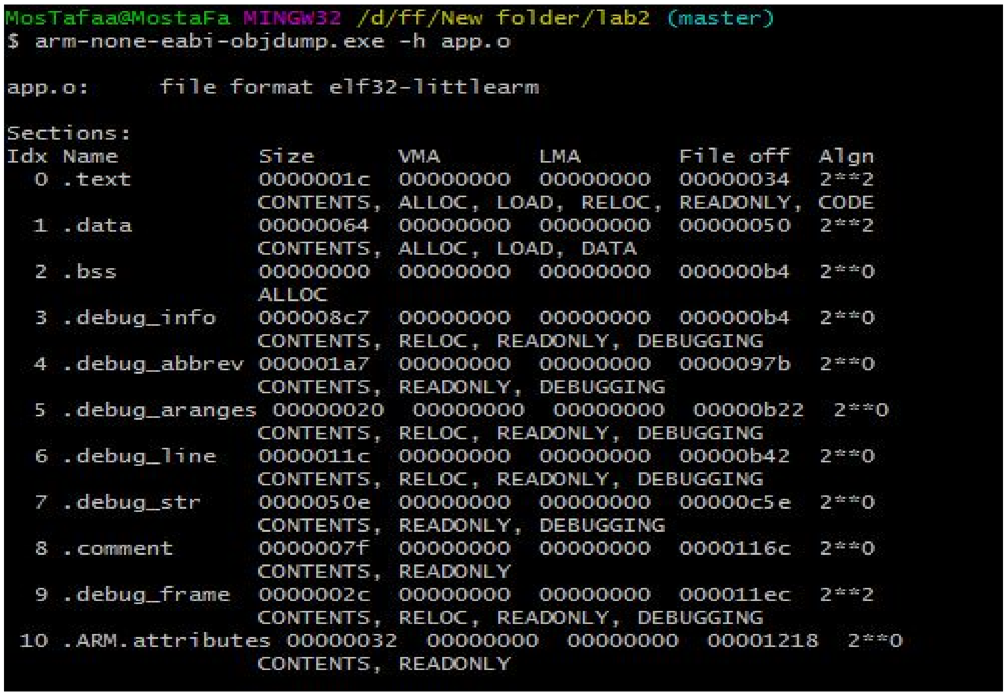
1- Sections of objects files

App.o

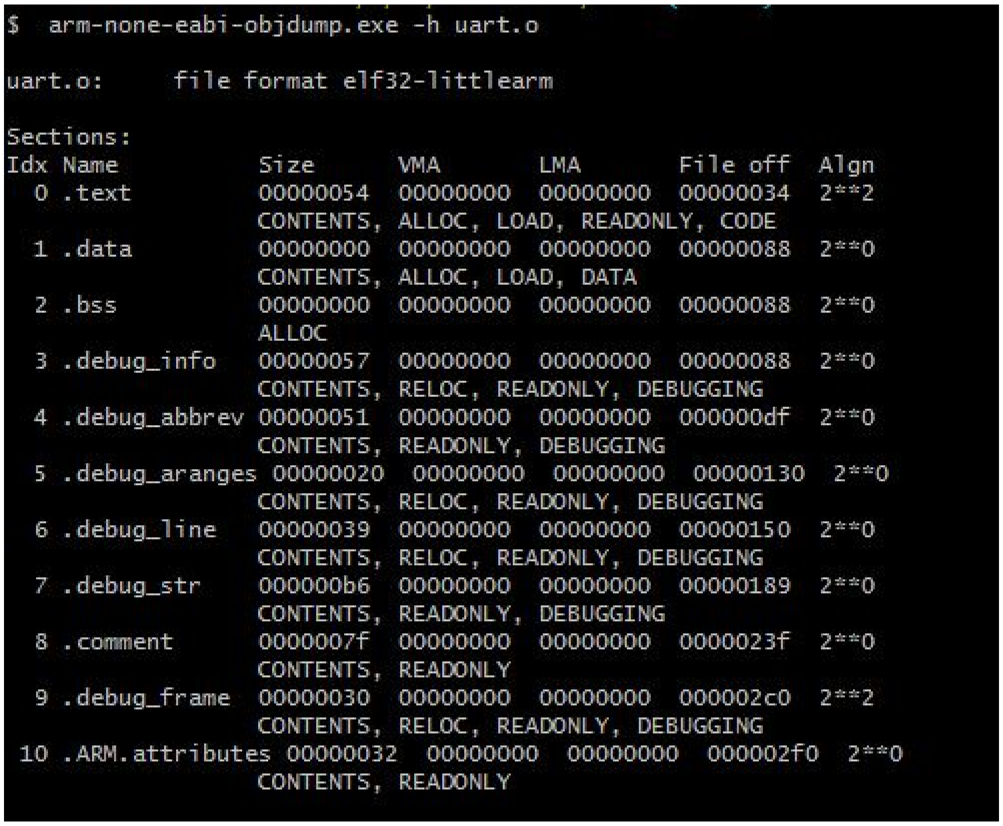


```
MosTafaa@MostaFa MINGW32 /d/ff/New folder/lab2 (master)
$ arm-none-eabi-objdump.exe -h app.o

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 .debug_info   000008c7  00000000  00000000  000000b4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001a7  00000000  00000000  0000097b  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000020  00000000  00000000  00000b22  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_line   0000011c  00000000  00000000  00000b42  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_str    0000050e  00000000  00000000  00000c5e  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  0000116c  2**0
                  CONTENTS, READONLY
  9 .debug_frame  0000002c  00000000  00000000  000011ec  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  00001218  2**0
                  CONTENTS, READONLY
```

Uart.o

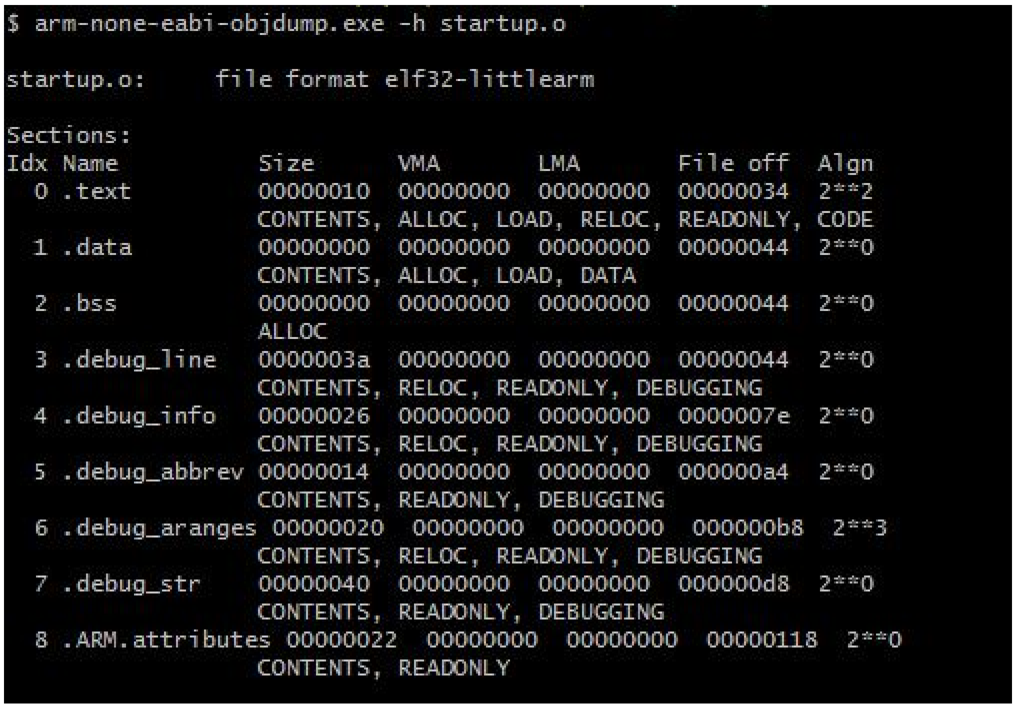


```
$  arm-none-eabi-objdump.exe -h uart.o

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 .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    000000b6  00000000  00000000  00000189  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  0000023f  2**0
                  CONTENTS, READONLY
  9 .debug_frame  00000030  00000000  00000000  000002c0  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  000002f0  2**0
                  CONTENTS, READONLY
```

Startup.o

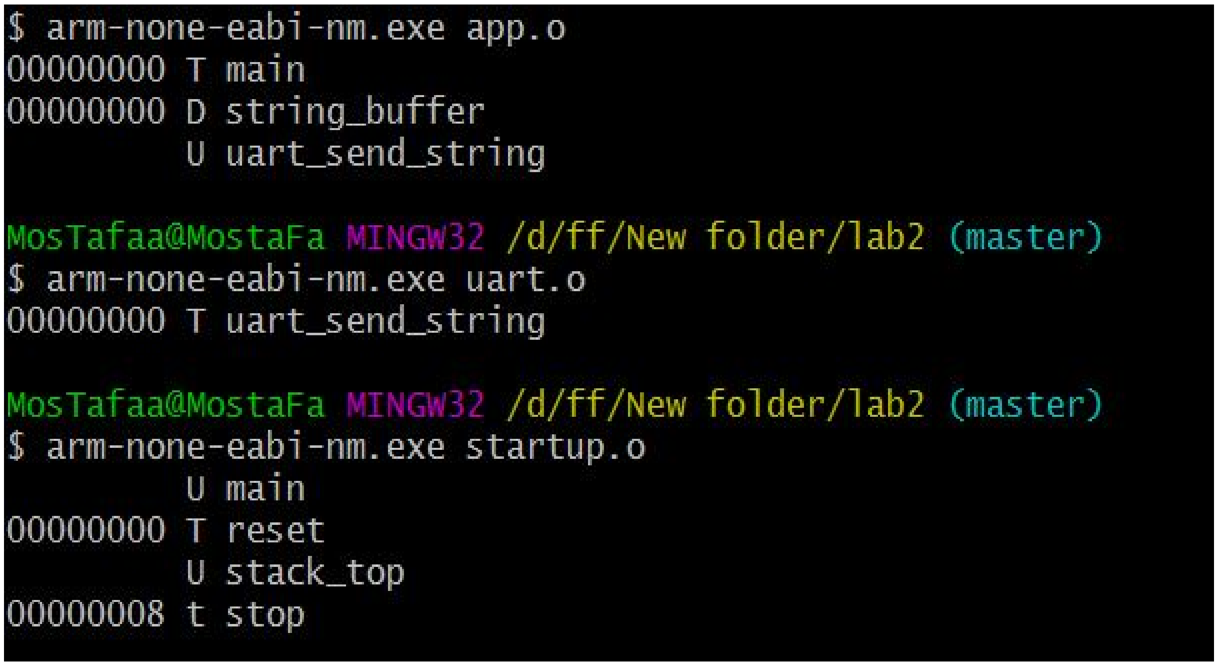


```
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         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    00000040  00000000  00000000  000000d8  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .ARM.attributes 00000022  00000000  00000000  00000118  2**0
                  CONTENTS, READONLY
```

2-Symbols Table

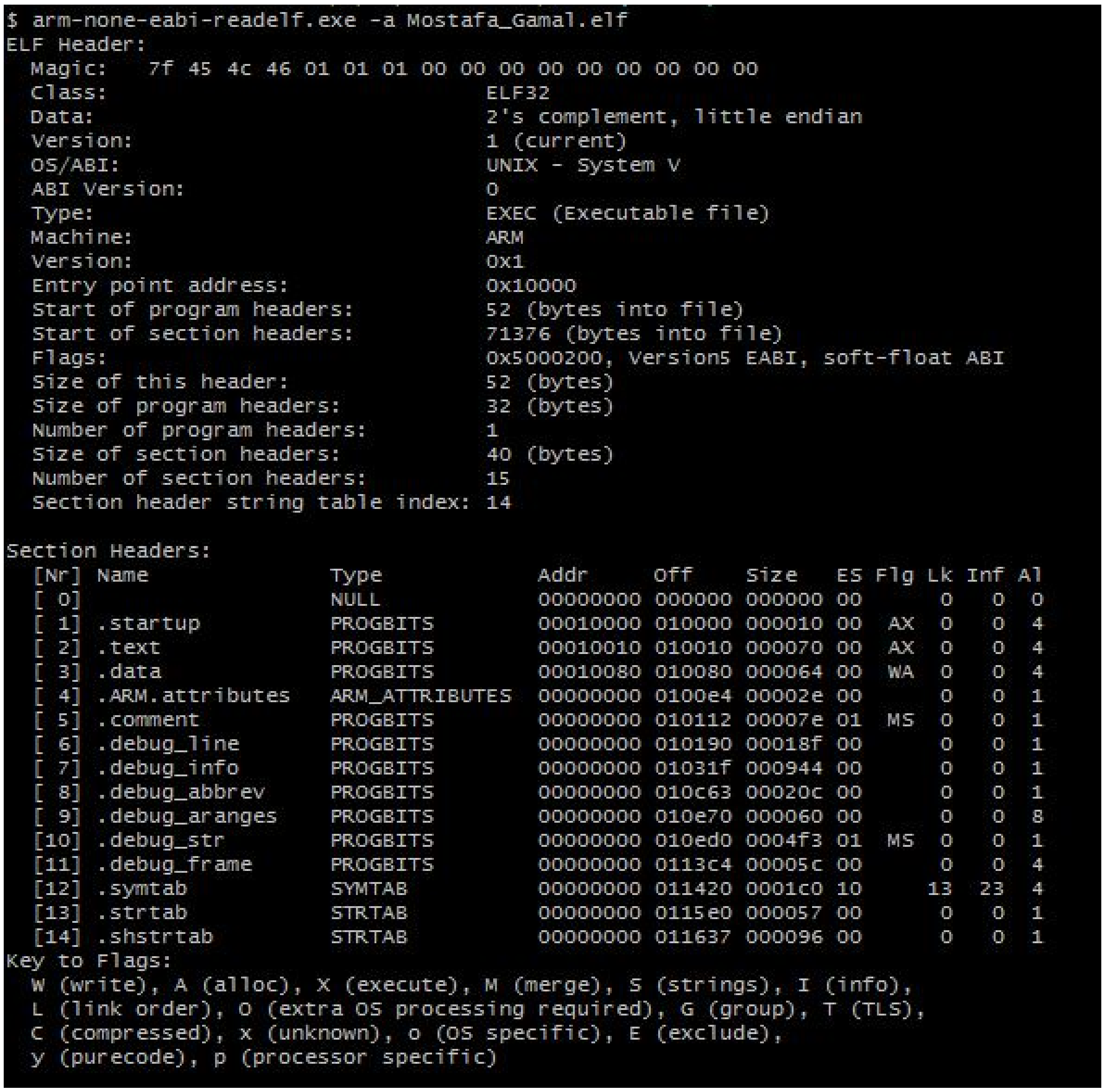


```
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
         U uart_send_string

MosTafaa@MostaFa MINGW32 /d/ff/New folder/lab2 (master)
$ arm-none-eabi-nm.exe uart.o
00000000 T uart_send_string

MosTafaa@MostaFa MINGW32 /d/ff/New folder/lab2 (master)
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

3-Analyzing elf file

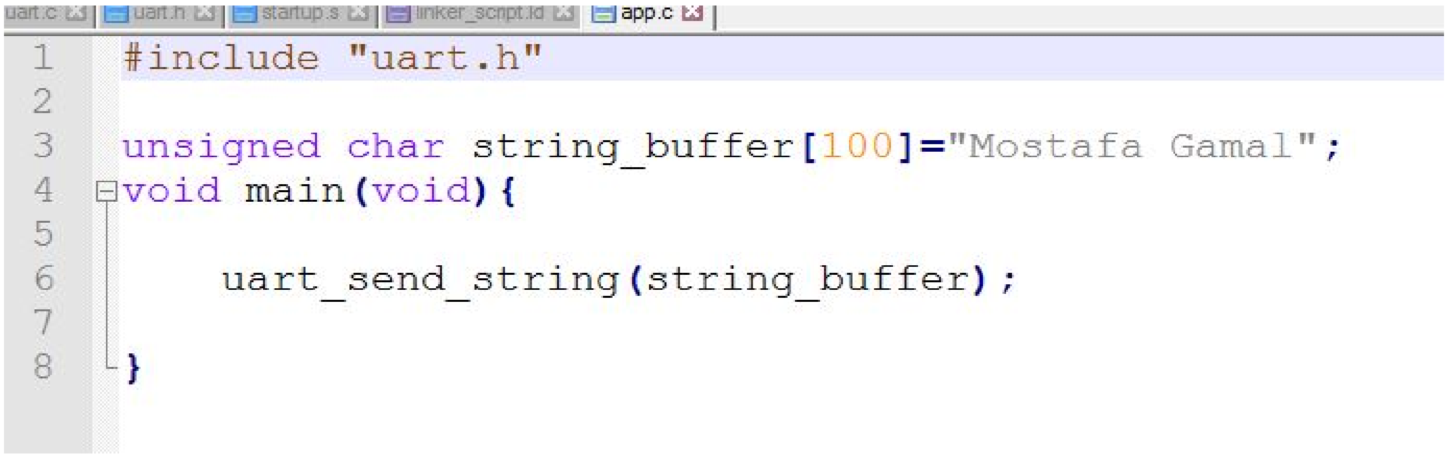


```
$ arm-none-eabi-readelf.exe -a Mostafa_Gamal.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            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:          71376 (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:         15
  Section header string table index: 14

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 000010 00  AX  0   0  4
  [ 2] .text             PROGBITS        00010010 010010 000070 00  AX  0   0  4
  [ 3] .data             PROGBITS        00010080 010080 000064 00  WA  0   0  4
  [ 4] .ARM.attributes   ARM_ATTRIBUTES  00000000 0100e4 00002e 00      0   0  1
  [ 5] .comment          PROGBITS        00000000 010112 00007e 01  MS  0   0  1
  [ 6] .debug_line       PROGBITS        00000000 010190 00018f 00      0   0  1
  [ 7] .debug_info       PROGBITS        00000000 01031f 000944 00      0   0  1
  [ 8] .debug_abbrev     PROGBITS        00000000 010c63 00020c 00      0   0  1
  [ 9] .debug_aranges    PROGBITS        00000000 010e70 000060 00      0   0  8
  [10] .debug_str        PROGBITS        00000000 010ed0 0004f3 01  MS  0   0  1
  [11] .debug_frame      PROGBITS        00000000 0113c4 00005c 00      0   0  4
  [12] .symtab           SYMTAB          00000000 011420 0001c0 10     13  23  4
  [13] .strtab           STRTAB          00000000 0115e0 000057 00      0   0  1
  [14] .shstrtab         STRTAB          00000000 011637 000096 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings), I (info),
  L (link order), O (extra OS processing required), G (group), T (TLS),
  C (compressed), x (unknown), o (OS specific), E (exclude),
  y (purecode), p (processor specific)
```

Codes

app.c



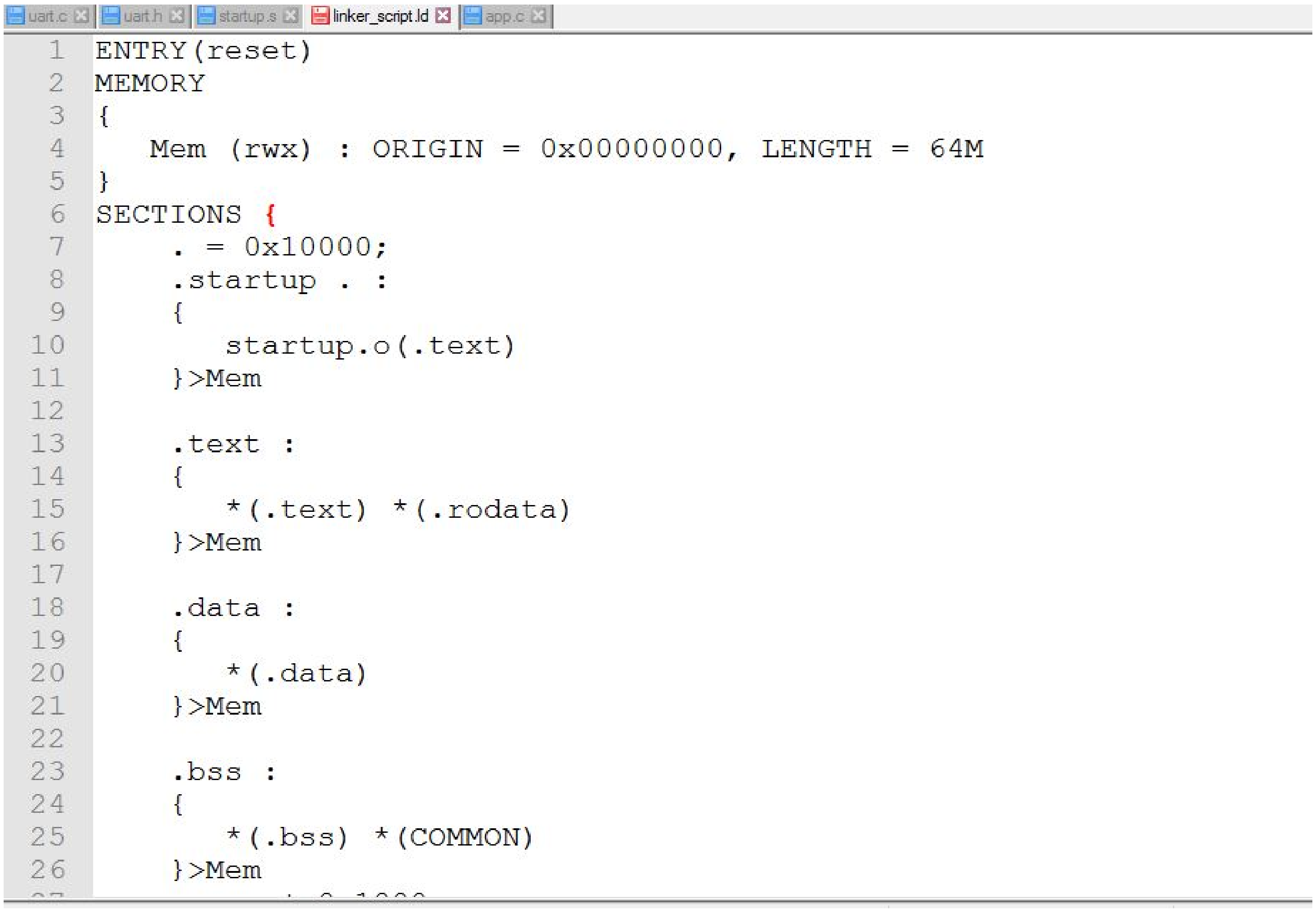
```c
#include "uart.h"

unsigned char string_buffer[100]="Mostafa Gamal";
void main(void){

    uart_send_string(string_buffer);

}
```

Linker_script



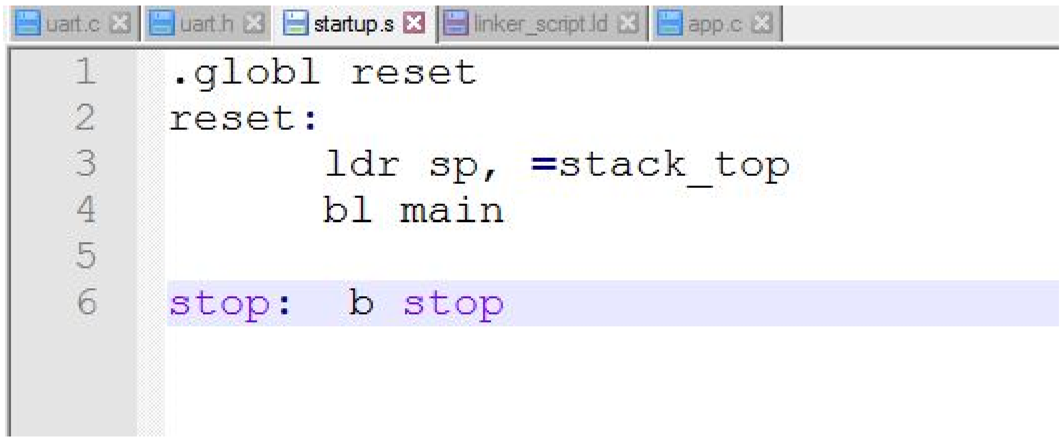
```
ENTRY(reset)
MEMORY
{
   Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
}
SECTIONS {
      . = 0x10000;
      .startup . :
      {
         startup.o(.text)
      }>Mem

      .text :
      {
         *(.text) *(.rodata)
      }>Mem

      .data :
      {
         *(.data)
      }>Mem

      .bss :
      {
         *(.bss) *(COMMON)
      }>Mem
```

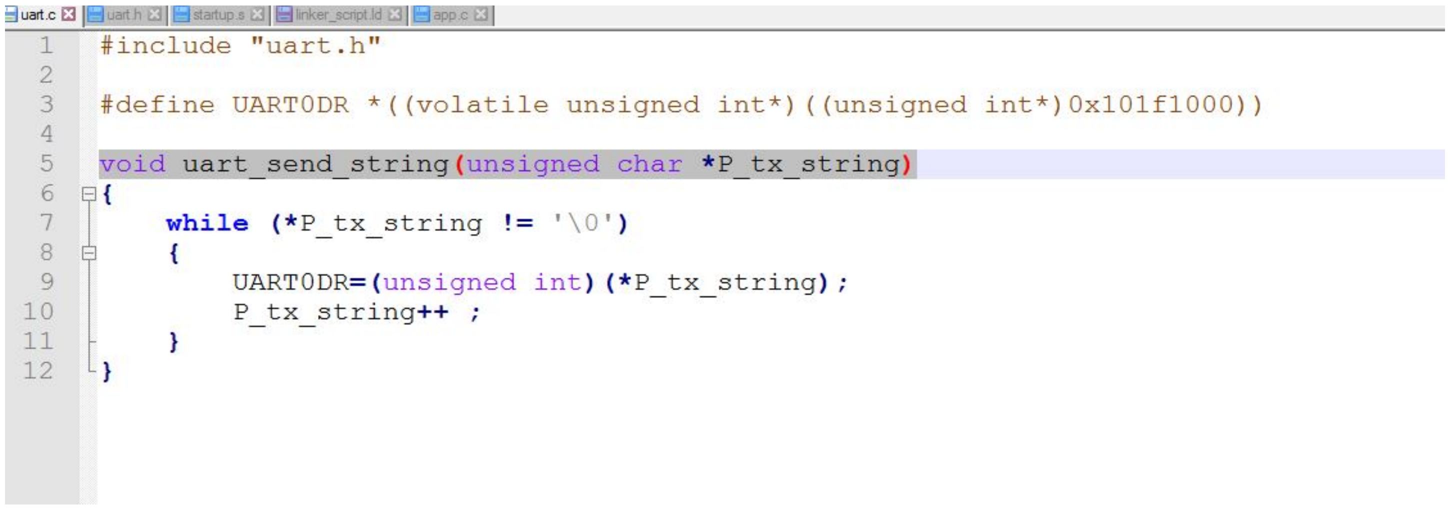
Startup.s



```
.globl reset
reset:
        ldr sp, =stack_top
        bl main

stop:  b stop
```

Uart.h

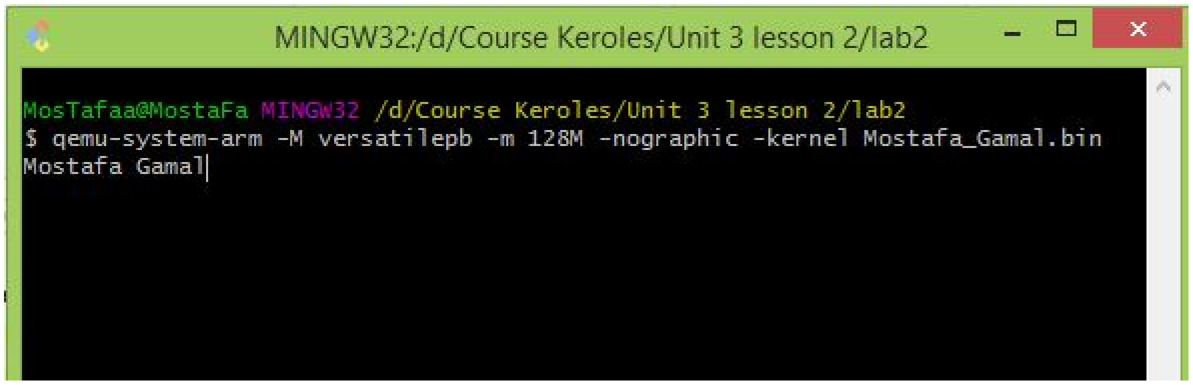


```
#include "uart.h"

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

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

Running on Qemu



```
MosTafaa@MostaFa MINGW32 /d/Course Keroles/Unit 3 lesson 2/lab2
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel Mostafa_Gamal.bin
Mostafa Gamal
```