Bootcamp 134 | Python

Course 17 | Linux – Part 2

Amir Hossein Chegouniyan

Head of the Technical Team at Dariche Tejarat

Lecturer of Python – Django at Maktab Sharif

Amirhossein-chegounian

Content

System and User Commands

- ping: Test network connectivity.
- man: Display manual pages for commands (how to read and navigate man pages).
- cal and jcal: Display calendar and Julian calendar.
- who and whoami: Identify logged-in users and current user.

Basic Filters

- Purpose: Process text from files or input streams.
- Key Commands:
  - grep: Search for patterns in files.
  - tee: Redirect output to multiple locations (file and console).
  - tr: Translate or delete characters.
  - uniq: Remove duplicate lines from sorted data.
  - sort: Sort file contents alphabetically or numerically.
  - wc: Count lines, words, and characters in files.

File Compression

- gzip and gunzip: Compress and decompress files.
- tar: Archive multiple files into one (tar -cvf, tar -xvf).
- zip and unzip: Create and extract .zip files.

File Security

- Concepts: Importance of file permissions and encryption.
- Key Commands:
  - chmod: Change file permissions (rwx and numeric mode).
  - chown: Change file ownership.
  - umask: Default permissions for new files.

File Security | Chmod

chmod u+x file.sh

chmod g-w file.txt

chmod o+r file.txt

chmod a+x script.py

chmod 644 file.txt

- **r** → read
- **w** → write
- **x** → execute
- **u** → user
- **g** → group
- **o** → others

File Security | Chown

- chown stands for change owner.
- With this command you can change the owner or group of a file or folder.
  - chown [owner]:[group] file_name # change owner and group of file
  - chown [owner] file_name # change owner of file
  - chown :[group] file_name # change group of file
  - chown -R [owner]:[group] directory # change owner and group of all file of directory

Create new group command:
sudo groupadd group_name

File Security | Umask

- The umask or user file creation mask is a value that determines the default permissions of files and folders that a user creates.
  - umask # print current default permissions
  - umask 077 # change default permissions

What is Bash Scripting?

- Automating repetitive tasks by executing a sequence of commands in a script file (.sh).
- Explain the shebang line (#!/bin/bash).

Basic Syntax

- Variables: Define and use variables in a script.
- Control Structures:
  - if, elif, else: Conditional statements.
  - for, while: Loops for iteration.
- Input and Output:
  - Reading user input with read.
  - Writing output with echo.

Key Commands for Scripting

- #!/bin/bash: Specify interpreter.
- Variables: Assign and reference (VAR="value"; echo $VAR).
- if, then, else: Control flow based on conditions.
- Loops: Simple for and while examples.
- read: Get input from the user.
- case: Pattern matching for multiple conditions.

Key Commands for Scripting | Example 1

```
#!/bin/bash
# Specify interpreter

# --- Variables ---
VAR="Hello, Bash!"
echo "Variable VAR contains: $VAR"

# --- read ---
echo "Enter your name:"
read NAME
echo "Hi, $NAME!"
```

```
declare -a my_indexed_array
my_indexed_array=( "apple" "banana" "cherry" )
# or
my_indexed_array[0]="apple"
my_indexed_array[1]="banana"
my_indexed_array[2]="cherry"
```

Key Commands for Scripting | Example 2

```
# --- if, then, else ---
if [ "$NAME" == "Alice" ]; then
    echo "Welcome back, Alice!"
else
    echo "You are not Alice."
fi
```

```
# --- for loop ---
echo "For loop example:"
for i in 1 2 3; do
    echo "Iteration $i"
done
```

```
for item in "${my_array[@]}"; do
    echo "Current item: $item"
done
```

Key Commands for Scripting | Example 3

```
# --- while loop ---
echo "While loop example:"
COUNT=1
while [ $COUNT -le 3 ]; do
    echo "Count is $COUNT"
    COUNT=$((COUNT + 1))
done
```

Key Commands for Scripting | Example 4

```
# --- case ---
echo "Enter a number (1-3):"
read NUM
case $NUM in
    1) echo "You chose One." ;;
    2) echo "You chose Two." ;;
    3) echo "You chose Three." ;;
    *) echo "Invalid choice." ;;
esac
```

Any question?

Next course

- Introduction to Regex
- Python Regex Library (re)
- Hands-On Activity:
- Understanding Python Packages
- Working with Third-Party Libraries
- Introduction to Virtual Environments
- The Zen of Python (PEP 20)
- PEP 8 - Python Style Guide
- Advanced Function Concepts