# Lab 1

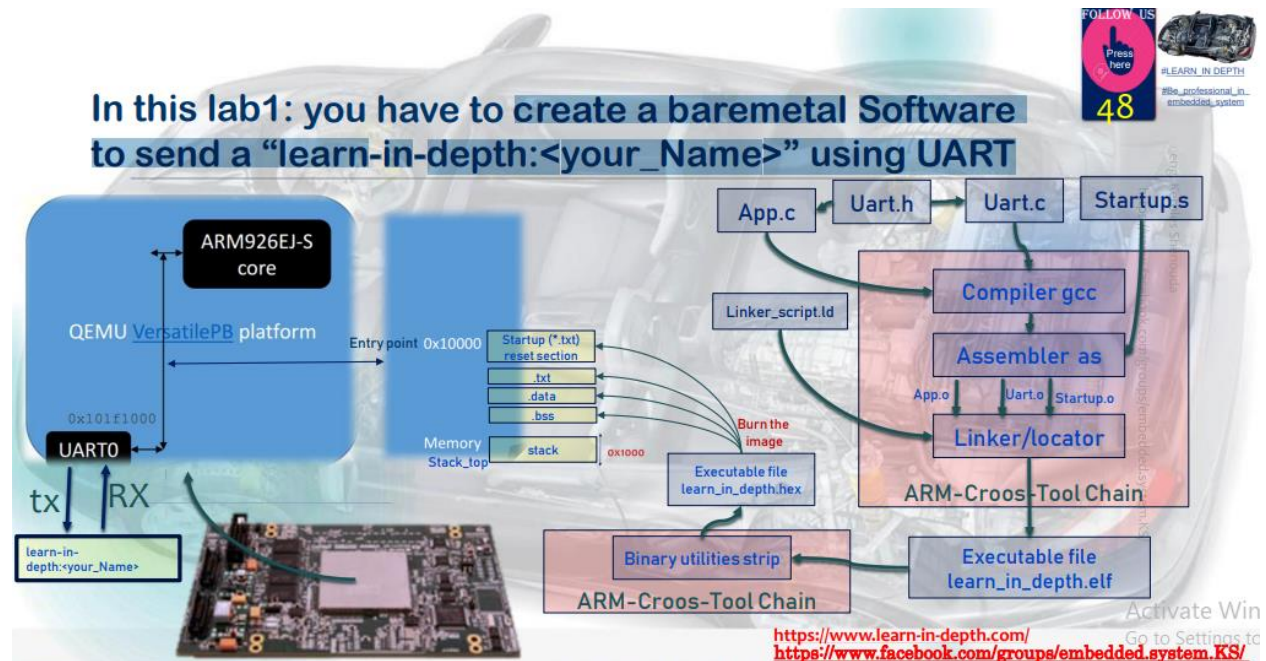## Description:

Create a bare-metal Software to send a "learn-in-depth: <<your name>>" using UART of the ARM Versatile PB board.



## Files Created:

- uart.h
- uart.c
- app.c
- startup.s
- linker_script.ld

## Executable Files:

- learn-in-depth.elf
- learn-in-depth.bin

## Analysis Files:

- uart.o
- app.o
- startup.o
- Map_file.map

## Git Commands Used In Compilation Process:

### To get the object files:

- $ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s uart.c -o uart.o
- $ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s app.c -o app.o
- $ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s startup.c -o startup.o

### To link the object files together using linker_script and get the .elf and .map files:

- $ arm-none-eabi-ld.exe -T linker_script.ld app.o uart.o startup.o -o learn-in-depth.elf -Map=Map_file.map

### To get the bin file:

- $ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin

### To run the program in the QEMU Simulator ("VersatilePB physical Board"):

- $ qemu-system-arm.exe -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin

# Git Commands Used In Analysis Process:

## To display the content of the section headers:

- $ arm-none-eabi-objdump.exe -h uart.o

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kero
los Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ 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         00000054  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000088  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000088  2**0
                  ALLOC
  3 .debug_info   00000057  00000000  00000000  00000088  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000051  00000000  00000000  000000df  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000020  00000000  00000000  00000130  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_line   00000039  00000000  00000000  00000150  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_str    000000fe  00000000  00000000  00000189  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  00000287  2**0
                  CONTENTS, READONLY
  9 .debug_frame  00000030  00000000  00000000  00000308  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  00000338  2**0
                  CONTENTS, READONLY

El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kero
los Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$
```

- $ arm-none-eabi-objdump.exe -h app.o

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ 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         0000001c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  00000050  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b4  2**0
                  ALLOC
  3 .debug_info   00000077  00000000  00000000  000000b4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 0000005a  00000000  00000000  0000012b  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000020  00000000  00000000  00000185  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_line   00000035  00000000  00000000  000001a5  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_str    0000010f  00000000  00000000  000001da  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  000002e9  2**0
                  CONTENTS, READONLY
  9 .debug_frame  0000002c  00000000  00000000  00000368  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  00000394  2**0
                  CONTENTS, READONLY
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$
```

- $ `arm-none-eabi-objdump.exe -h startup.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ 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 00000012  00000000  00000000  00000044  2**0
                  CONTENTS, READONLY

El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$
```

- $ `arm-none-eabi-objdump.exe -h learn-in-depth.elf`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

learn-in-depth.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      00000010  00010000  00010000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000070  00010010  00010010  00010010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  00010080  00010080  00010080  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .bss          00000064  000100e4  000100e4  000100e4  2**2
                  ALLOC
  4 .ARM.attributes 0000002e  00000000  00000000  000100e4  2**0
                  CONTENTS, READONLY
  5 .comment      0000007e  00000000  00000000  00010112  2**0
                  CONTENTS, READONLY
  6 .debug_info   000000ce  00000000  00000000  00010190  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_abbrev 000000ab  00000000  00000000  0001025e  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000040  00000000  00000000  00010309  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_line   0000006e  00000000  00000000  00010349  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    00000133  00000000  00000000  000103b7  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_frame  0000005c  00000000  00000000  000104ec  2**2
                  CONTENTS, READONLY, DEBUGGING

El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
```

## To display the assembler content of all the sections:

- $ `arm-none-eabi-objdump.exe -D uart.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-objdump.exe -D uart.o

uart.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <uart_send_string>:
   0:   e52db004    push    {fp}            ; (str fp, [sp, #-4]!)
   4:   e28db000    add     fp, sp, #0
   8:   e24dd00c    sub     sp, sp, #12
   c:   e50b0008    str     r0, [fp, #-8]
  10:   ea000006    b       30 <uart_send_string+0x30>
  14:   e51b3008    ldr     r3, [fp, #-8]
  18:   e5d32000    ldrb    r2, [r3]
  1c:   e59f302c    ldr     r3, [pc, #44]   ; 50 <uart_send_string+0x50>
  20:   e5832000    str     r2, [r3]
  24:   e51b3008    ldr     r3, [fp, #-8]
  28:   e2833001    add     r3, r3, #1
  2c:   e50b3008    str     r3, [fp, #-8]
  30:   e51b3008    ldr     r3, [fp, #-8]
  34:   e5d33000    ldrb    r3, [r3]
  38:   e3530000    cmp     r3, #0
  3c:   1afffff4    bne     14 <uart_send_string+0x14>
  40:   e1a00000    nop                     ; (mov r0, r0)
  44:   e28bd000    add     sp, fp, #0
  48:   e49db004    pop     {fp}            ; (ldr fp, [sp], #4)
  4c:   e12fff1e    bx      lr
  50:   101f1000    andsne  r1, pc, r0

Disassembly of section .debug_info:

00000000 <.debug_info>:
   0:   00000053    andeq   r0, r0, r3, asr r0
   4:   00000004    andeq   r0, r0, r4
   8:   01040000    mrseq   r0, (UNDEF: 4)
   c:   00000089    andeq   r0, r0, r9, lsl #1
```

- $ arm-none-eabi-objdump.exe -D app.o

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ 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:   e59f0008        ldr     r0, [pc, #8]    ; 18 <main+0x18>
   c:   ebfffffe        bl      0 <uart_send_string>
  10:   e1a00000        nop                     ; (mov r0, r0)
  14:   e8bd8800        pop     {fp, pc}
  18:   00000000        andeq   r0, r0, r0

Disassembly of section .data:

00000000 <string_buffer>:
   0:   7261656c        rsbvc   r6, r1, #108, 10        ; 0x1b000000
   4:   6e692d6e        cdpvs   13, 6, cr2, cr9, cr14, {3}
   8:   7065642d        rsbvc   r6, r5, sp, lsr #8
   c:   203a6874        eorscs  r6, sl, r4, ror r8
  10:   62453c3c        subvs   r3, r5, #60, 24 ; 0x3c00
  14:   206d6172        rsbcs   r6, sp, r2, ror r1
  18:   69626148        stmdbvs r2!, {r3, r6, r8, sp, lr}^
  1c:   003e3e62        eorseq  r3, lr, r2, ror #28
        ...

Disassembly of section .debug_info:

00000000 <.debug_info>:
   0:   00000073        andeq   r0, r0, r3, ror r0
   4:   00000004        andeq   r0, r0, r4
   8:   01040000        mrseq   r0, (UNDEF: 4)
   c:   000000a7        andeq   r0, r0, r7, lsr #1
  10:   0001040c        andeq   r0, r1, ip, lsl #8
  14:   00002b00        andeq   r2, r0, r0, lsl #22
  18:   00000000        andeq   r0, r0, r0
  1c:   00001c00        andeq   r1, r0, r0, lsl #24
  20:   00000000        andeq   r0, r0, r0
  24:   003c0200        eorseq  r0, ip, r0, lsl #4
  28:   00350000        eorseq  r0, r5, r0
  2c:   35030000        strcc   r0, [r3, #-0]
  30:   63000000        movwvs  r0, #0
  34:   07040400        streq   r0, [r4, -r0, lsl #8]
  38:   0000009a        muleq   r0, sl, r0
```

- $ arm-none-eabi-objdump.exe -D startup.o

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-objdump.exe -D startup.o

startup.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <reset>:
   0:   e59fd004        ldr     sp, [pc, #4]    ; c <stop+0x4>
   4:   ebfffffe        bl      0 <main>

00000008 <stop>:
   8:   eafffffe        b       8 <stop>
   c:   00000000        andeq   r0, r0, r0

Disassembly of section .ARM.attributes:

00000000 <.ARM.attributes>:
   0:   00001141        andeq   r1, r0, r1, asr #2
   4:   61656100        cmnvs   r5, r0, lsl #2
   8:   01006962        tsteq   r0, r2, ror #18
   c:   00000007        andeq   r0, r0, r7
  10:   Address 0x00000010 is out of bounds.

El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$
```

## To display the full content of all sections requested:

- $ arm-none-eabi-objdump.exe -s uart.o

```
$ arm-none-eabi-objdump.exe -s uart.o

uart.o:     file format elf32-littlearm

Contents of section .text:
 0000 04b02de5 00b08de2 0cd04de2 08000be5  ..-.......M.....
 0010 060000ea 08301be5 0020d3e5 2c309fe5  .....0.. ..,0..
 0020 002083e5 08301be5 013083e2 08300be5  . ...0...0...0..
 0030 08301be5 0030d3e5 000053e3 f4ffff1a  .0...0....S.....
 0040 0000a0e1 00d08be2 04b09de4 1eff2fe1  ............../.
 0050 00101f10                             ....
Contents of section .debug_info:
 0000 53000000 04000000 00000401 89000000  S...............
 0010 0ce60000 001a0000 00000000 00540000  .............T..
 0020 00000000 0002ed00 00000105 00000000  ................
 0030 54000000 019c4900 00000300 00000001  T.....I.........
 0040 05490000 00029174 0004044f 00000005  .I.....t...O....
 0050 01080c00 000000                      .......
Contents of section .debug_abbrev:
 0000 01110125 0e130b03 0e1b0e11 01120610  ...%............
 0010 17000002 2e013f19 030e3a0b 3b0b2719  ......?...:.;.'.
 0020 11011206 40189742 19011300 00030500  ....@..B........
 0030 030e3a0b 3b0b4913 02180000 040f000b  ..:.;.I.........
 0040 0b491300 00052400 0b0b3e0b 030e0000  .I....$...>.....
 0050 00                                   .
Contents of section .debug_aranges:
 0000 1c000000 02000000 00000400 00000000  ................
 0010 00000000 54000000 00000000 00000000  ....T...........
Contents of section .debug_line:
 0000 35000000 02001d00 00000201 fb0e0d00  5...............
 0010 01010101 00000001 00000100 75617274  ............uart
 0020 2e630000 00000000 05020000 00001783  .c..............
 0030 30836387 020a0001 01                 0.c.....
Contents of section .debug_str:
 0000 505f7478 5f737472 696e6700 756e7369  P_tx_string.unsi
 0010 676e6564 20636861 7200453a 5c446f77  gned char.E:\Dow
 0020 6e6c6f61 64735c45 6d626564 64656420  nloads\Embedded
 0030 48657265 20576520 476f2041 6761696e  Here We Go Again
 0040 5c4b6572 6f6c6f73 20536865 6e6f6461  \Kerolos Shenoda
 0050 27732044 69706c6f 6d615c43 6f64655c  's Diploma\Code\
 0060 4d617374 6572696e 675f456d 62656464  Mastering_Embedd
 0070 65645f53 79737465 6d735c55 6e697433  ed_Systems\Unit3
 0080 5c6c6573 736f6e32 00474e55 20433131  \lesson2.GNU C11
 0090 20372e32 2e312032 30313730 39303420   7.2.1 20170904
 00a0 2872656c 65617365 29205b41 524d2f65  (release) [ARM/e
 00b0 6d626564 6465642d 372d6272 616e6368  mbedded-7-branch
 00c0 20726576 6973696f 6e203235 35323034   revision 255204
 00d0 5d202d6d 6370753d 61726d39 3236656a  ] -mcpu=arm926ej
```

- $ arm-none-eabi-objdump.exe -s app.o

```
$ arm-none-eabi-objdump.exe -s app.o

app.o:     file format elf32-littlearm

Contents of section .text:
 0000 00482de9 04b08de2 08009fe5 feffffeb  .H-.............
 0010 0000a0e1 0088bde8 00000000           ...........
Contents of section .data:
 0000 6c656172 6e2d696e 2d646570 74683a20  learn-in-depth:
 0010 3c3c4562 72616d20 48616269 623e3e00  <<Ebram Habib>>.
 0020 00000000 00000000 00000000 00000000  ................
 0030 00000000 00000000 00000000 00000000  ................
 0040 00000000 00000000 00000000 00000000  ................
 0050 00000000 00000000 00000000 00000000  ................
 0060 00000000                             ....
Contents of section .debug_info:
 0000 73000000 04000000 00000401 a7000000  s...............
 0010 0c040100 002b0000 00000000 001c0000  .....+..........
 0020 00000000 00023c00 00003500 00000335  ......<...5....5
 0030 00000063 00040407 9a000000 0401080f  ...c............
 0040 00000005 1d000000 01032500 00000503  ..........%.....
 0050 00000000 05000000 00010425 00000005  ...........%....
 0060 03000000 00060a01 00000105 00000000  ................
 0070 1c000000 019c00                      .......
Contents of section .debug_abbrev:
 0000 01110125 0e130b03 0e1b0e11 01120610  ...%............
 0010 17000002 01014913 01130000 03210049  ......I......!.I
 0020 132f0b00 00042400 0b0b3e0b 030e0000  ./...$...>.....
 0030 05340003 0e3a0b3b 0b49133f 19021800  .4...:.;.I.?....
 0040 00062e00 3f19030e 3a0b3b0b 27191101  ....?...:.;.'...
 0050 12064018 96421900 0000                ..@..B....
Contents of section .debug_aranges:
 0000 1c000000 02000000 00000400 00000000  ................
 0010 00000000 1c000000 00000000 00000000  ................
Contents of section .debug_line:
 0000 31000000 02001c00 00000201 fb0e0d00  1...............
 0010 01010101 00000001 00000100 6170702e  ............app.
 0020 63000000 00000005 02000000 00174b4b  c.............KK
 0030 02060001 01                          .....
Contents of section .debug_str:
 0000 73747269 6e675f62 75666665 72320075  string_buffer2.u
 0010 6e736967 6e656420 63686172 00737472  nsigned char.str
 0020 696e675f 62756666 65720045 3a5c446f  ing_buffer.E:\Do
 0030 776e6c6f 6164735c 456d6265 64646564  wnloads\Embedded
```

- $ `arm-none-eabi-objdump.exe -s startup.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-objdump.exe -s startup.o

startup.o:     file format elf32-littlearm

Contents of section .text:
 0000 04d09fe5 feffffeb feffffea 00000000  ...............
Contents of section .ARM.attributes:
 0000 41110000 00616561 62690001 07000000  A....aeabi......
 0010 0801                                 ..

El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$
```

## To read the symbols and check the Entry Point Address:

- $ `arm-none-eabi-nm.exe uart.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-nm.exe uart.o
00000000 T uart_send_string
```

- $ `arm-none-eabi-nm.exe app.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
00000064 C string_buffer2
         U uart_send_string
```

- $ `arm-none-eabi-nm.exe startup.o`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```
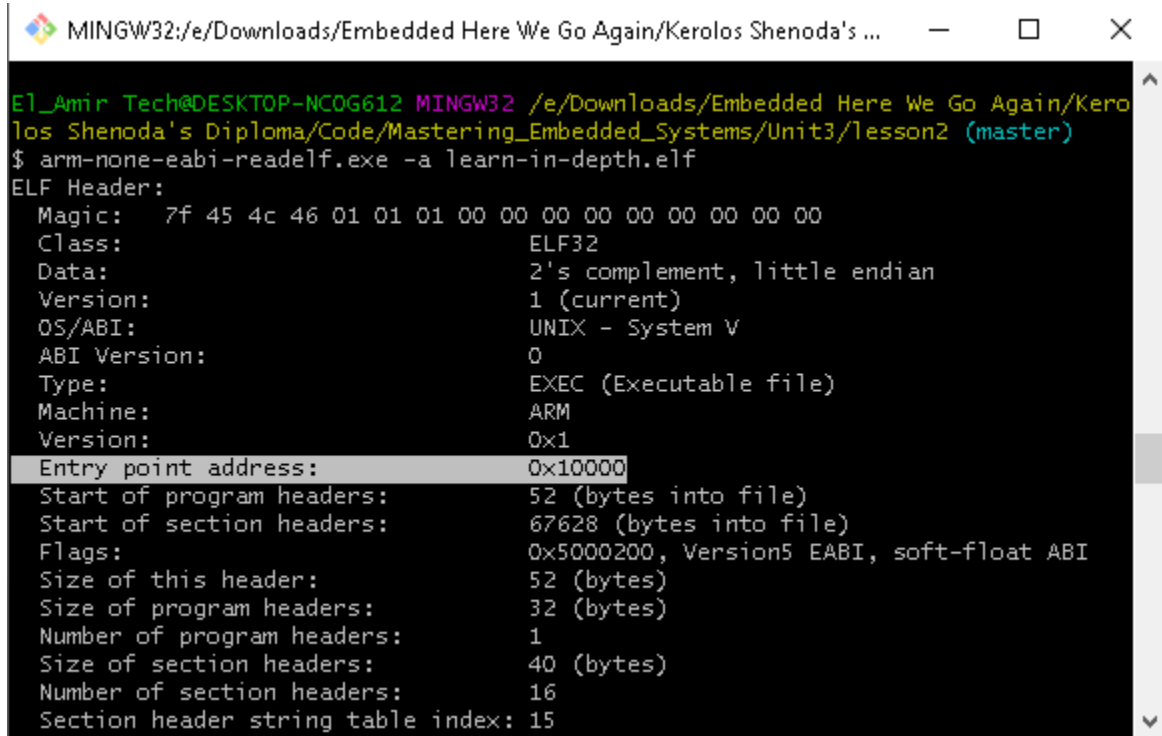
- $ `arm-none-eabi-nm.exe learn-in-depth.elf`

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
00011148 B stack_top
00010008 t stop
00010080 D string_buffer
000100e4 B string_buffer2
0001002c T uart_send_string
```

- `$ arm-none-eabi-readelf.exe -a learn-in-depth.elf`

MINGW32:/e/Downloads/Embedded Here We Go Again/Kerolos Shenoda's ... — □ ✕

```
El_Amir Tech@DESKTOP-NCOG612 MINGW32 /e/Downloads/Embedded Here We Go Again/Kero
los Shenoda's Diploma/Code/Mastering_Embedded_Systems/Unit3/lesson2 (master)
$ arm-none-eabi-readelf.exe -a learn-in-depth.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:          67628 (bytes into file)
  Flags:                             0x5000200, Version5 EABI, soft-float ABI
  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:         16
  Section header string table index: 15
```