-Shreesha H Shetty 1BM21CS209

1. Write a program for error detecting code using CRC-CCITT(16-bits).

Code:

```
#include<stdio.h>
char m[50],g[50],r[50],q[50],temp[50];
void caltrans(int);
void crc(int);
void calram();
void shiftl();
int main() {
       int n,i=0; char ch,flag=0;
       printf("Enter the frame bits:");
       while((ch=getc(stdin))!='\n')
               m[i++]=ch;
       n=i;
       for(i=0;i<16;i++)
               m[n++]='0';
       m[n]='\0';
       printf("Message after appending 16 zeros:%s",m);
       for(i=0;i<=16;i++)
               g[i]='0';
       g[0]=g[4]=g[11]=g[16]='1';
       g[17]='\0';
       printf("\ngenerator:%s\n",g);
       crc(n);
       printf("\n\nquotient:%s",q);
       caltrans(n);
       printf("\ntransmitted frame:%s",m);
       printf("\nEnter transmitted frame:");
       scanf("\n\%s",m);
       printf("CRC checking\n");
```

```
crc(n);
       printf("\n\nlast remainder:%s",r);
       for(i=0;i<16;i++)
               if(r[i]!='0')
                       flag=1;
               else
                       continue;
       if(flag==1)
                printf("Error during transmission");
       else
               printf("\n\nReceived freme is correct");
}
void crc(int n){
       int i,j;
       for(i=0;i<n;i++)
               temp[i]=m[i];
       for(i=0;i<16;i++)
               r[i]=m[i];
       for(i=0;i< n-16;i++){
               if(r[0]=='1'){
                       q[i]='1';
                       calram();
               }
               else\{
                       q[i]='0';
                       shiftl();
               }
               r[16]=m[17+i];
               r[17]='\0';
               for(j=0;j<=17;j++)
                       temp[j]=r[j];
       }
       q[n-16]='\0';
}
```

```
void calram() {
        int i,j;
        for(i=1;i \le 16;i++)
                r[i-1]=((int)temp[i]-48)^{((int)g[i]-48)+48};
}
void shiftl(){
        int i;
        for(i=1;i \le 16;i++)
                r[i-1]=r[i];
}
void caltrans(int n) {
        int i,k=0;
        for(i=n-16;i < n;i++)
                m[i]=((int)m[i]-48)^{((int)r[k++]-48)+48};
        m[i]='\0';
}
```

Output:

						DOM	Page N	lo.
						Date	1 1	
_					,			
Q	write	a proq	ram fo	remor	detecting	9 code	using	
E	CRC-	CCITT (16-bit	s).	1772 3	1.60	I DIN	11-11-0
				-0 B	9			100
)	N 600	ude < st	dia h	1 1 1 1 1	2	Eldin.	s stalies	i Vi
_				1 150	1, 9, 1507			
	char	m [30]	,9150	174.130	1,9,1901	. iemj	1 301	EH-
	void	caltra	ns(int);	- 150	18013	Joan S	ion
	void	orc(int	-):	- " - E	1	(D. 6.	LENE I	Sign .
	11010			5 4 6 1 1 1 d 4	Share and			
		shift10			1, 5, 6, 7, 7		Tiola V	
	roid	0.4,,,	,,	1		- Federale I	1-0-	MEM .
	int me	amc) {			3 300 N	200		a 202
			, i=0;	A Automati		37.4	PENDIN	VIII.
							- pur	
-	10 100 00	char	ch, fla	9=0;	1			
					rame bit			
		while	(Cch =	getclst	din))!=	= (\n')		1
	1		m[i++1= cl	v;			
		n=1;	- 1 3		1.1	100		LAR.
		for (i=	0; 1< 16	5; (++)	1.5			100
	=	,	m[n++	7 = '0';				17
		m[n]:	110%					7.31
		points (" Mess	000 011	44. 0004	d' · ·		. yeh
		for (i=	0; 14=	16; i+T	ен аррен	ding le	zeros	: 73
					,			1.12
	1		9LILE	'O':				
			g[il=		01441	447.	125	
		g[o]=	9[4] =	9 [117 =	9[16]=	'1';	13.0	
		g[o]=		9 [117 =	91161=	'1';	1 (14) 1 (14) 3 (14)	
		g[o]=	9[4] =	9 [117 =	9[16]=	a de l'ordo	15 Page	No
		g[o]=	9[4] =	9 [117 =	9[16]=	'1';	15 Page	No.
		g[o]=	9[4] =	9 [117 =	9[16]=	10C	15 Page	No.
		g[0]=	g[4] = 2 '\0';	g [11] z	g[16]=	Date	1	
		g[0]= g[17] prin	g[4] = 2 '\0';	g [117 z	on: 7.5°	Date	יריי על	
		g[0]= g[17] prin	g[4] = 2 '\0';	g [117 z	W W	Date	יריי על	
		g[o]= g[17] print crc(print calt	g[4] = 2 (0); ef("\n (n); ef("\n rans(n);	general	t:(7.8.1).	Date)) 1973 - 144 - 14	Mor
		g[o]= g[17] print crci print calt	g[4] = 2 '\0'; lef("\n (n); if ("\ne rans(n)	generali	00: 7.5°	Dote Date)) 1973 - 144 - 14	Mor
		g[o]= g[17] print crci print calt	g[4] = 2 '\0'; lef("\n (n); if ("\ne rans(n)	generali	00: 7.5°	Dote Date)) 1973 - 144 - 14	Mor
		g[0]= g[17] print crc1 print caut print	g[4] = (10); ef("\n (n); ef ("\n f ("\n f ("\n	generak guerak quotient i; transnii	t: (7.5); a	Dot Date	יייי נול אילייייי וול אילייייייייייייייייייייי	North
		g[0]= g[17] print crc1 print caut print	g[4] = (10); ef("\n (n); ef ("\n f ("\n f ("\n	generak guerak quotient i; transnii	t: (7.5); a	Dot Date	יייי נול אילייייי וול אילייייייייייייייייייייי	North
		g[0]= g[17] print crct print caut print scan prin	8[4] = (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	general guotant guotant i); transnút s", m);	on: Xsi; e (i) con - f theol fray (ii) con - f theol fray (iii) con - f (ii	Dot Date 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	1) 370 - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Nistra .
		print calt print scan print cre print	g[4] = (10); ef("\n); f("\n) f("\n) f("\n) f("\n); df("\n); df("\n);	generations generated gene	i: //s"; a i: //s"; a ii: //s"; a iii: //s"; iii: //s iiii: //s iii: //s ii: //s iii: //s ii: //s i	Dot Date India); 1- I quest (i 1 (i); 10 (i 7) (i 7) (i 7) (i 7) (i 7)	1) 370 - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Nistra .
		print calt print scan print cre print	g[4] = 2 '\0'; ep(c'\n (n); f c"\n	guerak quotient (); transnii 5", m); C chick (16; in	on: 1.5", con the frag	Dot Date India); 1- I quest (i 1 (i); 10 (i 7) (i 7) (i 7) (i 7) (i 7)	1) 370 - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Nistra .
		print calt print scan print cre print	g[4] = 2 '\0'; ep(c'\n (n); f c"\n	guerak quotient (); transmii 5", m); C check 1 (ast ris < 16; in	on: 1/s"; is the street of the street fraging in " () the street fraging in the street	Dot Date 154/19: 11 parts 11 parts 11 parts 11 parts 12 parts 13 parts 13 parts 13 parts 14 parts 15 parts	1) 370 - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Nistra .
		print calt print scan print cre print	g[4] = 2 '\0'; ep(c'\n (n); f c"\n	guerak quotient (); transmii 5", m); C check 1 (ast ris < 16; in	on: 1/s"; is the street of the street fraging in " () the street fraging in the street	Dot Date 154/19: 11 parts 11 parts 11 parts 11 parts 12 parts 13 parts 13 parts 13 parts 14 parts 15 parts	1) 370 - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Nistra .
		print calt print scan print cre print	g[4] = 2 '\0'; ef("\n ; f("\n f(\n f(\n f(\n f(\n f(\n f(\n f(\n f(generaling quotient (1); fransnút (2); ché cl	on: 1.5" (1/8"; color of the of frage in " (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1)	Dor Date	/ // // // // // // // // // // // // /	Nistra .
		print calt print scan print cre print	g[4] = 2 '\0'; ef("\n ; f("\n f(\n f(\n f(\n f(\n f(\n f(\n f(\n f(generaling quotient (1); fransnút (2); ché cl	on: 1.5" (1/8"; color of the of frage in " (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1) (1/100 - 1)	Dor Date	/ // // // // // // // // // // // // /	Nistra .
		print crec print scan print for (\$[4] = (10); \$(n); \$f("\n); \$f("\n); \$f("\n); \$f("\n'; \$f("\n'; \$f("\n'; \$f("\n'; \$f(\n'); \$f(\n	generalistics ge	on: Xs"; est	Dote Dote D	/ / / / / / / / / / / / / / / / / / /	Nicom
		print crec print scan print for (g[4] = g[generals quotient grotient s', m); c check 16; i+ 16g=1	on: Xs"; est	Dote Dote D	/ / / / / / / / / / / / / / / / / / /	Notes and the second se
		print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	guerak quotient (); fransnii 5", m); C check (16; in [1:1] = '0 flog = 1 continue 1)	on: 1/s" 1: 1/s"; a 1: 1/s"; a 1: 1/s"; a 1: 1/s" 1:	Dot Date Date); ser	history
		print crec print scan print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	guerak quotient (); fransnii 5", m); C check (16; in [1:1] = '0 flog = 1 continue 1)	on: 1/s" 1: 1/s"; a 1: 1/s"; a 1: 1/s"; a 1: 1/s" 1:	Dot Date Date); ser	history
		print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	guerak quotient (); fransnii 5", m); C check (16; in [1:1] = '0 flog = 1 continue 1)	on: Xs"; est	Dot Date); ser	history
		print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	guerak quotient (); fransnii 5", m); C check (16; in [1:1] = '0 flog = 1 continue 1)	on: 1/s" 1: 1/s"; a 1: 1/s"; a 1: 1/s"; a 1: 1/s" 1:	Dot Date); ser	history
	3	print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	generals quotient quotient (); fransnii (3", m); (C check (16; i+ [i]] = '0 flog = 1 continue 1) f("\n Re	on: 1/s" 1: 1/s"; a 1: 1/s"; a 1: 1/s"; a 1: 1/s" 1:	Dot Date 164.8. 164.	mission	· · · · · · · · · · · · · · · · · · ·
	}	print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	generals quotient quotient (); fransnii (3", m); (C check (16; i+ [i]] = '0 flog = 1 continue 1) f("\n Re	on: 1/s"; estated frag	Dot Date 164.8. 164.	",n),	· · · · · · · · · · · · · · · · · · ·
	3	print calt print crec print for (g[4] = 2 '\0'; ef("\n (n); ef ("\n f("\n f("\n (i=0); if (r) the f(ag== print	generations guidant guotient (1); fransnii 5", m); C chock 16; i+ [i]! = '0 flog = 1 continue 1) f("Crro	on: 1/s"; estated frag	Dote Date And 19: Transfine: 11: Transfine	"in);	· · · · · · · · · · · · · · · · · · ·
3	3	g[o]= g[i7] puin crci print calt print scan prin for (b elx	g[4] = 2 '\0'; ef("\n (n); f("\n f("\n f("\n (i=0); if(r) ene f(ene print print)	generations guident guotient (V; fransnii) (C) Check (V) (C) (C) (C) (C) (C) (C) (C) (C) (C) (C	on: 1/s"; established from the control of the contr	Dot Date 101/192 101/193 10	"in);	· · · · · · · · · · · · · · · · · · ·

4.50 2.75

	DOM5 Page No.
	Date / /
void Crc (int n) { 2 (colors	no 1 Martin
int is;	CHILE
for (i=0; icn; i++)	are also from
templil = m[il;	cathenistely
for (i=0; i(16; i++)	ort of 20 (Larry
r[il=m[il; quan	2004 (185)
for (1=0; i <n-16; f<="" i++)="" td=""><td>DANDREWA</td></n-16;>	DANDREWA
if (r[0]== 11){	D 41,7 1 12 Co. 12 Co.
1 calrom(l)	151,0=1) 40}
ene { (10) = 1	(:[(r[:)
alil= 0; col	1
3 shifte();	547
r[16]= m[17+1] poid	433
r[17] = '\0';	
(" " for (j=0; j = 17; j++)	Lead 1
} femp[j7=r[j); vile
19 [n-16] = 10 1 bank of) living
1 1 Je 1 2 3	
void calram(){	I R
int i, j;	
for(ir=1; i<=16; i++)	F.2
	[i1-48)^((int)g[i]-4

DOMS | Page No. word shiftles were trobe and many it will for (i=1; ix=16; i++) 3 3 3001 300 1 300 1000 r[i-1] = r[i]; דות אואל ביו ביותו לפווז ביו בפלבי ביותובוי roid cultrans (intn) 1 many but in with ment with for (i=n-16; i(n; i++) starte with fact m[:]=((int) m[:]-48)~((int) r[k++]-28) 48; m[il='10'; Output: Enter the frame bits: 1011 Mussage after appending 16 govos: 1011 0000 00000000 Generator: 10001000000100001 Transmitted frame: 10111011000101101011 Enter transmitted frame: 10111011000101101011 CRC checking Last remainder: 0000 0000 0000 0000 Received frame is correct.

2. Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#include<stdio.h>
int main(){
       int incoming, outgoing, buck size, n, store = 0;
       printf("Enter bucket size, outgoing rate and no of inputs: ");
       scanf("%d %d %d", &buck size, &outgoing, &n);
       while (n != 0)
               printf("Enter the incoming packet size : ");
               scanf("%d", &incoming);
               printf("Incoming packet size %d\n", incoming);
               if (incoming <= (buck size - store)){
                      store += incoming;
                      printf("Bucket buffer size %d out of %d\n", store, buck size);
               else {
                      printf("Dropped %d no of packets\n", incoming - (buck size - store));
                      printf("Bucket buffer size %d out of %d\n", store, buck size);
                      store = buck size;
               store = store - outgoing;
               printf("After outgoing %d packets left out of %d in buffer\n", store, buck size);
}
```

Output:

```
Enter bucket size, outgoing rate and no of inputs: 8 6 4
Enter the incoming packet size :
Incoming packet size 3
Bucket buffer size 3 out of 8
After outgoing -3 packets left out of 8 in buffer
Enter the incoming packet size
Incoming packet size 3
Bucket buffer size 0 out of 8
After outgoing -6 packets left out of 8 in buffer
Enter the incoming packet size: 4
Incoming packet size 4
Bucket buffer size -2 out of 8
After outgoing -8 packets left out of 8 in buffer
Enter the incoming packet size: 3
Incoming packet size 3
Bucket buffer size -5 out of 8
After outgoing -11 packets left out of 8 in buffer
```

3. Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Code:

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive") connectionSocket, addr =
    serverSocket.accept() sentence = connectionSocket.recv(1024).decode()

file=open(sentence,"r") l=file.read(1024)

connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
```

Output:

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerTCP.py The server is ready to receive
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ClientTCP.py
Enter file name: ServerTCP.py

From Server:
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
```

while 1: print("The server is ready to receive") connectionSocket, addr = serverSocket.accept() sentence = connectionSocket.recv(1024).decode() file = open(sentence, "r") l = file.read(1024) connectionSocket.send(l.encode()) print('\nSent contents of ' + sentence) file.close() connectionSocket.close()

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerTCP.py
The server is ready to receive

Sent contents of ServerTCP.py
The server is ready to receive
```

DOM:	5	Page No.
Date	1	1

and the second			
9.	125	ing TCP/IP sockels, write a	chiand -
0.	100	make client sending the file	culti server program
	10	section in the file	name and the serven
	to	send back the contents of th	e requested file if
		sent.	20061 - 1-910 ms.
_>		A GAT TINGT, SACE STOCKE	server Sorker - soc
	Cli	ent TCP. Pyrons , wood (1901)	Land The server street
		om sockel import *	
	Ser	ronName = 127.0.0.1	The state of the s
		verPort = 12000 1000 21 1860	
		nd Sockel = sockel CAF-INET.S	
-5.3		ntSocket connect ((server Nam	
	sen	tence = input (thenter file name	(P) 1000 = 100 V
	1.70		ו יין בין דע בנמכל לאל
	clies	al Bocket. send (sentence. enc	
		contents = client socket recv(1	
-		d ('In From Serven: In')	
	-	of (filecontents) (made)	10°C 40 10 17 110 110 110 110 110 110 110 110
	clie	nt Bocket. close ()	The state of the second
		The Park State of	The Residence
		A series series	100 100 100
+		A COLL	3650
+		1 7 24.	4
1			
		1	
			DOMS Page No
			Date / /
	W.17	Server TCP. py 1 1 1 11 21	1.4x2 1.41- 1941 V E
	331	from sockel import *	THE PARTY OF THE P
		server Name = "127-0.0.1"	as all donde twee of the co
		serverPort = 12000	- Average
		serverSocket = socket (AF-INE	T, SOCK_STR GAM)
		server Socket bind (server Name,	
			וליסאין בספוצעל ווייזף בי
		***	Servention of 127
_		print ("The server is ready	
			1000000
		sentence = connection Socket	
57			
à.	_2	file = open (sentence, "")	strikmer at input ("In
		1 = file.read (1024)	h. 1
1		connection socket send ().	
	3	print ("In Sent contents of	
			prist (In From Si
		connection Socket-close()	7.7092
		(C)	Stand Socked . Eloso
		The state of the s	Win I I
		100 Mg 1 a 1 a 1 a 1	C. P. Sepanson B.
		. V	
		V	
			The state of the s

		Dotts Page No.
	withing the transfer of the state of the	A Miller "UDF Likes HE"
_		P. Pyla Mada Stan of
		ecure of the Amore
+	THE STATE OF THE S	The of present of the
, 1	when you run Client TC	
		en TCD. py 119. Table tonis
1	From Serven:	them teached make a
(The file from Server TC	P. py will be copied and
	displayed here)	4100 Per E-12000
1		estim Ecope of madicked (A
>	In Server TCP. py 1	a foster de estapual / va Car
	The serven 13 ready to	receive and to the
	Sent contents of Server	
100	The surver is ready to ree	Mechanis serve Arasia
3	enderson (alexander	
	(6, 3-1) m;) m	part the consense dead
4	Walter British Born	of for own frombush
	the second in the second second	The production is and
		Sized Socked - class ()
		chart sockul closeld
-		
+		

4.Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Code:

```
ClientUDP.py
from socket import *
serverName ="127.0.0.1"
serverPort = 12000
clientSocket = socket(AF INET, SOCK DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
  # print(str(i), end = ")
clientSocket.close()
clientSocket.close()
ServerUDP.py
from socket import *
serverPort = 12000
serverSocket = socket(AF INET, SOCK DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
   sentence, clientAddress = serverSocket.recvfrom(2048)
   sentence = sentence.decode("utf-8")
   file=open(sentence,"r") con=file.read(2048)
   serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
   print ('\nSent contents of ', end = ' ')
   print (sentence)
   # for i in sentence:
```

```
# print (str(i), end = ")
file.close()
```

Output:

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerUDP.py
The server is ready to receive
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ClientUDP.py
Enter file name: ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
   sentence, clientAddress = serverSocket.recvfrom(2048)
   sentence = sentence.decode("utf-8")
   file = open(sentence, "r")
   con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
    print('\nSent contents of ', end=' ')
    print(sentence)
    # for i in sentence:
    # print (str(i), end = '')
    file.close()
Process finished with exit code 0
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerUDP.py
The server is ready to receive
Sent contents of ServerUDP.py
```

ž	DOM5 Page No. Date / /
4	+ Using UDP sockets, write a client-server program
	to make diene sending the filename and the
	server to send back a the contents of the requeste
	file if present.
→	2 when you now Converted for 11-6
	Client UDP. py in the second in which
	from socket import *
()	
	Server Port = 12000
	client Gockel - Sockel (AF. INET, SOCK_DGRAM)
	sentence = input (in Gulen file name: ")
6%	client socket. send to (bytes (sentence, "retf-8"),
	(serverName, server Port))
	filecontents, serven Address = client socket recrifrom (
	print ('In Reply from Serven: In')
	print (file contents decode ("ut f-8"))
	# for i'an filecontents:
	# print (str(i), end = ")
	client Socket-close()
	client Socket. close ()
	Caption of the fact
	Ort15 Page Ho.
	Data / /
Su	Date / /
	Data / /
fre	ruer UDP. py:
fre	ruer UDR py: om socket import * rver Port = 12000 og 1991 and my socket
fre ser	om socket import * ruer Port = 12000 og 111 mm men men men men men men men men men
fro ser se se	ruer UDP. py: om socket import * rver Port = 12000 og 101 in 10
fre ser se ser	ruer UDP. py: om socket import * rver Port = 12000 rver Socket = 30cket (AF. INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) int ("The server is ready to receive")
fre ser se ser	ruer UDP. py: om socket import * rver Port = 12000 rver Socket = 30cket (AF. INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) int ("The server is ready to receive")
ser ser ser pri	ruer UDP py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ser ser ser pri	ruer UDP py: om socket import * rver Port = 12000 rver Socket = 30cket (AF. INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) ent ("The server is ready to receive") ide 1: sentence, client Adobress = 3erver Socket. recufrom (200 sentence) file = open (sentence, "r")
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 og the name of the rver Port = 12000 og the name of the rver Port = 12000 og the name of the rver Port = 12000 og the name of the rver Socket bind (("127.0.0.1", server Port)) ent ("The server is ready to receive") inte 1: sendence, client Address = server Socket. recv from (20. sendence; surlence, decode("utf + 8") file = open (sentence, "r") con = file. read (2048)
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 rver Socket = socket (AF-INET, SOCK-DGRAM) rver Socket bind (("127.0.0.1", server Port)) int ("The server is ready to receive") sentence, client Address = server Socket recufrom (20. sentence: sentence, decode ("utf +8") file = open (sentence, "r") con = file. read (2048) server Socket. send to Coytes (con, "uef-8"), client Address)
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 rver Socket = socket (AF-INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) int ("The server is ready to receive") sentence, client Adobres = server Socket recufrom (20. sentence, client Adobres = server Socket recufrom (20. sentence = sentence decode ("utf-8") file = open (sentence, "r") com = file read (2048) server Socket send to Coytes (con, "uf-8"), print ("M Sent contents of", end=!!)
ser ser ser pri	ruer UDP py: om socket import * rver Port = 12000 og the inversocket = 30cket (AF. INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) ent ("The server is ready to receive") ide 1: sendence, client Address = server Socket. recufrom (20. sentence: sendence, decode ("utf-8") file = open (sentence, "r") con = file. read (2048) server Socket. send to (bytes (con, "utf-8"), print ("In Sent contents of", end: 2!!) print (sentence)
ser ser ser pri	ruer UDP. py: om socket import * rver Port = 12000 og the new port of the server is ready to receive of the server from (20. Sentence, client Address = server Socket. recv from (20. Sentence: sentence, "r") con = file. read (2048) Server Socket. send to Chytes (con, "ref-8"), print ("In Sent contents of "end="!") print ("In Sent contents of "end="!") # for i in sentence
ser ser ser pri	ruer UDP py: om socket import * rver Port = 12000 og the inversocket = 30cket (AF. INET, SOCK_DGRAM) rver Socket bind (("127.0.0.1", server Port)) ent ("The server is ready to receive") ide 1: sendence, client Address = server Socket. recufrom (20. sentence: sendence, decode ("utf-8") file = open (sentence, "r") con = file. read (2048) server Socket. send to (bytes (con, "utf-8"), print ("In Sent contents of", end: 2!!) print (sentence)
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og the new port of the server is ready to receive of the server from (20. Sentence, client Address = server Socket. recv from (20. Sentence: sentence, "r") con = file. read (2048) Server Socket. send to Chytes (con, "ref-8"), print ("In Sent contents of "end="!") print ("In Sent contents of "end="!") # for i in sentence
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
fra sea	ruer UDP. py: om socket import * rver Port = 12000 og 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

All P

	DOMS Page
	Date / /
	Output:
	Hom see held imported
\rightarrow	when you run serven UDP. py
_	The server is ready to receive the
	san socket blid (("1270.0) , since Per i)
_ <i>→</i>	when you run Client UDP. py (1500) AT I Diling
	Enter fite name: Server UDP. py
(3426)	Robbin Com. Cercion . 3 - 200 do A Cres do . Des 182 100
	(The folis from ServenUDP. py will be copied as
	(The folis from ServenUDP. py will be copied as displayed here)
	appropriate from the said (2004.)
→	In Servey UDP. py Steves transferences
1 234	The serven is ready to receive
	sul contents of server UDP. py 122 AP DARAGE
	The server is ready to receive a made and living
	12 v2 t2 v2 t v0 t t
	(" stores (Suppl Modific
_	file desct)
	The second secon

5. Tool Exploration - Wireshark

DOM5 Page No. 5. Wire shark: of affaill outs oran wireshork captures network packets from rasious interfaces st enables us to captures specific type of traffic bound proto cals, sounce and deslination adobress and all keywords within packet payload. It's real time monitoring capability is invaluable for observing ongoing network activities. This feature helps in diluting sudden traffic spikes and musual protocol vihaciona : Tr & babe un water : 10 101 \$4 5n command prompt: Type config survivos > ipconfig Ethernet adapter Ethernet: would be supplied with connection specific DNS suffix: Link weal IPV address ---: fe 80: e587: e9 fol : 4edb 1/2 Ipv4 address : 10.124.2.84 Bulonet mask ---- : 255.255.0.0 Default gateway . - 10.129.0.11

	45	DOM5	Page No.
		Date /	1
			10
	St shows us Tp V6, TP V4 add	uses. Submet m	ask
	- I would gate way.		
	The extentoroto col appears	at the bollow	with
N2	source and destination add	usses and all	key words
	written packet payload.	day build publ	ovi to the
15,	week to be a pocket from	es oud out to	שלבלינה
	when user clicks on these k	ywords for ex	ample:
	destination address, the dest		
13-17	be highlighted.	errette evente	do vol
Jak	be highlighted.	in delecting us	Refer
	For ex: when we geled a To		
	Source port: 5228		
	Destination port: 58545	e cemprand m	5 15
	Sequence number; 1 Sequence number; (raw): 42	4738453	SNIGH
	Acknowlegement Number: 1		
	Acknowlegement Numbers: 2	nit adoption fire	13.20
	1. Due 5140 11		
Fort		usus seve a	A Side
	: Uselb Y 2		
	140.120.284	ordinaris	Jev.
	U D 884 PAGE	La gradu	relati
	(1 37, 11 of .	- morning the	Mi. /