CHAT PROGRAM: AN APPLICATION OF SOCKET INTERFACE

Dr. Shankar K. Ghosh

Assistant Professor

Computer Science and Engineering

Shiv Nadar Institution of Eminence, Delhi NCR

Application requirement

- *Chat program:*
  - Is not delay sensitive.
  - Require guaranteed and ordered delivery.
  - Require connection oriented service.
  - Socket options: SOCK_STREAM, AF_INET.

Some necessary system calls

- ***sendto:*** sends a datagram from a buffer on a socket using a socket address and address length. Its prototype is essentially:

  int sendto(int sockfd, void *buffer, size_t len, int flags,struct sockaddr *to, socklen_t tolen);

  In normal use, the **flags** parameter can be kept **zero**. (For **stream socket** use **write** system call)

- ***recvfrom*****:** waits on a socket for a datagram from a specified address and receives it into a buffer. Its prototype is essentially

  int recvfrom(int sockfd, void *buffer, size_t len, int flags, struct sockaddr *from, socklen_t *fromlen);

  Again, in normal use, the **flags** parameter can be kept **zero**.(For **stream socket** use **read** system call)

Some necessary system calls

- #include <strings.h>

  ```
  void bzero(void *s, size_t  n);
  ```

  The ***bzero()*** function sets the first ***n*** bytes of the area starting at ***s*** to ***zero*** (bytes containing '\0').

Including necessary headers

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
```

The main() function

```
main()
{
   int sockid;
   int bindid;
   struct sockaddr_in myaddr;
   struct sockaddr_in client;
   int newsockid;
   int clientlen;
  int n;
  char msg[1000];
  int recvid, sendid;
  int port_id = 6000;
```

Creating the socket and initializing the addresses

```
sockid = socket(AF_INET, SOCK_STREAM, 0);

// i.e TCP protocol is in use

bzero((char*)&myaddr, sizeof(struct sockaddr));

myaddr.sin_family = AF_INET;

myaddr.sin_addr.s_addr = htonl(INADDR_ANY);

myaddr.sin_port = htons(port_id);
```

Creating the named socket and the queue

```
bindid = bind(sockid, (struct sockaddr*)&myaddr, sizeof(struct sockaddr_in));

if(bindid < 0)

printf("error \n");

listen(sockid, 5); // The server can chat with (maximum) 5 clients simultaniously
```

Accepting connections

```
clientlen = sizeof(struct sockaddr_in);

newsockid = accept(sockid, (struct sockaddr*)&client, &clientlen);

if(newsockid < 0)

    printf("error 2\n");
```

Read and write:chat

```
 while(1) {
recvid = recvfrom(newsockid, msg, sizeof(msg), 0, (struct sockaddr*)&client, &clientlen);
if(recvid < 0)
printf("error 2\n");
printf("%s \n", msg);
 bzero(msg,1000);
 n=0;
 printf("\n write to the client:");
 while((msg[n++]=getchar())!='\n');
 sendto(newsockid,msg,sizeof(msg),0,(struct sockaddr*)&client, sizeof(struct sockaddr_in));
 if(strncmp("bye",msg,3)==0)
          {
               printf("Exit session...\n");
               break;
            }
  }
```

Code for client side

```
servaddr.sin_family = AF_INET;

servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
 // IP address of the server, here loopback address

servaddr.sin_port = htons(port_id);

/* Connect to the server */

connectid = connect(sockid, (struct
sockaddr*)&servaddr, sizeof(struct sockaddr_in));
```

Code for client side

```
 While(1) {
        printf("\n write to the server:");
        n=0;
        while((msg[n++]=getchar())!='\n');
          sendto(sockid,msg,sizeof(msg),0,(struct sockaddr*)&servaddr, sizeof(struct
sockaddr_in));
          bzero(msg,sizeof(msg));
          recvid = recvfrom(sockid, msg, sizeof(msg), 0, (struct sockaddr*)&servaddr, &clientlen);
          printf("%s \n", msg);
          If (strncmp("bye",msg,3)==0)         {
              printf("Exit session...\n");
              break; }
 } // end of while loop
```

Recommended reading

- *UNIX Network Programming* by *W. Richard Stevens.*
- *Beginning Linux Programming, 4th Edition by*

  Neil Matthew, Richard Stones
- *http://mcalabprogram.blogspot.in/2012/01/tcp-sockets-chat-applicationserver.html*

- *http://man7.org/linux/manpages/man3/bzero.3.html*