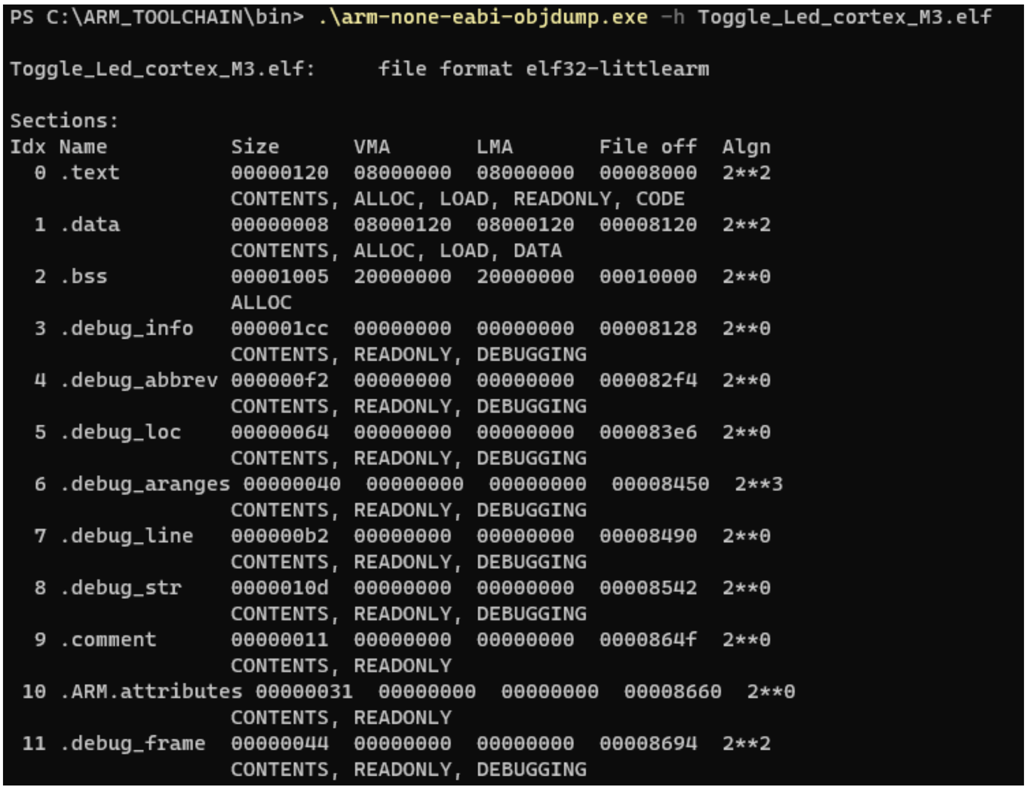
Unit 3 Lesson 3

Lab 2

Assembly Start up:

Sections:

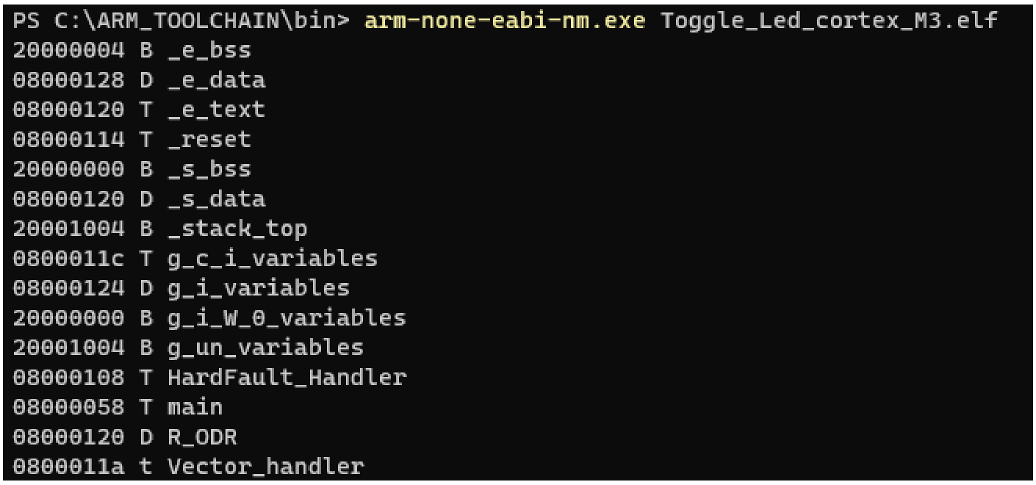


```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -h Toggle_Led_cortex_M3.elf

Toggle_Led_cortex_M3.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000120  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  08000120  08000120  00008120  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001005  20000000  20000000  00010000  2**0
                  ALLOC
  3 .debug_info   000001cc  00000000  00000000  00008128  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 000000f2  00000000  00000000  000082f4  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000064  00000000  00000000  000083e6  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000040  00000000  00000000  00008450  2**3
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000000b2  00000000  00000000  00008490  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    0000010d  00000000  00000000  00008542  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  0000864f  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000031  00000000  00000000  00008660  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000044  00000000  00000000  00008694  2**2
                  CONTENTS, READONLY, DEBUGGING
```

As we can see in the previous image the LMA and VMA of .text section is the same because both are located in the flash memory, also both address of the .data section is the same because the startup code did not take a copy of the values and kept them at the flash.

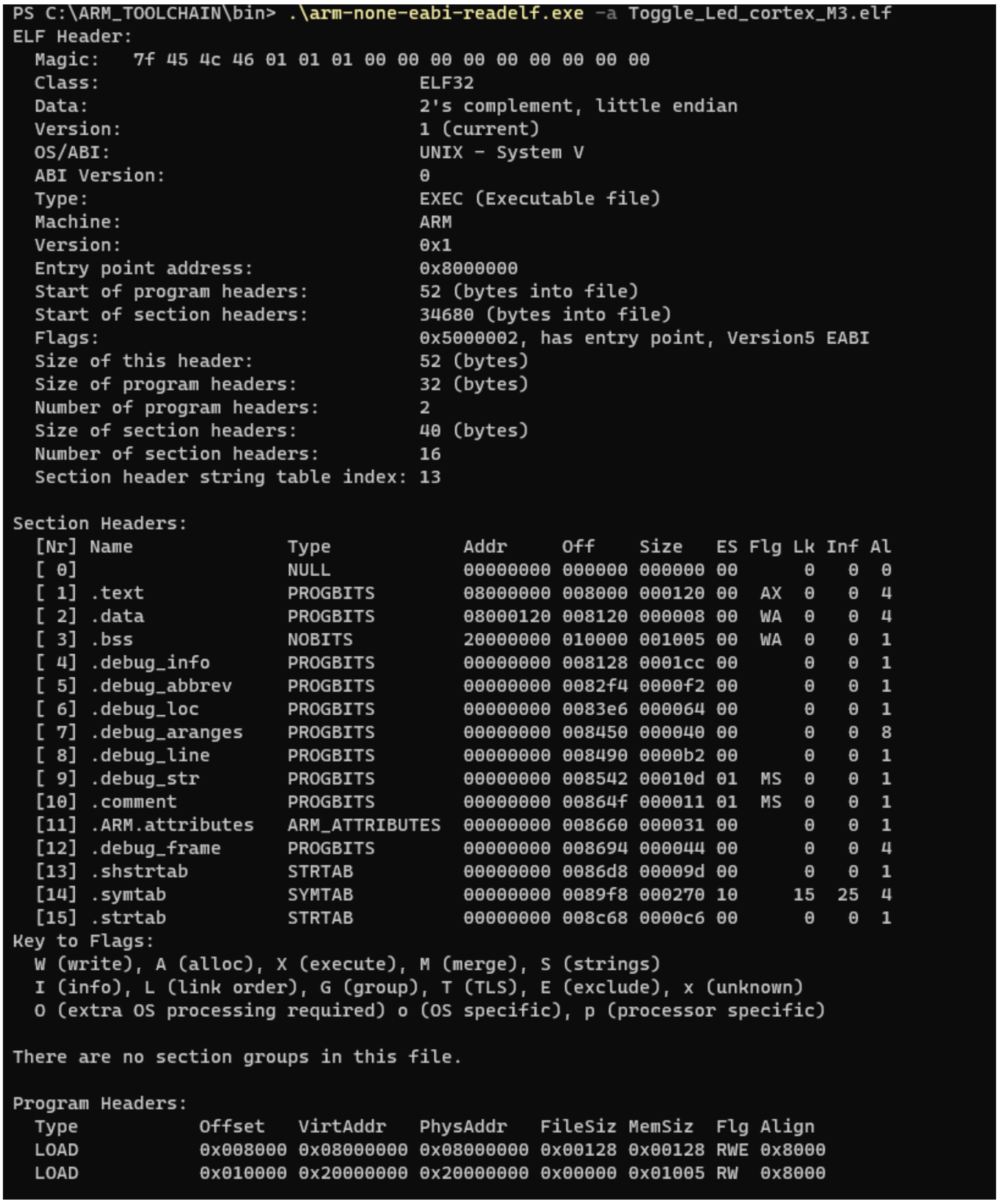
Symbol Table:

Toggle_Led_cortex_M3.elf



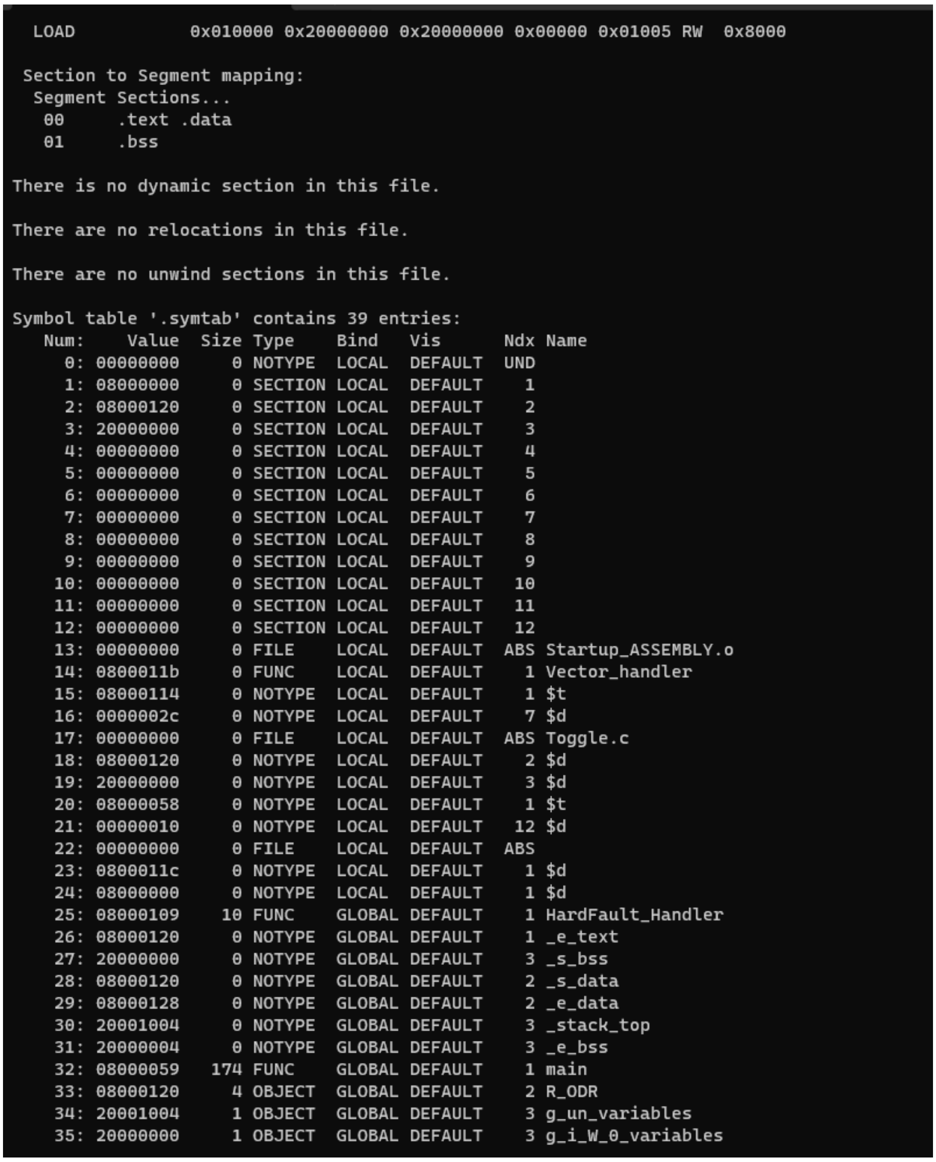
```
PS C:\ARM_TOOLCHAIN\bin> arm-none-eabi-nm.exe Toggle_Led_cortex_M3.elf
20000004 B _e_bss
08000128 D _e_data
08000120 T _e_text
08000114 T _reset
20000000 B _s_bss
08000120 D _s_data
20001004 B _stack_top
0800011c T g_c_i_variables
08000124 D g_i_variables
20000000 B g_i_W_0_variables
20001004 B g_un_variables
08000108 T HardFault_Handler
08000058 T main
08000120 D R_ODR
0800011a t Vector_handler
```

Read elf:

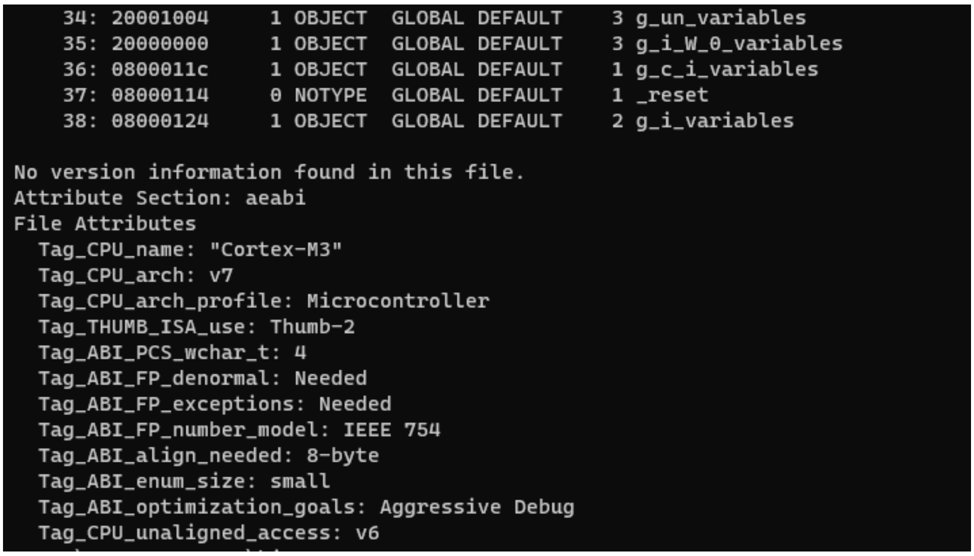


```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-readelf.exe -a Toggle_Led_cortex_M3.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x8000000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          34680 (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:         2
  Size of section headers:           40 (bytes)
  Number of section headers:         16
  Section header string table index: 13

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        08000000 008000 000120 00  AX  0   0  4
  [ 2] .data             PROGBITS        08000120 008120 000008 00  WA  0   0  4
  [ 3] .bss              NOBITS          20000000 010000 001005 00  WA  0   0  1
  [ 4] .debug_info       PROGBITS        00000000 008128 0001cc 00      0   0  1
  [ 5] .debug_abbrev     PROGBITS        00000000 0082f4 0000f2 00      0   0  1
  [ 6] .debug_loc        PROGBITS        00000000 0083e6 000064 00      0   0  1
  [ 7] .debug_aranges    PROGBITS        00000000 008450 000040 00      0   0  8
  [ 8] .debug_line       PROGBITS        00000000 008490 0000b2 00      0   0  1
  [ 9] .debug_str        PROGBITS        00000000 008542 00010d 01  MS  0   0  1
  [10] .comment          PROGBITS        00000000 00864f 000011 01  MS  0   0  1
  [11] .ARM.attributes   ARM_ATTRIBUTES  00000000 008660 000031 00      0   0  1
  [12] .debug_frame      PROGBITS        00000000 008694 000044 00      0   0  4
  [13] .shstrtab         STRTAB          00000000 0086d8 00009d 00      0   0  1
  [14] .symtab           SYMTAB          00000000 0089f8 000270 10     15  25  4
  [15] .strtab           STRTAB          00000000 008c68 0000c6 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 0x08000000 0x08000000 0x00128 0x00128 RWE 0x8000
  LOAD           0x010000 0x20000000 0x20000000 0x00000 0x01005 RW  0x8000
```

As we can see the start address of .text is at 0x08000000 which is the same as the start address of the flash memory as stated in the linker and followed by data section then the bss is alone in the SRAM at 0x20000000 as we did not move copy the data section from the flash to the RAM

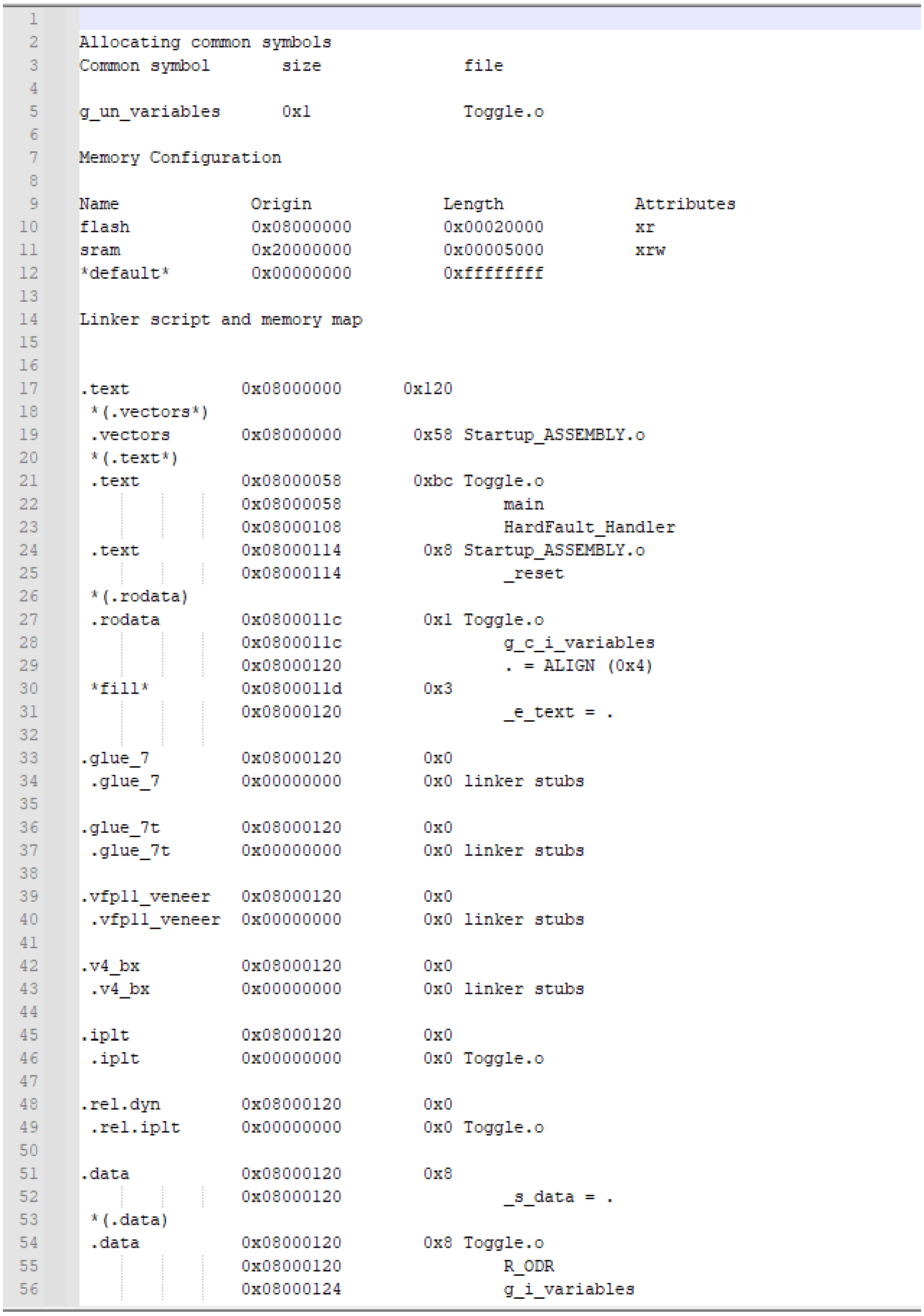


```
  LOAD           0x010000 0x20000000 0x20000000 0x00000 0x01005 RW  0x8000

 Section to Segment mapping:
  Segment Sections...
   00     .text .data
   01     .bss

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 39 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 08000000     0 SECTION LOCAL  DEFAULT    1
     2: 08000120     0 SECTION LOCAL  DEFAULT    2
     3: 20000000     0 SECTION LOCAL  DEFAULT    3
     4: 00000000     0 SECTION LOCAL  DEFAULT    4
     5: 00000000     0 SECTION LOCAL  DEFAULT    5
     6: 00000000     0 SECTION LOCAL  DEFAULT    6
     7: 00000000     0 SECTION LOCAL  DEFAULT    7
     8: 00000000     0 SECTION LOCAL  DEFAULT    8
     9: 00000000     0 SECTION LOCAL  DEFAULT    9
    10: 00000000     0 SECTION LOCAL  DEFAULT   10
    11: 00000000     0 SECTION LOCAL  DEFAULT   11
    12: 00000000     0 SECTION LOCAL  DEFAULT   12
    13: 00000000     0 FILE    LOCAL  DEFAULT  ABS Startup_ASSEMBLY.o
    14: 0800011b     0 FUNC    LOCAL  DEFAULT    1 Vector_handler
    15: 08000114     0 NOTYPE  LOCAL  DEFAULT    1 $t
    16: 0000002c     0 NOTYPE  LOCAL  DEFAULT    7 $d
    17: 00000000     0 FILE    LOCAL  DEFAULT  ABS Toggle.c
    18: 08000120     0 NOTYPE  LOCAL  DEFAULT    2 $d
    19: 20000000     0 NOTYPE  LOCAL  DEFAULT    3 $d
    20: 08000058     0 NOTYPE  LOCAL  DEFAULT    1 $t
    21: 00000010     0 NOTYPE  LOCAL  DEFAULT   12 $d
    22: 00000000     0 FILE    LOCAL  DEFAULT  ABS
    23: 0800011c     0 NOTYPE  LOCAL  DEFAULT    1 $d
    24: 08000000     0 NOTYPE  LOCAL  DEFAULT    1 $d
    25: 08000109    10 FUNC    GLOBAL DEFAULT    1 HardFault_Handler
    26: 08000120     0 NOTYPE  GLOBAL DEFAULT    1 _e_text
    27: 20000000     0 NOTYPE  GLOBAL DEFAULT    3 _s_bss
    28: 08000120     0 NOTYPE  GLOBAL DEFAULT    2 _s_data
    29: 08000128     0 NOTYPE  GLOBAL DEFAULT    2 _e_data
    30: 20001004     0 NOTYPE  GLOBAL DEFAULT    3 _stack_top
    31: 20000004     0 NOTYPE  GLOBAL DEFAULT    3 _e_bss
    32: 08000059   174 FUNC    GLOBAL DEFAULT    1 main
    33: 08000120     4 OBJECT  GLOBAL DEFAULT    2 R_ODR
    34: 20001004     1 OBJECT  GLOBAL DEFAULT    3 g_un_variables
    35: 20000000     1 OBJECT  GLOBAL DEFAULT    3 g_i_W_0_variables
```

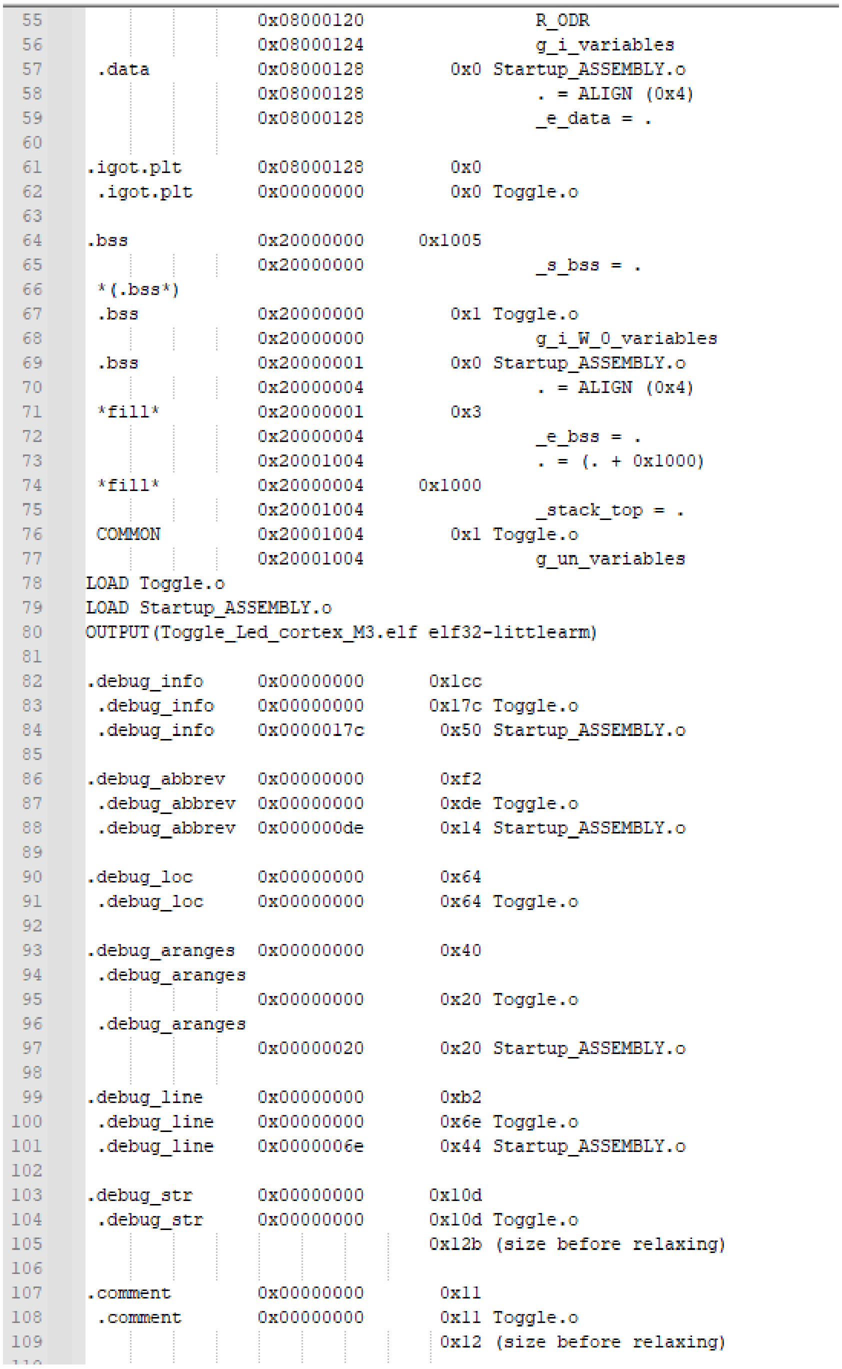


```
    34: 20001004     1 OBJECT  GLOBAL DEFAULT    3 g_un_variables
    35: 20000000     1 OBJECT  GLOBAL DEFAULT    3 g_i_W_0_variables
    36: 0800011c     1 OBJECT  GLOBAL DEFAULT    1 g_c_i_variables
    37: 08000114     0 NOTYPE  GLOBAL DEFAULT    1 _reset
    38: 08000124     1 OBJECT  GLOBAL DEFAULT    2 g_i_variables

No version information found in this file.
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "Cortex-M3"
  Tag_CPU_arch: v7
  Tag_CPU_arch_profile: Microcontroller
  Tag_THUMB_ISA_use: Thumb-2
  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
  Tag_ABI_optimization_goals: Aggressive Debug
  Tag_CPU_unaligned_access: v6
```

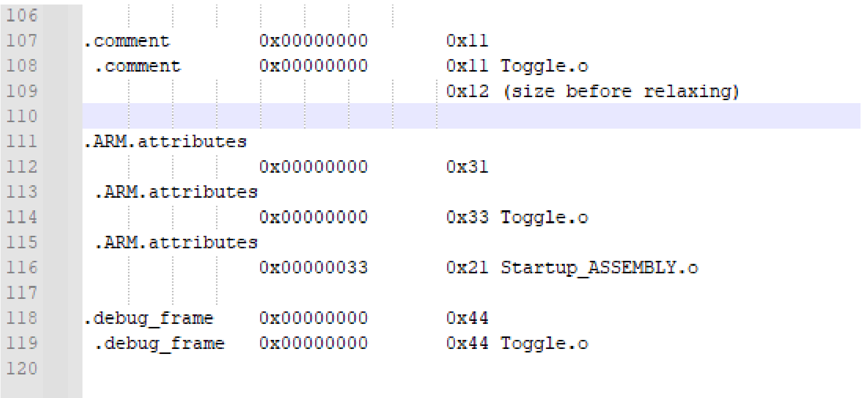
Map File:



```

Allocating common symbols
Common symbol       size              file

g_un_variables      0x1               Toggle.o

Memory Configuration

Name             Origin             Length             Attributes
flash            0x08000000         0x00020000         xr
sram             0x20000000         0x00005000         xrw
*default*        0x00000000         0xffffffff

Linker script and memory map


.text           0x08000000      0x120
 *(.vectors*)
 .vectors       0x08000000       0x58 Startup_ASSEMBLY.o
 *(.text*)
 .text          0x08000058       0xbc Toggle.o
                0x08000058                main
                0x08000108                HardFault_Handler
 .text          0x08000114        0x8 Startup_ASSEMBLY.o
                0x08000114                _reset
 *(.rodata)
 .rodata        0x0800011c        0x1 Toggle.o
                0x0800011c                g_c_i_variables
                0x08000120                . = ALIGN (0x4)
 *fill*         0x0800011d        0x3
                0x08000120                _e_text = .

.glue_7         0x08000120        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x08000120        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x08000120        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x08000120        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x08000120        0x0
 .iplt          0x00000000        0x0 Toggle.o

.rel.dyn        0x08000120        0x0
 .rel.iplt      0x00000000        0x0 Toggle.o

.data           0x08000120        0x8
                0x08000120                _s_data = .
 *(.data)
 .data          0x08000120        0x8 Toggle.o
                0x08000120                R_ODR
                0x08000124                g_i_variables
```



```
                0x08000120                R_ODR
                0x08000124                g_i_variables
 .data          0x08000128        0x0 Startup_ASSEMBLY.o
                0x08000128                . = ALIGN (0x4)
                0x08000128                _e_data = .

.igot.plt       0x08000128        0x0
 .igot.plt      0x00000000        0x0 Toggle.o

.bss            0x20000000     0x1005
                0x20000000                _s_bss = .
 *(.bss*)
 .bss           0x20000000        0x1 Toggle.o
                0x20000000                g_i_W_0_variables
 .bss           0x20000001        0x0 Startup_ASSEMBLY.o
                0x20000004                . = ALIGN (0x4)
 *fill*         0x20000001        0x3
                0x20000004                _e_bss = .
                0x20001004                . = (. + 0x1000)
 *fill*         0x20000004     0x1000
                0x20001004                _stack_top = .
 COMMON         0x20001004        0x1 Toggle.o
                0x20001004                g_un_variables
LOAD Toggle.o
LOAD Startup_ASSEMBLY.o
OUTPUT(Toggle_Led_cortex_M3.elf elf32-littlearm)

.debug_info     0x00000000      0x1cc
 .debug_info    0x00000000      0x17c Toggle.o
 .debug_info    0x0000017c       0x50 Startup_ASSEMBLY.o

.debug_abbrev   0x00000000       0xf2
 .debug_abbrev  0x00000000       0xde Toggle.o
 .debug_abbrev  0x000000de       0x14 Startup_ASSEMBLY.o

.debug_loc      0x00000000       0x64
 .debug_loc     0x00000000       0x64 Toggle.o

.debug_aranges  0x00000000       0x40
 .debug_aranges
                0x00000000       0x20 Toggle.o
 .debug_aranges
                0x00000020       0x20 Startup_ASSEMBLY.o

.debug_line     0x00000000       0xb2
 .debug_line    0x00000000       0x6e Toggle.o
 .debug_line    0x0000006e       0x44 Startup_ASSEMBLY.o

.debug_str      0x00000000      0x10d
 .debug_str     0x00000000      0x10d Toggle.o
                                0x12b (size before relaxing)

.comment        0x00000000       0x11
 .comment       0x00000000       0x11 Toggle.o
                                 0x12 (size before relaxing)
```

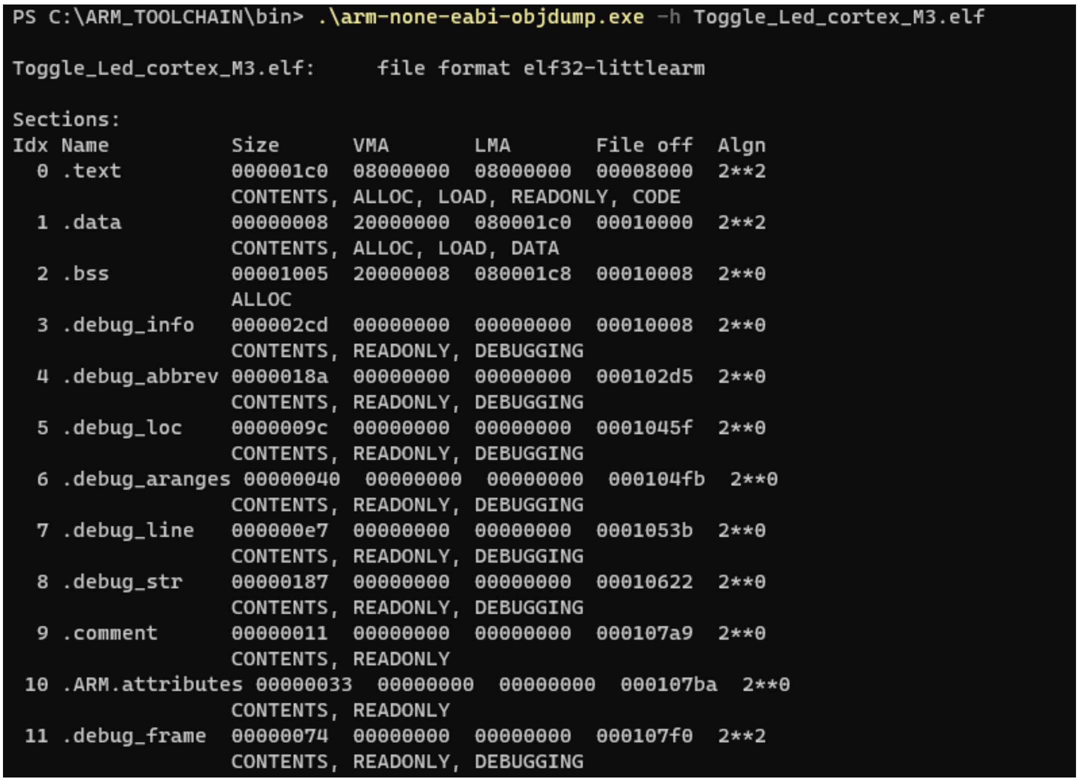


```

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

.ARM.attributes
                0x00000000       0x31
 .ARM.attributes
                0x00000000       0x33 Toggle.o
 .ARM.attributes
                0x00000033       0x21 Startup_ASSEMBLY.o

.debug_frame    0x00000000       0x44
 .debug_frame   0x00000000       0x44 Toggle.o

```

C code Start up:

Sections:

Toggle_Led_cortex_M3.elf

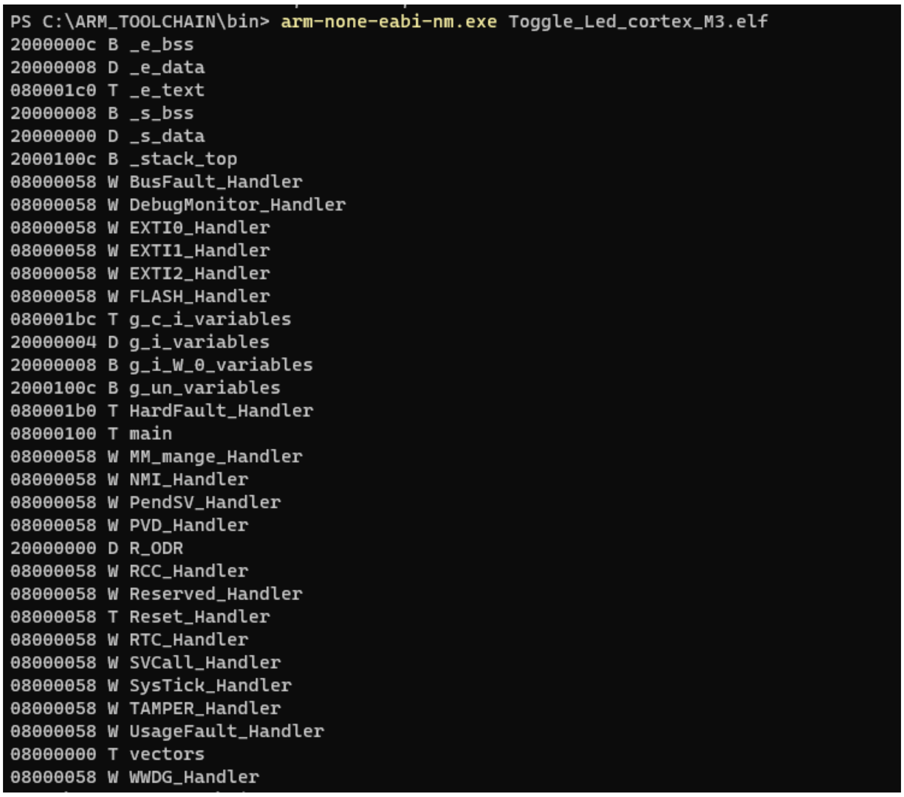


```
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-objdump.exe -h Toggle_Led_cortex_M3.elf

Toggle_Led_cortex_M3.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000001c0  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  20000000  080001c0  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001005  20000008  080001c8  00010008  2**0
                  ALLOC
  3 .debug_info   000002cd  00000000  00000000  00010008  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 0000018a  00000000  00000000  000102d5  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000009c  00000000  00000000  0001045f  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000040  00000000  00000000  000104fb  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000000e7  00000000  00000000  0001053b  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    00000187  00000000  00000000  00010622  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  000107a9  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  000107ba  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000074  00000000  00000000  000107f0  2**2
                  CONTENTS, READONLY, DEBUGGING
```

As we can see in the previous image the LMA and VMA of .text section is the same because both are located in the flash memory, on the other hand the we can detect a change on the address in the .data section because the startup code takes a copy of the values from the flash to the SRAM.

Symbol Table:

Toggle_Led_cortex_M3.elf



```
PS C:\ARM_TOOLCHAIN\bin> arm-none-eabi-nm.exe Toggle_Led_cortex_M3.elf
2000000c B _e_bss
20000008 D _e_data
080001c0 T _e_text
20000008 B _s_bss
20000000 D _s_data
2000100c B _stack_top
08000058 W BusFault_Handler
08000058 W DebugMonitor_Handler
08000058 W EXTI0_Handler
08000058 W EXTI1_Handler
08000058 W EXTI2_Handler
08000058 W FLASH_Handler
080001bc T g_c_i_variables
20000004 D g_i_variables
20000008 B g_i_W_0_variables
2000100c B g_un_variables
080001b0 T HardFault_Handler
08000100 T main
08000058 W MM_mange_Handler
08000058 W NMI_Handler
08000058 W PendSV_Handler
08000058 W PVD_Handler
20000000 D R_ODR
08000058 W RCC_Handler
08000058 W Reserved_Handler
08000058 T Reset_Handler
08000058 W RTC_Handler
08000058 W SVCall_Handler
08000058 W SysTick_Handler
08000058 W TAMPER_Handler
08000058 W UsageFault_Handler
08000000 T vectors
08000058 W WWDG_Handler
```

This image shows the symbols found in the elf file and their address as we can see most of the handlers “Interrupt vector table” the not used ones are under the same address as we used the preprocessor “alias” and the rest is overwritten because of the usage of “weak” to preserve the memory.

Where,

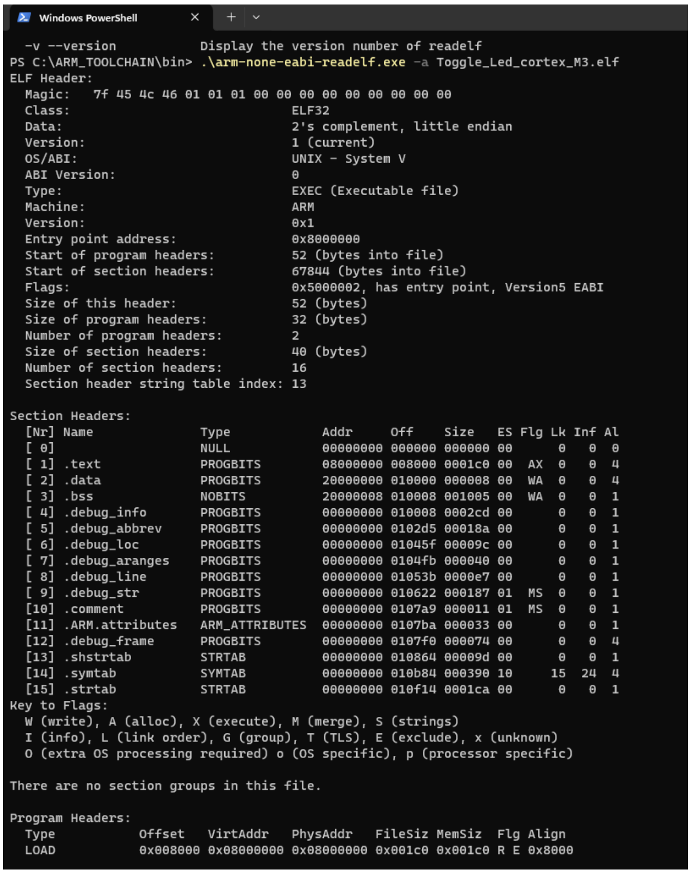
T is abbreviation for .text (code)

B is abbreviation for .bss (uninitialized data)

D is abbreviation for .Data (initialized data)

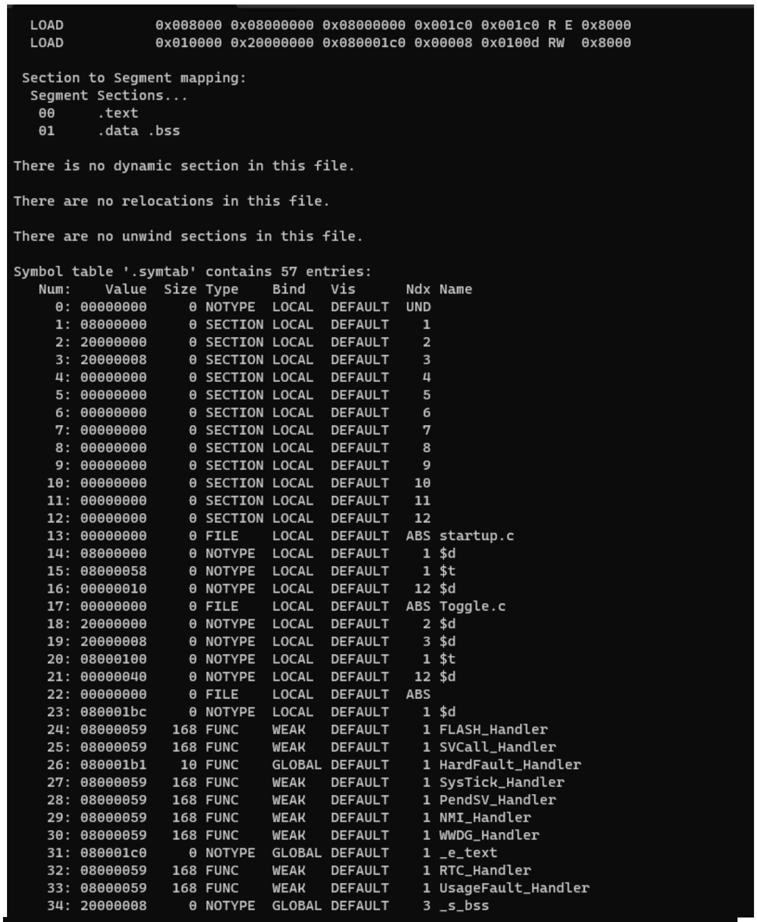
W is abbreviation for Weak Symbol

Read elf:

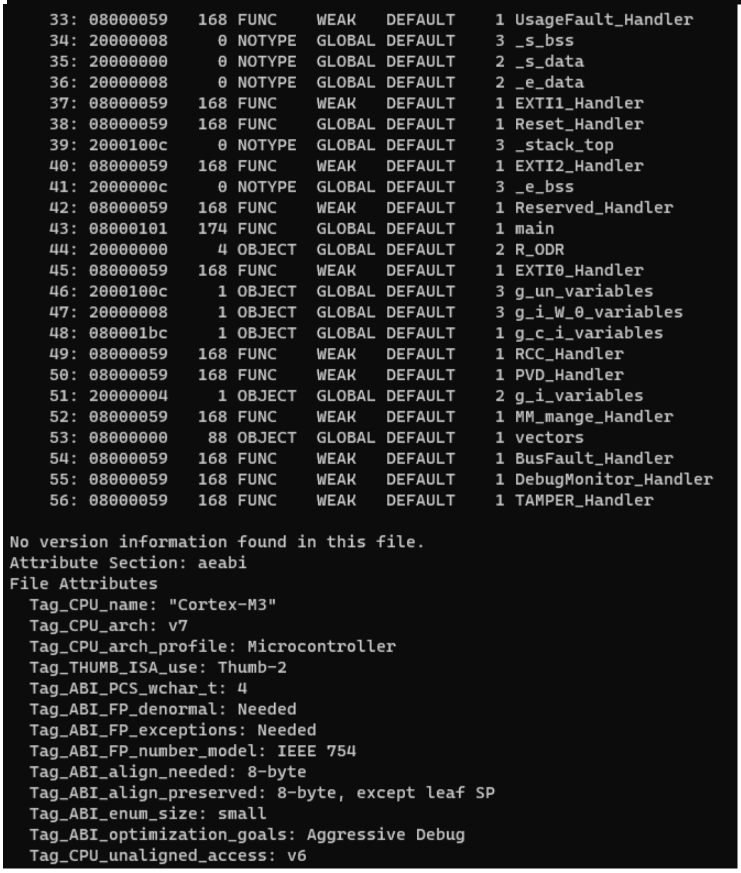


```
  -v --version           Display the version number of readelf
PS C:\ARM_TOOLCHAIN\bin> .\arm-none-eabi-readelf.exe -a Toggle_Led_cortex_M3.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's 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:               0x8000000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          67844 (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:         2
  Size of section headers:           40 (bytes)
  Number of section headers:         16
  Section header string table index: 13

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        08000000 008000 0001c0 00  AX  0   0  4
  [ 2] .data             PROGBITS        20000000 010000 000008 00  WA  0   0  4
  [ 3] .bss              NOBITS          20000008 010008 001005 00  WA  0   0  1
  [ 4] .debug_info       PROGBITS        00000000 010008 0002cd 00      0   0  1
  [ 5] .debug_abbrev     PROGBITS        00000000 0102d5 00018a 00      0   0  1
  [ 6] .debug_loc        PROGBITS        00000000 01045f 00009c 00      0   0  1
  [ 7] .debug_aranges    PROGBITS        00000000 0104fb 000040 00      0   0  1
  [ 8] .debug_line       PROGBITS        00000000 01053b 0000e7 00      0   0  1
  [ 9] .debug_str        PROGBITS        00000000 010622 000187 01  MS  0   0  1
  [10] .comment          PROGBITS        00000000 0107a9 000011 01  MS  0   0  1
  [11] .ARM.attributes   ARM_ATTRIBUTES  00000000 0107ba 000033 00      0   0  1
  [12] .debug_frame      PROGBITS        00000000 0107f0 000074 00      0   0  4
  [13] .shstrtab         STRTAB          00000000 010864 00009d 00      0   0  1
  [14] .symtab           SYMTAB          00000000 010b84 000390 10     15  24  4
  [15] .strtab           STRTAB          00000000 010f14 0001ca 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 0x08000000 0x08000000 0x001c0 0x001c0 R E 0x8000
```

As we can see the start address of .text is at 0x08000000 which is the same as the start address of the flash memory as stated in the linker and also the start of the data section is 0x20000000 which is the same as the start address of the SRAM memory and followed by the bss section

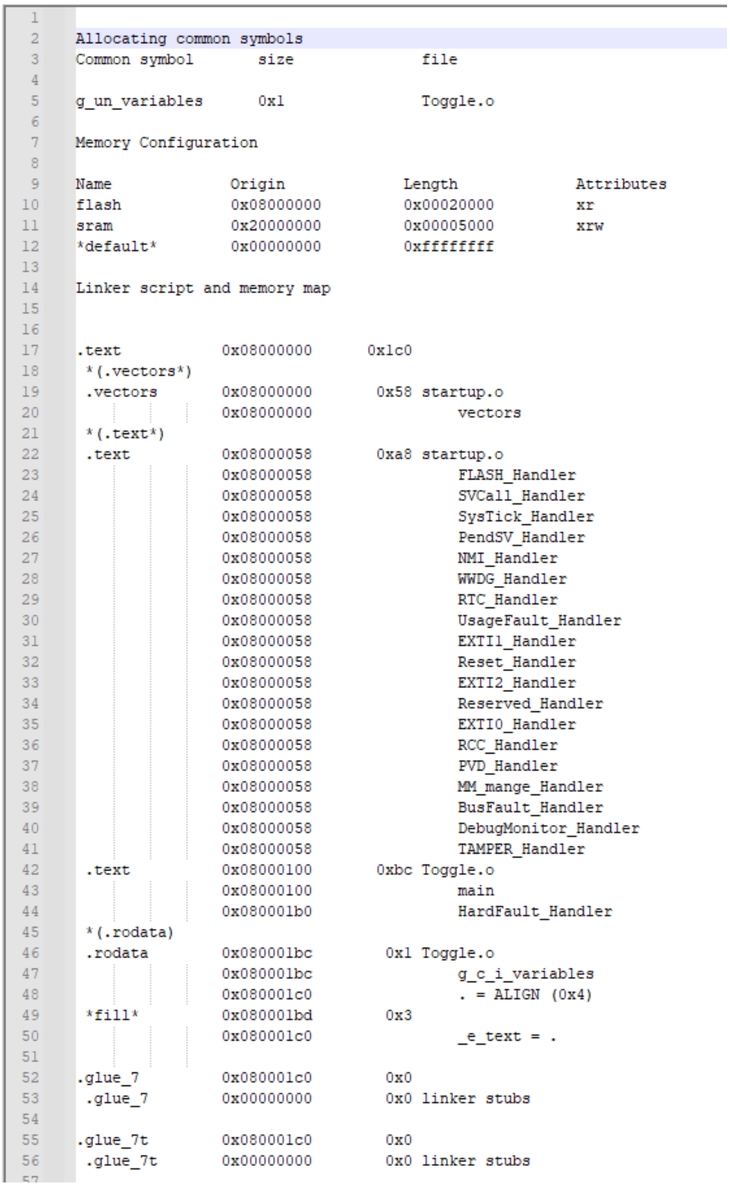


```
  LOAD           0x008000 0x08000000 0x08000000 0x001c0 0x001c0 R E 0x8000
  LOAD           0x010000 0x20000000 0x080001c0 0x00008 0x0100d RW  0x8000

 Section to Segment mapping:
  Segment Sections...
   00     .text
   01     .data .bss

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 57 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND 
     1: 08000000     0 SECTION LOCAL  DEFAULT    1 
     2: 20000000     0 SECTION LOCAL  DEFAULT    2 
     3: 20000008     0 SECTION LOCAL  DEFAULT    3 
     4: 00000000     0 SECTION LOCAL  DEFAULT    4 
     5: 00000000     0 SECTION LOCAL  DEFAULT    5 
     6: 00000000     0 SECTION LOCAL  DEFAULT    6 
     7: 00000000     0 SECTION LOCAL  DEFAULT    7 
     8: 00000000     0 SECTION LOCAL  DEFAULT    8 
     9: 00000000     0 SECTION LOCAL  DEFAULT    9 
    10: 00000000     0 SECTION LOCAL  DEFAULT   10 
    11: 00000000     0 SECTION LOCAL  DEFAULT   11 
    12: 00000000     0 SECTION LOCAL  DEFAULT   12 
    13: 00000000     0 FILE    LOCAL  DEFAULT  ABS startup.c
    14: 08000000     0 NOTYPE  LOCAL  DEFAULT    1 $d
    15: 08000058     0 NOTYPE  LOCAL  DEFAULT    1 $t
    16: 00000010     0 NOTYPE  LOCAL  DEFAULT   12 $d
    17: 00000000     0 FILE    LOCAL  DEFAULT  ABS Toggle.c
    18: 20000000     0 NOTYPE  LOCAL  DEFAULT    2 $d
    19: 20000008     0 NOTYPE  LOCAL  DEFAULT    3 $d
    20: 08000100     0 NOTYPE  LOCAL  DEFAULT    1 $t
    21: 00000040     0 NOTYPE  LOCAL  DEFAULT   12 $d
    22: 00000000     0 FILE    LOCAL  DEFAULT  ABS 
    23: 080001bc     0 NOTYPE  LOCAL  DEFAULT    1 $d
    24: 08000059   168 FUNC    WEAK   DEFAULT    1 FLASH_Handler
    25: 08000059   168 FUNC    WEAK   DEFAULT    1 SVCall_Handler
    26: 080001b1    10 FUNC    GLOBAL DEFAULT    1 HardFault_Handler
    27: 08000059   168 FUNC    WEAK   DEFAULT    1 SysTick_Handler
    28: 08000059   168 FUNC    WEAK   DEFAULT    1 PendSV_Handler
    29: 08000059   168 FUNC    WEAK   DEFAULT    1 NMI_Handler
    30: 08000059   168 FUNC    WEAK   DEFAULT    1 WWDG_Handler
    31: 080001c0     0 NOTYPE  GLOBAL DEFAULT    1 _e_text
    32: 08000059   168 FUNC    WEAK   DEFAULT    1 RTC_Handler
    33: 08000059   168 FUNC    WEAK   DEFAULT    1 UsageFault_Handler
    34: 20000008     0 NOTYPE  GLOBAL DEFAULT    3 _s_bss
```

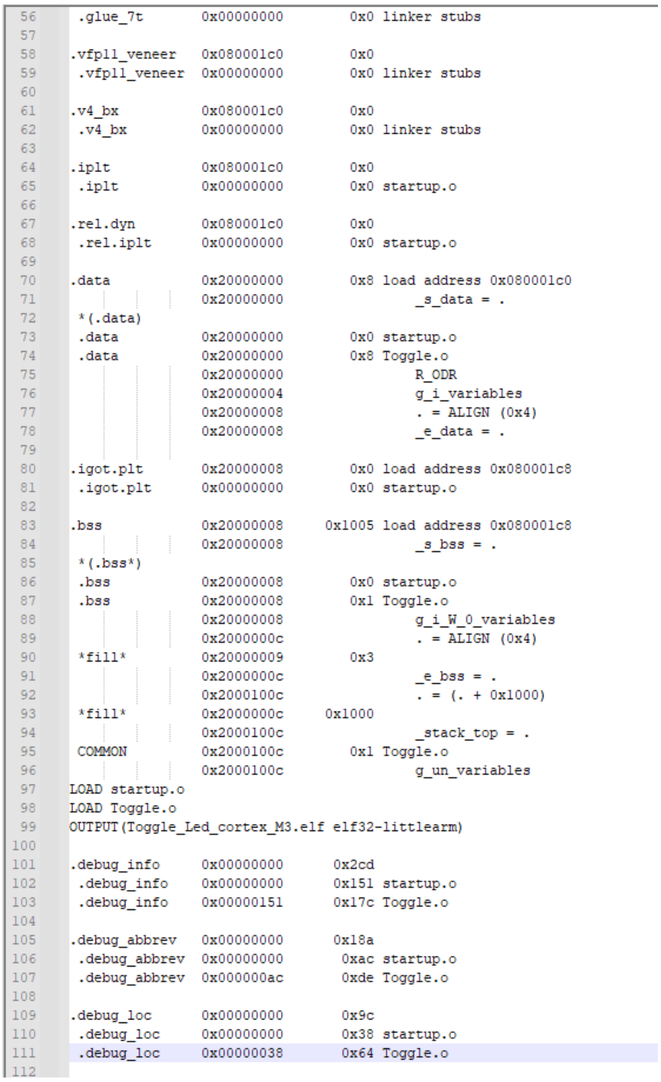


```
    33: 08000059   168 FUNC    WEAK   DEFAULT    1 UsageFault_Handler
    34: 20000008     0 NOTYPE  GLOBAL DEFAULT    3 _s_bss
    35: 20000000     0 NOTYPE  GLOBAL DEFAULT    2 _s_data
    36: 20000008     0 NOTYPE  GLOBAL DEFAULT    2 _e_data
    37: 08000059   168 FUNC    WEAK   DEFAULT    1 EXTI1_Handler
    38: 08000059   168 FUNC    GLOBAL DEFAULT    1 Reset_Handler
    39: 2000100c     0 NOTYPE  GLOBAL DEFAULT    3 _stack_top
    40: 08000059   168 FUNC    WEAK   DEFAULT    1 EXTI2_Handler
    41: 2000000c     0 NOTYPE  GLOBAL DEFAULT    3 _e_bss
    42: 08000059   168 FUNC    WEAK   DEFAULT    1 Reserved_Handler
    43: 08000101   174 FUNC    GLOBAL DEFAULT    1 main
    44: 20000000     4 OBJECT  GLOBAL DEFAULT    2 R_ODR
    45: 08000059   168 FUNC    WEAK   DEFAULT    1 EXTI0_Handler
    46: 2000100c     1 OBJECT  GLOBAL DEFAULT    3 g_un_variables
    47: 20000008     1 OBJECT  GLOBAL DEFAULT    3 g_i_W_0_variables
    48: 080001bc     1 OBJECT  GLOBAL DEFAULT    1 g_c_i_variables
    49: 08000059   168 FUNC    WEAK   DEFAULT    1 RCC_Handler
    50: 08000059   168 FUNC    WEAK   DEFAULT    1 PVD_Handler
    51: 20000004     1 OBJECT  GLOBAL DEFAULT    2 g_i_variables
    52: 08000059   168 FUNC    WEAK   DEFAULT    1 MM_mange_Handler
    53: 08000000    88 OBJECT  GLOBAL DEFAULT    1 vectors
    54: 08000059   168 FUNC    WEAK   DEFAULT    1 BusFault_Handler
    55: 08000059   168 FUNC    WEAK   DEFAULT    1 DebugMonitor_Handler
    56: 08000059   168 FUNC    WEAK   DEFAULT    1 TAMPER_Handler

No version information found in this file.
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "Cortex-M3"
  Tag_CPU_arch: v7
  Tag_CPU_arch_profile: Microcontroller
  Tag_THUMB_ISA_use: Thumb-2
  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_align_preserved: 8-byte, except leaf SP
  Tag_ABI_enum_size: small
  Tag_ABI_optimization_goals: Aggressive Debug
  Tag_CPU_unaligned_access: v6
```

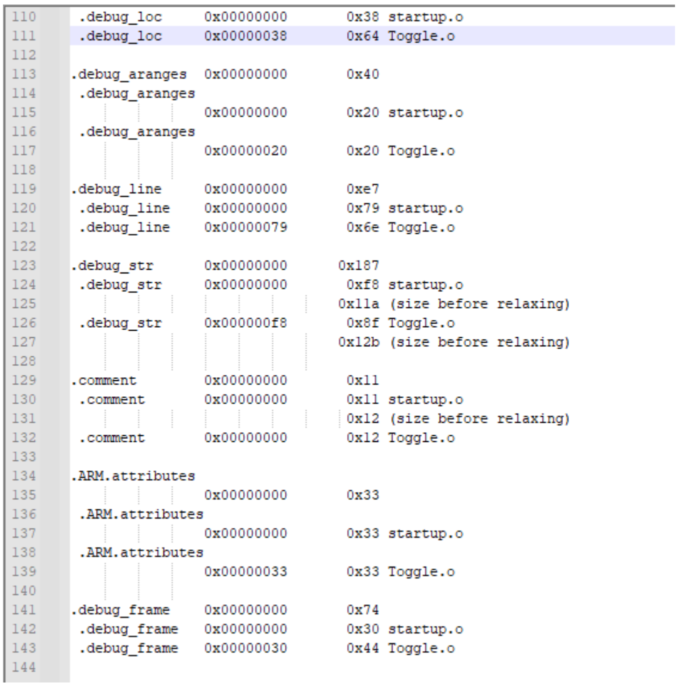
Map File:



```

Allocating common symbols
Common symbol       size              file

g_un_variables      0x1               Toggle.o

Memory Configuration

Name             Origin             Length             Attributes
flash            0x08000000         0x00020000         xr
sram             0x20000000         0x00005000         xrw
*default*        0x00000000         0xffffffff

Linker script and memory map


.text           0x08000000      0x1c0
 *(.vectors*)
 .vectors       0x08000000       0x58 startup.o
                0x08000000                vectors
 *(.text*)
 .text          0x08000058       0xa8 startup.o
                0x08000058                FLASH_Handler
                0x08000058                SVCall_Handler
                0x08000058                SysTick_Handler
                0x08000058                PendSV_Handler
                0x08000058                NMI_Handler
                0x08000058                WWDG_Handler
                0x08000058                RTC_Handler
                0x08000058                UsageFault_Handler
                0x08000058                EXTI1_Handler
                0x08000058                Reset_Handler
                0x08000058                EXTI2_Handler
                0x08000058                Reserved_Handler
                0x08000058                EXTI0_Handler
                0x08000058                RCC_Handler
                0x08000058                PVD_Handler
                0x08000058                MM_mange_Handler
                0x08000058                BusFault_Handler
                0x08000058                DebugMonitor_Handler
                0x08000058                TAMPER_Handler
 .text          0x08000100       0xbc Toggle.o
                0x08000100                main
                0x080001b0                HardFault_Handler
 *(.rodata)
 .rodata        0x080001bc        0x1 Toggle.o
                0x080001bc                g_c_i_variables
                0x080001c0                . = ALIGN (0x4)
 *fill*         0x080001bd        0x3
                0x080001c0                _e_text = .

.glue_7         0x080001c0        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x080001c0        0x0
 .glue_7t       0x00000000        0x0 linker stubs
```



```
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x080001c0        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x080001c0        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x080001c0        0x0
 .iplt          0x00000000        0x0 startup.o

.rel.dyn        0x080001c0        0x0
 .rel.iplt      0x00000000        0x0 startup.o

.data           0x20000000        0x8 load address 0x080001c0
                0x20000000                _s_data = .
 *(.data)
 .data          0x20000000        0x0 startup.o
 .data          0x20000000        0x8 Toggle.o
                0x20000000                R_ODR
                0x20000004                g_i_variables
                0x20000008                . = ALIGN (0x4)
                0x20000008                _e_data = .

.igot.plt       0x20000008        0x0 load address 0x080001c8
 .igot.plt      0x00000000        0x0 startup.o

.bss            0x20000008     0x1005 load address 0x080001c8
                0x20000008                _s_bss = .
 *(.bss*)
 .bss           0x20000008        0x0 startup.o
 .bss           0x20000008        0x1 Toggle.o
                0x20000008                g_i_W_0_variables
                0x2000000c                . = ALIGN (0x4)
 *fill*         0x20000009        0x3
                0x2000000c                _e_bss = .
                0x2000100c                . = (. + 0x1000)
 *fill*         0x2000000c     0x1000
                0x2000100c                _stack_top = .
 COMMON         0x2000100c        0x1 Toggle.o
                0x2000100c                g_un_variables
LOAD startup.o
LOAD Toggle.o
OUTPUT(Toggle_Led_cortex_M3.elf elf32-littlearm)

.debug_info     0x00000000      0x2cd
 .debug_info    0x00000000      0x151 startup.o
 .debug_info    0x00000151      0x17c Toggle.o

.debug_abbrev   0x00000000      0x18a
 .debug_abbrev  0x00000000       0xac startup.o
 .debug_abbrev  0x000000ac       0xde Toggle.o

.debug_loc      0x00000000       0x9c
 .debug_loc     0x00000000       0x38 startup.o
 .debug_loc     0x00000038       0x64 Toggle.o

```



```
 .debug_loc     0x00000000       0x38 startup.o
 .debug_loc     0x00000038       0x64 Toggle.o

.debug_aranges  0x00000000       0x40
 .debug_aranges
                0x00000000       0x20 startup.o
 .debug_aranges
                0x00000020       0x20 Toggle.o

.debug_line     0x00000000       0xe7
 .debug_line    0x00000000       0x79 startup.o
 .debug_line    0x00000079       0x6e Toggle.o

.debug_str      0x00000000      0x187
 .debug_str     0x00000000       0xf8 startup.o
                                0x11a (size before relaxing)
 .debug_str     0x000000f8       0x8f Toggle.o
                                0x12b (size before relaxing)

.comment        0x00000000       0x11
 .comment       0x00000000       0x11 startup.o
                                 0x12 (size before relaxing)
 .comment       0x00000000       0x12 Toggle.o

.ARM.attributes
                0x00000000       0x33
 .ARM.attributes
                0x00000000       0x33 startup.o
 .ARM.attributes
                0x00000033       0x33 Toggle.o

.debug_frame    0x00000000       0x74
 .debug_frame   0x00000000       0x30 startup.o
 .debug_frame   0x00000030       0x44 Toggle.o

```