

ME 333 Introduction to Mechatronics  
Quiz 4

1. Explain what is stored in a **heap** and what is stored in a **stack**. (2 points)

The heap contains dynamically allocated memory.

The stack contains temporary local variables.

2. Briefly describe what is in a map file. (2 pts)

Indicates where instructions are placed in program memory and where global variables are placed in data memory.

3. Which of these is responsible for setting the final virtual addresses for program instructions and global variables: the preprocessor, the compiler, the assembler, or the linker? (1 pt)

The linker

4. Explain one purpose that the C runtime startup code serves. (1 pt)

Initializes global variables, uninitialized global variables set to 0.

5. Explain why we disable the `-O1` when compiling to assembly code in Chapter 5. (1 pt)

`-O1` optimizes the code and would delete many of the "pointless" commands being timed/measured in Ch5.

6. Explain why the C line `LATAINV = 0x20;` generates fewer lines of assembly code than the line `LATABits.LATA5 = !LATABits.LATA5;` (Both lines simply invert one of LATA.) (2 pts)

The second method copied, manipulated, then recopied (9 steps) rather than just manipulating (3 steps).

7. After invoking `NU32_startup()`; at the beginning of your program, you must follow 7 steps to set up and use the interrupt. Fill in the missing steps. (Just a phrase, no code.) (3 pts)

1. Write the ISR, which clears the interrupt flag.
2. Disable interrupts at the CPU.
3. Configure the operating mode of the peripheral that will be generating IRQs.
4. Configure interrupt priority and subpriority.
5. Clear the interrupt flag to 0 (IFSx).
6. Set interrupt enable bit to 1 (IECx).
7. Enable interrupts at the CPU.

8. What is the difference between an IRQ number and an interrupt vector number? (2 pts)

There are more IRQs (96) than interrupt vectors (64). The IRQ defines what triggers the interrupt. The vector number determines the address of an ISR in virtual memory.

9. Explain the purpose of the IFS, IEC, and IPC special function registers. (3 pts)

IFS – Interrupt flag – Is interrupt triggered?

IPC – Interrupt priority – priority

IEC – Enable interrupt – interrupt enabled?

10. What is "context save and restore"? (1 pt)

When register data is copied and saved to RAM when an ISR is called, and then restored after the ISR completes.

11. Why is it a good idea to define a global variable, shared by an ISR and the main code, with the type qualifier `volatile`? (2 pts)

Volatile forces the value of the variable to always be read from RAM. This way, if the variable is changed in an ISR, the `main()` function will use the new value instead of the value that was stored in cache before the ISR update.