**Ex: 5**

**Regno :185001161**

**DAYTIME SERVER USING UDP**

Write a UDP socket program to implement daytime server. Use menu driven concept to get Day, Date, Time etc. from the client. Consider multiple client requests for the day time server

**Server Code:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<netdb.h>

#include<stdlib.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<time.h>

#define PORT 7228

struct input

{

int option;

int id;

};

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

{

int sockfd, ret,len,n,fromlen;

struct sockaddr\_in serverAddr;

int newfd;

struct sockaddr\_in clientAddr;

struct tm\* local;

//socklen\_t addr\_size;

char buffer[1024];

char str[1000];

pid\_t child;

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

if(sockfd < 0){

printf("---------Error in connection.\n");

exit(1);

}

//printf("---------Server Socket is created.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(PORT);

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

ret = bind(sockfd, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));

fromlen = sizeof(struct sockaddr\_in);

len = sizeof(clientAddr);

int no\_clients=1;

//time\_t rawtime;

//time(&rawtime); // 0 or NULL also works .....

// printf(" Time = ",asctime(local));

struct input i;

int x;

while(1)

{

n = recvfrom(sockfd,&i,sizeof(i),0,(struct sockaddr \*)&clientAddr,&len);

printf("\nRequest from Client %d\n",i.id);

time\_t rawtime;

time(&rawtime);

local = localtime(&rawtime);

x = i.option;

if (x == 1)

{

strftime(buffer,sizeof(buffer)," %x .",local);

}

else if(x == 2)

{

strftime(buffer,sizeof(buffer)," %A .",local);

}

else if(x == 3)

{

strftime(buffer,sizeof(buffer)," %B .",local);

}

else if(x == 4)

{

strftime(buffer,sizeof(buffer)," %Y .",local);

}

else if(x == 5)

{

strftime(buffer,sizeof(buffer)," %I:%M %p .",local);

}

else if(x == 6)

{

struct tm \*t;

time\_t toronto;

t = gmtime(&rawtime);

t -> tm\_hour -=4;

toronto = mktime(t);

t = localtime(&toronto);

strftime(buffer,sizeof(buffer)," %c .",local);

}

printf("\nMessage sent to client %d\n",i.id);

sendto(sockfd,buffer,sizeof(buffer),0,(struct sockaddr\*)&clientAddr,len);

}

return 0;

}

**Client Code:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<netdb.h>

#include<stdlib.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<time.h>

#define PORT 7228

struct input

{

int option;

int id;

};

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

{

/\* code \*/

int sockfd , ret,n,len;

struct sockaddr\_in serverAddr;

char buffer[1024];

char str[1000];

struct input i;

printf("\nEnter the client id :");

scanf("%d",&i.id);

struct tm\* local;

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

if(sockfd < 0)

{

printf("-----Error in Connection-----.\n");

exit(1);

}

//printf("-----Client Socket is Created!.\n");

bzero(&serverAddr,sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

serverAddr.sin\_port = htons(7228);

len = sizeof(serverAddr);

printf("\nRequesting to the server!");

//time\_t rawtime;

//time(&rawtime);

char ans;

int x;

while(1)

{

printf("\n1.Date\n2.Day\n3.Month\n4.Year\n5.Time\n6.Toronto\nEnter your option:");

scanf("%d",&i.option);

x = i.option;

if (x < 1|| x > 6)

continue;

sendto(sockfd,&i,sizeof(i),0,(struct sockaddr \*)&serverAddr,len);

recvfrom(sockfd,buffer,sizeof(buffer),0,(struct sockaddr \*)&serverAddr,&len);

if (x==1)

printf("\nThe Date is %s\n",buffer);

else if (x==2)

printf("\nThe Day is %s\n",buffer);

else if (x==3)

printf("\nThe Month is %s\n",buffer);

else if (x==4)

printf("\nThe year is %s\n",buffer);

else if (x==5)

printf("\nThe Time is %s\n",buffer);

else if (x==6)

{

time\_t rawtime;

time(&rawtime);

local = localtime(&rawtime);

strftime(str,sizeof(str),"%c ",local);

printf("\nThe Local time is %s\n",str);

printf("\nThe Toronto time is %s\n",buffer);

}

printf("\nDo you want to continue(1/0):");

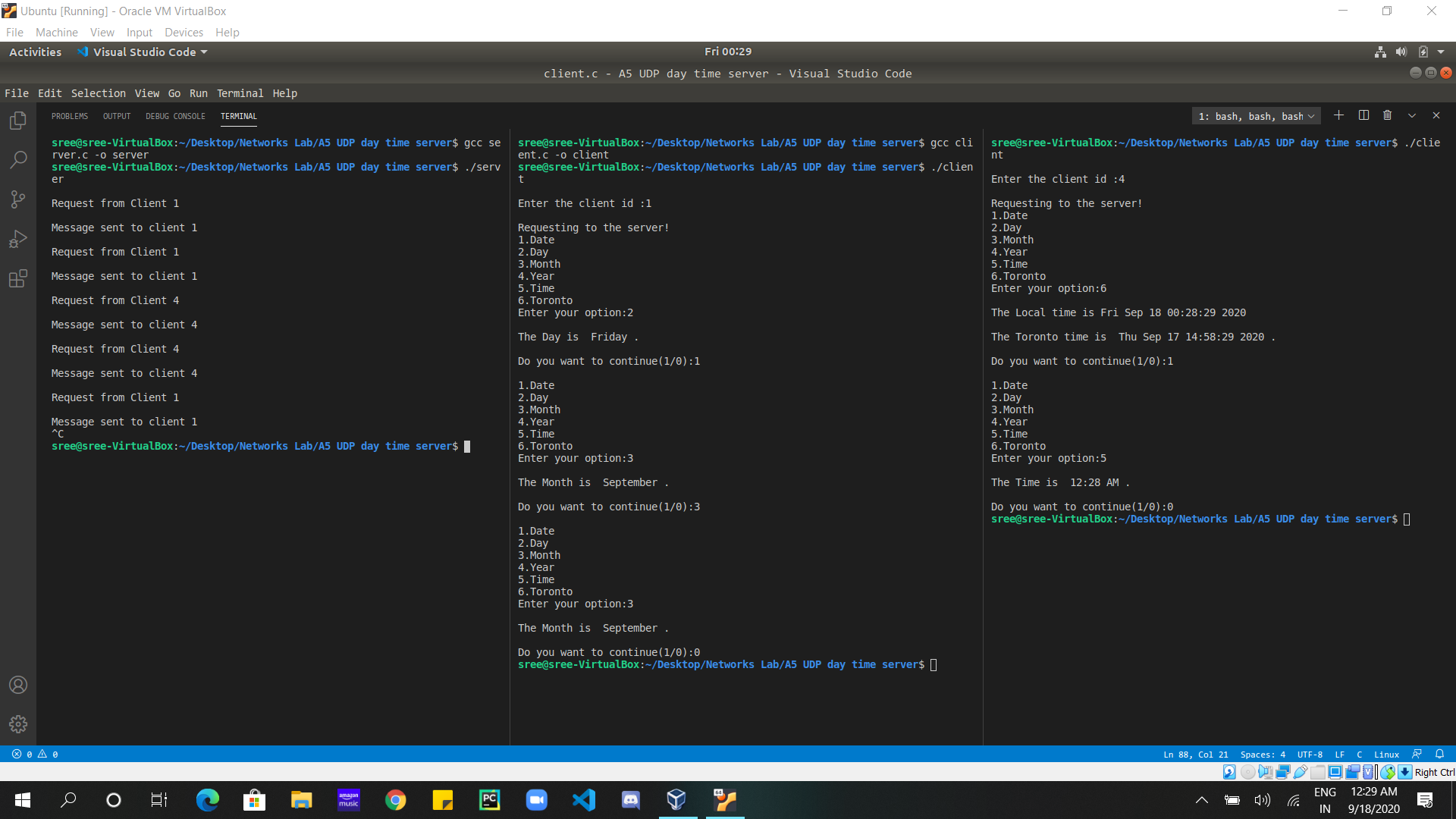
scanf("%d",&x);

if (x == 0 )

break;

}

}

****