Date

Expt. No. Nidhi K (1KV18CS104) Page No.

Lab internal

Q5 Implement concurrent chat server that allows current logged in users to chat with server

/* Sender */

```
#include<sys/types.h>
#include<sys/socket.h>
#include<sys/stat.h>
#include<unistd.h>
#include<stdlib.h>
#include<fcntl.h>
#include<stdio.h>
#include<arpa/inet.h>
#include<string.h>
#define max 255
void str_echo(int connfd)
{
  int n; char *buf = malloc(max);
  while(n=recv(connfd, buf, max, 0))>0)
  { fputs("From client: ", stdout);
    fputs(buf, stdout);
    fputs("\n", stdout);
    if(fgets(buf, max, stdin)!=NULL){
      send(connfd, buf, strlen(buf), 0);}
    bzero(buf, max); }}
int main()
{ int cont, listfd, connfd, addrlen, addrlen2, fd, pid, addrlen3;
  struct sockaddr_in addr, cli_addr;
  if((listfd=socket(AF_INET, SOCK_STREAM, 0))>0)
```

74/80
27.75/30

45/50

Teacher's Signature

Expt. No. Page No.

```
Printf("Socket created successfully");
else {printf("socket creation failed"); exit(0);}
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = INADDR_ANY;
addr.sin_port = htons(16001);
printf("The address before bind %s...\n", inet_ntoa
                                    (addr.sin_addr));
if (bind(listfd, (struct sockaddr *)&addr, sizeof(addr))==0)
printf("Binding socket\n");
else { printf("Bind failed"); exit(0);}
printf("The address after bind %s...\n", inet_ntoa
                                    (addr.sin_addr));
listen(listfd, 3);
printf("Server is listening\n");
getsockname(listfd, (struct sockaddr *)&addr, &addrlen3);
printf("The server's local address is %s, and port is %d\n",
                inet_ntoa(addr.sin_addr), htons(addr.sin_port));
for(;;)
addrlen = sizeof(struct sockaddr_in);
connfd = accept(listfd, (struct sockaddr *)&cli_addr,
                                    &addrlen);
addrlen2 = sizeof(struct sockaddr_in);
int i = getpeername(connfd, (struct sockaddr *)&cli_addr,
                                    &addrlen2);
printf("The client %s is connected on port %d\n",
inet_ntoa(cli_addr), htons(cli_addr.sin_port));
if(pid = fork())==0) { printf("Inside child\n");
    close(listfd); str_echo(connfd); exit(0);}
close(connfd); } return 0;}
```

Teacher's Signature

Date _____

Expt. No. _____ Page No. _____

```
/* client */
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<string.h>
#include<unistd.h>
#include<netinet/in.h>
#define max 25
void str_cli(FILE *fp, int sockfd)
{ int cont, char *buf=malloc(max);
  fputs("To server: ", stdout);
  while(fgets(buf, max, fp)!=NULL)
  {send(sockfd, buf, strlen(buf), 0);
  if((cont=recv(sockfd, buf, max, 0))>0){
  fputs("From server: "; stdout)); fputs(buf, stdout);
  if(strncmp("exit", 4)==0){
  printf("Client exit..\n"); break;}
  bzero(buf, 1024);
  fputs("\n To server:", stdout);}
  printf("EOF\n");}
int main(int argc, char *argv[])
{ int cr_sock, struct sockaddr_in addr;
  if(cr_sock=socket(AF_INET, SOCK_STREAM, 0)>0)
    printf("Socket created successfully");
else{printf("Socket creation failed"); exit(0);}
  addr.sin_port = htons(16001);
  addr.sin_family=AF_INET;
  inet_pton(AF_INET, argv[1], &addr.sin_addr);
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

```
if(connect(cr_sock, (struct sockaddr *)&addr, sizeof(addr))<0)
  printf("The connection is accepted by the server");
else printf("Error in connect");
strcpy(stdin, cr_sock);
return close(cr_sock);
}
```