School Of Engineering

Linux Programming
Assignment-8

Name: Neehara Lakshmi
USN No: ENG24CY0138
Roll No: 55
Section: B (CyberSecurity)
Semester: 3rd

Q1. What is a user-defined function in shell scripting? Explain with an example.

Ans: A user-defined function is a reusable block of code that you create to perform specific tasks, making scripts more organised and maintainable.

Syntax:

```
function_name() {
    commands
}
```

Example Code: #Function to Check File Existence

```
#!/bin/bash
# Define function
check_file() {
    if [ -f "$1" ]; then
        echo "File $1 exists"
    else
        echo "File $1 not found"
    fi
}

# Call function
check_file "script.sh"
check_file "nonexistent.txt"
```

Output:

```
File script.sh exists
File nonexistent.txt not found
```

Q2. Write a bash script with a function that multiply two integer numbers.

Ans:

Example Code:

```
#!/bin/bash
# Function to multiply two integers
multiply() {
    local result=$(( $1 * $2 ))
    echo $result
}
# Main script
echo "Enter first number:"
read num1

echo "Enter second number:"
read num2
```

```
# Call function and store result
product=$(multiply $num1 $num2)

# Display result
echo "The product of $num1 and $num2 is: $product"
```

Output:

```
Enter first number:
6
Enter second number:
7
The product of 6 and 7 is: 42
```

Q3. Explain how arrays (1D, 2D, and 3D) are declared in bash scripting.

Ans: Arrays in Bash Scripting

1D Arrays (One-Dimensional)

Declaration:

Example Code:

```
# Method 1: Explicit declaration
declare -a fruits=("apple" "banana" "cherry")
# Method 2: Direct assignment
colors=("red" "green" "blue")
# Method 3: Index-based assignment
fruits[0]="apple"
fruits[1]="banana"
fruits[2]="cherry"
```

2D Arrays (Simulated)

Bash doesn't have true 2D arrays, but we can simulate them:

Declaration:

Example Code:

```
declare -A matrix
matrix[0,0]="a"
matrix[0,1]="b"
matrix[1,0]="c"
matrix[1,1]="d"
```

3D Arrays (Simulated)

Similarly, we can simulate 3D arrays:

Declaration:

Example Code:

```
declare -A cube
cube[0,0,0]="x"
cube[0,0,1]="y"
cube[1,1,1]="z"
```

Q4. Write a shell script to display elements of an array.

Ans:

Example Code:

```
#!/bin/bash
# Define an array
my_array=("Apple" "Banana" "Cherry" "Date")
# Display all array elements
echo "Array elements:"
for element in "${my_array[@]}"; do
    echo "$element"
done
```

Q5. What is the purpose of cron in Linux?

Ans:

Purpose of Cron in Linux:

- Cron automates scheduled tasks on Linux systems.
- It runs commands at specified times without manual intervention.
- Cron helps schedule repetitive tasks like backups and updates.
- It executes system maintenance jobs automatically.
- Cron allows tasks to run during off-peak hours to reduce system load.
- It uses a time-based scheduling syntax with five time fields.
- Cron runs as a background daemon called crond.
- Users can create personal schedules using crontab files.
- System-wide cron jobs are stored in /etc/cron.* directories.
- Cron sends email notifications about job outputs unless configured otherwise.

Q6. Write a cron job to run a backup script every day at midnight.

Ans:

Example Code:

```
0 0 * * * /path/to/backup_script.sh
```

Explanation of the cron syntax:

- ‘ 0 ’- Minute (0 = on the hour)
- ‘ 0 ’- Hour (0 = midnight)
- ‘ * ’- Day of month (every day)
- ‘ * ’ - Month (every month)
- ‘ * ’ - Day of week (every day)

Q7. How do you schedule a one-time job using at command?
Ans:
Example Code:
echo "command_to_run" | at HH:MM

Q8. Write a script to display disk usage using df and du
Ans:
Example Code:
#!/bin/bash
echo "=== Disk Space Usage (df -h) ==="
df -h
echo -e "\n=== Directory Sizes in Current Path (du -sh *) ==="
du -sh *

This script shows:
‘ df -h ’ - Disk space usage for all mounted filesystems in human-readable format.
‘ du -sh * ’- Sizes of all files/directories in current location in human-readable format.

Q9. How can you log the output of a script using the tee command?
Ans :
Example Code:
./script.sh | tee output.log
Explanation: Displays output on screen AND saves copy to file simultaneously

Q10. Explain with an example how shell scripting can automate system administration tasks.
Ans:
Example Code:
#!/bin/bash
# Automated backup and system monitoring
BACKUP_DIR="/backups"
LOG_FILE="/var/log/auto_admin.log"

Automated backup
tar -czf $BACKUP_DIR/backup_$(date +%F).tar.gz /home /etc 2>/dev/null

Auto-cleanup of temp files
find /tmp -type f -mtime +7 -delete

Service status check
systemctl is-active --quiet apache2 || systemctl restart apache2

```
# Disk monitoring
[ $(df / --output=pcent | tail -1 | tr -d '% ') -gt 90 ] && echo "Disk critical" >> $LOG_FILE
```

- **Disk Monitoring** - Automatically logs disk usage without manual checks.
- **Service Management** - Detects and restarts failed services automatically.
- **File Cleanup** - Removes old temporary files on a schedule.
- **Health Reporting** - Creates logs for historical tracking and auditing.

THANK YOU