Faculty Initial: APA
Dept. of CSE
School of Engineering
CSE 362 Operating Systems Lab
Lab Tasks

Tasks

1 Hello World Tasks

1. Print your full name and current date in one line.
2. Print a message with a tab between two words.
3. Print three lines using one echo command.
4. Print your username using a variable.

2 cd Tasks

1. Go to /tmp, print path, return home.
2. Create a folder practice in your home, navigate into it.
3. From practice, go to parent, then back using .. and ..

3 ls Tasks

1. List all files (including hidden) in long format.
2. List only directories in /usr.
3. Show files in /var/log sorted by size.
4. Find all .conf files in /etc.

4 mkdir Tasks

1. Create: projects/python/web in one command.
2. Create three folders: backup1, backup2, backup3 with verbose output.

5 cp Tasks

1. Copy all .txt files from docs/ to backup/.
2. Copy config/ directory recursively to /tmp.
3. Try overwriting a file with and without -i.

6 rm Tasks

1. Create a file, delete it interactively.
2. Make a directory trash, put 3 files, delete recursively with verbose.
3. Delete all .tmp files in /tmp (safely!).

7 mv Tasks

1. Rename report.txt → final_report_v1.txt.
2. Move all .png images to images/ folder.
3. Try moving a file to existing name with/without -i.

8 pushd/popd Tasks

1. Navigate: home → /usr → /bin → /tmp using pushd.
2. Use popd twice to return.
3. Print directory stack.

9 touch Tasks

1. Create three empty files: a.txt, b.txt, c.txt.
2. Change only the access time of a.txt to now.
3. Create a file with timestamp 1 Jan 2025 12:00.

10 chown Tasks

1. Change owner of log.txt to alice.
2. Recursively change group of web/ to www-data.

11 sort Tasks

1. Sort /etc/passwd by UID (field 3).
2. Sort scores.txt numerically in descending order.
3. Remove duplicate lines from list.txt.

12 su / sudo Tasks

1. Switch to root with login shell.
2. Install htop using sudo.
3. Run whoami as user testuser.

13 ps Tasks

1. List all processes of current user.
2. Show full command line of PID 1234.
3. Count how many bash processes are running.

14 kill / killall Tasks

1. Gracefully stop process with PID 4321.
2. Force-kill a stuck python script.
3. Restart all apache2 processes.

15 vim Tasks

1. Open /etc/hosts in vim and go to line 10.
2. Create a new file hello.c and write a main() function.

16 nano Tasks

1. Edit /.bashrc with nano and add an alias.
2. Create a new file notes.txt and write three lines.

17 ping Tasks

1. Ping 8.8.8.8 four times.
2. Ping your router with 0.2 s interval.

18 traceroute Tasks

1. Trace route to cloudflare.com without DNS.
2. Limit hops to 10 for a local server.

19 find Tasks

1. Find all .log files modified in the last 2 days.
2. Locate every Makefile under /usr/src.
3. Delete all *.tmp files in /tmp (use -delete).

20 clear Tasks

1. Clear the screen after a long ls -R.

21 history Tasks

1. Show the last 5 commands.
2. Re-run the command that created myproject.

22 User Management Tasks

1. Add user testuser with home directory.
2. Add testuser to sudo group.
3. List all users.
4. Remove testuser and their home directory.

23 chmod Tasks

1. Create runme.sh, make it executable for owner only.
2. Make data.txt readable/writable by owner, read-only for others.
3. Give group execute permission on a directory.

24 chown/chgrp Tasks

1. Change owner of log.txt to alice.
2. Recursively change group of web/ to www-data.

25 File Viewing Tasks

1. Show first 3 lines of /etc/hosts.
2. Follow live changes in auth.log.
3. Count lines in passwd.
4. Find all lines containing bash in /etc/passwd.
5. Search recursively for main() in .c files.

26 I/O Redirection Tasks

1. Save ls -l output to listing.txt.
2. Append current date to log.txt.
3. Run a failing command → redirect error only.
4. Combine stdout and stderr into one file.

27 Pipelines Tasks

1. List all .sh scripts in /bin, count them.
2. Show top 5 largest files in current directory.
3. Extract unique IP addresses from access.log.
4. Find all users with /bin/bash shell from /etc/passwd.

28 Final Challenge: Mini File Manager Script

Create myfm.sh that does the following:

1. Creates a directory myproject
2. Creates 3 files inside
3. Copies one to backup/
4. Lists contents with permissions
5. Changes permission of one file to 700
6. Appends system info to log.txt
7. Displays log using less

29 What is Shell Script?

1. Explain in your own words the difference between interactive commands and a shell script.
2. Think of a repetitive task you do in the terminal that could be scripted.

30 Why Shell Script?

1. List two personal tasks you could automate with scripts.
2. Describe how a script could save time in system admin.

31 How to Write and Execute?

1. Create an empty script file named test.sh using nano.
2. Make it executable and run it (even if empty).
3. Try running without chmod – note the error.

32 Shell Script Format

1. Write a script with shebang and three comments.

2. Add inline comments to explain echo commands.
3. Run and verify comments are ignored.

33 A Sample Shell Script

1. Write and run the sample script.
2. Modify to include pwd and date.
3. Add comments explaining each line.

34 Variables

1. Print $USER and $PATH in a script.
2. Create user var os=Linux, echo it.
3. Assign numeric value, echo as is and +1.
4. Test case sensitivity: Var=1, var=2.

35 Variables (Continued)

1. Assign a variable with spaces using quotes.
2. Demonstrate double vs single quotes.
3. Escape $ to print literally.

36 Read User Input

1. Write script to read name and greet.
2. Read two numbers and echo their sum using expr.
3. Use read -p for prompt.

37 Shell Arithmetic

1. Compute 15 % 4 using expr.
2. Increment a variable by 10.
3. Check if 8 ¿ 5.

38 If-Else

1. Write script to compare two numbers read from user.
2. Check if a file exists.
3. Create age classifier (child/teen/adult).

39 Case Statement

1. Create menu: 1 → List files, 2 → Show date, else → Invalid.
2. Handle file extensions (.txt, .sh, etc.).

40 Command Line arguments

1. Print number of arguments.
2. Print first and second argument.
3. Loop through all arguments and number them.

41 For loop

1. Loop over ls output.
2. Print numbers 1 to 20 using C-style for.
3. Loop over command line arguments.

42 While loop

1. Write password checker with while.
2. Count from 1 to 10 using while.

43 Until loop

1. Rewrite password checker using until.
2. Countdown from 10 to 1.

44 Functions

1. Define and call simple function.
2. Pass argument to function.

45 Functions (continued)

1. Create function to add two numbers.
2. Use yes_or_no function.

46 Final Challenge: Simple Backup Script

Create backup.sh that:

1. Takes one directory as argument.
2. Checks if argument is given.
3. Creates dated backup folder.
4. Copies all files from source.
5. Logs actions.