

Tutorial 6

Question 1: (7)

- a) Compare and contrast assembly and C. (3)
- b) What is the ENTRY point of a program? Is it different to the reset vector? How so? (2)
- c) What do the -S and -Wall flags which are passed to GCC do? (2)

Question 2: (6)

- a) What is the advantage of writing our own linker script? (1)
- b) Into which section does executable machine code go? (1)
- c) Why do we want to pull out our vectors into a different section? (1)
- d) Write a line which could go into the linker script which would: (3)
 - create an output section called out_section
 - consume input sections: in1 from all files and in2 from file bar.o
 - cause the output section to be placed in memory block called RAM.

See the linker manual for help. Starting points:

<https://sourceware.org/binutils/docs/ld/SECTIONS.html#SECTIONS>

<https://sourceware.org/binutils/docs/ld/Input-Section-Basics.html#Input-Section-Basics>

<https://sourceware.org/binutils/docs/ld/Input-Section-Example.html#Input-Section-Example>

Question 3: (10)

- a) What does `#include <stdint.h>` give us access to? (1)
- b) We never want our main file to return, yet we give it return type of int. Why? (1)
- c) What does the "void" in `main(void)` indicate? (1)
- d) Explain what following code extract does: (2)
`*(int16_t*)0x08000100`
- e) The asterisk is used in two context in the above. What does it mean in each of these? (2)
- f) Write equivalent assembly code for: (3)
`*(uint16_t*)0x20000000 = (*(uint32_t*)0x20000FF0) + 42;`

Marked out of: 23