

## UNIX/Linux Operating Systems – Written Exam

Duration: 1 hour

Total Points: 20

**NAME:**

**Group:**

### Section A – Theoretical Questions (8 points)

1. (2 pts) Explain the architecture of a typical Linux operating system. Include the roles of each element.

**A typical Linux architecture consists of four layers:**

- **Hardware:** The physical components like CPU, memory, and storage.
- **Kernel:** The core component that manages hardware, memory, processes.
- **Shell:** Interface between user and kernel, interprets commands.
- **Applications:** User programs such as editors, compilers, browsers.

2. (2 pts) List and describe four essential Linux shell commands. Include what each one does

- **First command: ls:** Lists files and directories.
- **Second command: pwd:** Prints the current working directory
- **Third command: cd:** Changes the current directory
- **Fourth command: mkdir:** Creates a new directory.

3. (1 pt) What the difference between prompt, terminal, shell, and command line?

- **Shell:** Command-line interpreter that executes user commands.
- **Prompt:** Text shown to indicate readiness for input.
- **Terminal:** Interface to interact with the shell.
- **Command Line:** The area where users type commands.

4. (1 pt) How can 'stdout' and 'stderr' be redirected in Linux?

- **stdout:** > or >>
- **stderr:** 2>

5. (2 pts) Explain what a pipe (|) does in a command. Give an example of a useful pipeline with at least two commands and explain your example.

**A pipe (|) passes the output of one command as input to another.**

**Example: `cat file.txt | grep error`**

**This command shows lines in file.txt that contain 'error'.**

## **Section B – Command Analysis & Short Tasks (12 points)**

6. (1 pt) You are in a directory containing multiple .txt files. Write a command that: - Creates a subdirectory named texts - Moves all .txt files into that subdirectory.

**`mkdir texts && mv *.txt texts/`**

7. (1 pt) What does the following command do?

`ls /etc | grep conf | wc -l`

**Lists all files in /etc with 'conf' in the name and counts them.**

8. (2 pt) Write a command that lists the last 10 lines of a file called 'system.log' and counts how many of them contain the word 'ERROR'.

**Sol 1: `tail -n 10 system.log | grep -c ERROR`**

**Sol 2: `tail -n 10 system.log | grep 'ERROR' | wc -l`**

9. (1 pt) Write a shell command to extract and sort (alphabetically) all usernames from '/etc/passwd'.

**`cut -d: -f1 /etc/passwd | sort`**

10. (2 pts) You have a file called 'logs.txt'. Write a pipeline (|) that extracts all lines containing the word 'fail' (case-insensitive), shows only the first two words of each line, removes duplicates, and sorts the result in reverse order.

**Sol1: `grep -i 'fail' logs.txt | cut -d' ' -f1,2 | sort -r | sort`**

**Sol 2: `grep -i fail logs.txt | awk '{print $1, $2}' | sort | uniq | sort -r`**

- **awk is a powerful text-processing tool in Unix/Linux.**
- **It treats each line as a record and splits it into fields based on whitespace or a specified delimiter.**
- **{print \$1, \$2} tells awk to print the first and second fields.**

11 (1pts)

Create a script 'hello.sh' that:

- Ask the user for their name
  - reads the input
  - print "Hello your\_name" where your\_name is the name provided by the user
- Write the full script below:

```
#!/bin/bash
echo "What is your name?"
read your_name
echo "Hello $your_name."
```

12. (4 pts) Shell script

Create a script that:

- Asks the user to enter three file names.
- Assuming that all the files exist:
  - Find the number of lines in each file.
  - Use if statements to determine which file has the highest line count.
  - Display the name of the file with the most lines and how many lines it contains.

Write the full script below:

```
#!/bin/bash
echo "Enter the first file name:"
read file1
echo "Enter the second file name:"
read file2
echo "Enter the third file name:"
read file3
lines1=$(wc -l < "$file1")
lines2=$(wc -l < "$file2")
lines3=$(wc -l < "$file3")
if [ "$lines1" -ge "$lines2" ] && [ "$lines1" -ge "$lines3" ]; then
    echo "The file with the most lines is '$file1' with $lines1 lines."
elif [ "$lines2" -ge "$lines1" ] && [ "$lines2" -ge "$lines3" ]; then
    echo "The file with the most lines is '$file2' with $lines2 lines."
else
    echo "The file with the most lines is '$file3' with $lines3 lines."
fi
```