

I/O System

- The *I/O system* hides the peculiarities of I/O devices from the bulk of the kernel.
- Consists of a buffer caching system, general device driver code, and drivers for specific hardware devices.
- Only the device driver knows the peculiarities of a specific device.

4.3 BSD Kernel I/O Structure

system-call interface to the kernel				
socket	plain file	cooked block interface	raw block interface	raw tty interface
protocols	file system			
network interface	block-device driver		character-device driver	
the hardware				

Block Buffer Cache

- Consists of buffer headers, each of which can point to a piece of physical memory, as well as to a device number and a block number on the device.
- The buffer headers for blocks not currently in use are kept in several linked lists:
 - Buffers recently used, linked in LRU order (LRU list).
 - Buffers not recently used, or without valid contents (AGE list).
 - EMPTY buffers with no associated physical memory.
- When a block is wanted from a device, the cache is searched.
- If the block is found, it is used, and no I/O transfer is necessary.
- If it is not found, a buffer is chosen from the AGE list, or the LRU list if AGE is empty.

Block Buffer Cache (Cont'd)

- Buffer cache size effects system performance; if it is large enough, the percentage of cache hits can be high and the number of actual I/O transfers low.
- Data written to a disk file are buffered in the cache, and the disk driver sorts its output queue according to disk address — these actions allow the disk driver to minimize disk head seeks and to write data at times optimized for disk rotation.

Raw Device Interfaces

- Almost every block device has a character interface, or *raw device interface* — unlike the block interface, it bypasses the block buffer cache.
- Each disk driver maintains a queue of pending transfers.
- Each record in the queue specifies:
 - whether it is a read or a write
 - a main memory address for the transfer
 - a device address for the transfer
 - a transfer size
- It is simple to map the information from a block buffer to what is required for this queue.

C-Lists

- Terminal drivers use a character buffering system which involves keeping small blocks of characters in linked lists.
- A **write** system call to a terminal enqueues characters on a list for the device. An initial transfer is started, and interrupts cause dequeuing of characters and further transfers.
- Input is similarly interrupt driven.
- It is also possible to have the device driver bypass the canonical queue and return characters directly from the raw queue — *raw mode* (used by full-screen editors and other programs that need to react to every keystroke).