**Program 6 : Implementation of concurrent and iterative echo server using both connection and connectionless socket system calls  
echo\_sever\_tcpcon.c**

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <sys/stat.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <fcntl.h>

#include <arpa/inet.h>

void str\_echo(int connfd)

{

int n;

int bufsize = 1024;

char \*buffer = malloc(bufsize);

again:

while((n = recv(connfd, buffer, bufsize, 0)) > 0)

send(connfd,buffer,n,0);

if(n < 0)

goto again;

free(buffer);

}

int main()

{

int listenfd, connfd, addrlen, pid, addrlen3;

struct sockaddr\_in address, cli\_address;

if ((listenfd = socket(AF\_INET, SOCK\_STREAM, 0)) > 0)

printf("The socket was created\n");

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(15001);

printf("The address before bind %s ...\n", inet\_ntoa(address.sin\_addr));

if (bind(listenfd, (struct sockaddr \*)&address, sizeof(address)) == 0)

printf("Binding Socket\n");

printf("The address after bind %s ...\n",inet\_ntoa(address.sin\_addr));

listen(listenfd, 3);

printf("Server is listening\n");

getsockname(listenfd, (struct sockaddr \*)&address, &addrlen3);

printf("The server's local address %s ... and port %d\n", inet\_ntoa(address.sin\_addr), htons(address.sin\_port));

for(;;)

{

addrlen = sizeof(struct sockaddr\_in);

connfd = accept(listenfd, (struct sockaddr \*)&cli\_address, &addrlen);

int i = getpeername(connfd,(struct sockaddr \*)&cli\_address,&addrlen);

if (connfd > 0)

printf("The Client %s is connected ... on port %d\n", inet\_ntoa(cli\_address.sin\_addr), htons(cli\_address.sin\_port));

if ((pid = fork()) == 0)

{

printf("inside child\n");

close(listenfd);

str\_echo(connfd);

exit(0);

}

close(connfd);

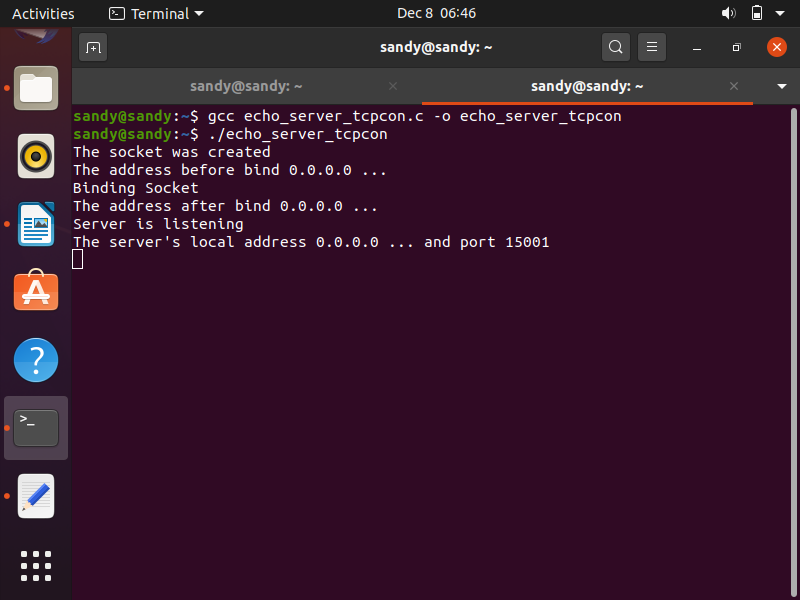
}

return 0 ;

}

**output:**

**after running server program :**



echo\_client\_tcpcon.c

// echo\_client\_tcpcon.c

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <arpa/inet.h>

void str\_cli(FILE \*fp, int sockfd)

{

int bufsize = 1024;

char \*buffer = malloc(bufsize);

while (fgets(buffer, bufsize, fp) != NULL)

{

send(sockfd, buffer, sizeof(buffer), 0);

if (recv(sockfd, buffer, bufsize, 0) > 0)

fputs(buffer, stdout);

}

printf("\nEOF\n");

free(buffer);

}

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

{

int create\_socket;

struct sockaddr\_in address;

if ((create\_socket = socket(AF\_INET,SOCK\_STREAM,0)) > 0)

printf("The socket was created\n");

address.sin\_family = AF\_INET;

address.sin\_port = htons(15001);

inet\_pton(AF\_INET, argv[1], &address.sin\_addr);

if (connect(create\_socket, (struct sockaddr \*)&address, sizeof(address)) == 0)

printf("The connection was accepted with the server %s...\n",argv[1]);

else

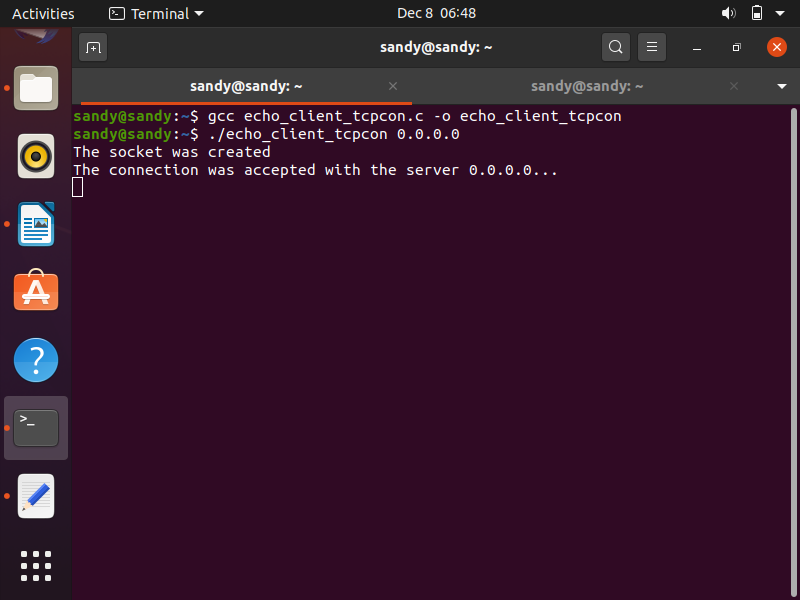
printf("Error in connect\n");

str\_cli(stdin, create\_socket);

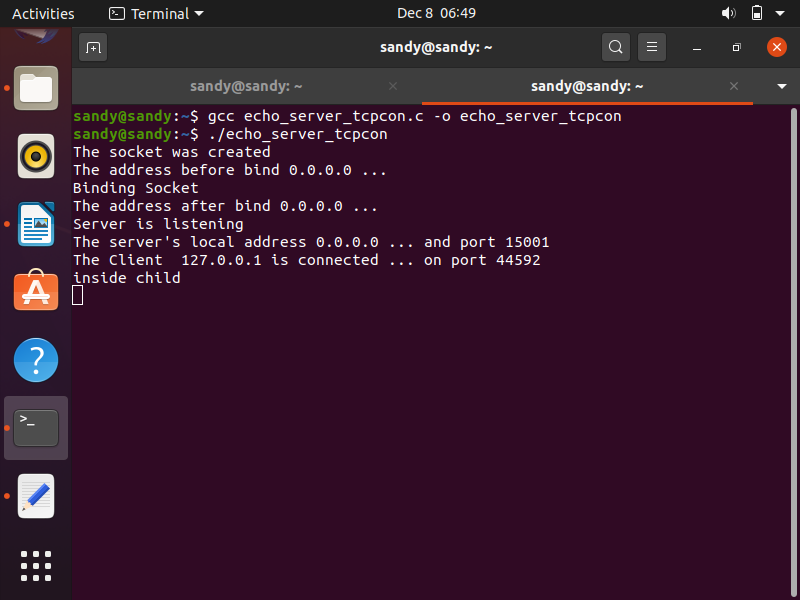
return close(create\_socket);

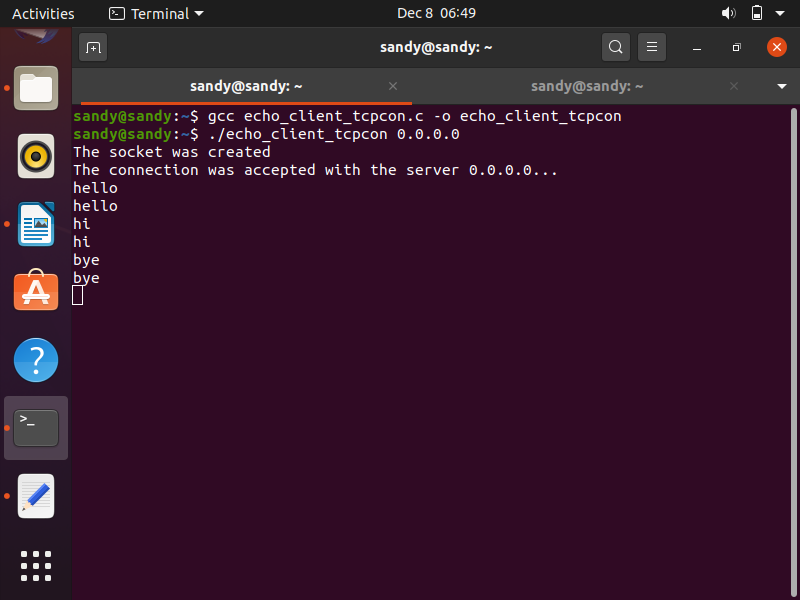
}

OUTPUT:



the server tab after running client program :



client output :

it\_echo\_server.c

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<sys/stat.h>

#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

#include<fcntl.h>

#include<arpa/inet.h>

void str\_echo(int connfd)

{

int n;

int bufsize = 1024;

char \*buffer = malloc(bufsize);

//printf("inside the function");

again: while((n=recv(connfd, buffer, bufsize, 0))>0)

send(connfd,buffer,n,0);

if(n<0)

goto again;

}

int main()

{

int cont,listenfd,connfd,addrlen,addrlen2,fd,pid,addrlen3;

//char fname[256];

struct sockaddr\_in address,cli\_address;

if ((listenfd = socket(AF\_INET,SOCK\_STREAM,0)) > 0)

printf("The socket was created\n");

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(15001);

printf("The address before bind %s ...\n",inet\_ntoa(address.sin\_addr) );

if (bind(listenfd,(struct sockaddr \*)&address,sizeof(address)) == 0)

printf("Binding Socket\n");

printf("The address after bind %s ...\n",inet\_ntoa(address.sin\_addr) );

listen(listenfd,3);

printf("server is listening\n");

for(;;){

addrlen = sizeof(struct sockaddr\_in);

connfd = accept(listenfd,(struct sockaddr \*)&cli\_address,&addrlen);

/\*if(connfd>0)

printf("The client is connected\n");\*/

printf("The Client %s is Connected...on port %d\n",inet\_ntoa(cli\_address.sin\_addr),htons(cli\_address.sin\_port));

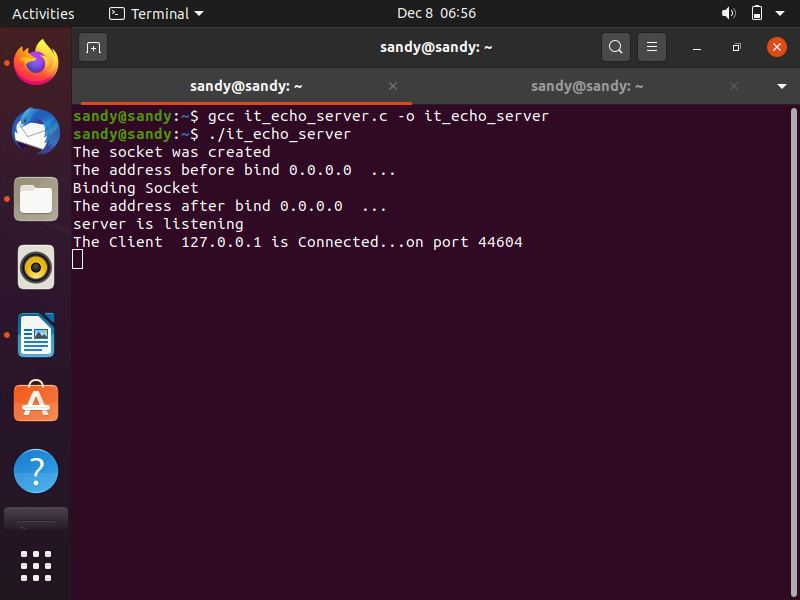
str\_echo(connfd);

close(connfd);

}

return 0 ;

}



it\_echo\_client.c

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<sys/stat.h>

#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

#include<fcntl.h>

#include<arpa/inet.h>

void str\_cli(FILE \*fp, int sockfd)

{

int bufsize = 1024, cont;

char \*buffer = malloc(bufsize);

while(fgets(buffer,bufsize,fp)!=NULL)

{

send(sockfd, buffer, sizeof(buffer), 0);

if((cont=recv(sockfd, buffer, bufsize, 0))>0) {

fputs(buffer,stdout);

}

}

printf("\nEOF\n");

}

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

{

int create\_socket;

struct sockaddr\_in address;

if ((create\_socket = socket(AF\_INET,SOCK\_STREAM,0)) > 0)

printf("The Socket was created\n");

address.sin\_family = AF\_INET;

address.sin\_port = htons(15001);

inet\_pton(AF\_INET,argv[1],&address.sin\_addr);

if (connect(create\_socket,(struct sockaddr \*) &address, sizeof(address)) == 0)

printf("The client is connecting to the server %s...\n",argv[1]);

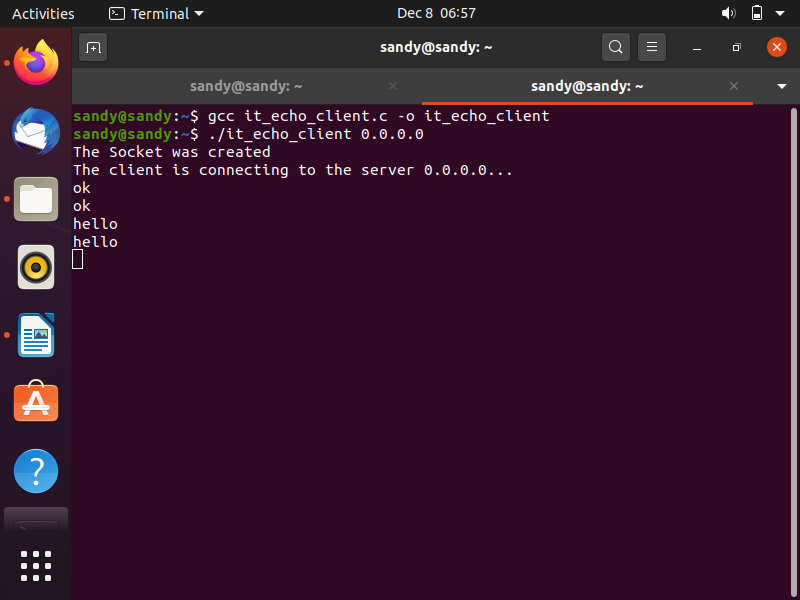
else

printf("error in connect \n");

str\_cli(stdin,create\_socket);

return close(create\_socket);

}



**udp\_echo\_server.c**

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<sys/stat.h>

#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

#include<fcntl.h>

#include <arpa/inet.h>

void str\_echo(int sockfd,struct sockaddr\* cli\_address, int clilen)

{

int n;

int bufsize = 1024;

char \*buffer = malloc(bufsize);

int addrlen;

for(;;){

addrlen = clilen;

n=recvfrom(sockfd,buffer,bufsize,0,cli\_address,&addrlen); //recvfrom

//printf("%s",buffer);

sendto(sockfd,buffer,n,0,cli\_address,addrlen);} //sendto

//printf("%d n",n);

}

int main()

{

int sockfd;

struct sockaddr\_in serv\_address,cli\_address;

if ((sockfd = socket(AF\_INET,SOCK\_DGRAM,0)) > 0) //sockfd

printf("The socket was created\n");

serv\_address.sin\_family = AF\_INET;

serv\_address.sin\_addr.s\_addr = INADDR\_ANY;

serv\_address.sin\_port = htons(16001);

printf("The address before bind %s ...\n",inet\_ntoa(serv\_address.sin\_addr) );

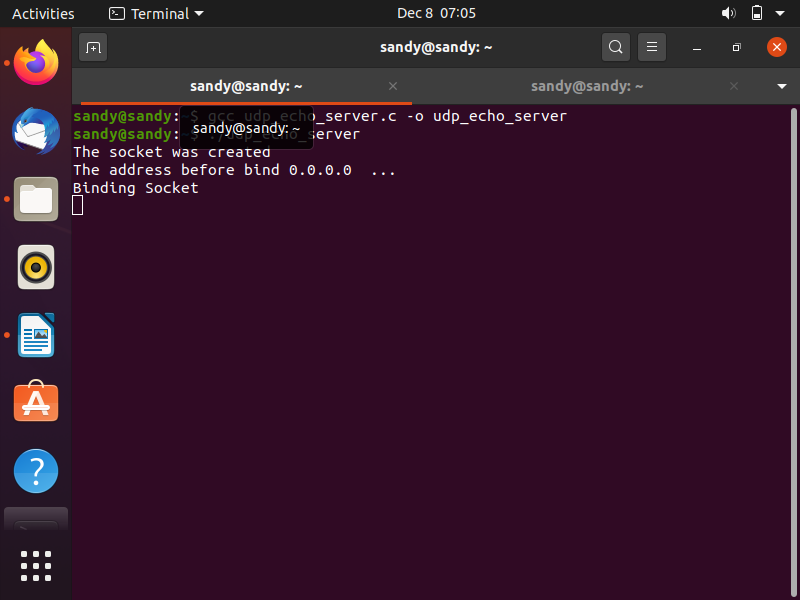
if (bind(sockfd,(struct sockaddr \*)&serv\_address,sizeof(serv\_address)) == 0) //bind

printf("Binding Socket\n");

str\_echo(sockfd,(struct sockaddr \*)&cli\_address,sizeof(cli\_address));

return 0 ;

}



udp\_echo\_client.c

/\*Client\*/

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

void str\_cli(FILE \*fp, int sockfd,struct sockaddr\* serv\_address, int servlen)

{

int bufsize = 1024, cont;

char \*buffer = malloc(bufsize);

int addrlen = sizeof(struct sockaddr\_in);

while(fgets(buffer,bufsize,fp)!=NULL){

sendto(sockfd, buffer, sizeof(buffer),0,serv\_address,servlen);

if((cont=recvfrom(sockfd, buffer, bufsize, 0,NULL,NULL)>0))

{

fputs(buffer,stdout); //echo printing

}}

printf("\nEOF\n");

}

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

{

int sockfd;

//char fname[256];

struct sockaddr\_in serv\_address;

if ((sockfd = socket(AF\_INET,SOCK\_DGRAM,0)) > 0)

printf("The Socket was created\n");

serv\_address.sin\_family = AF\_INET;

serv\_address.sin\_port = htons(16001);

inet\_pton(AF\_INET,argv[1],&serv\_address.sin\_addr);

str\_cli(stdin,sockfd,(struct sockaddr \*)&serv\_address,sizeof(serv\_address));

exit(0);

}

