|  |
| --- |
| Software Lab Based on CSHT-616 |
| UNIX NETWORK PROGRAMMING |
| SEMESTER -6 PRACTICAL FILE |

|  |
| --- |
| ***Submitted By :*** |
| **DEEPTI SHARMA** |
| **2014315 (14066270012)** |
| **SRCASW** |
| **3RD YEAR 6TH SEMESTER** |

TCP

**PRACTICAL 1: *Implement TCP Echo Client and Echo Server (Iterative).***

***Program:***

1. Client:

*#include "unp.h"*

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

{

char sendline[MAXLINE], recvline[MAXLINE];

while(fgets(sendline, MAXLINE, fp)!=NULL)

{

write(sockfd, sendline, strlen(sendline));

if(read(sockfd, recvline, MAXLINE)==0)

printf("str\_cli: server terminated prematurely");

fputs(recvline,stdout);

}

}

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

{

int sockfd;

struct sockaddr\_in servaddr;

if(argc!=2)

printf("usage: topcli <IPAddress>");

sockfd=socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(2002);

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

connect(sockfd, (SA\*) &servaddr, sizeof(servaddr));

//\*\*\*CALLING ECHO SERVER\*\*\*//

str\_cli(stdin, sockfd);

exit(0);

}

1. Server:

*#include "unp.h"*

void str\_echo(int sockfd)

{

ssize\_t n;

char buf[MAXLINE];

while((n=read(sockfd,buf,MAXLINE))>0)

write(sockfd,buf,n);

close(sockfd);

}

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

{

int listenfd, connfd;

pid\_t childpid;

struct sockaddr\_in servaddr;

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

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(2002);

bind(listenfd, (SA\*) &servaddr, sizeof(servaddr));

listen(listenfd, LISTENQ);

for(;;)

{

//clilen=sizeof(cliaddr);

connfd=accept(listenfd,(SA\*)NULL ,NULL);

str\_echo(connfd);

}

}

***Output:***

[u6302@linux6 eserv\_i]$ ./c1 127.0.0.1

Hi I am DEEPTI

Hi I am DEEPTI

wHO R U?

wHO R U?

**PRACTICAL 2: *Implement TCP Echo Client and Echo Server (Concurrent).***

***Program:***

1. Client:

*#include "unp.h"*

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

{

char sendline[MAXLINE], recvline[MAXLINE];

while(fgets(sendline, MAXLINE, fp)!=NULL)

{

write(sockfd, sendline, strlen(sendline));

if(read(sockfd, recvline, MAXLINE)==0)

printf("str\_cli: server terminated prematurely");

fputs(recvline,stdout);

}

}

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

{

int sockfd;

struct sockaddr\_in servaddr;

if(argc!=2)

printf("usage: topcli <IPAddress>");

sockfd=socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(2002);

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

connect(sockfd, (SA\*) &servaddr, sizeof(servaddr));

//\*\*\*CALLING ECHO SERVER\*\*\*//

str\_cli(stdin, sockfd);

exit(0);

}

1. Server:

*#include "unp.h"*

void str\_echo(int sockfd)

{

ssize\_t n;

char buf[MAXLINE];

while((n=read(sockfd,buf,MAXLINE))>0)

write(sockfd,buf,n);

}

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

{

int listenfd, connfd;

pid\_t childpid;

socklen\_t clilen;

struct sockaddr\_in cliaddr,servaddr;

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

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(2002);

bind(listenfd, (SA\*) &servaddr, sizeof(servaddr));

listen(listenfd, LISTENQ);

for(;;)

{

clilen=sizeof(cliaddr);

connfd=accept(listenfd,(SA\*)&cliaddr,&clilen);

if((childpid=fork())==0)

{

close(listenfd);

str\_echo(connfd);

exit(0);

}

}

close(connfd);

}

***Output:***

[u6302@linux6 eserv]$ ./c1 127.0.0.1

hello

hello

pc 1

pc 1

[u6302@linux6 eserv]$ ./c1 127.0.0.1

hello

hello

pc 2

pc 2

why talk?

why talk?

**PRACTICAL 3: *Implement TCP Daytime Client and Daytime Server (Iterative).***

***Program:***

1. Client:

*#include "unp.h"*

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

{

int sockfd,n;

char recvline[MAXLINE+1];

struct sockaddr\_in servaddr;

if(argc != 2)

printf("usage: a.out <IPaddress>");

if( (sockfd = socket(AF\_INET, SOCK\_STREAM, 0)) < 0)

printf("socket error");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(1315);

if(inet\_pton(AF\_INET, argv[1], &servaddr.sin\_addr) <= 0)

printf("inet\_pton error for %s", argv[1]);

if(connect(sockfd, (SA \*) &servaddr, sizeof(servaddr)) < 0)

printf("connect error");

while( (n = read(sockfd, recvline, MAXLINE)) > 0)

{

recvline[n] = 0;

if(fputs(recvline, stdout) == EOF)

printf("fputs error");

}

if(n < 0)

printf("read error");

exit(0);

}

1. Server:

*#include "unp.h"*

*#include <time.h>*

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

{

int listenfd, connfd;

char buff[MAXLINE];

time\_t ticks;

struct sockaddr\_in servaddr;

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

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(13);

bind(listenfd, (SA \*) &servaddr, sizeof(servaddr));

listen(listenfd, LISTENQ);

for( ; ;)

{

connfd = accept(listenfd, (SA\*) NULL, NULL);

ticks = time(NULL);

snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));

write(connfd, buff, strlen(buff));

close(connfd);

}

exit(0);

}

***Output:***

[u6302@linux6 dserv\_i]$ ./s1&

[1] 4812

[u6302@linux6 dserv\_i]$ ./c1 127.0.0.1

Mon Apr 3 16:52:50 2017

**PRACTICAL 4: *Implement TCP Daytime Client and Daytime Server (Concurrent).***

***Program:***

1. Client:

*#include "unp.h"*

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

{

int sockfd;

char recvline[MAXLINE+1];

struct sockaddr\_in servaddr;

if(argc!=2)

printf("usage: topcli <IPAddress>");

sockfd=socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(2002);

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

connect(sockfd, (SA\*) &servaddr, sizeof(servaddr));

printf("hELOO CONNECTED");

while( (n = read(sockfd, recvline, MAXLINE)) > 0)

{

recvline[n] = 0;

if(fputs(recvline, stdout) == EOF)

printf("fputs error");

}

if(n < 0)

printf("read error");

exit(0);

}

1. Server:

*#include "unp.h"*

*#include <time.h>*

void dt(int sockfd)

{

time\_t ticks;

char buff[MAXLINE];

ticks = time(NULL);

snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));

write(sockfd, buff, strlen(buff));

}

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

{

int listenfd, connfd;

pid\_t childpid;

socklen\_t clilen;

struct sockaddr\_in cliaddr,servaddr;

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

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(2002);

bind(listenfd, (SA\*) &servaddr, sizeof(servaddr));

listen(listenfd, LISTENQ);

for(;;)

{

clilen=sizeof(cliaddr);

connfd=accept(listenfd,(SA\*)&cliaddr,&clilen);

if((childpid=fork())==0)

{

close(listenfd);

dt(connfd);

exit(0);

}

close(connfd);

}

}

***Output:***

[u6302@linux6 dserv\_c]$ ./s1&

[2] 4832

[u6302@linux6 dserv\_c]$ ./c1 127.0.0.1

HereMon Apr 3 16:54:00 2017

[u6302@linux6 dserv\_c]$ ./s1&

(wd now: ~/deepti/dserv\_c)

[u6308@linux6 dserv\_c]$ ./c1 127.0.0.1

HereMon Apr 3 16:54:00 2017

UDP

**PRACTICAL 5: *Implement UDP Echo Client and Echo Server.***

***Program:***

1. Client:

*#include "unp.h"*

void dg\_cli(FILE \*fp, int sockfd, const SA\* pservaddr, socklen\_t servlen)

{

int n;

char sendline[MAXLINE], recvline[MAXLINE+1];

while(fgets(sendline,MAXLINE,fp)!=NULL)

{

sendto(sockfd,sendline, strlen(sendline), 0, pservaddr, servlen);

n=recvfrom(sockfd,recvline,MAXLINE,0,NULL,NULL);

recvline[n]=0;

fputs(recvline,stdout);

}

}

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

{

int sockfd;

struct sockaddr\_in servaddr;

if(argc!=2)

{

printf("usage: ./c1 <IP ADDRESS>");

}

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(1315);

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

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

dg\_cli(stdin, sockfd,(SA\*) &servaddr,sizeof(servaddr));

exit(0);

}

1. Server:

*#include "unp.h*"

void do\_echo(int sockfd,SA \*pcliaddr, socklen\_t clilen)

{

int n;

socklen\_t len;

char mesg[MAXLINE];

for(;;)

{

len=clilen;

n=recvfrom(sockfd,mesg,MAXLINE,0,pcliaddr,&len);

sendto(sockfd,mesg,n,0,pcliaddr,len);

}

}

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

{

int sockfd;

struct sockaddr\_in servaddr, cliaddr;

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

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(1315);

bind(sockfd,(SA\*)&servaddr,sizeof(servaddr));

do\_echo(sockfd,(SA\*)&cliaddr,sizeof(cliaddr));

}

***Output:***

[u6302@linux6 echo]$ ./s1&

[1] 4960

[u6302@linux6 echo]$ ./c1 127.0.0.1

HI

HI

How r u >^H?

How r u ?

Ho come u r here?

Ho come u r here?

**PRACTICAL 6: *Implement UDP Daytime Client and Daytime Server .***

***Program:***

1. Client:

*#include "unp.h"*

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

{

int sockfd,n;

char recvline[MAXLINE+1],sendline[MAXLINE];

struct sockaddr\_in servaddr;

if(argc!=2)

{

printf("usage: ./c1 <IP ADDRESS>");

}

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(2014);

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

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

socklen\_t len;

len=sizeof(servaddr);

sendto(sockfd,"hi",2,0,(SA\*)&servaddr,len);

while((n = recvfrom(sockfd, recvline, MAXLINE,0,(SA\*)&servaddr,&len))>0)

{

recvline[n] = 0;

if(fputs(recvline, stdout) == EOF)

printf("fputs error");

close(sockfd);

}

if(n < 0)

printf("read error");

exit(0);

}

1. Server:

*#include "unp.h"*

*#include<time.h>*

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

{

int sockfd,n;

struct sockaddr\_in servaddr,cliaddr;

char buff[MAXLINE];

time\_t ticks;

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

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(2014);

bind(sockfd,(SA\*)&servaddr,sizeof(servaddr));

socklen\_t len;

len=sizeof(cliaddr);

for(;;)

{

ticks = time(NULL);

n=recvfrom(sockfd,buff,MAXLINE,0,(SA\*)&cliaddr,&len);

snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));

sendto(sockfd,buff, sizeof(buff),0,(SA\*)&cliaddr,sizeof(cliaddr));

close(sockfd);

}

exit(0);

}

***Output:***

[u6302@linux6 daytime]$ ./s1&

[1] 4980

[u6302@linux6 daytime]$ ./c1 127.0.0.1

Mon Apr 3 16:59:45 2017

HOST INFORMATION

**PRACTICAL 7: *Write a program to display the name of the Host.***

***Program:***

*#include "unp.h"*

*#include<sys/utsname.h>*

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

{

struct utsname myname;

if(uname(&myname)<0)

printf("uname error");

printf("Name of the host: %s ",myname.nodename);

printf("\n");

exit(0);

}

***Output:***

[u6302@linux6 ~]$ gcc -o q7 q7.c

[u6302@linux6 ~]$ ./q7 127.0.0.1

Name of the host: linux6.com

**PRACTICAL 8: *Write a program to display all IP Addresses of the Host.***

***Program:***

*#include "unp.h"*

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

{

char \*ptr,\*\*pptr;

char str[INET\_ADDRSTRLEN];

struct hostent \*hptr;

while(--argc > 0)

{

ptr= \*++argv;

if((hptr=gethostbyname(ptr)) == NULL)

{

printf("gethostbyname error for host:%s",ptr);

continue;

}

printf("Official hostname: %s \n ", hptr ->h\_name);

for(pptr=hptr ->h\_aliases; \*pptr != NULL; pptr++)

printf("\tAliases:%s\n", \*pptr);

switch(hptr ->h\_addrtype)

{

case AF\_INET:

pptr=hptr->h\_addr\_list;

for(;\*pptr!=NULL;pptr++)

printf("\taddress:%s\n",inet\_ntop(hptr->h\_addrtype,\*pptr,str,sizeof(str)));

break;

default:

printf("Unknowm address type");

break;

}

}

exit(0);

}

***Output:***

[u6302@linux6 ipaddr]$ ./p1 127.0.0.1

Official hostname: 127.0.0.1

address:127.0.0.1

SOURCES

***UNP.H***

#include<sys/types.h>

#include<sys/socket.h>

#include<sys/time.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<sys/un.h>

#define bzero(ptr,n) memset(ptr,0,n)

#define LISTENQ 1024

#define MAXLINE 4096

#define BUFFSIZE 8192

#define SA struct sockaddr