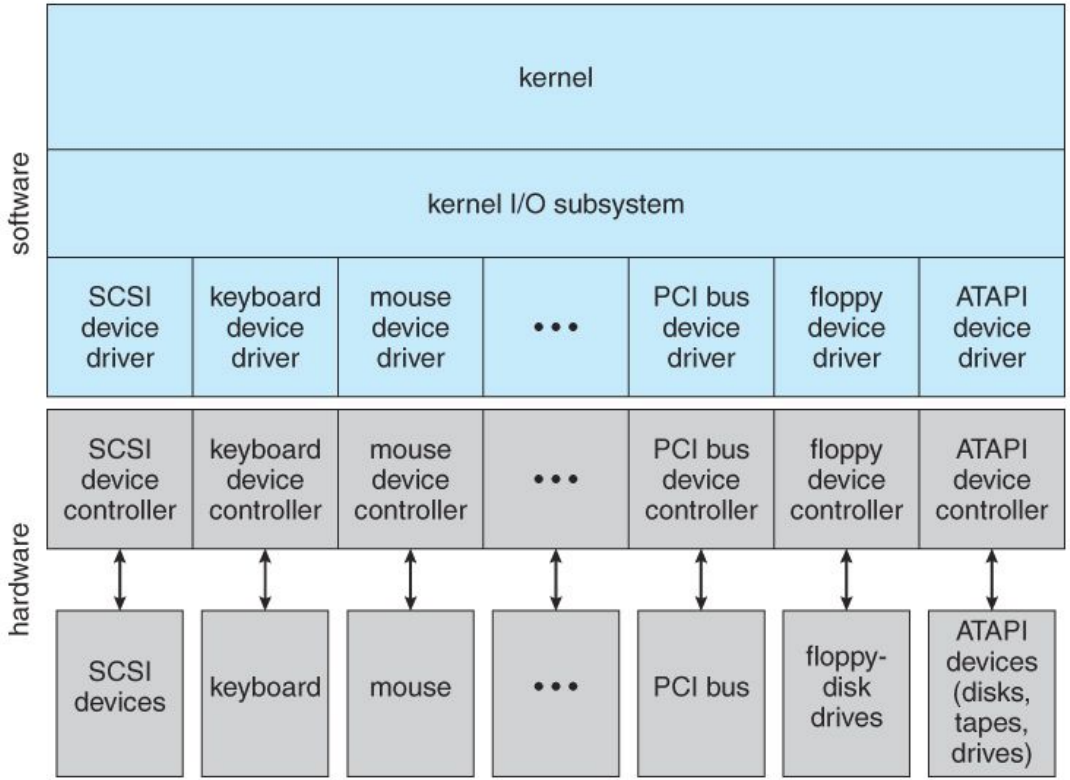
Application I/O Interface

Kernel I/O Structure



Characteristics of I/O Devices

- ***Character-stream or block:*** A character-stream device transfers bytes one by one, whereas a block device transfers a block of bytes as a unit
- ***Sequential or random access:*** A sequential device transfers data in a fixed order determined by the device, whereas the user of a random-access device can instruct the device to seek to any of the available data storage locations
- ***Synchronous or asynchronous:*** A synchronous device performs data transfers with predictable response times, in coordination with other aspects of the system. An asynchronous device exhibits irregular or unpredictable response times not coordinated with other computer events
- ***Sharable or dedicated:*** A sharable device can be used concurrently by several processes or threads; a dedicated device cannot

Characteristics of I/O Devices

- ***Speed of operation:*** Device speeds range from a few bytes per second to gigabytes per second.
- ***Read–write, read only, write once:*** Some devices perform both input and output, but others support only one data transfer direction. Some allow data to be modified after write, but others can be written only once and are read-only thereafter.

Characteristics of I/O Devices

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read–write	CD-ROM graphics controller disk

Block and Character Devices

- The name “block device” comes from the fact that the corresponding hardware typically reads and writes a whole block at a time (e.g. a sector on a hard disk)
- Data from block device can be cached in memory and read back from cache; writes can be buffered
- Character devices (also called character special files) behave like pipes, serial ports, etc. Writing or reading to them is an immediate action
- The name “character device” comes from the fact that each character is handled individually

Block and Character Devices

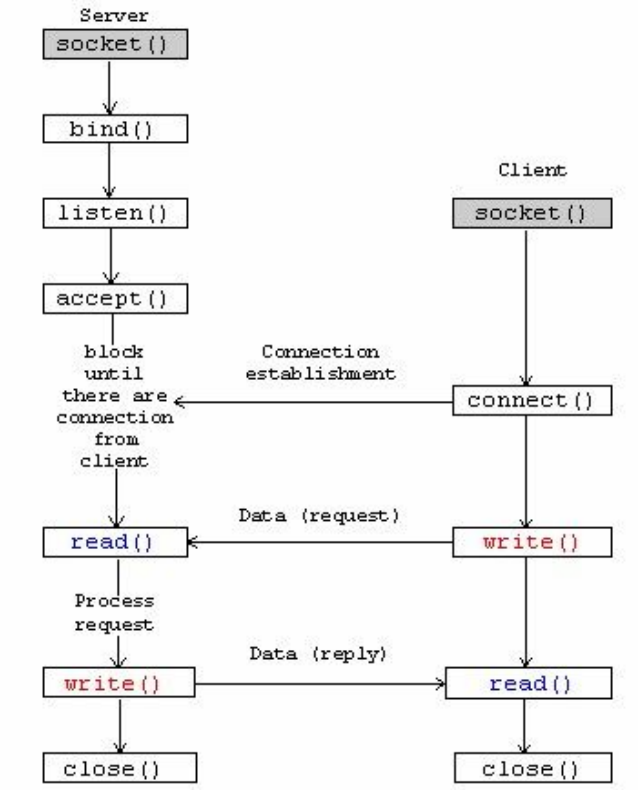
- when the kernel loads the correct driver (either at boot time, or via programs like udev) it scans the various buses to see if any devices handled by that driver are actually present on the system.
- If so, it sets up a device that 'listens' on the appropriate major/minor number.
- For instance, the Digital Signal Processor of the first audio card found by your system gets the major/minor number pair of 14/3; the second gets 14,35, etc.
- It's up to udev to create an entry in /dev named dsp as a character device marked major 14 minor 3.

Block and Character Devices

- Then, when a userspace program tries to access a file that's marked as a 'character special file' with the appropriate major/minor number (for instance, your audio player trying to send digital audio to /dev/dsp)
  - the kernel knows that this data needs to be transmitted via the driver that major/minor number is attached to
  - driver knows what to do with it in turn.
- `ls -l /dev/sda*`

```
brw-rw---- 1 root disk 8, 0 Mar 16 09:18 /dev/sda
brw-rw---- 1 root disk 8, 1 Mar 16 09:18 /dev/sda1
brw-rw---- 1 root disk 8, 2 Mar 16 09:18 /dev/sda2
brw-rw---- 1 root disk 8, 3 Mar 16 09:18 /dev/sda3
```

Network Devices - Socket



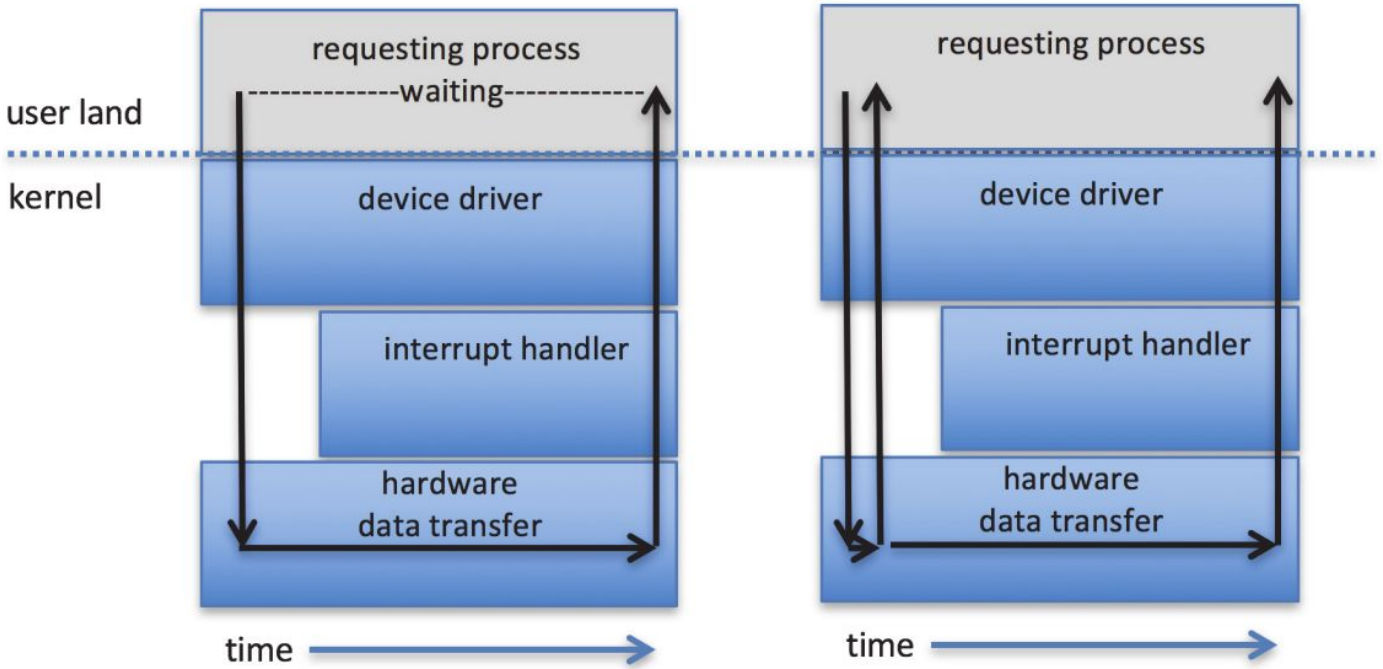
Clocks and Timers

- Give the current time.
- Give the elapsed time.
- Set a timer to trigger operation X at time T
- Wait certain time and generate interrupt
- Virtual timers

Nonblocking and Asynchronous I/O

- Blocking -
  - When an application issues a blocking system call, the execution of the calling thread is suspended.
  - The thread is moved from the operating system's run queue to a wait queue.
  - After the system call completes, the thread is moved back to the run queue, where it is eligible to resume execution.
  - When it resumes execution, it will receive the values returned by the system call.
- Nonblocking example -
  - A video application that reads frames from a file on disk while simultaneously decompressing and displaying the output on the display.

Synchronous vs Asynchronous



Asynchronous vs. Nonblocking?

Vectored I/O

Vectored I/O allows one system call to perform multiple I/O operations involving multiple locations (See code)