Solution for Lab exercise1. Write a program for error detection using CRC-CCITT (x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1.).

#include<stdio.h>

#include<string.h>

#define N strlen(g)

char t[28],cs[28],g[]="100000100110000010001110110110111";

inta,e,c;

voidxor(){

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

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

}

voidcrc(){

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("\n CRC checksum is : %s",cs);

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

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

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

printf("\nFinalcodeword transmitted 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);

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("\n CRC checksum is : %s",cs);

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

else

{

printf("\n CRC checksum is : %s",cs);

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

}

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

return 0;

}

2. Write a Program in C/ C++ for hamming code generation for error

detection/correction

#include<stdio.h>

void main() {

int data[10];

intdataatrec[10],c,c1,c2,c3,i;

printf("Enter 4 bits of data one by one\n");

scanf("%d",&data[0]);

scanf("%d",&data[1]);

scanf("%d",&data[2]);

scanf("%d",&data[4]);

//Calculation of even parity

data[6]=data[0]^data[2]^data[4];

data[5]=data[0]^data[1]^data[4];

data[3]=data[0]^data[1]^data[2];

printf("\nEncoded data is\n");

for(i=0;i<7;i++)

printf("%d",data[i]);

printf("\n\nEnter received data bits one by one\n");

for(i=0;i<7;i++)

scanf("%d",&dataatrec[i]);

c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];

c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];

c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];

c=c3\*4+c2\*2+c1 ;

if(c==0) {

printf("\nNo error while transmission of data\n");

}

else {

printf("\nError on position %d",c);

printf("\nData sent : ");

for(i=0;i<7;i++)

printf("%d",data[i]);

printf("\nData received : ");

for(i=0;i<7;i++)

printf("%d",dataatrec[i]);

printf("\nCorrect message is\n");

//if errorneous bit is 0 we complement it else vice versa

if(dataatrec[7-c]==0)

dataatrec[7-c]=1;

else

dataatrec[7-c]=0;

1

for (i=0;i<7;i++) {

printf("%d",dataatrec[i]);

}

}

}

Socket Program

4.Write a client-server program using TCP/IP sockets in which client requests

for a file by sending the file name to the server, and the server sends back the

contents of the requested file if present.

/\*SERVER - Create a file called hello.txt in the current directory and pass that as the file name

for server \*/

#include<stdio.h>

#include<arpa/inet.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define SERV\_TCP\_PORT 5035

#define MAX 60

int i, j, tem;

char buff[4096], t;

FILE \*f1;

int main(int afg, char \*argv)

{

int sockfd, newsockfd, clength;

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

char t[MAX], str[MAX];

strcpy(t,"exit");

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

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

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

serv\_addr.sin\_port=htons(SERV\_TCP\_PORT);

printf("\nBinded");

bind(sockfd,(struct sockaddr\*)&serv\_addr, sizeof(serv\_addr));

printf("\nListening...");

listen(sockfd, 5);

clength=sizeof(cli\_addr);

newsockfd=accept(sockfd,(struct sockaddr\*) &cli\_addr,&clength);

close(sockfd);

read(newsockfd, &str, MAX);

printf("\nClient message\n File Name : %s\n", str);

f1=fopen(str, "r");

while(fgets(buff, 4096, f1)!=NULL) {

write(newsockfd, buff,MAX);

printf("\n"); }

fclose(f1);

printf("\nFile Transferred\n");

return 0;

}

26

//CLIENT

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define SERV\_TCP\_PORT 5035

#define MAX 60

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

{

int sockfd,n;

struct sockaddr\_in serv\_addr;

struct hostent\*server;

char send[MAX],recvline[MAX],s[MAX],name[MAX];

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

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

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

serv\_addr.sin\_port=htons(SERV\_TCP\_PORT);

connect(sockfd,(struct sockaddr\*)&serv\_addr,sizeof(serv\_addr));

printf("\nEnter the source file name : \n");

scanf("%s",send);

write(sockfd,send,MAX);

while((n=read(sockfd,recvline,MAX))!=0) {

printf("%s",recvline);

}

close(sockfd);

return 0;

}