1. **ls: used for listing of files**

   ls -al; gives detailed information of the files

   ls -a:Listing Hidden Files

2. **cat: creating and viewing of files**

   cat > filename : create new file

   cat filename: to view file

   cat fileone finetwo > filecombined : combine two files into one

3. **rm: to delete files**

   rm filename

4. **mv: to move files**

   mv file location (location is the path where you want to move the file)

5. **chmod: To change permission**

chmod 777 filename (777 is the permission code)

chown: change ownership of file

chown owner:groupowner filename

6. **grep:To search pattern**

search test in all text files then grep command will be:

grep test *.txt

7. **tail: to display last lines of files**

tail -n 5 test.txt : it will display last 5 lines of test.txt file

8. **ssh: to connect to remote linux machines**

ssh user@IP

for example the command will look like; ssh sidharth@192.168.x.x

9. **traceroute:**

tracks the route that a particular packet of information takes to reach to the host

traceroute www.automationreinvented.blogspot.com

10. ftp and sftp:

to connect to remote server and download files

$ ftp IP/hostname

ftp> mget *.txt

11. ps: to check process

ps -ef: current running process

we can choose more or less with below
ps -ef | less or more

12. `dstat` – view processes, memory, paging, I/O, CPU, etc., in real-time. All-in-one for vmstat, iostat, netstat, and ifstat.

13. `iotop` – interactive I/O viewer. Get an overview of storage r/w activity.
14. `rsync` – remote file transfers and syncing.

What is Linux, and why is it used for testing?

Linux is an open-source operating system known for stability and security. It's commonly used for testing due to its flexibility and availability.

How do you check the Linux distribution and version?

Use the `lsb_release -a` command to check the distribution and version. For example:
css

```
lsb_release -a
```

What is the purpose of the `grep` command in Linux?

The `grep` command is used for searching and matching patterns in text files. For example, to find lines containing "error" in a log file:
perl

```
grep "error" logfile.txt
```

How do you list files in a directory using the `ls` command?

Use the `ls` command to list files and directories in the current directory. For example:
bash

```
ls -l
```

What is the `ps` command in Linux, and how do you list running processes?

The `ps` command is used to list running processes. To display all processes, use:

```
ps aux
```

How can you check disk space usage in Linux using the `df` command?

Use the `df` command to display disk space usage. For example:
bash

```
df -h
```

What is the purpose of the `top` command, and how do you use it to monitor system resources?

The `top` command provides real-time information about system resource usage. It can be used to monitor CPU, memory, and process activity. Press `q` to exit.

How do you create a new directory in Linux using the `mkdir` command?

To create a new directory named "test," use:
bash

```
mkdir test
```

What is the `chmod` command used for in Linux?

The `chmod` command is used to change file permissions. For example, to make a file executable:
bash

```
chmod +x filename
```

How can you compress and extract files in Linux using `tar`?

To compress files into a tarball, use:

```
tar -cvzf archive.tar.gz file1 file2
```

To extract files from a tarball, use:

```
tar -xvzf archive.tar.gz
```

What is the `curl` command used for in Linux?

The `curl` command is used to make HTTP requests. For example, to download a file:
arduino

```
curl -O https://example.com/file.txt
```

How do you search for a file in Linux using the `find` command?

Use the `find` command to search for a file by name or pattern. For example, to find all text files in the current directory:
arduino

```
find . -name "*.txt"
```

What is SSH, and how do you connect to a remote server using SSH?

SSH (Secure Shell) is a protocol for secure remote login. Use the `ssh` command to connect to a remote server:
css

```
ssh username@remote-server
```

How can you check the IP address of a Linux machine using the `ifconfig` command?

Use the `ifconfig` command to display network interface information, including IP addresses. For example:

```
ifconfig
```

What is the purpose of the `grep` command with the `-r` option?

The `grep -r` command is used to search for text recursively in directories and subdirectories. For example:
bash

```
grep -r "pattern" /path/to/directory
```

How do you view the contents of a text file in Linux using the `cat` command?

Use the `cat` command to display the contents of a text file. For example:
bash

```
cat myfile.txt
```

What is the `rsync` command used for in Linux?

The `rsync` command is used for efficient file synchronization and copying. For example, to synchronize two directories:
bash

```
rsync -avz /source-directory/ /destination-directory/
```

How can you schedule tasks in Linux using the `cron` daemon?

Use the `crontab` command to create and manage cron jobs. For example, to schedule a job to run every day at 3:00 PM:
javascript

```
0 15 * * * /path/to/script.sh
```

What is the purpose of the `tail` command in Linux, and how do you use it to view log files?

The `tail` command displays the last few lines of a file, commonly used for viewing log files. For example:
bash

```
tail -n 20 logfile.log
```

How can you check the available memory and swap space using the `free` command?

Use the `free` command to display information about memory and swap space. For example:
`free -m`

SCENARIO BASED REAL TIME

- Question: How would you identify and kill a process using its port number?

  Answer: Utilize the `lsof` command to identify the process using a specific port, and then use the `kill` command to terminate the identified process.

- **Question:** Explain how file permissions work in Linux, and how can you grant executable permission to a script?

  Answer: Linux uses the `chmod` command to modify file permissions. To grant executable permission to a script, use `chmod +x script.sh`.

- **Question:** How can you monitor system resource usage in real-time, and which command would you use?

  Answer: Use the `top` command to monitor real-time system resource usage, providing insights into CPU, memory, and process statistics.

- **Question:** How would you install a specific version of a package using the package manager?

  Answer: Use the package manager (e.g., `apt` for Debian-based systems) with the syntax `apt install <package_name>=<version>`.

- **Question:** Describe the steps you would take to troubleshoot network connectivity issues on a Linux machine.

  Answer: Use commands like `ping`, `traceroute`, and `netstat` to identify and troubleshoot network-related problems.

Question: Write a simple shell script that takes a filename as an argument and counts the number of lines in that file.

Answer:
bash

```
#!/bin/bash

file=$1

lines=$(wc -l < "$file")

echo "The file $file has $lines lines."
```

- **Question:** Explain the Linux boot process and the significance of key directories like /etc and /var.

  Answer: The Linux boot process involves stages such as BIOS/UEFI, bootloader, kernel, init, and user space. Key directories like /etc store system-wide configuration files, while /var holds variable data like logs and caches.

- **Question:** How would you use regular expressions to find all lines in a file containing a specific pattern?

  Answer: Employ the `grep` command with the regular expression pattern. For example, `grep "pattern" file.txt`.

- **Question:** What is the purpose of the `mount` command, and how can you mount an additional disk in Linux?

  Answer: The `mount` command is used to attach filesystems. To mount an additional disk, use a command like `sudo mount /dev/sdb1 /mnt`.

- **Question:** How can you set up passwordless SSH authentication between two Linux machines?

  Answer: Use `ssh-keygen` to generate a key pair, and then copy the public key to the remote machine's `authorized_keys` file. This allows for passwordless authentication.

CHEATSHEET

BASIC LINUX COMMANDS

FILES & NAVIGATING

ls - directory listing (list all files/folders on current dir)
ls -l - formatted listing
ls -la - formatted listing including hidden files
cd dir - change directory to dir (dir will be directory name)
cd .. - change to parent directory
cd ../dir - change to dir in parent directory
cd - change to home directory
pwd - show current directory
mkdir dir - create a directory dir
rm file - delete file
rm -f dir - force remove file
rm -r dir - delete directory dir
rm -rf dir - remove directory dir
rm -rf / - launch some nauclear bombs targeting your system
cp file1 file2 - copy file1 to file2
mv file1 file2 - rename file1 to file2
mv file1 dir/file2 - move file1 to dir as file2
touch file - create or update file
cat file - output contents of file
cat > file - write standard input into file
cat >> file - append standard input into file
tail -f file - output contents of file as it grows

NETWORKING

ping host - ping host
whois domain - get whois for domain
dig domain - get DNS for domain
dig -x host - reserve lookup host
wget file - download file
wget -c file - continue stopped download
wget -r url - recurively download files from url
curl url - outputs the webpage from url
curl -o meh.html url - writes the page to meh.html
ssh user@host - connect to host as user
ssh -p port user@host - connect using port
ssh -D user@host - connect & use bind port

PROCESSES

ps - display currently active processes
ps aux - detailed outputs
kill pid - kill process with process id (pid)
killall proc - kill all processes named proc

SYSTEM INFO

date - show current date/time
uptime - show uptime
whoami - who you're logged in as
w - display who is online
cat /proc/cpuinfo - display cpu info
cat /proc/meminfo - memory info
free - show memory and swap usage
du - show directory space usage
du -sh - displays readable sizes in GB
df - show disk usage
uname -a - show kernel config

COMPRESSING

tar cf file.tar files - tar files into file.tar
tar xf file.tar - untar into current directory
tar tf file.tar - show contents of archive

options:
c - create archive
t - table of contents
x - extract
z - use zip/gzip
f - specify filename
j - bzip2 compression
w - ask for comfirmation
k - do not overwrite
T - files from file
v - verbose

PERMISSIONS

chmod octal file - change permissions of file

4 - read (r)
2 - write (w)
1 - execute (x)

order: owner/group/world

chmod 777 - rwx for everyone
chmod 755 - rw for owner, rx for group world

SOME OTHERS

grep pattern files - search in files for pattern
grep -r pattern dir - search for pattern recursively in dir
locate file - find all instances of file
whereis app - show possible locaitions of app
man command - show manual page for command

Bash Commands

uname -a	Show system and kernel
head -n1 /etc/issue	Show distribution
mount	Show mounted filesy-stems
date	Show system date
uptime	Show uptime
whoami	Show your username
man *command*	Show manual for *command*

Bash Shortcuts

CTRL-c	Stop current command
CTRL-z	Sleep program
CTRL-a	Go to start of line
CTRL-e	Go to end of line
CTRL-u	Cut from start of line
CTRL-k	Cut to end of line
CTRL-r	Search history
!!	Repeat last command
!*abc*	Run last command starting with *abc*
!*abc*:p	Print last command starting with *abc*
!$	Last argument of previous command
ALT-.	Last argument of previous command
!*	All arguments of previous command
^*abc*^*123*	Run previous command, replacing *abc* with *123*

Bash Variables

env	Show environment variables
echo *$NAME*	Output value of *$NAME* variable

Bash Variables (cont)

export *NAME=value*	Set *$NAME* to *value*
$PATH	Executable search path
$HOME	Home directory
$SHELL	Current shell

IO Redirection

cmd < *file*

> Input of *cmd* from *file*

cmd1 <(*cmd2*)

> Output of *cmd2* as file input to *cmd1*

cmd > *file*

> Standard output (stdout) of *cmd* to *file*

cmd > /dev/null

> Discard stdout of *cmd*

cmd >> *file*

> Append stdout to *file*

cmd 2> *file*

> Error output (stderr) of *cmd* to *file*

cmd 1>&2

> stdout to same place as stderr

cmd 2>&1

> stderr to same place as stdout

cmd &> *file*

> Every output of *cmd* to *file*

cmd refers to a command.

Pipes

cmd1 | *cmd2*

> stdout of *cmd1* to *cmd2*

cmd1 |& *cmd2*

> stderr of *cmd1* to *cmd2*

Command Lists

cmd1 ; *cmd2*

> Run *cmd1* then *cmd2*

cmd1 && *cmd2*

> Run *cmd2* if *cmd1* is successful

cmd1 || *cmd2*

> Run *cmd2* if *cmd1* is not successful

cmd &

> Run *cmd* in a subshell

Directory Operations

pwd	Show current directory
mkdir *dir*	Make directory *dir*
cd *dir*	Change directory to *dir*
cd ..	Go up a directory
ls	List files

ls Options

-a	Show all (including hidden)
-R	Recursive list
-r	Reverse order
-t	Sort by last modified
-S	Sort by file size
-l	Long listing format
-1	One file per line
-m	Comma-separated output
-Q	Quoted output

Search Files

grep *pattern files*	Search for *pattern* in *files*
grep -i	Case insensitive search
grep -r	Recursive search
grep -v	Inverted search
grep -o	Show matched part of file only
find */dir/* -name *name**	Find files starting with *name* in *dir*

Search Files (cont)

find */dir/* -user *name*	Find files owned by *name* in *dir*
find */dir/* -mmin *num*	Find files modifed less than *num* minutes ago in *dir*
whereis *command*	Find binary / source / manual for *command*
locate *file*	Find *file* (quick search of system index)

File Operations

touch *file1*

> Create *file1*

cat *file1 file2*

> Concatenate files and output

less *file1*

> View and paginate *file1*

file *file1*

> Get type of *file1*

cp *file1 file2*

> Copy *file1* to *file2*

mv *file1 file2*

> Move *file1* to *file2*

rm *file1*

> Delete *file1*

head *file1*

> Show first 10 lines of *file1*

tail *file1*

> Show last 10 lines of *file1*

tail -F *file1*

> Output last lines of *file1* as it changes

Watch a Command

watch -n 5 'ntpq -p'

> Issue the 'ntpq -p' command every 5 seconds and display output

Process Management

ps	Show snapshot of processes
top	Show real time processes
kill *pid*	Kill process with id *pid*
pkill *name*	Kill process with name *name*
killall *name*	Kill all processes with names beginning *name*

Nano Shortcuts

Files	
Ctrl-R	Read file
Ctrl-O	Save file
Ctrl-X	Close file
Cut and Paste	
ALT-A	Start marking text
CTRL-K	Cut marked text or line
CTRL-U	Paste text
Navigate File	
ALT-/	End of file
CTRL-A	Beginning of line
CTRL-E	End of line
CTRL-C	Show line number
CTRL-_	Go to line number
Search File	
CTRL-W	Find
ALT-W	Find next
CTRL-\	Search and replace

More nano info at:
http://www.nano-editor.org/docs.php

Screen Shortcuts

screen

> Start a screen session.

screen -r

> Resume a screen session.

Screen Shortcuts (cont)

screen -list

> Show your current screen sessions.

CTRL-A

> Activate commands for screen.

CTRL-A c

> Create a new instance of terminal.

CTRL-A n

> Go to the next instance of terminal.

CTRL-A p

> Go to the previous instance of terminal.

CTRL-A "

> Show current instances of terminals.

CTRL-A A

> Rename the current instance.

More screen info at:
http://www.gnu.org/software/screen/

File Permissions

chmod 775 *file*

> Change mode of *file* to 775

chmod -R 600 *folder*

> Recursively chmod *folder* to 600

chown *user:group file*

> Change *file* owner to *user* and group to *group*

File Permission Numbers

First digit is owner permission, second is group and third is everyone.

Calculate permission digits by adding numbers below.

4	read (r)
2	write (w)
1	execute (x)