Using VersatilePB virtual board in QEMU and ARM toolchain

1. Writing source files, getting object files (with and without debug information) and analysing them.

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s app.c -o app.o

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s uart.c -o uart.o

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ ls *.o
app.o  uart.o
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/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         00000018  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .debug_info   0000006b  00000000  00000000  000000b0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000058  00000000  00000000  0000011b  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  00000173  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000019f  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000035  00000000  00000000  000001bf  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000090  00000000  00000000  000001f4  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  00000284  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000032  00000000  00000000  00000296  2**0
                  CONTENTS, READONLY
 11 .debug_frame  0000002c  00000000  00000000  000002c8  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/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         00000050  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000084  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000084  2**0
                  ALLOC
  3 .debug_info   0000005c  00000000  00000000  00000084  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000051  00000000  00000000  000000e0  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  00000131  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000015d  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000003d  00000000  00000000  0000017d  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000092  00000000  00000000  000001ba  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  0000024c  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000032  00000000  00000000  0000025e  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000028  00000000  00000000  00000290  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ arm-none-eabi-gcc.exe -c  -I . -mcpu=arm926ej-s app.c -o app.o

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$  arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/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         00000018  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000064  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .comment      00000012  00000000  00000000  000000b0  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  000000c2  2**0
                  CONTENTS, READONLY
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/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         00000050  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000084  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000084  2**0
                  ALLOC
  3 .comment      00000012  00000000  00000000  00000084  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000096  2**0
                  CONTENTS, READONLY
```

2. Writing startup code, getting object file and analysing it.

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ 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

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (master)
$ ls *.o
app.o  startup.o  uart.o

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diplo
ma/Embedded_C/Assignment2/lab1 (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 00000022  00000000  00000000  00000044  2**0
                  CONTENTS, READONLY
```

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

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(master)
$ arm-none-eabi-ld.exe -T linker_script.ld.exe startup.o app.o uart.o -o learn-in-depth.elf -Map=outMap.map
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-ld.exe: cannot open linker script file linker_script.ld.exe: No such file or
 directory
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(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  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000068  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  00010078  00010078  00008078  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000080dc  2**0
                  CONTENTS, READONLY
  4 .comment      00000011  00000000  00000000  0000810a  2**0
                  CONTENTS, READONLY
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(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:               0x10010
  Start of program headers:          52 (bytes into file)
  Start of section headers:          33124 (bytes into file)
  Flags:                             0x5000002, has entry point, Version5 EABI
  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:         9
  Section header string table index: 6

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .startup          PROGBITS        00010000 008000 000010 00  AX  0   0  4
  [ 2] .text             PROGBITS        00010010 008010 000068 00  AX  0   0  4
  [ 3] .data             PROGBITS        00010078 008078 000064 00  WA  0   0  4
  [ 4] .ARM.attributes   ARM_ATTRIBUTES  00000000 0080dc 00002e 00      0   0  1
  [ 5] .comment          PROGBITS        00000000 00810a 000011 01  MS  0   0  1
  [ 6] .shstrtab         STRTAB          00000000 00811b 000049 00      0   0  1
  [ 7] .symtab           SYMTAB          00000000 0082cc 000190 10      8  19  4
  [ 8] .strtab           STRTAB          00000000 00845c 00005e 00      0   0  1
Key to Flags:
```

4. Getting the symbol table for the object files and the final elf file.

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(master)
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
         U Uart_Send_String

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(master)
$ arm-none-eabi-nm.exe uart.o
00000000 T Uart_Send_String

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(master)
$ arm-none-eabi-nm.exe startup.o
         U main
         U resert
00000000 t reset
         U stack_top
00000008 t stop

HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment2/lab1
(master)
$ arm-none-eabi-nm.exe  learn-in-depth.elf
00010010 T main
         U resert
         U reset
00010000 t reset
000110dc D stack_top
00010008 t stop
00010078 D string_buffer
00010028 T Uart_Send_String
```

5. Getting the binary file and simulating the application using QEMU

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment
2/lab1 (master)
$  arm-none-eabi-objcopy.exe  -O binary learn-in-depth.elf   learn-in-depth.bin
```

```
HP@LAPTOP-PPV7E6PE MINGW64 /e/Embedded System/Basic/Embedded_System_Online_Diploma/Embedded_C/Assignment
2/lab1 (master)
$ ../../../../../Unit3/Setup/qemu/qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-dep
th.bin
Learn_in_depth: Mostafa
```