```
/*                      THE CLIENT PROCESS

      Please read the file server.c before you read this file. To run this,
      you must first change the IP address specified in the line:

            serv_addr.sin_addr.s_addr = inet_addr("144.16.202.221");

      to the IP-address of the machine where you are running the server.
*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

main()
{
      int               sockfd ;
      struct sockaddr_in      serv_addr;

      int i;
      char buf[100];

      /* Opening a socket is exactly similar to the server process */
      if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
            printf("Unable to create socket\n");
            exit(0);
      }

      /* Recall that we specified INADDR_ANY when we specified the server
         address in the server. Since the client can run on a different
         machine, we must specify the IP address of the server.

         TO RUN THIS CLIENT, YOU MUST CHANGE THE IP ADDRESS SPECIFIED
         BELOW TO THE IP ADDRESS OF THE MACHINE WHERE YOU ARE RUNNING
         THE SERVER.

      */
      serv_addr.sin_family          = AF_INET;
      serv_addr.sin_addr.s_addr     = inet_addr("144.16.202.221");
      serv_addr.sin_port            = htons(6000);

      /* With the information specified in serv_addr, the connect()
         system call establishes a connection with the server process.
      */
      if ((connect(sockfd, (struct sockaddr *) &serv_addr,
                                 sizeof(serv_addr))) < 0) {
            printf("Unable to connect to server\n");
            exit(0);
      }

      /* After connection, the client can send or receive messages.
         However, please note that recv() will block when the
         server is not sending and vice versa. Similarly send() will
         block when the server is not receiving and vice versa. For
```

```
         non-blocking modes, refer to the online man pages.
      */
      for(i=0; i < 100; i++) buf[i] = '\0';
      recv(sockfd, buf, 100, 0);
      printf("%s\n", buf);


      strcpy(buf,"Message from client");
      send(sockfd, buf, strlen(buf) + 1, 0);

      close(sockfd);
}
```