LAB 1

Using VersatilePB virtual board in QEMU and ARM toolchain

1. Writing source files, getting object files and analyzing them:

```
MINGW64:/d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1 — 
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1
$ touch app.c uart.c uart.h

Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1
$ arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s app.c -o app.o

Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1
$ arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o

Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1
$ arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o
```

```
MINGW64 /d/tull_diploma/ES_Online_Diploma_KS/tirst_term/u
 3/1abs/1ab1
 arm-none-eabi-objdump.exe -h app.o
          file format elf32-littlearm
app.o:
Sections:
Idx Name
                 Size
                            VMA
                                      LMA
                                                File off
                                                          Algn
 0 .text
                 0000001c
                           00000000
                                     00000000
                                                00000034
                           ALLOC, LOAD, RELOC, READONLY, CODE
                 CONTENTS,
 1 .data
                 00000064
                           00000000 00000000
                                               00000050
                           ALLOC, LOAD, DATA
                 CONTENTS,
                           00000000 00000000
 2 .bss
                 00000000
                                               000000b4
                 ALLOC
                 00000064
                           00000000 00000000 000000b4
 3 .rodata
                 CONTENTS, ALLOC, LOAD, READONLY, DATA
                 00000045 00000000 00000000 00000118
 4 .comment
                                                          2**0
                 CONTENTS, READONLY
 5 .ARM.attributes 0000002c 00000000 00000000 0000015d 2**0
                 CONTENTS, READONLY
.pex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/un
  /labs/lab1
```

2. Writing startup code, getting object file and analyzing it:

```
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/u
t3/labs/lab1
  arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o
startup.s: Assembler messages:
startup.s: Warning: end of file not at end of a line; newline inserted
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/u
t3/labs/lab1
$ arm-none-eabi-objdump.exe -h startup.o
              file format elf32-littlearm
startup.o:
Sections:
Idx Name
                  Size
                                                File off
                            VMA
                                      LMA
                                                          Alan
 0 .text
                  0000000c
                            00000000 00000000
                                                00000034
                                                           2**7
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
                                                          2**0
 1 .data
                  00000000 00000000
                                      00000000
                                                00000040
                  CONTENTS, ALLOC, LOAD, DATA
 2 .bss
                  00000000 00000000
                                     00000000
                                                00000040
                                                          2**0
                  ALLOC
  3 .ARM.attributes 00000022 00000000
                                        00000000
                                                  00000040
                                                            2**0
                  CONTENTS, READONLY
```

3. Writing the linker script, linking all objects, getting the elf file , analyzing it and get sections of it :

```
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term,
t3/labs/lab1
$ arm-none-eabi-objdump.exe -h learn-in-depth.elf
learn-in-depth.elf: file format elf32-littlearm
Sections:
Idx Name
                  Size
                           VMA
                                               File off
                                     LMA
                                                         Algn
 0 .startup
                 00000010 00010000 00010000
                                               00001000
                 CONTENTS, ALLOC, LOAD, READONLY, CODE
                  000000d8 00010010 00010010 00001010
 1 .text
                                                         2**2
                 CONTENTS, ALLOC, LOAD, READONLY, CODE
                 00000064 000100e8 000100e8 000010e8
 2 .data
                                                         2**2
                 CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e 00000000 00000000 0000114c 2**0
                 CONTENTS, READONLY
  4 .comment
                 00000044 00000000 00000000
                                               0000117a 2**0
                 CONTENTS, READONLY
 pex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/
 3/labs/lab1
```

```
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/uni
t3/labs/lab1
$ arm-none-eabi-ld.exe -T linker_script.ld startup.o app.o uart.o -o learn-in-de
pth.elf -Map=Map_file.map
C:\Program Files (x86)\Arm\bin\arm-none-eabi-ld.exe: warning: learn-in-depth.elf
has a LOAD segment with RWX permissions
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/uni
t3/labs/lab1
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
0001114c D stack_top
00010008 t stop
000100e8 D string_buffer
00010084 T string_buffer_2
0001002c T uart_send_strig
```

4. Getting the binary file and simulating the application using QEMU:

```
Apex@DESKTOP-006R3/L MINGW64 /d/tull_diploma/ES_Online_Diploma_KS/first_term/uni
t3/labs/lab1
$ arm-none-eabi-objcopy.exe -0 binary learn-in-depth.elf learn-in-depth.bin
```

```
Apex@DESKTOP-006R37L MINGW64 /d/full_diploma/ES_Online_Diploma_KS/first_term/unit3/labs/lab1
$ ../../../../c/PROGRAMS/qemu/qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin learn-in-depth:<tasneem>
```