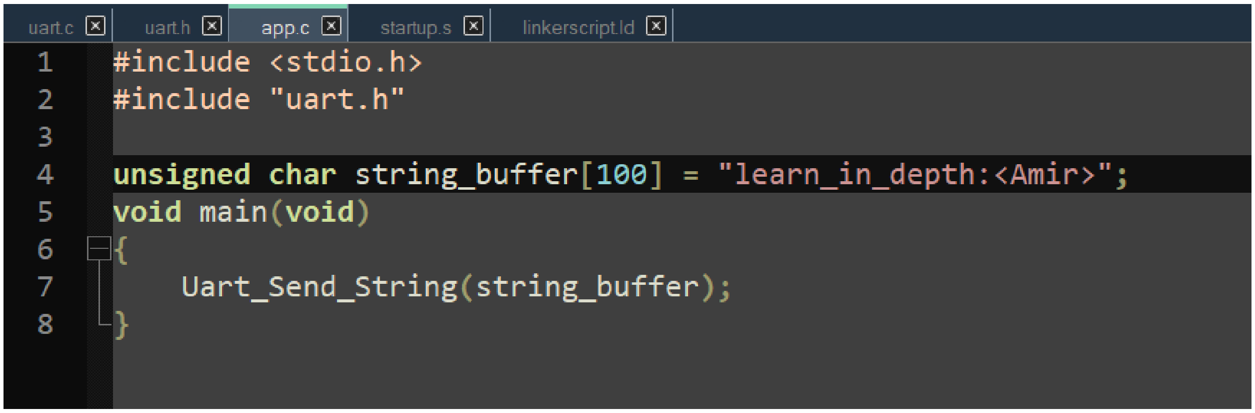
Full Analysis of bare-metal Software Runs on ARM "VersatilePB physical Board"

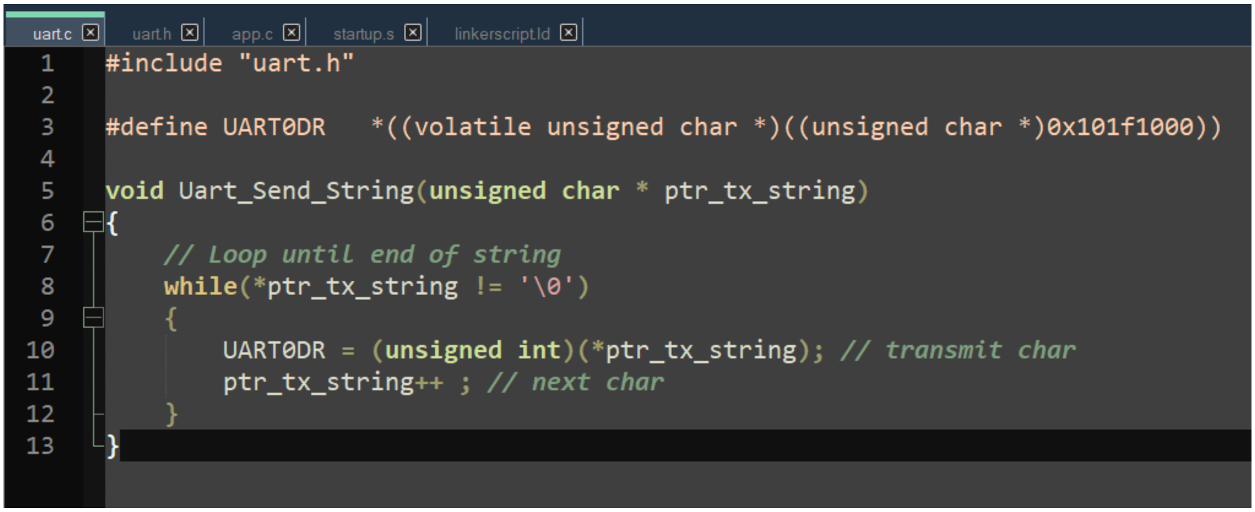
- **Files.c**

app.c



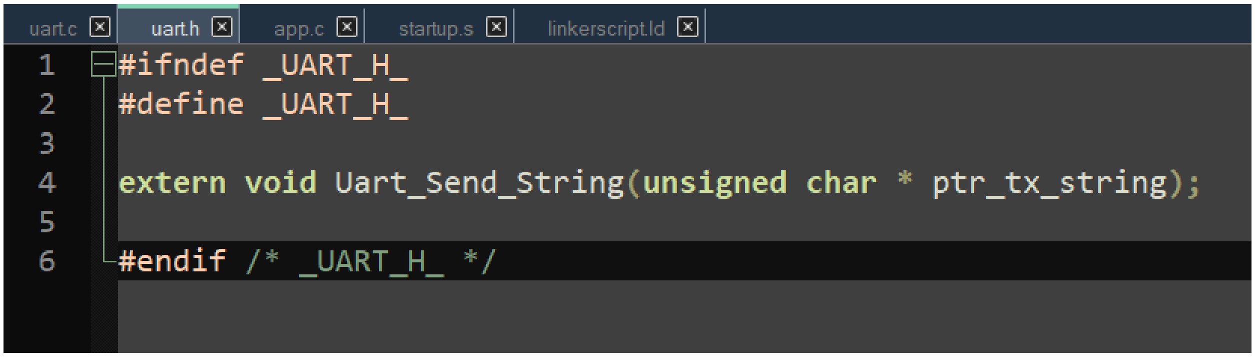
```c
#include <stdio.h>
#include "uart.h"

unsigned char string_buffer[100] = "learn_in_depth:<Amir>";
void main(void)
{
	Uart_Send_String(string_buffer);
}
```

uart.c



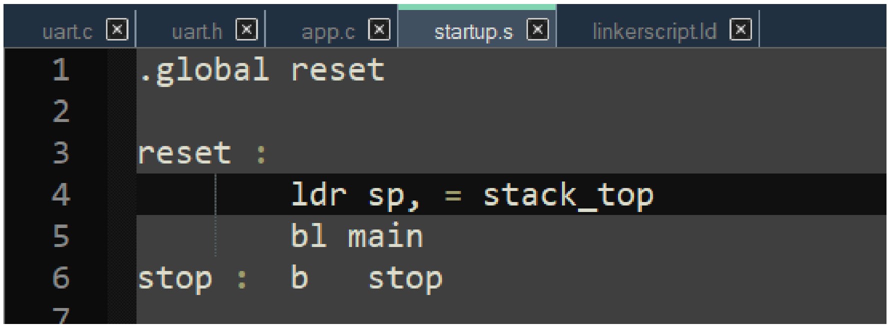
```c
#include "uart.h"

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

void Uart_Send_String(unsigned char * ptr_tx_string)
{
	// Loop until end of string
	while(*ptr_tx_string != '\0')
	{
		UART0DR = (unsigned int)(*ptr_tx_string); // transmit char
		ptr_tx_string++ ; // next char
	}
}
```

uart.h



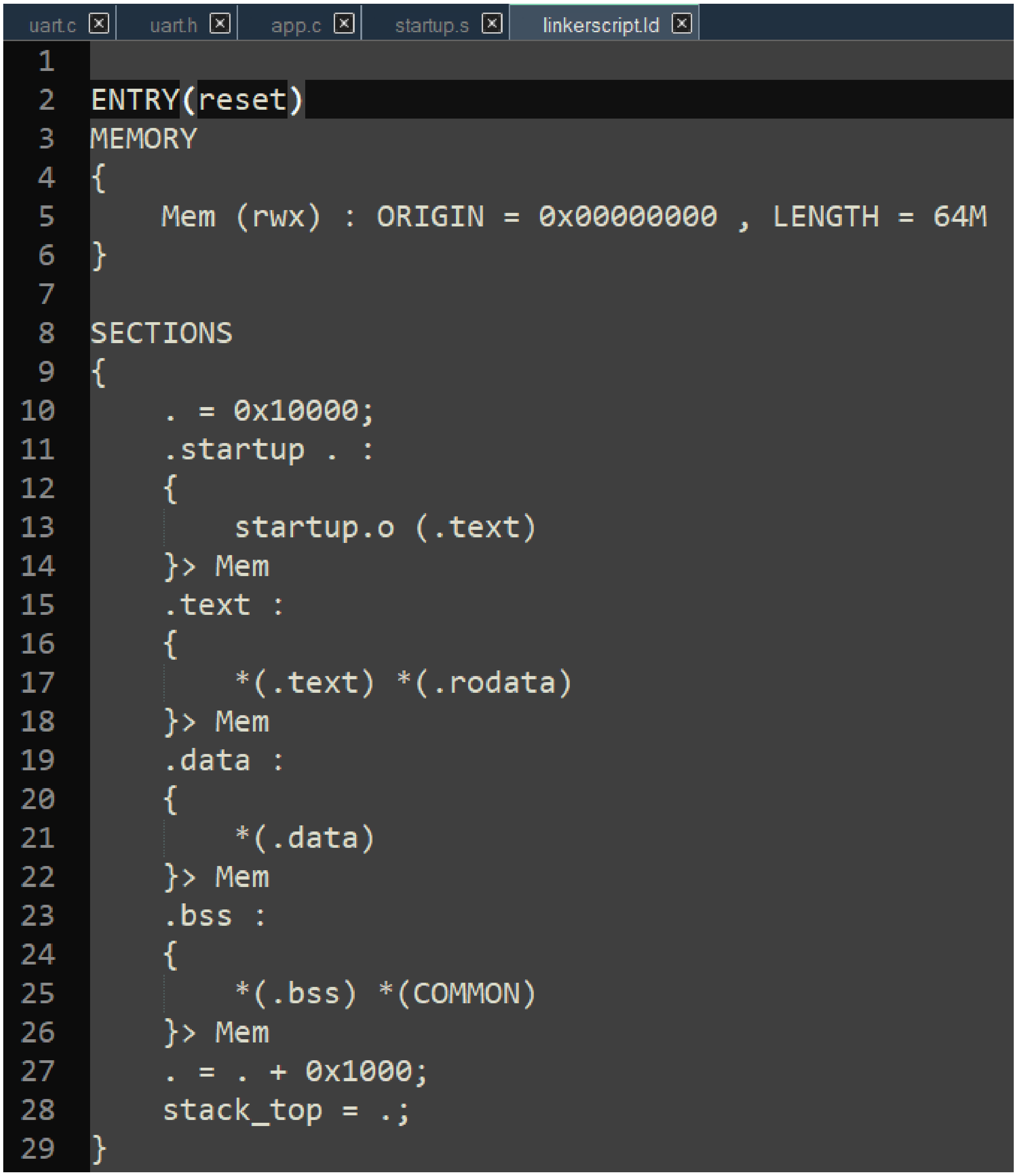
```c
#ifndef _UART_H_
#define _UART_H_

extern void Uart_Send_String(unsigned char * ptr_tx_string);

#endif /* _UART_H_ */
```

- **Startup.s**



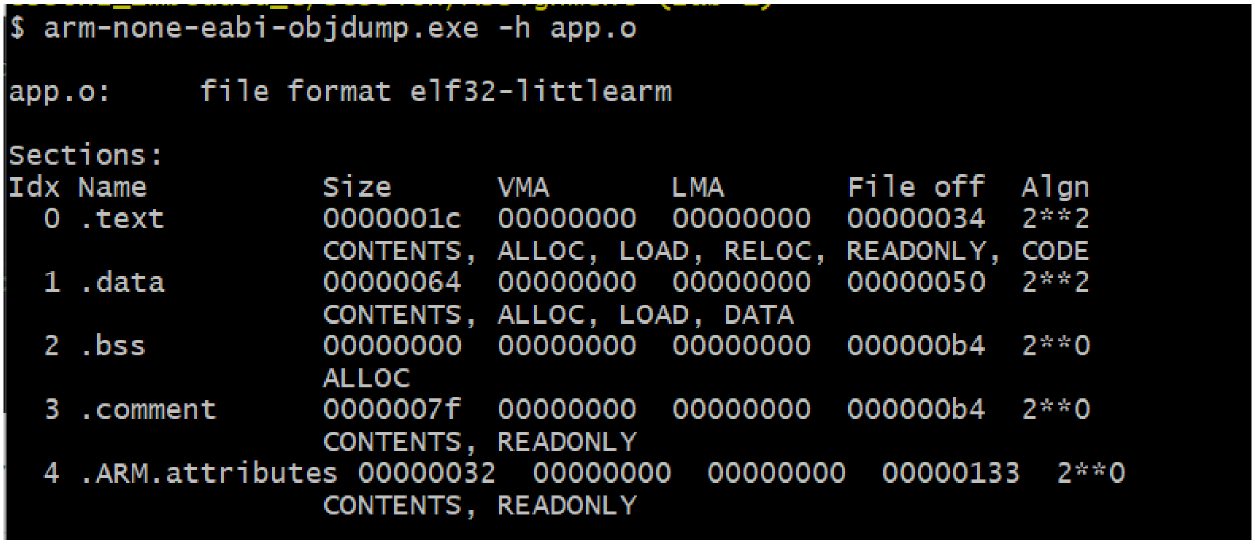
```
.global reset

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

- **Linkerscript.ld**



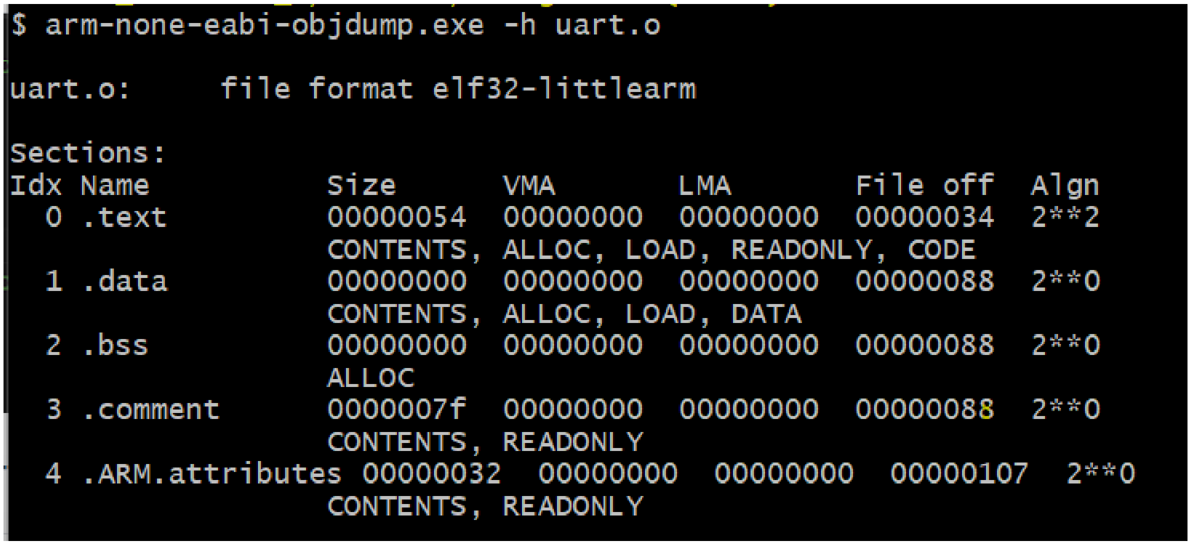
```

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;
    stack_top = .;
}
```

- **Analysis of The object files of the program :**
  - **App.o :**



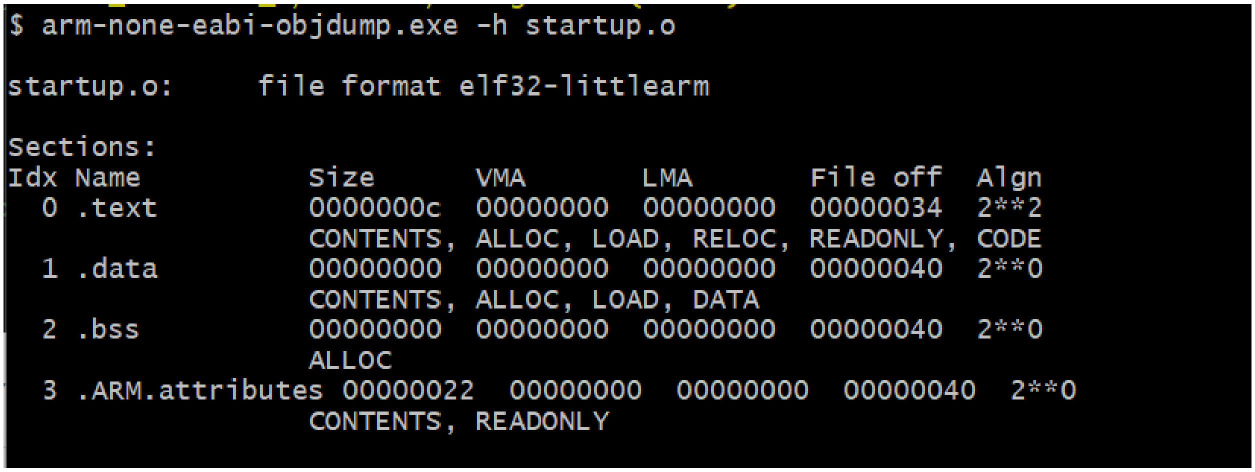
```
$ arm-none-eabi-objdump.exe -h app.o

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
```

  - **uart.o :**



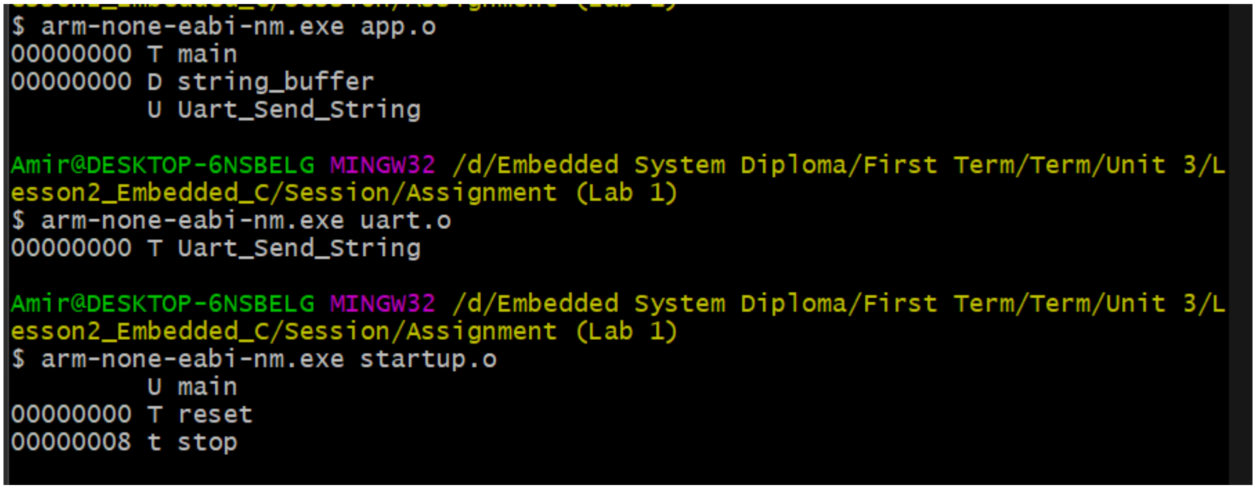
```
$ arm-none-eabi-objdump.exe -h uart.o

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
```

- **startup.o :**



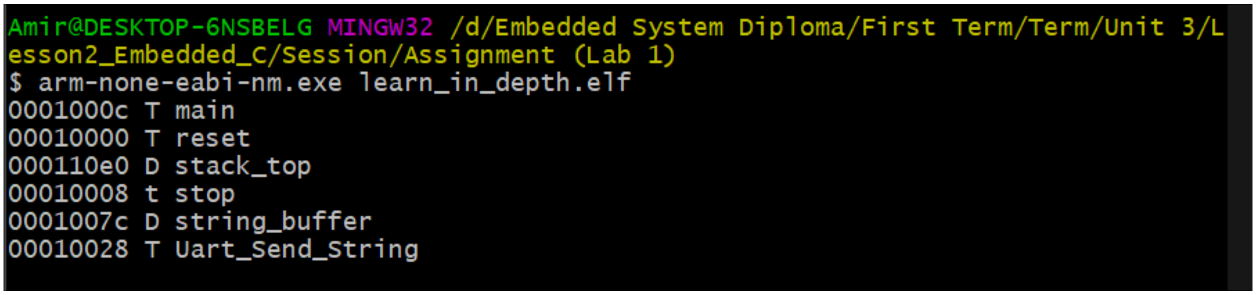
```
$ arm-none-eabi-objdump.exe -h startup.o

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
```

- **Symbols:**
- **Resolved /Unresolved Sybmols with Virtual Addresses (Before Linking) :**



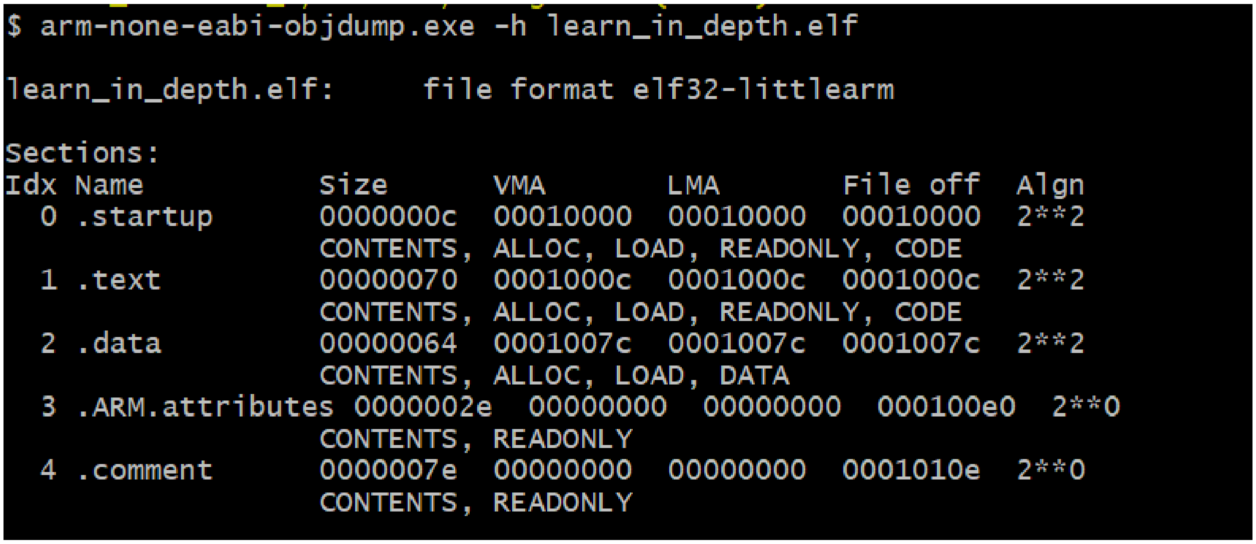
```
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
         U Uart_Send_String

Amir@DESKTOP-6NSBELG MINGW32 /d/Embedded System Diploma/First Term/Term/Unit 3/L
esson2_Embedded_C/Session/Assignment (Lab 1)
$ arm-none-eabi-nm.exe uart.o
00000000 T Uart_Send_String

Amir@DESKTOP-6NSBELG MINGW32 /d/Embedded System Diploma/First Term/Term/Unit 3/L
esson2_Embedded_C/Session/Assignment (Lab 1)
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
00000008 t stop
```

- **Resolved Sybmols (After Linking) :**



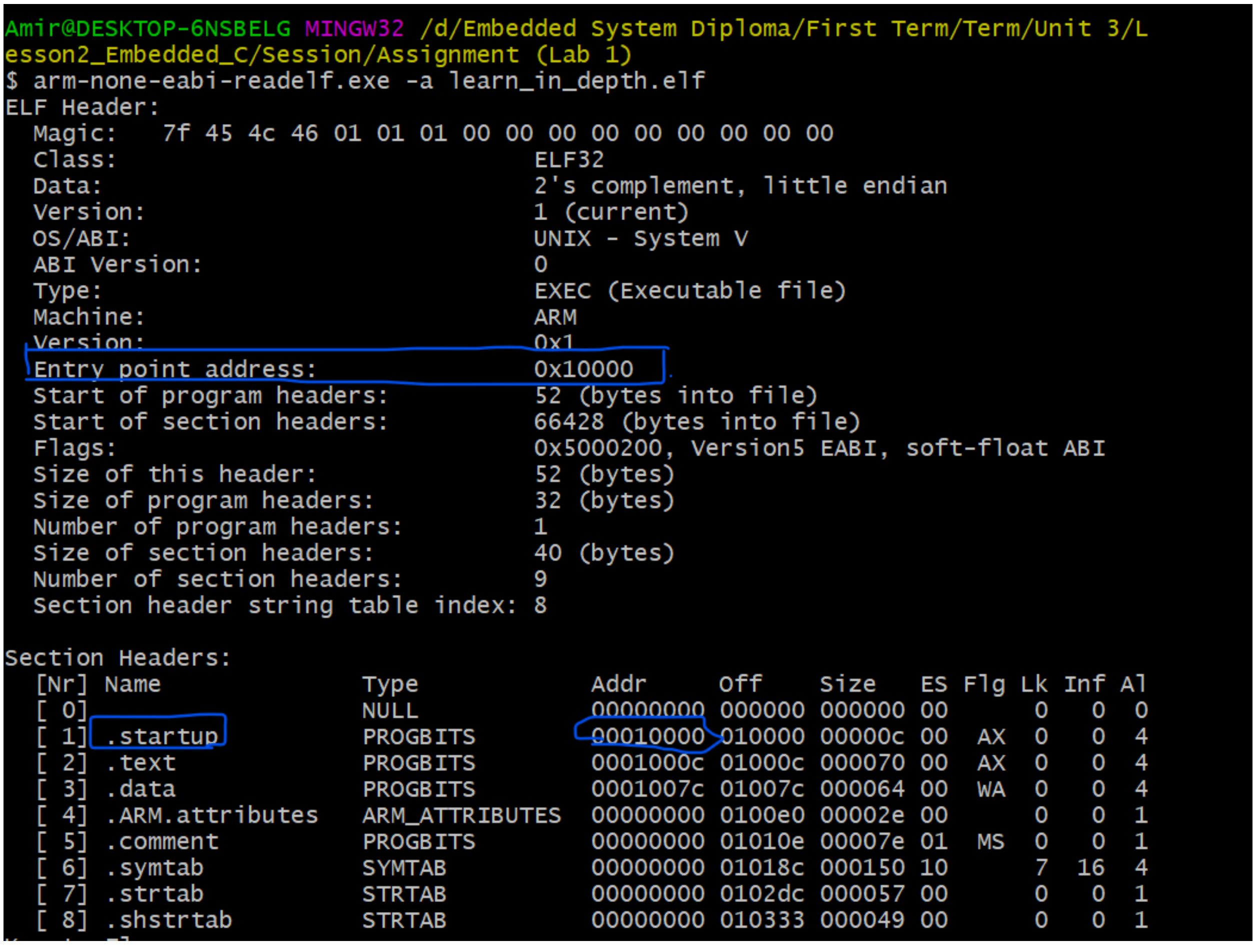
```
Amir@DESKTOP-6NSBELG MINGW32 /d/Embedded System Diploma/First Term/Term/Unit 3/L
esson2_Embedded_C/Session/Assignment (Lab 1)
$ arm-none-eabi-nm.exe learn_in_depth.elf
0001000c T main
00010000 T reset
000110e0 D stack_top
00010008 t stop
0001007c D string_buffer
00010028 T Uart_Send_String
```

- **Excutable_file after linking with(physical addresses) :**



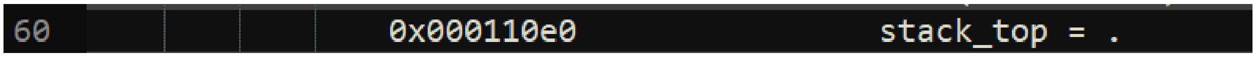
```
$ arm-none-eabi-objdump.exe -h learn_in_depth.elf

learn_in_depth.elf:     file format elf32-littlearm

Sections:
Idx Name              Size      VMA       LMA       File off  Algn
  0 .startup          0000000c  00010000  00010000  00010000  2**2
                      CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text             00000070  0001000c  0001000c  0001000c  2**2
                      CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data             00000064  0001007c  0001007c  0001007c  2**2
                      CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000100e0  2**0
                      CONTENTS, READONLY
  4 .comment          0000007e  00000000  00000000  0001010e  2**0
                      CONTENTS, READONLY
```

- **Address of Entry point :**



```
Amir@DESKTOP-6NSBELG MINGW32 /d/Embedded System Diploma/First Term/Term/Unit 3/L
esson2_Embedded_C/Session/Assignment (Lab 1)
$ 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:          66428 (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
  [ 1] .startup          PROGBITS        00010000 010000 00000c 00  AX  0   0  4
  [ 2] .text             PROGBITS        0001000c 01000c 000070 00  AX  0   0  4
  [ 3] .data             PROGBITS        0001007c 01007c 000064 00  WA  0   0  4
  [ 4] .ARM.attributes   ARM_ATTRIBUTES  00000000 0100e0 00002e 00      0   0  1
  [ 5] .comment          PROGBITS        00000000 01010e 00007e 01  MS  0   0  1
  [ 6] .symtab           SYMTAB          00000000 01018c 000150 10      7  16  4
  [ 7] .strtab           STRTAB          00000000 0102dc 000057 00      0   0  1
  [ 8] .shstrtab         STRTAB          00000000 010333 000049 00      0   0  1
```

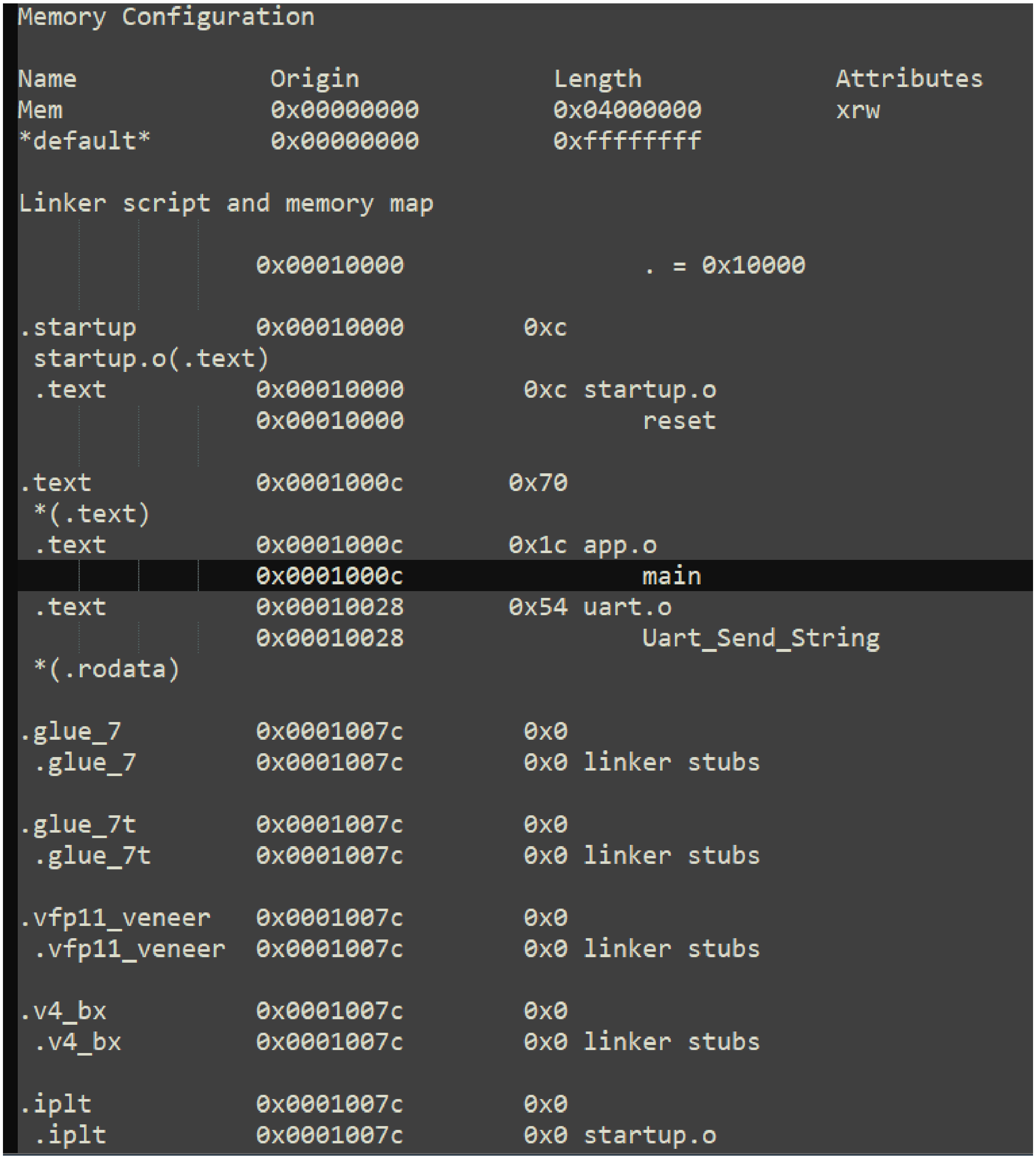
- **Address of Stack_Top :**



```
                    0x000110e0                stack_top = .
```

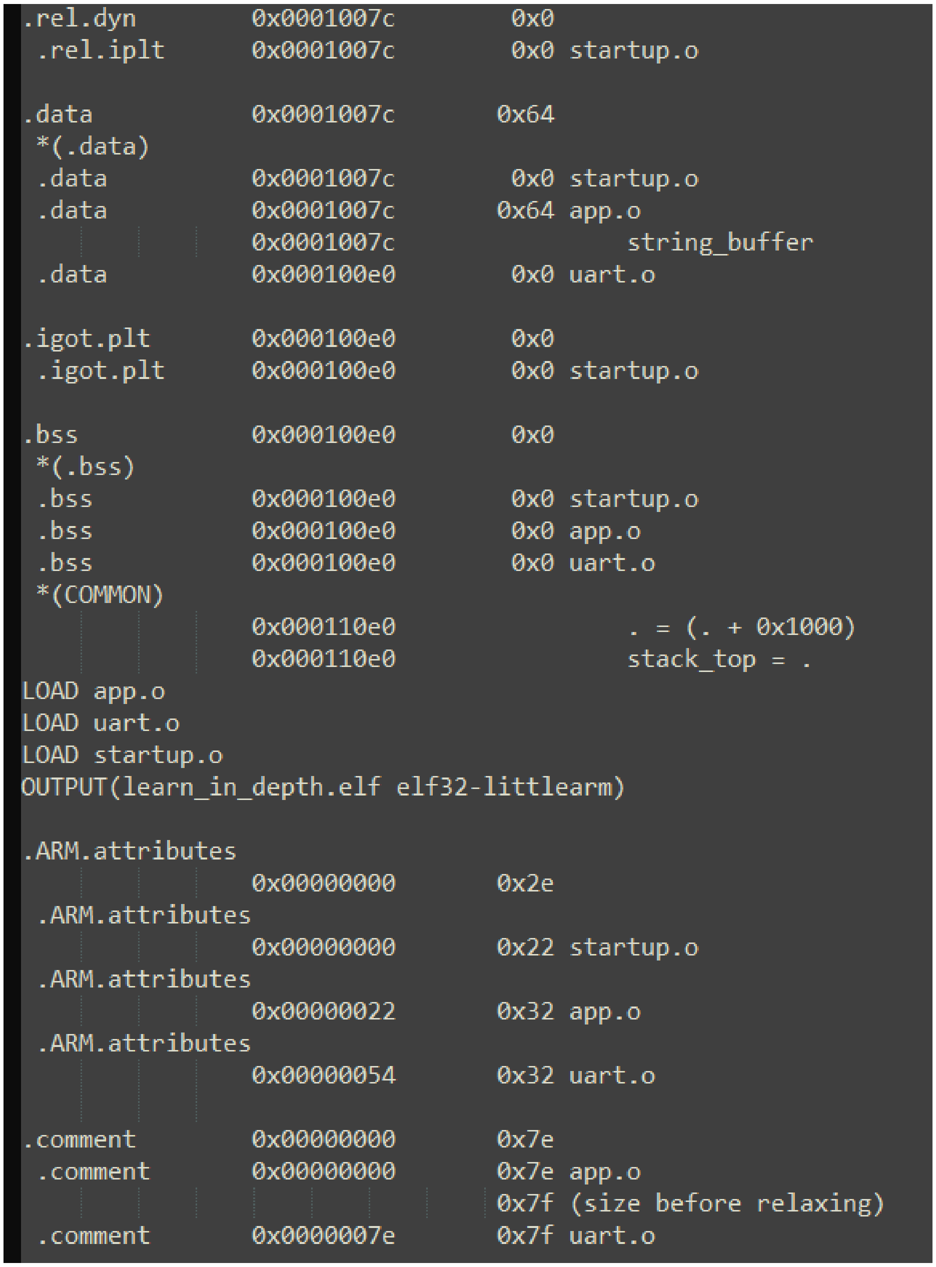
- **The Map File :**

Part (1) :



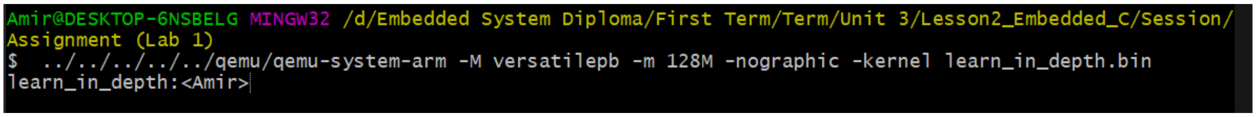
```
Memory Configuration

Name                Origin              Length              Attributes
Mem                 0x00000000          0x04000000          xrw
*default*           0x00000000          0xffffffff

Linker script and memory map

                0x00010000                . = 0x10000

.startup        0x00010000        0xc
 startup.o(.text)
 .text          0x00010000        0xc startup.o
                0x00010000                reset

.text           0x0001000c       0x70
 *(.text)
 .text          0x0001000c       0x1c app.o
                0x0001000c                main
 .text          0x00010028       0x54 uart.o
                0x00010028                Uart_Send_String
 *(.rodata)

.glue_7         0x0001007c        0x0
 .glue_7        0x0001007c        0x0 linker stubs

.glue_7t        0x0001007c        0x0
 .glue_7t       0x0001007c        0x0 linker stubs

.vfp11_veneer   0x0001007c        0x0
 .vfp11_veneer  0x0001007c        0x0 linker stubs

.v4_bx          0x0001007c        0x0
 .v4_bx         0x0001007c        0x0 linker stubs

.iplt           0x0001007c        0x0
 .iplt          0x0001007c        0x0 startup.o
```

Part (2) :



```
.rel.dyn        0x0001007c        0x0
 .rel.iplt      0x0001007c        0x0 startup.o

.data           0x0001007c       0x64
 *(.data)
 .data          0x0001007c        0x0 startup.o
 .data          0x0001007c       0x64 app.o
                0x0001007c                string_buffer
 .data          0x000100e0        0x0 uart.o

.igot.plt       0x000100e0        0x0
 .igot.plt      0x000100e0        0x0 startup.o

.bss            0x000100e0        0x0
 *(.bss)
 .bss           0x000100e0        0x0 startup.o
 .bss           0x000100e0        0x0 app.o
 .bss           0x000100e0        0x0 uart.o
 *(COMMON)
                0x000110e0                . = (. + 0x1000)
                0x000110e0                stack_top = .
LOAD app.o
LOAD uart.o
LOAD startup.o
OUTPUT(learn_in_depth.elf elf32-littlearm)

.ARM.attributes
                0x00000000       0x2e
 .ARM.attributes
                0x00000000       0x22 startup.o
 .ARM.attributes
                0x00000022       0x32 app.o
 .ARM.attributes
                0x00000054       0x32 uart.o

.comment        0x00000000       0x7e
 .comment       0x00000000       0x7e app.o
                                 0x7f (size before relaxing)
 .comment       0x0000007e       0x7f uart.o
```

- **Output of the Baremetal SW Using (QEMU) simulator :**



```
Amir@DESKTOP-6NSBELG MINGW32 /d/Embedded System Diploma/First Term/Term/Unit 3/Lesson2_Embedded_C/Session/
Assignment (Lab 1)
$  ../../../../../qemu/qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn_in_depth.bin
learn_in_depth:<Amir>
```