```
/************* muliple client handle SERVER CODE ******************/

#include <stdio.h>
#include <stdlib.h>

#include <netdb.h>
#include <netinet/in.h>

#include <string.h>
#include <sys/socket.h>

void doprocessing (int sock);

int main( int argc, char *argv[] ) {
   int sockfd, newsockfd, portno, clilen;
   char buffer[256];
   struct sockaddr_in serv_addr, cli_addr;
   int n, pid;

   /* First call to socket() function */
   sockfd = socket(AF_INET, SOCK_STREAM, 0);

   if (sockfd < 0) {
      perror("ERROR opening socket");
      exit(1);
   }

   /* Initialize socket structure */
   bzero((char *) &serv_addr, sizeof(serv_addr));
   portno = 5001;

   serv_addr.sin_family = AF_INET;
   serv_addr.sin_addr.s_addr = INADDR_ANY;
   serv_addr.sin_port = htons(portno);

   /* Now bind the host address using bind() call.*/
   if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) {
      perror("ERROR on binding");
      exit(1);
   }

   /* Now start listening for the clients, here
      * process will go in sleep mode and will wait
      * for the incoming connection
   */

   listen(sockfd,5);
   clilen = sizeof(cli_addr);

   while (1) {
      newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);

      if (newsockfd < 0) {
```

```
         perror("ERROR on accept");
         exit(1);
      }

      /* Create child process */
      pid = fork();

      if (pid < 0) {
         perror("ERROR on fork");
         exit(1);
      }

      if (pid == 0) {
         /* This is the client process */
         close(sockfd);
         doprocessing(newsockfd);
         exit(0);
      }
      else {
         close(newsockfd);
      }

   } /* end of while */
}
void doprocessing (int sock) {
   int n;
   char buffer[256];
   bzero(buffer,256);
   n = read(sock,buffer,255);

   if (n < 0) {
      perror("ERROR reading from socket");
      exit(1);
   }

   printf("Here is the message: %s\n",buffer);
   n = write(sock,"I got your message",18);

   if (n < 0) {
      perror("ERROR writing to socket");
      exit(1);
   }

}


/************* muliple client handle client CODE ******************/

//2ed arg --> ip address
//3rd arg --> port address
// ./client1 127.10.20.30(any) 5001

#include <stdio.h>
```

```
#include <stdlib.h>

#include <netdb.h>
#include <netinet/in.h>

#include <string.h>

int main(int argc, char *argv[])
{

   int sockfd, portno, n;
   struct sockaddr_in serv_addr;
   struct hostent *server;

   char buffer[256];

   if (argc < 3)
   {
      fprintf(stderr,"usage %s hostname port\n", argv[0]);
      exit(0);
   }

   portno = atoi(argv[2]);

   /* Create a socket point */
   sockfd = socket(AF_INET, SOCK_STREAM, 0);

   if (sockfd < 0)
   {
      perror("ERROR opening socket");
      exit(1);
   }

   server = gethostbyname(argv[1]);

   if (server == NULL)
   {
      fprintf(stderr,"ERROR, no such host\n");
      exit(0);
   }

   bzero((char *) &serv_addr, sizeof(serv_addr));
   serv_addr.sin_family = AF_INET;

   bcopy((char *)server->h_addr, (char *)&serv_addr.sin_addr.s_addr, server->h_length);
   serv_addr.sin_port = htons(portno);

   /* Now connect to the server */
   if (connect(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0)
   {
      perror("ERROR connecting");
      exit(1);
   }
```

```
   /* Now ask for a message from the user, this message
      * will be read by server
   */

while(1)
{
   printf("Please enter the message: ");
   bzero(buffer,256);
   //take msg from user and put in buffer
   fgets(buffer,255,stdin);

   /* Send message to the server */
   n = write(sockfd, buffer, strlen(buffer));

   if (n < 0)
   {
      perror("ERROR writing to socket");
      exit(2);
   }

   /* Now read server response */
   bzero(buffer,256);
   n = read(sockfd, buffer, 255);

   if (n < 0)
   {
      perror("ERROR reading from socket");
      exit(1);
   }

   printf("%s\n",buffer);

}
    return 0;
}
```