Week 2:

Grep command in Unix:

The grep command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files. Its name is derived from the ed (editor) command g/re/p (globally search for a regular expression and print matching lines), which reflects its core functionality. grep is widely used by programmers, system administrators, and users alike for its efficiency and versatility in handling text data. In this article, we will explore the various aspects of the grep command.

-c: This prints only a count of the lines that match a pattern

-h: Display the matched lines, but do not display the filenames.

-l: Displays list of a filenames only.

-n: Display the matched lines and their line numbers.

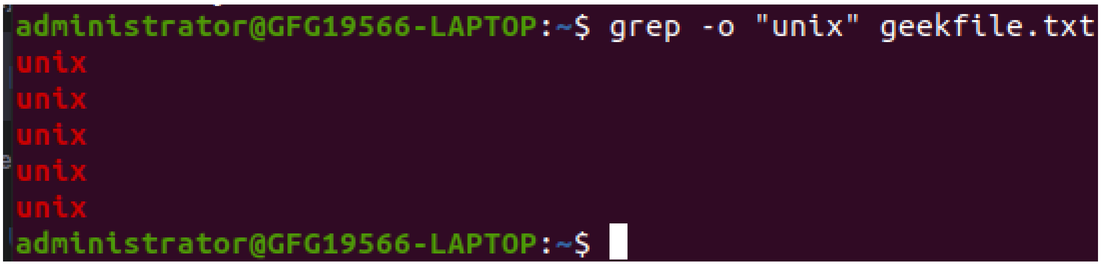
-v: This prints out all the lines that do not matches the pattern

-e : Specifies expression with this option. Can use multiple times.

-o :Print only the matched parts of a matching line, with each such part on a separate output line.

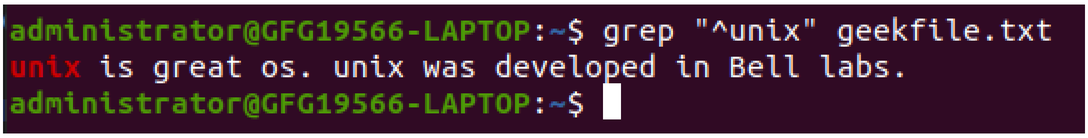
Example:

```
grep -o "unix" geekfile.txt
```



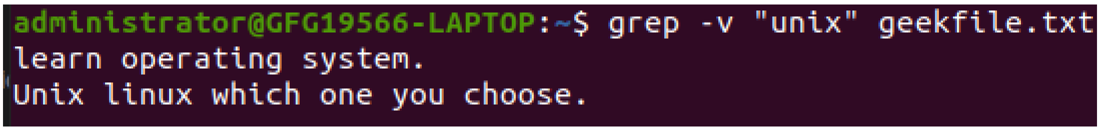
```
administrator@GFG19566-LAPTOP:~$ grep -o "unix" geekfile.txt
unix
unix
unix
unix
unix
administrator@GFG19566-LAPTOP:~$
```

```
grep "^unix" geekfile.txt
```



```
administrator@GFG19566-LAPTOP:~$ grep "^unix" geekfile.txt
unix is great os. unix was developed in Bell labs.
administrator@GFG19566-LAPTOP:~$
```

```
grep -v "unix" geekfile.txt
```



```
administrator@GFG19566-LAPTOP:~$ grep -v "unix" geekfile.txt
learn operating system.
Unix linux which one you choose.
```

Sed Command in Unix:

- **SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).**
- **SED command in unix supports regular expression which allows it perform complex pattern matching.**

Example:

Consider the below text file as an input.

$cat > geekfile.txt

unix is great os. unix is opensource. unix is free os.
learn operating system.
unix linux which one you choose.
unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

Replacing or substituting string : Sed command is mostly used to replace the text in a file. The below simple sed command replaces the word “unix” with “linux” in the file.

```
$sed 's/unix/linux/' geekfile.txt
```

Output :

```
linux is great os. unix is opensource. unix is free os.
learn operating system.
linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a
powerful.
```

Here the "s" specifies the substitution operation. The "/" are delimiters. The "unix" is the search pattern and the "linux" is the replacement string.
By default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second, third…occurrence in the line.

Example:

Replacing the nth occurrence of a pattern in a line :
Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word "unix" with "linux" in a line.

```
$sed 's/unix/linux/2' geekfile.txt
```

Output:

```
unix is great os. linux is opensource. unix is free os.
learn operating system.
unix linux which one you choose.
unix is easy to learn.linux is a multiuser os.Learn unix .unix is a
powerful.
```

Try it:

- Sed 's/hello/HELLO/g' file1.txt
- Sed '2 s/hello/HELLO/2g' file1.txt
- Sed '2d' file1.txt : removing second line of the paragraph

- Sed ‘2,4d’ file1.txt : removing second line and 4 th line of the paragraph
- Sed ‘/core/d’ file1.txt : removing the total line of the core
- $ sed '$d' file1.txt
- $sed '2,$ s/unix/linux/' file1.txt
- $sed 's/unix/linux/p' file1.txt

Sort Command:

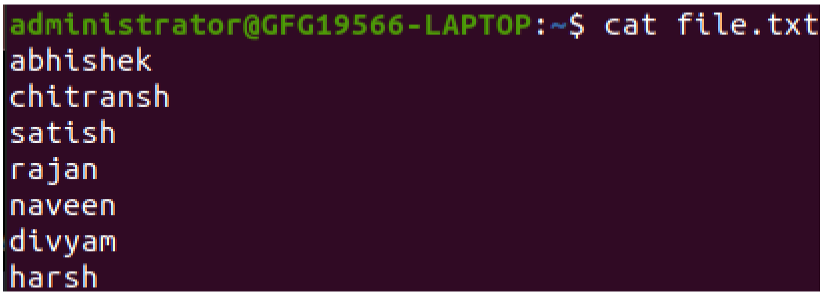
In Unix-based operating systems, the sort command is used to sort the lines of a text file or input stream. It's a versatile command with various options to customize the sorting behavior.

Uses of sort command:

1. It can sort any type of file be it table file text file numeric file and so on.
2. Sorting can be directly implemented from one file to another without the present work being hampered.
3. Sorting of table files on the basis of columns has been made way simpler and easier.
4. So many options are available for sorting in all possible ways.
5. The most beneficial use is that a particular data file can be used many times as no change is made in the input file provided.

Example:1

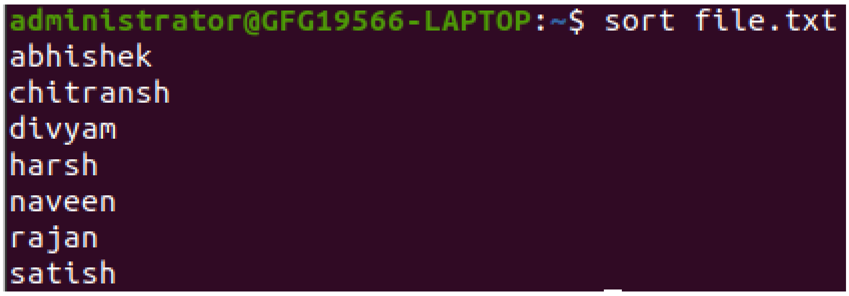
```
cat file.txt
```



```
administrator@GFG19566-LAPTOP:~$ cat file.txt
abhishek
chitransh
satish
rajan
naveen
divyam
harsh
```

```
sort file.txt
```

output:



```
administrator@GFG19566-LAPTOP:~$ sort file.txt
abhishek
chitransh
divyam
harsh
naveen
rajan
satish
```

Try it:

- **-o**: Specifies an output file for the sorted data. Functionally equivalent to redirecting output to a file.
- **-r:** Sorts data in reverse order (descending).
- **-n:** Sorts a file numerically (interprets data as numbers).
- **-nr:** Sorts a file with numeric data in reverse order. Combines -n and -r options.
- **-k:** Sorts a table based on a specific column number.
- **-c:** Checks if the file is already sorted and reports any disorder.

- **-u:** Sorts and removes duplicate lines, providing a unique sorted list.

- -M: Sorts by month names.

chmod Command:

In Unix operating systems, the **chmod** command is used to change the access mode of a file. The name is an abbreviation of **change mode**. Which states that every file and directory has a set of permissions that control the permissions like who can read, write or execute the file. In this the permissions have three categories: read, write, and execute simultaneously represented by `r`, `w` and `x`. These letters combine together to form a specific permission for a group of users.
The `chmod` command is used to modify this permission so that it can grant or restrict access to directories and files. Let's have a look at the syntax and options for the `chmod` command in Linux Operating System.

The following operators can be used with the symbolic mode:

Operators	**Definition:**
`+`	Add permissions
`-`	Remove permissions
`=`	Set the permissions to the specified values

The following letters that can be used in symbolic mode:

Letters	**Definition**
`r`	Read permission
`w`	Write permission
`x`	Execute permission

The following Reference that are used:

Reference	**Class or categories of users**
u	Owner
g	Group
o	Others
a	All (owner,groups,others)

Octal mode:
It is also a method for specifying permissions. In this method we specify permission using three-digit number. Where..

First digit specify the permission for Owner.
Second digit specify the permission for Group.
Third digit specify the permission for Others. The digits
NOTE: The digits are calculated by adding the values of the individual permissions.

Value	**Permission**	
4	Read Permission	**- r**
2	Write Permission	**- w**
1	Execute Permission	**- x**

Grant execute permission to the owner:

- chmod u+x filename
- chmod u-x filename

Remove write permission for group and others:

- chmod go-w filename

Try it:

```
chmod u+x filename      Add execute permission for the owner
chmod go-w filename     Remove write permission for group
chmod a=rwx filename    Set read, write, and execute permissions
chmod a=w filename
chmod ugo=w filename
```

AWK command in Unix:

awk is a powerful text processing tool available in Unix and Unix-like operating systems. It is particularly useful for processing structured text files, such as CSV files or log files. **awk** operates on a line-by-line basis, processing each line of input according to the specified commands. Here are some common **awk** commands and their usage:

Example:

Consider the following text file as the input file for all cases below:

```
$cat > employee.txt
```

```
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
```

satvik director purchase 80000

Print the lines which match the given pattern.

```
$ awk '/manager/ {print}' employee.txt
```

Output:

```
ajay manager account 45000
varun manager sales 50000
amit manager account 47000
```

try it:

Splitting a Line Into Fields : For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

```
$ awk '{print $1,$4}' employee.txt
```

- Use of NR built-in variables (Display Line Number)

```
$ awk '{print NR,$0}' employee.txt
```

- To return the second column/item from file1.txt:

```
$ awk '{print $2}' file1.txt
```

- To count the lines in a file:

```
$ awk '{ print NR }' file.txt
```

```
awk  free  -M
```

- awk ‘/Mem/{print}/’ free -m

- awk ‘NR==2{printNR,$0}’ free -m

shutdown command:

The shutdown command in Linux is used to shutdown the system in a safe way. You can shutdown the machine immediately, or schedule a shutdown using 24 hour format.It brings the system down in a secure way. When the shutdown is initiated, all logged-in users and processes are notified that the system is going down, and no further logins are allowed.

```
shutdown [options] [time] [message]
```

```
shutdown -r +5 reboot
```

options – Shutdown options such as halt, power-off (the default option) or reboot the system.
time – The time argument specifies when to perform the shutdown process.
message – The message argument specifies a message which will be broadcast to all users.

Examples:

```
 $ shutdown -H
 $shutdown -P
$ shutdown -r now
$ shutdown -r +5
$  sudo shutdown 15:00
$ sudo shutdown +10
$ sudo shutdown -c
```

Options: try it:

r : Requests that the system be rebooted after it has been brought down.
-h : Requests that the system be either halted or powered off after it has been brought down, with the choice as to which left up to the system.
-H : Requests that the system be halted after it has been brought down.
-P : Requests that the system be powered off after it has been brought down.
-c : Cancels a running shutdown. TIME is not specified with this option, the first argument is MESSAGE.
-k : Only send out the warning messages and disable logins, do not actually bring the system down.

SSH :

SSH, or Secure Shell, constitutes a cryptographic network protocol designed to enable secure communication between two systems over networks that may not be secure. This protocol is widely employed for remote access to servers and the secure transmission of files between computers. In essence, SSH acts as a secure conduit, establishing a confidential channel for communication in scenarios where the network may pose security risks. This technology is instrumental for professionals seeking a reliable and secure method of managing servers and transferring sensitive data across computers in a controlled and protected manner. ssh runs at TCP/IP port 22.

```
ssh [username]@[hostname or IP address
```

Replace [username] with your remote server username, and [hostname or IP address] with the server's hostname or IP address.

Install SSH Component on Linux

Setting up SSH on Linux may be necessary, as some distributions don't come with it pre-installed. Installing OpenSSH, a widely used SSH implementation, or opting for a graphical user interface (GUI) solution like the PuTTY client for Ubuntu can address this. Here's a step-by-step guide on installing and configuring OpenSSH on both the client and server sides:

```
sudo yum install openssh-clients openssh-server
```

- This command is used to connect to a remote server.

```
ssh [username]@[hostname or IP address]
```

ex:

```
ssh user@example.com
```

- ssh-copy-id: This command is used to copy your SSH public key to a remote server's authorized_keys file, enabling passwordless SSH login.

```
ssh-copy-id [username]@[hostname or IP address]
```

ex:

```
ssh-copy-id user@example.com
```

- ssh-add: This command is used to add private key identities to the SSH authentication agent.

```
ssh-add [path to private key]
```

ex:

```
ssh-add ~/.ssh/id_rsa
```

- This command connects to the remote server using port 2222. Adjust the port number as needed

```
ssh -p 2222 username@remote_server_ip
```

command consists of 3 different parts:

- ssh command instructs the system to establish an encrypted secure connection with the host machine.
- user_name represents the account that is being accessed on the host.
- host refers to the machine which can be a computer or a router that is being accessed. It can be an IP address (e.g., 192.168.1.24) or domain e.g., www.domainname.com).

The three major encryption techniques used by SSH are:

- Symmetrical encryption
- Asymmetrical encryption
- Hashing

SSH
https://sdf.org/?signup
Check mail
SDF Username: sujoysarkar
SDF Password: kseFUqtOXGlNwg

Please login to your account via the following methods:

```
ssh sujoysarkar@tty.sdf.org
password
```

```
mkdir dir1
cd dir1
touch abc.txt
ls
exit
```

FTP:

File transfer protocol (FTP) is an Internet tool provided by TCP/IP. The first feature of FTP is developed by Abhay Bhushan in 1971. It helps to transfer files from one computer to another by providing access to directories or folders on remote computers and allows software, data, text file to be transferred between different kinds of computers. The end-user in the connection is known as localhost and the server which provides data is known as the remote host.

FTP Clients
FTP works on a client-server model. The FTP client is a program that runs on the user's computer to enable the user to talk to and get files from remote computers. It is a set of commands that establishes the connection between two hosts, helps to transfer the files, and then closes the connection. Some of the commands are: get filename(retrieve the file from server), mget filename(retrieve multiple files from the server), ls(lists files available in the current directory of the server). There are also built-in FTP programs, which makes it easier to transfer files and it does not require remembering the commands.

Type of FTP Connections

FTP connections are of two types:

Active FTP connection: In an Active FTP connection, the client establishes the command channel and the server establishes the data channel. When the client requests the data over the connection the server initiates the transfer of the data to the client. It is not the default connection because it may cause problems if there is a firewall in between the client and the server.

Passive FTP connection: In a Passive FTP connection, the client establishes both the data channel as well as the command channel. When the client requests the data over the connection, the server sends a random port number to the client, as soon as the client receives this port number it establishes the data channel. It is the default connection, as it works better even if the client is protected by the firewall.

When an FTP connection is established, there are two types of communication channels are also established and they are known as command channel and data channel. The command channel is used to transfer the commands and responses from client to server and server to client. FTP uses the same approach as TELNET or SMTP to communicate across the control connection. It uses the NVT ASCII character set for communication. It uses port number 21. Whereas the data channel is used to actually transfer the data between client and server. It uses port number 20.

FTP Commands

1. **cd** Changes the working directory on the remote host
2. **close** Closes the FTP connection
3. **quit** Quits FTP
4. **pwd** displays the current working Directory on the rem remote host
5. **dis or ls** Provides a Directory Listing of the current working directory
6. **help** Displays a list of all client FTP commands
7. **remotehelp** Displays a list of all server FTP commands
8. **type** Allows the user to specify the file type
9. **struct** specifies the files structure

https://dlptest.com/ftp-test/

ftp
ftp://dlpuser:rNrKYTX9g7z3RgJRmxWuGHbeu@ftp.dlptest.com/

```
ls
get some_text.txt
```

Service:

Linux operating systems are known for their robustness and versatility, and managing system services is a crucial aspect of maintaining a well-functioning system. With the advent of systemd, a system and service manager for Linux operating systems, the systemctl command has become an essential tool for managing services. In this article, we will explore the intricacies of systemctl and how it can be used to control and monitor system services.

Starting and Stopping Services

```
systemctl start [service]
```

- Enabling Services

```
systemctl enable [service]
```

If we want to enable our firewall service.

```
systemctl enable firewalld
```

- Disabling Services

```
systemctl disable [service]
```

If we want to disable our firewall service.

```
systemctl disable firewalld
```

Restarting and Reloading Services

- Restarting Services

```
systemctl restart [service]
```

If we want to restart our SSH service.

```
systemctl restart sshd
```

- Reloading Services

```
systemctl reload [service]
```

If we want to reload our Apache service.

```
systemctl reload httpd
```

chown Command

the Linux operating system, file ownership is a crucial aspect of system security and user management. The `**chown**` command, short for "change owner," is a powerful tool that allows users to change the owner of files and directories. This command is particularly useful in scenarios where administrators need to grant or revoke access to specific resources. In this article, we will explore the fundamentals of file ownership in Linux and delve into the usage of the chown command.

- **Root User:** It is a superuser who has access to all the directories and files in our system and it can perform any operation. An important thing to note is that only the root user can perform changing of permissions or ownerships of the files that are not owned by them.
- **Regular User:** These users have limited access to files and directories and can only modify the files that they own.

Ownership and Permissions:

To protect and secure files and directories in Linux we use permissions to control what a user can do with a file or directory.

Linux uses three types of permissions:

Read: This permission allows the user to read files in directories, it lets the user read directories and subdirectories stored in it.

Write: This permission allows a user to modify and delete a file. Also, it allows a user to modify its contents (create, delete, and rename files in it) for the directories. Unless the execution permission is given to directories changes do affect them.

Execute This permission on a file allows it to get executed. For example, if we have a file named php.sh unless we don't give it execute permission it won't run.

Here's a breakdown of the components:

Here's a breakdown of the components:

- **`chown`**: The base command.
- **`options`**: Optional flags that modify the behavior of the `chown` command.
- **`new_owner[:new_group]`:** The new owner and optionally the new **group**. If `new_group` is omitted, only the owner is changed.

Here's a breakdown of the components:

How to File Ownership in Linux

To Change the owner of a file in Linux, you can use the following basic syntax:

```
chown owner_name file_name
```

```
chown master file1.txt
```

To change the group ownership of a file, utilize the following syntax:

```
chown :group1 file1.txt
```