LEC 2 Task (Youssef Samy Youssef)

In this lab1: you have to create a baremetal Software to send a “learn-in-depth:” using UART

Tools:-

- QEMU
- GNU ARM TOOLCHAIN

Board:

```
versatileab             ARM Versatile/AB (ARM926EJ-S)
versatilepb             ARM Versatile/PB (ARM926EJ-S)
vexpress-a15            ARM Versatile Express for Cortex-A
```

Codes:

Create files

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> touch main.c uart.h uart.c
```

uart.c

```c

#define UARTDR_BASE_ADDRESS (*((volatile unsigned int *)0x101f1000))

#include "uart.h"

void UART_send (unsigned char *str){
    while(*str != '\0'){
        UARTDR_BASE_ADDRESS = *str;
        str++;
    }
}
```

uart.h

```c
#ifndef _UART_H_
#define _UART_H_

void UART_send (unsigned char *str);


#endif
```

main.c

```c
#include "uart.h"

unsigned char data[30] = "Learn in Depth: Youssef Samy";

void main(){
    UART_send(data);
}
```

Help

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-gcc --help
Usage: arm-none-eabi-gcc.exe [options] file...
Options:
  -pass-exit-codes         Exit with highest error code from a phase
  --help                   Display this information
  --target-help            Display target specific command line options
  --help={common|optimizers|params|target|warnings|[^]{joined|separate|undocumented}}[,...]
                           Display specific types of command line options
  (Use '-v --help' to display command line options of sub-processes)
  --version                Display compiler version information
  -dumpspecs               Display all of the built in spec strings
  -dumpversion             Display the version of the compiler
  -dumpmachine             Display the compiler's target processor
  -print-search-dirs       Display the directories in the compiler's search path
  -print-libgcc-file-name  Display the name of the compiler's companion library
  -print-file-name=<lib>   Display the full path to library <lib>
  -print-prog-name=<prog>  Display the full path to compiler component <prog>
  -print-multi-directory   Display the root directory for versions of libgcc
  -print-multi-lib         Display the mapping between command line options and
                           multiple library search directories
  -print-multi-os-directory Display the relative path to OS libraries
  -print-sysroot           Display the target libraries directory
  -print-sysroot-headers-suffix Display the sysroot suffix used to find headers
  -Wa,<options>            Pass comma-separated <options> on to the assembler
  -Wp,<options>            Pass comma-separated <options> on to the preprocessor
  -Wl,<options>            Pass comma-separated <options> on to the linker
  -Xassembler <arg>        Pass <arg> on to the assembler
  -Xpreprocessor <arg>     Pass <arg> on to the preprocessor
  -Xlinker <arg>           Pass <arg> on to the linker
  -save-temps              Do not delete intermediate files
  -save-temps=<arg>        Do not delete intermediate files
  -no-canonical-prefixes   Do not canonicalize paths when building relative
                           prefixes to other gcc components
  -pipe                    Use pipes rather than intermediate files
  -time                    Time the execution of each subprocess
  -specs=<file>            Override built-in specs with the contents of <file>
```

Build .c to .o

```
PS C:\Users\asus> arm-none-eabi-gcc -c -g -I . uart.c -o uart.o -mcpu=arm926ej-s
```

```
PS C:\Users\asus> arm-none-eabi-gcc -c -g -I . main.c -o main.o -mcpu=arm926ej-s
```

Objdump help

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump --help
Usage: C:\ARM_tool_chain\bin\arm-none-eabi-objdump.exe <option(s)> <file(s)>
 Display information from object <file(s)>.
 At least one of the following switches must be given:
  -a, --archive-headers    Display archive header information
  -f, --file-headers       Display the contents of the overall file header
  -p, --private-headers    Display object format specific file header contents
  -P, --private=OPT,OPT... Display object format specific contents
  -h, --[section-]headers  Display the contents of the section headers
  -x, --all-headers        Display the contents of all headers
  -d, --disassemble        Display assembler contents of executable sections
  -D, --disassemble-all    Display assembler contents of all sections
  -S, --source             Intermix source code with disassembly
  -s, --full-contents      Display the full contents of all sections requested
  -g, --debugging          Display debug information in object file
  -e, --debugging-tags     Display debug information using ctags style
  -G, --stabs              Display (in raw form) any STABS info in the file
  -W[lLiaprmfFsoRt] or
  --dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,
          =frames-interp,=str,=loc,=Ranges,=pubtypes,
          =gdb_index,=trace_info,=trace_abbrev,=trace_aranges]
                           Display DWARF info in the file
  -t, --syms               Display the contents of the symbol table(s)
  -T, --dynamic-syms       Display the contents of the dynamic symbol table
  -r, --reloc              Display the relocation entries in the file
  -R, --dynamic-reloc      Display the dynamic relocation entries in the file
  @<file>                  Read options from <file>
  -v, --version            Display this program's version number
```

Objdump sections headers

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -h main.o

main.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         00000020  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  0000006c  2**0
                  ALLOC
  3 .debug_info   0000006b  00000000  00000000  0000006c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000058  00000000  00000000  000000d7  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  0000012f  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000015b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000036  00000000  00000000  0000017b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000070  00000000  00000000  000001b1  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  00000221  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000032  00000000  00000000  00000233  2**0
                  CONTENTS, READONLY
 11 .debug_frame  0000002c  00000000  00000000  00000268  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

Objdump dissembler

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -d main.o

main.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <main>:
   0:   e92d4800        push    {fp, lr}
   4:   e28db004        add     fp, sp, #4
   8:   e59f0004        ldr     r0, [pc, #4]    ; 14 <main+0x14>
   c:   ebfffffe        bl      0 <UART_send>
  10:   e8bd8800        pop     {fp, pc}
  14:   00000000        .word   0x00000000
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2>
```

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -d uart.o

uart.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <UART_send>:
   0:   e52db004        push    {fp}            ; (str fp, [sp, #-4]!)
   4:   e28db000        add     fp, sp, #0
   8:   e24dd00c        sub     sp, sp, #12
   c:   e50b0008        str     r0, [fp, #-8]
  10:   ea000006        b       30 <UART_send+0x30>
  14:   e59f3030        ldr     r3, [pc, #48]   ; 4c <UART_send+0x4c>
  18:   e51b2008        ldr     r2, [fp, #-8]
  1c:   e5d22000        ldrb    r2, [r2]
  20:   e5832000        str     r2, [r3]
  24:   e51b3008        ldr     r3, [fp, #-8]
  28:   e2833001        add     r3, r3, #1
  2c:   e50b3008        str     r3, [fp, #-8]
  30:   e51b3008        ldr     r3, [fp, #-8]
  34:   e5d33000        ldrb    r3, [r3]
  38:   e3530000        cmp     r3, #0
  3c:   1afffff4        bne     14 <UART_send+0x14>
  40:   e28bd000        add     sp, fp, #0
  44:   e8bd0800        ldmfd   sp!, {fp}
  48:   e12fff1e        bx      lr
  4c:   101f1000        .word   0x101f1000
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2>
```

Objdump source

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -s uart.o

uart.o:     file format elf32-littlearm

Contents of section .text:
 0000 04b02de5 00b08de2 0cd04de2 08000be5  ..-.......M.....
 0010 060000ea 30309fe5 08201be5 0020d2e5  ....00... ... ..
 0020 002083e5 08301be5 013083e2 08300be5  . ...0...0...0..
 0030 08301be5 0030d3e5 000053e3 f4ffff1a  .0...0....S.....
 0040 00d08be2 0008bde8 1eff2fe1 00101f10  ........../.....
Contents of section .debug_info:
 0000 58000000 02000000 00000401 0e000000  X...............
 0010 01240000 002b0000 00000000 00500000  .$...+.......P..
 0020 00000000 0002011a 00000001 06010000  ................
 0030 00005000 00000000 0000014e 00000003  ..P........N....
 0040 73747200 01064e00 00000291 74000404  str...N.....t...
 0050 54000000 05010800 00000000           T...........
Contents of section .debug_abbrev:
 0000 01110125 0e130b03 0e1b0e11 01120110  ...%............
 0010 06000002 2e013f0c 030e3a0b 3b0b270c  ......?...:.;.'.
 0020 11011201 40069742 0c011300 00030500  ....@..B........
 0030 03083a0b 3b0b4913 020a0000 040f000b  ..:.;.I.........
 0040 0b491300 00052400 0b0b3e0b 030e0000  .I....$...>.....
 0050 00                                   .
Contents of section .debug_loc:
 0000 00000000 04000000 02007d00 04000000  ..........}.....
 0010 08000000 02007d04 08000000 50000000  ......}.....P...
 0020 02007b04 00000000 00000000           ..{.........
Contents of section .debug_aranges:
 0000 1c000000 02000000 00000400 00000000  ................
 0010 00000000 50000000 00000000 00000000  ....P...........
Contents of section .debug_line:
 0000 39000000 02001d00 00000201 fb0e0d00  9...............
 0010 01010101 00000001 00000100 75617274  ............uart
 0020 2e630000 00000000 05020000 00001783  .c..............
 0030 2f830002 04016486 02080001 01        /.....d......
Contents of section .debug_str:
 0000 756e7369 676e6564 20636861 7200474e  unsigned char.GN
 0010 55204320 342e372e 32005541 52545f73  U C 4.7.2.UART_s
 0020 656e6400 75617274 2e630044 3a5c5072  end.uart.c.D:\Pr
 0030 6f6a6563 74735c6c 6561726e 5f696e5f  ojects\learn_in_
 0040 64657074 685f776f 726b7370 6163655c  depth_workspace\
 0050 756e6974 5f335f65 6d626564 6465645f  unit_3_embedded_
 0060 635c6c65 633200                      c\lec2.
Contents of section .comment:
 0000 00474343 3a202847 4e552920 342e372e  .GCC: (GNU) 4.7.
 0010 3200                                 2.
Contents of section .ARM.attributes:
 0000 41310000 00616561 62690001 27000000  A1...aeabi..'...
 0010 0541524d 39323645 4a2d5300 06050801  .ARM926EJ-S.....
 0020 09011204 14011501 17031801 19011a01  ................
 0030 1e06                                 ..
Contents of section .debug_frame:
 0000 0c000000 ffffffff 0100027c 0e0c0d00  ...........|....
 0010 14000000 00000000 50000000           ............P...
 0020 420e048b 01420d0b                    B....B..
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2>
```

Startup.s file

```
startup.s | main.c | uart.h | uart.c | linker_script.ld

1  .globl reset
2
3  reset:
4      ldr sp, =stack_top
5      bl main
6  stop: b stop
```

Assemlber startup.s

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-as -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
```

Objdump

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -d startup.o

startup.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <reset>:
   0:   e59fd004        ldr     sp, [pc, #4]    ; c <stop+0x4>
   4:   ebfffffe        bl      0 <main>

00000008 <stop>:
   8:   eafffffe        b       8 <stop>
   c:   00000000        .word   0x00000000
```

See symbols

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-nm main.o
00000000 D data
00000000 T main
         U UART_send
```

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-nm uart.o
00000000 T UART_send
```

Linker script

```
ENTRY(reset)

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

SECTIONS{
    . = 0x10000;

    .startup . :
    {
        startup.o(.text)
    }>Mem

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

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

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

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

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

Linke object files

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-ld -T linker_script.ld uart.o main.o -o learn-in-depth.elf
```

Objdump

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-nm learn-in-depth.elf
00010078 D data
00010060 T main
00010000 T reset
00011098 D stack_top
00010008 t stop
00010010 T UART_send
```

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objdump -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         00000020  00010078  00010078  00008078  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  00008098  2**0
                  CONTENTS, READONLY
  4 .comment      00000011  00000000  00000000  000080c6  2**0
                  CONTENTS, READONLY
  5 .debug_info   000000c7  00000000  00000000  000080d7  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev 000000a9  00000000  00000000  0000819e  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_loc    00000058  00000000  00000000  00008247  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000040  00000000  00000000  0000829f  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_line   00000073  00000000  00000000  000082df  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    00000081  00000000  00000000  00008352  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_frame  00000054  00000000  00000000  000083d4  2**2
                  CONTENTS, READONLY, DEBUGGING
```

Read elf

```
                  CONTENTS, READONLY, DEBUGGING
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-readelf -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:               0x10000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          33996 (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:         16
  Section header string table index: 13

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 000020 00  WA  0   0  4
  [ 4] .ARM.attributes   ARM_ATTRIBUTES  00000000 008098 00002e 00      0   0  1
  [ 5] .comment          PROGBITS        00000000 0080c6 000011 01  MS  0   0  1
  [ 6] .debug_info       PROGBITS        00000000 0080d7 0000c7 00      0   0  1
  [ 7] .debug_abbrev     PROGBITS        00000000 00819e 0000a9 00      0   0  1
  [ 8] .debug_loc        PROGBITS        00000000 008247 000058 00      0   0  1
  [ 9] .debug_aranges    PROGBITS        00000000 00829f 000040 00      0   0  1
  [10] .debug_line       PROGBITS        00000000 0082df 000073 00      0   0  1
  [11] .debug_str        PROGBITS        00000000 008352 000081 01  MS  0   0  1
  [12] .debug_frame      PROGBITS        00000000 0083d4 000054 00      0   0  4
  [13] .shstrtab         STRTAB          00000000 008428 0000a1 00      0   0  1
  [14] .symtab           SYMTAB          00000000 00874c 000200 10     15  27  4
  [15] .strtab           STRTAB          00000000 00894c 000048 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)
```

Generate binary files

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> arm-none-eabi-objcopy -O binary learn-in-depth.elf learn-in-depth.bin
```

Simulate the code

```
PS D:\Projects\learn_in_depth_workspace\unit_3_embedded_c\lec2> qemu-system-arm -M versatilepb
 -m 128M -nographic -kernel learn-in-depth.bin
Learn in Depth: Youssef Samy
```