Shell Scripting Notes – Beginner Level

Day 1 — Fundamentals of OS & Shell

1. Kernel

The **kernel** is the heart of any operating system.
It acts as a bridge between **hardware and software**, managing system resources.

Responsibilities of Kernel

1. **Device Management**
   Handles communication between OS and hardware devices.
2. **Memory Management**
   Allocates and tracks usage of system memory.
3. **Process Management**
   Handles process scheduling, creation, termination.
4. **System Handling/Protection**
   Ensures controlled execution and access.

2. System Libraries

Libraries provide reusable functions to applications.
Example:
• **lib C** provides system functions used by C, Java, Python, etc.

3. Operating System Stack

```
User Applications / Compilers / System Software
-------------------------------
System Libraries
-------------------------------
Kernel
-------------------------------
Operating System
```

4. Shell

A **shell is a command interpreter** — the way users communicate with the OS.
Examples:
• bash
• sh
• dash
• ksh
Popular Linux Distros:
• CentOS
• Fedora
• Ubuntu

Shell Commands – Basics

Command	Description
ls	List files/directories
pwd	Show present working directory
cd	Change directory
cd bundle	Go into *bundle* directory
cd ..	Go one level back
cd ../..	Go two levels back
cd ubuntu/bundle	Directly change path
ls -ltr	List files with timestamps
touch kk	Create an empty file named *kk*
vi kkk	Create/open file *kkk* in editor
cat kk	View file contents
mkdir kk	Create folder kk
rm kk	Delete file kk
rm -r kk	Delete directory kk recursively
rmdir kk	Remove empty directory
free	Check memory usage
free -g	Display memory in GB
nproc	Check number of CPUs
df -h	Show disk usage
top	Display real-time system info

Using Vi editor

1. **vi filename** → open file
2. Press **i** → insert mode (typing allowed)
3. Press **Esc** → exit insert mode
4. Type **:wq!** → save & exit
5. Type **:q!** → exit without saving

Day 2 — Shell Scripting Introduction

Why Automation?

Automation reduces **manual effort** and **repetitive tasks**.
Shell scripting is a major automation tool.

Creating Files & Listing

```
touch first-shell-script.sh   # create file
ls                           # list files
ls -ltr                      # list with timestamp
```

Manual Pages (Help System)

```
man ls
man touch
man <command>
```

Exit man using:

```
:q!
```

Writing Your First Shell Script

1. Open file:

```
vi first-shell-script.sh
```

2. Enter insert mode and write:

```
#!/bin/bash
echo "my name is karthik"
```

3. Save and exit **:wq!**.

How to View Script Content

```
cat first-shell-script.sh
```

Executing the Script

Method 1:

```
sh first-shell-script.sh
```

Method 2:

```
./first-shell-script.sh
```

Permission Issue
If permission denied occurs, apply execute permissions.

File Permissions & chmod
Linux security relies on access control:

```
chmod 777 filename
```

Meaning of 777:
- First 7 → User (Owner)
- Second 7 → Group
- Third 7 → Others
  Each 7 = 4(read) + 2(write) +1(execute)

Examples:

Command	Meaning
chmod 444 file	Read-only for all
chmod 755 file	Owner full, others read+execute
chmod 777 file	Full permissions to everyone

History Command

```
history     # shows previously used commands
```

Day 3 — Simple Script Exercise

Step 1: Create a Working Directory

```
mkdir myfirst-shell-scripting-folder
cd myfirst-shell-scripting-folder
```

Step 2: Create Script File

```
vi sample-shell-script.sh
```

Write Script:

```
#!/bin/bash

# create a folder
mkdir karthik

# creates files
touch firstfile secondfile
```

Save using **:wq!**

Step 3: Grant Execute Permission

```
chmod 777 sample-shell-script.sh
```

Step 4: Run Script

```
./sample-shell-script.sh
```

After execution:
- A directory **karthik** will be created.
- Two files **firstfile** and **secondfile** will be created.

Summary Table — Frequently Used Commands

Category	Command Examples
Directory	cd, mkdir, ls, pwd
File	touch, vi, cat, rm
Permissions	chmod
Monitoring	top, df -h, free, nproc
Others	history, man

Beginner Script Templates

Print Text Script

```
#!/bin/bash
echo "Hello, this is my first shell script"
```

File Creation Script

```
#!/bin/bash
mkdir demo
touch demo/file1 demo/file2
```

System Info Script

```
#!/bin/bash
echo "Memory Information:"
free -h

echo "CPU Information:"
nproc

echo "Disk Usage:"
df -h
```