The I/O operations that you used earlier in writing assembly language programs are not real – they work with the SPIM simulator only. This lecture outlines the major things that you need to know to perform I/O operations with real machines.

**Memory mapped I/O**

Determines how the devices are addressed. Each device register is assigned an address similar to the memory addresses.
The address space is shared by ROM, RAM, and Input Output device registers. Thus any device can be accessed and controlled in the same way a memory location is read or written

```
lw $t0, 0x00000004  # Read ROM
sw $t0, 0x00000004  # Write ROM error!
lbu $t0, 0x0000fffc1 # Read RAM
sb $t0, 0x0000fffc1 # Write RAM
lbu $t0, 0xffff0000  # Read an I/O dev
sb $t0, 0xffff0004  # Write to I/O dev
```
Communicating with I/O devices

1. Programmed I/O or Polled I/O

We will discuss about interfacing a keyboard as an input device. Polled I/O is very basic, but inefficient, since it uses spin-wait or busy-wait.

SPIM uses

0xffff0004 as data register (only the last 8-bits), and
0xffff0000 as the status and control register
(bit 0 is the ready bit, automatically cleared after the data is read, bit 1 is the interrupt enable bit)
2. Interrupt driven I/O
Efficient method of communication with I/O devices

3. Direct Memory Access (DMA)
Used for communication with high-speed devices like Disk.

*These will be explained in the class*