

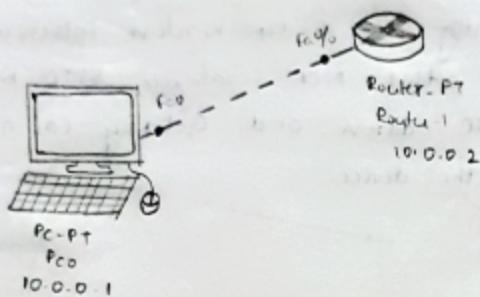
PROGRAM -10

To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

18/8/23

Aim : To understand the operation of TELNET by accessing the router in server room from PC in IT office.

Topology:



Procedure:

- ① Create the topology as above
- ② Give IP address of PC as 10.0.0.1 & config the router ip address as 10.0.0.2
- ③ Give the Gateway as 10.0.0.2 in the PC
- ④ Now, in the Router give the following commands to create hostname, password.

```
Router>enable  
Router# config  
Router(Config)# hostname r1  
r1(config)# enable secret 1  
r1(config)# interface fastethernet 0/0  
r1(config-if)# ip address 10.0.0.1 255.0.0.0  
r1(config-if)# no shut  
r1(config-if)# line vty 0 5  
r1(config-line)# login
```

T1 (config-line) # password po

T1 (config-line) # exit

exit (config) # exit

⑤ Command wr is used to save changes in router

Output:-

Ping 10.0.0.2

Pc > ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 : bytes = 32 time = TTL =

Reply from 10.0.0.2 : bytes = 32 time = TTL =

Reply from 10.0.0.2 : bytes = 32 time = TTL =

Reply from 10.0.0.2 : bytes = 32 time = TTL =

Ping statistics for 10.0.0.2:

Packet: Sent = 4 , Received = 4 , Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0 , Maximum = , Average =

Pc > telnet 10.0.0.2

Trying 10.0.0.2 ... open

User Access Verification

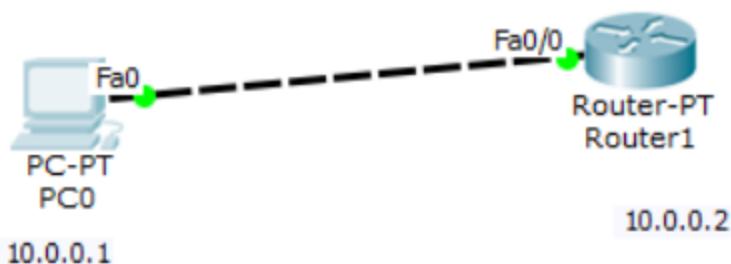
10/10 Password: po

T1 > enable

✓ Password: pi

22/8/23 T1# show ip route

Codes : C - connected , S - static, I - IGRP, R - RIP, M - mobile



The screenshot shows a "Command Prompt" window from the Packet Tracer software. The window title is "PC0". The tabs at the top are "Physical", "Config", "Desktop", and "Custom Interface". The main area displays the following command-line session:

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

Password:
rl>enable
Password:
rl#|
```

CYCLE - 2

Write a program for error detecting code using CRC_CCITT(16 bits)

18/8/23 CYCLE 2

Aim : Write a program for error detecting code using CRC-CCITT (16 bits)

```
# include <stdio.h>
# include <conio.h>

char m[50], g[50], r[50], q[50], temp[50];
void caltrans(int);
void crc(int);
void caltrans();
void shift();
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 + i] = '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[8] = g[12] = '1'; g[1] = g[5] = g[9] = g[13] = '0';
    printf("\n generator : %s\n", g);
    crc(n);
    printf("\n quotient : %s", q);
    caltrans(n);
}
```

```

printf("In transmitted frame : %s", m);
printf("In Enter transmitted frame:");
scanf("In %s", m);
printf(" CRC checking");
crc(n);
printf("In last remainder : %s", r);
for (i=0; i<16; i++)
    if (r[i] != '0')
        flag = 0;
    else
        continue;
    if (flag == 1)
        printf(" Error during transmission ");
    else
        printf(" Received frame 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';
            calc();
        }
    }
}

```

```

        for(j=0; j <= 17; j++)
            temp[j] = r[j];
    }
    q[n-16] = '\0';
}

void colram()
{
    int i, j;
    for(i=1; i <= 16; i++)
    {
        r[i-1] = ((int)temp[i]-48) ^ ((int)q[i]-48)+48;
    }
}

void shift()
{
    int i;
    for(i=1; i <= 16; i++)
        r[i-1] = r[i];
}

void colram(int n)
{
    int i, k=0;
    for(i=n-16; i < n; i++)
        m[i] = ((int)m[i]-48) ^ ((int)r[k+i]-48)+48;
    m[i] = '\0';
}

```

Output:

Enter frame bits : 1011

Message after appending 16 zeros : 1011 0000 0000 0000 0000

Generator : 10001000000100001

Quotient : 1011

```
Enter the frame bits:1011
Message after appending 16 zeros:10110000000000000000
generator:1000100000100001

quotient:1011
transmitted frame:10111011000101101011
Enter transmitted frame:10111011000101101011
CRC checking

last remainder:0000000000000000

Received frame is correct
Process returned 0 (0x0)    execution time : 14.468 s
Press any key to continue.
```

```
Enter the frame bits:1001
Message after appending 16 zeros:10010000000000000000
generator:1000100000100001

quotient:1001
transmitted frame:10011001000100101001
Enter transmitted frame:10011001000000101001
CRC checking

last remainder:000000100000000Error during transmission
Process returned 0 (0x0)    execution time : 19.597 s
Press any key to continue.
```

Write a program for congestion control using leaky bucket algorithm

18/8/23

```
Write a program for congestion control using leaky  
bucket algorithm

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

return 0;
}

Output:

Enter bucket size, outgoing rate & no. of inputs.

20 10 2

Enter the incoming packet size : 30

Incoming packet size 30

Dropped 10 no. of packets

Bucket buffer size 0 out of 20

After outgoing 10 packets left out 20 in
buffer.

Enter the incoming packet size : 10

Incoming packet size 10

Bucket buffer size 10 out of 20

After outgoing 10 packets left out of 20 in buffer

10/10
N
22/8/23

```
Enter bucket size, outgoing rate and no of inputs: 20 10 2
Enter the incoming packet size : 30
Incoming packet size 30
Dropped 10 no of packets
Bucket buffer size 0 out of 20
After outgoing 10 packets left out of 20 in buffer
Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size 20 out of 20
After outgoing 10 packets left out of 20 in buffer

Process returned 0 (0x0)  execution time : 22.003 s
Press any key to continue.
```