LAB 1
Using VersatilePB virtual board in QEMU and ARM toolchain

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

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s app.c -o app.o

D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-gcc.exe -c -g -I . -mcpu=arm926ej-s uart.c -o uart.o

D:\KS\4.Embedded C\3. Lesson2\Assignment>ls *.o
app.o  uart.o
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>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 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .debug_info   000000cd  00000000  00000000  00000114  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 00000079  00000000  00000000  000001e1  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    0000002c  00000000  00000000  0000025a  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00000286  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   00000049  00000000  00000000  000002a6  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    000000e0  00000000  00000000  000002ef  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      00000012  00000000  00000000  000003cf  2**0
                  CONTENTS, READONLY
 11 .ARM.attributes 00000032  00000000  00000000  000003e1  2**0
                  CONTENTS, READONLY
 12 .debug_frame  0000002c  00000000  00000000  00000414  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>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   000000c1  00000000  00000000  00000084  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000070  00000000  00000000  00000145  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  000001b5  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  000001e1  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000051  00000000  00000000  00000201  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000000e4  00000000  00000000  00000252  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  00000336  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000032  00000000  00000000  00000348  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000028  00000000  00000000  0000037c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o

D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s app.c -o app.o
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>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 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000012  00000000  00000000  00000114  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000126  2**0
                  CONTENTS, READONLY
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>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 analyzing it.**

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>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

D:\KS\4.Embedded C\3. Lesson2\Assignment>ls *.o
app.o  startup.o  uart.o

D:\KS\4.Embedded C\3. Lesson2\Assignment>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.**

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-ld.exe -T linker_script.ld startup.o app.o uart.o -o
 lab1.elf -Map=outMap.map
arm-none-eabi-ld.exe: warning: section `.bss' type changed to PROGBITS

D:\KS\4.Embedded C\3. Lesson2\Assignment>ls *.elf
lab1.elf

D:\KS\4.Embedded C\3. Lesson2\Assignment>ls *.map
outMap.map
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-objdump.exe -h lab1.elf

lab1.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         000000cc  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000064  000100dc  000100dc  000080dc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .bss          00000011  00010140  00010140  00008140  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  4 .ARM.attributes 0000002e  00000000  00000000  00008151  2**0
                  CONTENTS, READONLY
```

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-readelf.exe -a lab1.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:          33220 (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 0000cc 00  AX  0   0  4
  [ 3] .data             PROGBITS        000100dc 0080dc 000064 00  WA  0   0  4
  [ 4] .bss              PROGBITS        00010140 008140 000011 00  WA  0   0  1
  [ 5] .ARM.attributes   ARM_ATTRIBUTES  00000000 008151 00002e 00      0   0  1
  [ 6] .shstrtab         STRTAB          00000000 00817f 000045 00      0   0  1
  [ 7] .symtab           SYMTAB          00000000 00832c 0001a0 10      8  20  4
  [ 8] .strtab           STRTAB          00000000 0084cc 000066 00      0   0  1
```

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

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
00000000 R string_buffer2
         U Uart_Send_String

D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-nm.exe uart.o
00000000 T Uart_Send_String

D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop

D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-nm.exe lab1.elf
00010010 T main
00010000 T reset
00011151 B stack_top
00010008 t stop
000100dc D string_buffer
00010078 T string_buffer2
00010028 T Uart_Send_String
```

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

```
D:\KS\4.Embedded C\3. Lesson2\Assignment>arm-none-eabi-objcopy.exe -O binary lab1.elf lab1.bin

D:\KS\4.Embedded C\3. Lesson2\Assignment>ls *.bin
lab1.bin
```

```
C:\Program Files (x86)\qemu>qemu-system-arm -M versatilepb -m 128M -nographic -kernel lab1.bin
Learn-in-depth: Mai
```