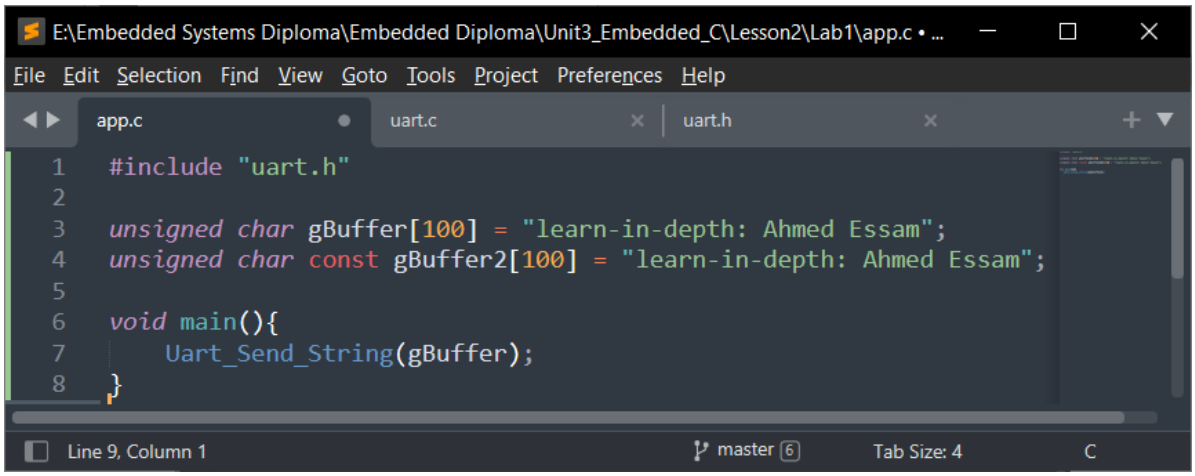
Mastering Embedded Systems

Unit 3

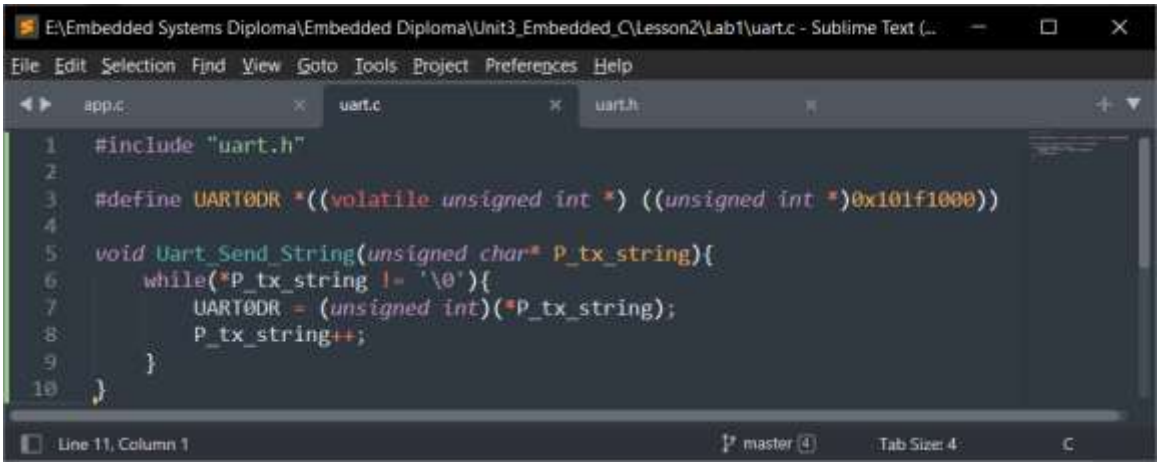
Embedded C

Lab 1

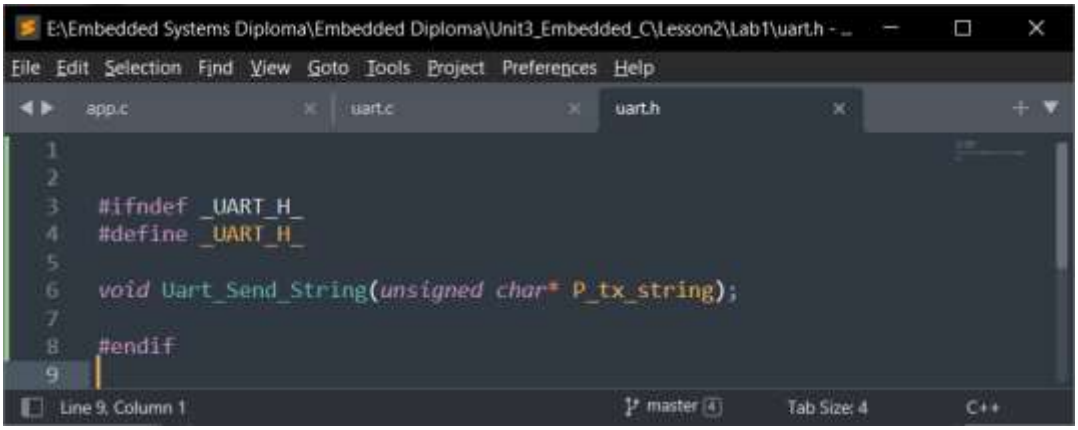
App.c:



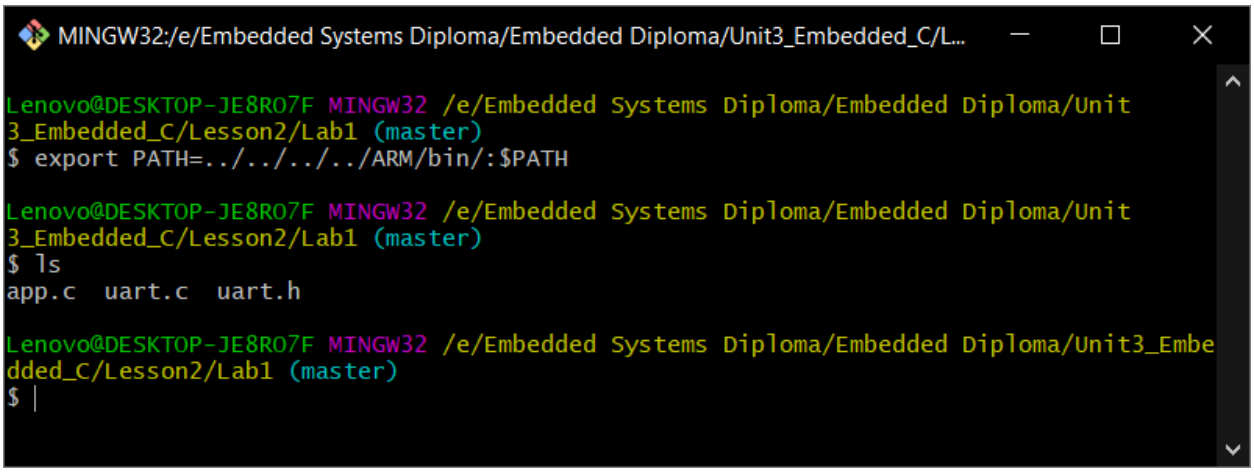
Uart.c:



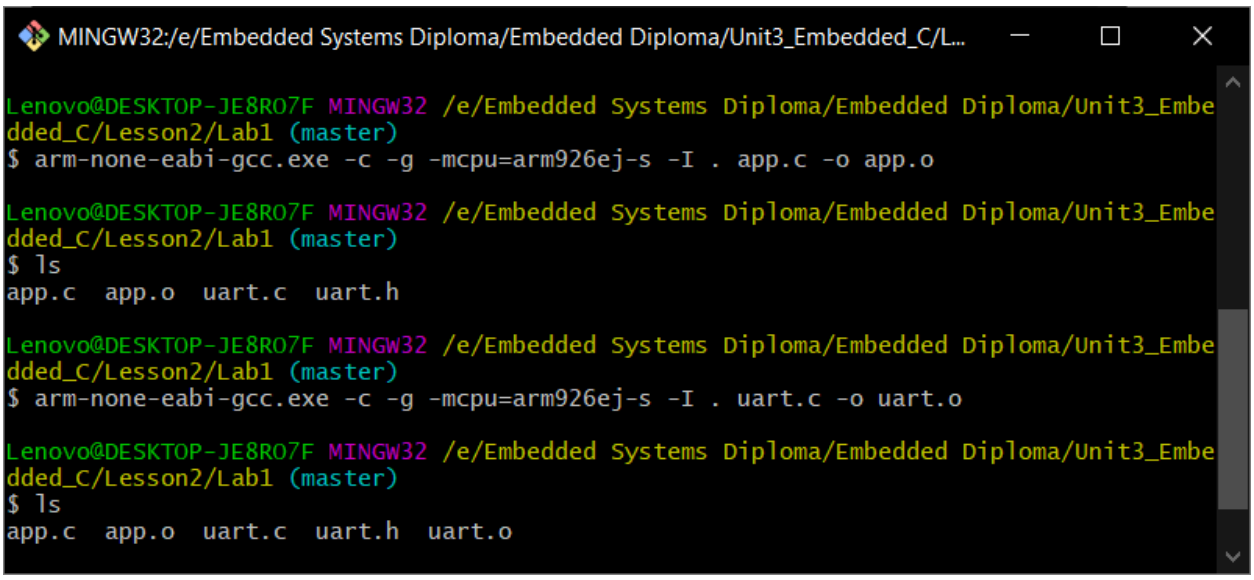
Uart.h:



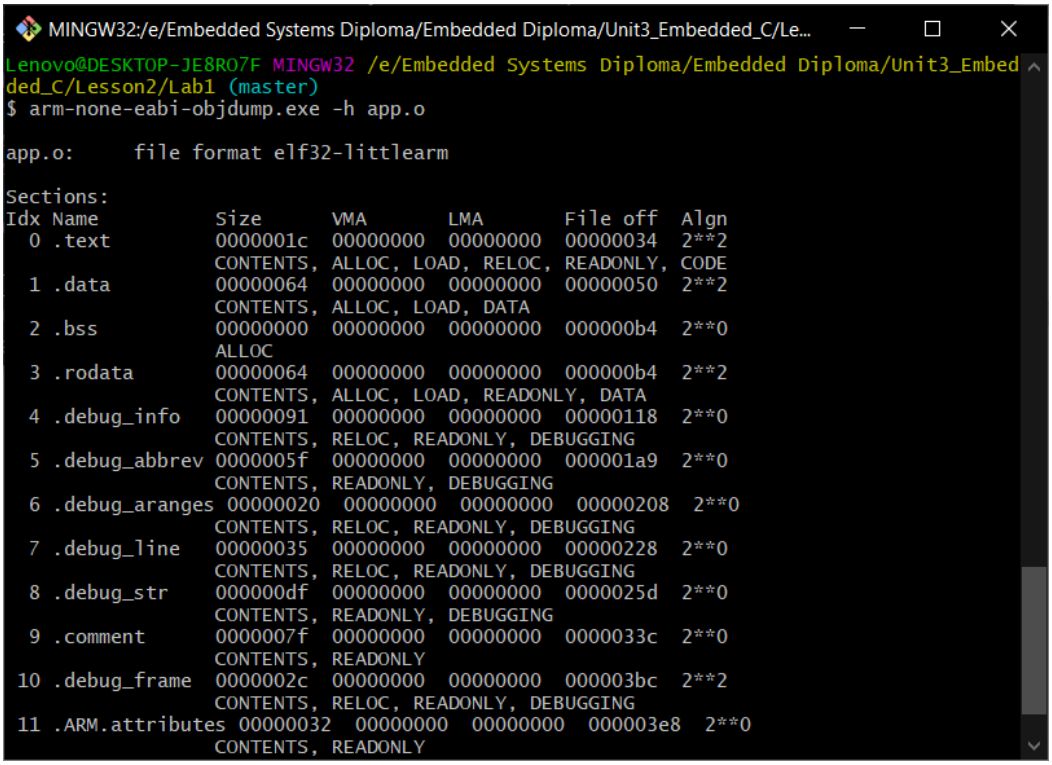
Exporting ARM cross tool chain to our path for easier use then viewing current directory files:



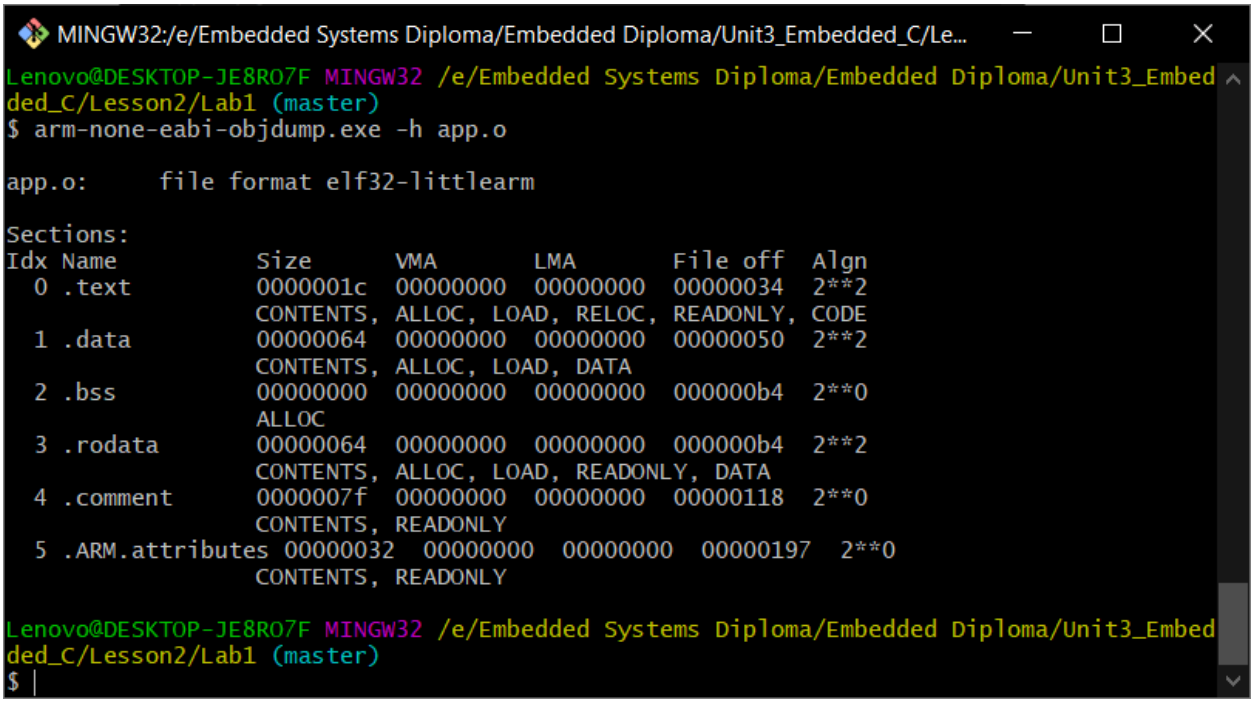
Exporting app.o & uart.o object files “AKA: Relocatable Binary” with debugging info:



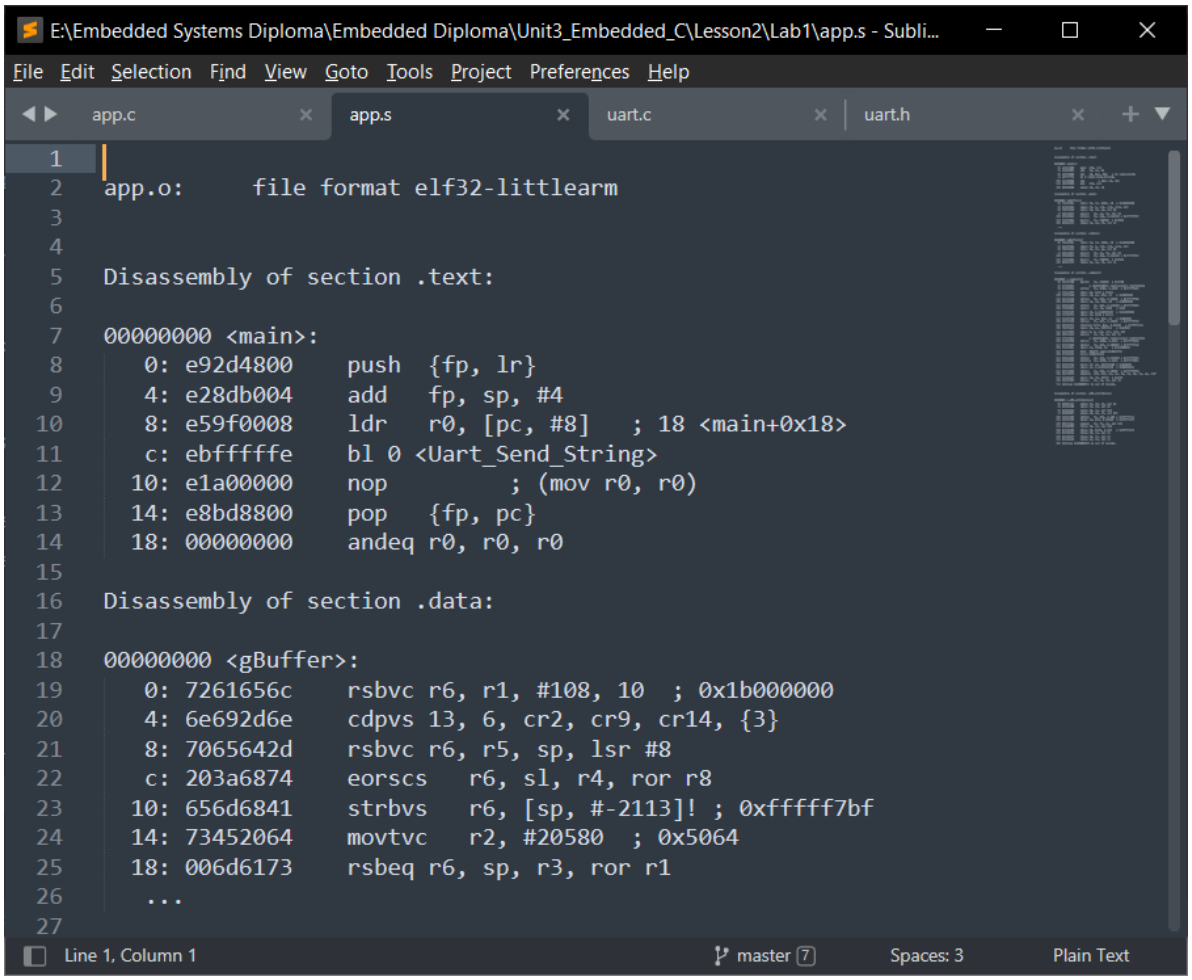
Section Headers for app.o with debugging info:



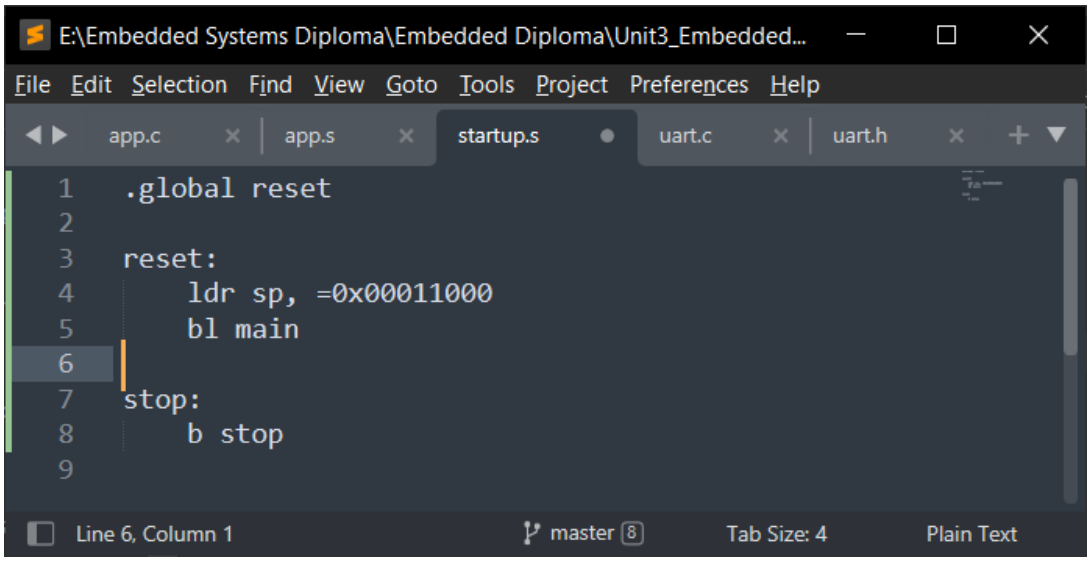
Section Headers for app.o without debugging info:



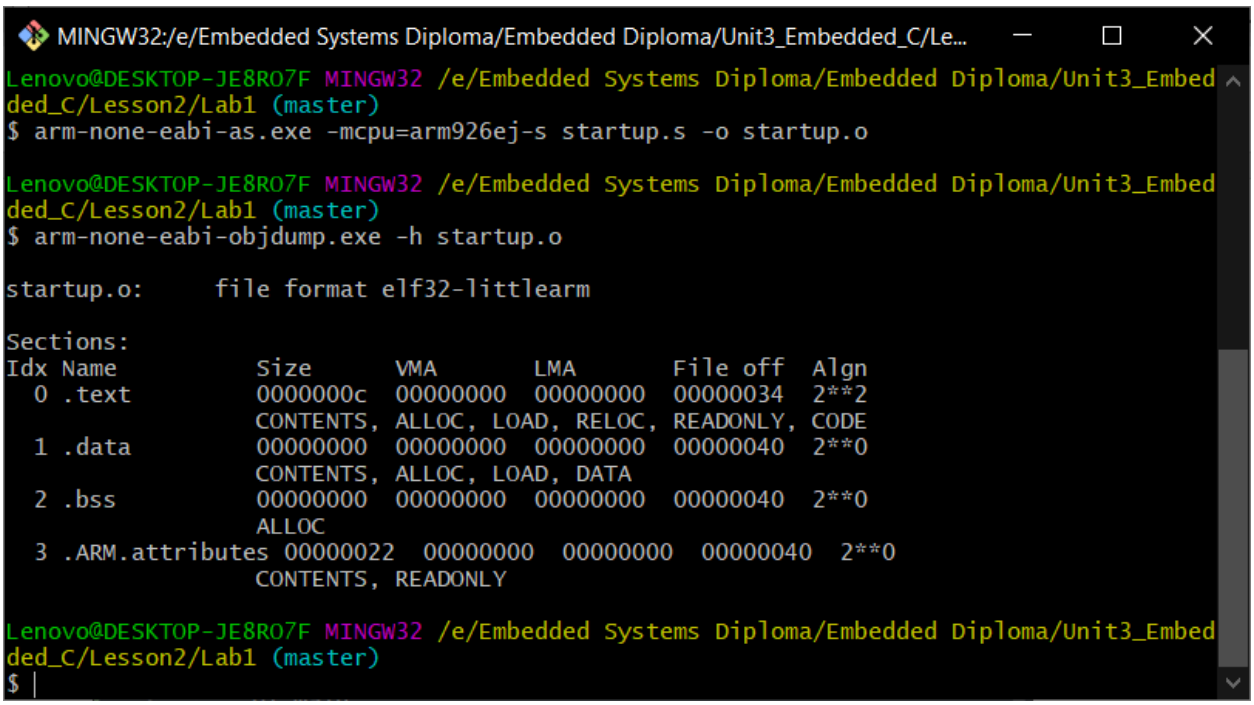
Disassembly of app.o file:



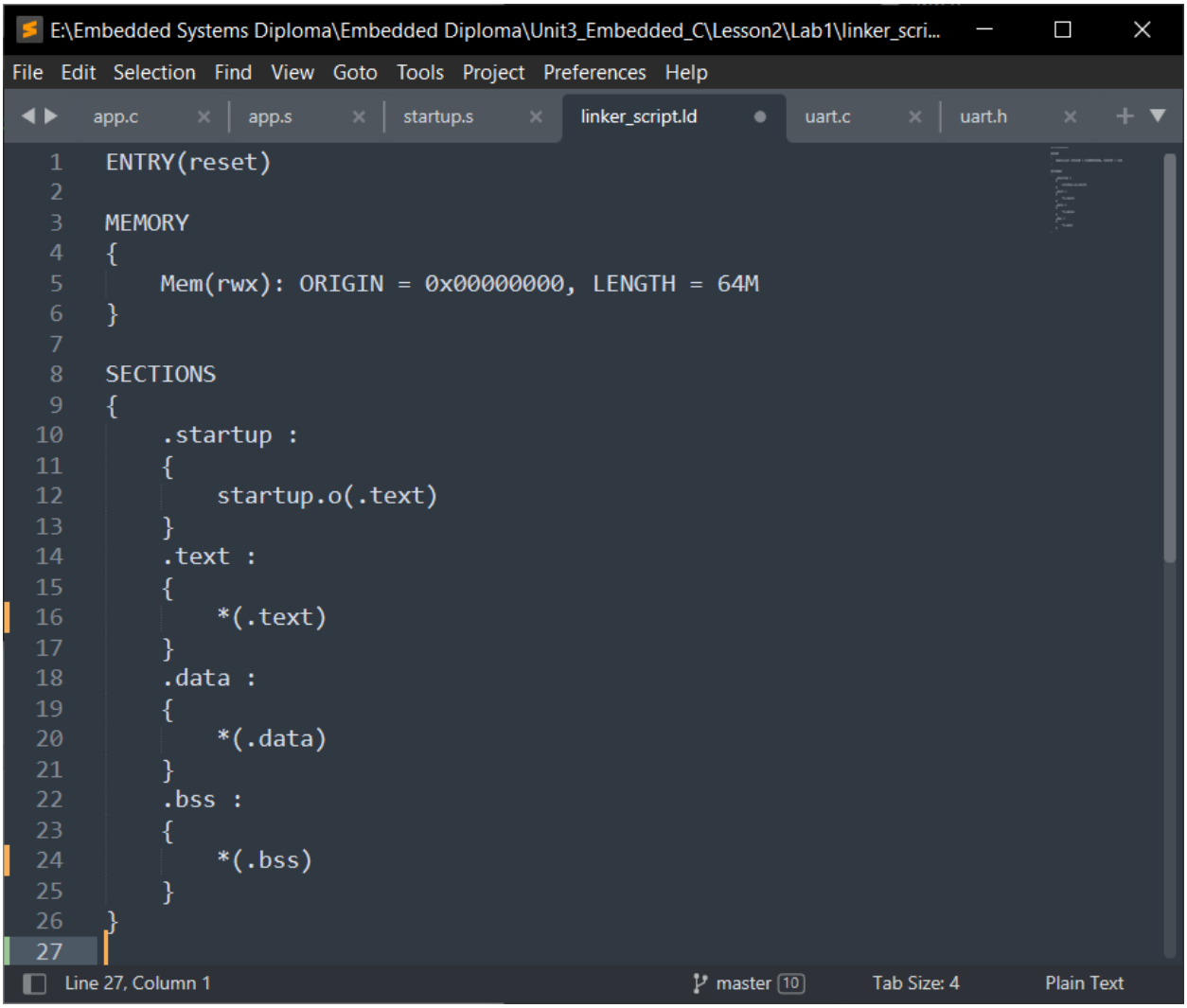
Startup.s demo with hard-coded stack pointer address:



Startup.o objdump:



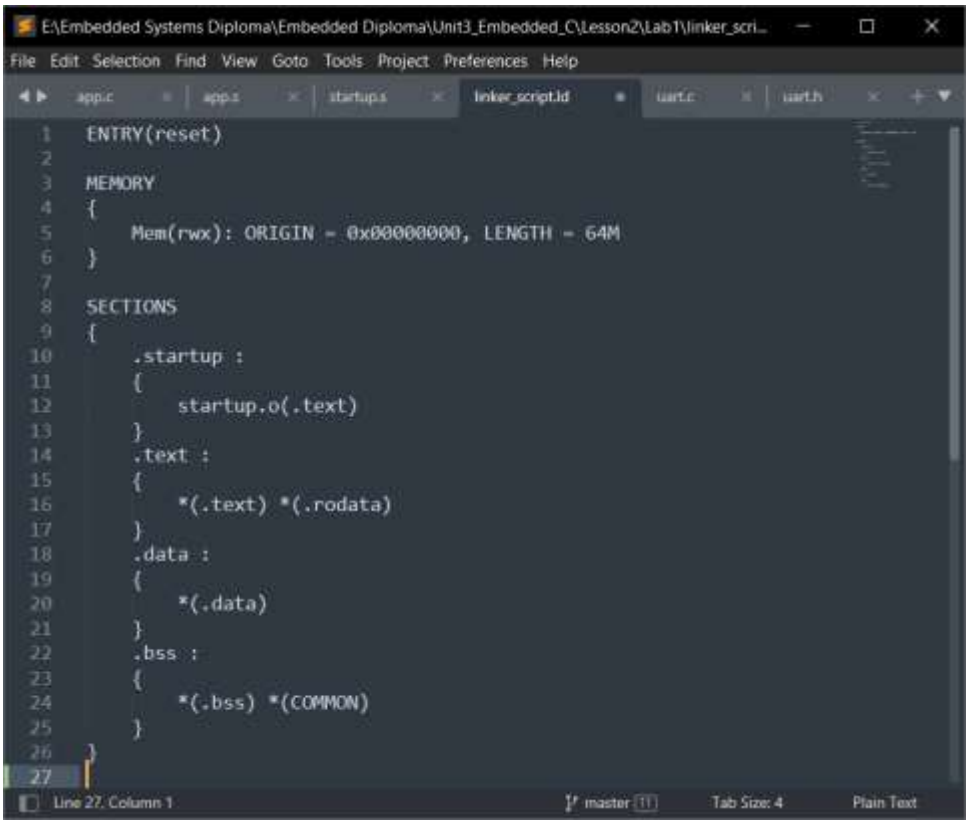
Linker_script.ld initial:



Linking all object files with linker_script.ld in learn-in-depth.elf and dumping data:

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8RO7F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

learn-in-depth.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      0000000c  00000000  00000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000070  0000000c  0000000c  0001000c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .rodata       00000064  0000007c  0000007c  0001007c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .data         00000064  000000e0  000000e0  000100e0  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  4 .ARM.attributes 0000002e  00000000  00000000  00010144  2**0
                  CONTENTS, READONLY
  5 .comment      0000007e  00000000  00000000  00010172  2**0
                  CONTENTS, READONLY

Lenovo@DESKTOP-JE8RO7F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$
```

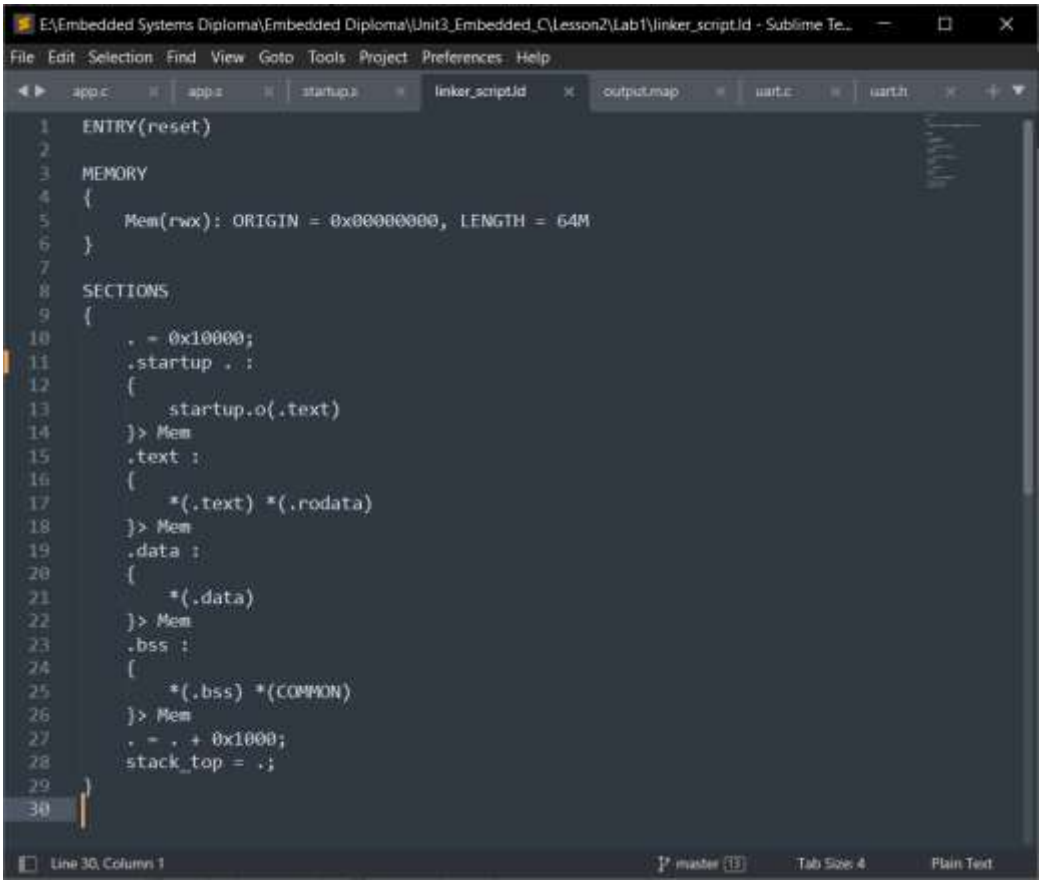
Linker_script after adding .rodata to .text section:



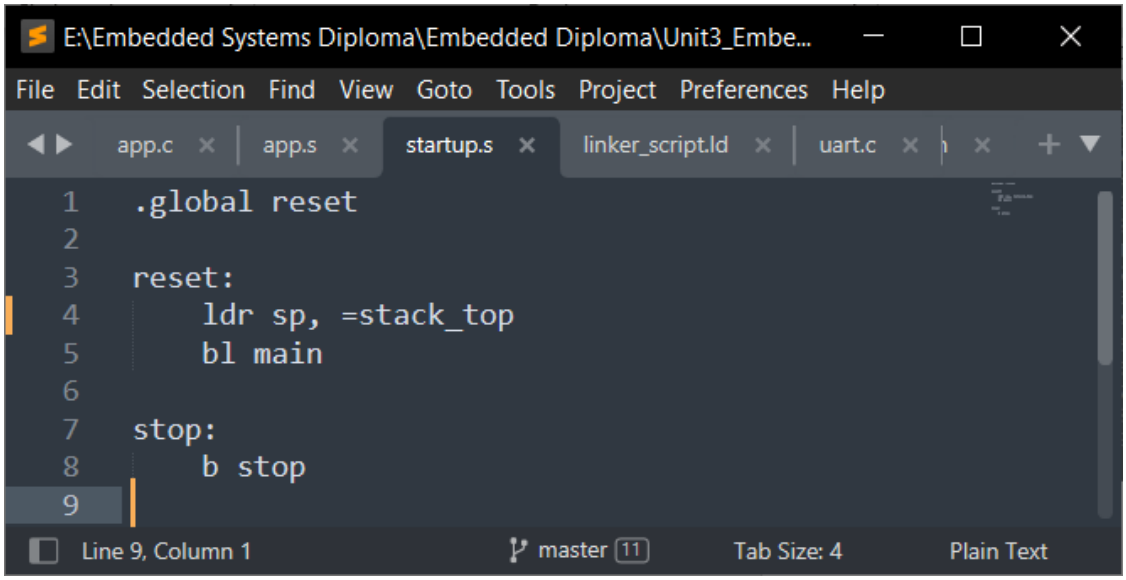
Dumping data in learn-in-depth.elf:

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1

Lenovo@DESKTOP-JE8RO7F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-ld.exe -T linker_script.ld app.o uart.o startup.o -o learn-in-depth.elf

Lenovo@DESKTOP-JE8RO7F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

learn-in-depth.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      0000000c  00000000  00000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         000000d4  0000000c  0000000c  0001000c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  000000e0  000000e0  000100e0  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  00010144  2**0
                  CONTENTS, READONLY
  4 .comment      0000007e  00000000  00000000  00010172  2**0
                  CONTENTS, READONLY
```

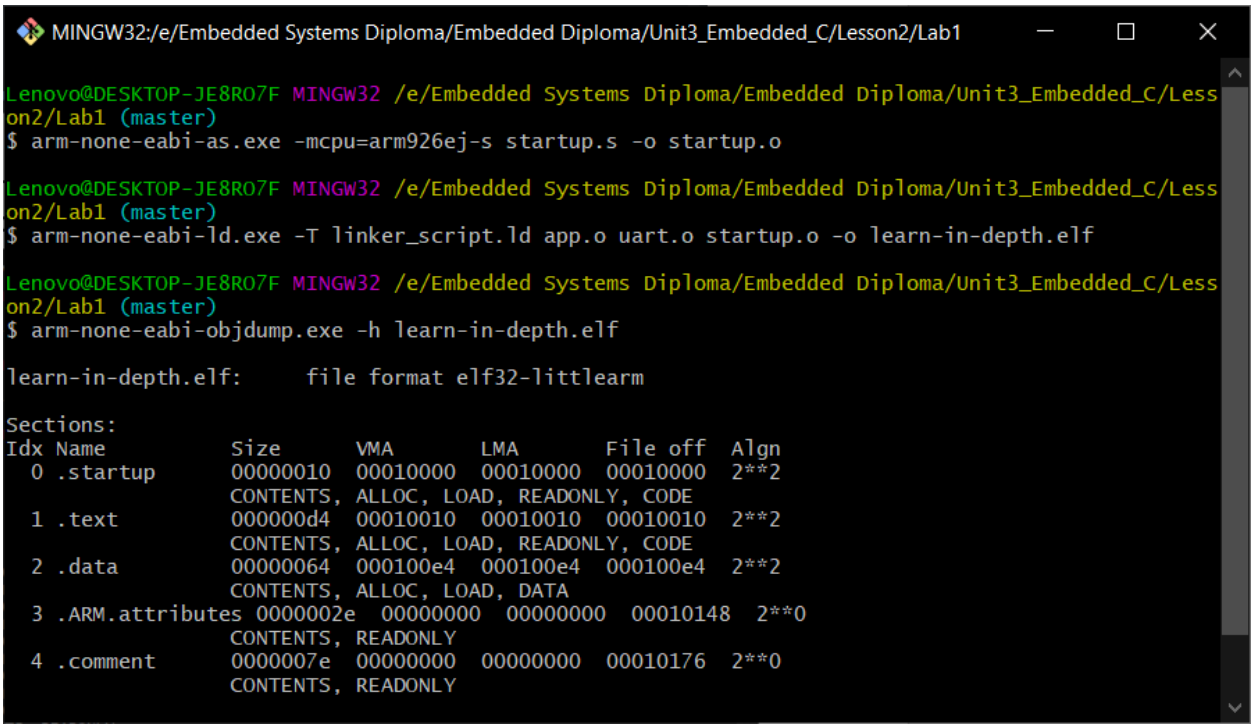
Adding VMA/LMA to current sections:



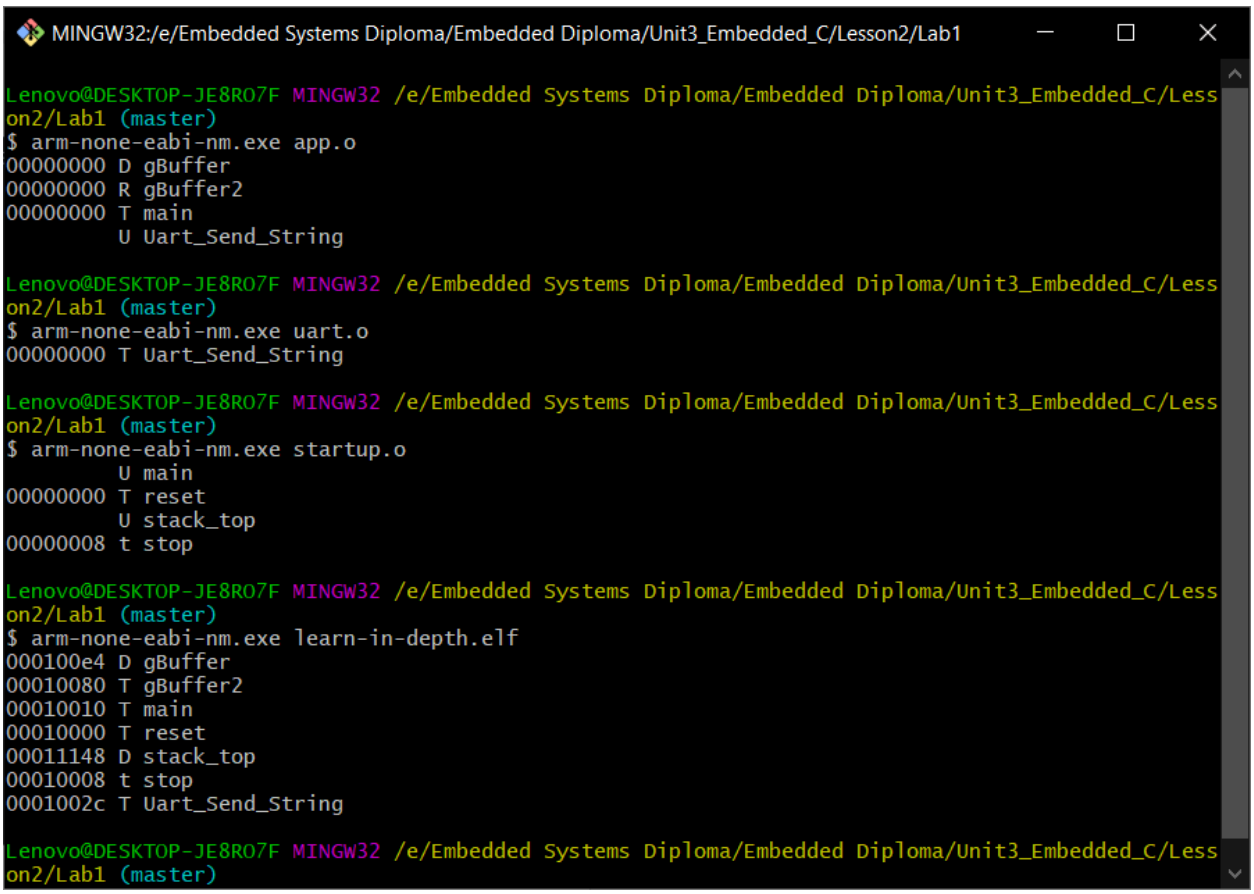
Modified startup.s after adding stack_top address:



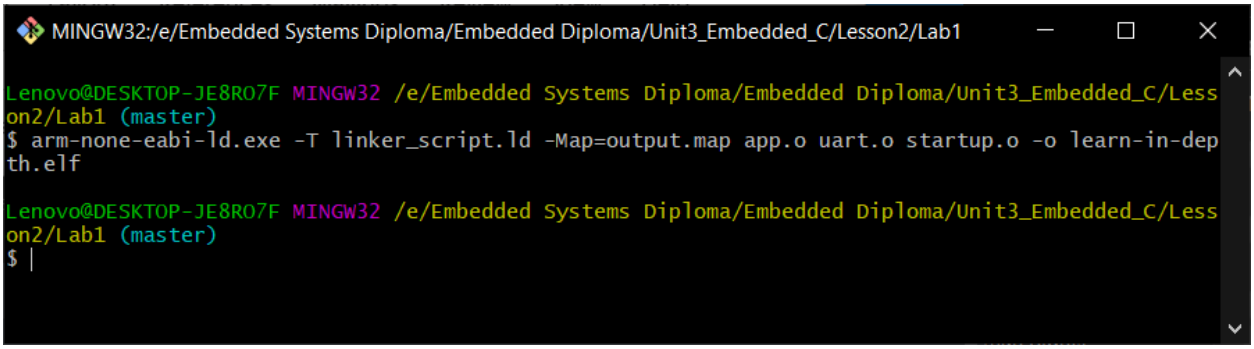
Dumping data in learn-in-depth.elf after adding VMA/LMA to all sections and modifying startup.s:



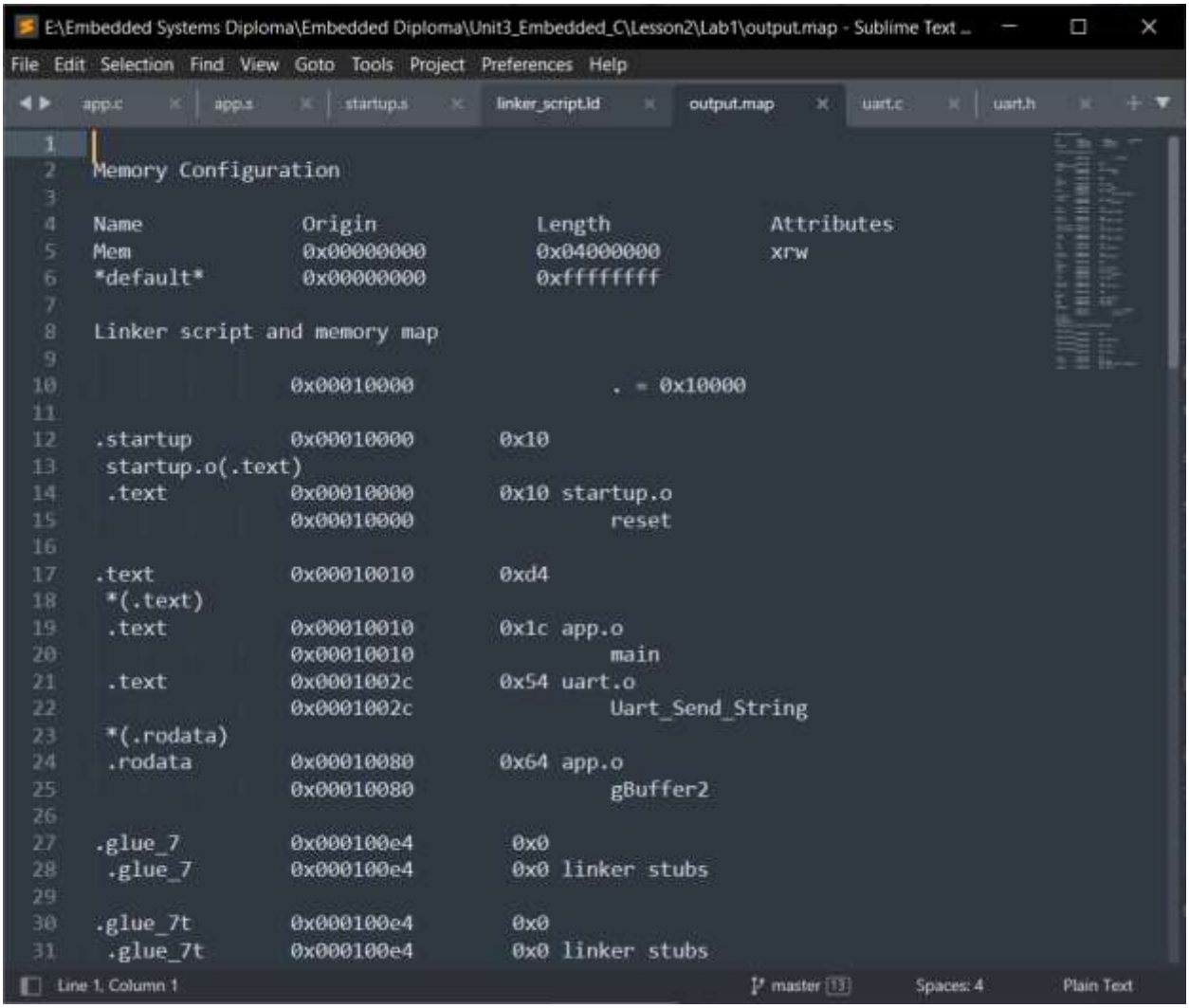
Reading symbols in each of the following files to elaborate resolving symbols after linking/locating stage:



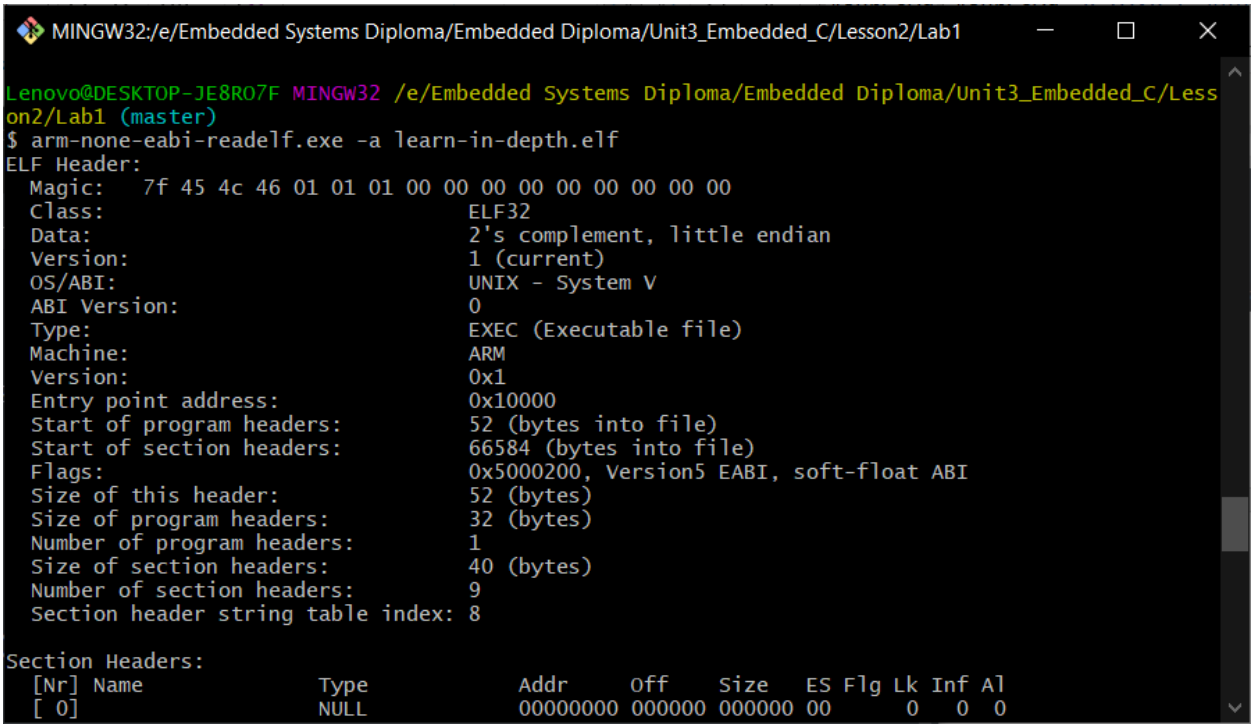
Exporting map file that contains complete list of code with their addresses:



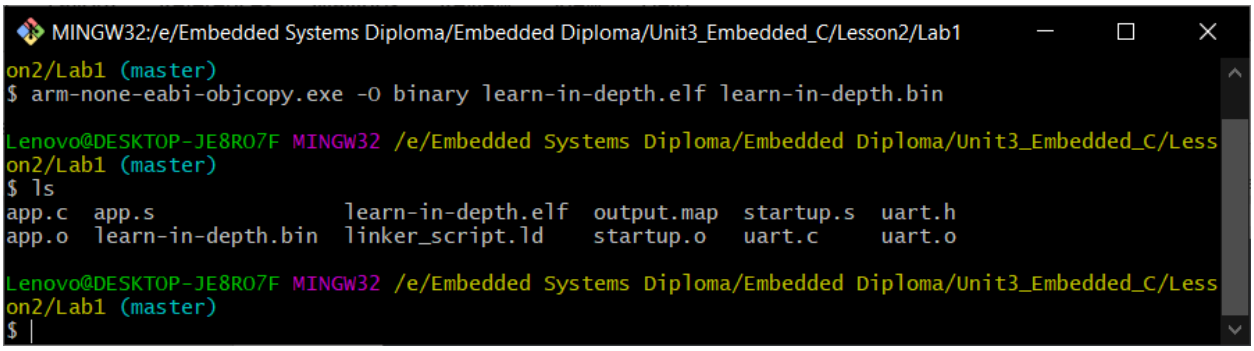
Output.map:



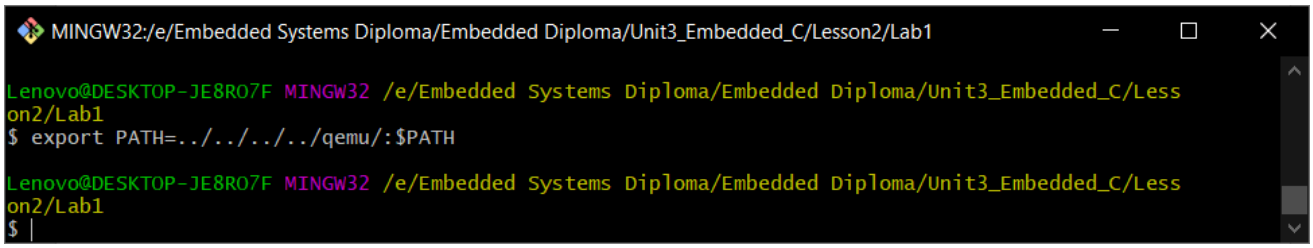
Using readelf utility to check entry point address:



Using binary utilities to strip .elf file to extract .bin file:



Exporting qemu to our path for easier use then viewing current directory files:



Running learn-in-depth.bin on qemu: