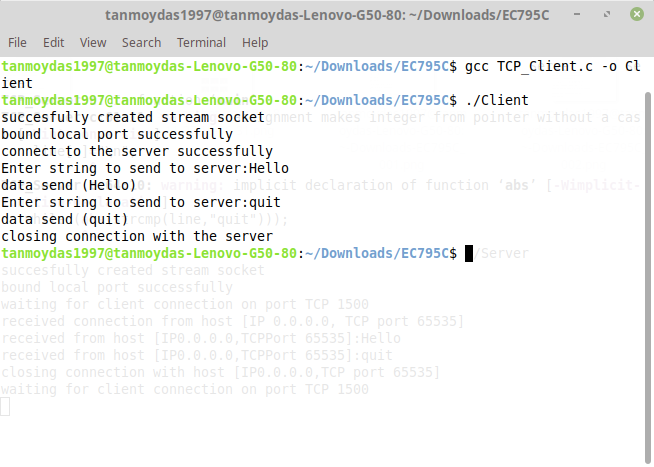
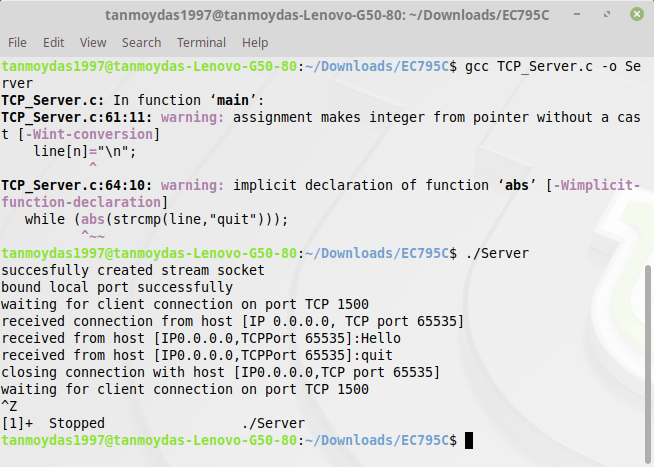
Output:

TCP Client



TCP Server

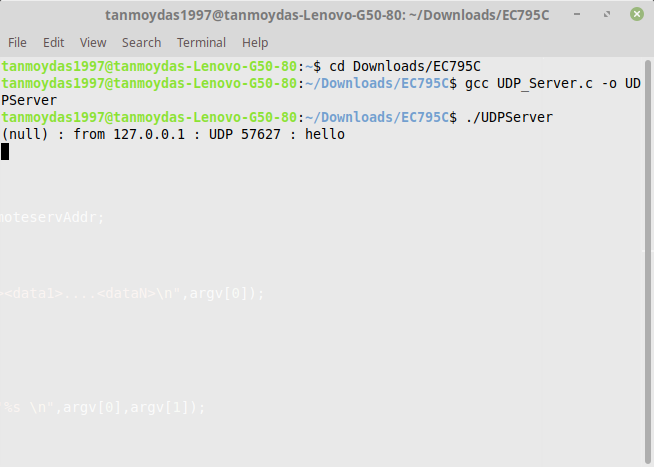


Output:

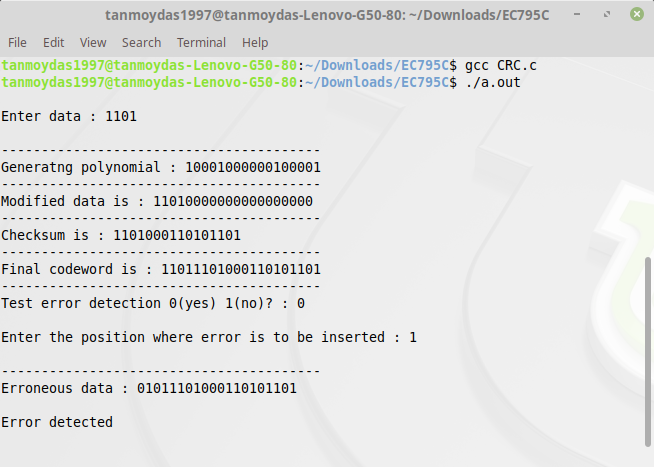
UDP Client



UDP Server



Output:



Program:

TCP Client

**#include** <sys/types.h>

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

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

**#include** <arpa/inet.h>

**#include** <netdb.h>

**#include** <unistd.h>

**#include** <math.h>

**#include** <string.h>

**#include** <stdio.h>

**#define** MAX\_MSG 100

**#define** SERVER\_ADDR "127.0.0.1"

**#define** CLIENT\_ADDR "127.0.0.1"

**#define** SERVER\_PORT 1500

**#define** CLIENT\_PORT 1500

int **main**()

{

    int sd, rc, i;

    struct sockaddr\_in clientAddr, servAddr;

    char line[MAX\_MSG];

**bzero**((char \*)&servAddr, sizeof(servAddr));

    servAddr**.**sin\_family = AF\_INET;

    servAddr**.**sin\_addr**.**s\_addr = **inet\_addr**(SERVER\_ADDR);

    servAddr**.**sin\_port = **htons**(SERVER\_PORT);

**bzero**((char \*)&clientAddr, sizeof(clientAddr));

    clientAddr**.**sin\_family = AF\_INET;

    clientAddr**.**sin\_addr**.**s\_addr = INADDR\_ANY;

    clientAddr**.**sin\_port = **htons**(0);

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

    printf("succesfully created stream socket \n");

**bind**(sd, (struct sockaddr \*)&clientAddr, sizeof(clientAddr));

    printf("bound local port successfully \n");

**connect**(sd, (struct sockaddr \*)&servAddr, sizeof(servAddr));

    printf("connect to the server successfully\n");

**do**

    {

        printf("Enter string to send to server:");

        scanf("%s", line);

**send**(sd, line, strlen(line) + 1, 0);

        printf("data send (%s)\n", line);

    } **while** (strcmp(line, "quit"));

    printf("closing connection with the server \n");

**close**(sd);

}

TCP Server

**#include** <sys/types.h>

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

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

**#include** <arpa/inet.h>

**#include** <netdb.h>

**#include** <unistd.h>

**#include** <math.h>

**#include** <string.h>

**#include** <stdio.h>

**#define** MAX\_MSG 100

**#define** SERVER\_ADDR "127.0.0.1"

**#define** SERVER\_PORT 1500

int **main**()

{

    int sd, newsd, clilen, n;

    struct sockaddr\_in cliAddr, servAddr;

    char line[MAX\_MSG];

**bzero**((char \*)&servAddr, sizeof(servAddr));

    servAddr**.**sin\_family = AF\_INET;

    servAddr**.**sin\_addr**.**s\_addr = **inet\_addr**(SERVER\_ADDR);

    servAddr**.**sin\_port = **htons**(SERVER\_PORT);

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

    printf("succesfully created stream socket \n");

**bind**(sd, (struct sockaddr \*)&servAddr, sizeof(servAddr));

    printf("bound local port successfully \n");

**listen**(sd, 5);

**while** (1)

    {

        printf("waiting for client connection on port TCP %u\n", SERVER\_PORT);

        newsd = **accept**(sd, (struct sockaddr \*)&cliAddr, &clilen);

        printf("received connection from host [IP %s, TCP port %d]\n", **inet\_ntoa**(cliAddr**.**sin\_addr), **ntohs**(cliAddr**.**sin\_port));

**do**

        {

            memset(line, 0x0, MAX\_MSG);

            n = **recv**(newsd, line, MAX\_MSG, 0);

            line[n] = "\n";

            printf("received from host [IP%s,TCPPort %d]:%s\n", **inet\_ntoa**(cliAddr**.**sin\_addr), **ntohs**(cliAddr**.**sin\_port), line);

        } **while** (abs(strcmp(line, "quit")));

        printf("closing connection with host [IP%s,TCP port %d]\n", **inet\_ntoa**(cliAddr**.**sin\_addr), **ntohs**(cliAddr**.**sin\_port));

**close**(newsd);

    }

}

Program:

UDP Client

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

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

**#include** <arpa/inet.h>

**#include** <netdb.h>

**#include** <unistd.h>

**#include** <math.h>

**#include** <string.h>

**#include** <stdio.h>

**#include** <stdlib.h>

**#define** MAX\_MSG 100

**#define** REMOTE\_SERVER\_PORT 1500

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

{

int sd, rc, i;

struct sockaddr\_in cliAddr, remoteservAddr;

struct hostent \*h;

**if** (argc **<** 3)

{

printf("Usage : %s <server><data1>....<dataN>\n", argv[0]);

exit(1);

}

h = **gethostbyname**(argv[1]);

**if** (h **==** **NULL**)

{

printf("%s : unknown host '%s \n", argv[0], argv[1]);

exit(1);

}

remoteservAddr**.**sin\_family = h**->**h\_addrtype;

memcpy((char \*)&remoteservAddr**.**sin\_addr**.**s\_addr, h**->**h\_addr\_list[0], h**->**h\_length);

remoteservAddr**.**sin\_port = **htons**(REMOTE\_SERVER\_PORT);

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

**if** (sd **<** 0)

{

printf("%s : cannot open socket \n", argv[0]);

exit(1);

}

cliAddr**.**sin\_family = AF\_INET;

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

cliAddr**.**sin\_port = **htons**(0);

rc = **bind**(sd, (struct sockaddr \*)&cliAddr, sizeof(cliAddr));

**if** (rc **<** 0)

{

printf("%s : cannot bind port \n", argv[0]);

exit(1);

}

**for** (i = 2; i **<** argc; i++)

{

rc = **sendto**(sd, argv[i], strlen(argv[1]) + 1, 0, (struct sockaddr\*)&remoteservAddr, sizeof(remoteservAddr));

**if** (rc **<** 0)

{

printf("%s : cannot send data %d \n", argv[0], i - 1);

**close**(sd);

exit(1);

}

}

**return** 1;

}

UDP Server

**#include** <sys/types.h>

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

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

**#include** <arpa/inet.h>

**#include** <netdb.h>

**#include** <unistd.h>

**#include** <math.h>

**#include** <string.h>

**#include** <stdio.h>

**#include** <stdlib.h>

**#define** MAX\_MSG 100

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

{

int sd, rc, n, clilen;

struct sockaddr\_in cliAddr, servAddr;

char msg[MAX\_MSG];

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

**if** (sd **<** 0)

{

printf("%s : Cannot open socket \n", argv[0]);

exit(1);

}

servAddr**.**sin\_family = AF\_INET;

servAddr**.**sin\_addr**.**s\_addr = **ntohl**(INADDR\_ANY);

servAddr**.**sin\_port = **htons**(1500);

rc = **bind**(sd, (struct sockaddr \*)&servAddr, sizeof(servAddr));

**if** (rc **<** 0)

{

printf("Cannot bind \n");

exit(1);

}

**while** (1)

{

memset(msg, 0x0, MAX\_MSG);

clilen = sizeof(cliAddr);

n = **recvfrom**(sd, msg, MAX\_MSG, 0, (struct sockaddr \*)&cliAddr, &clilen);

**if** (n **<** 0)

{

printf("%s : Cannot receive data \n", argv[0]);

**continue**;

}

printf("%s : from %s : UDP %u : %s \n", argv[1], **inet\_ntoa**(cliAddr**.**sin\_addr), **ntohs**(cliAddr**.**sin\_port), msg);

}

**return** 0;

}

Program:

**#include** <stdio.h>

**#include** <string.h>

**#define** N strlen(g)

char t[28], cs[28], g[] = "10001000000100001";

int a, e, c;

void **xor** () {

**for** (c = 1; c **<** N; c++)

cs[c] = ((cs[c] **==** g[c]) **?** '0' **:** '1');

}

void **crc**()

{

**for** (e = 0; e **<** N; e++)

cs[e] = t[e];

**do**

{

**if** (cs[0] **==** '1')

**xor**();

**for** (c = 0; c **<** N - 1; c++)

cs[c] = cs[c + 1];

cs[c] = t[e++];

} **while** (e **<=** a + N - 1);

}

int **main**()

{

printf("\nEnter data : ");

scanf("%s", t);

printf("\n----------------------------------------");

printf("\nGeneratng polynomial : %s", g);

a = strlen(t);

**for** (e = a; e **<** a + N - 1; e++)

t[e] = '0';

printf("\n----------------------------------------");

printf("\nModified data is : %s", t);

printf("\n----------------------------------------");

**crc**();

printf("\nChecksum is : %s", cs);

**for** (e = a; e **<** a + N - 1; e++)

t[e] = cs[e - a];

printf("\n----------------------------------------");

printf("\nFinal codeword is : %s", t);

printf("\n----------------------------------------");

printf("\nTest error detection 0(yes) 1(no)? : ");

scanf("%d", &e);

**if** (e **==** 0)

{

**do**

{

printf("\nEnter the position where error is to be inserted : ");

scanf("%d", &e);

} **while** (e **==** 0 **||** e **>** a + N - 1);

t[e - 1] = (t[e - 1] **==** '0') **?** '1' **:** '0';

printf("\n----------------------------------------");

printf("\nErroneous data : %s\n", t);

}

**crc**();

**for** (e = 0; (e **<** N - 1) **&&** (cs[e] **!=** '1'); e++)

;

**if** (e **<** N - 1)

printf("\nError detected\n\n");

**else**

printf("\nNo error detected\n\n");

printf("\n----------------------------------------\n");

**return** 0;

}