CSI STUDENT
BRANCH, DDU

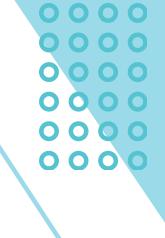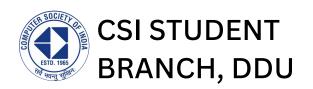# SHELL & EXCEL: LINUX UNRAVELED

Me 5 minutes before my project submission looking for my FINAL FINAL code file amongst 20 others

Kaha gaya use dhundo

FIND COMMAND

- It helps locate files in the <u>directory hierarchy</u>.
- *"-name"* for case sensitive search.
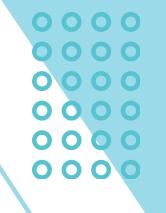- *"-iname"* for case-insensitive search.

```
find . -name "xyz.txt"    # Case-sensitive search for 'xyz.txt' in the current directory
find . -iname "XYZ.TXT"    # Case-insensitive search for 'XYZ.TXT' in the current directory
```

- You can count the <u>number of files</u> in a directory using *find* combined with *"wc -l"*.

```
find . -type f -iname "*.txt" | wc -l  # Count all .txt files #Used to find content of specific type
find . -type f | wc -l                  # Count all files in the current directory
```

SEARCHING FOR EXCLUSIVE FILES

- For finding <u>Files Owned by a Specific User</u> ,one can use the *"-user"* option.

```
find /path/to/search -user xyz   # Replace 'xyz' with the username
```

- For finding <u>Files with Specific Permissions</u>, one can use the *"-perm"* option.

```
find /path/to/search -perm xyz   # Replace 'xyz' with the required permission
# (e.g., 777 for full access)
```

SEARCHING FOR EXCLUSIVE FILES

- For finding Files <u>Based on Size</u>, one can use the "*-size*" option.

```
find . -size +10M    # Files larger than 10MB
find . -size -500k   # Files smaller than 500KB
find . -size 100c    # Files that are exactly 100 bytes
```

- *Common File size units:*
  c: Bytes
  k: Kilobytes
  M: Megabytes
  G: Gigabytes

SEARCHING FOR CONTENT

- If you wish to <u>search a file based on its content</u> or the phrases used within, you can use *find* with *"grep"*.

- The *"grep -l"* <u>lists only the file names</u> that contain the specified text, <u>without displaying the matching lines.</u>

```
find . -name "*.txt" -exec grep -l "word to search" {} + # Search within .txt file & list file name
find . -type f -exec grep -l "xyz" {} +   # Search inside all types of files available
```
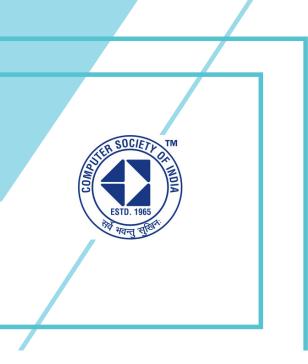
Grep command can also be used to find multiple forms of content in multiple files separately as well as simustaneously

GRIP OVER "GREP"

- You can <u>search for text across multiple files</u> by specifying multiple file names.

```
grep "xyz" file.name file1.name  # Search for 'xyz' in multiple files
```

- One can use *"-w"* with *grep* to <u>search for exact word matches</u>. This avoids partial matches.

```
grep -iw "xyz" file.name  # Search for exact match of 'xyz'
# "-i" is used for case-insensitive search
```
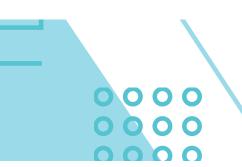
GRIP OVER "GREP"

- In order to <u>search multiple keywords in a file</u>, one can use *"-e"* option.

```
grep -i -e "xyz" -e "abc" file.name  # Search for 'xyz' OR 'abc' in a file
```

- When searching for <u>multiple keywords, in multiple files</u> one can *combine -e with multiple file names searches*.

```
grep -i -e "xyz" -e "abc" file.name file1.name  # Search for 'xyz' OR 'abc' in multiple files
```

- The *"-v"* option <u>inverts the search</u>, displaying lines from file that <u>do not contain</u> the specified text.

```
grep -v "xyz" file.name  # Show lines that do NOT contain 'xyz'
```

SECURE SHELL

- SSH (Secure Shell) is a protocol used to <u>securely connect to remote systems</u> over a network.
- *Basic command Syntax:*

ssh user@host -p port [options] where:

- *"**user**"*: The username on the remote system.

*"**host**"*: The IP address or domain name of the remote system.

*"**-p port**"*: Specifies the port to connect on (default is 22).

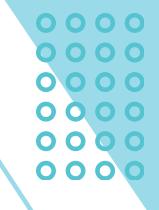*[options]*: Additional options to modify the SSH behavior.

SECURE SHELL

SSH Communication summed up:

SFTP (SECURE FILE TRANSFER PROTOCOL)

- SFTP is a secure file transfer protocol that allows file transfers over an encrypted SSH connection.

Basic SFTP Command Format

Command Syntax:

```
sftp user@host
```

- **user** : username of the remote system
- **host** : The IP address or domain name of the remote system.

Example

```
sftp nisarg@192.168.1.10
```

SFTP (SECURE FILE TRANSFER PROTOCOL)

Uploading Files

- Single file upload:

```
put filename.txt
```

- Multiple files upload:

```
put file1.txt file2.txt file3.txt
```

- Uploading Files to a Specific Directory:

```
put filename.txt /path/to/directory/
```

SFTP (SECURE FILE TRANSFER PROTOCOL)

Downloading Files

- Single file download:

```
get filename.txt
```

- Multiple files download:

```
get file1.txt file2.txt file3.txt
```

- Downloading Files to a Specific Directory:

```
get /path/to/directory/filename.txt
```

SFTP (SECURE FILE TRANSFER PROTOCOL)

Important SFTP Commands

- **ls :** Lists files in the remote directory.

- **pwd** : Shows the current remote working directory.

- **cd** : Changes the remote working directory.

- **bye/exit** : Exits the SFTP session.

FEEDBACK FORM