

Linux Programming - Assignment 8

Date: 11-10-2025

Name – Samarth Kale

Batch - 3C

Roll no. - 57

USN - ENG24CY0154

1. Q11. What is a user-defined function in shell scripting? Explain with an example.

A user-defined function groups commands under a name to be called later; it improves modularity and reuse. Syntax: `function_name() { commands; }` Example below:

```
#!/bin/bash
greet() {
    echo "Hello, $1"
}
# call
greet "Samarth"
```

2. Q12. Write a bash script with a function that multiply two integer numbers.

Function returns or prints product; use arithmetic expansion.

```
#!/bin/bash
multiply() {
    local a=$1
    local b=$2
    echo $((a * b))
}
result=$(multiply 6 7)
echo "Product = $result"
```

3. Q13. Explain how arrays (1D, 2D, and 3D) are declared in bash scripting.

Bash supports indexed and associative 1D arrays natively. Multidimensional arrays are simulated by using indices or nested naming.

Examples:

- 1D: `arr=(1 2 3)` or `arr[0]=A`
- 2D (simulated): use combined keys like `arr[0,0]=val` or index math: `arr[${i*COL + j}]`
- 3D: further index encoding or associative arrays with composite keys.

4. Q14. Write a shell script to display elements of an array.

Loop over indices or use `${arr[@]}` to expand all elements.

```
#!/bin/bash
arr=(123 "Abs" -2.3 A 23.56 0)
for i in "${!arr[@]}"; do
    echo "arr[$i] = ${arr[$i]}"
done
# or echo all
# echo "${arr[@]}"
```

5. Q15. What is the purpose of cron in Linux?

cron is a time-based job scheduler to run tasks automatically at specified times or intervals. It uses crontab files to configure scheduled commands, useful for backups, maintenance, and recurring jobs.

6. Q16. Write a cron job to run a backup script every day at midnight.

Add this line to the crontab (edit with `crontab -e`):

```
# run backup.sh at 00:00 daily
0 0 * * * /path/to/backup.sh >> /var/log/backup.log 2>&1
```

7. Q17. How do you schedule a one-time job using at command?

Use `at` with a time and provide commands via stdin or a file. Example:

```
# schedule 'backup.sh' to run tomorrow at 02:30
echo "/path/to/backup.sh" | at 02:30 tomorrow
# or
at 02:30 tomorrow
> /path/to/backup.sh
> <EOT>
```

8. Q18. Write a script to display disk usage using `df` and `du`.

`df` shows filesystem-level usage; `du` shows directory/file sizes. Combine both for overview.

```
#!/bin/bash
# filesystem usage
echo "Filesystem usage (df -h):"
df -h

echo "\nTop disk consumers (du):"
du -h --max-depth=1 | sort -hr | head -n 10
```

9. Q19. How can you log the output of a script using the `tee` command?

`tee` duplicates stdout to a file while preserving display. Example: `./script.sh | tee script.log` will show output and save it. Use `'tee -a'` to append. Inside script, redirect commands: `some_command |& tee -a logfile` to capture stderr too.

10. Q20. Explain with an example how shell scripting can automate system administration tasks.

Shell scripts chain commands to automate tasks such as backups, user provisioning, monitoring, and log rotation. Example: a script that compresses `/var/log`, uploads to remote storage, and rotates logs can run nightly via cron, eliminating manual steps.