

The symbols before linking are unresolved yet & take relocatable addresses (virtual addresses) After linking files.o into file.elf : all symbols are resolved and takes it's physical addresses

Symbols for lab_startup_c

```
$ arm-none-eabi-nm startup.o
U _E_BSS
U _E_DATA
U _E_text
U _S_BSS
U _S_DATA
U _stack_top
00000000 T Default_Handler
00000000 W LL_Fault_Handler
U main
00000000 W MM_Bus
00000000 W MM_Fault_Handler
00000000 W NMI_Handler
00000000 T Reset_Handler
00000000 W Usage_Fault_Handler
00000000 D vectors

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/GET-REF
embedded_c_course/lesson3/lab_startup.s (master)
$ arm-none-eabi-nm main.o
00000000 D access_a
00000000 T main

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/GET-REF
embedded_c_course/lesson3/lab_startup.s (master)
$ arm-none-eabi-nm LEARN_IN_DEPTH_COTREX-M3.elf
20000004 B _E_BSS
20000004 D _E_DATA
08000168 T _E_text
20000004 B _S_BSS
20000000 D _S_DATA
20001004 B _stack_top
20000000 D access_a
0800001c T Default_Handler
0800001c W LL_Fault_Handler
080000d8 T main
0800001c W MM_Bus
0800001c W MM_Fault_Handler
0800001c W NMI_Handler
08000028 T Reset_Handler
0800001c W Usage_Fault_Handler
08000000 T vectors
```

Handwritten annotations for the first screenshot:

- Arrows pointing to the unresolved symbols in the first section: **Un resolved**
- An arrow pointing to the resolved symbols in the second section: **resolved**

Symbols for lab_startup_c

```
$ arm-none-eabi-nm main.o
00000000 D access_a
00000000 T main

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/GET-REF
embedded_c_course/lesson3/lab_startup.s (master)
$ arm-none-eabi-nm startup.o
00000000 t _reset
00000000 U _stack_top
00000000 t _vector_handler
00000000 U main

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/GET-REF
embedded_c_course/lesson3/lab_startup.s (master)
$ arm-none-eabi-nm LEARN_IN_DEPTH_COTREX-M3.elf
20000000 B _E_BSS
080000ec D _E_DATA
080000e8 T _E_text
080000e0 t _reset
20000000 B _S_BSS
080000e8 D _S_DATA
20001000 B _stack_top
080000e6 t _vector_handler
080000e8 D access_a
08000050 T main
```

Handwritten annotations for the second screenshot:

- Arrows pointing to the unresolved symbols in the first section: **virtual addresses**
- A box around the unresolved symbols in the second section with the text: **unresolved before linking**
- Arrows pointing to the resolved symbols in the third section: **all symbols is resolved now**
- An arrow pointing to the resolved symbols in the third section: **Physical addresses**

Main_c sections: has virtual addresses

Main_s

No LMA or VMA addresses

```
main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000090 00000000 00000000 00000034 2**2
   CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000004 00000000 00000000 000000c4 2**2
   CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000 00000000 00000000 000000c8 2**0
   ALLOC
 3 .debug_info    00000144 00000000 00000000 000000c8 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev  000000db 00000000 00000000 0000020c 2**0
   CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     00000038 00000000 00000000 000002e7 2**0
   CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000 00000000 0000031f 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    0000005d 00000000 00000000 0000033f 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     00000118 00000000 00000000 0000039c 2**0
   CONTENTS, READONLY, DEBUGGING
 9 .comment       00000012 00000000 00000000 000004b4 2**0
   CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000004c6 2**0
   CONTENTS, READONLY
11 .debug_frame   0000002c 00000000 00000000 000004fc 2**2
   CONTENTS, RELOC, READONLY, DEBUGGING
```

```
main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000090 00000000 00000000 00000034 2**2
   CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000004 00000000 00000000 000000c4 2**2
   CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000 00000000 00000000 000000c8 2**0
   ALLOC
 3 .debug_info    00000144 00000000 00000000 000000c8 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev  000000db 00000000 00000000 0000020c 2**0
   CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     00000038 00000000 00000000 000002e7 2**0
   CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000 00000000 0000031f 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    0000005d 00000000 00000000 0000033f 2**0
   CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     00000116 00000000 00000000 0000039c 2**0
   CONTENTS, READONLY, DEBUGGING
 9 .comment       00000012 00000000 00000000 000004b2 2**0
   CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000004c4 2**0
   CONTENTS, READONLY
11 .debug_frame   0000002c 00000000 00000000 000004f8 2**2
   CONTENTS, RELOC, READONLY, DEBUGGING
```

see sections after linking in .elf: all section has physical LMA & VMA

c headers:

s headers:

```
LEARN_IN_DEPTH_COTREX-M3.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000168 00000000 00000000 00000000 2**2
   CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data          00000004 20000000 00000168 00010000 2**2
   CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00001000 20000004 0000016c 00010004 2**0
   ALLOC
 3 .debug_info    000002b6 00000000 00000000 00010004 2**0
   CONTENTS, READONLY, DEBUGGING
 4 .debug_abbrev  0000019b 00000000 00000000 000102ba 2**0
   CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     0000009c 00000000 00000000 00010455 2**0
   CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000040 00000000 00000000 000104f1 2**0
   CONTENTS, READONLY, DEBUGGING
 7 .debug_line    00000109 00000000 00000000 00010531 2**0
   CONTENTS, READONLY, DEBUGGING
 8 .debug_str     00000183 00000000 00000000 0001063a 2**0
   CONTENTS, READONLY, DEBUGGING
 9 .comment       00000011 00000000 00000000 000107bd 2**0
   CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000107ce 2**0
   CONTENTS, READONLY
11 .debug_frame   00000078 00000000 00000000 00010804 2**2
   CONTENTS, READONLY, DEBUGGING
```

```
LEARN_IN_DEPTH_COTREX-M3.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          000000e8 00000000 00000000 00000000 2**2
   CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data          00000004 00000004 000000e8 000000e8 2**2
   CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00001000 20000000 20000000 00010000 2**0
   ALLOC
 3 .debug_info    000001a0 00000000 00000000 000000ec 2**0
   CONTENTS, READONLY, DEBUGGING
 4 .debug_abbrev  000000ef 00000000 00000000 0000028c 2**0
   CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     00000038 00000000 00000000 0000037b 2**0
   CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000040 00000000 00000000 000003b8 2**3
   CONTENTS, READONLY, DEBUGGING
 7 .debug_line    00000098 00000000 00000000 000003f8 2**0
   CONTENTS, READONLY, DEBUGGING
 8 .debug_str     000000e2 00000000 00000000 00000490 2**0
   CONTENTS, READONLY, DEBUGGING
 9 .comment       00000011 00000000 00000000 00000572 2**0
   CONTENTS, READONLY
10 .ARM.attributes 00000031 00000000 00000000 00000583 2**0
   CONTENTS, READONLY
11 .debug_frame   0000002c 00000000 00000000 000005b4 2**2
   CONTENTS, READONLY, DEBUGGING
```

MEMORY MAP:

LAB_STARTUP_S:

.text	0x08000000	0xe8
(.vectors)		
.vectors	0x08000000	0x50 startup.o
(.text)		
.text	0x08000050	0x90 main.o
.text	0x08000050	main
*(.rodata)		
	0x080000e8	_E_text_ = .
.data	0x080000e8	0x4
*(.data)		
.data	0x080000e8	0x4 main.o
.data	0x080000e8	access_a
.data	0x080000ec	0x0 startup.o
	0x080000ec	. = ALIGN (0x4)
	0x080000ec	_E_DATA_ = .
.igot.plt	0x080000ec	0x0
.igot.plt	0x080000ec	0x0 main.o
.bss	0x20000000	0x1000
	0x20000000	_S_BSS_ = .
*(.bss)		
.bss	0x20000000	0x0 main.o
.bss	0x20000000	0x0 startup.o
	0x20000000	. = ALIGN (0x4)
	0x20000000	_E_BSS_ = .
	0x20001000	. = (. + 0x1000)
fill	0x20000000	0x1000
	0x20001000	_stack_top = .

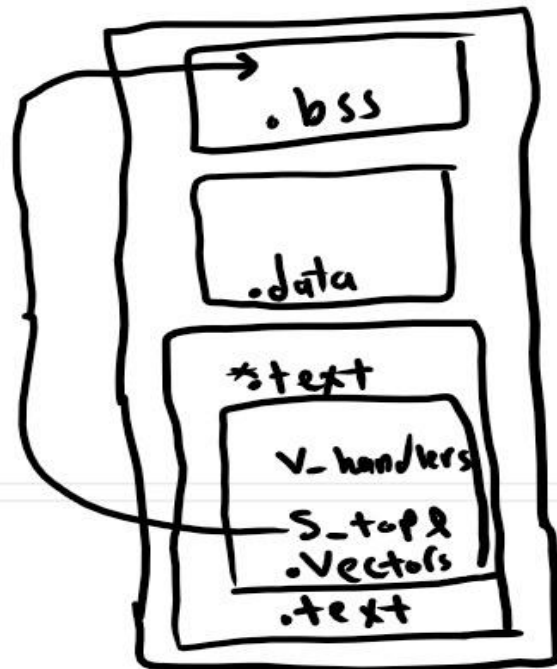
Sections Produced as I describe in linker script I create a .text section starts it with the vector handler table (.vectors) then: other .text in other Files.o (c-code in main) then I create (.data sec) but it's empty (no global or static) then: .bss section I located 0x1000 for BSS section and locate stack_top on the top of BSS section.

LAB_STARTUP_C:

Linker script and memory map

.text	0x08000000	0x168
(.vectors)		
.vectors	0x08000000	0x1c startup.o
(.text)		
.text	0x0800001c	0xbc startup.o
	0x0800001c	NMI_Handler
	0x0800001c	LL_Fault_Handler
	0x0800001c	MM_Fault_Handler
	0x0800001c	Usage_Fault_Handler
	0x0800001c	Default_Handler
	0x0800001c	MM_Bus
	0x08000028	Reset_Handler
.text	0x080000d8	0x90 main.o
	0x080000d8	main
*(.rodata)		
	0x08000168	_E_text_ = .
.data	0x20000000	0x4 load address 0x08000168
*(.data)		
.data	0x20000000	0x0 startup.o
.data	0x20000000	0x4 main.o
	0x20000000	access_a
	0x20000004	. = ALIGN (0x4)
	0x20000004	_E_DATA_ = .
.igot.plt	0x20000004	0x0 load address 0x0800016c
.igot.plt	0x08000000	0x0 startup.o
.bss	0x20000004	0x1000 load address 0x0800016c
	0x20000004	_S_BSS_ = .
*(.bss)		
.bss	0x20000004	0x0 startup.o
.bss	0x20000004	0x0 main.o
	0x20000004	. = ALIGN (0x4)
	0x20000004	_E_BSS_ = .
	0x20001004	. = (. + 0x1000)
fill	0x20000004	0x1000
	0x20001004	_stack_top = .

LOAD startup.o
LOAD main.o
OUTPUT(LEARN_IN_DEPTH_COTREX-M3.elf elf32-littlearm)



I copied .data from Rom to S RAM (LMA) → VMA