LAB - 3
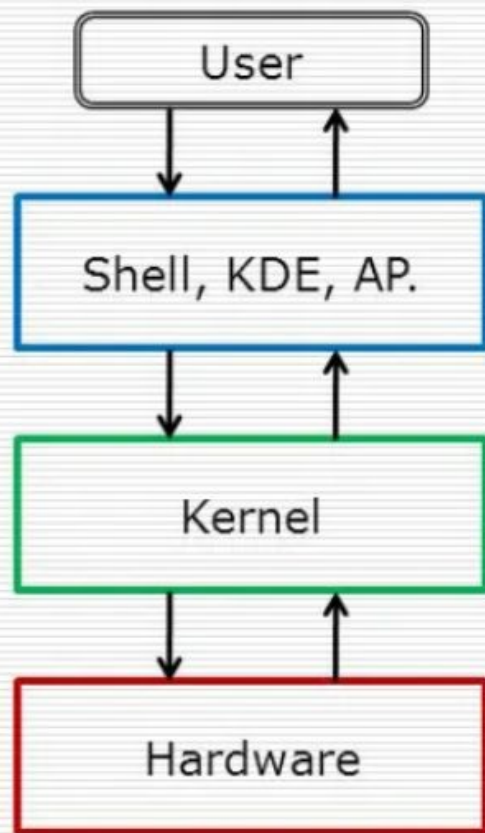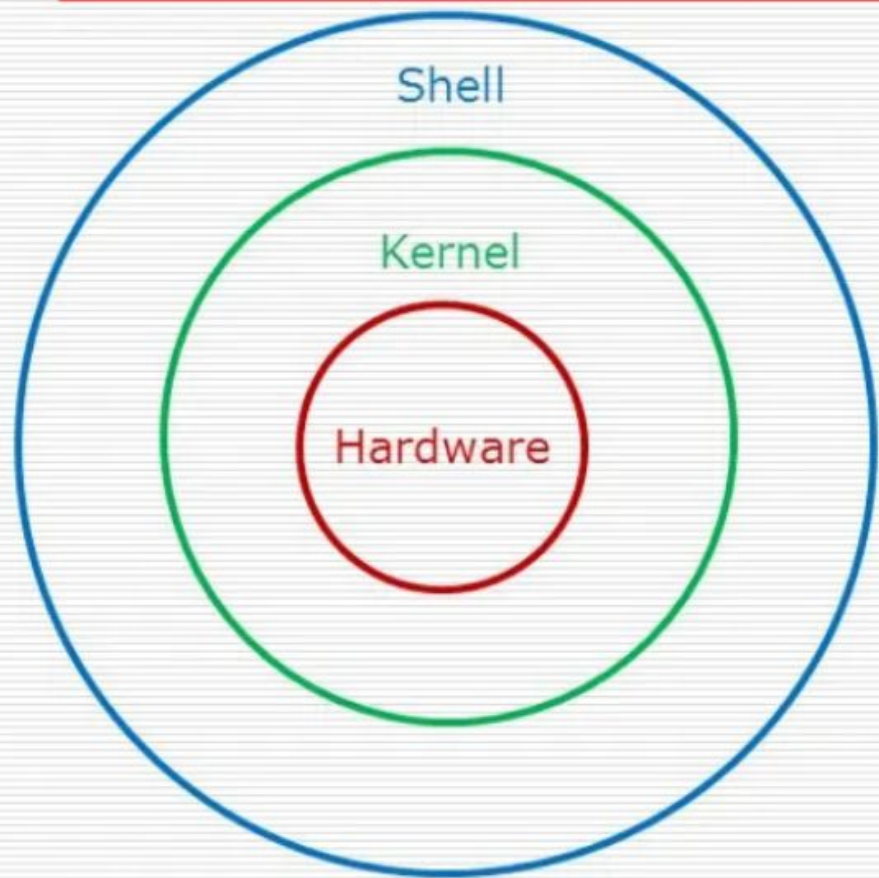
We can start guessing what a shell is ¿Can't we?

A shell is a *program working as the middle man between the Kernel and the user.*

It's a command line interpreter which receives inputs from the user and passes them onto the Kernel.

What is a Shell Program?

A **shell program** (or shell script) is a file containing a sequence of shell commands and control structures (like loops, conditionals, and functions). It is written to automate tasks that can be executed in a shell.

- It is essentially a script written in a **shell scripting language**, such as Bash, Zsh, or sh.
- Shell programs are interpreted by the shell (not compiled), meaning the shell reads and executes the script line by line.
- Commonly used for tasks like file manipulation, program execution, and text processing.

- **mkdir**: Creates directories.
- **chmod**: Changes file or directory permissions.
- **ls**: Lists directory contents.
- **cp**: Copies files or directories.

Scripting Commands

Examples:

- **if**, **for**, **while**: Control structures for logic and looping.
- **echo**: Outputs text (also works in the terminal, but widely used in scripts).
- **read**: Accepts user input.
- **exit**: Ends a script or program.

Scenario: Automated Backup System

Imagine a scenario where you need to back up a specific directory (`/home/user/documents`) to another location (`/home/user/backup`) every day. Instead of manually copying the files daily, you can write a **shell script** to automate the process.

Shell Script to create a FOLDER

Create a new script file. Let's name it create_folder.sh:

nano create_folder.sh
Paste the following script into the editor:

```
#!/bin/bash

# Define the Desktop path
DESKTOP_PATH="$HOME/Desktop"

# Define the folder name you want to create
FOLDER_NAME="my_new_folder"

# Create the folder on the Desktop
mkdir -p "$DESKTOP_PATH/$FOLDER_NAME"

# Print a confirmation message
echo "Folder '$FOLDER_NAME' created on Desktop."

# List the contents of the Desktop to verify
echo "Contents of Desktop:"
ls "$DESKTOP_PATH"
```

In nano, press Ctrl + O, press Enter to save, and then press Ctrl + X to exit.

MAke the script Executable

`chmod +x create_folder.sh`

Run the script.

`./create_folder.sh`

Verify the folder

`ls $HOME/Desktop`

```
This message is shown once a day. To disable it please create the
/home/kiit/.hushlogin file.
kiit@KIIT-107615:~$ nano script1.sh
kiit@KIIT-107615:~$ chmod +x demo_script.sh
chmod: cannot access 'demo_script.sh': No such file or directory
kiit@KIIT-107615:~$ chmod +x script1.sh
kiit@KIIT-107615:~$ ./script1.sh
Enter your name:
bj
Hello, bj!
Creating a file called demo.txt...
Contents of demo.txt:
This is a demo file created by bj.
Cleaning up...
Done!
kiit@KIIT-107615:~$
```

Take Back up Regularly

```bash
#!/bin/bash

# Define the Desktop path
DESKTOP_PATH="$HOME/Desktop"

# Define the directory names
DIR1="dir1"
DIR2="dir2"

# Define a sample file to create and copy (e.g., a text file)
SAMPLE_FILE="$DIR1/sample.txt"

# Create the two directories on the Desktop
mkdir -p "$DESKTOP_PATH/$DIR1" "$DESKTOP_PATH/$DIR2"

# Create a sample file inside dir1
echo "This is a sample file for copying." > "$DESKTOP_PATH/$DIR1/sample.txt"

# Copy the file from dir1 to dir2
cp "$DESKTOP_PATH/$DIR1/sample.txt" "$DESKTOP_PATH/$DIR2/"

# Print a confirmation message
echo "Directories '$DIR1' and '$DIR2' created."
echo "File 'sample.txt' copied from '$DIR1' to '$DIR2'."

# List the contents of the Desktop to verify
echo "Contents of Desktop:"
ls "$DESKTOP_PATH"
```

```bash
#!/bin/bash

# Demonstrating basic mechanisms in a shell script

# Reading input
echo "Enter your name:"
read name

# Writing output
echo "Hello, $name!"

# Creating a file
echo "Creating a file called demo.txt..."
echo "This is a demo file created by $name." > demo.txt

# Reading the file
echo "Contents of demo.txt:"
cat demo.txt

# Cleaning up
echo "Cleaning up..."
rm demo.txt
echo "Done!"
```

```bash
#!/bin/bash

# Function to check if a number is prime
is_prime() {
        local num=$1

        if [ "$num" -le 1 ]; then
        return 1 # Not prime
        fi

        for ((i = 2; i <= num / 2; i++)); do
        if ((num % i == 0)); then
        return 1 # Not prime
        fi
        done

        return 0 # Prime
}

# Main script
echo "Enter a number:"
read number

if is_prime "$number"; then
        echo "$number is a prime number."
else
        echo "$number is not a prime number."
fi
```