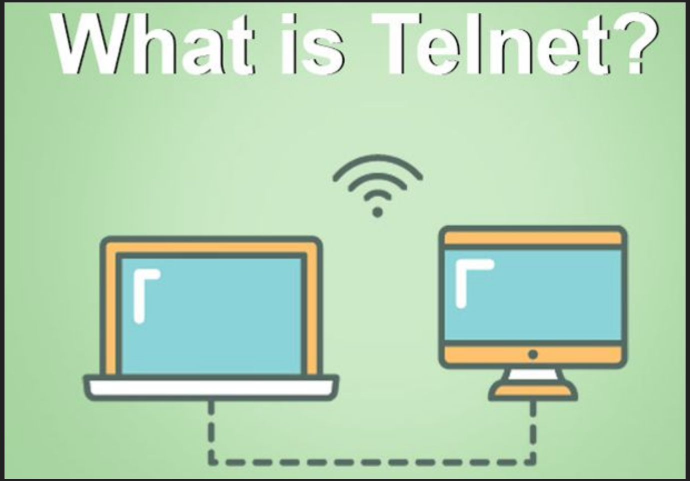
TELNET

BY:
Aditya Singh Mertia - IIT2022125
Rishabh Kumar - IIT2022131
Karan Singh - IIT2022132
Tejas Sharma - IIT2022161
Varun Naik - IIT2022220
Atharva Chavan - IIT2022221

Introduction - What is Telnet ?

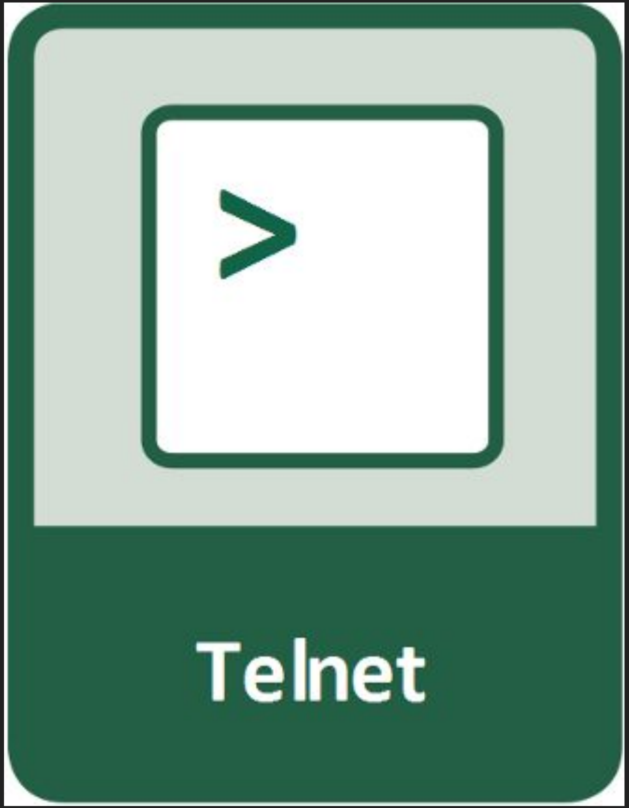
The full form of "TELNET" is "TELetype NETwork."

The term "Teletype" originated from the combination of "telegraph" and "typewriter."



TELNET vs telnet

- TELNET is a protocol that provides "a general, bi-directional, eight-bit byte oriented communications facility". TELNET is a Network Layer Protocol
- telnet is a program that supports the TELNET protocol over TCP. Telnet operates on the Application layer of OSI Model

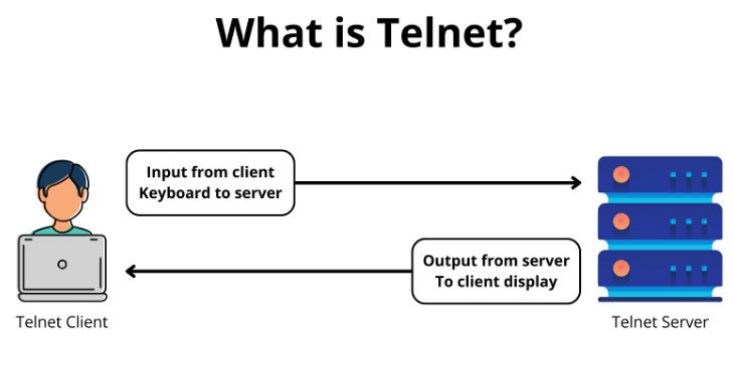


RFCs - Request for Comments

RFCs, or Request for Comments, are documents published by the Internet Engineering Task Force (IETF) that serve as standards for various Internet technologies. They provide detailed specifications for protocols and technologies, facilitating effective implementation by developers and engineers. For example, RFCs define the Telnet protocol used for remote login and terminal emulation (e.g., RFC 97, RFC 854, RFC 855).

History of Telnet

Telnet was developed in the early days of computer networking, with the first implementation appearing in 1969 as part of the ARPANET project (Advanced Research Projects Agency NET), which laid the foundation for the modern Internet. Originally designed for remote terminal access to Unix systems, Telnet quickly became a widely used protocol for accessing and managing remote systems.


What is Telnet?
Input from client Keyboard to server
Output from server To client display
Telnet Client
Telnet Server

Key Features of Telnet

Remote Login: Telnet enables users to log in to a remote system and access its command-line interface (CLI) as if they were physically present at the system's console

Text-Based Communication: Telnet provides a text-based interface for communication between the client and the server

Bidirectional Communication: Telnet supports bidirectional communication, allowing both the client and the server to send and receive text-based data

Portability: Telnet is platform-independent and can be used on a wide range of operating systems, including Unix, Linux, Windows, and macOS.

Remote Login

Remote login, also known as remote access, refers to the ability to connect to and interact with a computer system or network from a remote location. It enables users to access the resources and services of a remote system as if they were physically present at its location.

Logging Process

Local Login

- Local Login refers to accessing the user's own device directly.

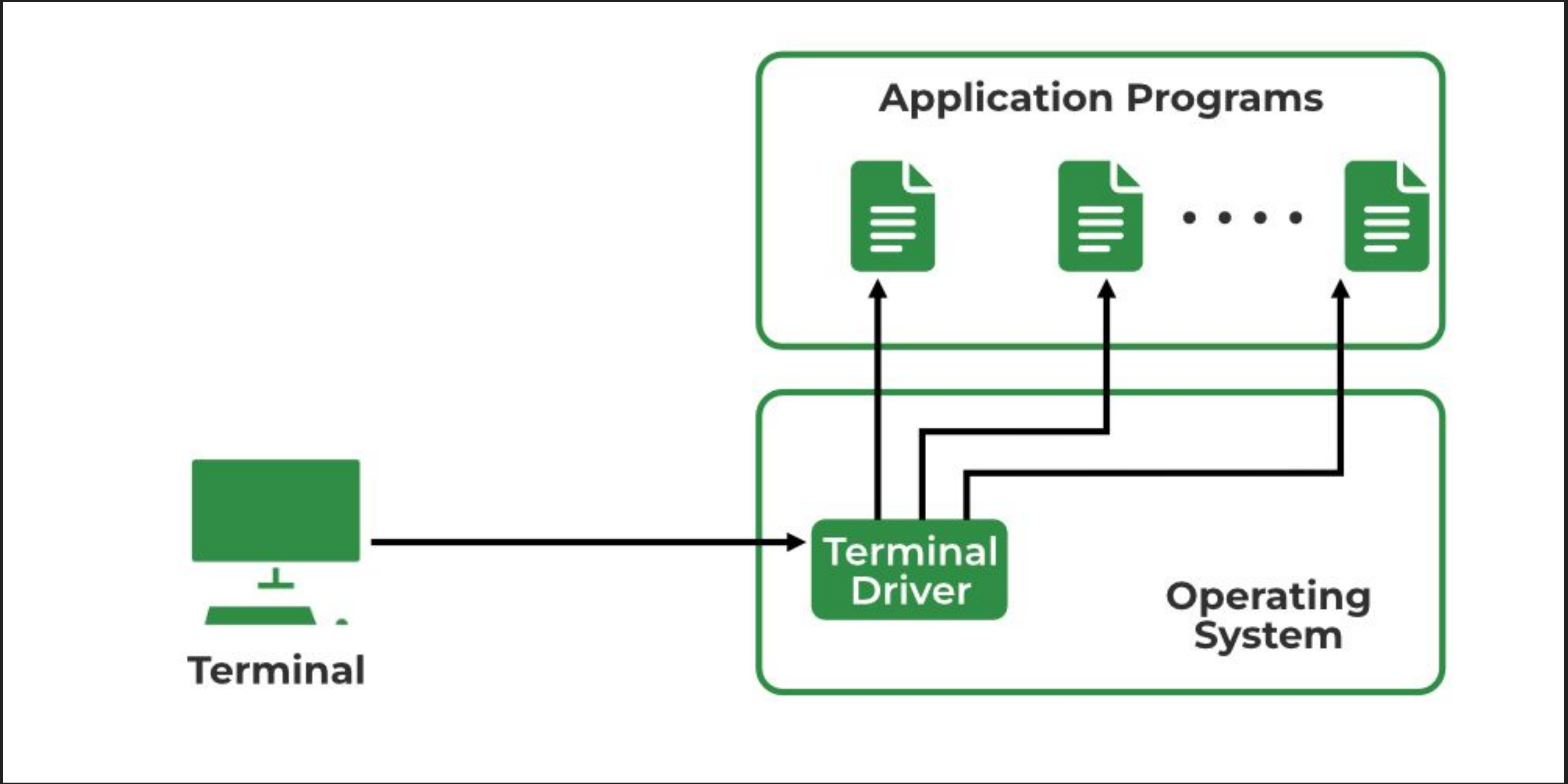
- When the user logs into a local timesharing system, it is called local log-in

Remote Login

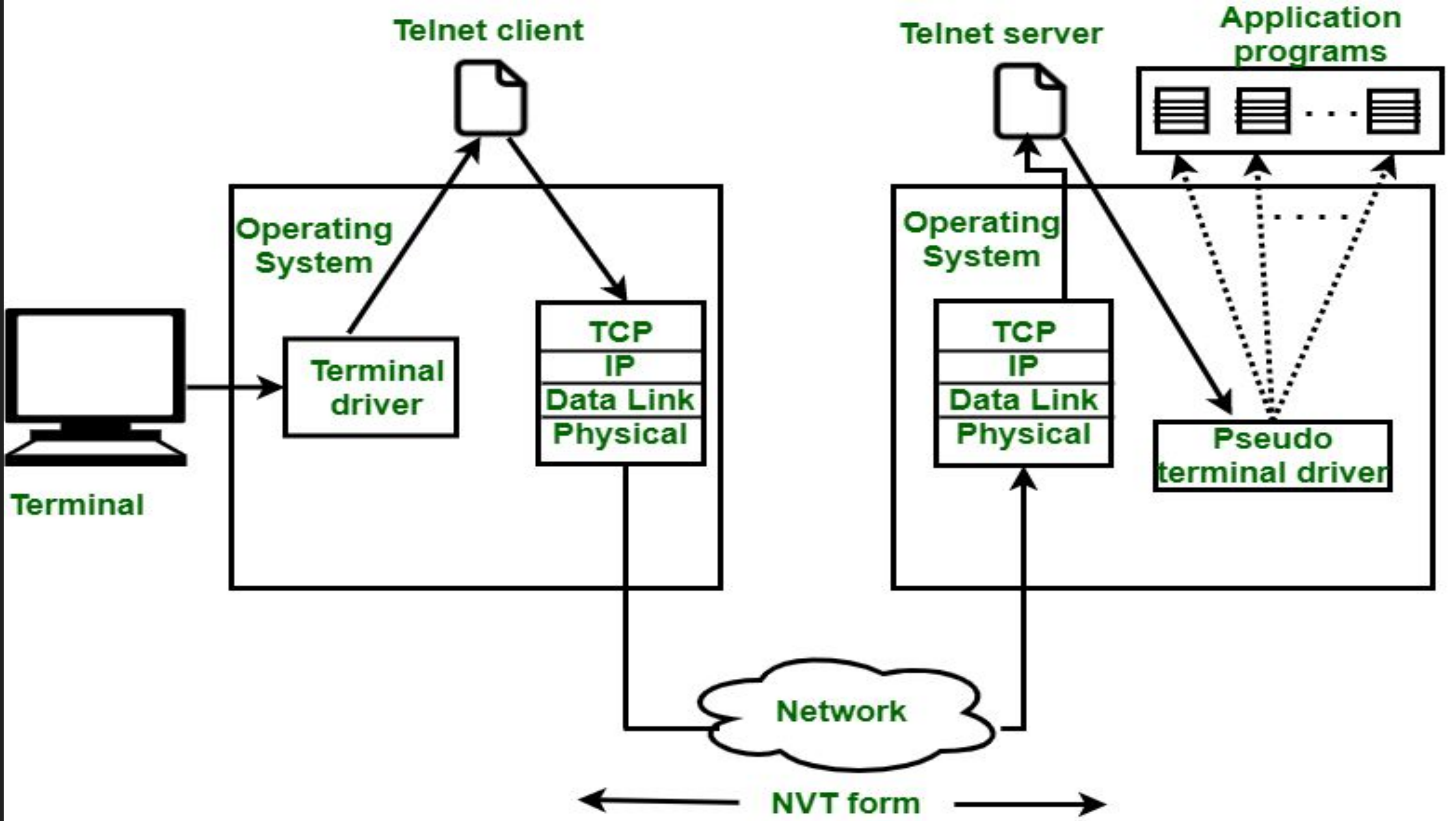
- Remote Login is a process in which users can log in to a remote site i.e. computer and use services that are available on the remote computer.

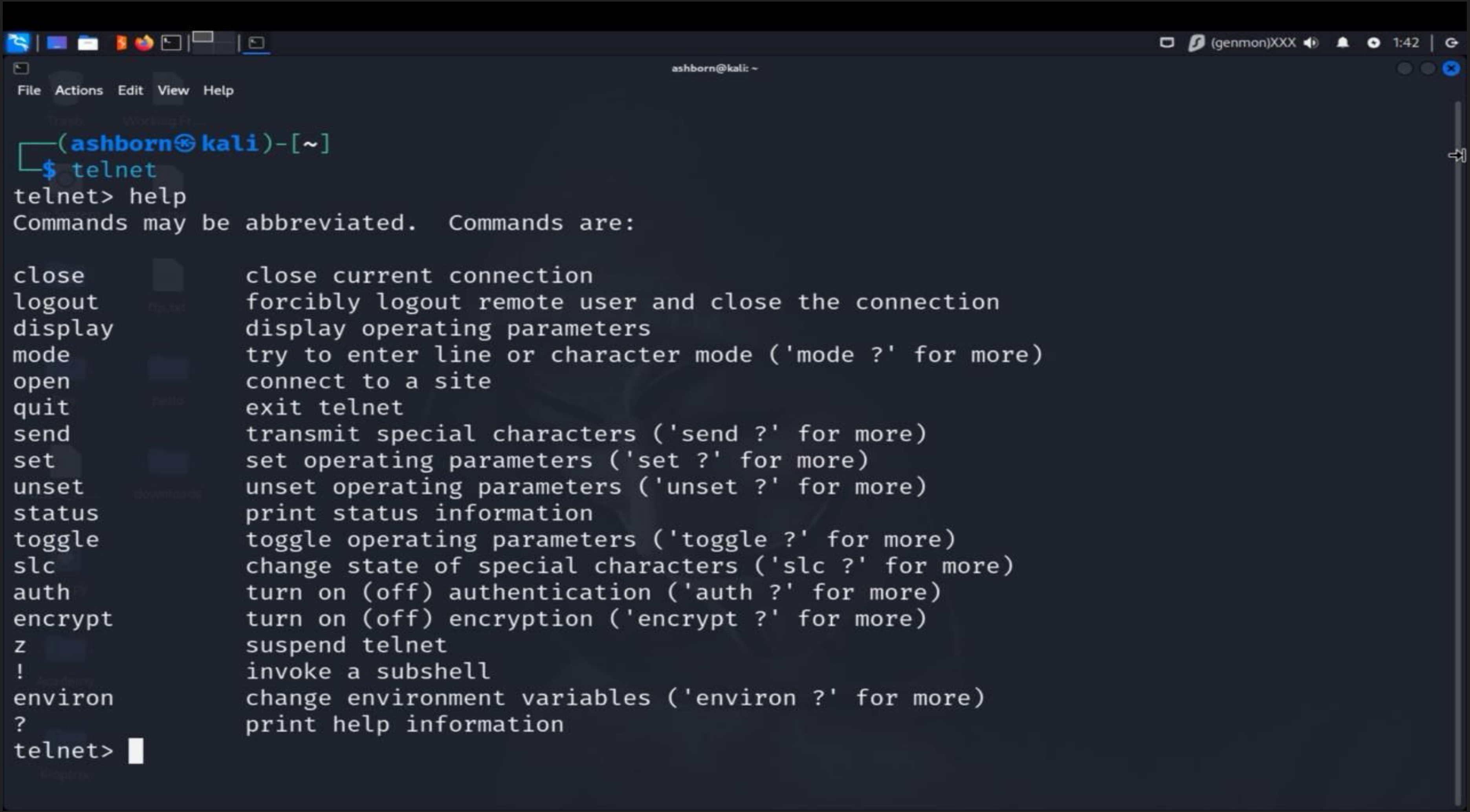
- When the user wants to access an application program or utility located on a remote machine, it is called remote log-in.

Local Login

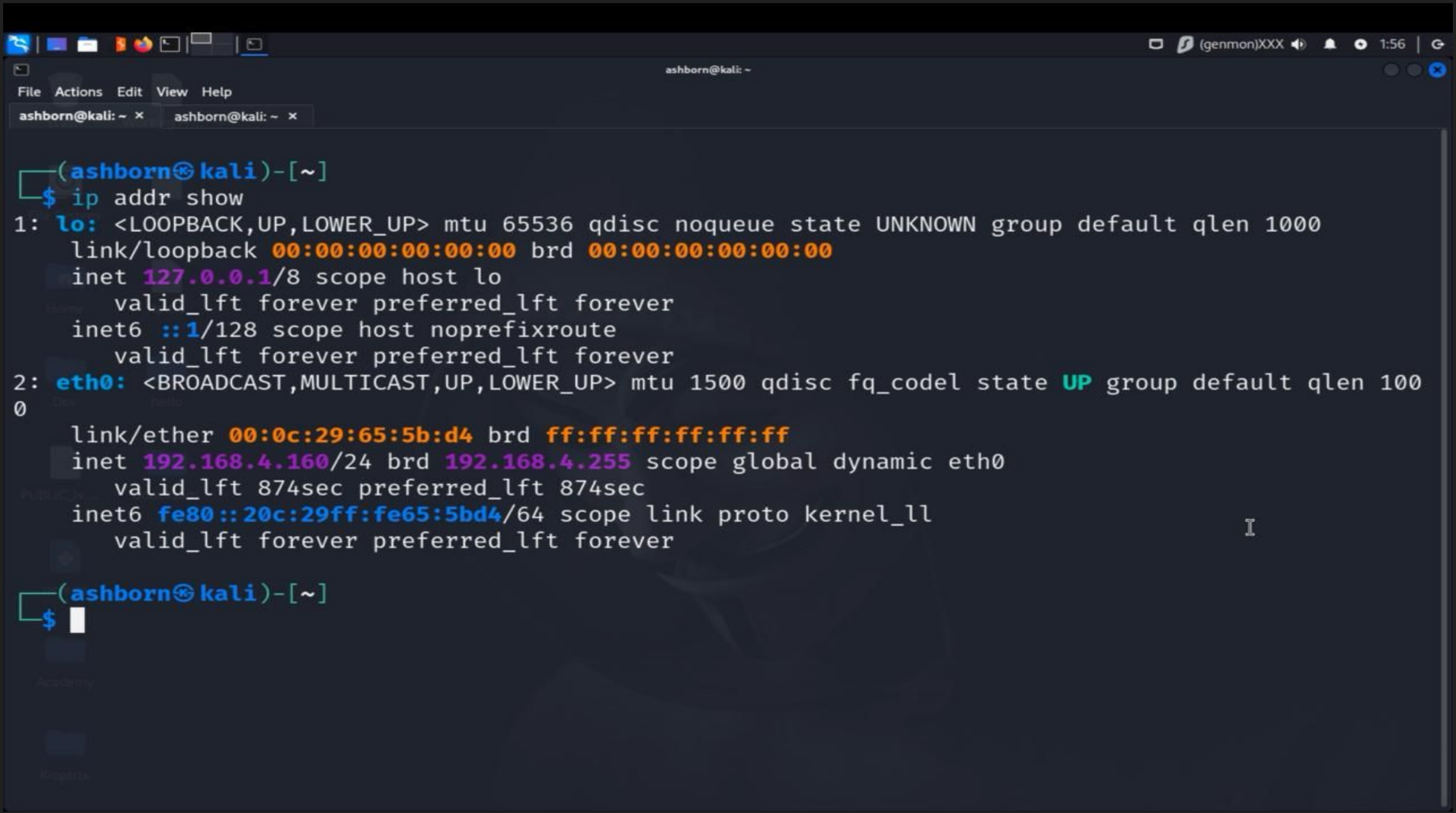

Application Programs
Terminal Driver
Operating System
Terminal

Remote Login

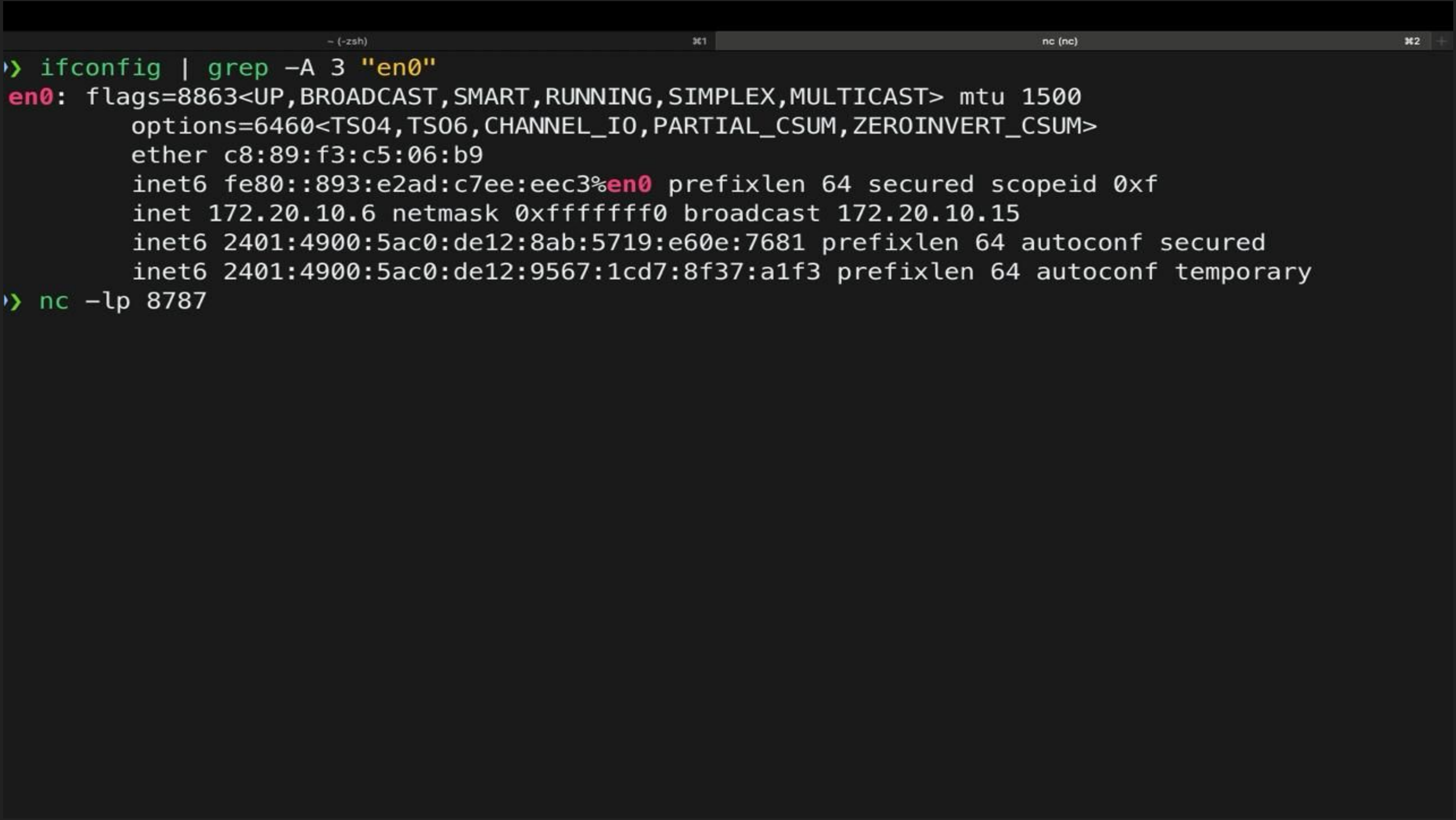




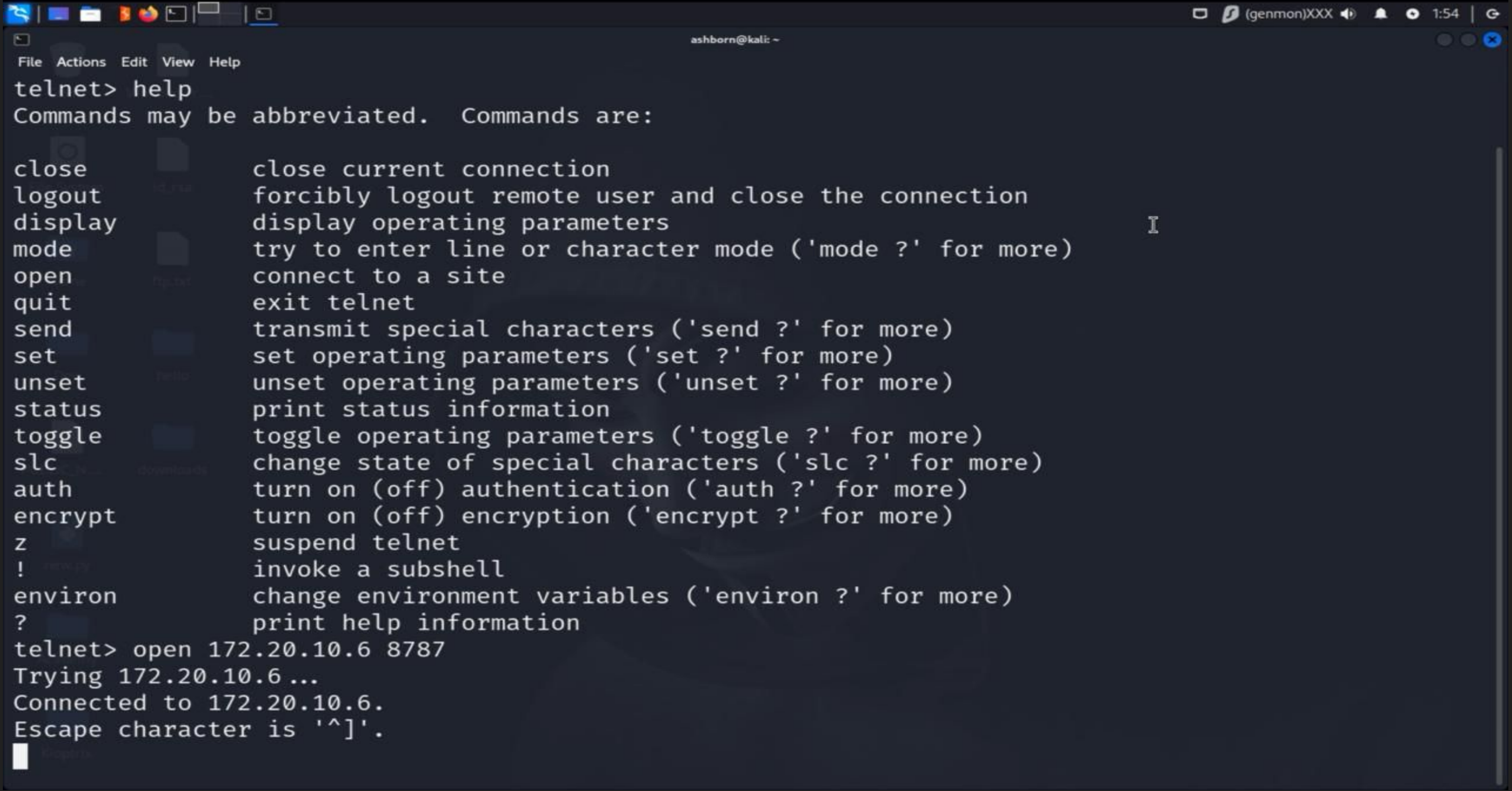
```
┌──(ashborn㉿kali)-[~]
└─$ telnet
telnet> help
Commands may be abbreviated.  Commands are:

close           close current connection
logout          forcibly logout remote user and close the connection
display         display operating parameters
mode            try to enter line or character mode ('mode ?' for more)
open            connect to a site
quit            exit telnet
send            transmit special characters ('send ?' for more)
set             set operating parameters ('set ?' for more)
unset           unset operating parameters ('unset ?' for more)
status          print status information
toggle          toggle operating parameters ('toggle ?' for more)
slc             change state of special characters ('slc ?' for more)
auth            turn on (off) authentication ('auth ?' for more)
encrypt         turn on (off) encryption ('encrypt ?' for more)
z               suspend telnet
!               invoke a subshell
environ         change environment variables ('environ ?' for more)
?               print help information
telnet> 
```



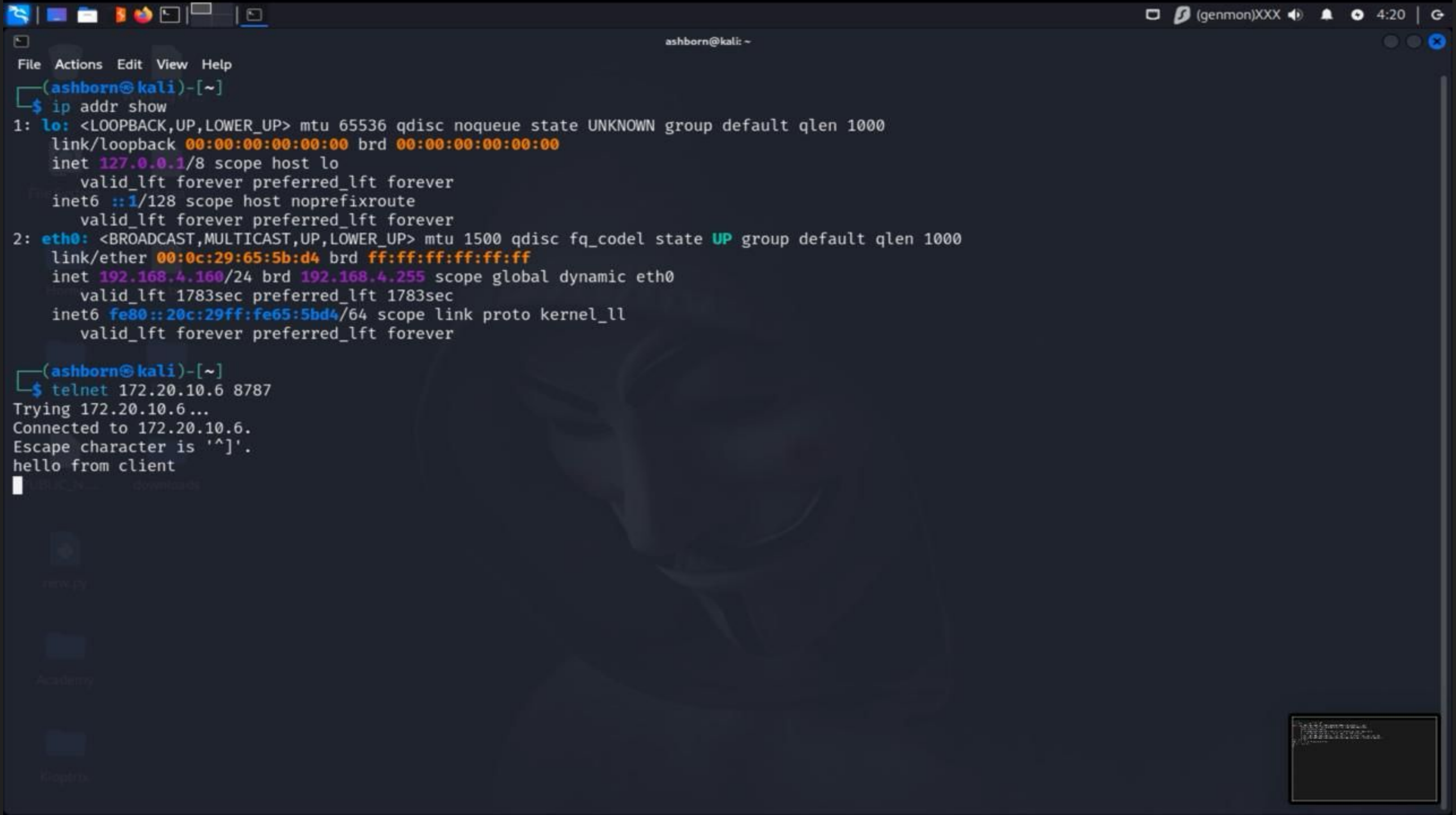
```
┌──(ashborn㉿kali)-[~]
└─$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 00:0c:29:65:5b:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.160/24 brd 192.168.4.255 scope global dynamic eth0
       valid_lft 874sec preferred_lft 874sec
    inet6 fe80::20c:29ff:fe65:5bd4/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever

┌──(ashborn㉿kali)-[~]
└─$ 
```



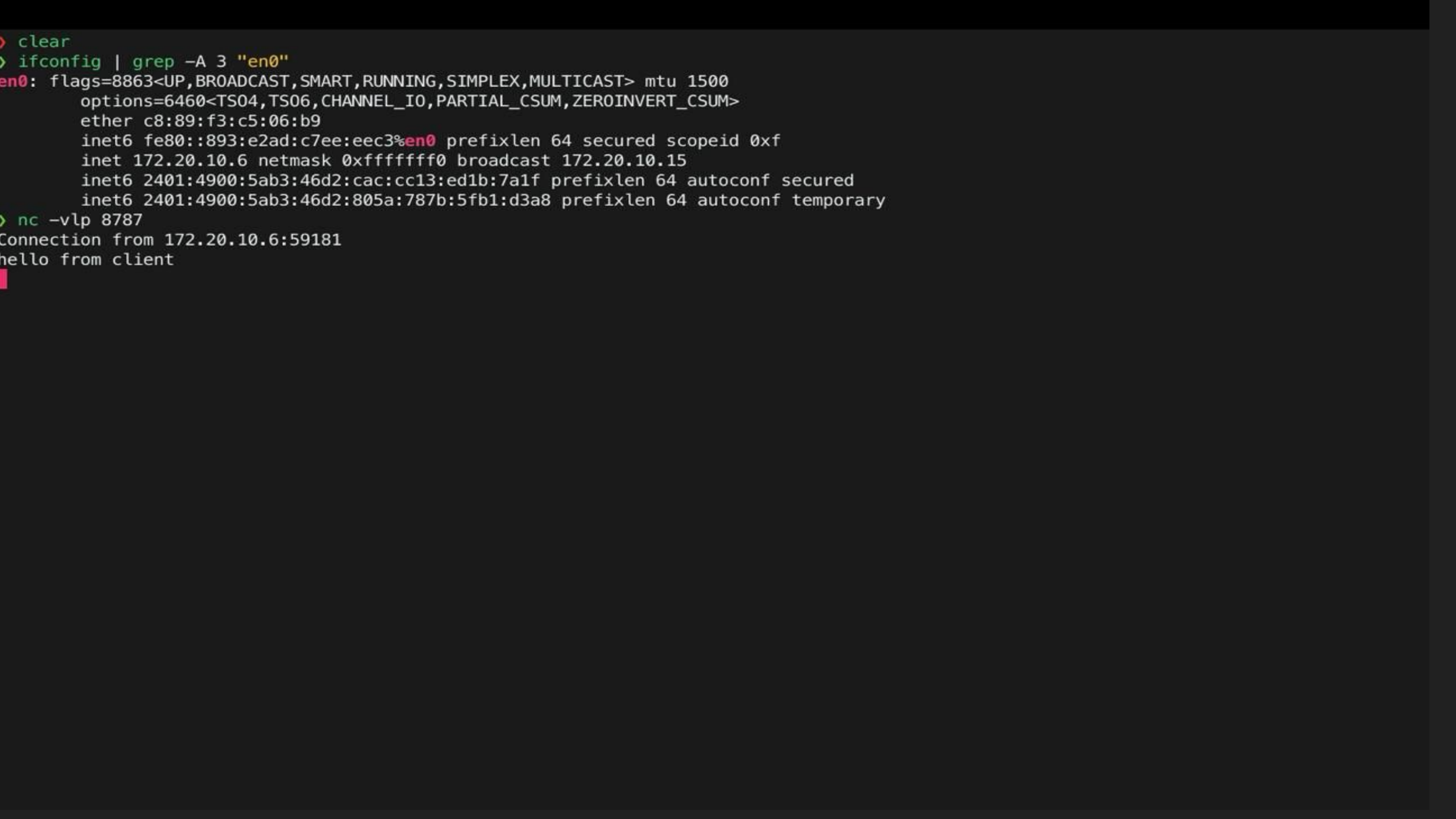
```
❯ ifconfig | grep -A 3 "en0"
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether c8:89:f3:c5:06:b9
        inet6 fe80::893:e2ad:c7ee:eec3%en0 prefixlen 64 secured scopeid 0xf
        inet 172.20.10.6 netmask 0xfffffff0 broadcast 172.20.10.15
        inet6 2401:4900:5ac0:de12:8ab:5719:e60e:7681 prefixlen 64 autoconf secured
        inet6 2401:4900:5ac0:de12:9567:1cd7:8f37:a1f3 prefixlen 64 autoconf temporary
❯ nc -lp 8787
```



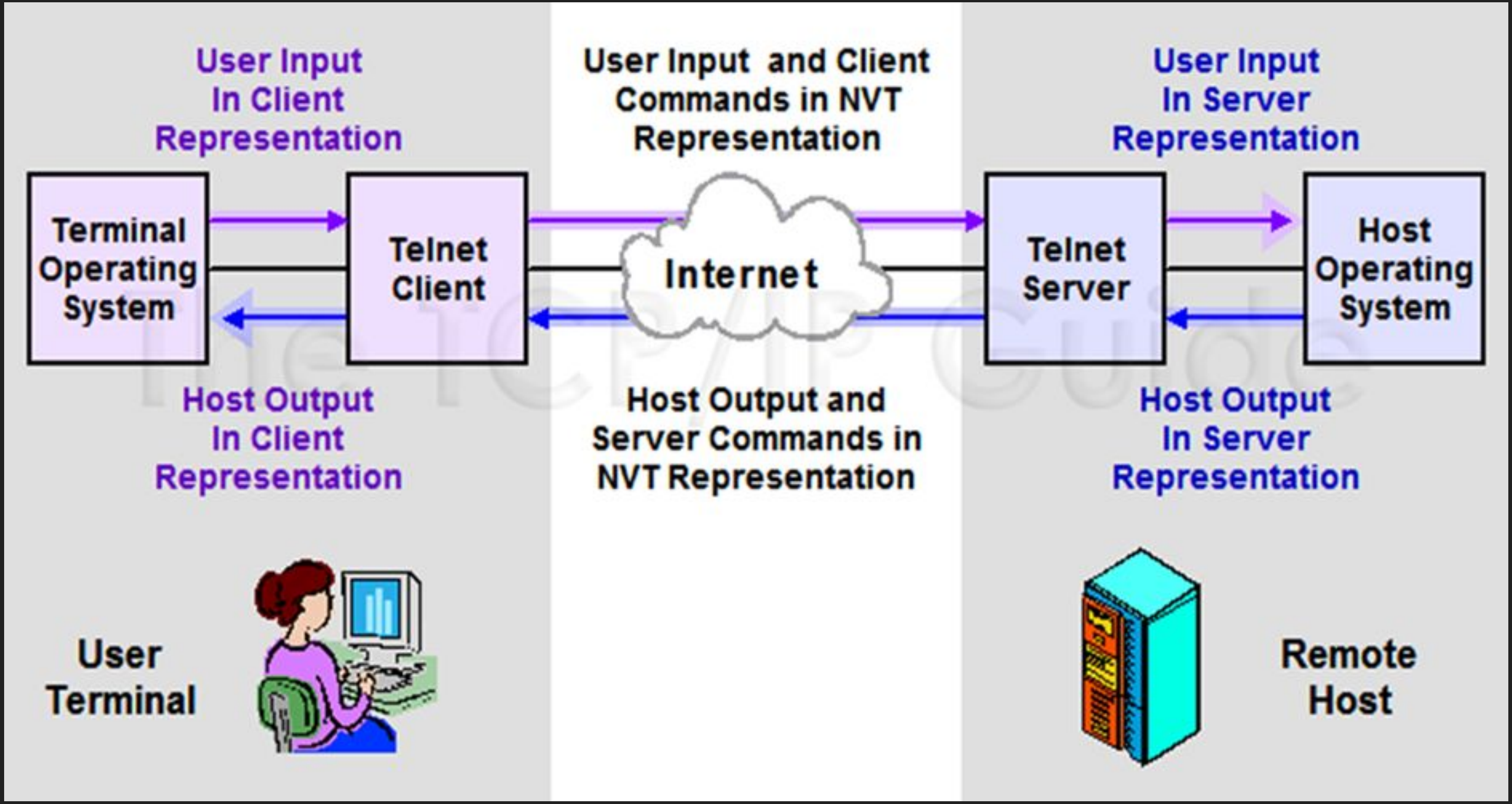
```
telnet> help
Commands may be abbreviated.  Commands are:

close           close current connection
logout          forcibly logout remote user and close the connection
display         display operating parameters
mode            try to enter line or character mode ('mode ?' for more)
open            connect to a site
quit            exit telnet
send            transmit special characters ('send ?' for more)
set             set operating parameters ('set ?' for more)
unset           unset operating parameters ('unset ?' for more)
status          print status information
toggle          toggle operating parameters ('toggle ?' for more)
slc             change state of special characters ('slc ?' for more)
auth            turn on (off) authentication ('auth ?' for more)
encrypt         turn on (off) encryption ('encrypt ?' for more)
z               suspend telnet
!               invoke a subshell
environ         change environment variables ('environ ?' for more)
?               print help information
telnet> open 172.20.10.6 8787
Trying 172.20.10.6...
Connected to 172.20.10.6.
Escape character is '^]'.
```



```
┌──(ashborn㉿kali)-[~]
└─$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:65:5b:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.160/24 brd 192.168.4.255 scope global dynamic eth0
       valid_lft 1783sec preferred_lft 1783sec
    inet6 fe80::20c:29ff:fe65:5bd4/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever

┌──(ashborn㉿kali)-[~]
└─$ telnet 172.20.10.6 8787
Trying 172.20.10.6...
Connected to 172.20.10.6.
Escape character is '^]'.
hello from client
```



```
❯ clear
❯ ifconfig | grep -A 3 "en0"
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether c8:89:f3:c5:06:b9
        inet6 fe80::893:e2ad:c7ee:eec3%en0 prefixlen 64 secured scopeid 0xf
        inet 172.20.10.6 netmask 0xfffffff0 broadcast 172.20.10.15
        inet6 2401:4900:5ab3:46d2:cac:cc13:ed1b:7a1f prefixlen 64 autoconf secured
        inet6 2401:4900:5ab3:46d2:805a:787b:5fb1:d3a8 prefixlen 64 autoconf temporary
❯ nc -vlp 8787
Connection from 172.20.10.6:59181
hello from client
```


User Input In Client Representation
User Input and Client Commands in NVT Representation
User Input In Server Representation
Terminal Operating System
Telnet Client
Internet
Telnet Server
Host Operating System
Host Output In Client Representation
Host Output and Server Commands in NVT Representation
Host Output In Server Representation
User Terminal
Remote Host

Problem due to Heterogeneous Systems

What if the remote host has different operating system than that of the client?

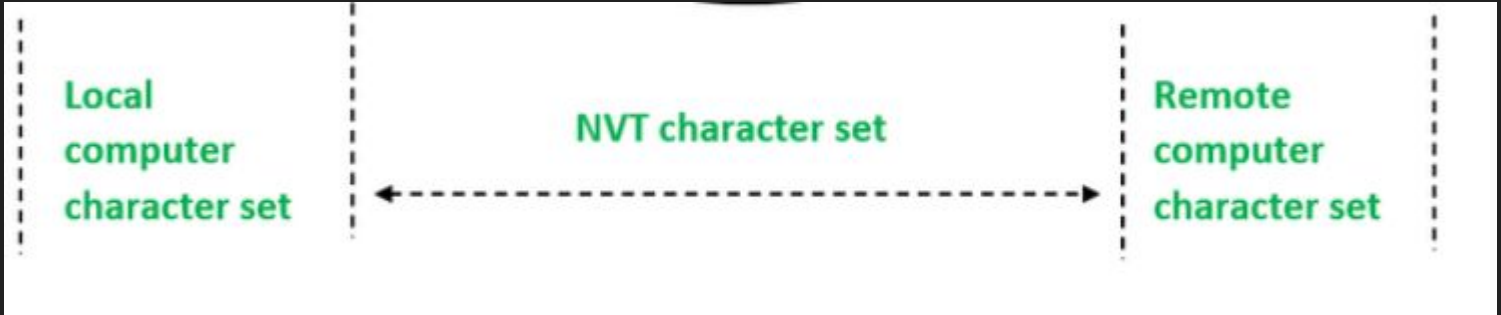
So we need something that will standardize the representation of characters and control functions cross platforms.

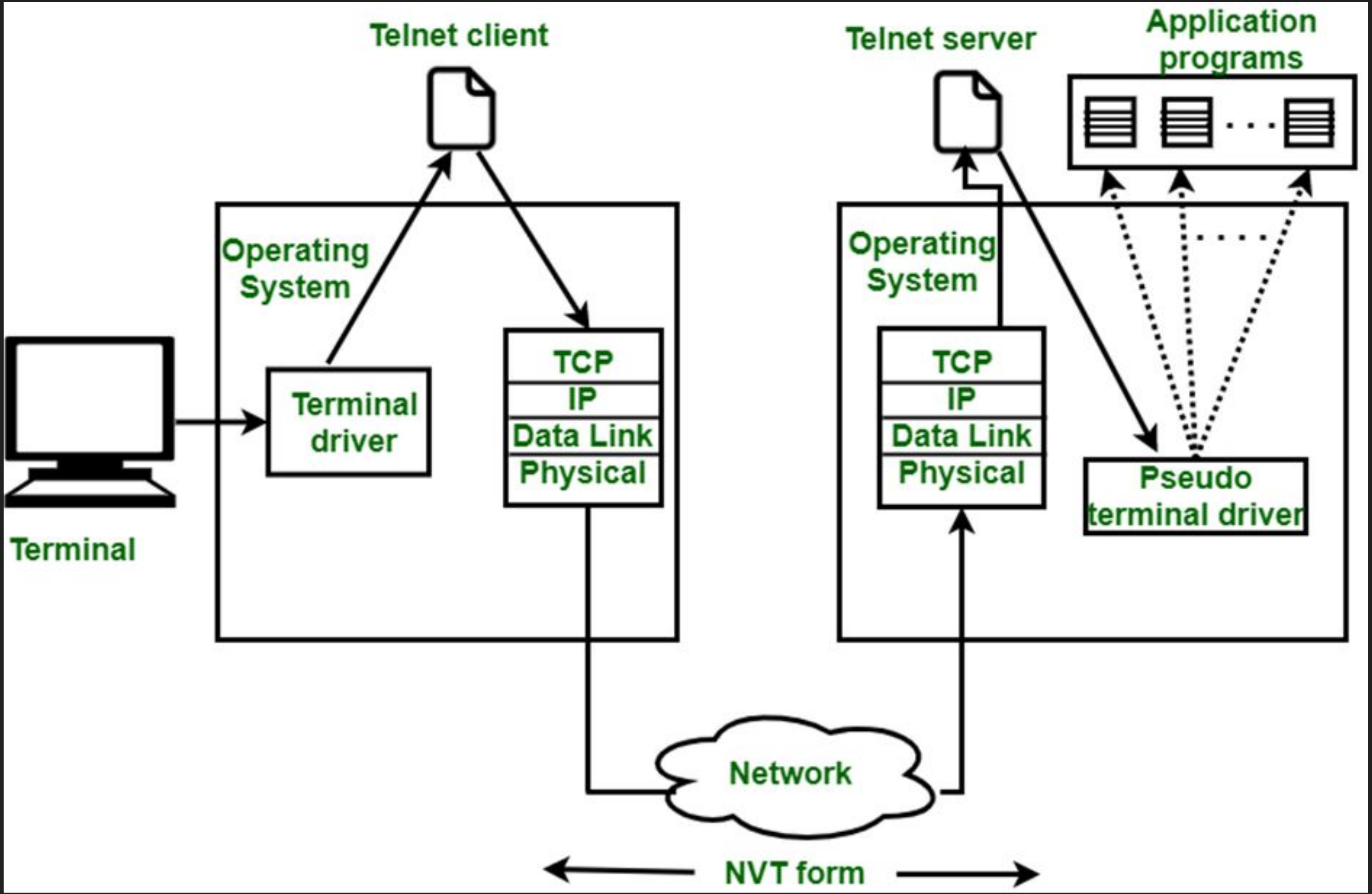
Therefore, we need a virtual terminal to do the above.

NVT (Network Virtual Terminal)

NVT (Network Virtual Terminal) is a virtual terminal in TELNET that has a fundamental structure that is shared by many different types of real terminals.

TELNET clients and servers send data, commands, and server output using a virtual terminal type called NVT




Telnet client
Telnet server
Application programs
Operating System
Operating System
Terminal
Terminal driver
TCP
IP
Data Link
Physical
TCP
IP
Data Link
Physical
Pseudo terminal driver
Network
NVT form

How Does NVT Work?

It works as follows and is also represented in the above figure:

- Local terminal character → NVT format and then send.
-
- NVT-formatted data and commands → Remote Computer’s character set

The Standarized Character Set

Network Virtual Terminal (NVT) Character Set:

The Network Virtual Terminal (NVT) primarily employs two sets of characters: one for data and another for control.

- 8-bit character set for data = 7 lowest-order bits (identical to ASCII) + highest bit set to 0.
-
- 8-bit bit character set to communicate control characters = 7 lowest-order bits (Commands) + highest-order bit set to 1.

NVT ASCII CODES

NULL , 0

```
echo -e "This is a null character: \x00End of message"
```

LINE FEED , 10

```
echo -e "Line 1\nLine 2\nLine 3\n"
```

Carriage Return , 13

```
echo -e “Line 1\rLine 2\rLine 3”
```

BACKSPACE , 8

```
echo -e "Hello\b\b\bWorld"
```

VERTICAL TAB , 11

```
echo -e "Line 1\vLine 2\vLine 3\v"
```

FORM FEED , 12

```
echo -e "Page 1\fPage 2\fPage 3\f"
```

END OF LINE MARKER

```
echo -e "This is line 1\r\nThis is line 2\r\nThis is line 3\r\n"
```

Negotiation Option

Command Structure - Syntax and format

Client Action

WILL : WILL <option>

WONT : WONT <option>

Server Action

DO : DO <option>

DONT : DONT <option>

Negotiated Options EXAMPLE

- Client sends: `IAC WILL ECHO`
- Server responds: `IAC DO ECHO`
- Client and server agree: `ECHO`

Modes Of Operation

Character Mode

Line Mode

Character Set

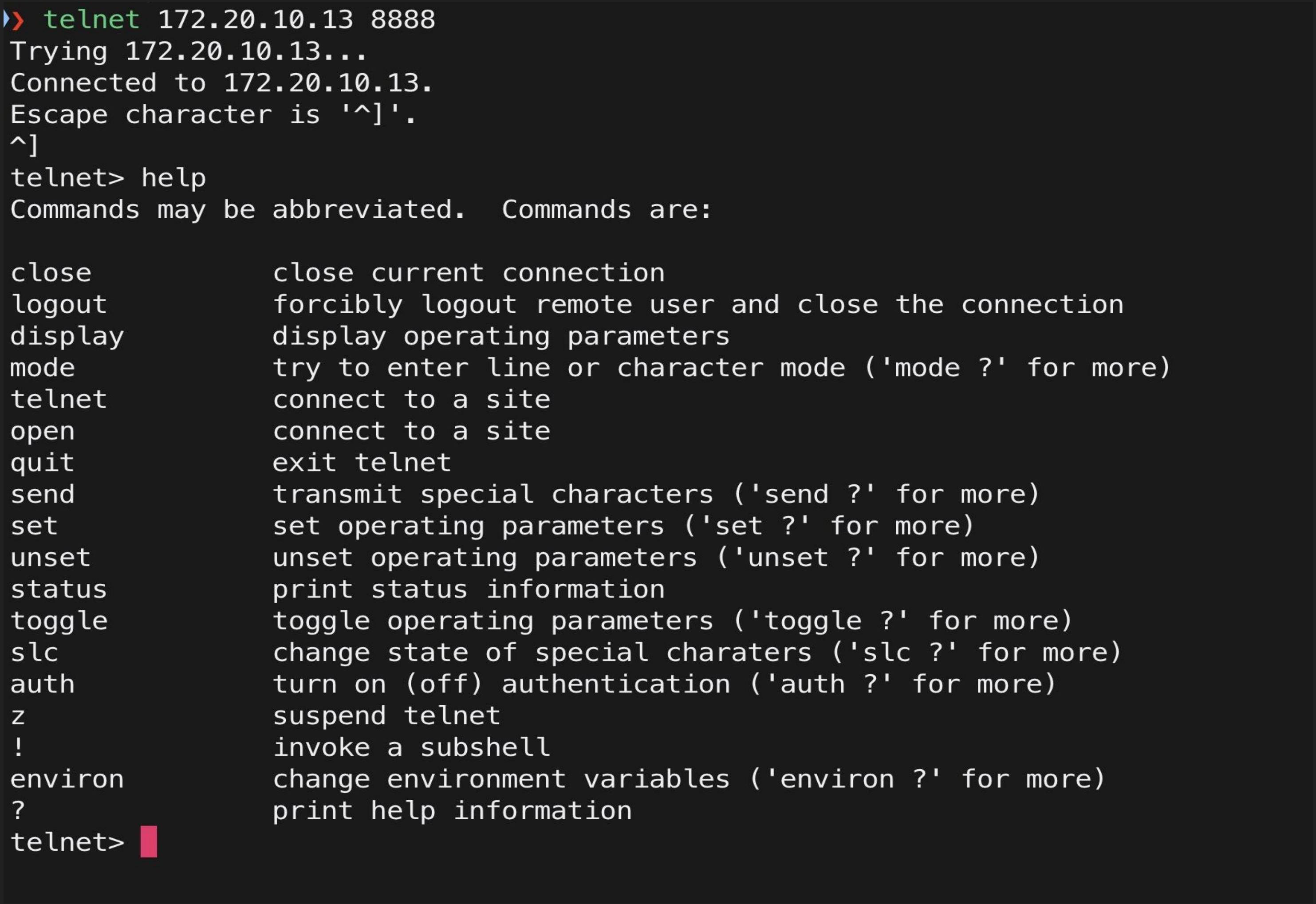
>>ASCII

>>EBCDIC

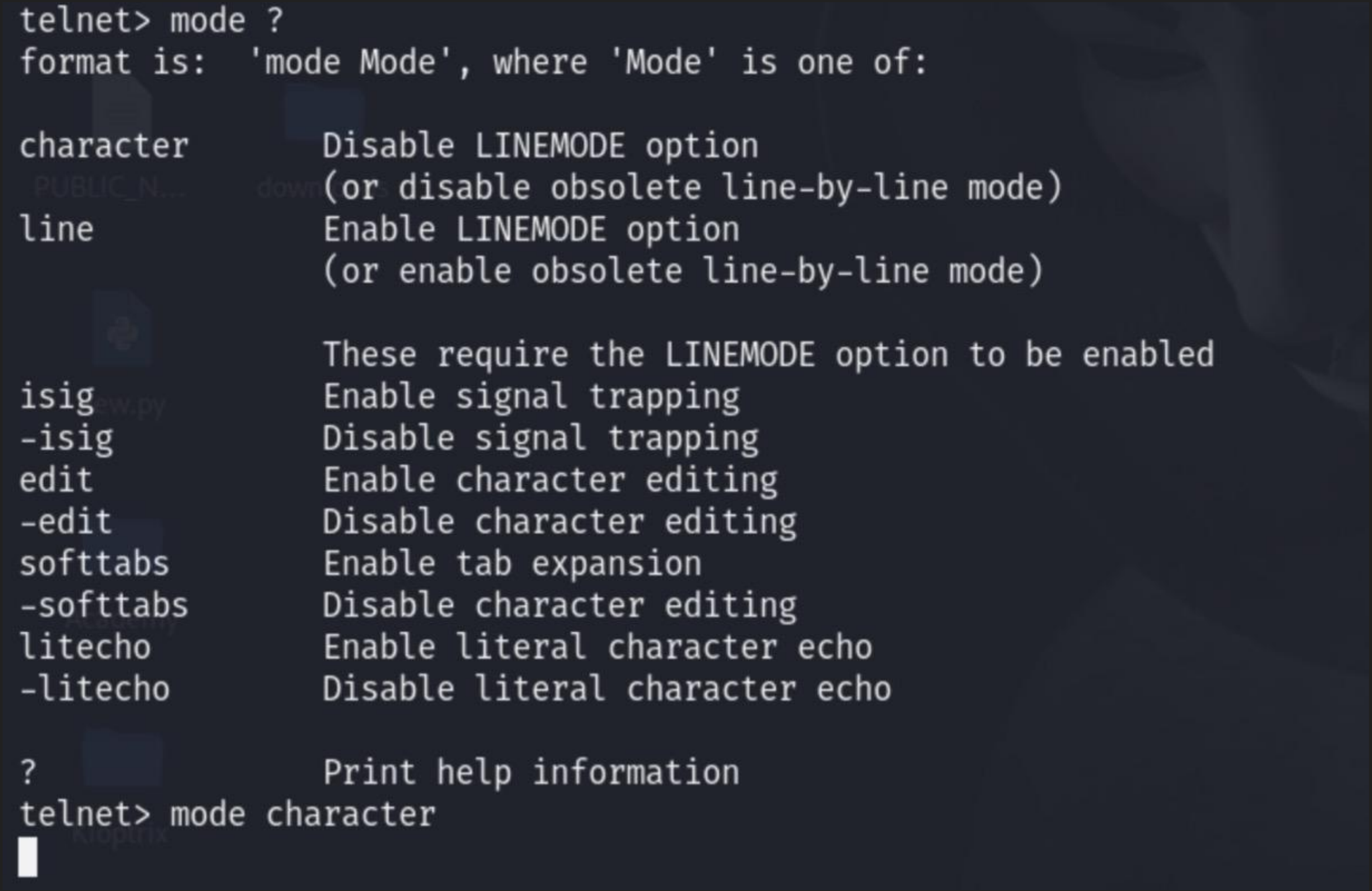
Echo Mode

>>Local Echo

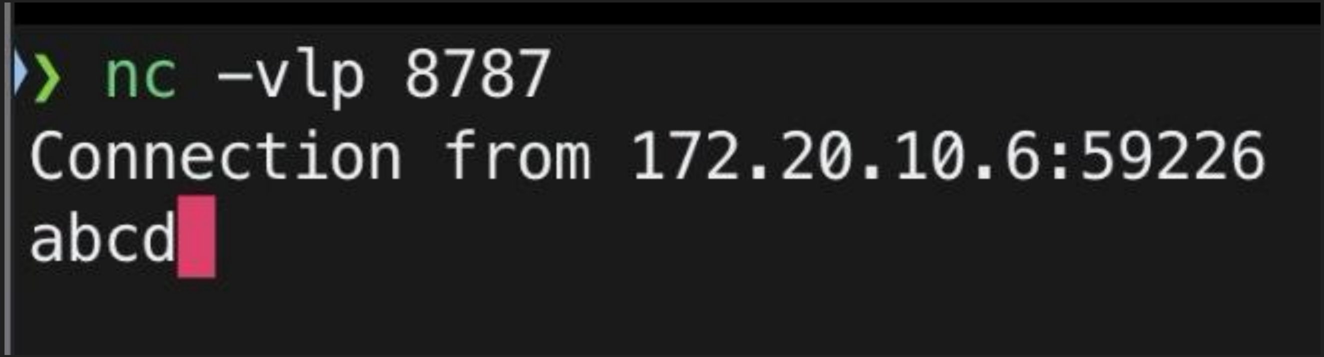
>>Remote Echo



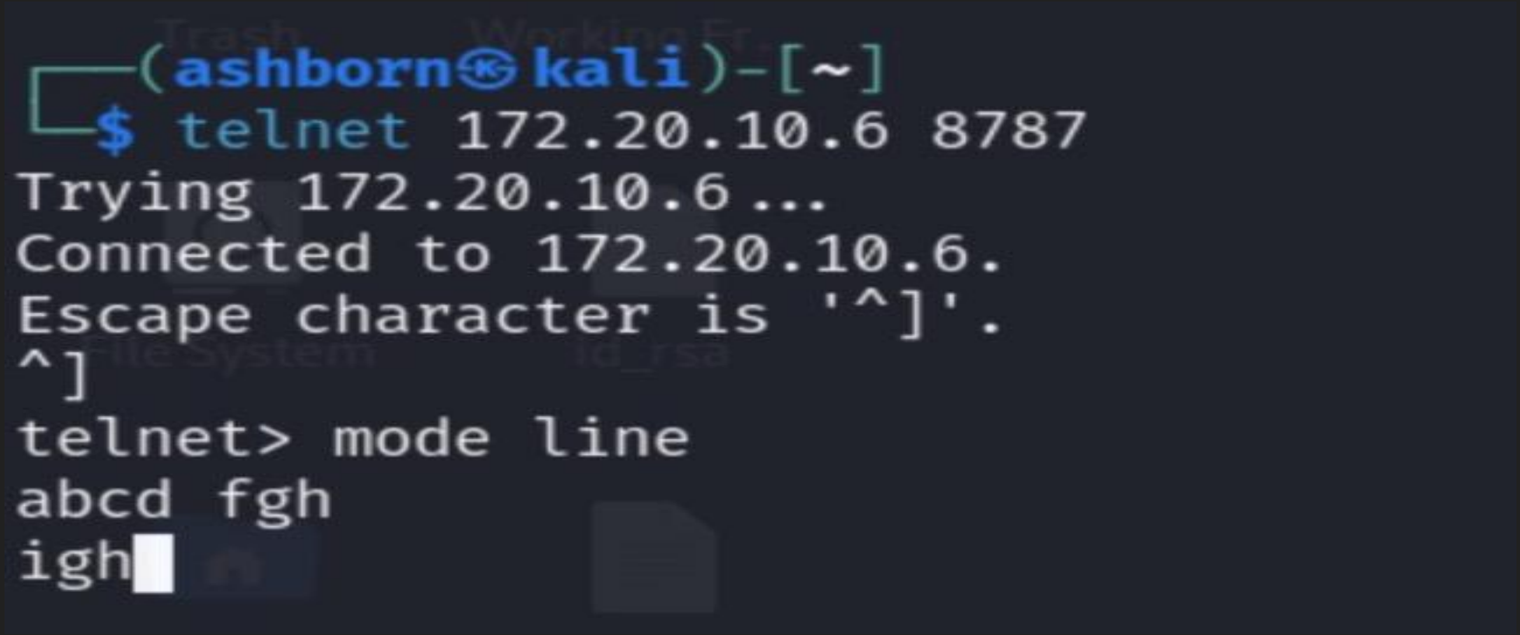
```
❯ telnet 172.20.10.13 8888
Trying 172.20.10.13...
Connected to 172.20.10.13.
Escape character is '^]'.
^]
telnet> help
Commands may be abbreviated.  Commands are:

close           close current connection
logout          forcibly logout remote user and close the connection
display         display operating parameters
mode            try to enter line or character mode ('mode ?' for more)
telnet          connect to a site
open            connect to a site
quit            exit telnet
send            transmit special characters ('send ?' for more)
set             set operating parameters ('set ?' for more)
unset           unset operating parameters ('unset ?' for more)
status          print status information
toggle          toggle operating parameters ('toggle ?' for more)
slc             change state of special charaters ('slc ?' for more)
auth            turn on (off) authentication ('auth ?' for more)
z               suspend telnet
!               invoke a subshell
environ         change environment variables ('environ ?' for more)
?               print help information
telnet> 
```



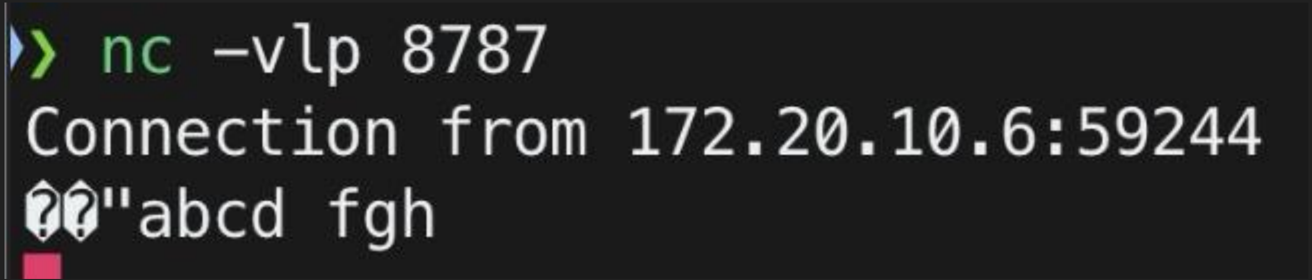
```
telnet> mode ?
format is:  'mode Mode', where 'Mode' is one of:

character       Disable LINEMODE option
                (or disable obsolete line-by-line mode)
line            Enable LINEMODE option
                (or enable obsolete line-by-line mode)

                These require the LINEMODE option to be enabled
isig            Enable signal trapping
-isig           Disable signal trapping
edit            Enable character editing
-edit           Disable character editing
softtabs        Enable tab expansion
-softtabs       Disable character editing
litecho         Enable literal character echo
-litecho        Disable literal character echo

?               Print help information
telnet> mode character
```



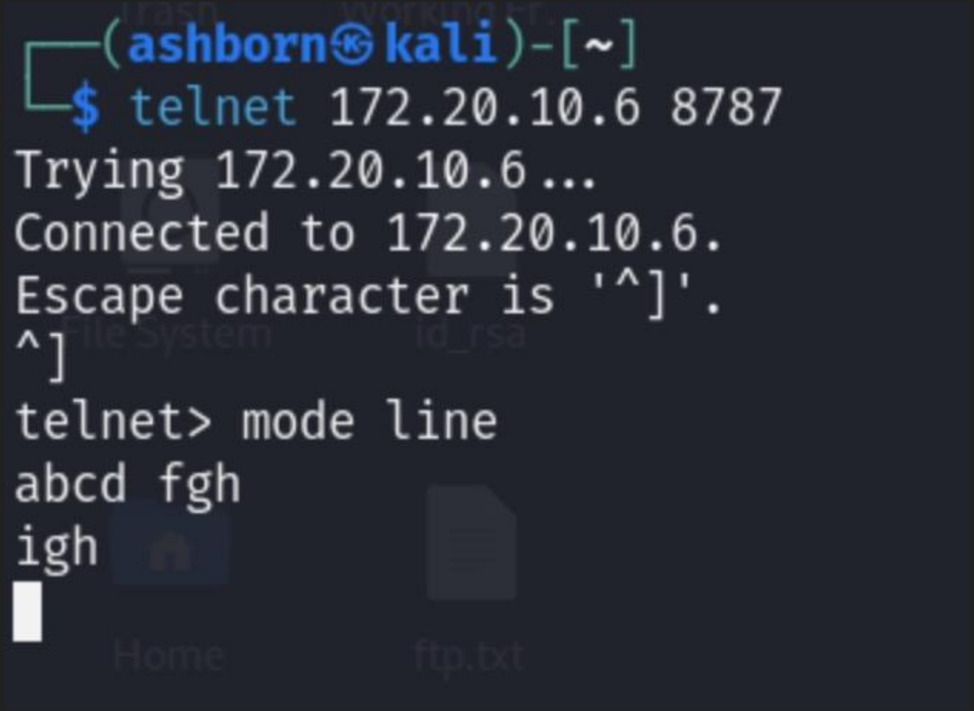
```
❯ nc -vlp 8787
Connection from 172.20.10.6:59226
abcd
```



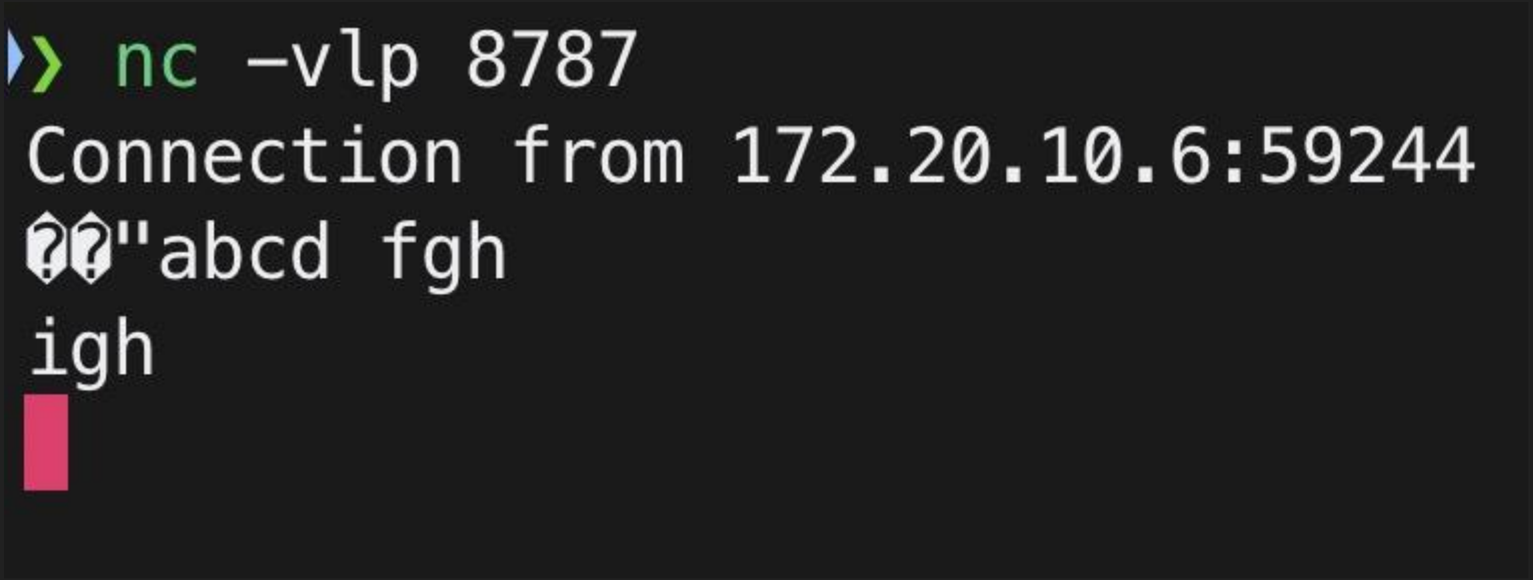
```
┌──(ashborn㉿kali)-[~]
└─$ telnet 172.20.10.6 8787
Trying 172.20.10.6...
Connected to 172.20.10.6.
Escape character is '^]'.
^]
telnet> mode line
abcd fgh
igh
```



```
❯❯ nc -vlp 8787
Connection from 172.20.10.6:59244
��"abcd fgh
```



```
┌──(ashborn㉿kali)-[~]
└─$ telnet 172.20.10.6 8787
Trying 172.20.10.6...
Connected to 172.20.10.6.
Escape character is '^]'.
^]
telnet> mode line
abcd fgh
igh
```



```
❯❯ nc -vlp 8787
Connection from 172.20.10.6:59244
��"abcd fgh
igh
```

Advantages Of TELNET

Simplicity: Straightforward protocol and easy to set up and use.

Portability: It can be accessed from a wide range of devices.

Low Overhead: Telnet is a lightweight protocol, leading to low overhead.

Disadvantages Of TELNET

Lack of Encryption: All data, including login credentials and commands, are sent in plain text, making it vulnerable to interception and eavesdropping.

Security Risks: Using Telnet poses significant security risks.

Man-in-the-Middle Attacks: Without encryption, Telnet sessions are susceptible to man-in-the-middle attacks.

How do we overcome these shortcomings?

As we saw, TELNET has some serious flaws, and security implications.

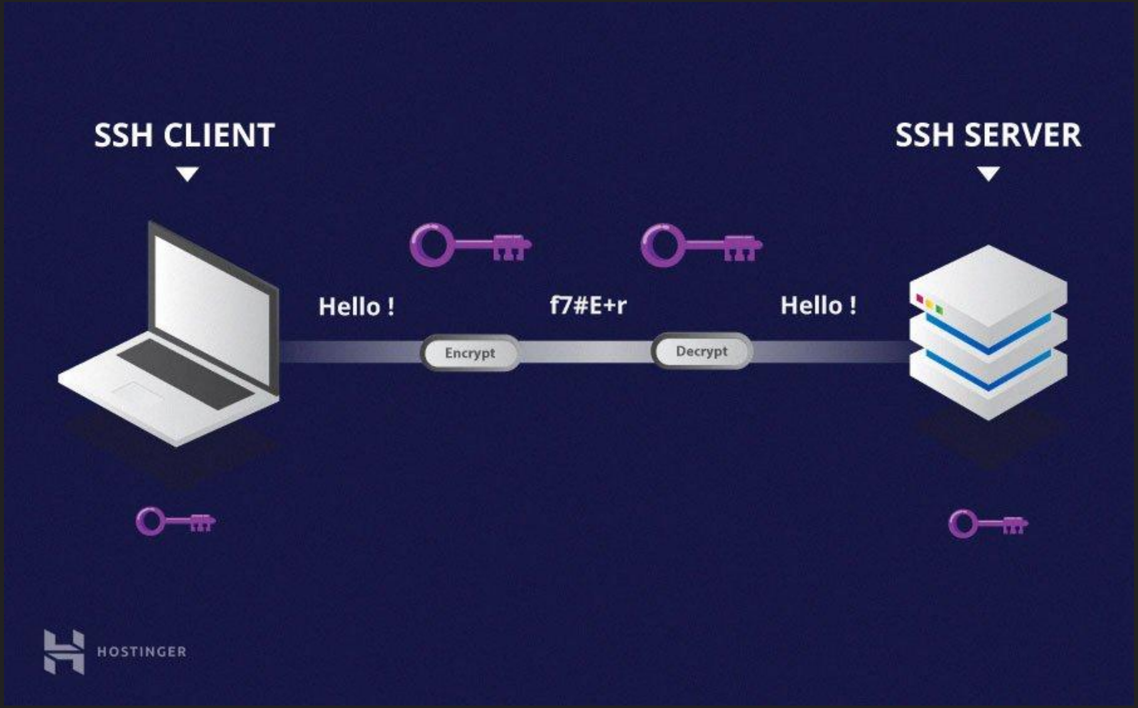
Naturally, the question arises, what is the way forward?

The Answer: Secure Shell Protocol

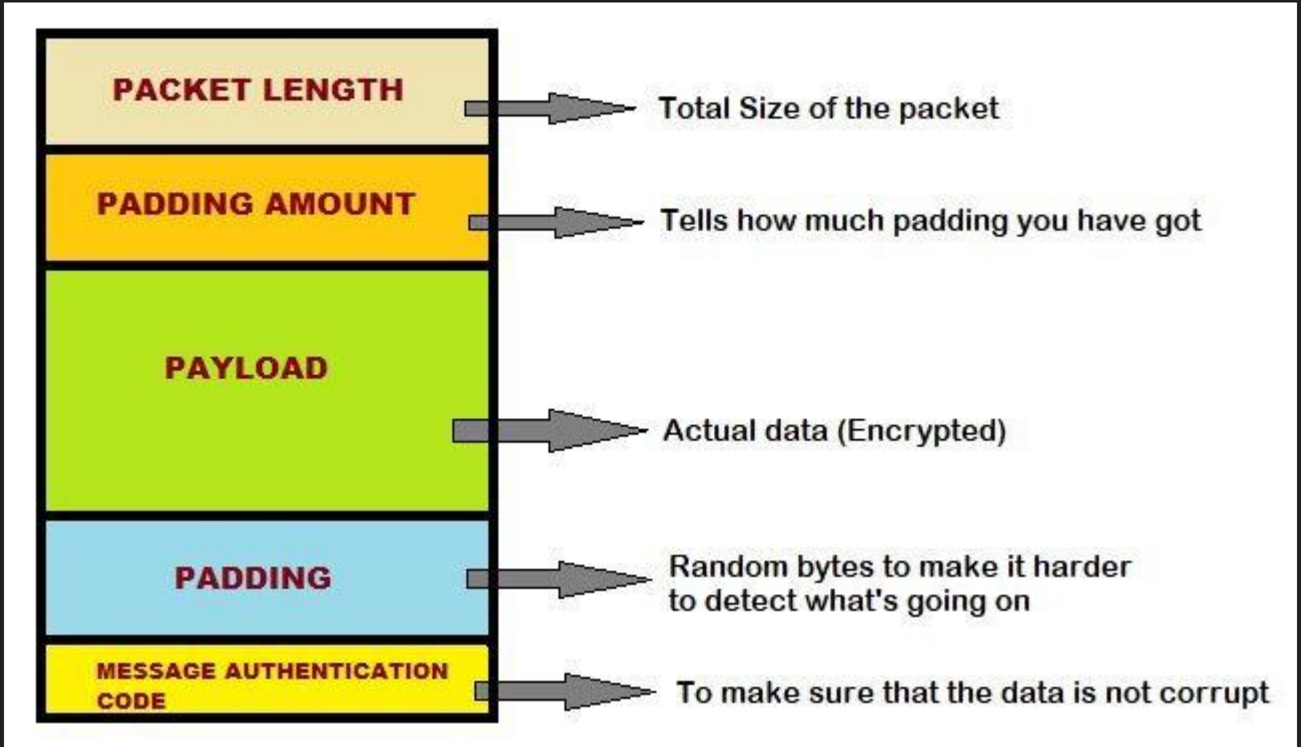
SSH stands for Secure Shell or Secure Socket Shell. It is a cryptographic network protocol that allows two computers to communicate and share the data over an insecure network such as the internet. It is used to login to a remote server to execute commands and data transfer from one machine to another machine.

The SSH protocol was developed by SSH communication security Ltd to safely communicate with the remote machine.

A simple example can be understood, such as suppose you want to transfer a package to one of your friends. Without SSH protocol, it can be opened and read by anyone. But if you will send it using SSH protocol, it will be encrypted and secured with the public keys, and only the receiver can open it.



Okay, but how is data sent?



PACKET LENGTH → Total Size of the packet

PADDING AMOUNT → Tells how much padding you have got

PAYLOAD → Actual data (Encrypted)

PADDING → Random bytes to make it harder to detect what's going on

MESSAGE AUTHENTICATION CODE → To make sure that the data is not corrupt

What can be transferred using SSH?

The SSH protocol can transfer the following:

oData

oText

oCommands

oFiles

The files are transferred using the SFTP(Secure file transfer protocol), the encrypted version of FTP that provides security to prevent any threat

Thank You