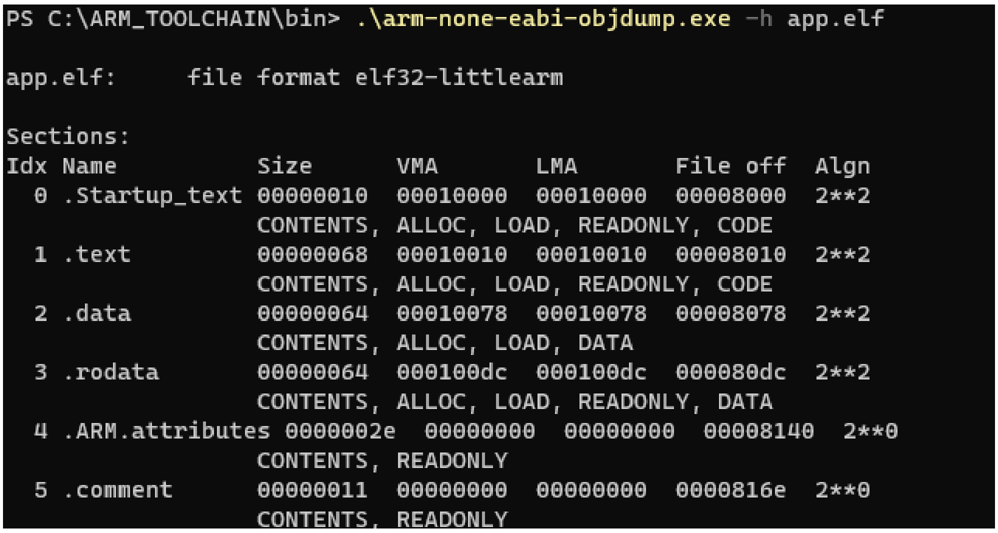
Unit 3 Lesson 2

Lab 1

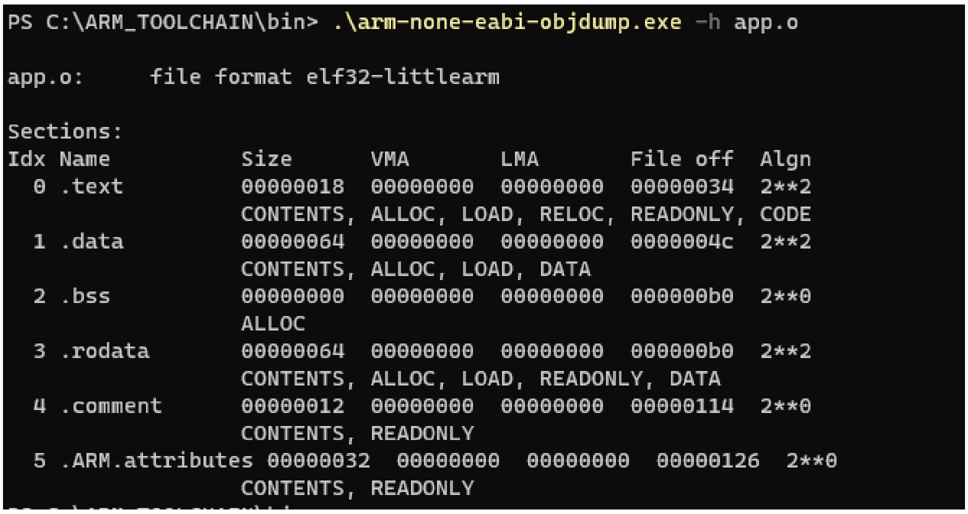
Sections:

App.elf



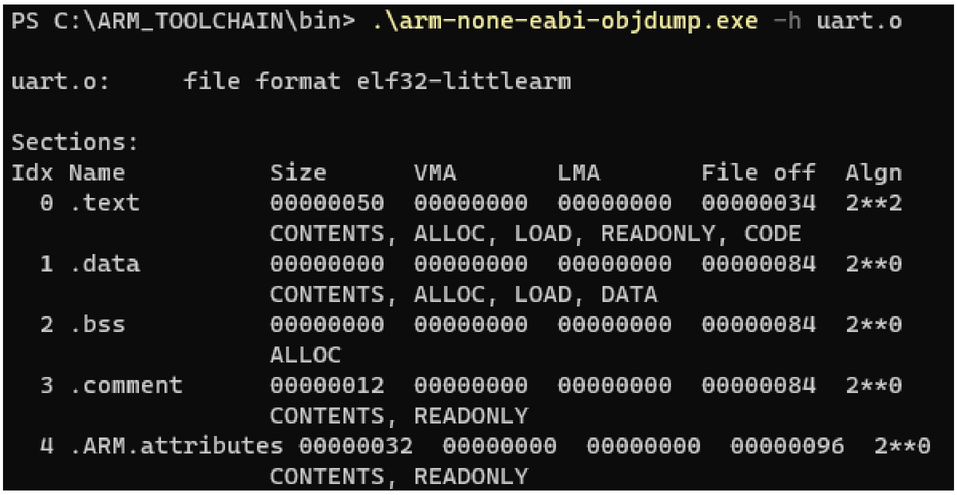
```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -h app.elf

app.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .Startup_text 00000010  00010000  00010000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000068  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  00010078  00010078  00008078  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .rodata       00000064  000100dc  000100dc  000080dc  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .ARM.attributes 0000002e  00000000  00000000  00008140  2**0
                  CONTENTS, READONLY
  5 .comment      00000011  00000000  00000000  0000816e  2**0
                  CONTENTS, READONLY
```

App.o



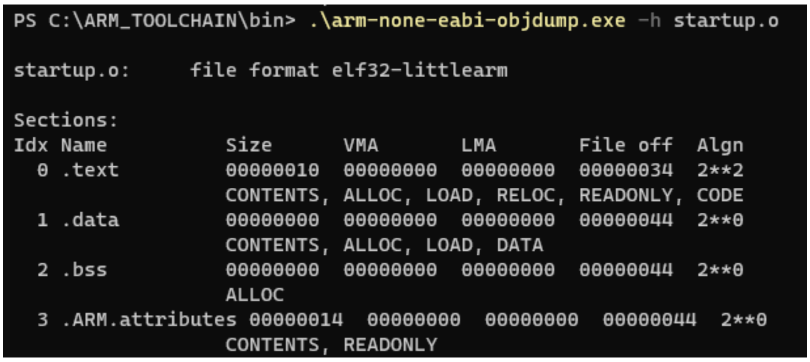
```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -h app.o

app.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000018  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000012  00000000  00000000  00000114  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000126  2**0
                  CONTENTS, READONLY
```

Uart.o



```
PS C:\ARM_TOOLCHAIN\bin> .\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
```

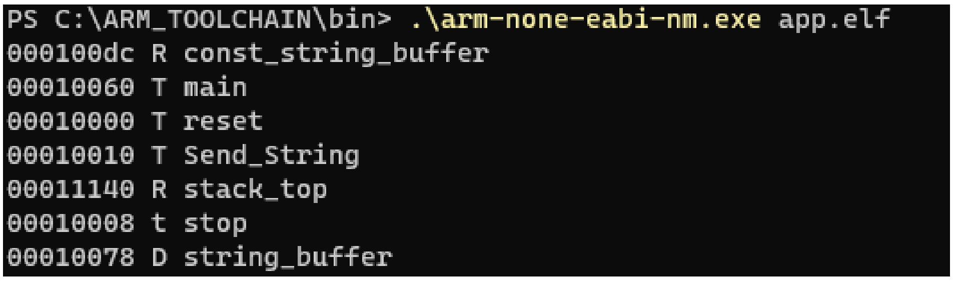
Startup.o



```
PS C:\ARM_TOOLCHAIN\bin> .\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 00000014  00000000  00000000  00000044  2**0
                  CONTENTS, READONLY
```

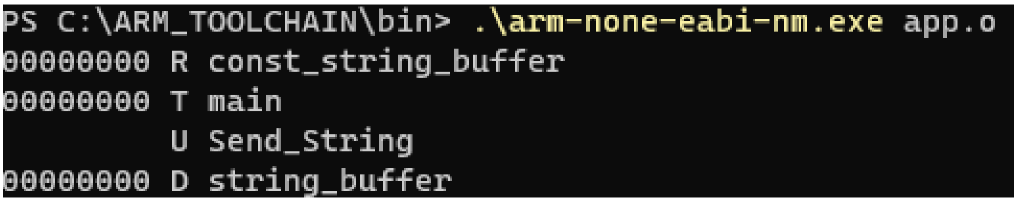
Symbol table:

App.elf



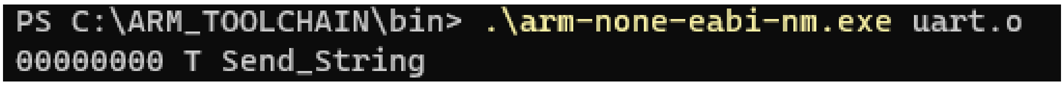
```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-nm.exe app.elf
000100dc R const_string_buffer
00010060 T main
00010000 T reset
00010010 T Send_String
00011140 R stack_top
00010008 t stop
00010078 D string_buffer
```

App.o



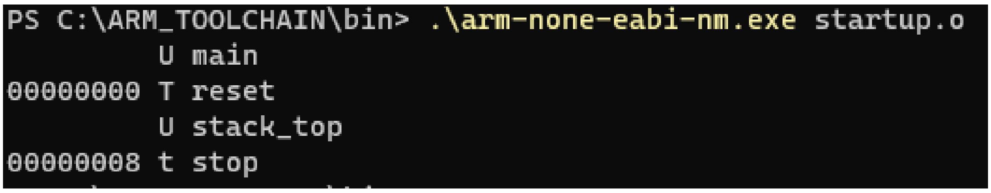
```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-nm.exe app.o
00000000 R const_string_buffer
00000000 T main
         U Send_String
00000000 D string_buffer
```

uart.o



```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-nm.exe uart.o
00000000 T Send_String
```

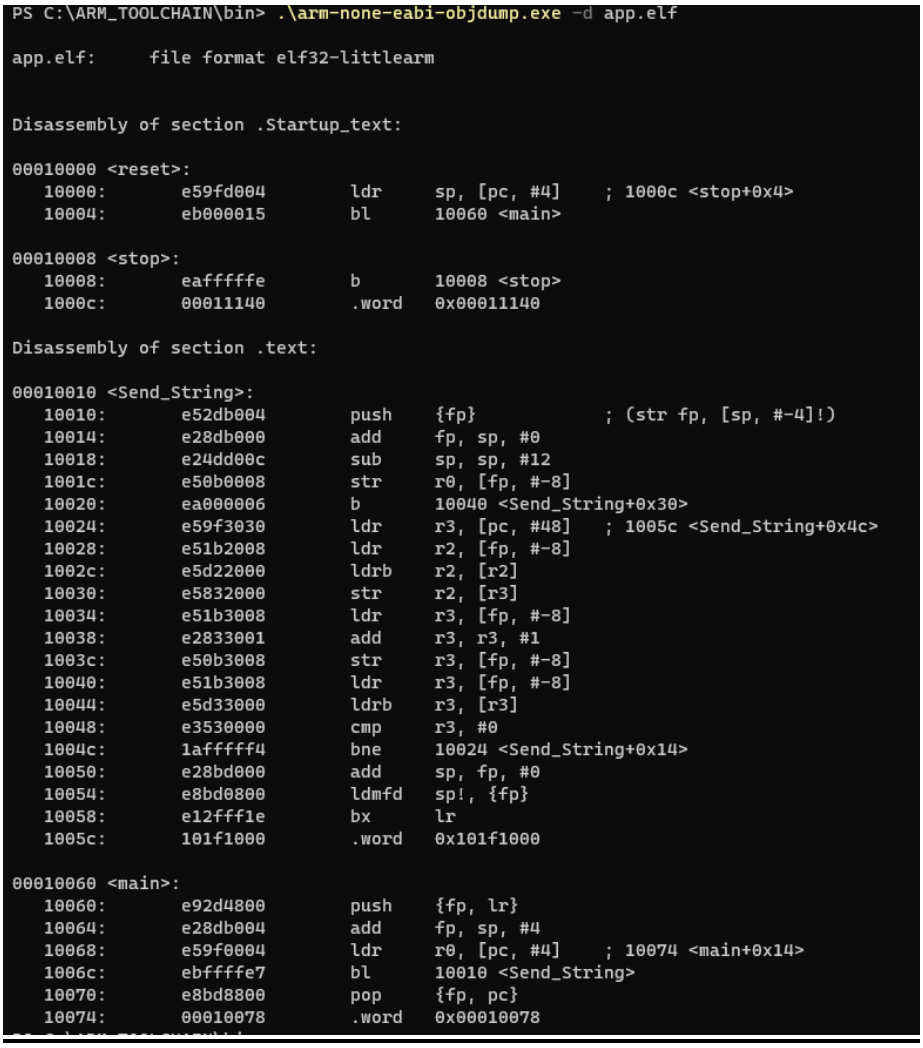
Startup.o



```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

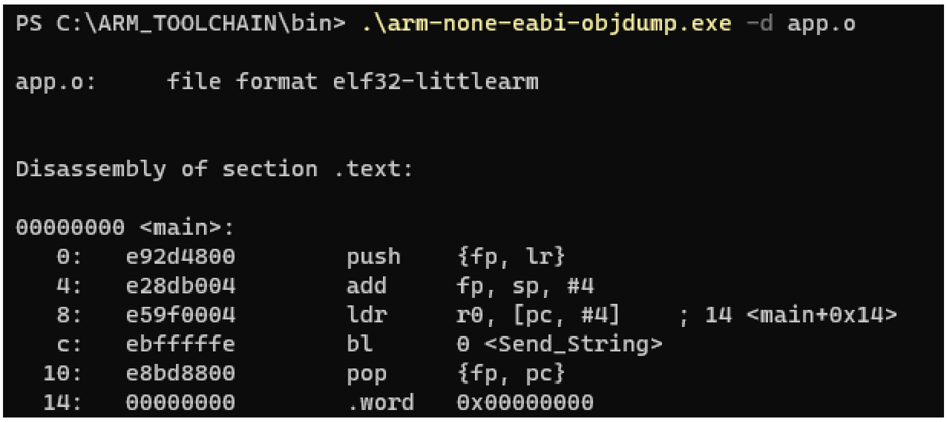
Disassembly:

App.elf



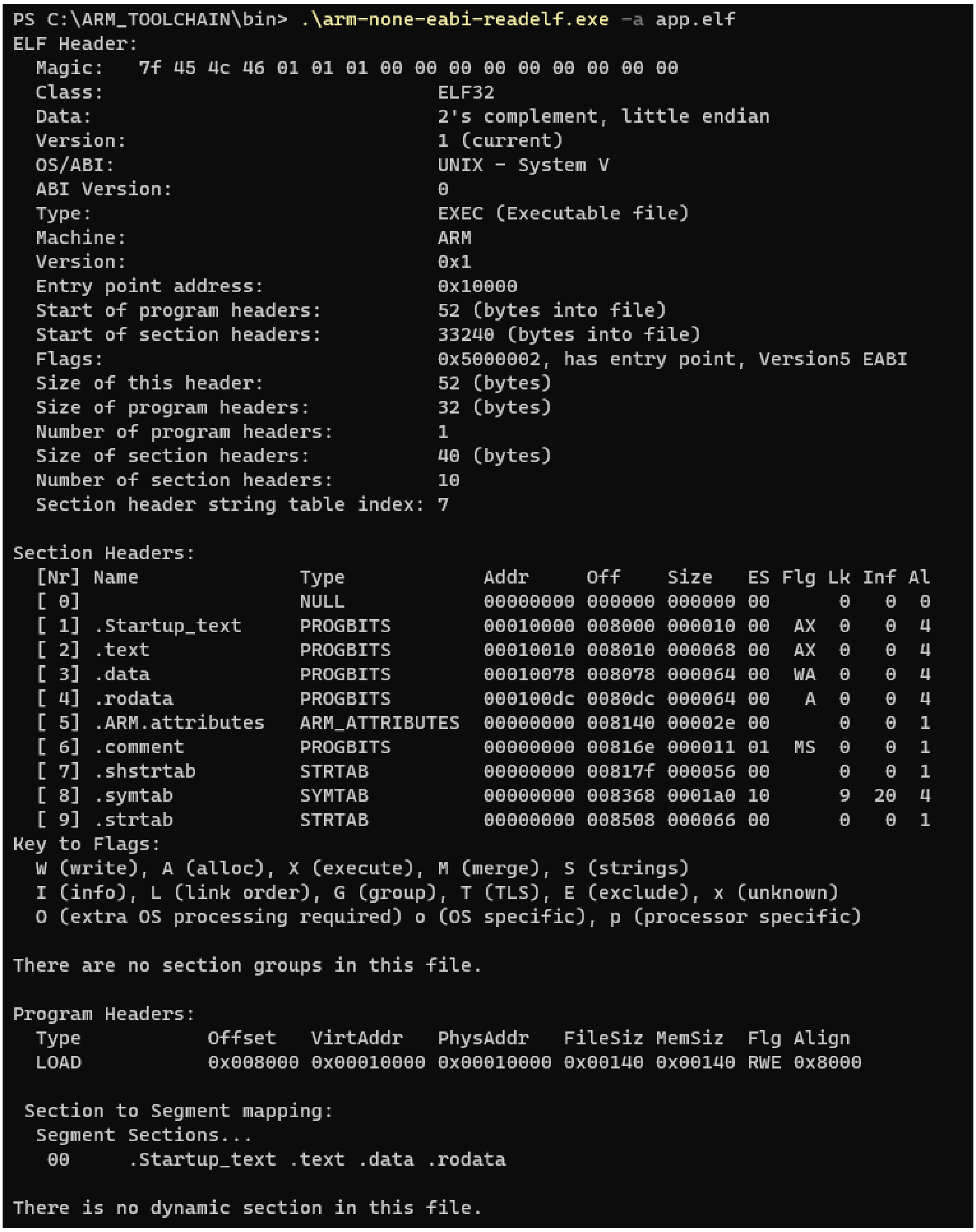
```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -d app.elf

app.elf:     file format elf32-littlearm


Disassembly of section .Startup_text:

00010000 <reset>:
   10000:       e59fd004        ldr     sp, [pc, #4]    ; 1000c <stop+0x4>
   10004:       eb000015        bl      10060 <main>

00010008 <stop>:
   10008:       eafffffe        b       10008 <stop>
   1000c:       00011140        .word   0x00011140

Disassembly of section .text:

00010010 <Send_String>:
   10010:       e52db004        push    {fp}            ; (str fp, [sp, #-4]!)
   10014:       e28db000        add     fp, sp, #0
   10018:       e24dd00c        sub     sp, sp, #12
   1001c:       e50b0008        str     r0, [fp, #-8]
   10020:       ea000006        b       10040 <Send_String+0x30>
   10024:       e59f3030        ldr     r3, [pc, #48]   ; 1005c <Send_String+0x4c>
   10028:       e51b2008        ldr     r2, [fp, #-8]
   1002c:       e5d22000        ldrb    r2, [r2]
   10030:       e5832000        str     r2, [r3]
   10034:       e51b3008        ldr     r3, [fp, #-8]
   10038:       e2833001        add     r3, r3, #1
   1003c:       e50b3008        str     r3, [fp, #-8]
   10040:       e51b3008        ldr     r3, [fp, #-8]
   10044:       e5d33000        ldrb    r3, [r3]
   10048:       e3530000        cmp     r3, #0
   1004c:       1afffff4        bne     10024 <Send_String+0x14>
   10050:       e28bd000        add     sp, fp, #0
   10054:       e8bd0800        ldmfd   sp!, {fp}
   10058:       e12fff1e        bx      lr
   1005c:       101f1000        .word   0x101f1000

00010060 <main>:
   10060:       e92d4800        push    {fp, lr}
   10064:       e28db004        add     fp, sp, #4
   10068:       e59f0004        ldr     r0, [pc, #4]    ; 10074 <main+0x14>
   1006c:       ebffffe7        bl      10010 <Send_String>
   10070:       e8bd8800        pop     {fp, pc}
   10074:       00010078        .word   0x00010078
```

App.o

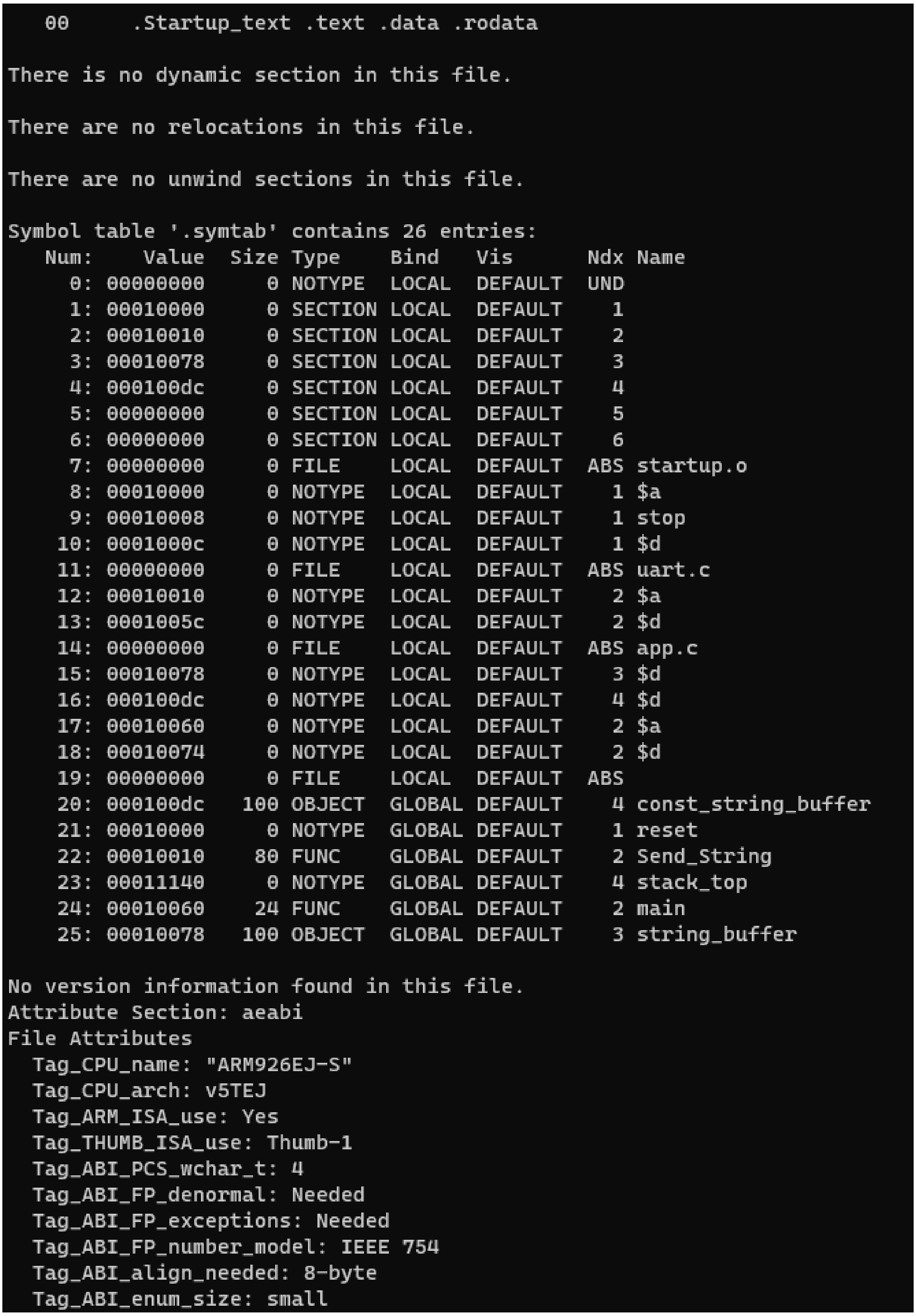


```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -d app.o

app.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <main>:
   0:   e92d4800        push    {fp, lr}
   4:   e28db004        add     fp, sp, #4
   8:   e59f0004        ldr     r0, [pc, #4]    ; 14 <main+0x14>
   c:   ebfffffe        bl      0 <Send_String>
  10:   e8bd8800        pop     {fp, pc}
  14:   00000000        .word   0x00000000
```

Read elf:

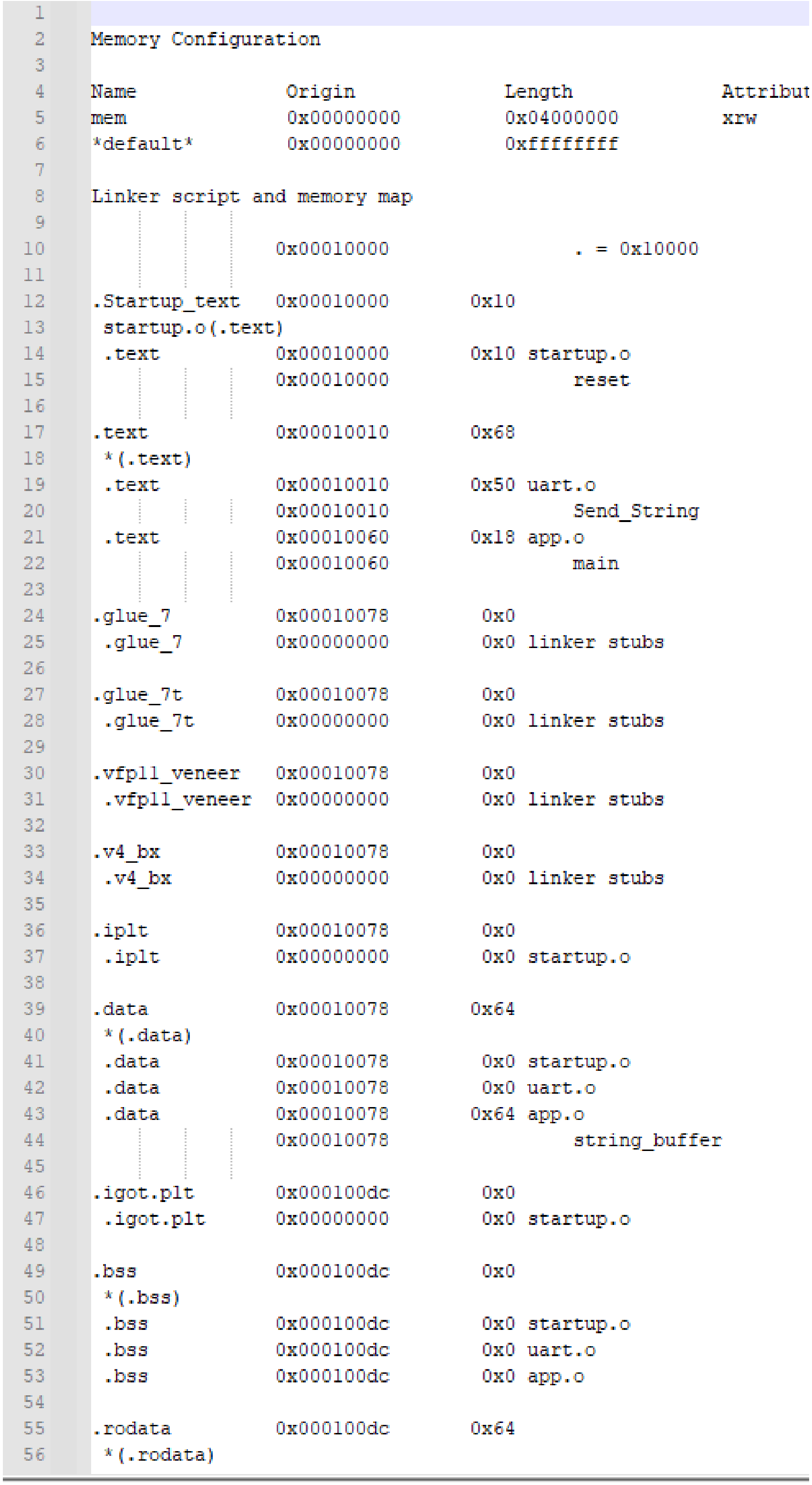


```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-readelf.exe -a app.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            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:          33240 (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:         10
  Section header string table index: 7

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .Startup_text     PROGBITS        00010000 008000 000010 00  AX  0   0  4
  [ 2] .text             PROGBITS        00010010 008010 000068 00  AX  0   0  4
  [ 3] .data             PROGBITS        00010078 008078 000064 00  WA  0   0  4
  [ 4] .rodata           PROGBITS        000100dc 0080dc 000064 00   A  0   0  4
  [ 5] .ARM.attributes   ARM_ATTRIBUTES  00000000 008140 00002e 00      0   0  1
  [ 6] .comment          PROGBITS        00000000 00816e 000011 01  MS  0   0  1
  [ 7] .shstrtab         STRTAB          00000000 00817f 000056 00      0   0  1
  [ 8] .symtab           SYMTAB          00000000 008368 0001a0 10      9  20  4
  [ 9] .strtab           STRTAB          00000000 008508 000066 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)

There are no section groups in this file.

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  LOAD           0x008000 0x00010000 0x00010000 0x00140 0x00140 RWE 0x8000

 Section to Segment mapping:
  Segment Sections...
   00     .Startup_text .text .data .rodata

There is no dynamic section in this file.
```

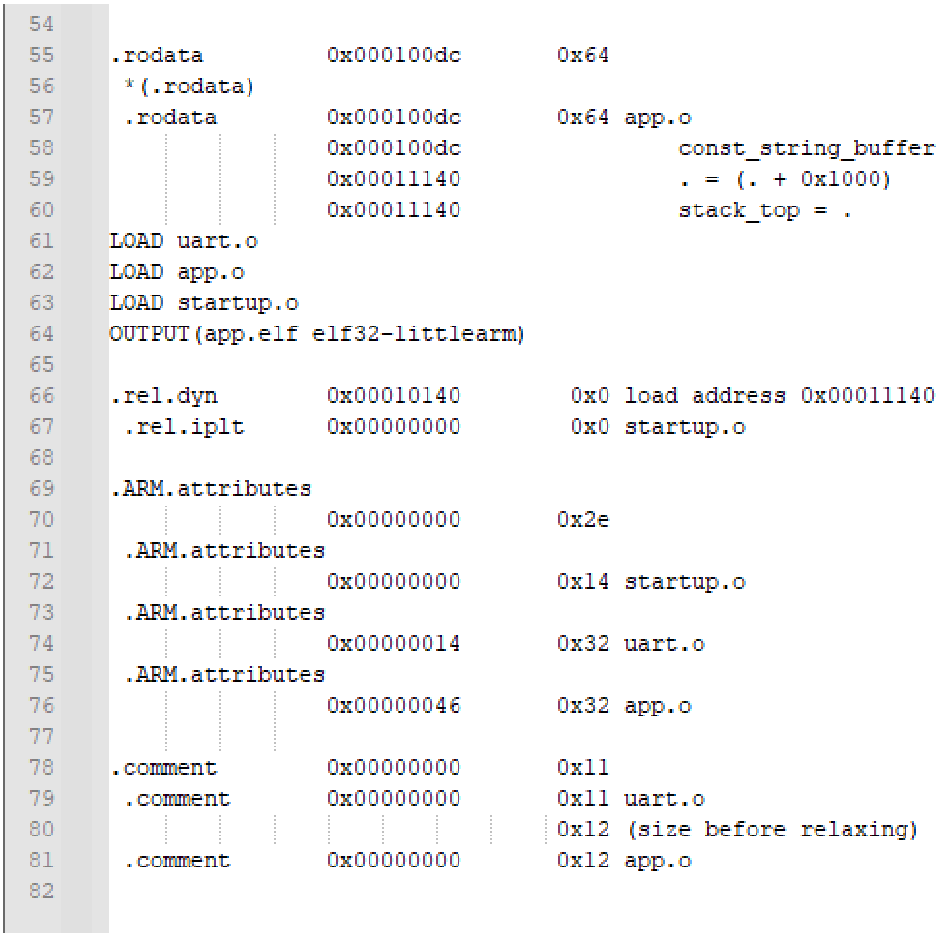


```
   00     .Startup_text .text .data .rodata

There is no dynamic section in this file.

There are no relocations in this file.

There are no unwind sections in this file.

Symbol table '.symtab' contains 26 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 00010000     0 SECTION LOCAL  DEFAULT    1
     2: 00010010     0 SECTION LOCAL  DEFAULT    2
     3: 00010078     0 SECTION LOCAL  DEFAULT    3
     4: 000100dc     0 SECTION LOCAL  DEFAULT    4
     5: 00000000     0 SECTION LOCAL  DEFAULT    5
     6: 00000000     0 SECTION LOCAL  DEFAULT    6
     7: 00000000     0 FILE    LOCAL  DEFAULT  ABS startup.o
     8: 00010000     0 NOTYPE  LOCAL  DEFAULT    1 $a
     9: 00010008     0 NOTYPE  LOCAL  DEFAULT    1 stop
    10: 0001000c     0 NOTYPE  LOCAL  DEFAULT    1 $d
    11: 00000000     0 FILE    LOCAL  DEFAULT  ABS uart.c
    12: 00010010     0 NOTYPE  LOCAL  DEFAULT    2 $a
    13: 0001005c     0 NOTYPE  LOCAL  DEFAULT    2 $d
    14: 00000000     0 FILE    LOCAL  DEFAULT  ABS app.c
    15: 00010078     0 NOTYPE  LOCAL  DEFAULT    3 $d
    16: 000100dc     0 NOTYPE  LOCAL  DEFAULT    4 $d
    17: 00010060     0 NOTYPE  LOCAL  DEFAULT    2 $a
    18: 00010074     0 NOTYPE  LOCAL  DEFAULT    2 $d
    19: 00000000     0 FILE    LOCAL  DEFAULT  ABS
    20: 000100dc   100 OBJECT  GLOBAL DEFAULT    4 const_string_buffer
    21: 00010000     0 NOTYPE  GLOBAL DEFAULT    1 reset
    22: 00010010    80 FUNC    GLOBAL DEFAULT    2 Send_String
    23: 00011140     0 NOTYPE  GLOBAL DEFAULT    4 stack_top
    24: 00010060    24 FUNC    GLOBAL DEFAULT    2 main
    25: 00010078   100 OBJECT  GLOBAL DEFAULT    3 string_buffer

No version information found in this file.
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "ARM926EJ-S"
  Tag_CPU_arch: v5TEJ
  Tag_ARM_ISA_use: Yes
  Tag_THUMB_ISA_use: Thumb-1
  Tag_ABI_PCS_wchar_t: 4
  Tag_ABI_FP_denormal: Needed
  Tag_ABI_FP_exceptions: Needed
  Tag_ABI_FP_number_model: IEEE 754
  Tag_ABI_align_needed: 8-byte
  Tag_ABI_enum_size: small
```

Mapfile:

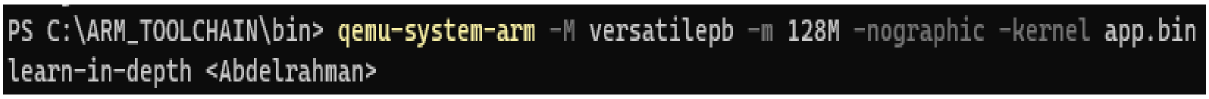


```

Memory Configuration

Name             Origin             Length             Attribut
mem              0x00000000         0x04000000         xrw
*default*        0x00000000         0xffffffff

Linker script and memory map

                0x00010000                . = 0x10000

.Startup_text   0x00010000       0x10
 startup.o(.text)
 .text          0x00010000       0x10 startup.o
                0x00010000                reset

.text           0x00010010       0x68
 *(.text)
 .text          0x00010010       0x50 uart.o
                0x00010010                Send_String
 .text          0x00010060       0x18 app.o
                0x00010060                main

.glue_7         0x00010078        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x00010078        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x00010078        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x00010078        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x00010078        0x0
 .iplt          0x00000000        0x0 startup.o

.data           0x00010078       0x64
 *(.data)
 .data          0x00010078        0x0 startup.o
 .data          0x00010078        0x0 uart.o
 .data          0x00010078       0x64 app.o
                0x00010078                string_buffer

.igot.plt       0x000100dc        0x0
 .igot.plt      0x00000000        0x0 startup.o

.bss            0x000100dc        0x0
 *(.bss)
 .bss           0x000100dc        0x0 startup.o
 .bss           0x000100dc        0x0 uart.o
 .bss           0x000100dc        0x0 app.o

.rodata         0x000100dc       0x64
 *(.rodata)
```



```

.rodata         0x000100dc        0x64
 *(.rodata)
 .rodata        0x000100dc        0x64 app.o
                0x000100dc                 const_string_buffer
                0x00011140                 . = (. + 0x1000)
                0x00011140                 stack_top = .
LOAD uart.o
LOAD app.o
LOAD startup.o
OUTPUT(app.elf elf32-littlearm)

.rel.dyn        0x00010140         0x0 load address 0x00011140
 .rel.iplt      0x00000000         0x0 startup.o

.ARM.attributes
                0x00000000        0x2e
 .ARM.attributes
                0x00000000        0x14 startup.o
 .ARM.attributes
                0x00000014        0x32 uart.o
 .ARM.attributes
                0x00000046        0x32 app.o

.comment        0x00000000        0x11
 .comment       0x00000000        0x11 uart.o
                                  0x12 (size before relaxing)
 .comment       0x00000000        0x12 app.o

```

Burn the binary on the board using qemu:



```
PS C:\ARM_TOOLCHAIN\bin> qemu-system-arm -M versatilepb -m 128M -nographic -kernel app.bin
learn-in-depth <Abdelrahman>
```