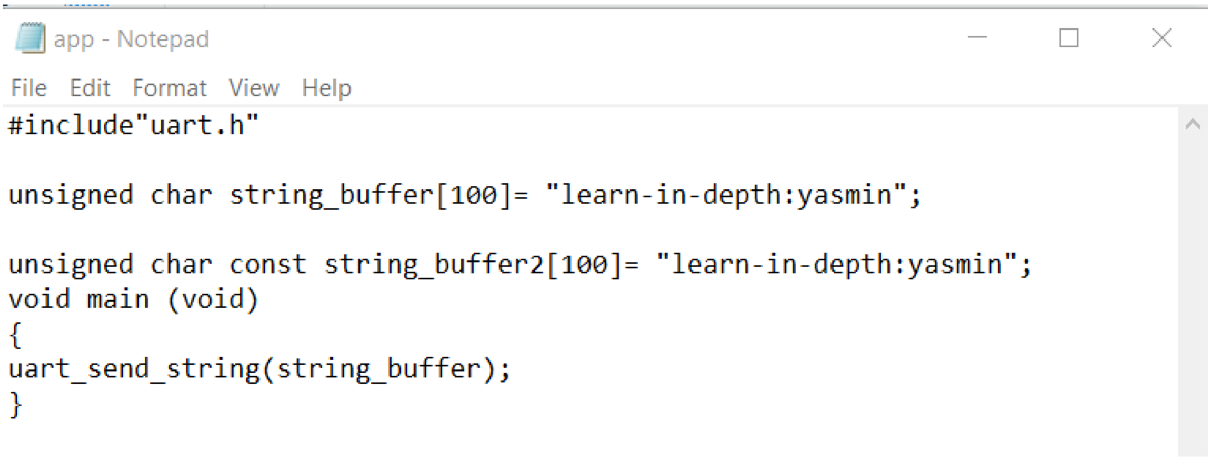
Embedded C Lesson 2

Lab 1

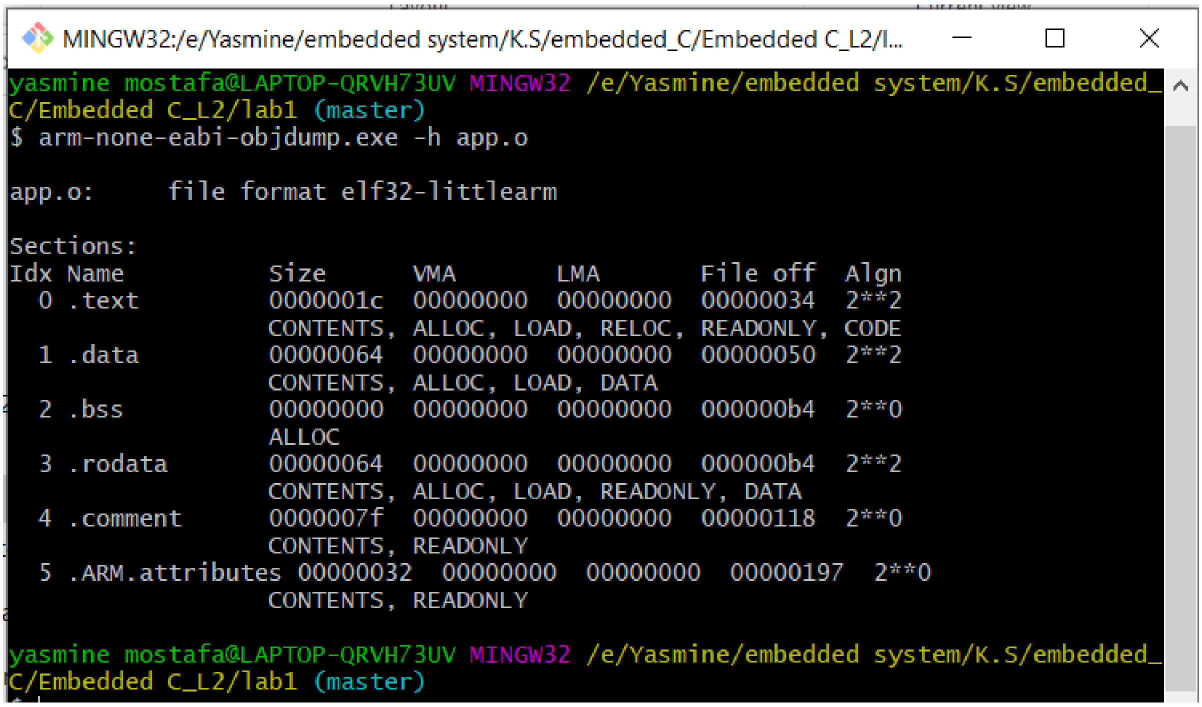
Yasmin Mostafa Abdelazem

App.c



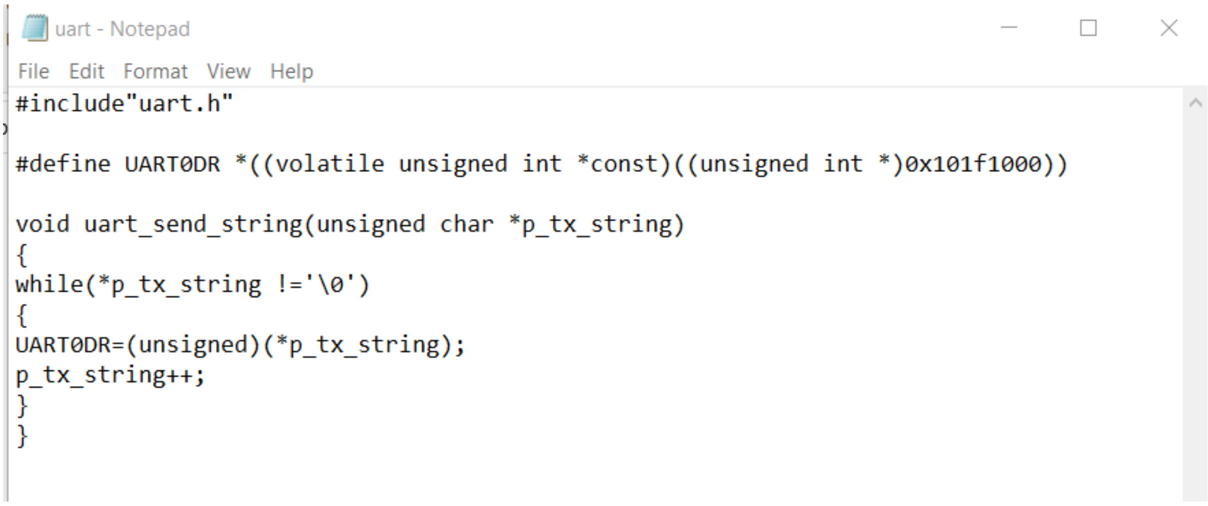
```
#include"uart.h"

unsigned char string_buffer[100]= "learn-in-depth:yasmin";

unsigned char const string_buffer2[100]= "learn-in-depth:yasmin";
void main (void)
{
uart_send_string(string_buffer);
}
```

App.o objdump .exe


MINGW32:/e/Yasmine/embedded system/K.S/embedded_C/Embedded C_L2/l...

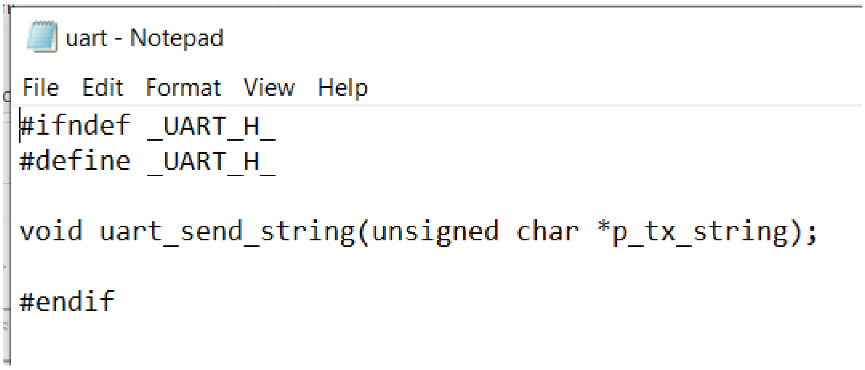
```
yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedded_
C/Embedded C_L2/lab1 (master)
$ arm-none-eabi-objdump.exe -h app.o

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 .rodata       00000064  00000000  00000000  000000b4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      0000007f  00000000  00000000  00000118  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000197  2**0
                  CONTENTS, READONLY

yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedded_
C/Embedded C_L2/lab1 (master)
```

uart.c



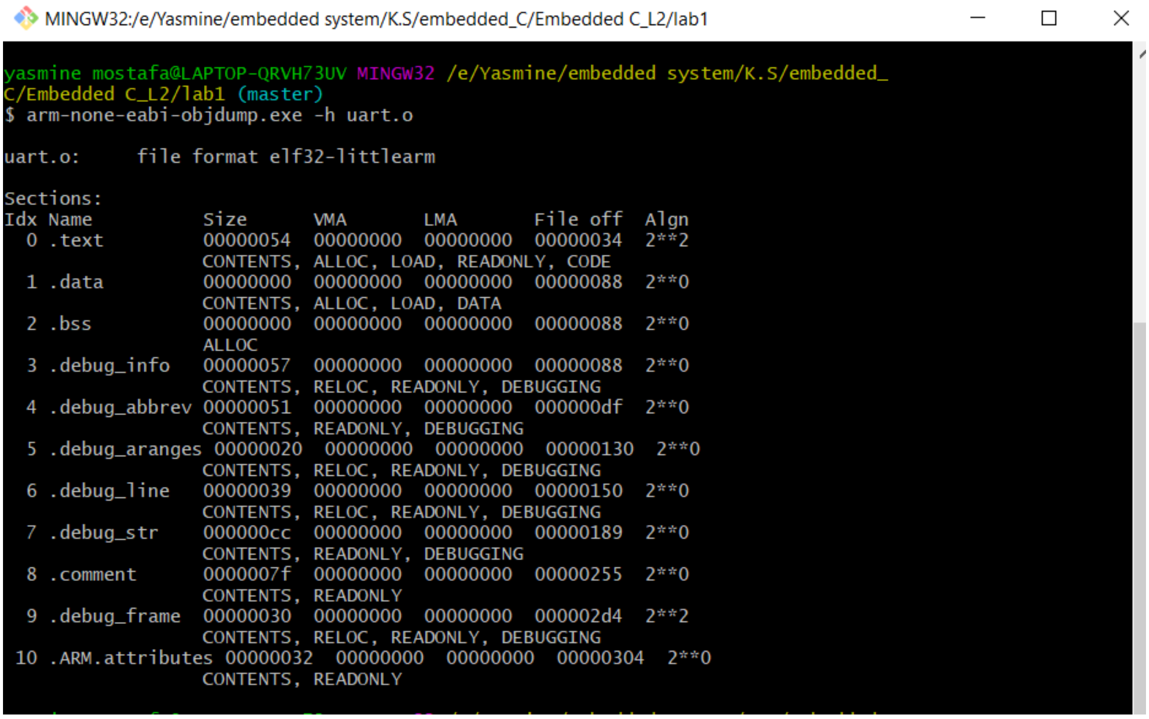
```c
#include"uart.h"

#define UART0DR *((volatile unsigned int *const)((unsigned int *)0x101f1000))

void uart_send_string(unsigned char *p_tx_string)
{
while(*p_tx_string !='\0')
{
UART0DR=(unsigned)(*p_tx_string);
p_tx_string++;
}
}
```

Uart.h



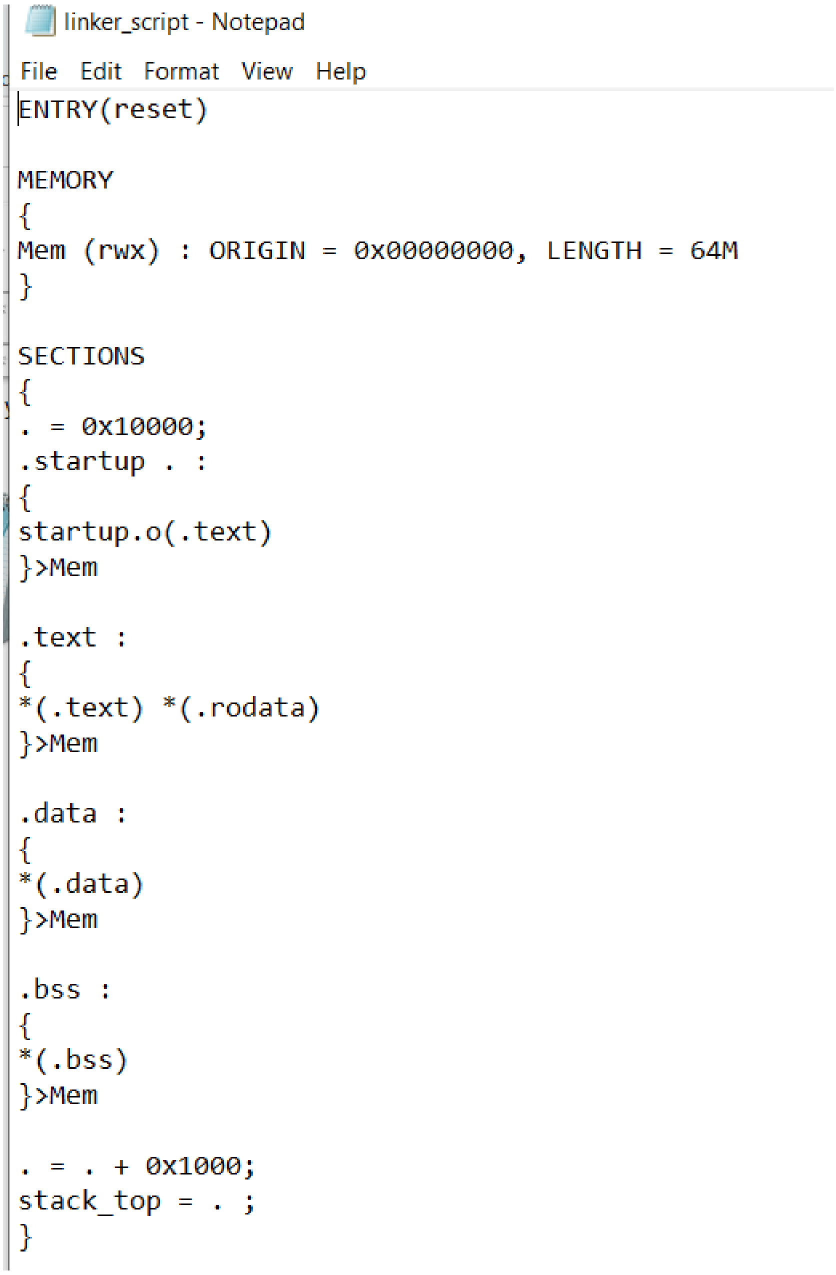
```c
#ifndef _UART_H_
#define _UART_H_

void uart_send_string(unsigned char *p_tx_string);

#endif
```

Uart.o objdump .exe


MINGW32:/e/Yasmine/embedded system/K.S/embedded_C/Embedded C_L2/lab1

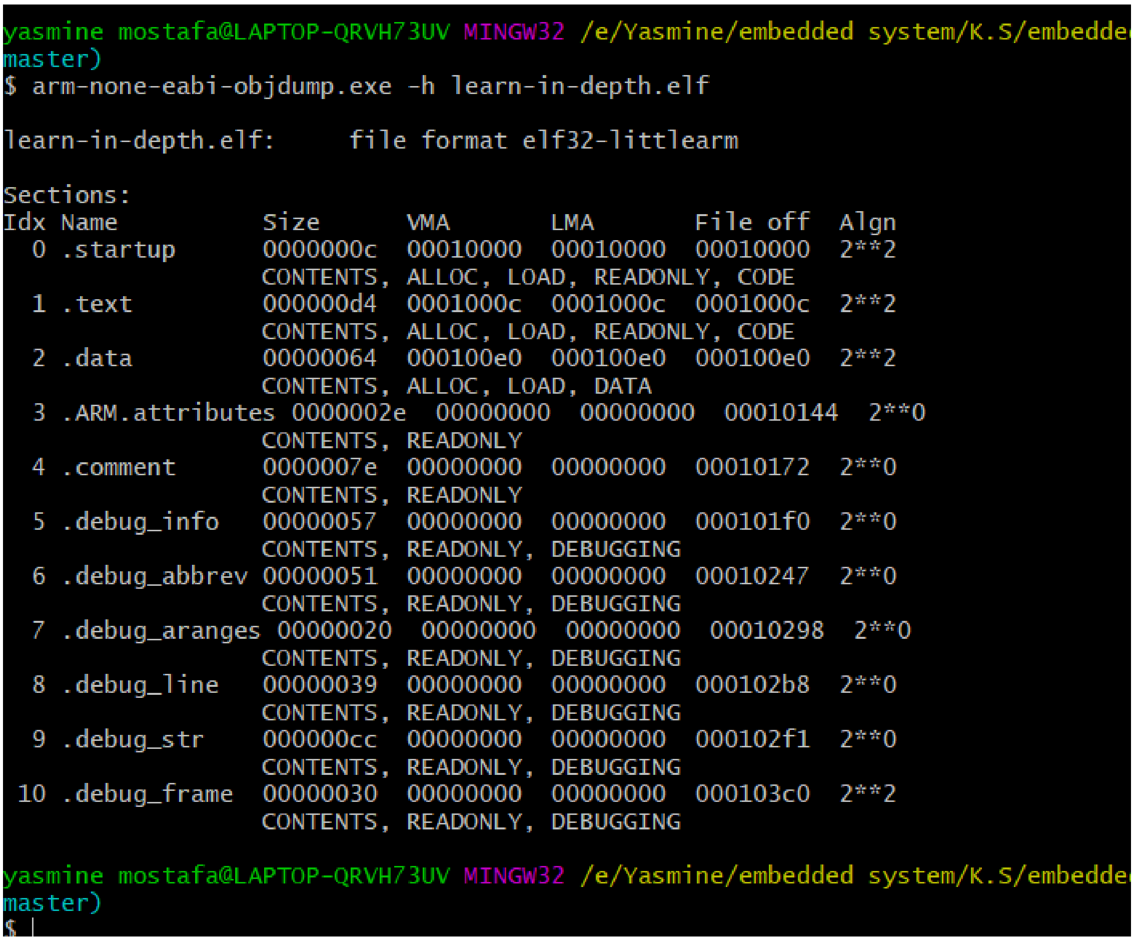
```
yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedded_
C/Embedded C_L2/lab1 (master)
$ arm-none-eabi-objdump.exe -h uart.o

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    000000cc  00000000  00000000  00000189  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007f  00000000  00000000  00000255  2**0
                  CONTENTS, READONLY
  9 .debug_frame  00000030  00000000  00000000  000002d4  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 10 .ARM.attributes 00000032  00000000  00000000  00000304  2**0
                  CONTENTS, READONLY
```

Linker_script.ld



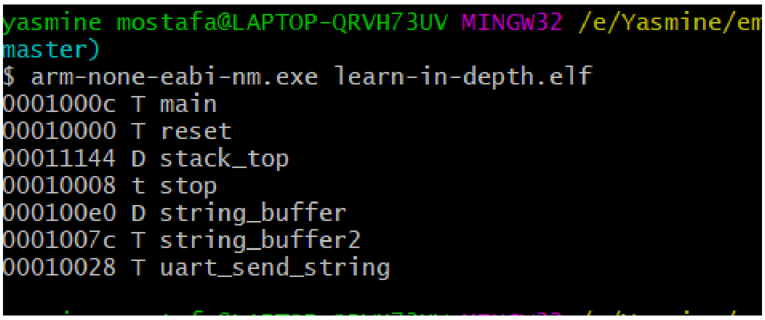
```
ENTRY(reset)

MEMORY
{
Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
}

SECTIONS
{
. = 0x10000;
.startup . :
{
startup.o(.text)
}>Mem

.text :
{
*(.text) *(.rodata)
}>Mem

.data :
{
*(.data)
}>Mem

.bss :
{
*(.bss)
}>Mem

. = . + 0x1000;
stack_top = . ;
}
```

Learn-in-depth objdump



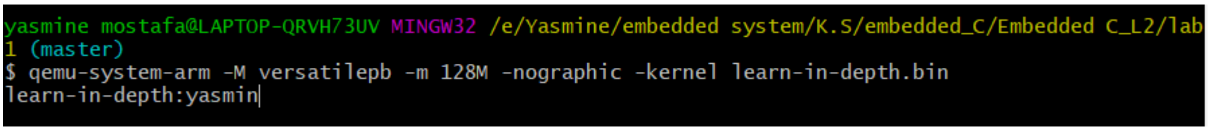
```
yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedde
master)
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf

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

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      0000000c  00010000  00010000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         000000d4  0001000c  0001000c  0001000c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  000100e0  000100e0  000100e0  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  00010144  2**0
                  CONTENTS, READONLY
  4 .comment      0000007e  00000000  00000000  00010172  2**0
                  CONTENTS, READONLY
  5 .debug_info   00000057  00000000  00000000  000101f0  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev 00000051  00000000  00000000  00010247  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00010298  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_line   00000039  00000000  00000000  000102b8  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_str    000000cc  00000000  00000000  000102f1  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_frame  00000030  00000000  00000000  000103c0  2**2
                  CONTENTS, READONLY, DEBUGGING

yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedde
master)
$ |
```

Learn-in-depth objdump



```
yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/em
master)
$ arm-none-eabi-nm.exe learn-in-depth.elf
0001000c T main
00010000 T reset
00011144 D stack_top
00010008 t stop
000100e0 D string_buffer
0001007c T string_buffer2
00010028 T uart_send_string
```

Output in qemu



```
yasmine mostafa@LAPTOP-QRVH73UV MINGW32 /e/Yasmine/embedded system/K.S/embedded_C/Embedded C_L2/lab
1 (master)
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn-in-depth:yasmin
```