Assignment report Lesson 2

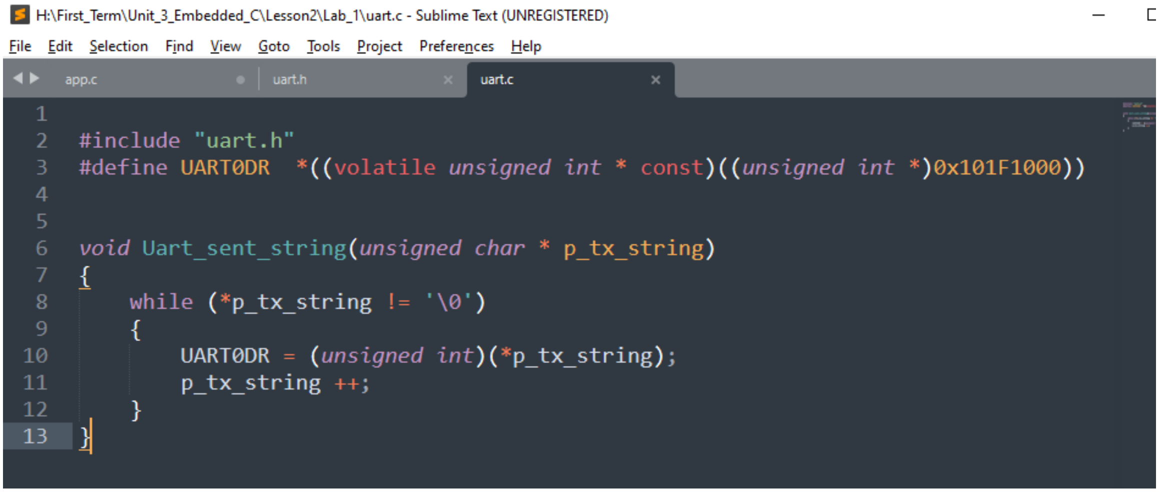
*** Qemu emulator support the versatilepb platform that contains arm926ej-s core**

*** UART0 → the first serial port that works as a terminal in adders 0x101f1000.**

*** UARTDR →used a transmit when writing in the register and receiver when reading bytes placed in 0x0**

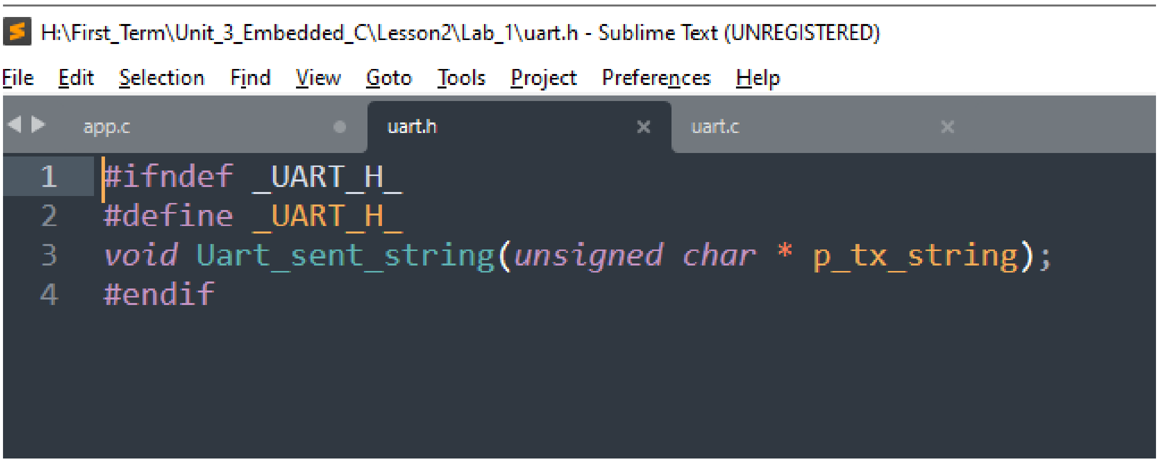
1- C Cood (app.c , uart.c , uart.h)

* uart.c →


H:\First_Term\Unit_3_Embedded_C\Lesson2\Lab_1\uart.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.c
uart.h
uart.c

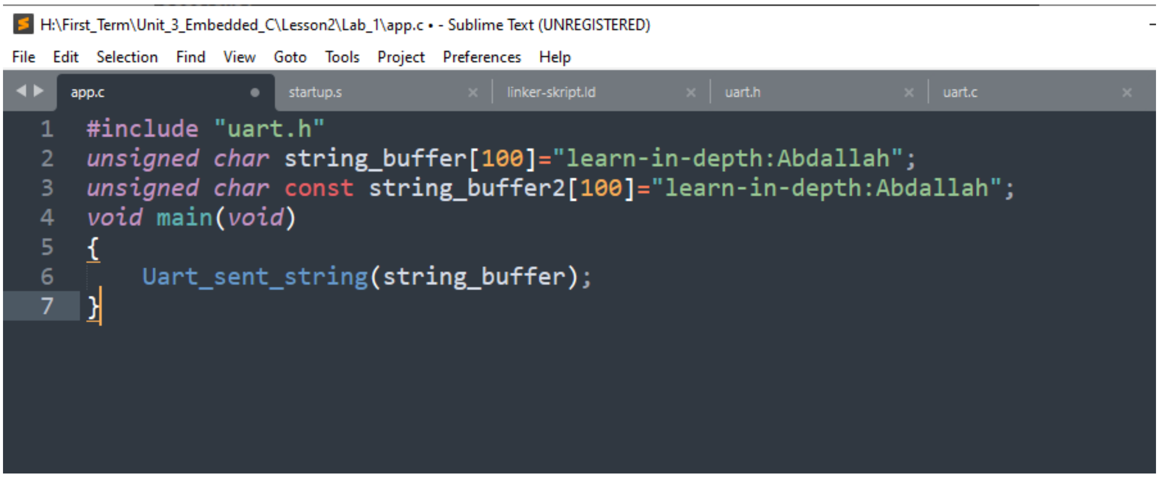
```c

#include "uart.h"
#define UART0DR  *((volatile unsigned int * const)((unsigned int *)0x101F1000))


void Uart_sent_string(unsigned char * p_tx_string)
{
    while (*p_tx_string != '\0')
    {
        UART0DR = (unsigned int)(*p_tx_string);
        p_tx_string ++;
    }
}
```

* uart.h →


H:\First_Term\Unit_3_Embedded_C\Lesson2\Lab_1\uart.h - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.c
uart.h
uart.c

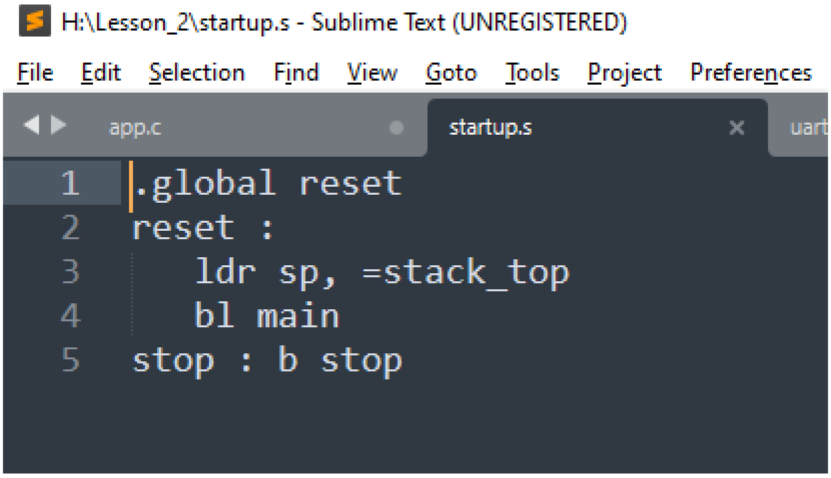
```c
#ifndef _UART_H_
#define _UART_H_
void Uart_sent_string(unsigned char * p_tx_string);
#endif
```

* app.c →


H:\First_Term\Unit_3_Embedded_C\Lesson2\Lab_1\app.c • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.c
startup.s
linker-skript.ld
uart.h
uart.c

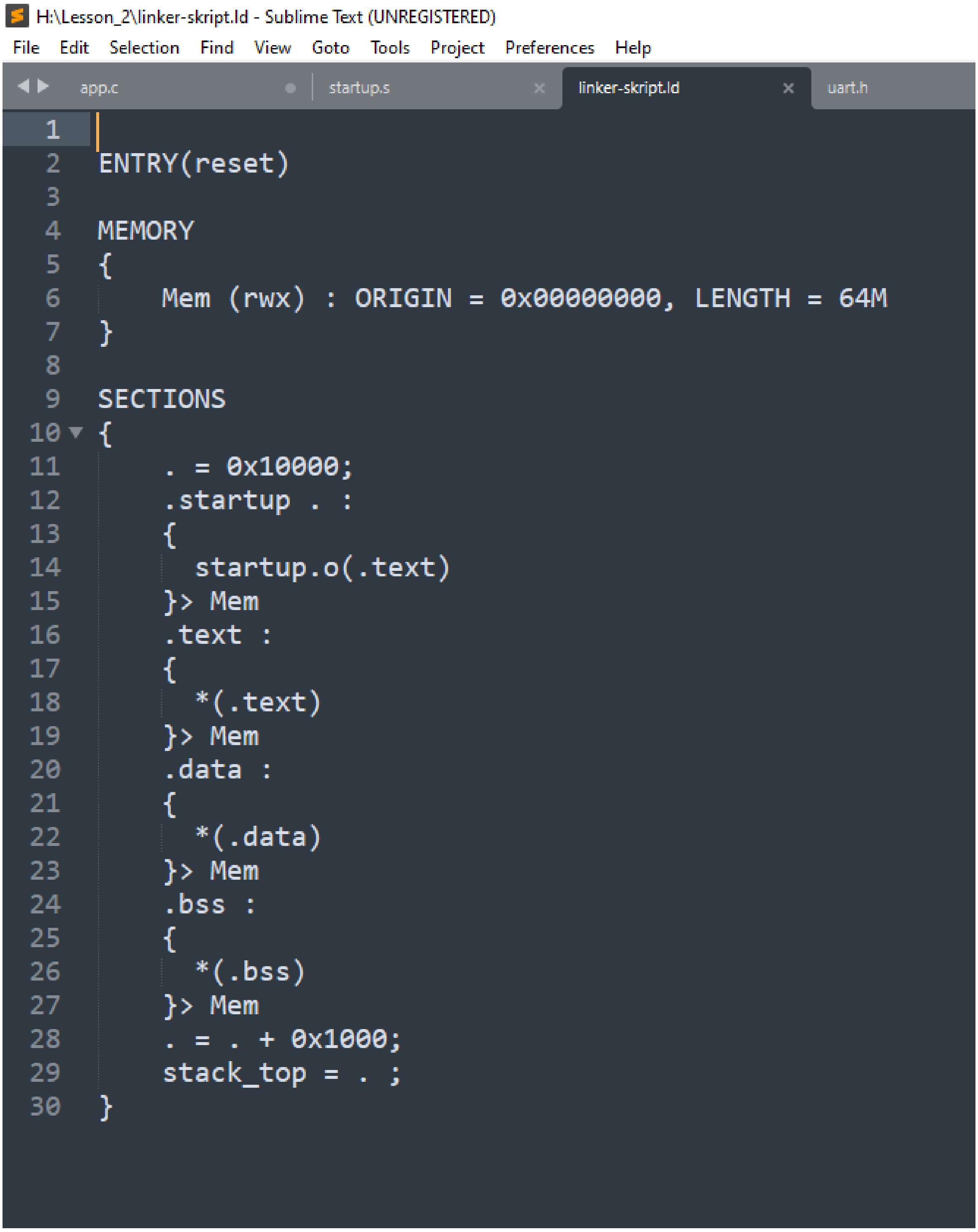
```c
#include "uart.h"
unsigned char string_buffer[100]="learn-in-depth:Abdallah";
unsigned char const string_buffer2[100]="learn-in-depth:Abdallah";
void main(void)
{
    Uart_sent_string(string_buffer);
}
```

2- startup


H:\Lesson_2\startup.s - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences
app.c
startup.s
uart

```
.global reset
reset :
    ldr sp, =stack_top
    bl main
stop : b stop
```

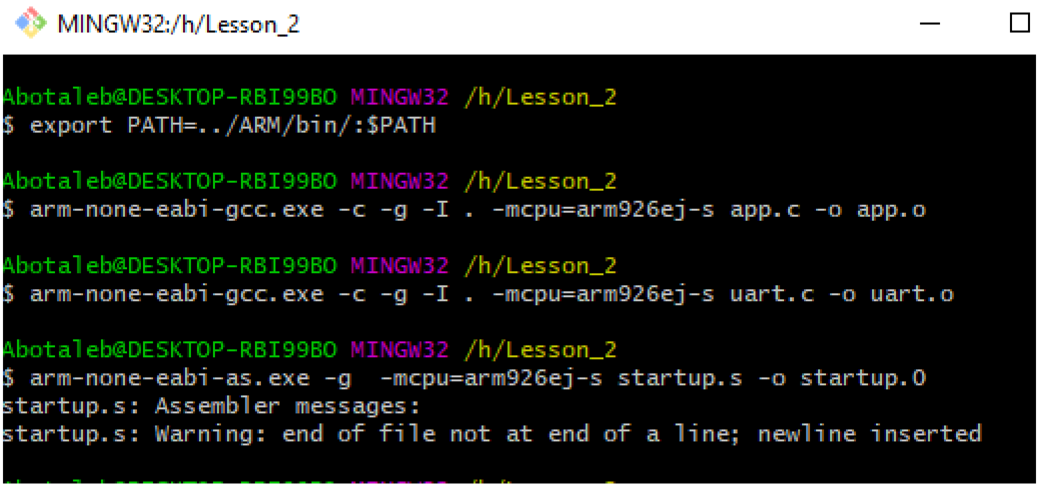
3- linker script



H:\Lesson_2\linker-skript.ld - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.c
startup.s
linker-skript.ld
uart.h

```

ENTRY(reset)

MEMORY
{
	Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
}

SECTIONS
{
	. = 0x10000;
	.startup . :
	{
		startup.o(.text)
	}> Mem
	.text :
	{
		*(.text)
	}> Mem
	.data :
	{
		*(.data)
	}> Mem
	.bss :
	{
		*(.bss)
	}> Mem
	. = . + 0x1000;
	stack_top = . ;
}
```

4- object-files (app.o , uart.o ,startup.o)

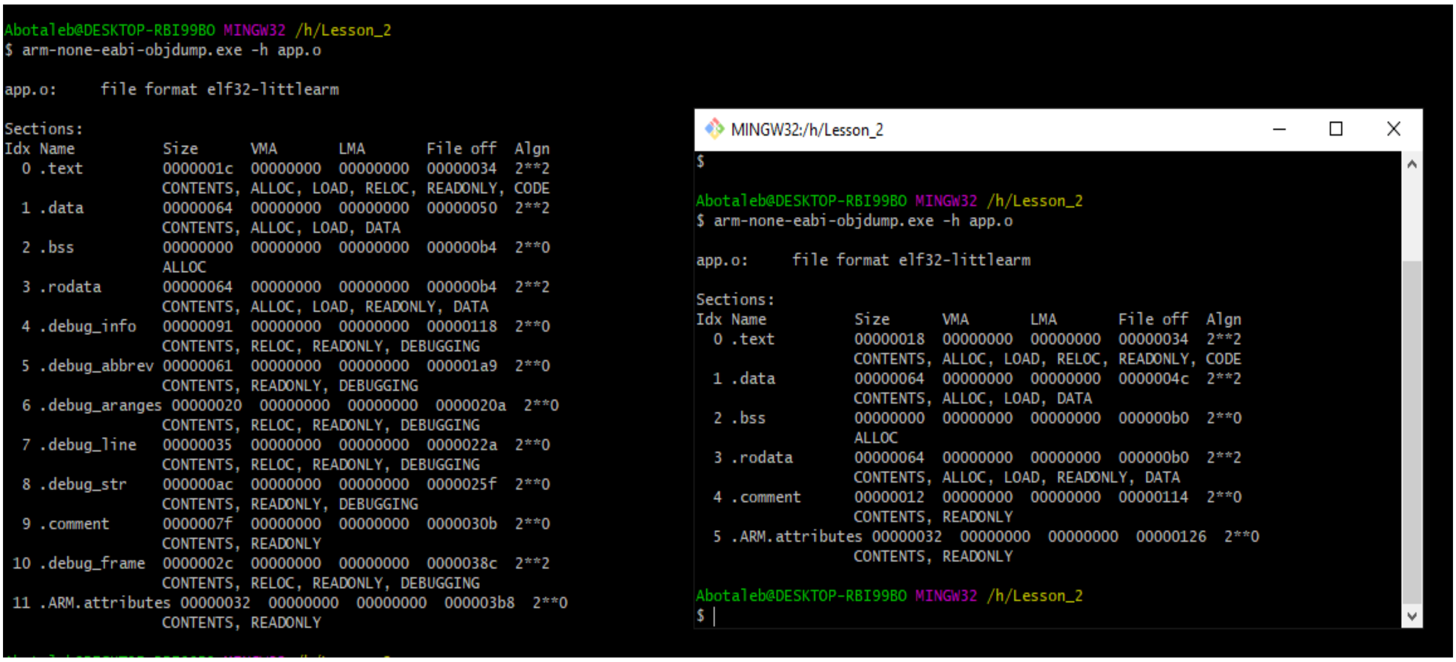


```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ export PATH=../ARM/bin/:$PATH

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s app.c -o app.o

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s uart.c -o uart.o

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-as.exe -g  -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
```

5-bin utilities(objdump[-h→section headers , -D → disassemble])

***app.o with debug**

*** app.o without debug**

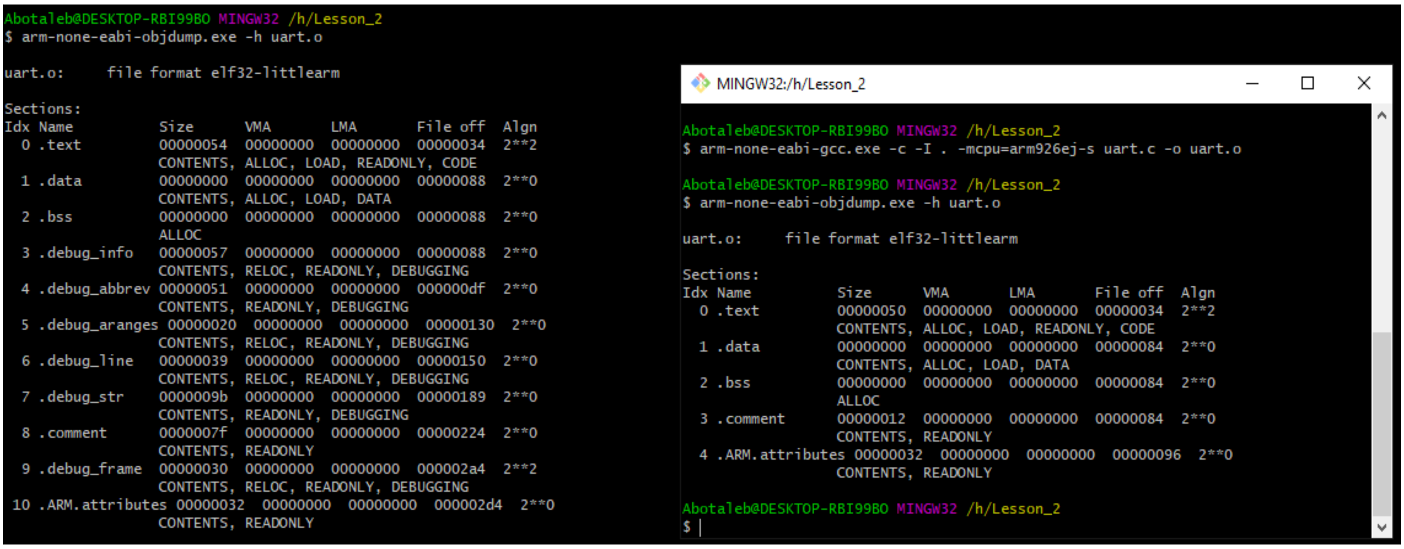


```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-objdump.exe -h app.o

app.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         0000001c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  00000050  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b4  2**0
                  ALLOC
  3 .rodata       00000064  00000000  00000000  000000b4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .debug_info   00000091  00000000  00000000  00000118  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 00000061  00000000  00000000  000001a9  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000020a  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000035  00000000  00000000  0000022a  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000000ac  00000000  00000000  0000025f  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  0000030b  2**0
                  CONTENTS, READONLY
 10 .debug_frame  0000002c  00000000  00000000  0000038c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000032  00000000  00000000  000003b8  2**0
                  CONTENTS, READONLY
```

```
$

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ 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

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ |
```

*** uart.o with debug**

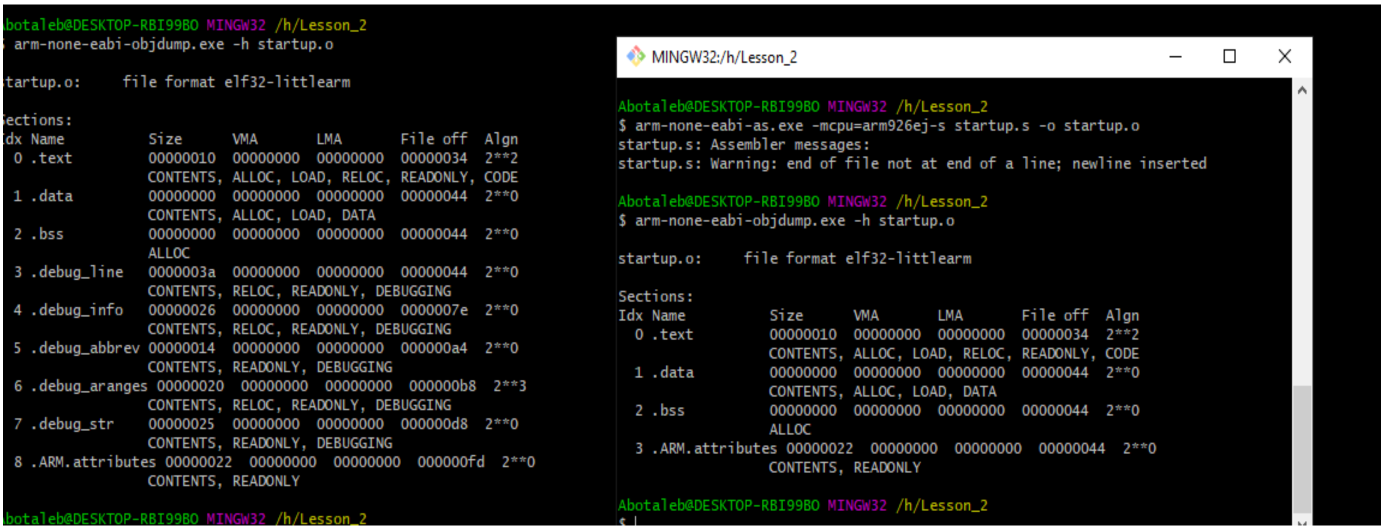
*** uart.o without debug**



```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ 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         00000054  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000088  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000088  2**0
                  ALLOC
  3 .debug_info   00000057  00000000  00000000  00000088  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000051  00000000  00000000  000000df  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000020  00000000  00000000  00000130  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_line   00000039  00000000  00000000  00000150  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_str    0000009b  00000000  00000000  00000189  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  00000224  2**0
                  CONTENTS, READONLY
  9 .debug_frame  00000030  00000000  00000000  000002a4  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  000002d4  2**0
                  CONTENTS, READONLY
```

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ 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

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ |
```

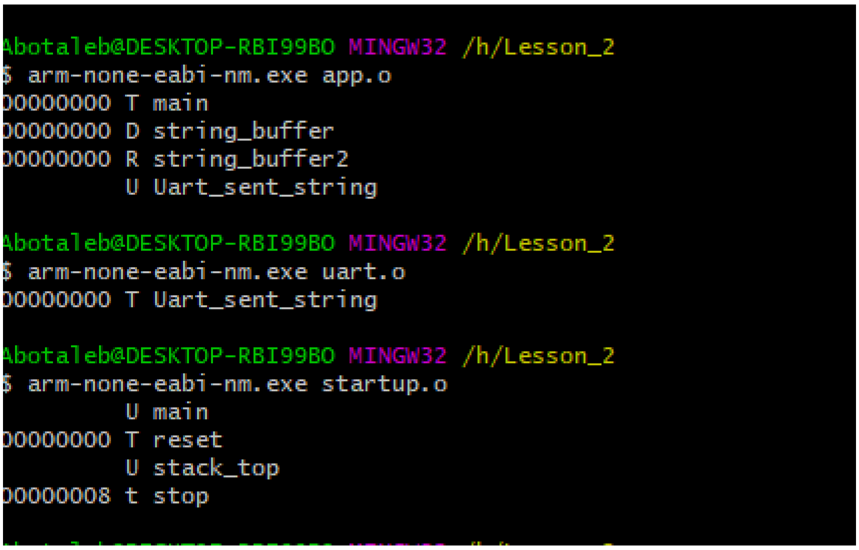
*** startup.o with debug** *** startup.o without debug**



```
botaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
 arm-none-eabi-objdump.exe -h startup.o

tartup.o:     file format elf32-littlearm

ections:
dx 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 .debug_line   0000003a  00000000  00000000  00000044  2**0
                 CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_info   00000026  00000000  00000000  0000007e  2**0
                 CONTENTS, RELOC, READONLY, DEBUGGING
 5 .debug_abbrev 00000014  00000000  00000000  000000a4  2**0
                 CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020  00000000  00000000  000000b8  2**3
                 CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_str    00000025  00000000  00000000  000000d8  2**0
                 CONTENTS, READONLY, DEBUGGING
 8 .ARM.attributes 00000022  00000000  00000000  000000fd  2**0
                 CONTENTS, READONLY

botaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
```

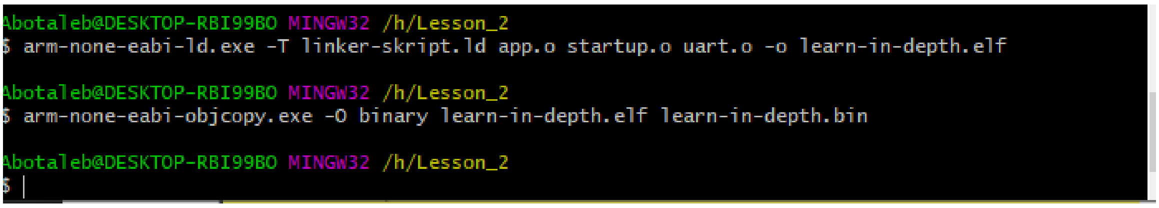
```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o
startup.s: Assembler messages:
startup.s: Warning: end of file not at end of a line; newline inserted

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ 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

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ |
```

6- symbol table (app.o , uart.o , startup.o)



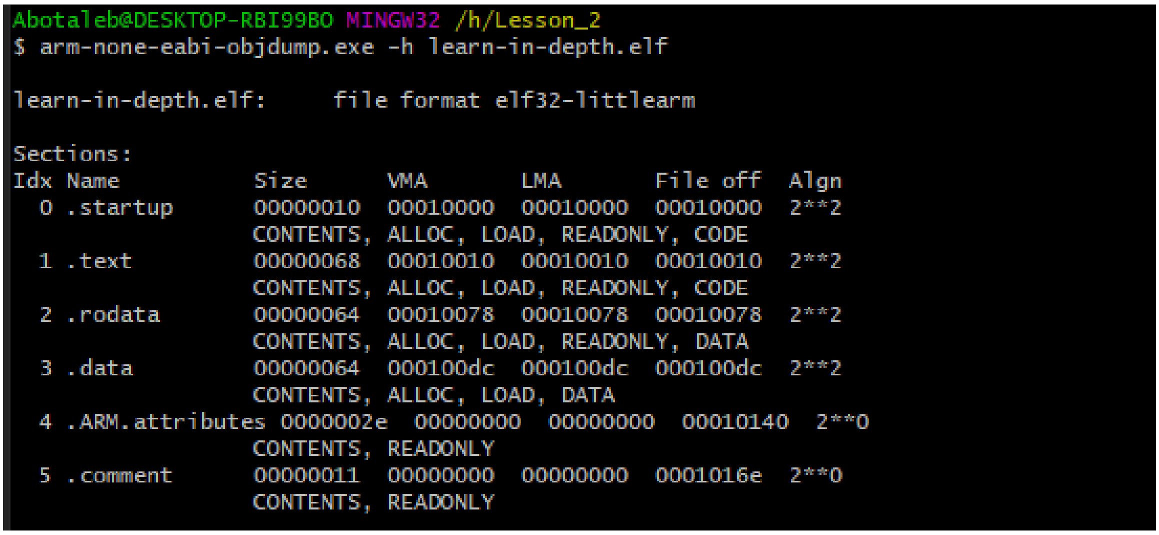
```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
00000000 R string_buffer2
         U Uart_sent_string

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-nm.exe uart.o
00000000 T Uart_sent_string

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

7- linking and getting the binary file (learn-in-depth.bin)



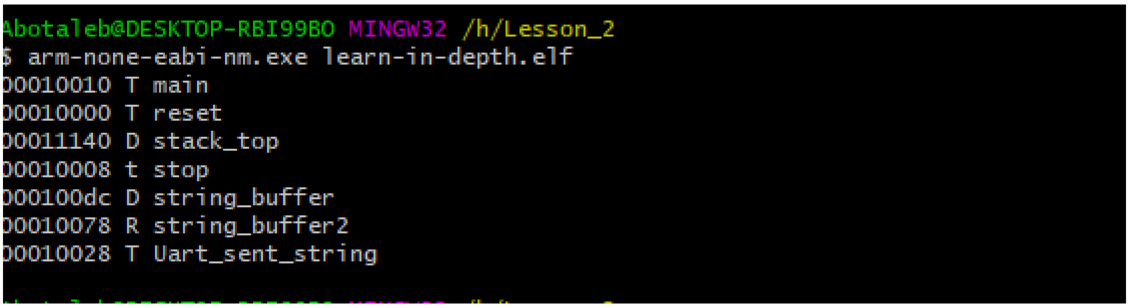
```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-ld.exe -T linker-skript.ld app.o startup.o uart.o -o learn-in-depth.elf

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin

Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ |
```

8-sections of learn-in-depth.elf



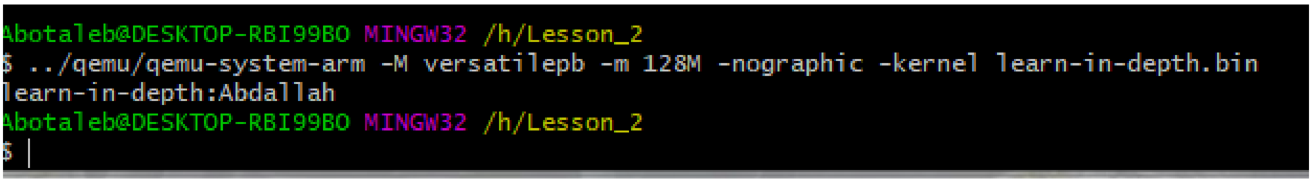
```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

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

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      00000010  00010000  00010000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000068  00010010  00010010  00010010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .rodata       00000064  00010078  00010078  00010078  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .data         00000064  000100dc  000100dc  000100dc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  4 .ARM.attributes 0000002e  00000000  00000000  00010140  2**0
                  CONTENTS, READONLY
  5 .comment      00000011  00000000  00000000  0001016e  2**0
                  CONTENTS, READONLY
```

9-symbol table for learn-in-depth.elf



```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
00011140 D stack_top
00010008 t stop
000100dc D string_buffer
00010078 R string_buffer2
00010028 T Uart_sent_string
```

10- run the program in the Qemu simulator



```
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ ../qemu/qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn-in-depth:Abdallah
Abotaleb@DESKTOP-RBI99BO MINGW32 /h/Lesson_2
$ |
```