

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

```
MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/units/unit3/lesson2/assignment
$ arm-none-eabi-nm app.o
00000000 T main
00000000 D $B
          U UART_SEND_STRING_

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/units/unit3/lesson2/assignment
$ arm-none-eabi-nm startup.o
          U main
00000000 T reset
          U Stack_Top
00000008 t stop

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/units/unit3/lesson2/assignment
$ arm-none-eabi-nm uart.o
00000000 T UART_SEND_STRING_

MOSTAFA@DESKTOP-SKJU3NL MINGW32 /d/Diploma/units/unit3/lesson2/assignment
$ arm-none-eabi-nm UART_LAB.elf
00010010 T main
00010000 T reset
00010080 D $B
000110e4 D Stack_Top
00010008 t stop
00010030 T UART_SEND_STRING_
```

MEMPRY MAP:

Memory configuration: memory starts at 0x0000 0000 & ends at 0x 0400 0000

Takes attributes to enable it to read, write and execute

#### Memory Configuration

Name	Origin	Length	Attributes
FLASH	0x00000000	0x04000000	xrw

Sections:

1- program counter locator will starts to locate sections at 0x0001 0000

Linker script and memory map

```
. = 0x00010000 . = 0x10000
```

```
SECTIONS
{
    . = 0x10000;
```

2- startup section: program starts at it's entry point at the at startup section

I exit the .text data from startup.o to startup section

The first instruction in startup is to load the stack top into sp register

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

```
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
reset () at startup.s:3
3          ldr sp, = Stack_Top
(gdb) _
```

The first step instruction after the sp initialization is to branch to main.

```
(gdb) si
reset () at startup.s:4
4          bl main
(gdb) _
```

3- .text section: the c code after compiler exit in .text section

```
.text      0x00010010      0x70
*(.text)
.text      0x00010010      0x20 app.o
            0x00010010      main
.text      0x00010030      0x50 uart.o
            0x00010030      UART_SEND_STRING_
```

4- .bss section: it contains the uninitialized global data: my code hasn't uninitialized data

```

.bss                0x000100e4        0x0
*(.bss)
.bss                0x000100e4        0x0 startup.o
.bss                0x000100e4        0x0 app.o
.bss                0x000100e4        0x0 uart.o
                   0x000110e4
                   0x000110e4        . = (. + 0x1000)
                                   Stack_Top = .

LOAD app.o
LOAD uart.o
LOAD startup.o
OUTPUT(UART_LAB.elf elf32-littlearm)

```

## DEBUGGER TOUTORIAL

The image shows two side-by-side windows from a debugger environment. The left window displays a list of breakpoints for the UART\_SEND\_STRING function, with addresses ranging from 0x10093 to 0x1009c. The right window shows a terminal output with a command prompt and a series of commands, including 'qemu-system-arm -M versatilepb -n 128M -nographic -s -S -kernel UART\_LAB.elf', followed by a series of 'learn in depth: mostafa emary' messages.

```

MINGW32/d/Diploma/GET-REPO/embedded_system_online_diplome/embedded_c_course/les...
Breakpoint 2, UART_SEND_STRING_ (p=0x10093 <S_B+19> "tafa emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10094 <S_B+20> "afa emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10095 <S_B+21> "fa emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10096 <S_B+22> "a emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10097 <S_B+23> " emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10098 <S_B+24> "emary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x10099 <S_B+25> "mary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x1009a <S_B+26> "ary") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x1009b <S_B+27> "ry") at uart.c:7
7
(gdb) c
Continuing.
Breakpoint 2, UART_SEND_STRING_ (p=0x1009c <S_B+28> "y") at uart.c:7
7
(gdb) c
Continuing.
[Inferior 1 (Remote target) exited normally]
(gdb) c
The program is not being run.
(gdb) quit

MOSTAFAPADESKTOP-SKHU3NL MINGW32 /d/Diploma/GET-REPO/embedded_system_online_diplome/
embedded_c_course/lesson2 <master>
$ qemu-system-arm -M versatilepb -n 128M -nographic -s -S -kernel UART_LAB.elf
learn in depth: mostafa emarylearn in depth: mostafa emary
MOSTAFAPADESKTOP-SKHU3NL MINGW32 /d/Diploma/GET-REPO/embedded_system_online_diplome/
embedded_c_course/lesson2 <master>
$ qemu-system-arm -M versatilepb -n 128M -nographic -s -S -kernel UART_LAB.elf
learn in depth: mostafa emary
MOSTAFAPADESKTOP-SKHU3NL MINGW32 /d/Diploma/GET-REPO/embedded_system_online_diplome/
embedded_c_course/lesson2 <master>
$

```