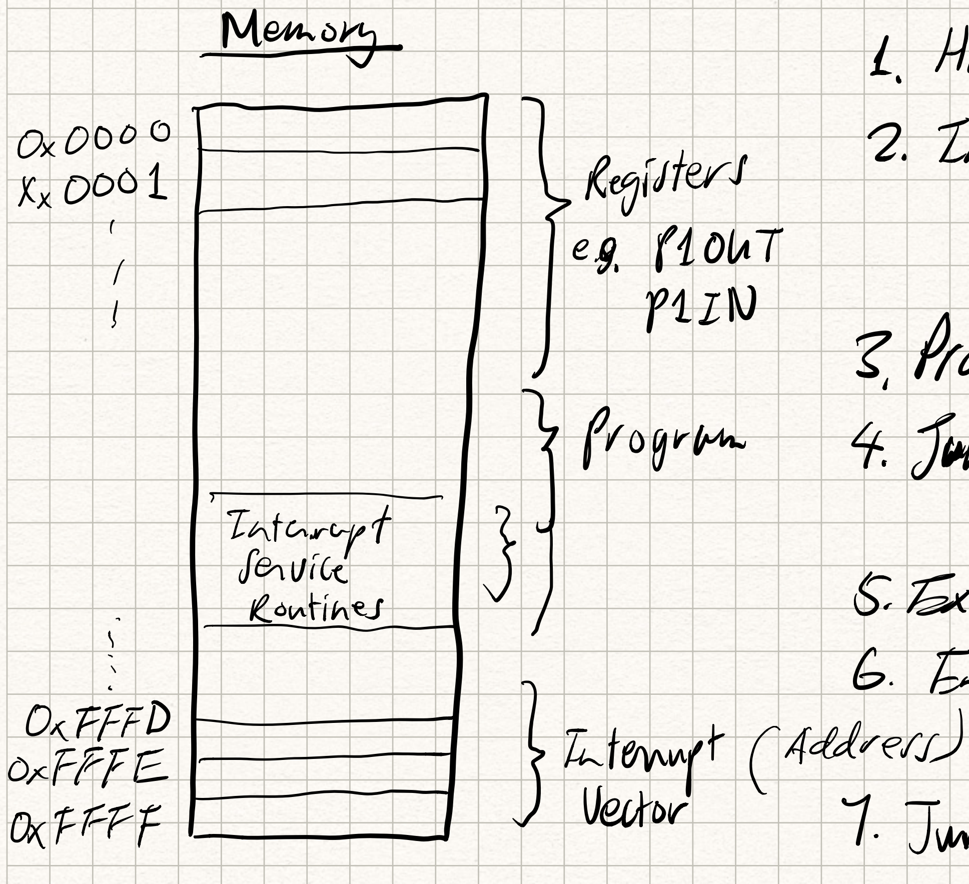
MECH 423 Lecture #7

Anatomy of an MCU program (Simple Version)

```
#include ....      ← header files

int main (void)    ← start of C program
{
    // Turn off the Watch dog timer (WDT)
    // Set up registers (SFR)   ← most of your code.
    while (1)   // infinite loop
      {  // more code
      }
         ← Note: MCU programs cannot end.
```

Interrupts



Interrupt Sequence

1. Hardware event.
2. Interrupt flag (1 bit)
   - IFG
   - Set regardless if interrupt is enable
3. Program Halts
4. Jump to address in the interrupt Vector
5. Execute ISR
6. End of ISR ⇒ Reset IFG
7. Jump back to original program

Potential Problems

① If interrupt is enabled

The interrupt vector must have a valid address

$\Rightarrow$ Must have an ISR

② IFG must be cleared at the end of the ISR. Otherwise, the program will interrupt again.

Caveat: Some interrupts are auto-clearing (e.g. UART)

Structure of an MCU program (interrupt version)

```
# include ...
int main (void)
    // Stop WDT
    // Set up registers                      } most
        - enable specific interrupts        } of your code
    // Global interrupt enable
        _EINT();
    // Infinite loop     while (1);
                                              look up
                                              in header
                                              file
// ISR
#pragma     vector = VECTOR_NAME
__interrupt  void  ISR_NAME  <-- user defined.
{
    // ISR code
    // Reset IFG
}
```