I/O Channel Behavior Depends on Type of Channel

But what's the point?

Er.*

Remember that file descriptors are
- indices into an array of I/O channels
- controlled by the OS?

The behavior of each I/O channel
- depends on the type of the channel
- (keyboard, display, file, and so forth).

*Sorry.

Behavior is Implemented with Function Pointers

In **the I/O channel** array,
- each **array element**
- **includes several function pointers**.

The functions define a channel's behavior.

To implement a new channel type,
- implement a function for each operation (read, write, and so forth), then
- use the addresses of the new functions to form the array element for a channel of the new type.

Can Also Specialize Data Structures or Other Functions

Function pointers can be used to specialize behavior in other ways.

Behavior of **operations on a structure**:
- different **subtypes use different functions**;
- called **'virtual' functions** in **C++** and **Java**

Behavior of **a function X**:
- pointer to **Y** passed as argument;
- **X calls back to Y** to execute an operation;
- **X** is said to **use a callback** to implement the operation.

Let's Generalize Insertion Sort to Operate on Any Type

Do you remember
- that I said we could copy the code
- for the insertion sort on integers
- to create an insertion sort on strings?

Now let's
- **generalize the code** to
- **sort an array of "things."**

We'll use function pointers.

