Lab1
Build a Bare-metal software from scratch using terminal

- **Application:**

  Printing a string to UART of Arm versatilepb board by writing in the register UARTDT.

- **Board :**

  Arm versatilepb.

- **Processor:**

  Arm926ej-s.

- **Tool Chain:**

  Arm cross tool chain

- **Steps:**

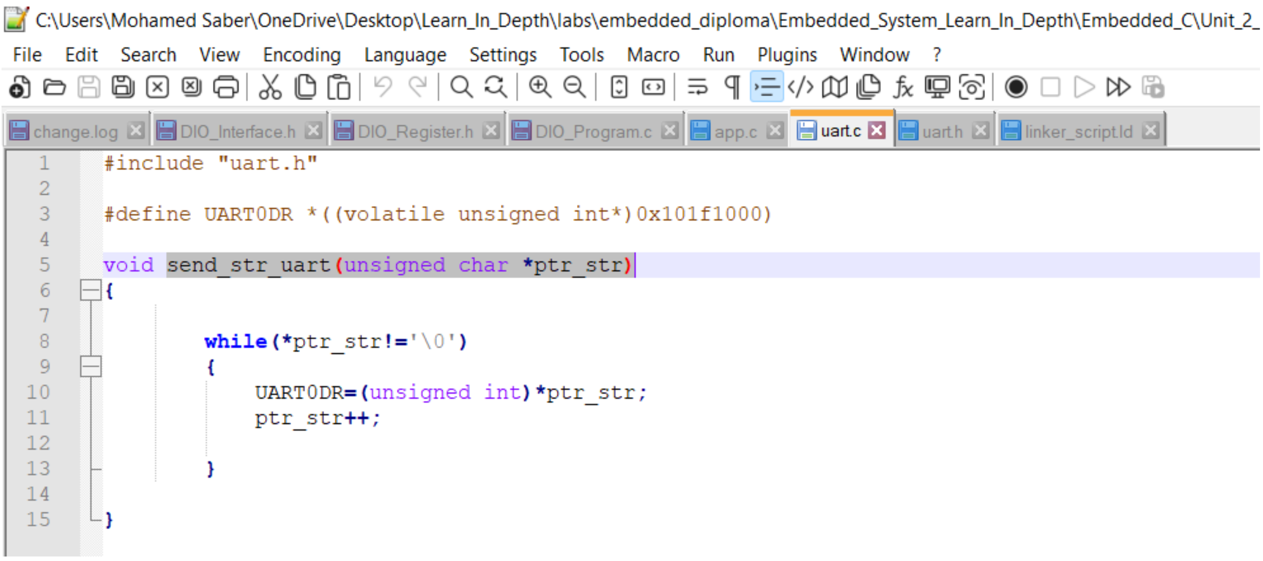
  1- Writing the C code.
  2- Writing Startup file.
  3- Writing linker script.
  4- Running the application using QEMU emulator.

By: Mohamed M.saber
Email:mohamed.m.sabir@gmail.com

C code files:

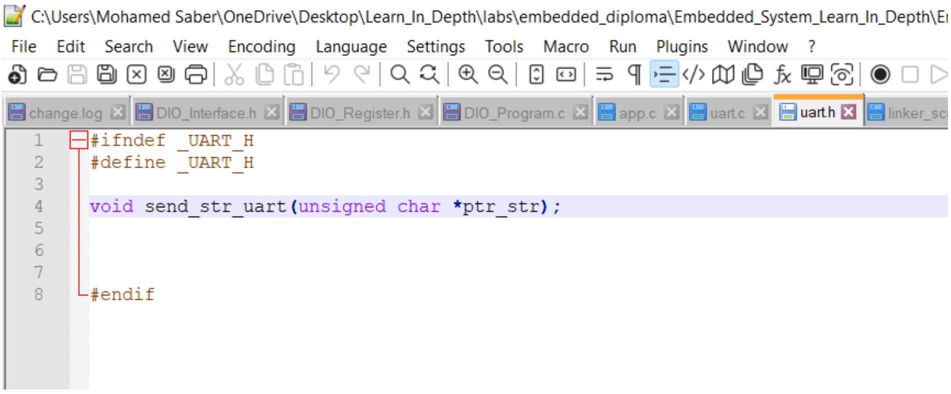
Versatilepb board has 4 UART terminals , we used UART0 The logic of c code is that we send the string which we want to print to UARTDT Register which prints anything once we write on it . For doing this we used a pointer refers to our string then we stored the value of location pointed to by the pointer , in the UARTDT Register so the character gets printed directly , after this we incremented the value of the pointer to step to the next character and do the same process. Eventually the string will be printed.

Uart.c:


C:\Users\Mohamed Saber\OneDrive\Desktop\Learn_In_Depth\labs\embedded_diploma\Embedded_System_Learn_In_Depth\Embedded_C\Unit_2_
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
DIO_Interface.h
DIO_Register.h
DIO_Program.c
app.c
uart.c
uart.h
linker_script.ld

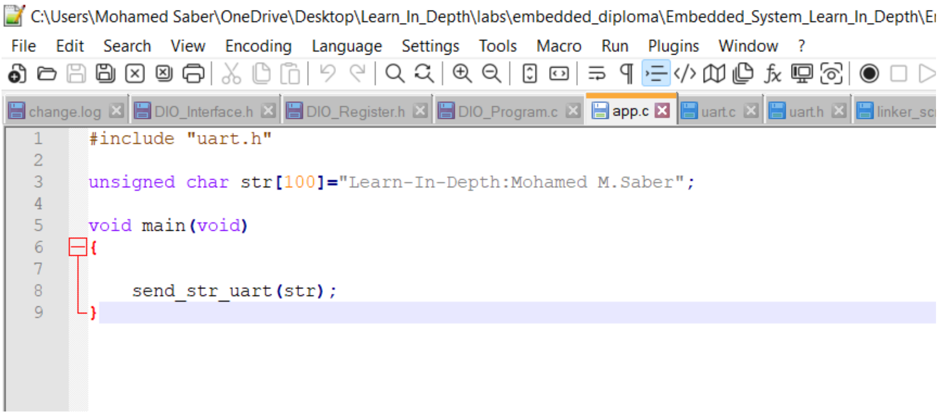
```
#include "uart.h"

#define UART0DR *((volatile unsigned int*)0x101f1000)

void send_str_uart(unsigned char *ptr_str)
{

        while(*ptr_str!='\0')
        {
            UART0DR=(unsigned int)*ptr_str;
            ptr_str++;

        }

}
```

Uart.h:


C:\Users\Mohamed Saber\OneDrive\Desktop\Learn_In_Depth\labs\embedded_diploma\Embedded_System_Learn_In_Depth\E
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
DIO_Interface.h
DIO_Register.h
DIO_Program.c
app.c
uart.c
uart.h
linker_sc

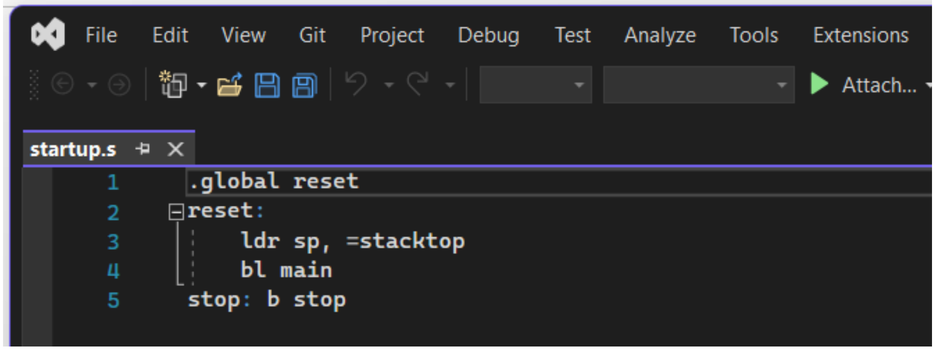
```
#ifndef _UART_H
#define _UART_H

void send_str_uart(unsigned char *ptr_str);



#endif
```

App.c:


C:\Users\Mohamed Saber\OneDrive\Desktop\Learn_In_Depth\labs\embedded_diploma\Embedded_System_Learn_In_Depth\E
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
DIO_Interface.h
DIO_Register.h
DIO_Program.c
app.c
uart.c
uart.h
linker_sc

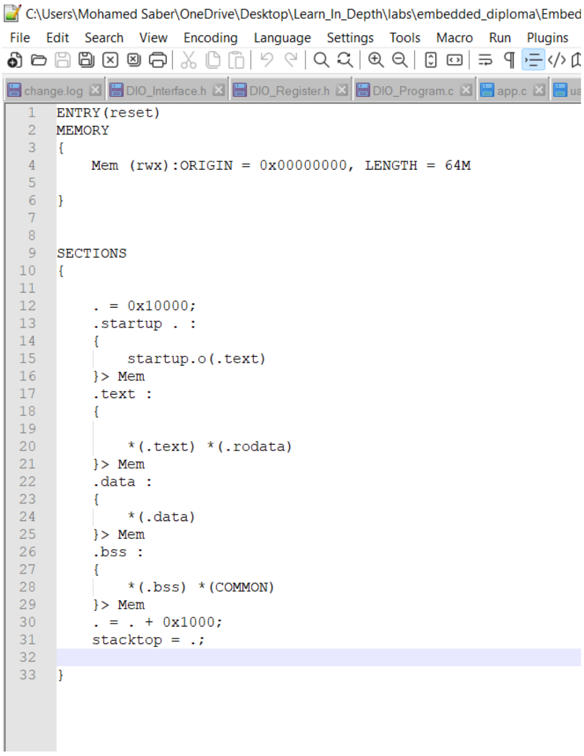
```c
#include "uart.h"

unsigned char str[100]="Learn-In-Depth:Mohamed M.Saber";

void main(void)
{

    send_str_uart(str);
}
```

Startup file:



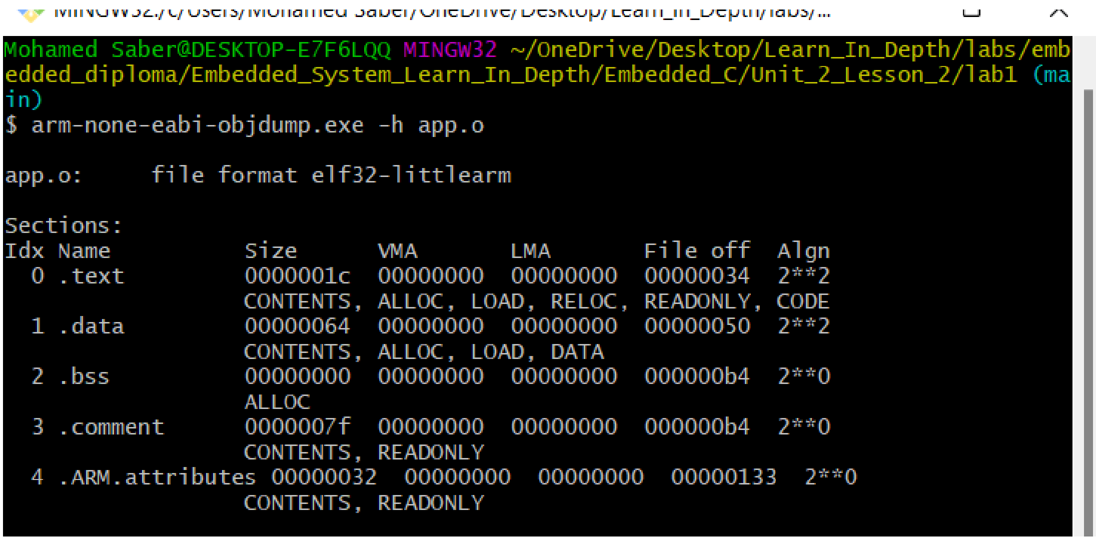
```asm
 .global reset
reset:
     ldr sp, =stacktop
     bl main
 stop: b stop
```

Linker script:


C:\Users\Mohamed Saber\OneDrive\Desktop\Learn_In_Depth\labs\embedded_diploma\Embed
File Edit Search View Encoding Language Settings Tools Macro Run Plugins
change.log
DIO_Interface.h
DIO_Register.h
DIO_Program.c
app.c
ua

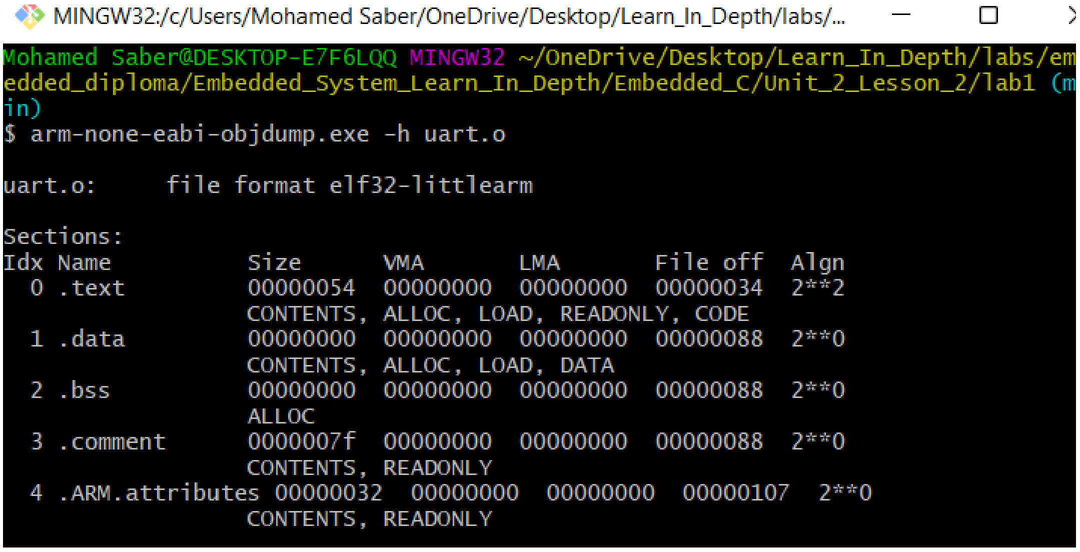
```
ENTRY(reset)
MEMORY
{
    Mem (rwx):ORIGIN = 0x00000000, LENGTH = 64M

}


SECTIONS
{

    . = 0x10000;
    .startup . :
    {
        startup.o(.text)
    }> Mem
    .text :
    {

        *(.text) *(.rodata)
    }> Mem
    .data :
    {
        *(.data)
    }> Mem
    .bss :
    {
        *(.bss) *(COMMON)
    }> Mem
    . = . + 0x1000;
    stacktop = .;

}
```

Sections of app.o:



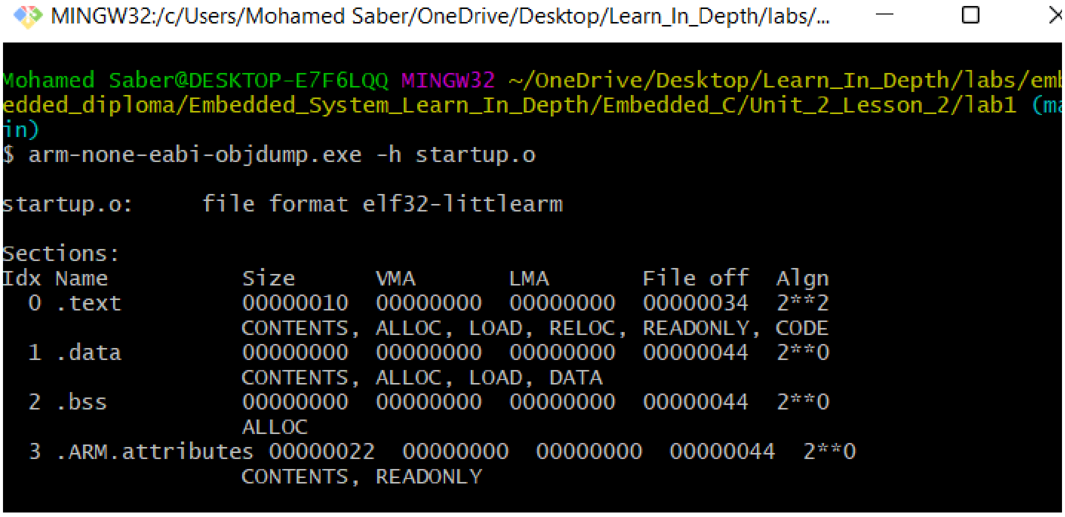
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-objdump.exe -h app.o

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 .comment      0000007f  00000000  00000000  000000b4  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000133  2**0
                  CONTENTS, READONLY
```

Sections of uart.o:


MINGW32:/c/Users/Mohamed Saber/OneDrive/Desktop/Learn_In_Depth/labs/...

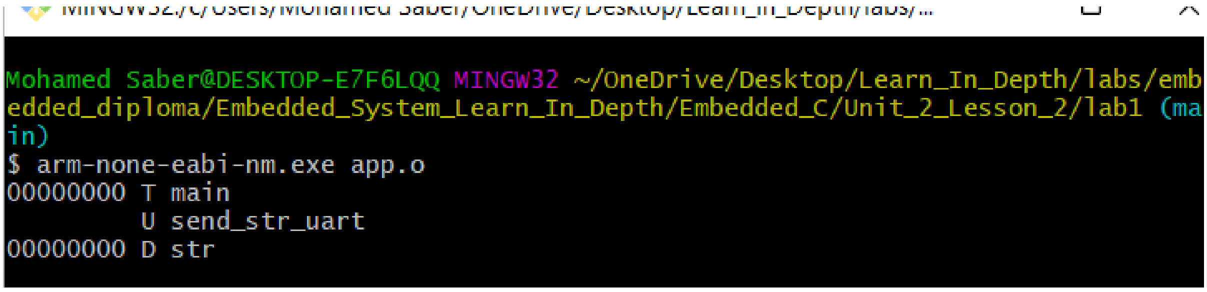
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/em
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (m
in)
$ arm-none-eabi-objdump.exe -h uart.o

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 .comment      0000007f  00000000  00000000  00000088  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000107  2**0
                  CONTENTS, READONLY
```

Sections of startup.o:


MINGW32:/c/Users/Mohamed Saber/OneDrive/Desktop/Learn_In_Depth/labs/...

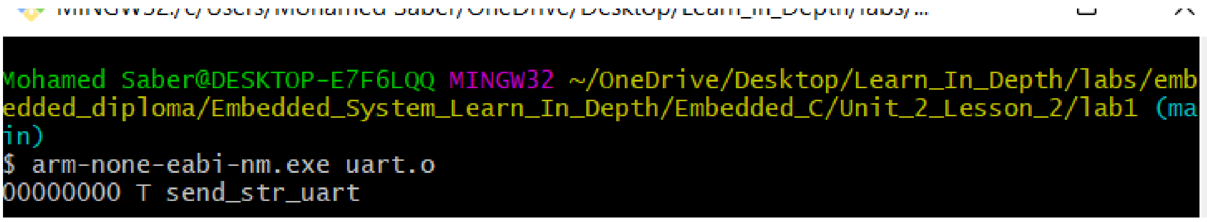
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-objdump.exe -h startup.o

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
```

Symbols of app.o:



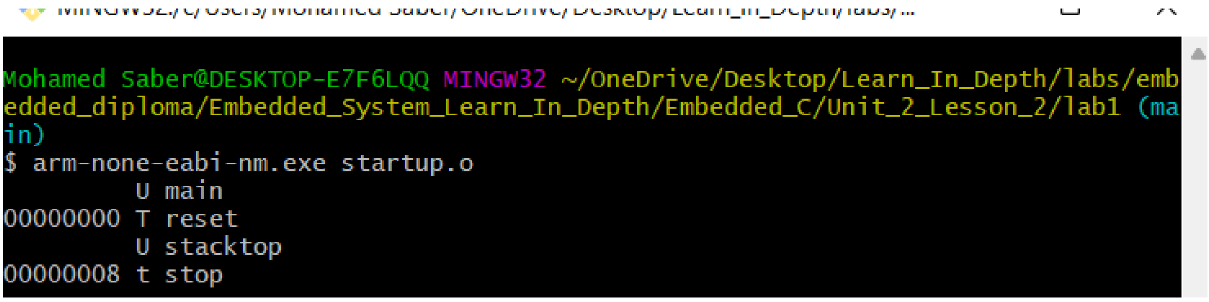
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-nm.exe app.o
00000000 T main
         U send_str_uart
00000000 D str
```

Symbols of uart.o:



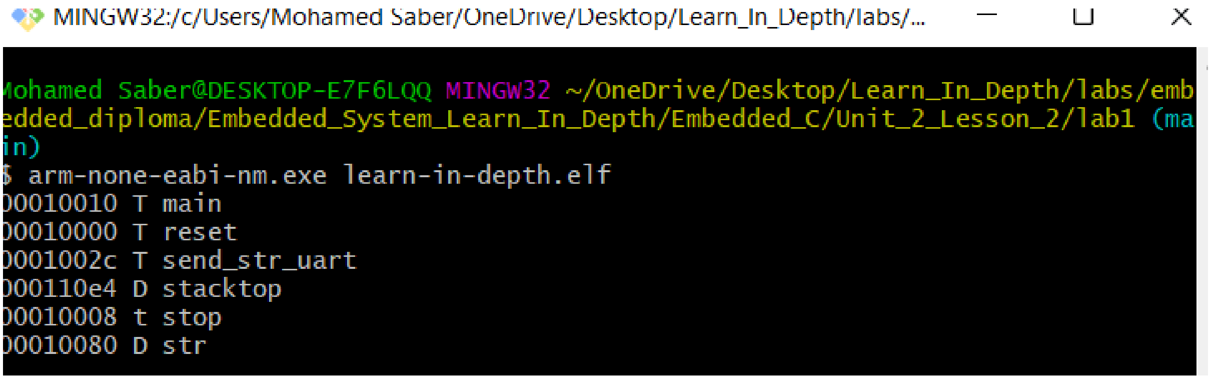
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-nm.exe uart.o
00000000 T send_str_uart
```

Symbols of startup.o:



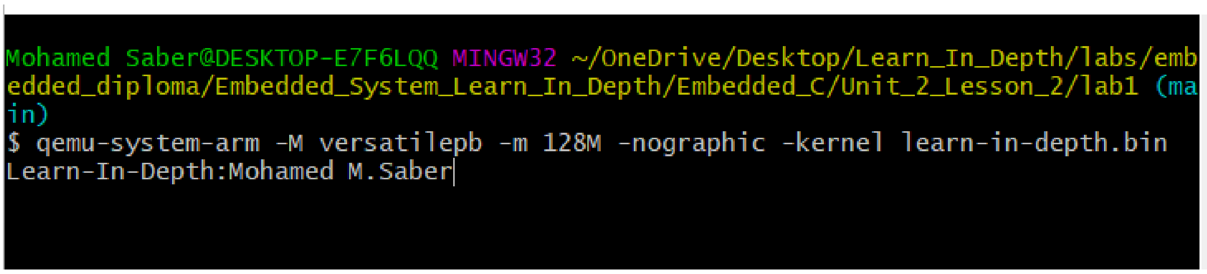
```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stacktop
00000008 t stop
```

Symbols of learn-in-depth.elf:


MINGW32:/c/Users/Mohamed Saber/OneDrive/Desktop/Learn_In_Depth/labs/...

```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
0001002c T send_str_uart
000110e4 D stacktop
00010008 t stop
00010080 D str
```

Running the application using QEMU emulator:



```
Mohamed Saber@DESKTOP-E7F6LQQ MINGW32 ~/OneDrive/Desktop/Learn_In_Depth/labs/emb
edded_diploma/Embedded_System_Learn_In_Depth/Embedded_C/Unit_2_Lesson_2/lab1 (ma
in)
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
Learn-In-Depth:Mohamed M.Saber
```