Embedded C Lesson 2

Object Files

Uart.o

- **Header contents**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded_
$ 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         00000050  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000084  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000084  2**0
                  ALLOC
  3 .comment      00000012  00000000  00000000  00000084  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000096  2**0
                  CONTENTS, READONLY
```

- **Symbols**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64
$ arm-none-eabi-nm.exe uart.o
00000000 T sendData
```

App.o

- **Header contents**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded_
$ 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         00000020  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  00000054  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b8  2**0
                  ALLOC
  3 .comment      00000012  00000000  00000000  000000b8  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  000000ca  2**0
                  CONTENTS, READONLY
```

- **Symbols**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64
$ arm-none-eabi-nm.exe app.o
00000000 T main
         U sendData
00000000 D str_data
```

Startup.o

- **Header contents**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded
$ 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**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

Output Files

Output.elf

- **Header contents**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded
$ arm-none-eabi-objdump.exe -h output.elf

output.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         00000070  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  00010080  00010080  00008080  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000080e4  2**0
                  CONTENTS, READONLY
  4 .comment      00000011  00000000  00000000  00008112  2**0
                  CONTENTS, READONLY
```

- **Symbols**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64
$ arm-none-eabi-nm.exe output.elf
00010010 T main
00010000 T reset
00010030 T sendData
000110e4 D stack_top
00010008 t stop
00010080 D str_data
```

- **Content of ELF file**

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded_Systems_Projects/
$ arm-none-eabi-readelf.exe -a output.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:          33132 (bytes into file)
  Flags:                             0x5000002, has entry point, Version5 EABI
  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: 6

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 008000 000010 00  AX  0   0  4
  [ 2] .text             PROGBITS        00010010 008010 000070 00  AX  0   0  4
  [ 3] .data             PROGBITS        00010080 008080 000064 00  WA  0   0  4
  [ 4] .ARM.attributes   ARM_ATTRIBUTES  00000000 0080e4 00002e 00      0   0  1
  [ 5] .comment          PROGBITS        00000000 008112 000011 01  MS  0   0  1
  [ 6] .shstrtab         STRTAB          00000000 008123 000049 00      0   0  1
  [ 7] .symtab           SYMTAB          00000000 0082d4 000170 10      8  18  4
  [ 8] .strtab           STRTAB          00000000 008444 00004a 00      0   0  1
```

Final Output

```
Mohamed Ramadan@DESKTOP-SC9BGHC MINGW64 /d/Embedded systems course/Embedded_
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel output.bin
learn-in-depth:<Mohamed>
```