Introduction to Socket Programming

By Pedro Paulo Ferreira Bueno and Antonio Pires de Castro Junior.

Published in Issue 47 of Linux Gazette, November 1999
http://linuxgazette.net/issue47/bueno.html

- Wei-li Tang
- July 1st, 2009.

Outline

- Introduction
- The Socket Function
- The Mini-chat Server structure
- The Mini-chat Client structure
- Conclusion
- Lab

Introduction

- Socket
  - A socket is an endpoint used by a process for bi-directional communication with a socket associated with another process.
  - Introduced in Berkeley Unix, are a basic mechanism for IPC on a computer system
  - On different computer systems connected by local or wide area networks

The Socket Function

- Most network applications can be divided into two pieces: a client and a server.
- Creating a socket
  - #include <sys/types.h>
  - #include <sys/socket.h>
- Specify 3 main parameters
  - the domain
  - the type
  - the protocol
- int socket(int domain, int type, int protocol);

The Socket Function

- domain: Communication Domain
  - IPv4: AF_INET
  - IPv6: AF_INET6
- type: Communication Semantics
  - SOCK_STREAM: Reliable, 2-way stream (e.g. TCP)
  - SOCK_DGRAM: Unreliable, connectionless datagram (e.g. UDP)
- protocol: The Protocol Type
  - 0: IP, 1: ICMP, 6: TCP, 17: UDP, etc.

The Socket Function

- s = socket(AF_INET , SOCK_STREAM , 0);
  - s: The file descriptor returned by the socket function.

- Since our mini chat is divided in two parts we will divided the explanation in the server, the client and the both, showing the basic differences between them, as we will see next.

The Mini-chat Server structure

- Binding a socket to a port and waiting for the connections
  - The sockets are always associated with a port
    - e.g. Telnet: port 23, FTP: port 21.
    - Since the system pre-defined a lot of ports between 1 and 7000 (/etc/services) we choose the port number larger than 10000. (For example: 15000.)
  - Step 1: Bind a port
  - Step 2: Listen the socket to wait for connections
  - Step 3: Accept a connection

The Mini-chat Server structure

- Step 1: Bind a port
  - int bind(int s, struct sockaddr *addr, int addrlen);
    - struct sockaddr_in: the information about socket
- struct sockaddr_in
  - address.sin_family = AF_INET
    - Use a internet domain
  - address.sin_addr.s_addr = INADDR_ANY
    - Use a specific IP of host
  - address.sin_port = htons(15000);
    - Use a specific port number
- bind(create_socket , (struct sockaddr *)&address,sizeof(address));

The Mini-chat Server structure

- Step 2: Listen the socket to wait for connections
  - Prepare a socket to accept messages from clients
  - listen (create_socket, MAXNUMBER);
    - MAXNUMBER: the maximum number of pending connections. (For example, 3)

The Mini-chat Server structure

- Step 3: Accept a connection.
  - Accept is used with connection based sockets such as streams.
  - accept(create_socket,(struct sockaddr *)&address,&addrlen);

The Mini-chat Client structure

- The biggest difference is that client needs a Connect() function.
  - Used on the client side to identify and, possibly, start the connection to the server.
- connect(create_socket,(struct sockaddr *)&address,sizeof(address)) ;

The common structure

- struct hostent
  - Used to represent an entry in the hosts database.
- The use of the Send and Recv functions are another common codes.
  - send(new_socket,buffer,bufsize,0);
    - Send the buffer to the server.
  - recv(new_socket,buffer,bufsize,0);
    - Receive the buffer from the server

Conclusion

- The exactly interface between an application and the TCP/IP protocols depends of the details of the operating system.
- In our case we examine the UNIX BSD socket interface because Linux follow this.
- 與其介紹 Client/Server model 不如動手實做簡單的聊天室！

Further Study

- IPC (Inter-process Communication)
- fork
- threads
- … and much more.

Lab

- Example Codes
  - http://linuxgazette.net/issue47/misc/bueno/client.c
  - http://linuxgazette.net/issue47/misc/bueno/server.c
- Compile it
  - gcc -o client client.c
  - gcc -o server server.c
- Run the server
- Run the client with the address of the server

Resource

- Books
  - Internetworking with TCP/IP Vol1 - Doulgas Commer
  - Unix Network Programming, Vol2 , Richard Stevens
  - Unix Network Programming, Vol1, Richard Stevens
    - Third Edition (Recommended)
- Documents
  - Socket Linux Manual Page (man socket)
  - /etc/protocols
  - /etc/services

Q&A

Thanks!

Wei-li Tang
alextwl@ms11.voip.edu.tw

July 1st, 2009.