

Lab Assignment 1

Network and Communication

Name: Kulvir Singh

Reg. No.: 19BCE2074

Question 1:

Study about Networking Commands

Output Screenshots :

a)ifconfig

ifconfig (interface configurator) is one of the most basic and commonly used commands for finding network details. It is also used to configure network interface parameters.

```
kulvir06@KV06:~$ ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.80.1 netmask 255.255.255.0 broadcast 192.168.80.255
    inet6 fe80::69b4:7997:7f5a:b35e prefixlen 64 scopeid 0xfd<compat,link,site,host>
    ether 00:50:56:c0:00:01 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

b)ip address

ip address command can be used to show all IP Addresses of all network interfaces.

```
kulvir06@KV06:~$ ip address
11: eth0: <> mtu 1500 group default qlen 1
    link/ether d0:c5:d3:3f:3f:d4
    inet 169.254.19.55/16 brd 169.254.255.255 scope global dynamic
        valid_lft forever preferred_lft forever
    inet6 fe80::9869:b963:b73f:1337/64 scope link dynamic
        valid_lft forever preferred_lft forever
24: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1
    link/ether 00:50:56:c0:00:01
    inet 192.168.80.1/24 brd 192.168.80.255 scope global dynamic
        valid_lft 1032sec preferred_lft 1032sec
    inet6 fe80::69b4:7997:7f5a:b35e/64 scope link dynamic
        valid_lft forever preferred_lft forever
```

c)ping

ping (Packet Internet Groper) is another most commonly used network command to check connectivity between two network nodes. It is used to measure the average response.

```
kulvir06@KV06:~$ ping google.com
PING google.com(bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e)) 56 data bytes
64 bytes from bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e): icmp_seq=7 ttl=117 time=83.4 ms
64 bytes from bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e): icmp_seq=9 ttl=117 time=78.7 ms
64 bytes from bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e): icmp_seq=12 ttl=117 time=86.9 ms
64 bytes from bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e): icmp_seq=15 ttl=117 time=79.4 ms
64 bytes from bom07s30-in-x0e.1e100.net (2404:6800:4009:820::200e): icmp_seq=16 ttl=117 time=92.1 ms
```

d)netstat

netstat command used to review each network connection and open sockets on the Linux device. It provides connections, open sockets, routing tables information.

```
kulvir06@KV06:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type                   State                  I-Node   Path
kulvir06@KV06:~$
```

e)nslookup

nslookup (Name Server Lookup) command used to query DNS to get a domain name, IP address mapping, or DNS records.

```
kulvir06@KV06:~$ nslookup google.com
Server:          192.168.43.1
Address:         192.168.43.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.183.14
Name:   google.com
Address: 2404:6800:4009:820::200e
```

f)dig

dig (Domain Information Groper) is another command used to investigate DNS. It is an updated version of nslookup. It performs a DNS Lookup query and displays the response returned from name servers. It is also used to verify DNS mappings, MX records, and other DNS records.

kulvir06@KV06: ~

kulvir06@KV06:~\$ dig

```
; <<>> DiG 9.16.1-Ubuntu <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8489
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
; .                IN      NS

;; ANSWER SECTION:
.                518400 IN      NS      m.root-servers.net.
.                518400 IN      NS      l.root-servers.net.
.                518400 IN      NS      d.root-servers.net.
.                518400 IN      NS      c.root-servers.net.
.                518400 IN      NS      i.root-servers.net.
.                518400 IN      NS      b.root-servers.net.
.                518400 IN      NS      a.root-servers.net.
.                518400 IN      NS      f.root-servers.net.
.                518400 IN      NS      e.root-servers.net.
.                518400 IN      NS      h.root-servers.net.
.                518400 IN      NS      j.root-servers.net.
.                518400 IN      NS      k.root-servers.net.
.                518400 IN      NS      g.root-servers.net.
```

g)route

route command is used to shows or modifies the system's routing table. Using this command, you can troubleshoot the network issue caused by a wrong entry in the system routing table.

kulvir06@KV06:~\$ route

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.80.0	0.0.0.0	255.255.255.0	U	256	0	0	eth1
192.168.80.1	0.0.0.0	255.255.255.255	U	256	0	0	eth1
192.168.80.255	0.0.0.0	255.255.255.255	U	256	0	0	eth1
224.0.0.0	0.0.0.0	240.0.0.0	U	256	0	0	eth1
255.255.255.255	0.0.0.0	255.255.255.255	U	256	0	0	eth1
192.168.159.0	0.0.0.0	255.255.255.0	U	256	0	0	eth2
192.168.159.1	0.0.0.0	255.255.255.255	U	256	0	0	eth2
192.168.159.255	0.0.0.0	255.255.255.255	U	256	0	0	eth2
224.0.0.0	0.0.0.0	240.0.0.0	U	256	0	0	eth2
255.255.255.255	0.0.0.0	255.255.255.255	U	256	0	0	eth2
127.0.0.0	0.0.0.0	255.0.0.0	U	256	0	0	lo
127.0.0.1	0.0.0.0	255.255.255.255	U	256	0	0	lo

h)host

host command is used to display domain name for an IP address or IP address for a domain name. It can also be used to query DNS.

```
kulvir06@KV06:~$ host google.com
google.com has address 172.217.163.174
google.com has IPv6 address 2404:6800:4007:80f::200e
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 20 alt1.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 40 alt3.aspmx.l.google.com.
kulvir06@KV06:~$
```

i)iwconfig

iwconfig command is used to configure the WLAN interface. It can view or set up basic wireless network interface properties like SSID and encryption type.

```
kulvir06@KV06:~$ iwconfig
eth0      no wireless extensions.

eth1      no wireless extensions.

eth2      no wireless extensions.

lo        no wireless extensions.

eth3      no wireless extensions.

wifi0     no wireless extensions.

wifi1     no wireless extensions.

wifi2     no wireless extensions.
```

j)curl

curl is a utility used to transfer data to and from a server without user interaction. It can communicate using HTTP, HTTPS, FTP, SFTP, and SCP protocols. It can be used to upload or download data using any of the above protocols.

```
kulvir06@KV06:~$ curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

k)telnet

telnet command uses the Telnet protocol to communicate with the destination host. You must need to specify the host with port

```
kulvir06@KV06:~$ telnet fosslinux.com 433
Trying 2606:4700:3034::ac43:80b2...
```

l)whois

whois command is used to get all the information about a website. You can get all registration and ownership details using it. You need to install the whois package before using it.

```
kulvir06@KV06:~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
```

m)ifplugstatus

ifplugstatus command is used to check if the network cable is connected to the network interface. To use the command, you first need to install it on Ubuntu.

```
kulvir06@KV06:~$ ifplugstatus
eth0: link beat detected
eth1: link beat detected
eth2: link beat detected
lo: link beat detected
eth3: link beat detected
wifi0: link beat detected
wifi1: link beat detected
wifi2: link beat detected
```

n)nload

nload command is used to monitor your network bandwidth. It can show the total amount of data usage and min/max bandwidth usage. You need to install the nload package to run it.

```
Device eth0 [169.254.19.55] (1/8):
=====
Incoming:

Curr: 0.00 Bit/s
Avg: 0.00 Bit/s
Min: 0.00 Bit/s
Max: 0.00 Bit/s
Ttl: 0.00 Byte

Outgoing:

Curr: 0.00 Bit/s
Avg: 0.00 Bit/s
Min: 0.00 Bit/s
Max: 0.00 Bit/s
Ttl: 0.00 Byte
```

o)w

w command is used to get a list of currently logged in users on a system. It also provides valuable information like host, login time, idle time, JCPU.

```
kulvir06@KV06:~$ w
 21:43:37 up 34 min,  0 users,  load average: 0.52, 0.58, 0.59
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
kulvir06@KV06:~$
```

p)wget

wget is also a pre-installed package. It is used to download files using HTTP, HTTPS, FTP Protocols. It provides the ability to download multiple files, resume downloads, download in the background, etc.


```
kulvir06@KV06:~$ wget google.com
--2021-03-19 21:44:42-- http://google.com/
Resolving google.com (google.com)... 2404:6800:4009:820::200e, 142.250.183.14
Connecting to google.com (google.com)|2404:6800:4009:820::200e|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2021-03-19 21:45:00-- http://www.google.com/
Resolving www.google.com (www.google.com)... 2404:6800:4009:81a::2004, 142.250.77.36
Connecting to www.google.com (www.google.com)|2404:6800:4009:81a::2004|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html           [ <=> ] 14.83K --.-KB/s in 0.08s

2021-03-19 21:45:01 (194 KB/s) - 'index.html' saved [15191]
```

q)hostname

gives information about the hostname

```
 kulvir06@KV06: ~
kulvir06@KV06:~$ hostname
KV06
kulvir06@KV06:~$
```

Question 2:

Implement the following Error Detection and correction mechanism

- a) Checksum
- b) Parity Check
- c) CRC

Aim :

To find out the result after implementing error detection and error correction mechanism of checksum, parity check and crc mechanism.

Checksum

Algorithm :

The message is divided into 16-bit words. The value of the checksum word is set to 0. All words including the checksum are added using one's complement addition. The sum is complemented and becomes the checksum. The checksum is sent with the data. The message (including checksum) is divided into 16-bit words. All words are added using one's complement addition. The sum is complemented and becomes the new checksum. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

Code Text :

```
#include<iostream>
#include<string.h>

using namespace std;

int main()
{
    char a[20],b[20];
    char sum[20],complement[20];
    int i;
    cout<<"Kulvir Singh - 19BCE2074\n";
    cout<<"Enter first binary string\n";
    cin>>a;
    cout<<"Enter second binary string\n";
    cin>>b;

    if(strlen(a)==strlen(b))
    {
        char carry='0';
        int length=strlen(a);

        for(i=length-1;i>=0;i--)
```



```
{
  if(a[i]=='0' && b[i]=='0' && carry=='0')
  {
    sum[i]='0';
    carry='0';
  }
  else if(a[i]=='0' && b[i]=='0' && carry=='1')
  {
    sum[i]='1';
    carry='0';
  }
  else if(a[i]=='0' && b[i]=='1' && carry=='0')
  {
    sum[i]='1';
    carry='0';
  }
  else if(a[i]=='0' && b[i]=='1' && carry=='1')
  {
    sum[i]='0';
    carry='1';
  }
  else if(a[i]=='1' && b[i]=='0' && carry=='0')
  {
    sum[i]='1';
    carry='0';
  }
  else if(a[i]=='1' && b[i]=='0' && carry=='1')
  {
    sum[i]='0';
    carry='1';
  }
  else if(a[i]=='1' && b[i]=='1' && carry=='0')
  {
    sum[i]='0';
    carry='1';
  }
  else if(a[i]=='1' && b[i]=='1' && carry=='1')
  {
    sum[i]='1';
    carry='1';
  }
}
```

```
        else
            break;
    }
    cout<<"\nSum="<<carry<<sum;

    for(i=0;i<length;i++)
    {
        if(sum[i]=='0')
            complement[i]='1';
        else
            complement[i]='0';
    }

        if(carry=='1')
            carry='0';
        else
            carry='1';

        cout<<"\nChecksum="<<carry<<complement;
    }
    else
        cout<<"\nWrong input strings";

    return 0;
}
```

Code Screenshots :

```
1  #include<iostream>
2  #include<string.h>
3  using namespace std;
4  int main()
5  {
6      char a[20],b[20];
7      char sum[20],complement[20];
8      int i;
9
10     cout<<"Enter first binary string\n";
11     cin>>a;
12     cout<<"Enter second binary string\n";
13     cin>>b;
14
15     if(strlen(a)==strlen(b))
16     {
17         char carry='0';
18         int length=strlen(a);
19
20         for(i=length-1;i>=0;i--)
21         {
22             if(a[i]=='0' && b[i]=='0' && carry=='0')
23             {
24                 sum[i]='0';
25                 carry='0';
26             }
27             else if(a[i]=='0' && b[i]=='0' && carry=='1')
28             {
```

```
28     {
29         sum[i]='1';
30         carry='0';
31     }
32 }
33 else if(a[i]=='0' && b[i]=='1' && carry=='0')
34 {
35     sum[i]='1';
36     carry='0';
37 }
38 }
39 else if(a[i]=='0' && b[i]=='1' && carry=='1')
40 {
41     sum[i]='0';
42     carry='1';
43 }
44 }
45 else if(a[i]=='1' && b[i]=='0' && carry=='0')
46 {
47     sum[i]='1';
48     carry='0';
49 }
50 }
51 else if(a[i]=='1' && b[i]=='0' && carry=='1')
52 {
53     sum[i]='0';
54     carry='1';
```

```

56     }
57     else if(a[i]=='1' && b[i]=='1' && carry=='0')
58     {
59         sum[i]='0';
60         carry='1';
61     }
62     }
63     else if(a[i]=='1' && b[i]=='1' && carry=='1')
64     {
65         sum[i]='1';
66         carry='1';
67     }
68     }
69     else
70         break;
71 }
72 cout<<"\nSum="<<carry<<sum;
73
74 for(i=0;i<length;i++)
75 {
76     if(sum[i]=='0')
77         complement[i]='1';
78     else
79         complement[i]='0';
80 }
81
82 if(carry=='1')

```

```

83     carry='0';
84     else
85         carry='1';
86
87     cout<<"\nChecksum="<<carry<<complement;
88 }
89 else
90     cout<<"\nWrong input strings";
91
92 return 0;
93 }

```

Output Screenshots :

```
Kulvir Singh - 19BCE2074
Enter first binary string
101001
Enter second binary string
100111

Sum=1010000
Checksum=0101111
```

Parity Check

Algorithm :

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

1 is added to the block if it contains odd number of 1's, and

0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.

Code Text :

```
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{
int data[8];
int i=0,c=0;
printf("Kulvir Singh - 19BCE2074\n");
printf("Enter the 7 bits\n");
for(i=0;i<7;i++)
scanf("%d",&data[i]);
for(i=0;i<7;i++)
{
if(data[i]==1)
c=c+1;
}
for(i=0;i<7;i++)
printf("%d",data[i]);
if(c%2==0)
{
printf("\nEven parity\n");
data[7]=0;
}
else
{
printf("\nOdd parity\n");
data[7]=1;
}
```

```

}
printf("original data:");
for(i=0;i<7;i++)
printf("%d",data[i]);
srand(time(0));
int r=0,j=0;
for(j=0;j<5;j++)
r=rand()%((8-0)+1);
printf("\nBit changed randomly at position:%d\n",r+1);
if(data[r]==1)
data[r]=0;
else
data[r]=1;
printf("\nData after adding the parity bits:");
for(i=0;i<8;i++)
printf("%d",data[i]);
int cn=0;
for(i=0;i<8;i++)
{
if(data[i]==1)
cn=cn+1;
}
if(cn%2==0)
printf("\nno error");
else
printf("\nerror");
return 0;
}

```


Code Screenshots :

```
1  #include<iostream>
2  #include<stdio.h>
3  #include<stdlib.h>
4  #include<time.h>
5  int main()
6  {
7  int data[8];
8  int i=0,c=0;
9  printf("Enter the 7 bits\n");
10 for(i=0;i<7;i++)
11 scanf("%d",&data[i]);
12 for(i=0;i<7;i++)
13 {
14 if(data[i]==1)
15 c=c+1;
16 }
17 for(i=0;i<7;i++)
18 printf("%d",data[i]);
19 if(c%2==0)
20 {
21 printf("Even parity\n");
22 data[7]=0;
23 }
24 else
25 {
26 printf("\nOdd parity\n");
27 data[7]=1;
28 }
```

```
29  printf("original data:");
30  for(i=0;i<7;i++)
31  printf("%d",data[i]);
32  srand(time(0));
33  int r=0,j=0;
34  for(j=0;j<5;j++)
35  r=rand()%((8-0)+1);
36  printf("\nBit changed randomly at position:%d\n",r+1);
37  if(data[r]==1)
38  data[r]=0;
39  else
40  data[r]=1;
41  printf("\nData after adding the parity bits:");
42  for(i=0;i<8;i++)
43  printf("%d",data[i]);
44  int cn=0;
45  for(i=0;i<8;i++)
46  {
47  if(data[i]==1)
48  cn=cn+1;
49  }
50  if(cn%2==0)
51  printf("\nno error");
52  else
53  printf("\nerror");
54  return 0;
55  }
```

Output Screenshots :

```
Kulvir Singh - 19BCE2074
Enter the 7 bits
1
0
0
1
0
1
0
1001010
Odd parity
original data:1001010
Bit changed randomly at position:8

Data after adding the parity bits:10010100
error

...Program finished with exit code 0
Press ENTER to exit console. 
```

CRC (Cyclic Redundancy Check)

Algorithm :

The communicating parties agree upon the size of message, $M(x)$ and the generator polynomial, $G(x)$. If r is the order of $G(x)$, r bits are appended to the low order end of $M(x)$. This makes the block size bits, the value of which is $x^r M(x)$. The block $x^r M(x)$ is divided by $G(x)$ using modulo 2 division. The remainder after division is added to $x^r M(x)$ using modulo 2 addition. The result is the frame to be transmitted, $T(x)$. The encoding procedure makes exactly divisible by $G(x)$. The receiver divides the incoming data frame $T(x)$ unit by $G(x)$ using modulo 2 division. Mathematically, if $E(x)$ is the error, then modulo 2 division of $[M(x) + E(x)]$ by $G(x)$ is done. If there is no remainder, then it implies that $E(x)$. The data frame is accepted. A remainder indicates a non-zero value of $E(x)$, or in other words presence of an error. So, the data frame is rejected. The receiver may then send an erroneous acknowledgment back to the sender for retransmission.

Code Text :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main() {
    int i,j,keyl,msgl;
    char input[100], key[30],temp[30],q[100],rem[30],key1[30];
    printf("enter data: ");
    gets(input);
    printf("enter key: ");
    gets(key);
    keyl=strlen(key);
    msgl=strlen(input);
    strcpy(key1,key);
    for (i=0;i<keyl-1;i++) {
        input[msgl+i]='0';
    }
    for (i=0;i<keyl;i++)
        temp[i]=input[i];
    for (i=0;i<msgl;i++) {
        q[i]=temp[0];
        if(q[i]=='0')
            for (j=0;j<keyl;j++)
                key[j]='0'; else
```

```
for (j=0;j<keyl;j++)
key[j]=key1[j];
for (j=keyl-1;j>0;j--) {
if(temp[j]==key[j])
rem[j-1]='0'; else
rem[j-1]='1';
}
rem[keyl-1]=input[i+keyl];
strcpy(temp,rem);
}
strcpy(rem,temp);
printf("\nQuotient is ");
for (i=0;i<msgl;i++)
printf("%c",q[i]);
printf("\nRemainder is ");
for (i=0;i<keyl-1;i++)
printf("%c",rem[i]);
printf("\nFinal data is: ");
for (i=0;i<msgl;i++)
printf("%c",input[i]);
for (i=0;i<keyl-1;i++)
printf("%c",rem[i]);
getch();
}
```

Code Screenshots :

```
1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  void main() {
5  int i,j,keyl,msgl;
6  char input[100], key[30],temp[30],q[100],rem[30],key1[30];
7  printf("enter data: ");
8  gets(input);
9  printf("enter key: ");
10 gets(key);
11 keyl=strlen(key);
12 msgl=strlen(input);
13 strcpy(key1,key);
14 for (i=0;i<keyl-1;i++) {
15 input[msgl+i]='0';
16 }
17 for (i=0;i<keyl;i++)
18 temp[i]=input[i];
19 for (i=0;i<msgl;i++) {
20 q[i]=temp[0];
21 if(q[i]=='0')
22 for (j=0;j<keyl;j++)
23 key[j]='0'; else
24 for (j=0;j<keyl;j++)
25 key[j]=key1[j];
```

```
26     for (j=keyl-1;j>0;j--) {
27         if(temp[j]==key[j])
28             rem[j-1]='0'; else
29             rem[j-1]='1';
30     }
31     rem[keyl-1]=input[i+keyl];
32     strcpy(temp,rem);
33 }
34 strcpy(rem,temp);
35 printf("\nQuotient is ");
36 for (i=0;i<msgl;i++)
37     printf("%c",q[i]);
38 printf("\nRemainder is ");
39 for (i=0;i<keyl-1;i++)
40     printf("%c",rem[i]);
41 printf("\nFinal data is: ");
42 for (i=0;i<msgl;i++)
43     printf("%c",input[i]);
44 for (i=0;i<keyl-1;i++)
45     printf("%c",rem[i]);
46 getch();
47 }
48 |
```

Output Screenshots :

Creating Codeword

```
Kulvir Singh - 19BCE2074
enter data: 1010101010
enter key: 11001

Quotient is 1100010010
Remainder is 0010
Final data is: 10101010100010

...Program finished with exit code 0
Press ENTER to exit console.
```

Checking the error. No error as remainder is 0

```
Kulvir Singh - 19BCE2074
enter data: 10101010100010
enter key: 11001

Quotient is 11000100100000
Remainder is 0000
Final data is: 101010101000100000

...Program finished with exit code 0
Press ENTER to exit console.
```