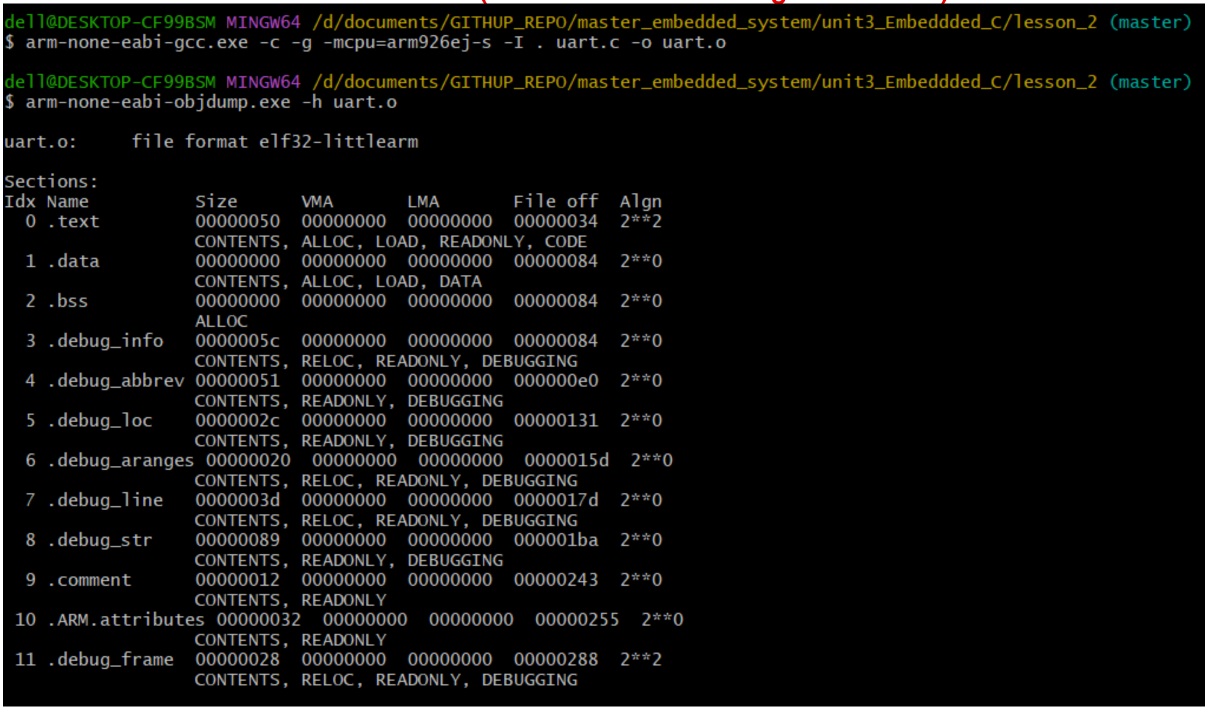
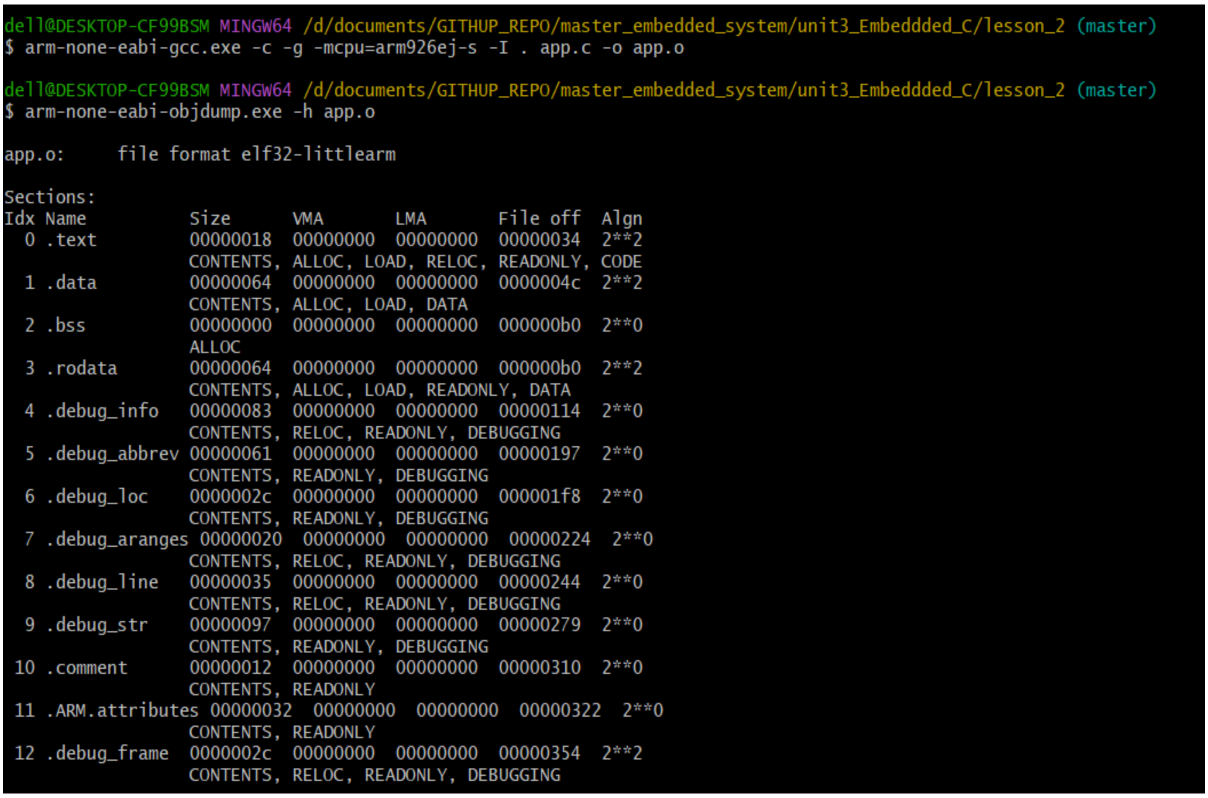
LAB 1
Using VersatilePB virtual board in Qemu and ARM toolchain

1. Writing source files, getting object files and analyzing them.

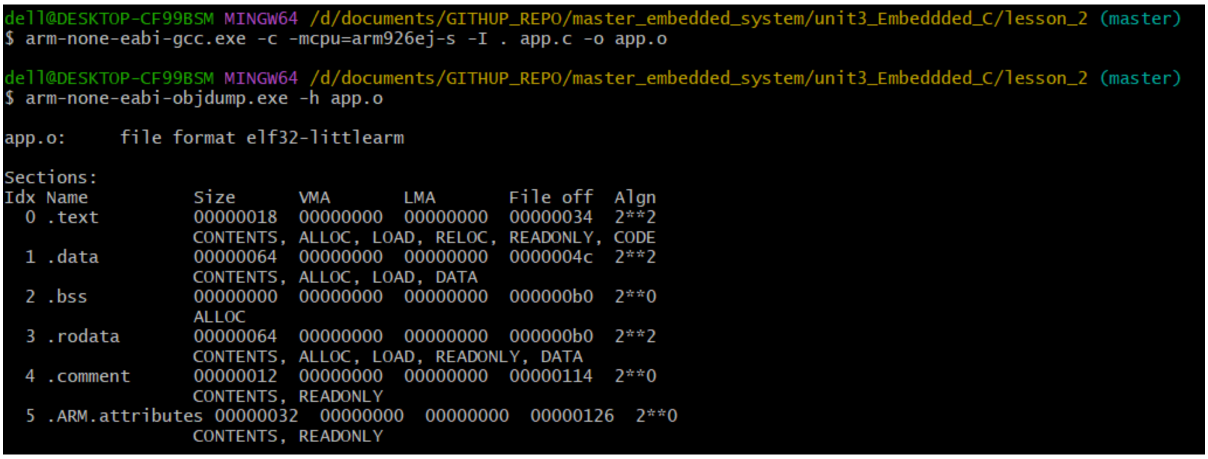
1.1(with and without debug information)



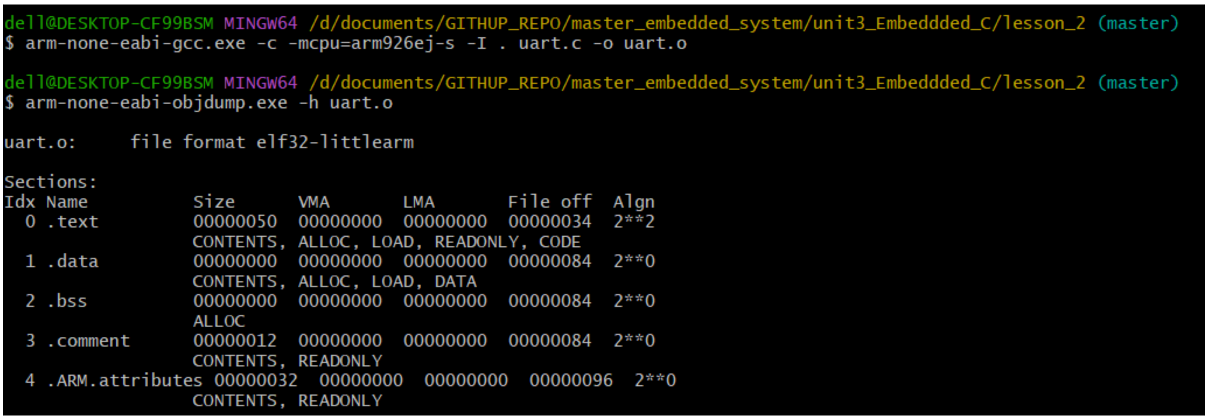
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-gcc.exe -c -g -mcpu=arm926ej-s -I . uart.c -o uart.o

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h uart.o

uart.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000050  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000084  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000084  2**0
                  ALLOC
  3 .debug_info   0000005c  00000000  00000000  00000084  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000051  00000000  00000000  000000e0  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  00000131  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000015d  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000003d  00000000  00000000  0000017d  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000089  00000000  00000000  000001ba  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  00000243  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000032  00000000  00000000  00000255  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000028  00000000  00000000  00000288  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```



```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-gcc.exe -c -g -mcpu=arm926ej-s -I . app.c -o app.o

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h app.o

app.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000018  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .debug_info   00000083  00000000  00000000  00000114  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 00000061  00000000  00000000  00000197  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    0000002c  00000000  00000000  000001f8  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00000224  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   00000035  00000000  00000000  00000244  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    00000097  00000000  00000000  00000279  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      00000012  00000000  00000000  00000310  2**0
                  CONTENTS, READONLY
 11 .ARM.attributes 00000032  00000000  00000000  00000322  2**0
                  CONTENTS, READONLY
 12 .debug_frame  0000002c  00000000  00000000  00000354  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

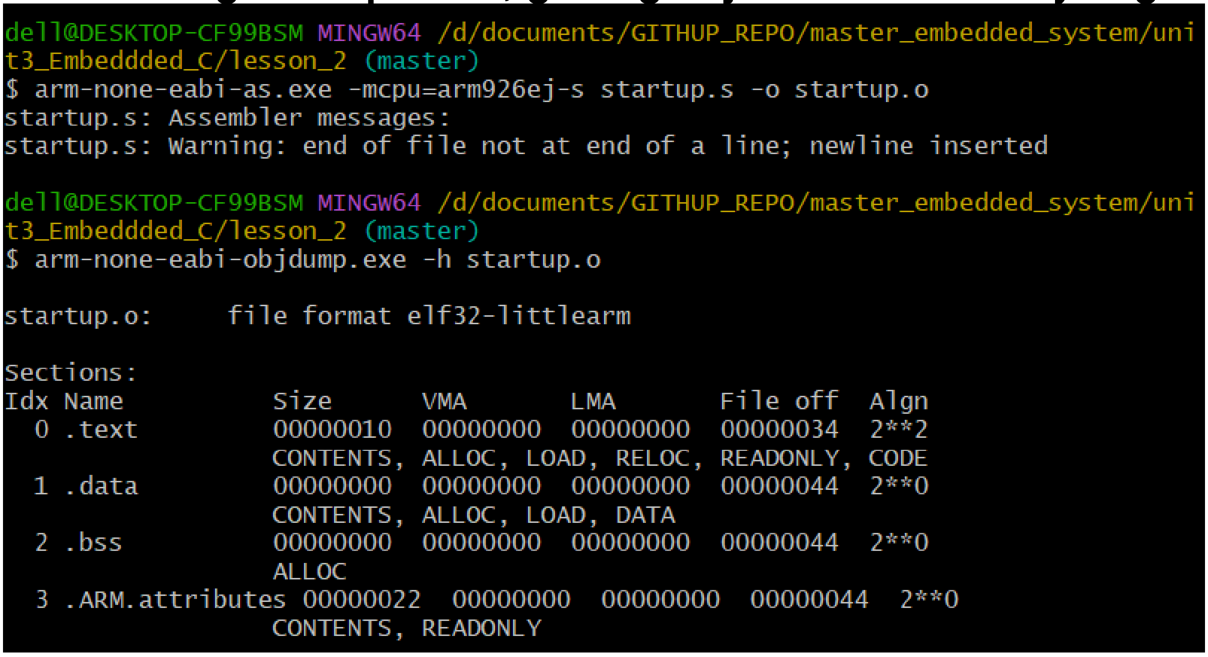


```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . app.c -o app.o

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h app.o

app.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000018  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000012  00000000  00000000  00000114  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000126  2**0
                  CONTENTS, READONLY
```



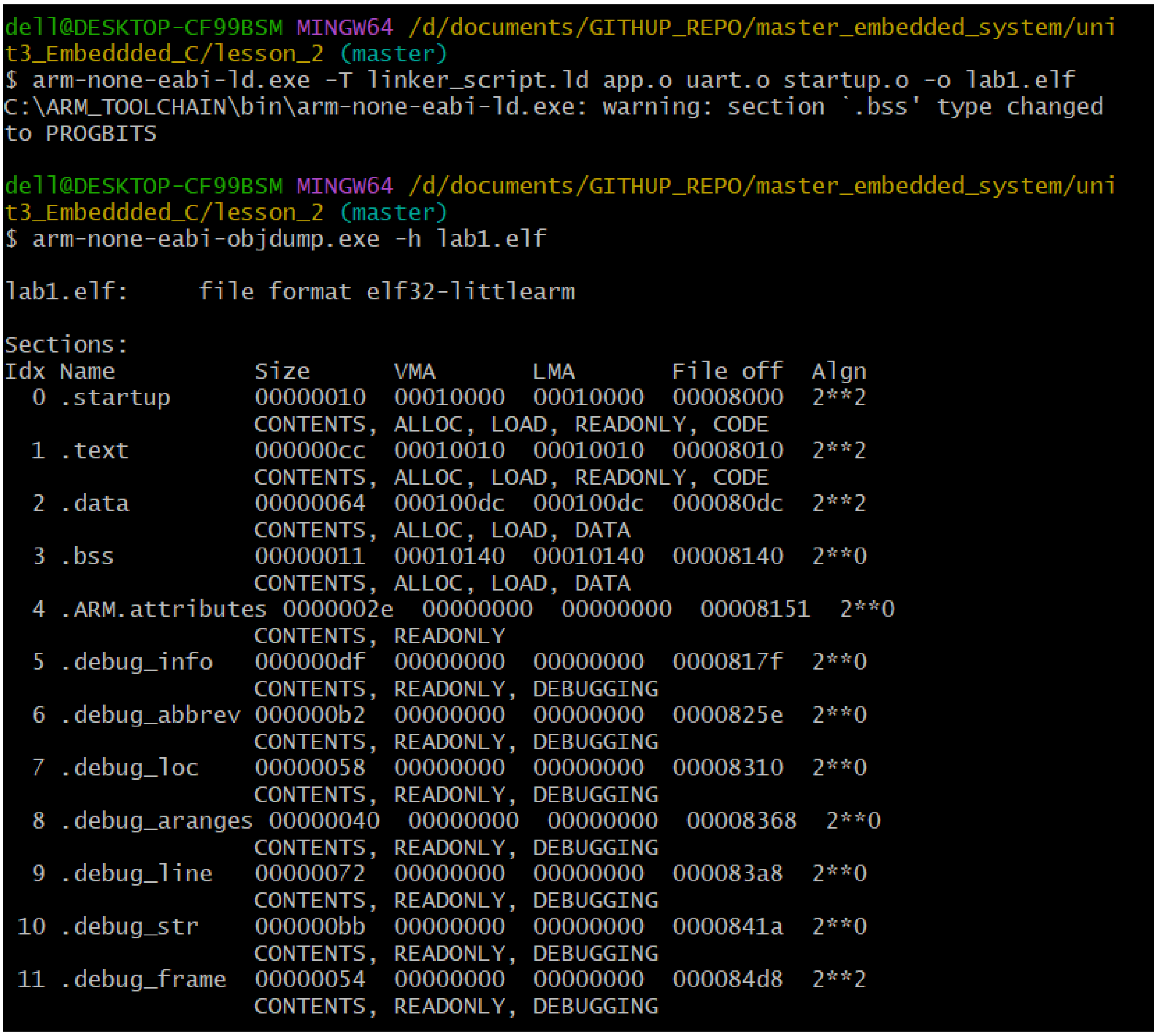
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . uart.c -o uart.o

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h uart.o

uart.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000050  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000084  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000084  2**0
                  ALLOC
  3 .comment      00000012  00000000  00000000  00000084  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000096  2**0
                  CONTENTS, READONLY
```

2. Writing startup code, getting object file and analyzing



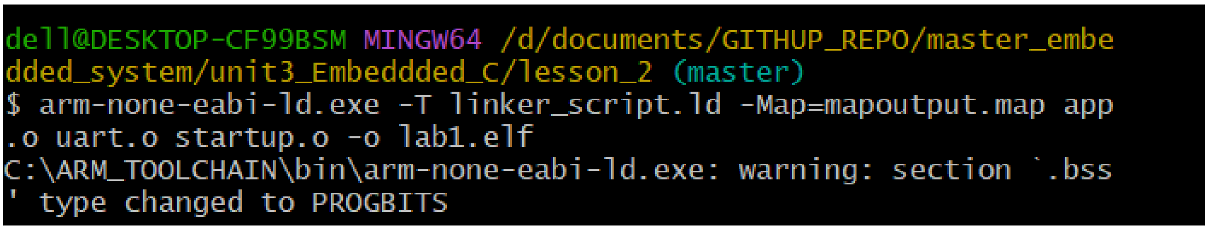
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/uni
t3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o
startup.s: Assembler messages:
startup.s: Warning: end of file not at end of a line; newline inserted

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/uni
t3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000010  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000044  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000044  2**0
                  ALLOC
  3 .ARM.attributes 00000022  00000000  00000000  00000044  2**0
                  CONTENTS, READONLY
```

3. Writing linker script, linking all object, getting elf file and analyzing



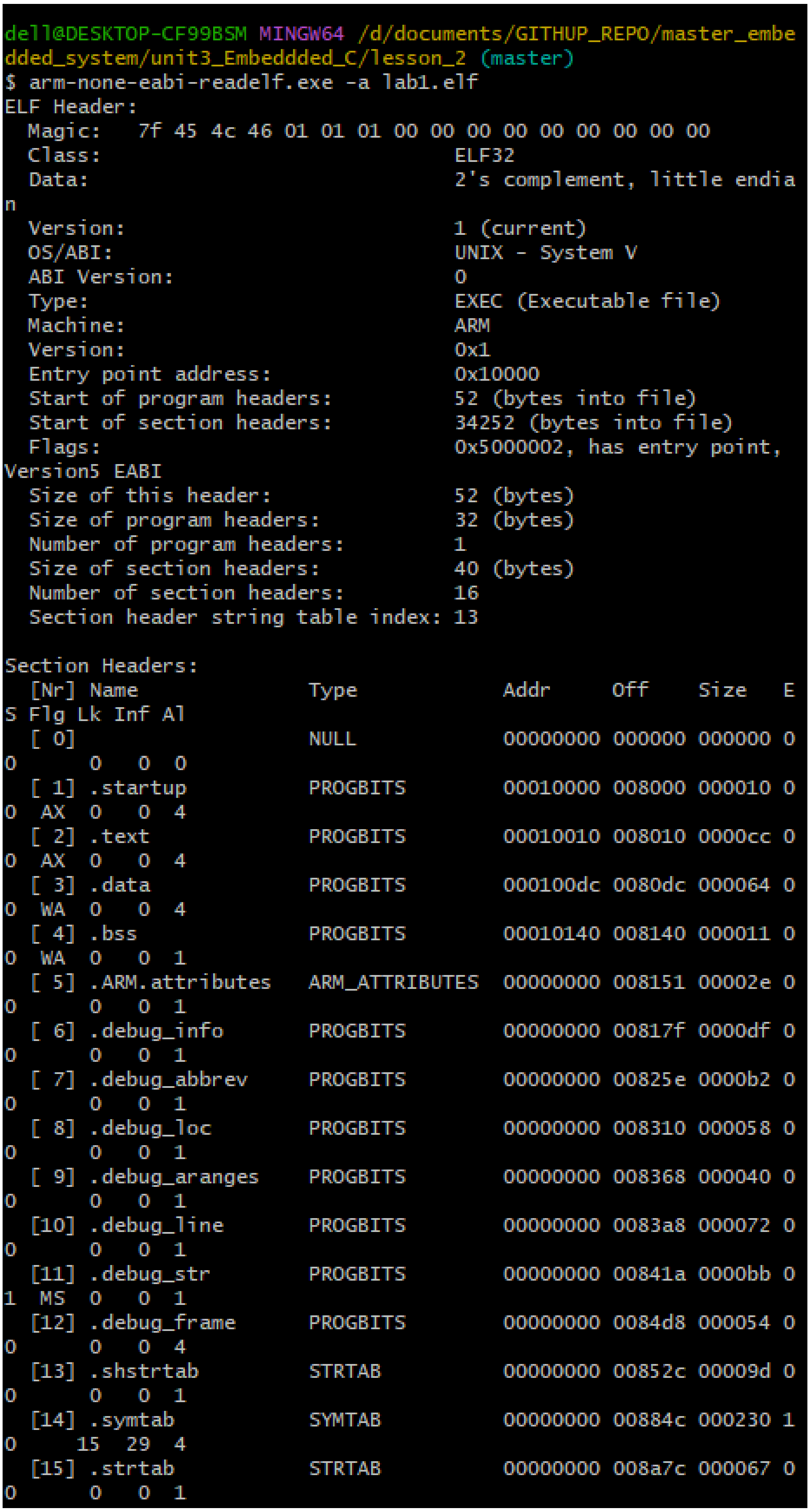
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/uni
t3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-ld.exe -T linker_script.ld app.o uart.o startup.o -o lab1.elf
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-ld.exe: warning: section `.bss' type changed
to PROGBITS

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedded_system/uni
t3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objdump.exe -h lab1.elf

lab1.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      00000010  00010000  00010000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         000000cc  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  000100dc  000100dc  000080dc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .bss          00000011  00010140  00010140  00008140  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  4 .ARM.attributes 0000002e  00000000  00000000  00008151  2**0
                  CONTENTS, READONLY
  5 .debug_info   000000df  00000000  00000000  0000817f  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev 000000b2  00000000  00000000  0000825e  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_loc    00000058  00000000  00000000  00008310  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000040  00000000  00000000  00008368  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_line   00000072  00000000  00000000  000083a8  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    000000bb  00000000  00000000  0000841a  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_frame  00000054  00000000  00000000  000084d8  2**2
                  CONTENTS, READONLY, DEBUGGING
```

3.1 We can get .map file



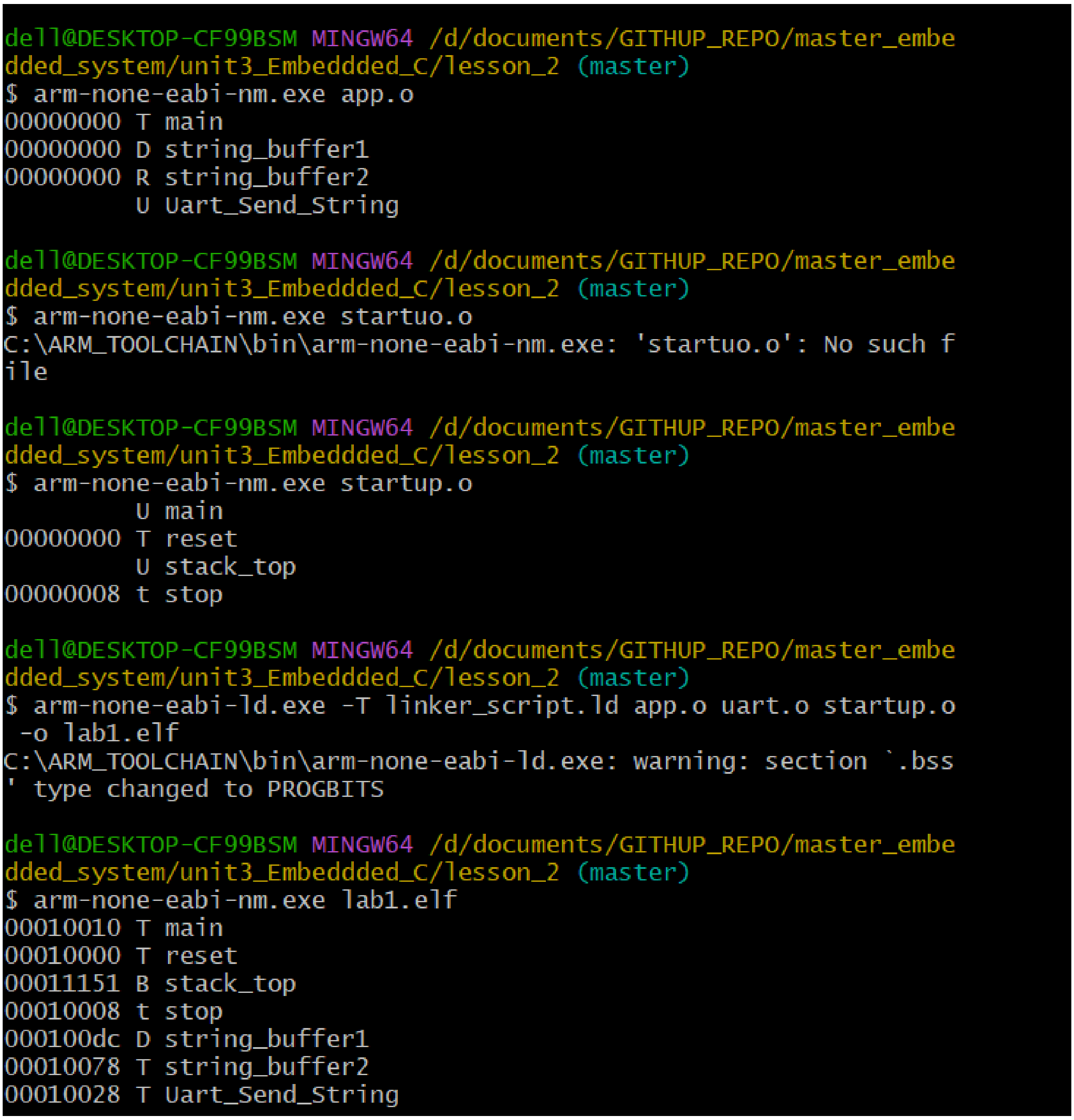
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-ld.exe -T linker_script.ld -Map=mapoutput.map app
.o uart.o startup.o -o lab1.elf
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-ld.exe: warning: section `.bss
' type changed to PROGBITS
```

3.2 We can use readelf.exe to make sure about the entry point at the address



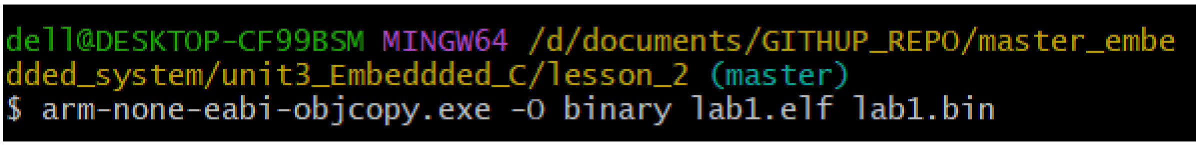
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-readelf.exe -a lab1.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endia
n
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x10000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          34252 (bytes into file)
  Flags:                             0x5000002, has entry point,
Version5 EABI
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         1
  Size of section headers:           40 (bytes)
  Number of section headers:         16
  Section header string table index: 13

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   E
S Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 0
0      0   0  0
  [ 1] .startup          PROGBITS        00010000 008000 000010 0
0  AX  0   0  4
  [ 2] .text             PROGBITS        00010010 008010 0000cc 0
0  AX  0   0  4
  [ 3] .data             PROGBITS        000100dc 0080dc 000064 0
0  WA  0   0  4
  [ 4] .bss              PROGBITS        00010140 008140 000011 0
0  WA  0   0  1
  [ 5] .ARM.attributes   ARM_ATTRIBUTES  00000000 008151 00002e 0
0      0   0  1
  [ 6] .debug_info       PROGBITS        00000000 00817f 0000df 0
0      0   0  1
  [ 7] .debug_abbrev     PROGBITS        00000000 00825e 0000b2 0
0      0   0  1
  [ 8] .debug_loc        PROGBITS        00000000 008310 000058 0
0      0   0  1
  [ 9] .debug_aranges    PROGBITS        00000000 008368 000040 0
0      0   0  1
  [10] .debug_line       PROGBITS        00000000 0083a8 000072 0
0      0   0  1
  [11] .debug_str        PROGBITS        00000000 00841a 0000bb 0
1  MS  0   0  1
  [12] .debug_frame      PROGBITS        00000000 0084d8 000054 0
0      0   0  4
  [13] .shstrtab         STRTAB          00000000 00852c 00009d 0
0      0   0  1
  [14] .symtab           SYMTAB          00000000 00884c 000230 1
0     15  29  4
  [15] .strtab           STRTAB          00000000 008a7c 000067 0
0      0   0  1
```

4. Getting the symbol table for the object files and the final elf file



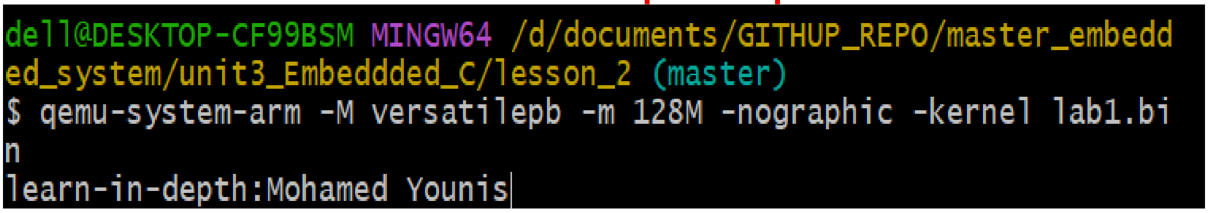
```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer1
00000000 R string_buffer2
         U Uart_Send_String

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-nm.exe startuo.o
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-nm.exe: 'startuo.o': No such f
ile

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-ld.exe -T linker_script.ld app.o uart.o startup.o
 -o lab1.elf
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-ld.exe: warning: section `.bss
' type changed to PROGBITS

dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-nm.exe lab1.elf
00010010 T main
00010000 T reset
00011151 B stack_top
00010008 t stop
000100dc D string_buffer1
00010078 T string_buffer2
00010028 T Uart_Send_String
```

5. Getting the binary file and simulating the application using QEMU



```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embe
dded_system/unit3_Embeddded_C/lesson_2 (master)
$ arm-none-eabi-objcopy.exe -O binary lab1.elf lab1.bin
```

6. Go to the qemu path



```
dell@DESKTOP-CF99BSM MINGW64 /d/documents/GITHUP_REPO/master_embedd
ed_system/unit3_Embeddded_C/lesson_2 (master)
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel lab1.bi
n
learn-in-depth:Mohamed Younis
```