**LEAKY BUCKET PROGRAM**

#include <iostream>

using namespace std;

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

cout << "Enter size of buffer: ";

int bufferSize;

cin >> bufferSize;

cout << "Enter output rate: ";

int outputRate;

cin >> outputRate;

cout << "Enter grant rate: ";

int grantRate;

cin >> grantRate;

int dataSize;

int bufferFilled = 0;

int grantFilled = 0;

cout << "Enter simulation time: ";

int stime;

cin>>stime;

for (int i=0; i<stime; i++) {

cout << "Enter size of data: ";

cin >> dataSize;

if (dataSize <= bufferSize - bufferFilled) {

cout << dataSize << " packets added to buffer" <<endl;

bufferFilled += dataSize;

}

else {

cout << (dataSize - (bufferSize - bufferFilled)) << " packets dropped, insufficient space available in buffer" <<endl;

bufferFilled = bufferSize;

}

if (bufferFilled != 0) {

if (bufferFilled > outputRate) {

cout << outputRate << " packets sent to grant buffer" << endl;

bufferFilled -= outputRate;

grantFilled+= outputRate;

}

else {

cout << bufferFilled << " packets sent to grant buffer" << endl;

grantFilled+= bufferFilled;

bufferFilled = 0;

}

}

else {

cout << "No data to send to grant buffer, main buffer empty" << endl;

}

for (int j=0; j<grantRate; j++) { //One packet in 1/g seconds

if (grantFilled > 0) {

cout<<"1 packet sent"<<endl;

grantFilled--;

}

else break;

}

cout << "Grant buffer has " << grantFilled << " packets." <<endl;

}

return 0;

}

**SERVER AND CLIENT (SOCKET CONNECTION)**

**//server.c**

#include <stdio.h>

#include <arpa/inet.h>

#include <fcntl.h>

#include <unistd.h>

int main()

{

int welcome, new\_soc, fd, n;

char buffer[1024], fname[50];

struct sockaddr\_in addr;

welcome = socket(PF\_INET, SOCK\_STREAM, 0);

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(7891);

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

bind(welcome, (struct sockaddr \*) &addr, sizeof(addr));

printf("\nServer is Online");

listen(welcome, 5);

new\_soc = accept(welcome, NULL, NULL);

recv(new\_soc, fname, 50, 0);

printf("\nRequesting for file: %s\n", fname);

fd = open(fname, O\_RDONLY);

if (fd < 0)

send(new\_soc, "\nFile not found\n", 15, 0);

else

while ((n = read(fd, buffer, sizeof(buffer))) > 0)

send(new\_soc, buffer, n, 0);

printf("\nRequest sent\n");

close(fd);

return 0;

}

**//Client.c**

#include <stdio.h>

#include <arpa/inet.h>

#include <fcntl.h>

#include <unistd.h>

int main()

{

int soc, n;

char buffer[1024], fname[50];

struct sockaddr\_in addr;

soc = socket(PF\_INET, SOCK\_STREAM, 0);

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(7891);

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

while(connect(soc, (struct sockaddr \*) &addr, sizeof(addr))) ;

printf("\nClient is connected to Server");

printf("\nEnter file name: ");

scanf("%s", fname);

send(soc, fname, sizeof(fname), 0);

printf("\nRecieved response\n");

while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)

printf("%s", buffer);

return 0;

}

**Hamming Code**

#include<stdio.h>

void main() {

int data[7],rec[7],i,c1,c2,c3,c;

printf("this works for message of 4bits in size \nenter message bit one by one: ");

scanf("%d%d%d%d",&data[0],&data[1],&data[2],&data[3]);

data[6]=data[0]^data[2]^data[3];

data[5]=data[0]^data[1]^data[3];

data[4]=data[0]^data[1]^data[2];

printf("\nthe encoded bits are given below: \n");

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

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

}

printf("\nenter the received data bits one by one: ");

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

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

}

c1=rec[6]^rec[3]^rec[2]^rec[0];

c2=rec[5]^rec[3]^rec[1]^rec[0];

c3=rec[4]^rec[2]^rec[1]^rec[0];

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

if(c==0) {

printf("\ncongratulations there is no error: ");

} else {

printf("\nerror on the postion: %d\nthe correct message is \n",c);

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

rec[7-c]=1;

else

rec[7-c]=0;

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

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

}

}

}

**CRC PROGRAM**

#include<iostream>

using namespace std;

void division(int temp[],int gen[],int n,int r)

{

for(int i=0;i<n;i++)

{

if (gen[0]==temp[i])

{

for(int j=0,k=i;j<r+1;j++,k++)

if(!(temp[k]^gen[j]))

temp[k]=0;

else

temp[k]=1;

} }}

int main()

{int n,r=16,message[50],temp[50];

int gen[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};

cout<<"At Sender's End "<<endl;

cout<<"Enter the number of message bits : ";

cin>>n;

cout<<"Enter the message : ";

for(int i=0;i<n;i++)

cin>>message[i];

for(int i=0;i<r;i++)

message[n+i] = 0;

for(int i=0;i<n+r;i++)

temp[i] = message[i];

division(temp,gen,n,r);

cout<<"CRC : ";

for(int i=0;i<r;i++)

{

cout<<temp[n+i]<<" ";

message[n+i] = temp[n+i];

}

cout<<endl<<"Transmitted Message : ";

for(int i=0;i<n+r;i++)

cout<<message[i]<<" ";

cout<<endl<<endl<<"At Receiver's End "<<endl;

cout<<"Enter the received message : ";

for(int i=0;i<n+r;i++)

cin>>message[i];

for(int i=0;i<n+r;i++)

temp[i] = message[i];

division(temp,gen,n,r);

for(int i=0;i<r;i++)

{

if(temp[n+i])

{

cout<<"Error detected in received message.";

return 0;

} }

cout<<"No error in received Message.\nReceived Message : ";

for(int i=0;i<n;i++)

cout<<message[i]<<" ";

return 0;

}