

LP ASSIGNMENT-3

ENG24CY0159

SHADIK KHAN

1. Distinguish between man and whatis commands? Justify with proper example.

Think of it like asking for information about a movie.

- **whatis** gives you the one-sentence summary you'd read in a TV guide. It's quick, to the point, and tells you just the basics.
- **man** (manual) is like watching the entire movie with director's commentary. It gives you *everything*: the plot, the characters, all the options, and examples.

Example: If you run `whatis ls`, you get something short and sweet:

`ls (1)` - list directory contents

But if you run `man ls`, you get a full-length manual page with pages of detailed information about every possible way to use the `ls` command. You use `whatis` to find out what a command does, and `man` to learn how to use it properly.

2. Use the tee command to save the output of ls -l into a file while also displaying it.

No problem. The `tee` command is perfect for this. It acts like a T-splitter in plumbing, sending the output in two directions at once: to your screen and to a file.

You use the pipe `|` symbol to connect the output of `ls -l` to the input of `tee`.

Command:

Bash

```
ls -l | tee file_listing.txt
```

When you run this, you'll see the detailed file listing (ls -l) on your terminal as usual, AND a new file named file_listing.txt will be created containing that exact same output.

3. Explain with an example how the tee command can be used in logging.

tee is a lifesaver for logging because it lets you **watch a script run in real-time while saving a record of it**.

Imagine you have a long-running backup script. You want to see its progress, but you also need a log file to check for errors later. tee is the perfect tool for the job.

Example:

Let's say you run your script like this:

Bash

```
./my_backup_script.sh | tee -a backup.log
```

- You'll see all the progress messages from my_backup_script.sh on your screen.
 - At the same time, every line of that output is also being saved to backup.log.
 - The **-a** flag is the secret ingredient here; it means **append**. This adds the new log entries to the end of the file instead of wiping it clean every time you run the script.
-

4. List the steps involved in installing Ubuntu 25.04 LTS on Oracle VirtualBox.

First, a quick heads-up: Ubuntu's LTS (Long-Term Support) versions are released in even-numbered years, so you'd be looking at 24.04 LTS or 26.04 LTS. But the steps are the same!

Here's the friendly guide to getting it done:

1. **Get the Goods:** First, download two things: the latest version of Oracle VirtualBox and the Ubuntu Desktop ISO file from the official Ubuntu website. The ISO is basically a digital version of an installation CD.

2. Create Your Virtual Computer:

- Open VirtualBox and click the "New" button.
- Give your new virtual machine (VM) a name, like "My Ubuntu VM".
- Assign it some memory (RAM) – 4GB (4096MB) is a good start.
- Choose to create a new virtual hard disk. The default settings (VDI, dynamically allocated) are perfectly fine. Give it at least 25GB of space.

3. Load the "CD":

- Go into your new VM's "Settings."
- Click on "Storage," select the empty CD drive, and then click the little CD icon on the right to choose your downloaded Ubuntu ISO file.

4. Launch and Install:

- Start the VM! It will boot from the ISO file you just selected.
- Follow the on-screen Ubuntu installer. It's very straightforward: you'll choose your language, keyboard layout, connect to the internet, and then select "Erase disk and install Ubuntu." (Don't worry, this only erases the *virtual* disk, not your actual computer!)
- Set your time zone and create your user account and password.

5. **Final Touches:** After the installation finishes, it will ask you to reboot. Once it's back up, use the VirtualBox menu (Devices > Insert Guest Additions CD Image...) to install the Guest Additions, which will make the screen resolution and mouse work much more smoothly.

5. During Ubuntu OS installation, you face a Kernel Panic Error. How would you troubleshoot it?

Ah, the dreaded Kernel Panic! It's Linux's version of the Blue Screen of Death, but don't worry. When it happens in a virtual machine, your main computer is perfectly safe. Here's how to play detective:

1. **Read the Clues:** The screen full of text isn't just nonsense. Look for a specific line that mentions a file name or module, like `VBoxGuestAdditions.ko` or `ata_piix`. This is your biggest clue as to what's causing the problem.
2. **Check Your VM Settings:** The most common culprit is a setting mismatch. Go back into the VM's settings and check:
 - **RAM:** Did you give it enough memory? Try bumping it up a bit.
 - **Video Memory:** This is a big one. Try increasing the video memory under the "Display" settings.
 - **Processor:** Ensure "Enable PAE/NX" is checked in the processor settings.
3. **Change Virtual Hardware:** Sometimes the default virtual hardware just doesn't agree with the installer. Try changing the "Graphics Controller" under "Display" from VMSVGA to VBoxVGA or vice-versa.
4. **Suspect a Bad Download:** It's possible your Ubuntu ISO file is corrupted. The best way to check is to find the official checksum (usually an SHA256 hash) on the Ubuntu download page and verify your file against it.
5. **Try Safe Graphics Mode:** When the installer first boots up, there's usually an option for "Safe Graphics" or a way to edit the boot options. Adding `nomodeset` as a boot parameter tells Linux to use a very basic graphics driver, which can bypass a lot of kernel panic issues related to video.

6. Write the command to display the system's hostname? How to change hostname using `sysctl` command?

Displaying the hostname is one of the easiest commands out there.

To display:

Bash

`hostname`

To change it (temporarily): You can use `sysctl` to talk directly to the Linux kernel and change its `kernel.hostname` parameter.

Bash

```
sudo sysctl kernel.hostname=new-server-name
```

Important: Think of this like writing on a whiteboard. It works right now, but the change is **temporary** and will be gone the next time you reboot the system. For a permanent change, you need to edit the `/etc/hostname` file.

7. Which command is used to show the calendar of the year 1984 with August month?

You can use the `cal` command for this trip back in time! Just tell it the month and year you want to see.

Command:

Bash

```
cal 8 1984
```

This will display a nicely formatted calendar for just August 1984.

8. Write a command to display system uptime and logged-in users together.

While `uptime` gives you the uptime, there's an even better command that shows you both at once: `w`.

Command:

Bash

```
w
```

The very first line of the output from `w` is the same as the `uptime` command. Below that, it gives you a handy list of who is logged in, where they're connecting from, and what they are currently doing. It's the perfect all-in-one command for a quick system status check.

9. Use the `find` command to list all “.c” files in `/home/user`.

The `find` command is like a search dog for your filesystem. You just need to tell it where to look and what to look for.

Command:

Bash

```
find /home/user -type f -name "*.c"
```

Let's break that down:

- **find /home/user:** Tells the "dog" to start searching in the /home/user directory.
 - **-type f:** Narrows the search to only look for files (and ignore directories).
 - **-name "*.c":** This is the scent! It tells it to find anything whose name ends with .c. The quotes are a good habit to prevent the shell from getting confused.
-

10. How do you change file permissions to allow only the owner to read and write?

This is the "Top Secret" or "For Your Eyes Only" permission setting. You want the owner to have full access but lock everyone else out completely.

The magic number for this is **600**.

Command:

Bash

```
chmod 600 your_secret_file.txt
```

Here's why it works:

- The **owner** gets a 6 (which is Read(4) + Write(2)).
 - The **group** gets a 0 (no permissions).
 - **Everyone else** also gets a 0 (no permissions).
-