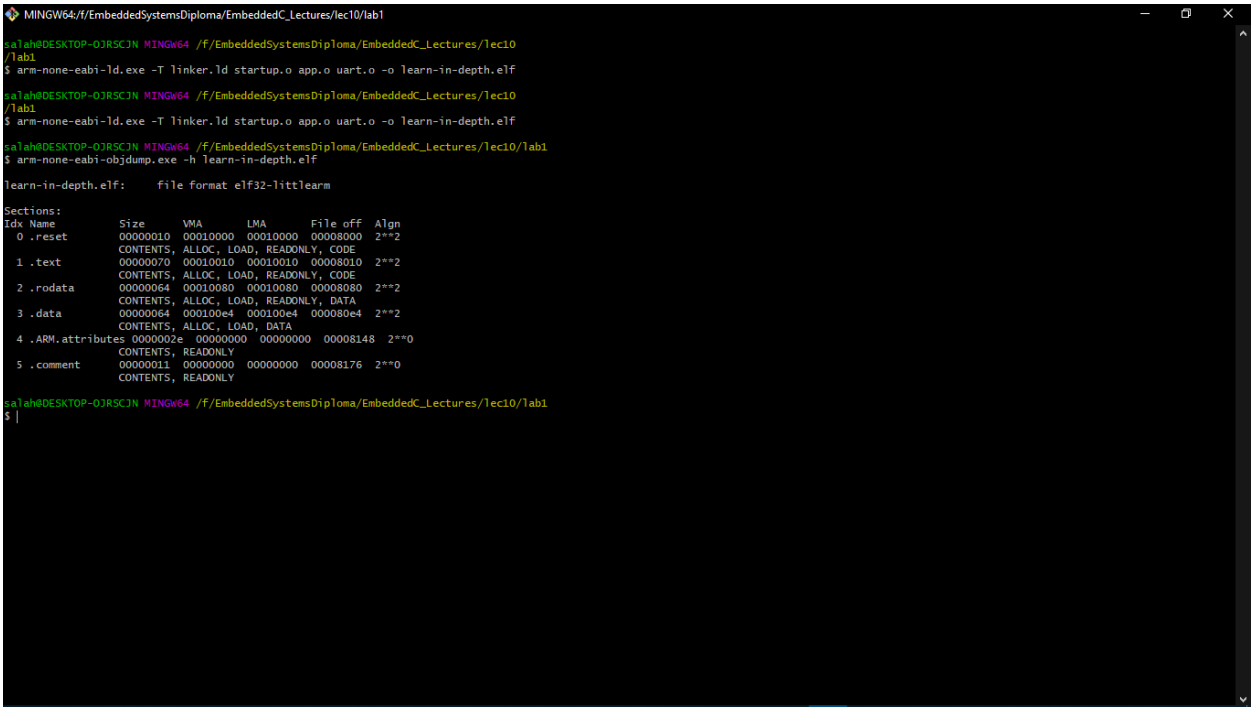
MINGW64:/f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1

```
salah@DESKTOP-OJRSCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10
/lab1
$ arm-none-eabi-obj
arm-none-eabi-objcopy.exe  arm-none-eabi-objdump.exe

salah@DESKTOP-OJRSCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10
/lab1
$ arm-none-eabi-objdump.exe -s app.o

app.o:     file format elf32-littlearm

Contents of section .text:
 0000 00482de9 04b08de2 0c009fe5 feffffeb  .H-.............
 0010 0030a0e3 0300a0e1 0088bde8 00000000  .0..............
Contents of section .data:
 0000 6c656172 6e2d696e 2d646570 74683a53  learn-in-depth:S
 0010 414c4148 00000000 00000000 00000000  ALAH............
 0020 00000000 00000000 00000000 00000000  ................
 0030 00000000 00000000 00000000 00000000  ................
 0040 00000000 00000000 00000000 00000000  ................
 0050 00000000 00000000 00000000 00000000  ................
 0060 00000000                             ....
Contents of section .rodata:
 0000 6c656172 6e2d696e 2d646570 74683a53  learn-in-depth:S
 0010 414c4148 00000000 00000000 00000000  ALAH............
 0020 00000000 00000000 00000000 00000000  ................
 0030 00000000 00000000 00000000 00000000  ................
 0040 00000000 00000000 00000000 00000000  ................
 0050 00000000 00000000 00000000 00000000  ................
 0060 00000000                             ....
Contents of section .comment:
 0000 00474343 3a202847 4e552920 342e372e  .GCC: (GNU) 4.7.
 0010 3200                                 2.
Contents of section .ARM.attributes:
 0000 41310000 00616561 62690001 27000000  A1...aeabi..'...
 0010 0541524d 39323645 4a2d5300 06050801  .ARM926EJ-S.....
 0020 09011204 14011501 17031801 19011a01  ................
 0030 1e06                                 ..

salah@DESKTOP-OJRSCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10
/lab1
$
```

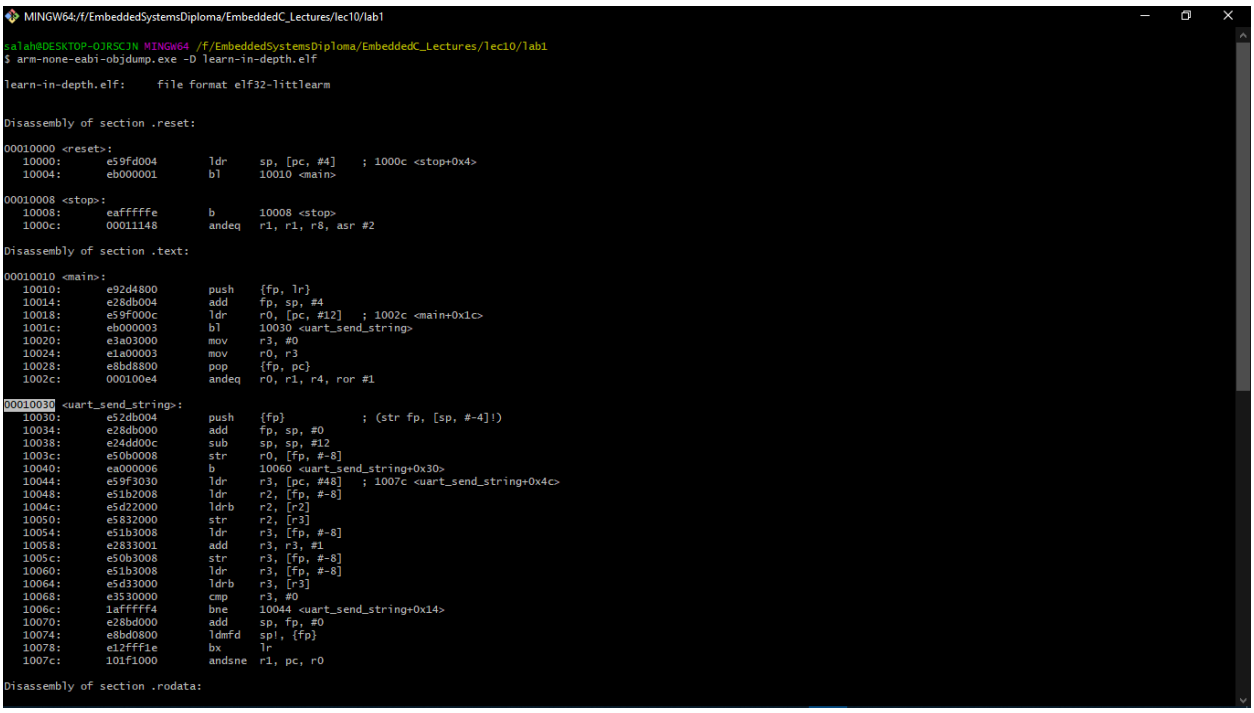
Similar information for the startup file.



Using obj-dump utility (-h) to show sections headers information of the final executable after linking.

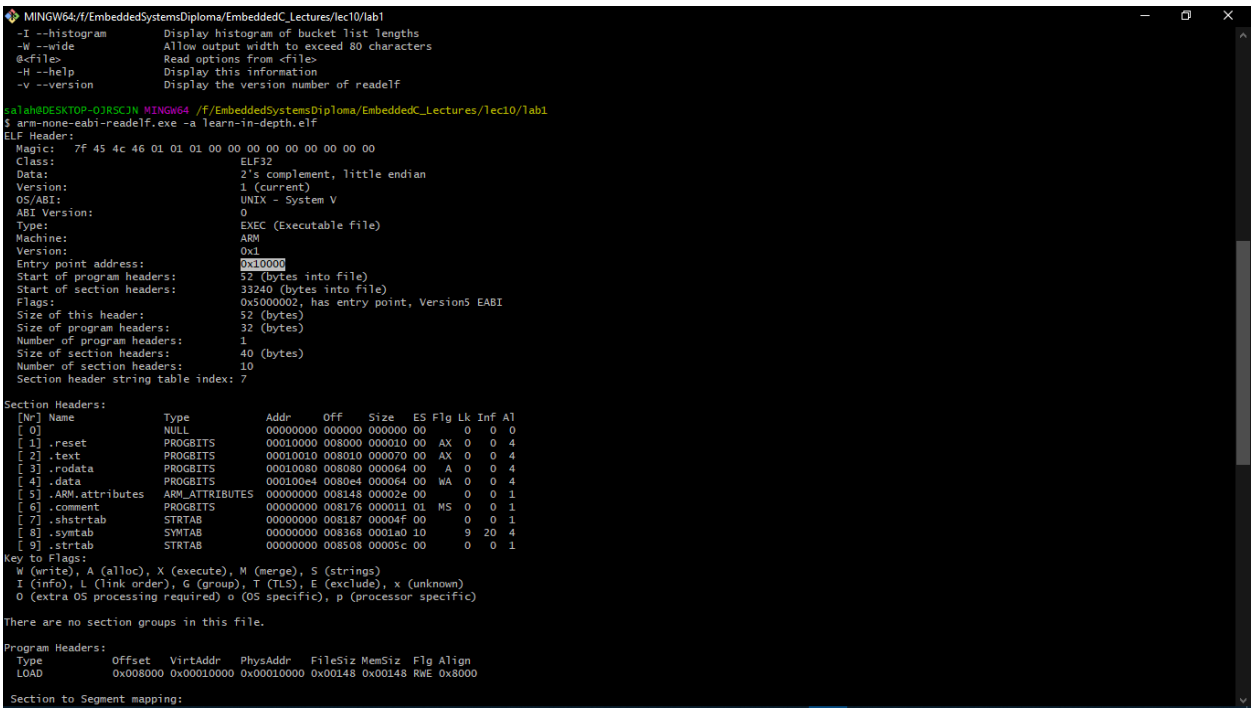


Using obj-dump utility (-D) to show disassembly of the final executable after linking



Using nm cross toolchain utility to show symbols of the objects and executable files.

Using read-elf cross toolchain utility to show the entry point address.



Run the program on the QEMU simulator.