Lab1

-write code to send string to UART0 and display it from scratch without IDE using arm-none-eabi tool chain

-we will write Startup code, Linker Scribt and C code

Board name : Verastilepb

MCPU name : arm926ej-s

According to specs :

Entry point : 0x10000

To write string on UART0 write only on UART0DR register

UART0DR address : 0x101f1000

C code Divided into (app.c – uart.c – uart.h)

App.c

```c
#include "Uart.h"

unsigned char myname[] = "Learn-In-Depth : Abdelrhman";

unsigned char const myName[] = "Learn-In-Depth : Abdelrhman";

void main(void)
{
    Uart_Send_String(myname);
}
```

Uart.c

```c
#include "Uart.h"

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


void Uart_Send_String(unsigned char *myString)
{
    while(*myString != '\0')
    {
        UART0DR = (unsigned int)(*myString);
        myString++;
    }
}
```

Uart.h

```c
#ifndef _UART_H_
#define _UART_H_

void Uart_Send_String(unsigned char *myString);

#endif
```

Startup.s

```
.global reset
reset:
    ldr sp, = Stack_Point
    bl main
stop: b stop
```

Linker_Scribt.ld

```
ENTRY (reset)

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

SECTIONS
{

    . = 0x10000;
    .startup . :
    {
        startup.o(.text)
    }>MEM
    .text :
    {
        *(.text)
    }>MEM
    .data :
    {
        *(.data)
    }>MEM
    .bss :
    {
        *(.bss)
    }>MEM
    .rodata :
    {
        *(.rodata)
    }>MEM
    . = . + 0x1000;
    Stack_Point = .;
}
```

next step :

is to generate obj file from app.c,Uart.c and startup.s by using this command line.

```
2M@DESKTOP-PD1LFOS MINGW64 /d/Abdo2/Embedded System/Kerolos/Unit3/Lesson2-Unit3/lap1
$ arm-none-eabi-gcc.exe -c app.c Uart.c

2M@DESKTOP-PD1LFOS MINGW64 /d/Abdo2/Embedded System/Kerolos/Unit3/Lesson2-Unit3/lap1
$ arm-none-eabi-gcc.exe -c Startup.s
```

next -> we need to link obj file using linker scribt and generate .bin file to burn it.

```
2M@DESKTOP-PD1LFOS MINGW64 /d/Abdo2/Embedded System/Kerolos/Unit3/Lesson2-Unit3/lap1
$ arm-none-eabi-ld.exe -T linker_scribt.ld Startup.o app.o Uart.o -o learn-in-depth.elf

2M@DESKTOP-PD1LFOS MINGW64 /d/Abdo2/Embedded System/Kerolos/Unit3/Lesson2-Unit3/lap1
$ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin
```

Here Using some binary utilities :

objdump –h to show all sections in file.

```
$ 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         00000074  00010010  00010010  00010010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         0000001c  00010084  00010084  00010084  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .rodata       0000001c  000100a0  000100a0  000100a0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .ARM.attributes 00000028  00000000  00000000  000100bc  2**0
                  CONTENTS, READONLY
  5 .comment      00000049  00000000  00000000  000100e4  2**0
                  CONTENTS, READONLY
  6 .debug_line   000000c1  00000000  00000000  0001012d  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  7 .debug_info   00000113  00000000  00000000  000101ee  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  8 .debug_abbrev 000000ce  00000000  00000000  00010301  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  9 .debug_aranges 00000060  00000000  00000000  000103d0  2**3
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 10 .debug_str    00000104  00000000  00000000  00010430  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 11 .debug_frame  0000005c  00000000  00000000  00010534  2**2
                  CONTENTS, READONLY, DEBUGGING, OCTETS
```

nm.exe to show all sympol

```
2M@DESKTOP-PD1LFOS MINGW64 /d/
lap1
$ arm-none-eabi-nm.exe learn-i
00010068 T main
00010084 D myname
000100a0 R myName
00010000 T reset
000110bc R Stack_Point
00010008 t stop
00010010 T Uart_Send_String
```

Simulation on Qemu

```
2M@DESKTOP-PD1LFOS MINGW64 /d/Abdo2/Embedded System/Kerolos/Unit3/Lesson2-Unit3/
lap1
$ C:/Program\ Files\ \(x86\)/qemu/qemu-system-arm -M versatilepb -m 128M -nograp
hic -kernel learn-in-depth.bin
Learn-In-Depth : Abdelrhman
```