Unix and Shell Scripting - Day 1

By Dhandapani Yedappalli Krishnamurthi Sep 29, 2025

1. Overview, History, Features

- **Overview**: Unix is a multitasking, multiuser operating system.
- **History**: Developed in 1969 at Bell Labs by Ken Thompson, Dennis Ritchie, and others.
- **Features**:
  - Multiuser, Multitasking
  - Portability (written in C)
  - Security & Permissions
  - Hierarchical File System
  - Tools for development and automation

🔹 2. Kernel, Shell, File System, User Space

- **Kernel**: Core of OS; manages CPU, memory, devices.
- **Shell**: Command-line interpreter (bash, ksh, csh).
- **File System**: Hierarchical tree (/, /home, /bin, /etc).
- **User Space**: Where user applications run (not kernel).

🔹 3. Basic Commands

```
pwd          # Print working directory
ls           # List files
cd /path     # Change directory
cp a.txt b.txt   # Copy file
mv a.txt b.txt   # Move/rename file
rm file.txt      # Remove file
mkdir newdir     # Create directory
rmdir olddir     # Remove empty directory
echo "Hello"     # Print message
man ls           # Manual for command
```

🔹 4. Hello World Script

```
# hello.sh
#!/bin/bash
echo "Hello World!"
```

Run:

```
chmod +x hello.sh   # make executable
./hello.sh
```

🔹 5. File System Concepts

- **Hierarchical structure**: `/` → root → `/bin`, `/etc`, `/home`, `/usr`
- **Navigation**: `cd`, `pwd`, `ls`
- **File types**: Regular file (`-`), Directory (`d`), Symbolic link (`l`), Device files (`b`, `c`)

🔹 6. Permissions & Ownership

```
chmod 755 file.sh    # rwxr-xr-x
chown user1 file.sh  # Change owner
chgrp dev file.sh    # Change group
stat file.sh         # Show details
```

🔹 7. File Handling

```
cat file.txt      # Show contents
touch new.txt     # Create empty file
wc file.txt       # Count lines/words/chars
more file.txt     # Scroll file (forward)
less file.txt     # Scroll file (both ways)
```

🔹 8. Redirection & Pipes

- **stdin**: standard input (keyboard)

- **stdout**: standard output (screen)
- **stderr**: error messages

```
ls > out.txt      # Redirect stdout to file
ls >> out.txt     # Append output
ls notafile 2> err.txt  # Redirect errors
cat file.txt | grep "word"   # Pipe output
```

🔹 9. Shell Types, Variables

- **Types**: bash, ksh, csh, zsh
- **System variables**: `$HOME`, `$PATH`, `$USER`
- **Set option**:

```
set -x   # Debug mode
set +x   # Disable debug
```

🔹 10. Process Management

```
ps -ef       # List processes
top          # Monitor CPU/memory
bg job_id    # Send job to background
fg job_id    # Bring job to foreground
kill 1234    # Kill process
nohup cmd &  # Run process ignoring hangups
jobs         # Show background jobs
```

🔹 11. Scheduling Jobs

```
crontab -e         # Edit cron jobs
# Run backup at 2 AM daily
0 2 * * * /home/user/backup.sh

at now + 5 minutes # Run a job 5 min later
```

🔹 12. Text Processing

```
grep "error" logfile     # Search text
head -5 file.txt         # First 5 lines
tail -10 file.txt        # Last 10 lines
tr a-z A-Z < file.txt    # Translate to uppercase
sort file.txt            # Sort lines
find /home -name "*.sh"  # Find files
```

🔹 13. sed (Stream Editor)

```
sed 's/error/warning/g' file.txt    # Replace
sed -n '1,5p' file.txt              # Print lines
1–5
```

🔹 14. awk (Pattern Scanning)

```
awk '{print $1, $3}' file.txt   # Print 1st & 3rd columns
awk '/error/ {print $0}' logfile  # Print lines with "error"
```

🔹 15. System Monitoring

```
top                # CPU & memory
vmstat 5           # System performance
df -h              # Disk usage
du -sh /var/log    # Size of directory
```

🔹 16. VI Editor Basics

- **Modes**: Command, Insert, Ex
- **Navigation**: `h` (left), `l` (right), `j` (down), `k` (up)
- **Input**: `i` (insert), `a` (append), `o` (new line)
- **Save & Quit**: `:wq` (save & quit), `:q!` (quit without save)

✅ Example Script – File Operations

```
#!/bin/bash
# file_ops.sh
echo "Creating test files..."
mkdir test_dir
cd test_dir
touch file1.txt file2.txt
echo "Hello World" > file1.txt
cp file1.txt file2.txt
ls -l
```

Run:

```
chmod +x file_ops.sh
./file_ops.sh
```

---

https://www.youtube.com/watch?v=j56eSh_DKNw