

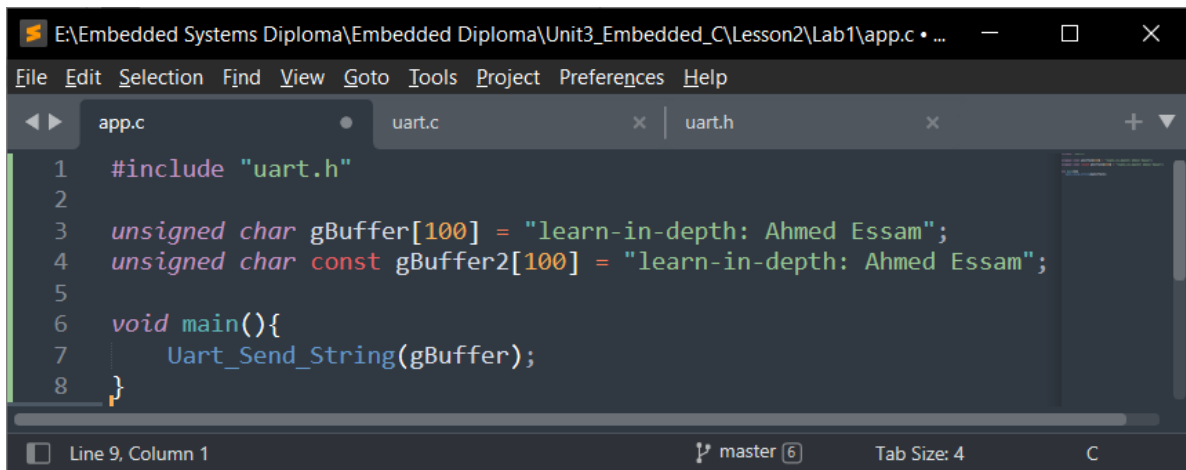
# **Mastering Embedded Systems**

## **Unit 3**

### **Embedded C**

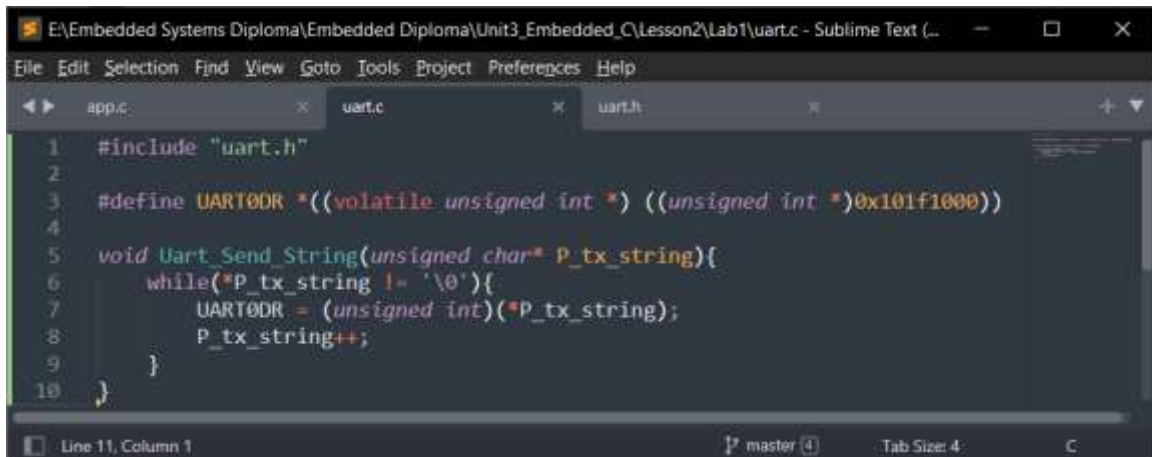
#### **Lab 1**

## App.c:



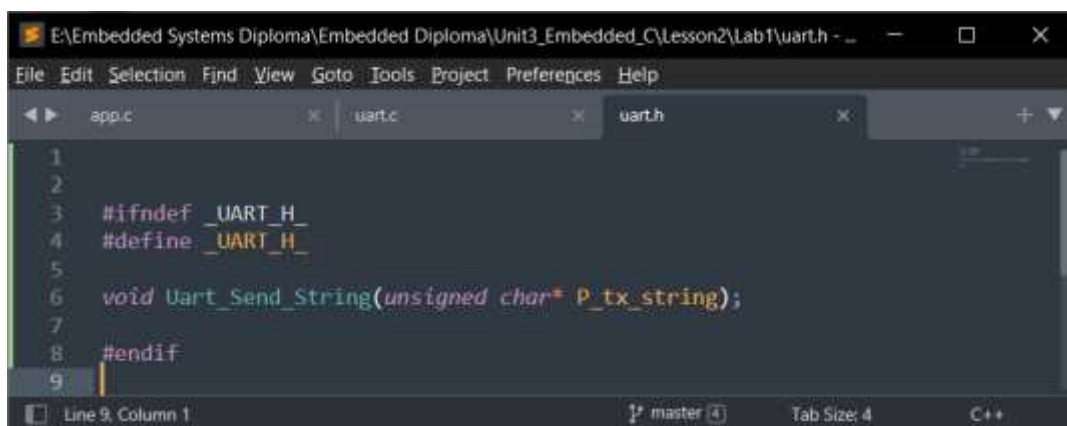
```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\app.c • ...
File Edit Selection Find View Goto Tools Project Preferences Help
app.c uart.c uart.h
1 #include "uart.h"
2
3 unsigned char gBuffer[100] = "learn-in-depth: Ahmed Essam";
4 unsigned char const gBuffer2[100] = "learn-in-depth: Ahmed Essam";
5
6 void main(){
7     Uart_Send_String(gBuffer);
8 }
Line 9, Column 1 master [6] Tab Size: 4 C
```

## Uart.c:



```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\uart.c - Sublime Text (...)
File Edit Selection Find View Goto Tools Project Preferences Help
app.c uart.c uart.h
1 #include "uart.h"
2
3 #define UART0DR *((volatile unsigned int *) ((unsigned int *)0x101f1000))
4
5 void Uart_Send_String(unsigned char* P_tx_string){
6     while(*P_tx_string != '\0'){
7         UART0DR = (unsigned int)(*P_tx_string);
8         P_tx_string++;
9     }
10 }
Line 11, Column 1 master [4] Tab Size: 4 C
```

## Uart.h:



```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\uart.h - ...
File Edit Selection Find View Goto Tools Project Preferences Help
app.c uart.c uart.h
1
2
3 #ifndef _UART_H_
4 #define _UART_H_
5
6 void Uart_Send_String(unsigned char* P_tx_string);
7
8 #endif
9
Line 9, Column 1 master [4] Tab Size: 4 C++
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/L...
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ export PATH=../../../../../ARM/bin/:$PATH

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ ls
app.c  uart.c  uart.h

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ |
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/L...
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-gcc.exe -c -g -mcpu=arm926ej-s -I . app.c -o app.o

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ ls
app.c  app.o  uart.c  uart.h

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-gcc.exe -c -g -mcpu=arm926ej-s -I . uart.c -o uart.o

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ ls
app.c  app.o  uart.c  uart.h  uart.o
```

## Section Headers for app.o with debugging info:

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Le...
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (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          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  0000005f 00000000 00000000 000001a9 2**0
                CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000 00000000 00000208 2**0
                CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    00000035 00000000 00000000 00000228 2**0
                CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     000000df 00000000 00000000 0000025d 2**0
                CONTENTS, READONLY, DEBUGGING
 9 .comment       0000007f 00000000 00000000 0000033c 2**0
                CONTENTS, READONLY
10 .debug_frame   0000002c 00000000 00000000 000003bc 2**2
                CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000032 00000000 00000000 000003e8 2**0
                CONTENTS, READONLY
```

## Section Headers for app.o without debugging info:

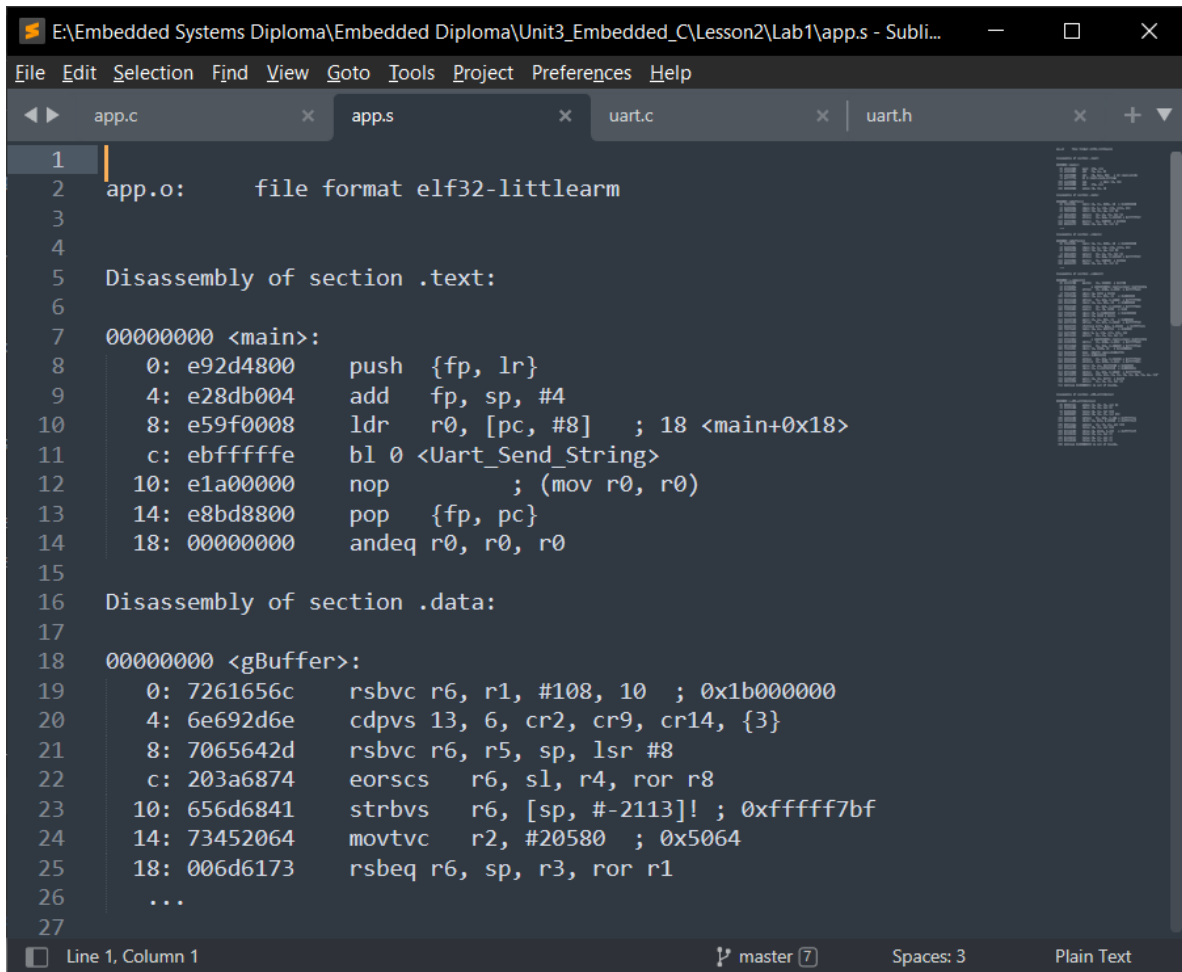
```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Le...
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objdump.exe -h app.o

app.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
 0 .text          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 .comment       0000007f 00000000 00000000 00000118 2**0
                CONTENTS, READONLY
 5 .ARM.attributes 00000032 00000000 00000000 00000197 2**0
                CONTENTS, READONLY

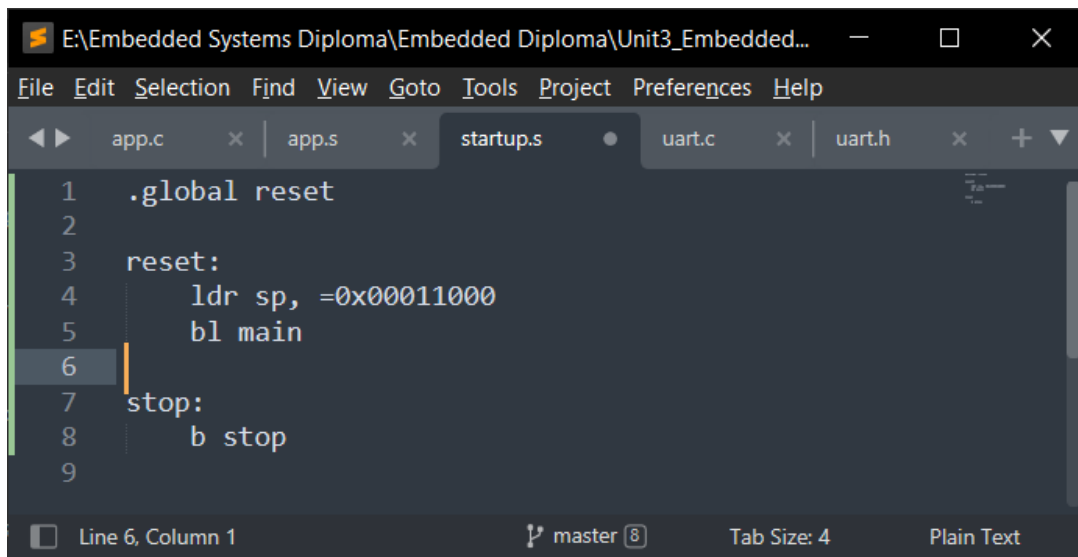
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ |
```

## Disassembly of app.o file:



```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\app.s - Subli...
File Edit Selection Find View Goto Tools Project Preferences Help
app.c app.s uart.c uarth
1
2 app.o:    file format elf32-littlearm
3
4
5 Disassembly of section .text:
6
7 00000000 <main>:
8     0: e92d4800    push    {fp, lr}
9     4: e28db004    add     fp, sp, #4
10    8: e59f0008    ldr     r0, [pc, #8]    ; 18 <main+0x18>
11    c: ebfffffe    bl     0 <Uart_Send_String>
12   10: e1a00000    nop
13   14: e8bd8800    pop     {fp, pc}
14   18: 00000000    andeq   r0, r0, r0
15
16 Disassembly of section .data:
17
18 00000000 <gBuffer>:
19     0: 7261656c    rsbvc   r6, r1, #108, 10 ; 0x1b000000
20     4: 6e692d6e    cdpvs   13, 6, cr2, cr9, cr14, {3}
21     8: 7065642d    rsbvc   r6, r5, sp, lsr #8
22    c: 203a6874    eorscs   r6, sl, r4, ror r8
23   10: 656d6841    strbvs   r6, [sp, #-2113]! ; 0xfffff7bf
24   14: 73452064    movtvc   r2, #20580 ; 0x5064
25   18: 006d6173    rsbeq    r6, sp, r3, ror r1
26   ...
27
Line 1, Column 1 master (7) Spaces: 3 Plain Text
```

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

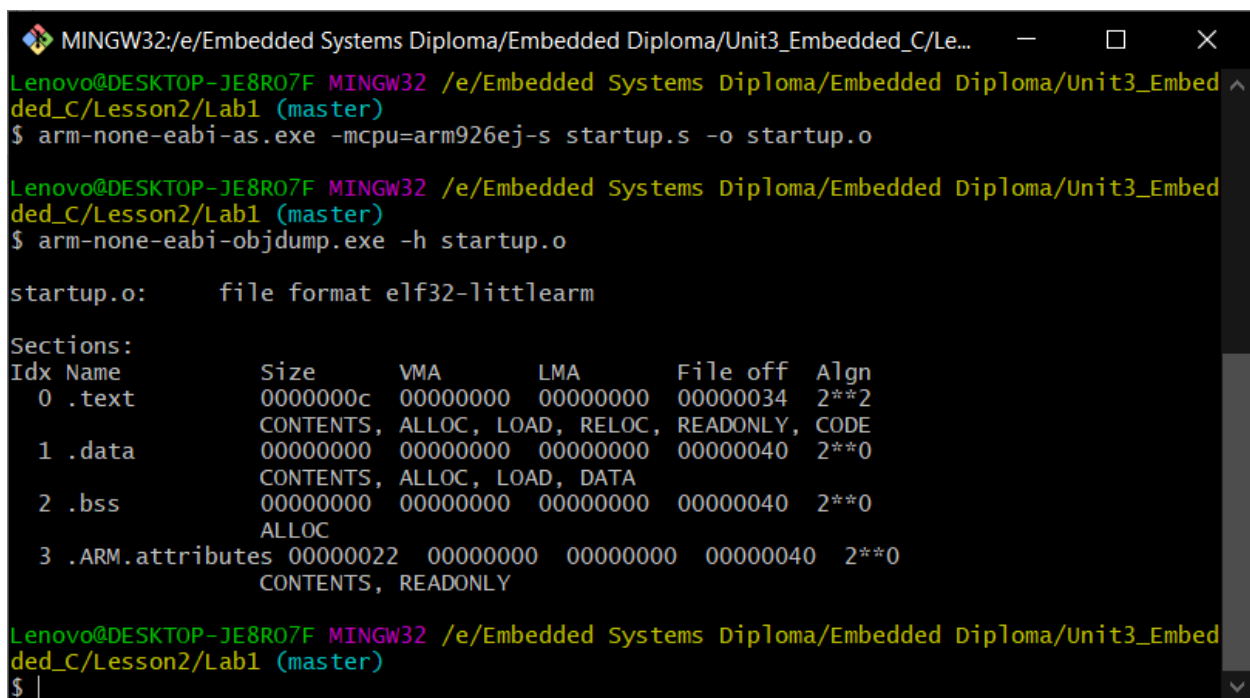


The screenshot shows a text editor window with the file path `E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded...`. The editor has tabs for `app.c`, `app.s`, `startup.s` (active), `uart.c`, and `uart.h`. The `startup.s` file contains the following assembly code:

```
1  .global reset
2
3  reset:
4      ldr sp, =0x00011000
5      bl main
6
7  stop:
8      b stop
9
```

The status bar at the bottom indicates "Line 6, Column 1", "master (8)", "Tab Size: 4", and "Plain Text".

## Startup.o objdump:



The screenshot shows a terminal window with the following commands and output:

```
MINGW32/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Le...
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o

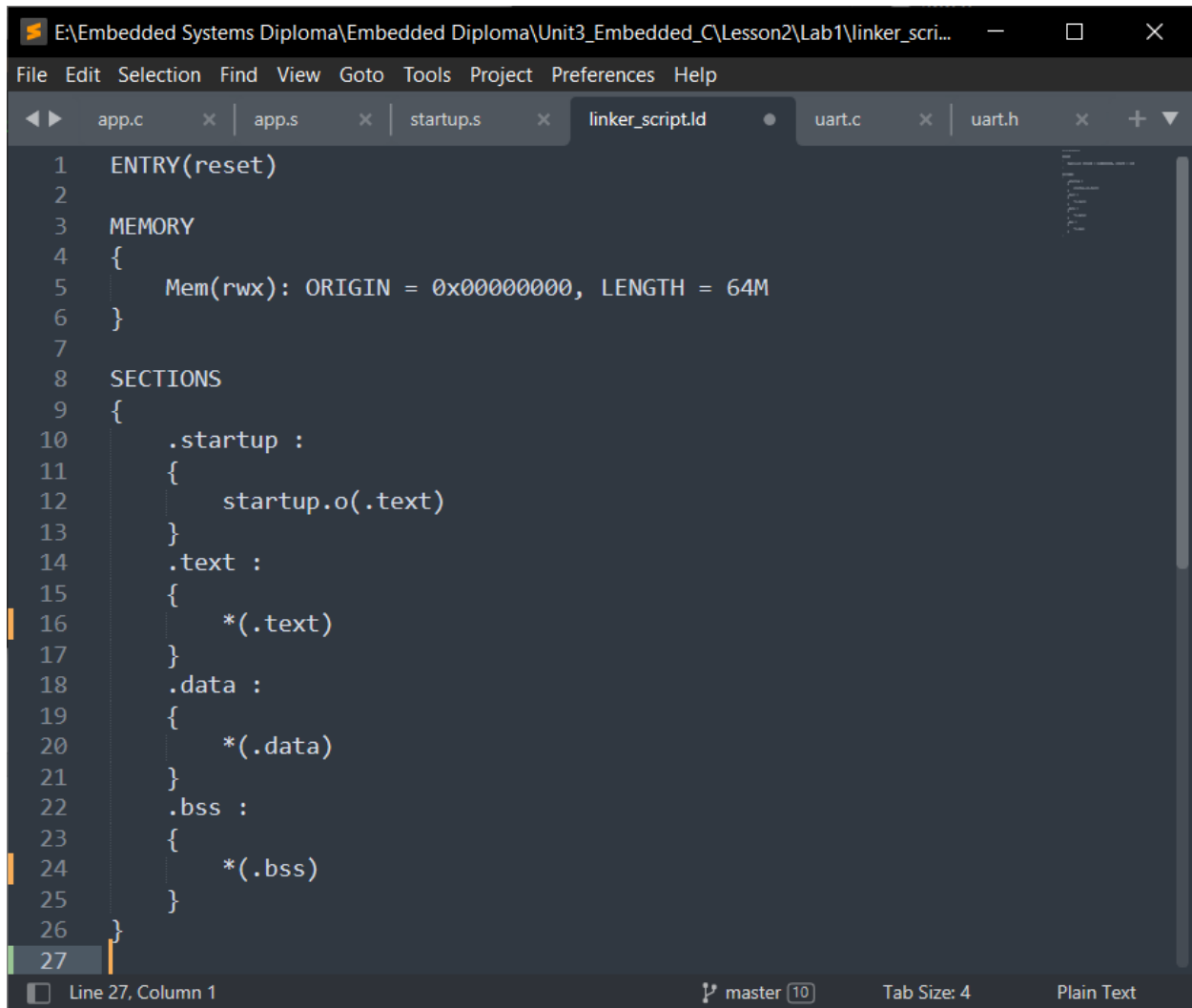
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
 0  .text          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
```

The terminal prompt is `Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)`.

## Linker\_script.ld initial:



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

Line 27, Column 1      master 10      Tab Size: 4      Plain Text

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-JE8R07F 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-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$
```

Linker\_script after adding .rodata to .text section:

```
File Edit Selection Find View Goto Tools Project Preferences Help
appic  app.s  startup.s  linker_script.ld  uart.c  uart.h

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

Line 27, Column 1    master (1)    Tab Size: 4    Plain Text



## Dumping data in learn-in-depth.elf:

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F 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-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

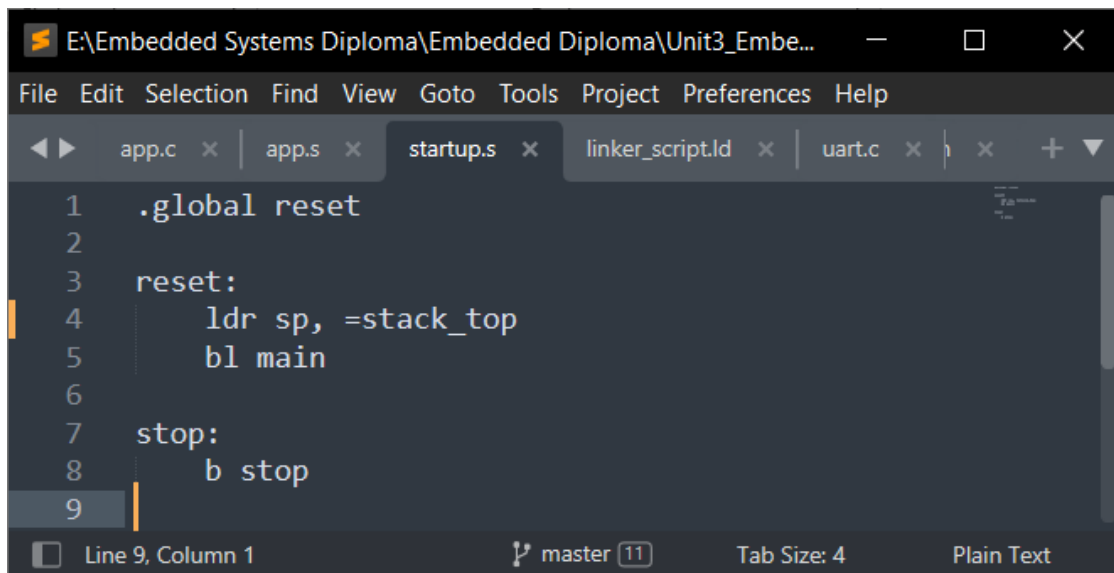
learn-in-depth.elf:      file format elf32-littlearm

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:

```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\linker_script.ld - Sublime Te...
File Edit Selection Find View Goto Tools Project Preferences Help
app.c | app.s | startup.s | linker_script.ld | outputmap | uart.c | uart.h
1 ENTRY(reset)
2
3 MEMORY
4 {
5     Mem(rwx): ORIGIN = 0x00000000, LENGTH = 64M
6 }
7
8 SECTIONS
9 {
10     . = 0x10000;
11     .startup :
12     {
13         startup.o(.text)
14     }> Mem
15     .text :
16     {
17         *(.text) *(.rodata)
18     }> Mem
19     .data :
20     {
21         *(.data)
22     }> Mem
23     .bss :
24     {
25         *(.bss) *(COMMON)
26     }> Mem
27     . = . + 0x1000;
28     stack_top = .;
29
30
```

**Modified startup.s after adding stack\_top address:**

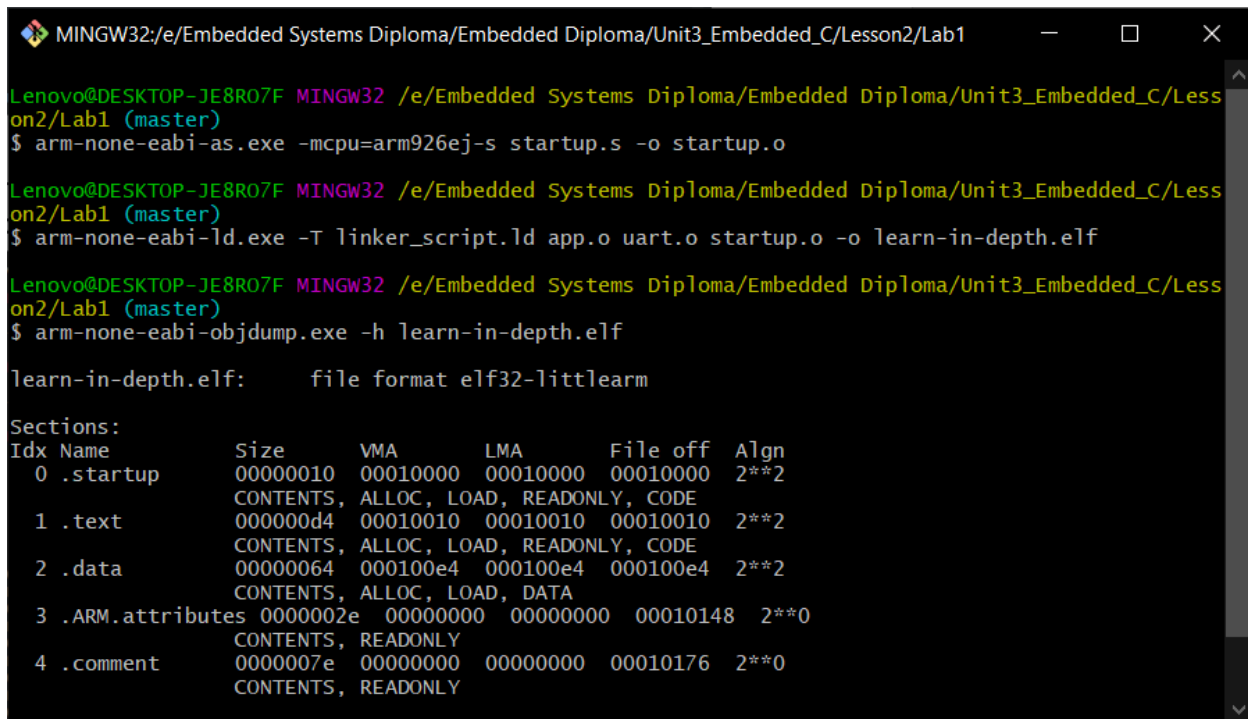


The screenshot shows a code editor window with the title "E:\Embedded Systems Diploma\Embedded Diploma\Unit3\_Embe...". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The tab bar shows several files: app.c, app.s, startup.s (active), linker\_script.ld, and uart.c. The main editor area displays the following assembly code:

```
1  .global reset
2
3  reset:
4      ldr sp, =stack_top
5      bl main
6
7  stop:
8      b stop
9
```

The status bar at the bottom indicates "Line 9, Column 1", "master (11)", "Tab Size: 4", and "Plain Text".

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



The screenshot shows a terminal window with the title "MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3\_Embedded\_C/Lesson2/Lab1". The terminal output is as follows:

```
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o

Lenovo@DESKTOP-JE8R07F 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-JE8R07F 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        00000010  00010000  00010000  00010000  2**2
   CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .text           000000d4  00010010  00010010  00010010  2**2
   CONTENTS, ALLOC, LOAD, READONLY, CODE
 2 .data           00000064  000100e4  000100e4  000100e4  2**2
   CONTENTS, ALLOC, LOAD, DATA
 3 .ARM.attributes 0000002e  00000000  00000000  00010148  2**0
   CONTENTS, READONLY
 4 .comment         0000007e  00000000  00000000  00010176  2**0
   CONTENTS, READONLY
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-nm.exe app.o
00000000 D gBuffer
00000000 R gBuffer2
00000000 T main
00000000 U Uart_Send_String

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-nm.exe uart.o
00000000 T Uart_Send_String

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-nm.exe startup.o
00000000 U main
00000000 T reset
00000000 U stack_top
00000008 t stop

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-nm.exe learn-in-depth.elf
000100e4 D gBuffer
00010080 T gBuffer2
00010010 T main
00010000 T reset
00011148 D stack_top
00010008 t stop
0001002c T Uart_Send_String

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-ld.exe -T linker_script.ld -Map=output.map app.o uart.o startup.o -o learn-in-depth.elf
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ |
```

**Output.map:**

```
E:\Embedded Systems Diploma\Embedded Diploma\Unit3_Embedded_C\Lesson2\Lab1\output.map - Sublime Text ...
File Edit Selection Find View Goto Tools Project Preferences Help
app.c app.s startup.s linker_script.ld output.map uart.c uart.h
1
2 Memory Configuration
3
4 Name Origin Length Attributes
5 Mem 0x00000000 0x04000000 xrw
6 *default* 0x00000000 0xffffffff
7
8 Linker script and memory map
9
10 0x00010000 . = 0x10000
11
12 .startup 0x00010000 0x10
13 startup.o(.text)
14 .text 0x00010000 0x10 startup.o
15 0x00010000 reset
16
17 .text 0x00010010 0xd4
18 *(.text)
19 .text 0x00010010 0x1c app.o
20 0x00010010 main
21 .text 0x0001002c 0x54 uart.o
22 0x0001002c Uart_Send_String
23 *(.rodata)
24 .rodata 0x00010080 0x64 app.o
25 0x00010080 gBuffer2
26
27 .glue_7 0x000100e4 0x0
28 .glue_7 0x000100e4 0x0 linker stubs
29
30 .glue_7t 0x000100e4 0x0
31 .glue_7t 0x000100e4 0x0 linker stubs
Line 1, Column 1 master Spaces: 4 Plain Text
```

## Using readelf utility to check entry point address:

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-readelf.exe -a learn-in-depth.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 endian
  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:              66584 (bytes into file)
  Flags:                                0x5000200, Version5 EABI, soft-float ABI
  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:              9
  Section header string table index:      8

Section Headers:
 [Nr] Name                Type            Addr      Off      Size    ES Flg Lk Inf Al
 [ 0]                     NULL            00000000  000000  000000  00   0  0  0
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ ls
app.c  app.s          learn-in-depth.elf  output.map  startup.s  uart.h
app.o  learn-in-depth.bin  linker_script.ld    startup.o   uart.c     uart.o

Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1 (master)
$ |
```

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

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
$ export PATH=../../../../../qemu/:$PATH
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
$ |
```

**Running learn-in-depth.bin on qemu:**

```
MINGW32:/e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
$ export PATH=../../../../../qemu/:$PATH
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
$ qemu-system-arm
qemu-system-arm.exe      qemu-system-armw.exe
Lenovo@DESKTOP-JE8R07F MINGW32 /e/Embedded Systems Diploma/Embedded Diploma/Unit3_Embedded_C/Lesson2/Lab1
$ qemu-system-arm.exe -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn-in-depth: Ahmed Essam
```