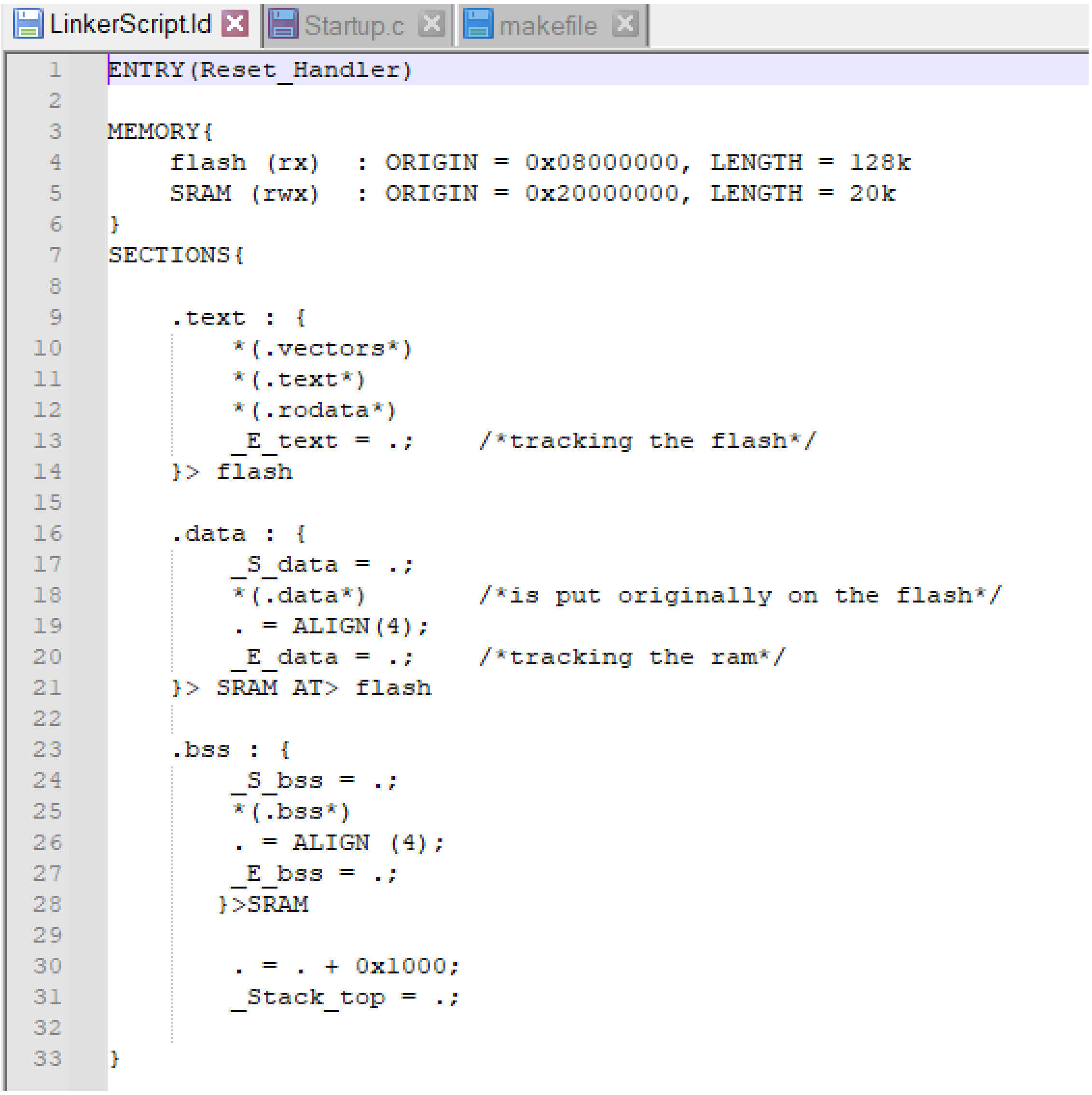
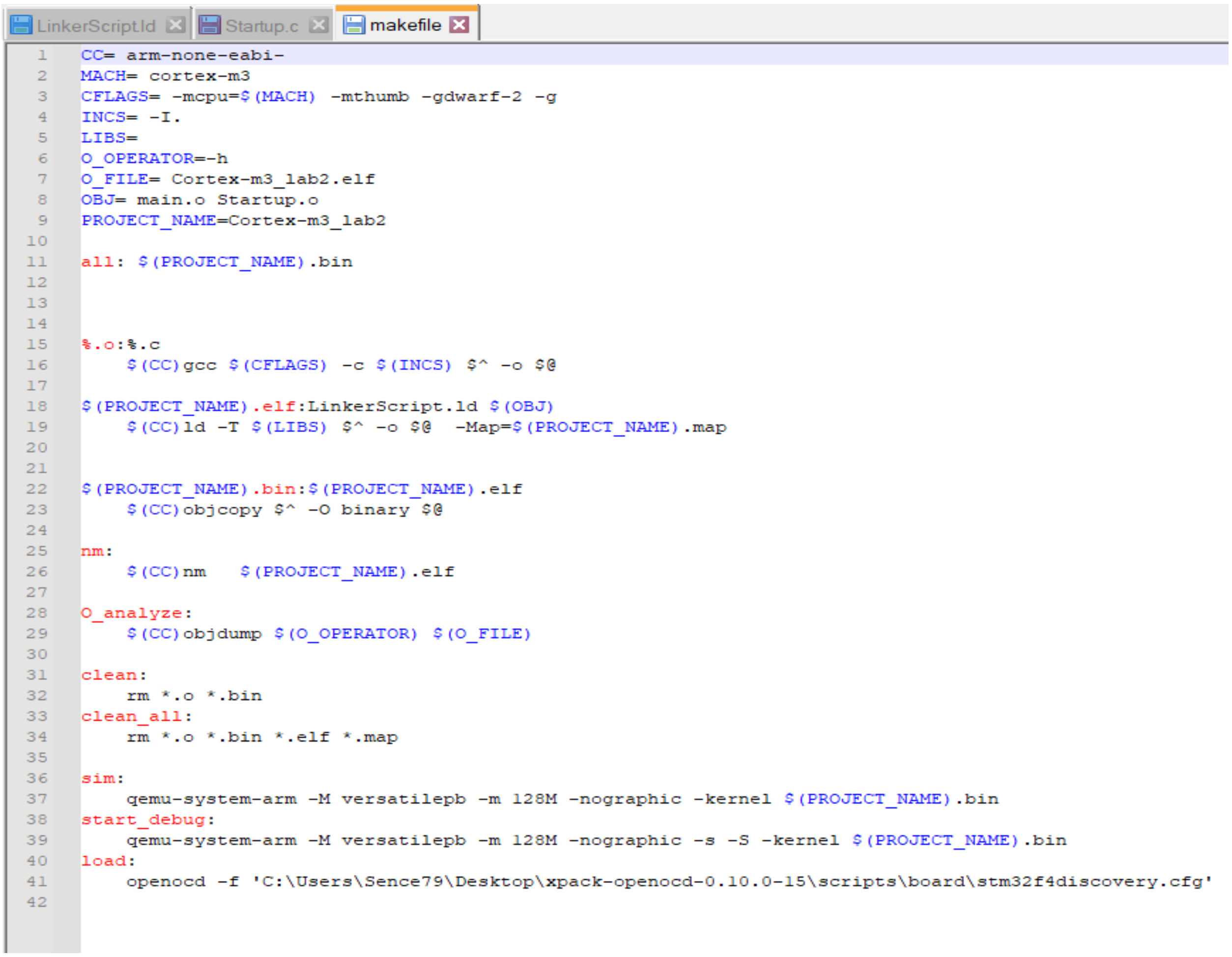
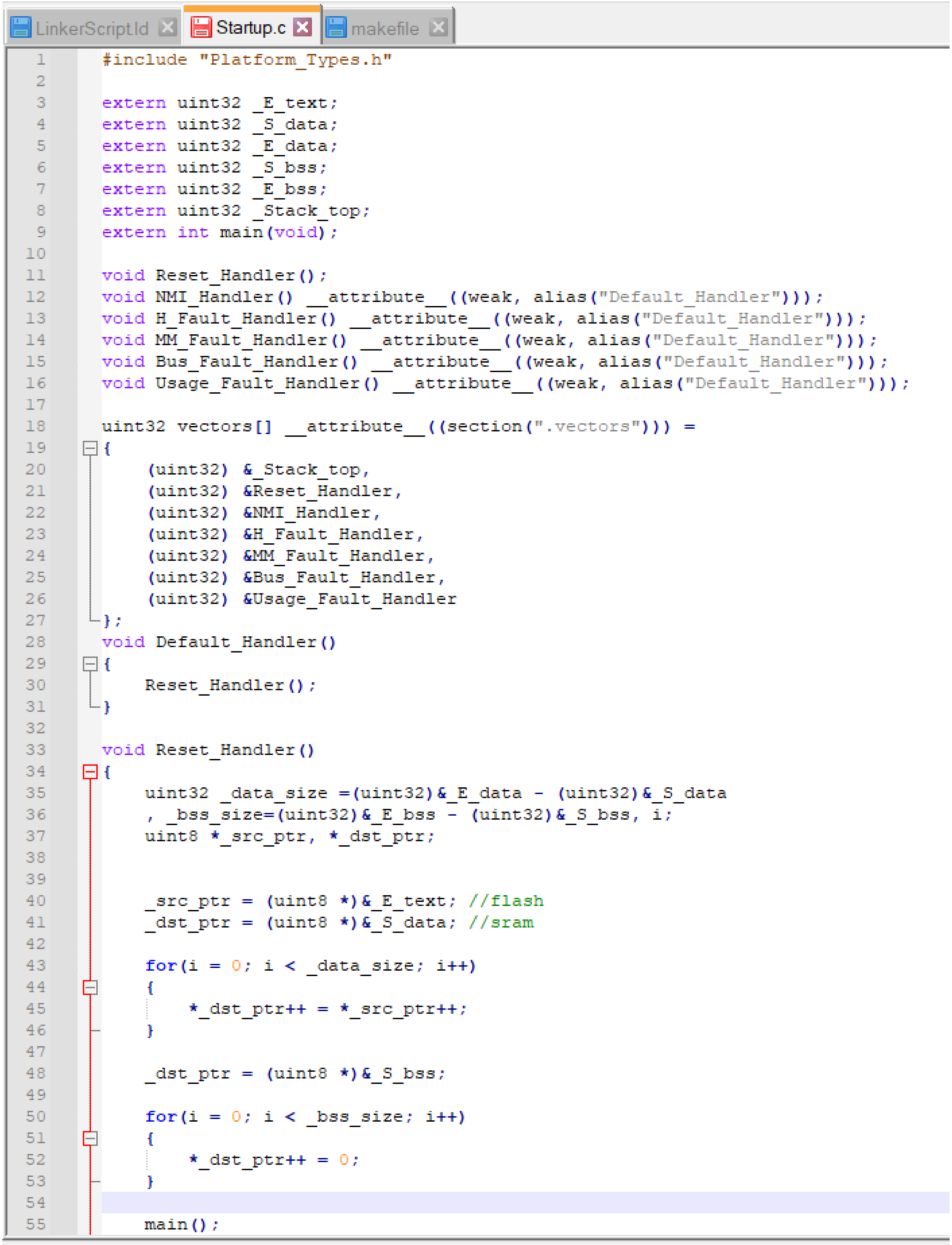
Files needed to build and compile the main.c file :



```
ENTRY(Reset_Handler)

MEMORY{
    flash (rx)  : ORIGIN = 0x08000000, LENGTH = 128k
    SRAM (rwx)  : ORIGIN = 0x20000000, LENGTH = 20k
}
SECTIONS{

    .text : {
        *(.vectors*)
        *(.text*)
        *(.rodata*)
        _E_text = .;    /*tracking the flash*/
    }> flash

    .data : {
        _S_data = .;
        *(.data*)       /*is put originally on the flash*/
        . = ALIGN(4);
        _E_data = .;    /*tracking the ram*/
    }> SRAM AT> flash

    .bss : {
        _S_bss = .;
        *(.bss*)
        . = ALIGN (4);
        _E_bss = .;
       }>SRAM

        . = . + 0x1000;
        _Stack_top = .;

}
```

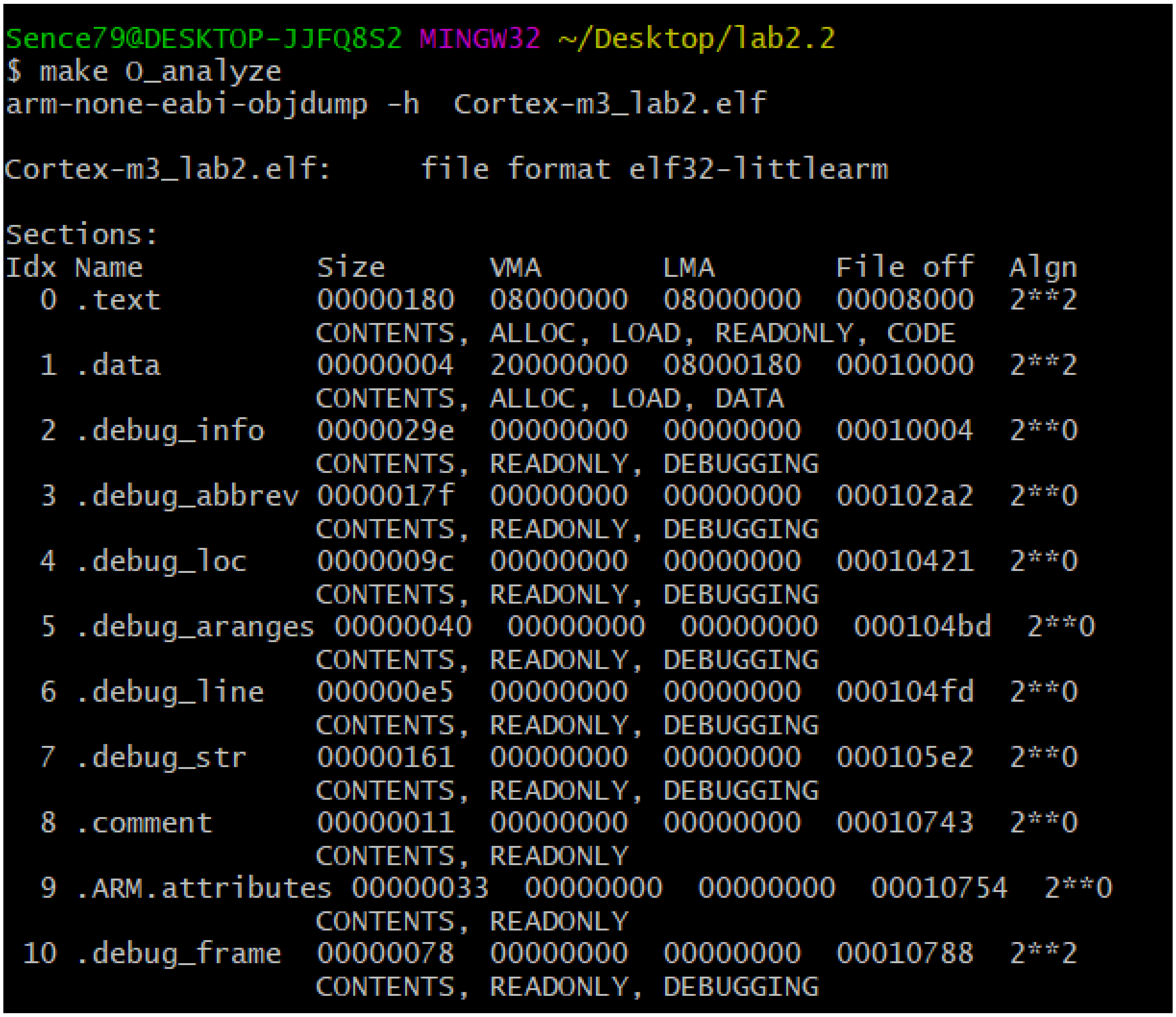


```
CC= arm-none-eabi-
MACH= cortex-m3
CFLAGS= -mcpu=$(MACH) -mthumb -gdwarf-2 -g
INCS= -I.
LIBS=
O_OPERATOR=-h
O_FILE= Cortex-m3_lab2.elf
OBJ= main.o Startup.o
PROJECT_NAME=Cortex-m3_lab2

all: $(PROJECT_NAME).bin



%.o:%.c
	$(CC)gcc $(CFLAGS) -c $(INCS) $^ -o $@

$(PROJECT_NAME).elf:LinkerScript.ld $(OBJ)
	$(CC)ld -T $(LIBS) $^ -o $@  -Map=$(PROJECT_NAME).map


$(PROJECT_NAME).bin:$(PROJECT_NAME).elf
	$(CC)objcopy $^ -O binary $@

nm:
	$(CC)nm   $(PROJECT_NAME).elf

O_analyze:
	$(CC)objdump $(O_OPERATOR) $(O_FILE)

clean:
	rm *.o *.bin
clean_all:
	rm *.o *.bin *.elf *.map

sim:
	qemu-system-arm -M versatilepb -m 128M -nographic -kernel $(PROJECT_NAME).bin
start_debug:
	qemu-system-arm -M versatilepb -m 128M -nographic -s -S -kernel $(PROJECT_NAME).bin
load:
	openocd -f 'C:\Users\Sence79\Desktop\xpack-openocd-0.10.0-15\scripts\board\stm32f4discovery.cfg'

```

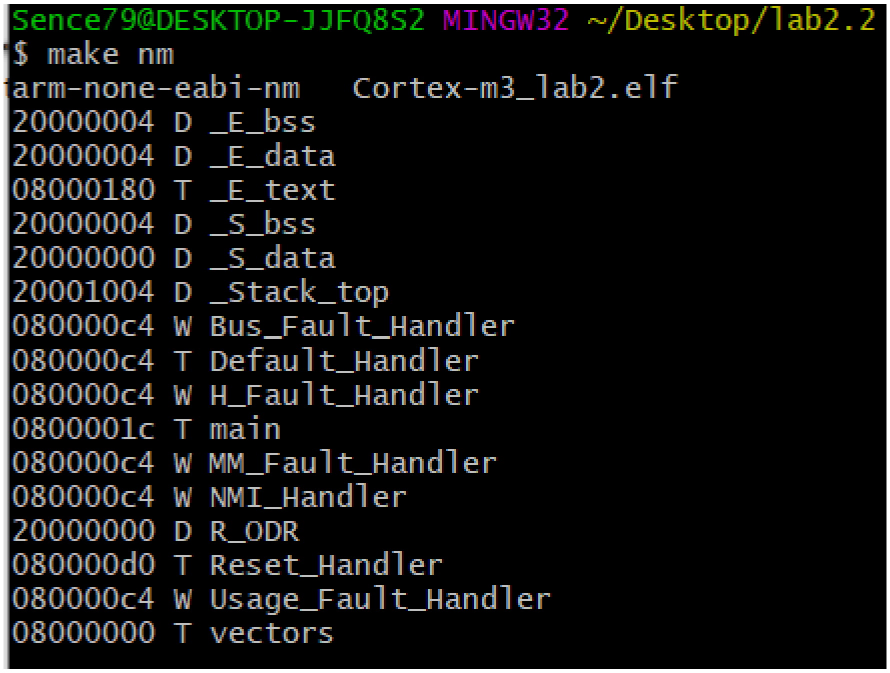


```c
#include "Platform_Types.h"

extern uint32 _E_text;
extern uint32 _S_data;
extern uint32 _E_data;
extern uint32 _S_bss;
extern uint32 _E_bss;
extern uint32 _Stack_top;
extern int main(void);

void Reset_Handler();
void NMI_Handler() __attribute__((weak, alias("Default_Handler")));
void H_Fault_Handler() __attribute__((weak, alias("Default_Handler")));
void MM_Fault_Handler() __attribute__((weak, alias("Default_Handler")));
void Bus_Fault_Handler() __attribute__((weak, alias("Default_Handler")));
void Usage_Fault_Handler() __attribute__((weak, alias("Default_Handler")));

uint32 vectors[] __attribute__((section(".vectors"))) =
{
    (uint32) &_Stack_top,
    (uint32) &Reset_Handler,
    (uint32) &NMI_Handler,
    (uint32) &H_Fault_Handler,
    (uint32) &MM_Fault_Handler,
    (uint32) &Bus_Fault_Handler,
    (uint32) &Usage_Fault_Handler
};
void Default_Handler()
{
    Reset_Handler();
}

void Reset_Handler()
{
    uint32 _data_size =(uint32)&_E_data - (uint32)&_S_data
    , _bss_size=(uint32)&_E_bss - (uint32)&_S_bss, i;
    uint8 *_src_ptr, *_dst_ptr;


    _src_ptr = (uint8 *)&_E_text; //flash
    _dst_ptr = (uint8 *)&_S_data; //sram

    for(i = 0; i < _data_size; i++)
    {
        *_dst_ptr++ = *_src_ptr++;
    }

    _dst_ptr = (uint8 *)&_S_bss;

    for(i = 0; i < _bss_size; i++)
    {
        *_dst_ptr++ = 0;
    }

    main();
```

Analyzing some .o files:

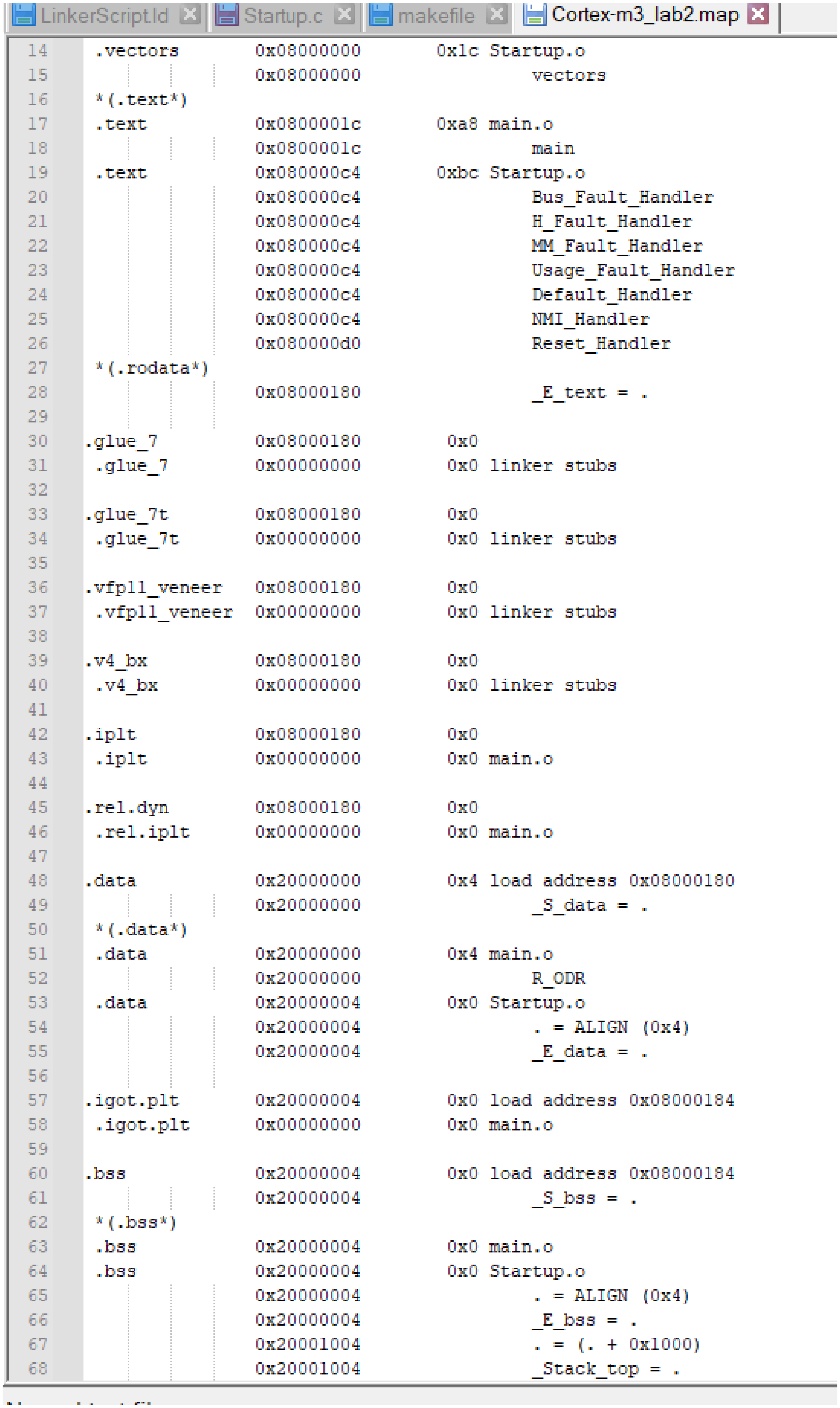


```
Sence79@DESKTOP-JJFQ8S2 MINGW32 ~/Desktop/lab2.2
$ make O_analyze
arm-none-eabi-objdump -h  Cortex-m3_lab2.elf

Cortex-m3_lab2.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000180  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000004  20000000  08000180  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .debug_info   0000029e  00000000  00000000  00010004  2**0
                  CONTENTS, READONLY, DEBUGGING
  3 .debug_abbrev 0000017f  00000000  00000000  000102a2  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_loc    0000009c  00000000  00000000  00010421  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000040  00000000  00000000  000104bd  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_line   000000e5  00000000  00000000  000104fd  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_str    00000161  00000000  00000000  000105e2  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      00000011  00000000  00000000  00010743  2**0
                  CONTENTS, READONLY
  9 .ARM.attributes 00000033  00000000  00000000  00010754  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000078  00000000  00000000  00010788  2**2
                  CONTENTS, READONLY, DEBUGGING
```



```
Sence79@DESKTOP-JJFQ8S2 MINGW32 ~/Desktop/lab2.2
$ make nm
arm-none-eabi-nm   Cortex-m3_lab2.elf
20000004 D _E_bss
20000004 D _E_data
08000180 T _E_text
20000004 D _S_bss
20000000 D _S_data
20001004 D _Stack_top
080000c4 W Bus_Fault_Handler
080000c4 T Default_Handler
080000c4 W H_Fault_Handler
0800001c T main
080000c4 W MM_Fault_Handler
080000c4 W NMI_Handler
20000000 D R_ODR
080000d0 T Reset_Handler
080000c4 W Usage_Fault_Handler
08000000 T vectors
```

.map file:



```
 .vectors         0x08000000       0x1c Startup.o
                  0x08000000                vectors
 *(.text*)
 .text            0x0800001c       0xa8 main.o
                  0x0800001c                main
 .text            0x080000c4       0xbc Startup.o
                  0x080000c4                Bus_Fault_Handler
                  0x080000c4                H_Fault_Handler
                  0x080000c4                MM_Fault_Handler
                  0x080000c4                Usage_Fault_Handler
                  0x080000c4                Default_Handler
                  0x080000c4                NMI_Handler
                  0x080000d0                Reset_Handler
 *(.rodata*)
                  0x08000180                _E_text = .

.glue_7           0x08000180        0x0
 .glue_7          0x00000000        0x0 linker stubs

.glue_7t          0x08000180        0x0
 .glue_7t         0x00000000        0x0 linker stubs

.vfp11_veneer     0x08000180        0x0
 .vfp11_veneer    0x00000000        0x0 linker stubs

.v4_bx            0x08000180        0x0
 .v4_bx           0x00000000        0x0 linker stubs

.iplt             0x08000180        0x0
 .iplt            0x00000000        0x0 main.o

.rel.dyn          0x08000180        0x0
 .rel.iplt        0x00000000        0x0 main.o

.data             0x20000000        0x4 load address 0x08000180
                  0x20000000                _S_data = .
 *(.data*)
 .data            0x20000000        0x4 main.o
                  0x20000000                R_ODR
 .data            0x20000004        0x0 Startup.o
                  0x20000004                . = ALIGN (0x4)
                  0x20000004                _E_data = .

.igot.plt         0x20000004        0x0 load address 0x08000184
 .igot.plt        0x00000000        0x0 main.o

.bss              0x20000004        0x0 load address 0x08000184
                  0x20000004                _S_bss = .
 *(.bss*)
 .bss             0x20000004        0x0 main.o
 .bss             0x20000004        0x0 Startup.o
                  0x20000004                . = ALIGN (0x4)
                  0x20000004                _E_bss = .
                  0x20001004                . = (. + 0x1000)
                  0x20001004                _Stack_top = .
```