



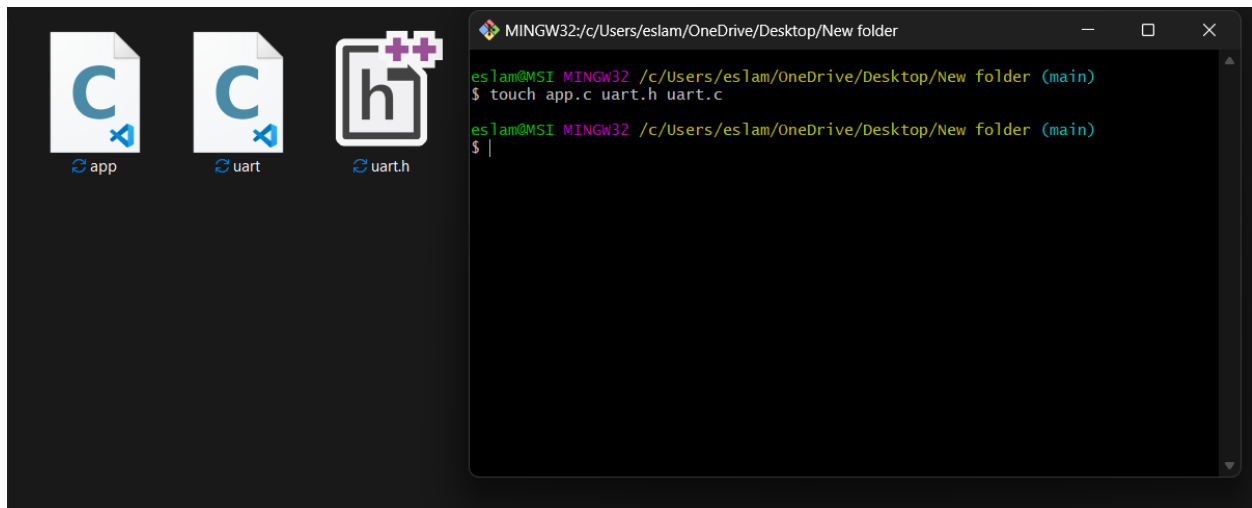
REPORT FOR LAB OF UNIT 3

LESSON 2



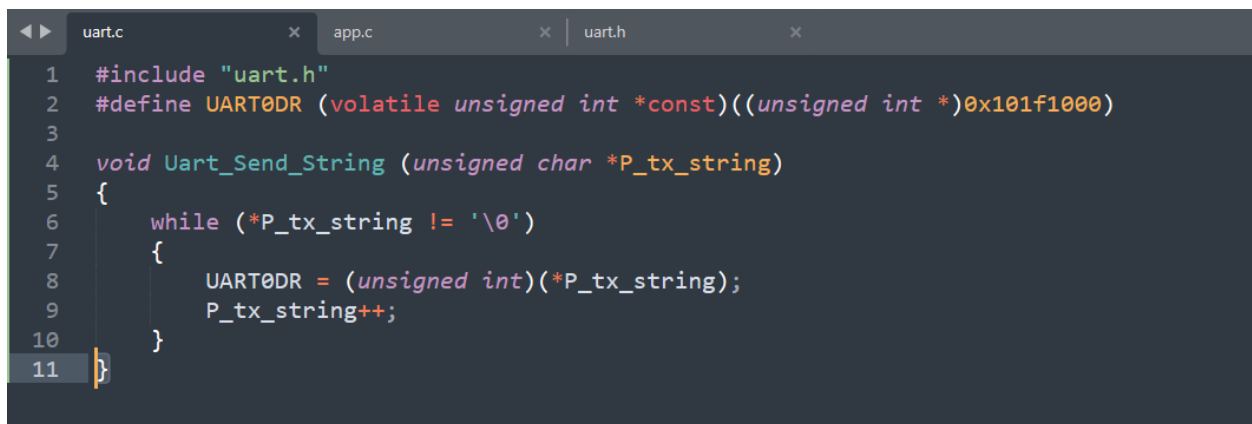
Name : Eslam Mostafa Mohamed

Using command at git Bash to create files :



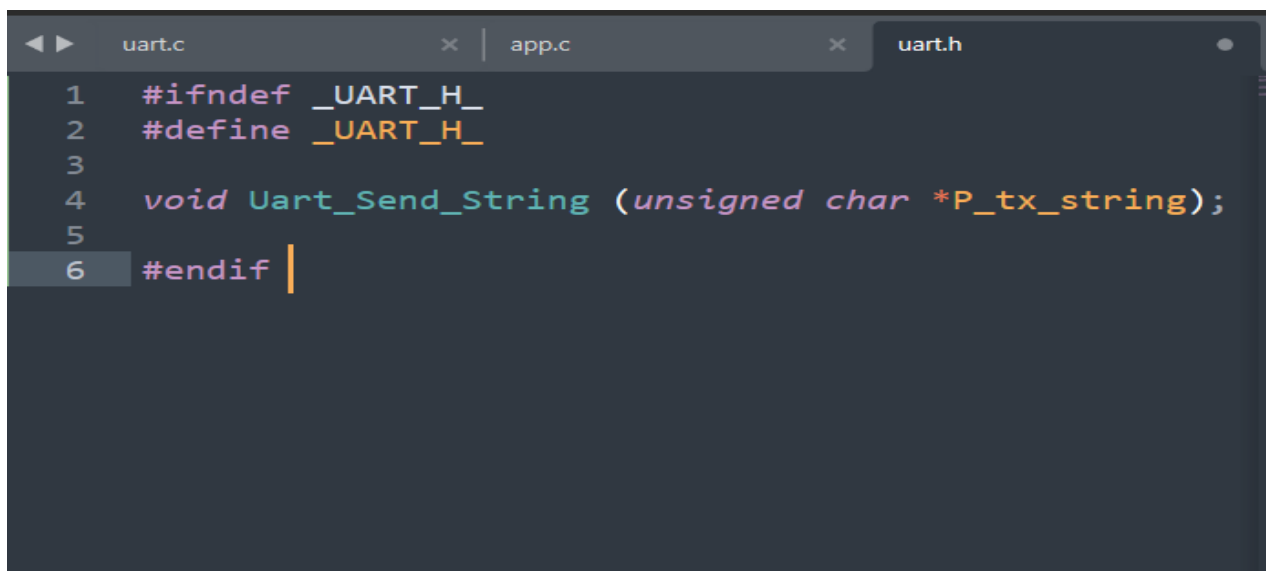
```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ touch app.c uart.h uart.c
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ |
```

Uart.c file :



```
1 #include "uart.h"
2 #define UART0DR (volatile unsigned int *const)((unsigned int *)0x101f1000)
3
4 void Uart_Send_String (unsigned char *P_tx_string)
5 {
6     while (*P_tx_string != '\0')
7     {
8         UART0DR = (unsigned int)(*P_tx_string);
9         P_tx_string++;
10    }
11 }
```

Uart.h file :



```
1 #ifndef _UART_H_
2 #define _UART_H_
3
4 void Uart_Send_String (unsigned char *P_tx_string);
5
6 #endif
```

App.c file :

```
1  #include "uart.h"
2
3  unsigned char string_buffer[100]="learn-in-depth:<Eslam>";
4  unsigned char const string_buffer_2[100]="to create a rodata section";
5
6  void main (void)
7  {
8      Uart_Send_String (string_buffer);
9  }
```

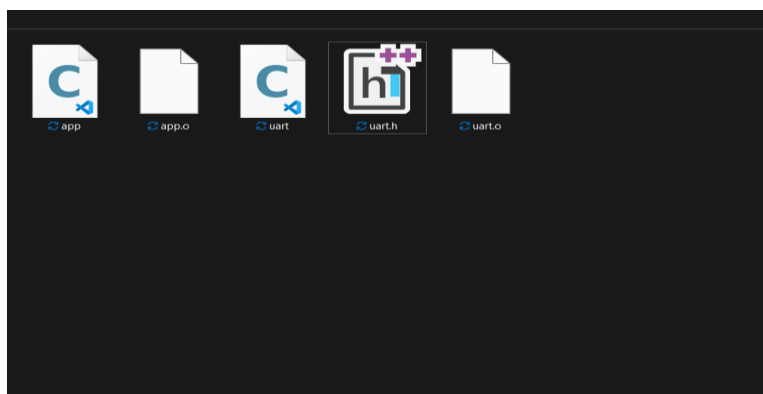
Creating object files with debugging Information :

```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s app.c -o app.o

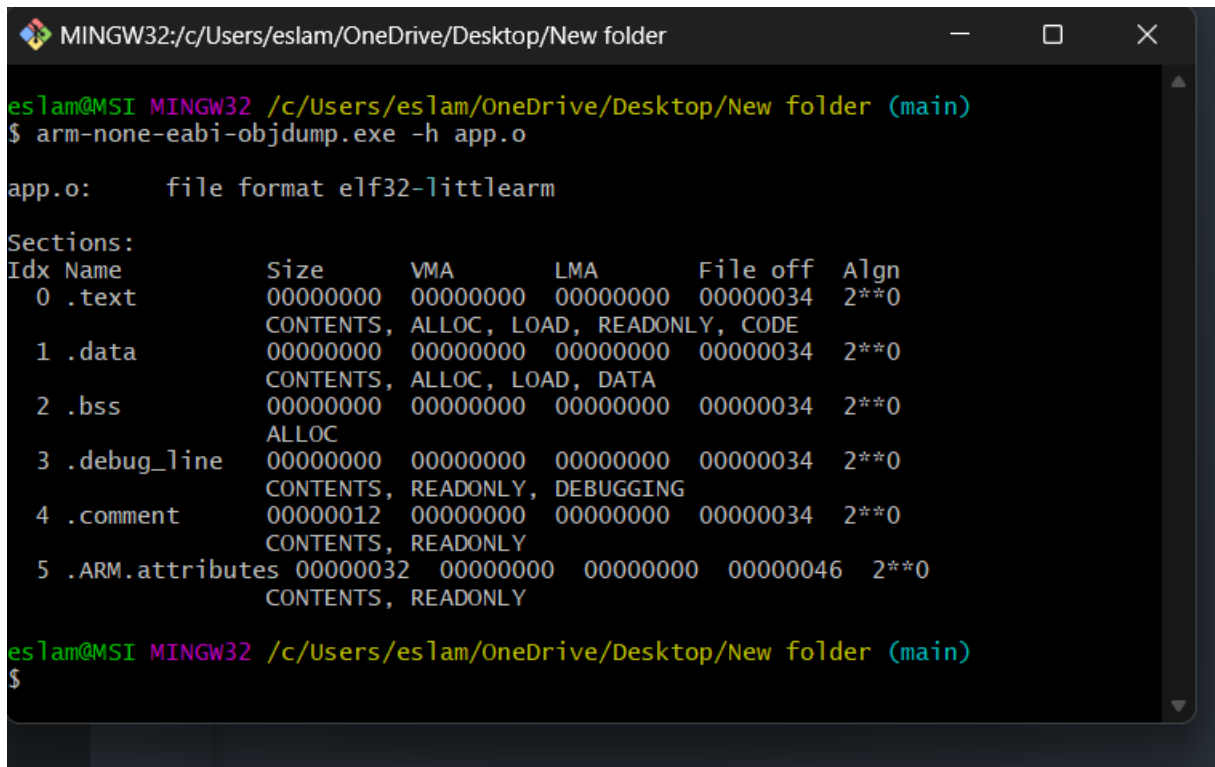
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s uart.c -o uart.o

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ ls
1.png 2.png 3.png 4.png app.c app.o uart.c uart.h uart.o

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ |
```



To watch obj file section :



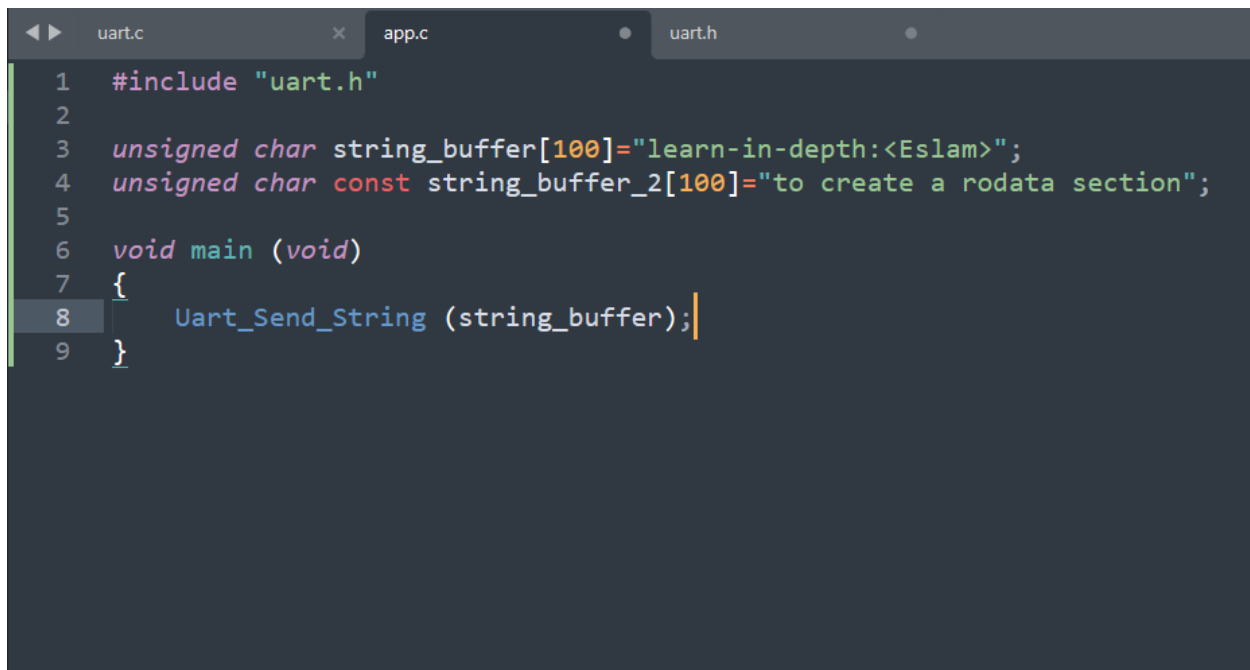
```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ arm-none-eabi-objdump.exe -h app.o

app.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00000000  00000000  00000000  00000034  2**0
CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data           00000000  00000000  00000000  00000034  2**0
CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  00000034  2**0
ALLOC
  3 .debug_line     00000000  00000000  00000000  00000034  2**0
CONTENTS, READONLY, DEBUGGING
  4 .comment        00000012  00000000  00000000  00000034  2**0
CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000046  2**0
CONTENTS, READONLY

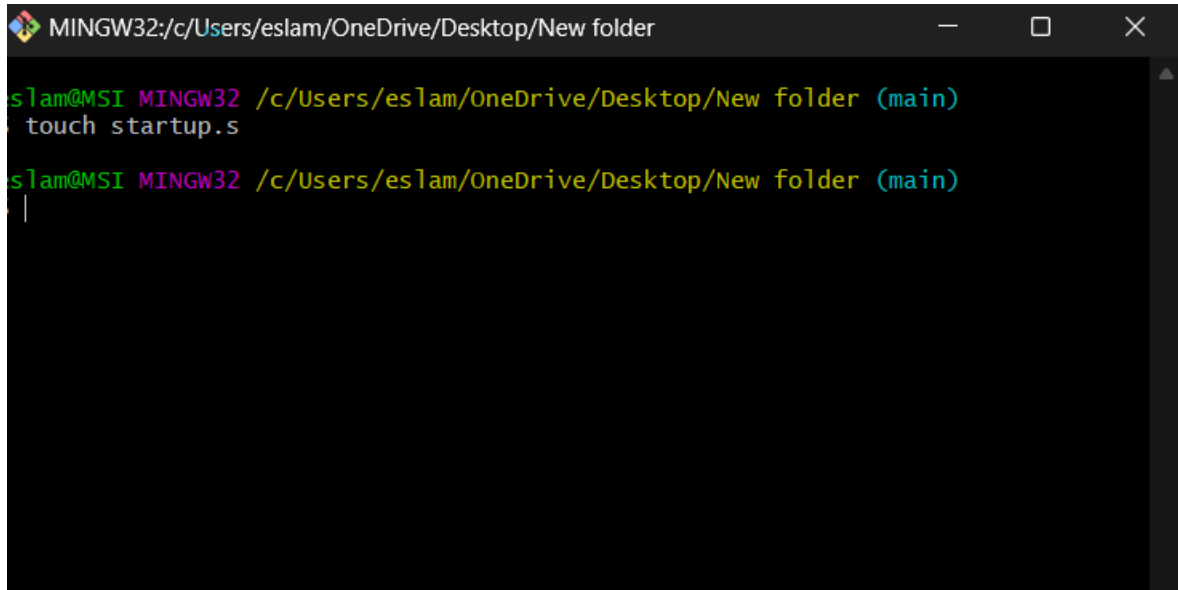
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$
```

Create .rodata section :



```
uart.c  app.c  uart.h
1  #include "uart.h"
2
3  unsigned char string_buffer[100]="learn-in-depth:<Eslam>";
4  unsigned char const string_buffer_2[100]="to create a rodata section";
5
6  void main (void)
7  {
8      Uart_Send_String (string_buffer);
9  }
```

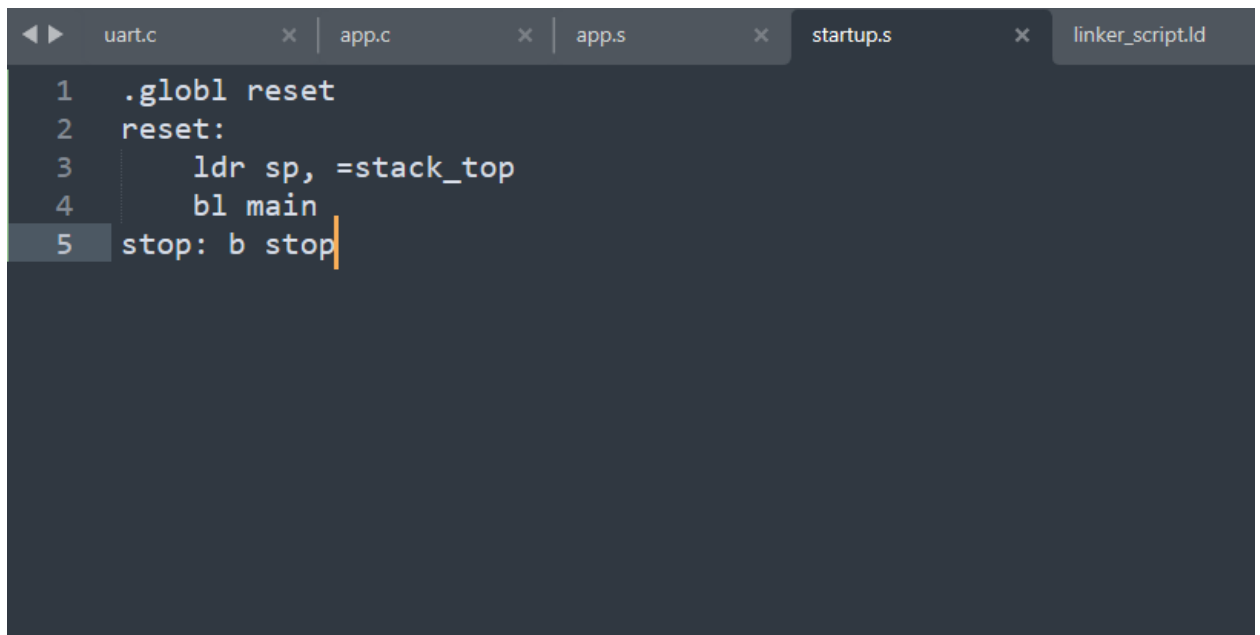
Creating startup file :



```
MINGW32:/c:/Users/eslam/OneDrive/Desktop/New folder

slam@MSI MINGW32 /c:/Users/eslam/OneDrive/Desktop/New folder (main)
$ touch startup.s

slam@MSI MINGW32 /c:/Users/eslam/OneDrive/Desktop/New folder (main)
$ |
```



```
uart.c  app.c  app.s  startup.s  linker_script.ld

1  .globl reset
2  reset:
3      ldr sp, =stack_top
4      bl main
5  stop: b stop
```

Creating .o file and watch object file sections of startup file :

```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder
eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ 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

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ 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          0000000c  00000000  00000000  00000034  2**2
             CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data          00000000  00000000  00000000  00000040  2**0
             CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  00000040  2**0
             ALLOC
  3 .ARM.attributes 00000022  00000000  00000000  00000040  2**0
             CONTENTS, READONLY

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ |
```

Creating linker script file :

```
uart.c  x  app.c  •  app.s  x  startup.s  x  linker_script.ld  x  uart.h

1 ENTRY(reset)
2 MEMORY
3 {
4     Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
5 }
6 SECTIONS
7 {
8     . = 0x10000;
9     .startup . :
10    {
11        startup.o(.text)
12    }>Mem
13    .text :
14    {
15        *(.text) *(.rodata)
16    }>Mem
17    .data :
18    {
19        *(.data)
20    }>Mem
21    .bss :
22    {
23        *(.bss) *(COMMON)
24    }>Mem
25    . = . + 0x1000;
26    stack_top = .;
27 }
```

Watching symbols of app.o and uart.o before linking with relocatable addresses :

```
MINGW32:/e/Courses/Embedded System Diploma/unit 3/Assignment/Embed...
eslam@MSI MINGW32 /e/Courses/Embedded System Diploma/unit 3/Assignment/Embedded_
C/lesson_2/lab_1
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
00000000 R string_buffer_2
          U Uart_Send_String

eslam@MSI MINGW32 /e/Courses/Embedded System Diploma/unit 3/Assignment/Embedded_
C/lesson_2/lab_1
$ |
```

watching symbols in learn-in-depth.elf File :

```
MINGW32:/e/Courses/Embedded System Diploma/unit 3/Assignment/Embed...
eslam@MSI MINGW32 /e/Courses/Embedded System Diploma/unit 3/Assignment/Embedded_
C/lesson_2/lab_1
$ 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 T string_buffer_2
00010028 T Uart_Send_String

eslam@MSI MINGW32 /e/Courses/Embedded System Diploma/unit 3/Assignment/Embedded_
C/lesson_2/lab_1
$
```

Watching sections in learn-in-depth.elf file with debugging information :

```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ 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  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 .ARM.attributes 0000002e  00000000  00000000  00008140  2**0
                CONTENTS, READONLY
  4 .comment       00000011  00000000  00000000  0000816e  2**0
                CONTENTS, READONLY
  5 .debug_info     000000fa  00000000  00000000  0000817f  2**0
                CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev   000000d5  00000000  00000000  00008279  2**0
                CONTENTS, READONLY, DEBUGGING
  7 .debug_loc      00000058  00000000  00000000  0000834e  2**0
                CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges  00000040  00000000  00000000  000083a6  2**0
                CONTENTS, READONLY, DEBUGGING
  9 .debug_line     00000072  00000000  00000000  000083e6  2**0
                CONTENTS, READONLY, DEBUGGING
 10 .debug_str       0000009b  00000000  00000000  00008458  2**0
                CONTENTS, READONLY, DEBUGGING
 11 .debug_frame    00000054  00000000  00000000  000084f4  2**2
                CONTENTS, READONLY, DEBUGGING

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$
```

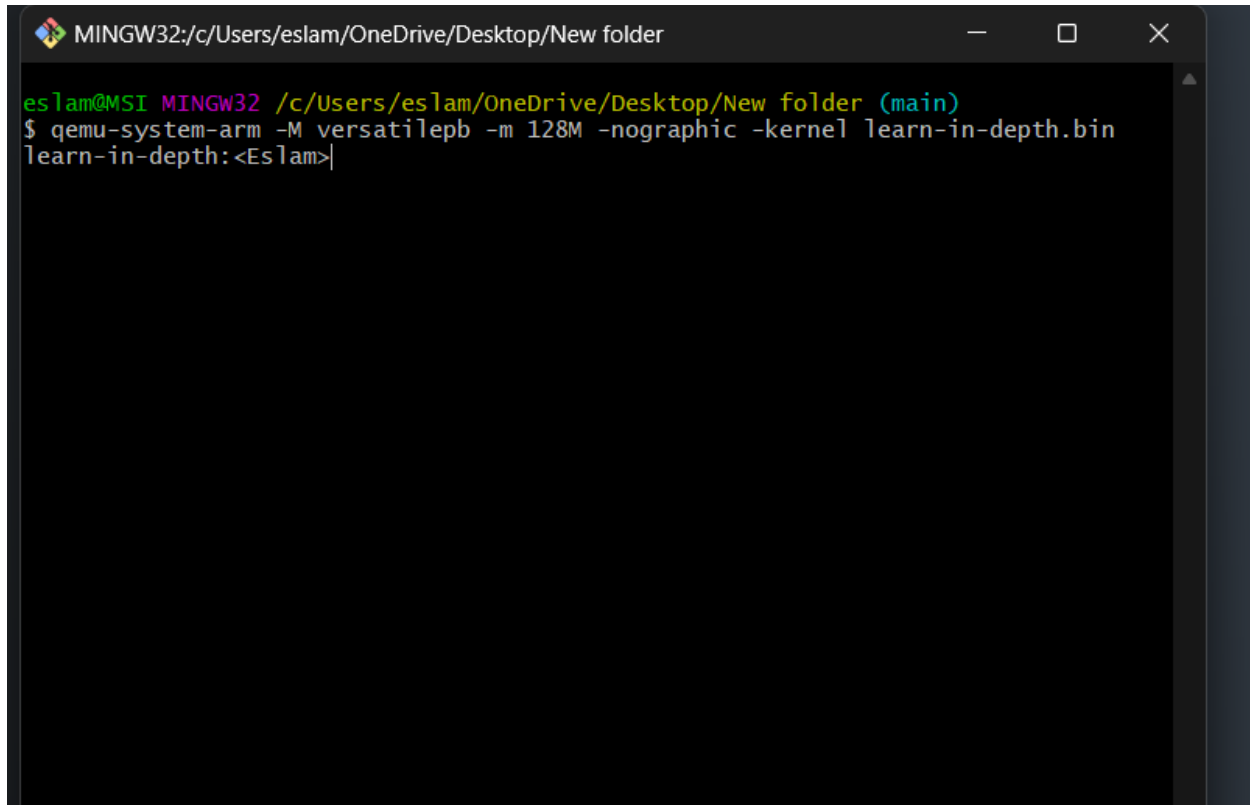
generating binary file from learn-in-depth.elf File

```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$
```


Output :



```
MINGW32:/c/Users/eslam/OneDrive/Desktop/New folder

eslam@MSI MINGW32 /c/Users/eslam/OneDrive/Desktop/New folder (main)
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn-in-depth:<Eslam>
```