EXPERIMENT 12:

Advanced linux commands curl, wget, ftp, ssh and grep

1. curl

- **curl** is a tool to transfer data from or to a server, using one of the supported protocols (HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction.

Download a single file

- The following command will get the content of the URL and display it in the STDOUT (i.e on your terminal).

```
$ curl http://www.cet.ac.in
```

- To store the output in a file, you an redirect it as shown below. This will also display some additional download statistics.

```
$ curl http://www.cet.ac.in > cet.html
```

Save the curl Output to a file

- We can save the result of the curl command to a file by using -o/-O options.

  - o (lowercase o) the result will be saved in the filename provided in the command line

    ```
    curl -o  test.html  http://www.cet.ac.in
    ```

  - -O (uppercase O) the filename in the URL will be taken and it will be used as the filename to store the result.

    ```
    $ curl -O  https://www.tutorialspoint.com/unix_commands/curl.html
    ```

Fetch Multiple Files at a time

- We can download multiple files in a single shot by specifying the URLs on the command line.

```
$ curl -O URL1 -O URL2
```

Follow HTTP Location Headers with -L option

- By default curl doesn't follow the HTTP Location headers. It is also termed as Redirects. When a requested web page is moved to another place, then an HTTP Location header will be sent as a Response and it will have where the actual web page is located.
- Eg:

```
$ curl http://www.google.com

<TITLE>302 Moved</TITLE>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.in/">here</A>
```

- The above output says that the requested document is moved to 'http://www.google.co.in/'. We can insists curl to follow the redirection using -L option, as shown below. Now it will download the google.co.in's html source code.

```
$ curl -L http://www.google.com
```

Continue/Resume a Previous Download

- Using curl -C option, you can continue a download which was stopped already for some reason. This will be helpful when you download large files, and the download got interrupted.

- Start a big download using curl, and press Ctrl-C to stop it in between the download.
- Eg:

```
$ curl -O http://www.gnu.org/software/gettext/manual/gettext.html
##############            20.1%
```

- Now the above download was stopped at 20.1%. Using "curl -C -", we can continue the download from where it left off earlier. Now the download continues from 20.1%.

```
curl -C -  -O http://www.gnu.org/software/gettext/manual/gettext.html
###############            21.1%
```

Pass HTTP Authentication in curl

- Sometime, websites will require a username and password to view the content (can be done with .htaccess file). With the help of -u option, we can pass those credentials from cURL to the web server as shown below.

```
$ curl -u username:password URL
```

Download a file only if it is modified before/after the given time

- We can get the files that are modified after a particular time using -z option in curl. This will work for both FTP & HTTP.

```
$ curl -z 21-Dec-11 http://www.example.com/yy.html
```

- The above command will download the yy.html only if it is modified later than the given date and time.

```
$ curl -z -21-Dec-11 http://www.example.com/yy.html
```

- The above command will download the yy.html, if it is modified before than the given date and time.

2. wget

- **wget** stands for "web get". It is a command-line utility which downloads files over a network.

Download Single File with wget

- The following example downloads a single file from internet and stores in the current directory.

```
$ wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

Specify Download Speed / Download Rate Using wget –limit-rate

- We can limit the download speed using the –limit-rate. In the following example, the download speed is limited to 200k.

```
$ wget --limit-rate=200k http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

Continue the Incomplete Download Using wget -c

- Restart a download which got stopped in the middle using wget -c option as shown below.

```
$ wget -c http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

Download in the Background Using wget -b

- For a huge download, put the download in background using wget option -b as shown below.

```
$ wget -b http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

Download Multiple Files / URLs Using Wget -i

- First, store all the download files or URLs in a text file as:

```
$ cat > download-file-list.txt
  URL1
  URL2
  URL3
  URL4
```

- Next, give the download-file-list.txt as argument to wget using -i option as shown below.

```
$ wget -i download-file-list.txt
```

3. FTP

- FTP is File Transfer Protocol.

Connect to a FTP site

- You can directly open connection with a remote host using it's IP or host name from the command line.
- Syntax:

```
$ ftp IP/hostname
```

or

```
$ ftp
ftp> open IP/hostnam
```

Download a file using ftp

- Use the get command to download file from a remote ftp server as shown below.

```
ftp> get FILENAME
```

Changing FTP Mode to binary or ascii

- Go to ftp Ascii mode

```
ftp> ascii
200 Type set to A
```

- Go to ftp Binary mode

```
ftp> binary
200 Type set to I.
```

Uploading a file to FTP server

- Use put command to upload a file to a remote ftp server as shown below.

```
ftp> put filename
```

Listing the contents of remote directory from FTP

- You can view the content of a remote directory using the **ls** / **dir** command.

```
ftp> ls
```

Close a FTP connection

- Without exiting the ftp prompt you may want to open a connection to another server. In that case, execute **close** command.

```
ftp> close
```

4. ssh

- SSH, or *Secure Shell*, is a protocol used to securely log onto remote systems. It is the most common way to access remote Linux and Unix-like servers.
- The most basic form of the command is:

```
$ ssh remote_host
```

- The *remote_host* in this example is the IP address or domain name that you are trying to connect to.

- This command assumes that your username on the remote system is the same as your username on your local system.
- If your username is different on the remote system, you can specify it by using this syntax:

```
$ ssh remote_username@remote_host
```

- To exit back into your local session, simply type

```
$exit
```

File transfer to/from remote host:

- Another common use of ssh client is to copy files from/to remote host using scp.
- Copy file from the remotehost to the localhost:

```
localhost$scp jsmith@remotehost.example.com:/home/jsmith/remotehostfile.txt
remotehostfile.txt
```

- Copy file from the localhost to the remotehost:

```
localhost$scp localhostfile.txt
jsmith@remotehost.example.com:/home/jsmith/localhostfile.txt
```

5. grep

- grep, which stands for "global regular expression print," processes text line by line and prints any lines which match a specified pattern.

Search for the given string in a single file

- The basic usage of grep command is to search for a specific string in the specified file as shown below.
- Syntax:

```
$ grep "literal_string" filename
```

- Eg:

```
$ grep "this" demo_file
```

Checking for the given string in multiple files.

- Syntax:

```
grep "string" FILE_PATTERN
```

- For example, if demo_file and demo_file1 are the files, then

```
$ grep "this" demo_*
```

Case insensitive search using grep -i

- This searches for the given string/pattern case insensitively. So it matches all the words such as “the”, “THE” and “The” case insensitively as shown below.

```
$ grep -i "the" demo_file
```

Match regular expression in files

- This is a very powerful feature, if you can use use regular expression effectively. In the following example, it searches for all the pattern that starts with “lines” and ends with “empty” with anything in-between. i.e To search “lines[anything in-between]empty” in the demo_file.

```
$ grep "lines.*empty" demo_file
```