

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				

Sockets

- A socket is an endpoint of communication.
- An in-use socket it usually bound with an *address*; the nature of the address depends on the *communication domain* of the socket.
- A characteristic property of a domain is that processes communicating in the same domain use the same *address format*.
- A single socket can communicate in only one domain — the three domains currently implemented in 4.3 BSD are:
 - the UNIX domain (AF_UNIX)
 - the Internet domain (AF_INET)
 - the XEROX Network Services (NS) domain (AF_NS)

Socket Types

- **Stream sockets** provide reliable, duplex, sequenced data streams. Supported in Internet domain by the TCP protocol. In UNIX domain, pipes are implemented as a pair of communicating stream sockets.
- **Sequenced packet sockets** provide similar data streams, except that record boundaries are provided. Used in XEROX AF_NS protocol.
- **Datagram sockets** transfer messages of variable size in either direction. Supported in Internet domain by UDP protocol.
- **Reliably delivered message sockets** transfer messages that are guaranteed to arrive. Currently unsupported.
- **Raw sockets** allow direct access by processes to the protocols that support the other socket types; e.g., in the Internet domain, it is possible to reach TCP, IP beneath that, or a deeper Ethernet protocol. Useful for developing new protocols.

Socket System Calls

- The **socket** call creates a socket; takes as arguments specifications of the communication domain, socket type, and protocol to be used and returns a small integer called a *socket descriptor*.
- A name is bound to a socket by the **bind** system call.
- The **connect** system call is used to initiate a connection.
- A server process uses **socket** to create a socket and **bind** to bind the well-known address of its service to that socket.
 - Uses **listen** to tell the kernel that it is ready to accept connections from clients.
 - Uses **accept** to accept individual connections.
 - Uses **fork** to produce a new process after the **accept** to service the client while the original server process continues to listen for more connections.

Socket System Calls (Cont'd)

- The simplest way to terminate a connection and to destroy the associated socket is to use the **close** system call on its socket descriptor.
- The **select** system call can be used to multiplex data transfers on several file descriptors and/or socket descriptors.

Network Support

- Networking support is one of the most important features in 4.3 BSD.
- The socket concept provides the programming mechanism to access other processes, even across a network.
- Sockets provide an interface to several sets of protocols.
- Almost all current UNIX systems support UUCP.
- 4.3 BSD supports the DARPA Internet protocols UDP, TCP, IP, and ICMP on a wide range of Ethernet, token-ring, and ARPANET interfaces.
- The 4.3 BSD networking implementation, and to a certain extent the *socket* facility, is more oriented toward the ARPANET Reference Model (ARM).

Network Reference Models and Layering

ISO reference model	ARPANET reference model	4.2BSD layers	example layering
application	process applications	user programs and libraries	telnet
presentation			sock_stream
session		sockets	
transport	host–host	protocol	TCP
network data link			IP
hardware	network interface	network interfaces	Ethernet driver
	network hardware	network hardware	interlan controller