SE 4C03 Winter 2002

10. The Socket Interface

Instructor: W. M. Farmer

Revised: 31 March 2002

The Socket Interface

- The **socket interface** is an interface for application programs to establish communication channels using TCP/IP protocols (as well as other protocols)
- The socket interface can be implemented either
  - Directly in the operating system or
  - By a set of library routines
- A **socket** is the end point of a communication channel
  - Is a generalization of a Unix file
- The socket interface is becoming a de facto standard

Functions for Constructing a Socket

- **Create**: creates a socket
- **Bind**: establishes a local protocol port for a socket
  - Usually only called by a server process
- **Connect**: connects a socket to a destination IP address and protocol port
  - For TCP, a TCP connection is established
  - For UDP, no connection is made, but the destination address and port number are stored
- **Listen**: enables a server process to listen to a socket
- **Accept**: blocks a server process until a connection request arrives and then creates a new socket

Sending and Receiving Data

- Data is sent through a socket using various kinds of **write** functions
- Data is received through a socket using various kinds of **read** functions

Other Socket Interface Functions

- Functions for getting socket attributes such as:
  - Source and destination IP addresses
  - Source and destination IP protocol ports
- Functions for getting and setting socket options such as:
  - Buffer sizes
  - Timeout parameters
- Library functions that provide network services such as:
  - DNS queries
  - Host information
  - Network information
  - Protocol information
- **Close**: closes a socket