1. Connecting Windows to Ubuntu

If SSH isn't available on Windows, you can use **PowerShell**, virtual machines, or alternative tools to connect. Here's the process:

Commands:

1. **Start a virtual machine:**
2. `Start-VM -Name "UBUNTU" -Passthru`
    - **What it does:** Starts an Ubuntu VM if you're using Hyper-V on Windows. The `-Passthru` flag ensures that the VM's status is displayed.
    - **Example:** If your Ubuntu VM is named `UBUNTU`, this command powers it on.
3. **Connect via SSH (if supported):**
4. `ssh user@ip`
    - **What it does:** Connects to the Ubuntu server using the SSH protocol. Replace `user` with your Ubuntu username and `ip` with the VM's IP address.
    - **Example:**
    - `ssh john@192.168.1.10`

File Transfer:

- **Sending files to Ubuntu:**
- `scp act.txt user@ip:/home/user/`
    - **What it does:** Copies the file `act.txt` from your Windows system to `/home/user/` on the Ubuntu server.
    - **Example:**
    - `scp C:\Users\John\act.txt john@192.168.1.10:/home/john/`
- **Retrieving files from Ubuntu:**
- `scp user@ip:/home/user/acting.txt .`
    - **What it does:** Copies `acting.txt` from the Ubuntu server to the current directory on your Windows system.
    - **Example:**
    - `scp john@192.168.1.10:/home/john/acting.txt C:\Users\John\`

---

2. AWK Executes on Specific Files

AWK processes text files and extracts or manipulates data based on patterns.

- **What:** AWK can be used to process specific files by passing filenames as arguments.
- **Why:** To handle specific data files based on your requirements.

Example:

```
awk '/error/' logs.txt
```

- Searches for lines containing "error" in the file `logs.txt`.

To process multiple files:

```
awk '/error/' file1.txt file2.txt
```

- Processes `file1.txt` and `file2.txt` for the pattern `error`.

---

3. Difference Between `tr` and `sed`

- **`tr`:**
  - **What:** A command to translate or delete characters.
  - **Example:**
  - `echo "hello world" | tr 'a-z' 'A-Z'`

    Output: `HELLO WORLD` (converts lowercase to uppercase).

- **`sed`:**
  - **What:** A stream editor for search, find-and-replace, or insertion.
  - **Example:**
  - `echo "hello world" | sed 's/world/Linux/'`

    Output: `hello Linux` (replaces "world" with "Linux").

---

4. Running a Shell Script

Shell scripts are executed with two main methods:

1. **Using `sh`:**
2. `sh hello_world.sh`
   - Runs the script using the `sh` shell interpreter.
3. **Direct execution:**
4. `./hello_world.sh`
   - Runs the script directly, provided it has execution permissions (`chmod +x hello_world.sh`).

---

5. Understanding Brackets in Shell Scripts

1. Single Brackets `[]`:

- **What:** Test or evaluate a condition in POSIX shell scripts.
- **Example:**
- `if [ -f file.txt ]; then`
- `    echo "File exists."`
- `fi`

2. Double Brackets `[[]]`:

- **What:** Extended test syntax in bash. Supports pattern matching and logical operators.
- **Example:**

```
if [[ "$var" == "value" ]]; then
    echo "Value matches."
fi
```

3. Double Quotes `""`:

- **What:** Used to preserve spaces in strings.
- **Example:**

```
name="John Doe"
echo "Hello, $name"
```

Output: `Hello, John Doe`.

Why Use These Commands Together?

1. **Automating Tasks:** Combine commands like SSH, SCP, AWK, and `sed` to automate tasks such as file processing, backups, and remote connections.
2. **Improved Efficiency:** Shell scripting with structured commands makes repetitive tasks faster and less error-prone.