Operating Systems
CS-2006 (Spring 25)
Assignment-1

Submission Date 26th February 2025

Instructions:

- *This is an individual Assignment.*
- ***All parties involved in any kind of plagiarism/cheating (even in a single line of code) will be given zero marks in all the assignments.***
- ***Assignment deadline won't be extended***
- ***Late submissions will be discarded so submit your assignment on-time***
- *You must follow below submission guidelines, otherwise your submission won't be accepted*
    - *create a new directory*
    - *change its name to the following format*
      YOURSECTION_ROLLNUMBER_NAME.
      E.g. A_20I-0012_Muhammad Ahamd
    - *put all your files (.cpp & .sh & .txt only) into this newly created directory*
    - *Compress the directory into a compressed .zip file*
    - *Submit it on google classroom*
- *Use good programming practices (well commented and indented code; meaningful variable names, readable code etc.).*
- *Understanding the problem is also part of the assignment.*

QUESTION 1

Scenario:
Your **personal PC is running low on disk space**, and you need an automated cleanup script. The script should:

1. **Identify the top 10 largest files** in the home directory and prompt the user whether to delete them.

2. **Detect duplicate files** across directories and provide options to:
    - **Delete** duplicates.
    - **Hard-link** duplicates to save space.
    - **Move duplicates** to a separate folder.
3. **Clean unnecessary system files**, including:
    - Cache files (.cache/, browser temp files).
    - Old logs (/var/log/ if run as root).
    - Large unused packages (apt autoremove or equivalent for package managers).

Bonus Challenge: Implement a feature where the script **suggests** which files can be deleted based on their last access time (atime).

QUESTION 2

Write a script that displays a main menu and perform tasks based on the input value.

Valid input values = {1, 2, 3, 4, exit}.

The different options 1,2,3,4 will display the output as follow:

1. Input a filename from user and display permissions of that particular file. Then invert the permissions e.g. If permissions were r-x change them to -w-. Then again

display the updated permissions of that file.

2. Input a filename and a string and search it in the file. Output the lines of file where that string is found. But if the string contains a dot(.) it means any character can

fill the place. For example:

string = c.t = {cat, cot, c t,}

3. Create a file dummy.txt and add the content of all the files in the current directory to dummy. But copy the content in such a way that if files in current directory = {f1, f2, f3, f4, ..., fn}. Then copy first N lines of files at even location {f2, f4,.} and last N lines of files at odd location {f1,f3,. }. Input value of N from user.

4. Input a filename from user and check modified date of that file. If modified date is greater than 24 hours from the current time change the modified date to current date. Along

with displaying the output on terminal, maintain a log file that contains the information of the script. Format of the log file is given below:

Format of the log File

Option 01 selected at *date and time*
File name: *filename.txt*
Permissions of *filename.txt*: *Show permissions*
Permissions changed
Updated Permissions of *filename.txt*: *Show permissions*

Option 02 selected at *date and time*
Filename: *filename.txt*
String: *string*
Output all the lines in *filename* where *string* is found.

Option 03 selected at *date and time*
Files at odd location: *f1, f3, f5*
Files at even location: *f2, f4*
Dummy.txt is created and *N* lines of each file copied in it.

Option 04 selected at *date and time*
Filename: *filename.txt*
Current modified date: *date*
Modified time updated or not

Option exit
Script terminated at *date and time*.

Question 3

Scenario:
You want to track **how much time you spend on different applications** on your PC daily. The script should:

1. **Monitor which application/window is active** every minute.
2. **Log the time spent per application** (e.g., Browser: 2 hours, VS Code: 3 hours, VLC Player: 30 mins).
3. **Generate a daily/weekly report** of application usage.
4. **Provide an alert** if a non-productive app (e.g., social media, YouTube) is used for more than 1 hour.

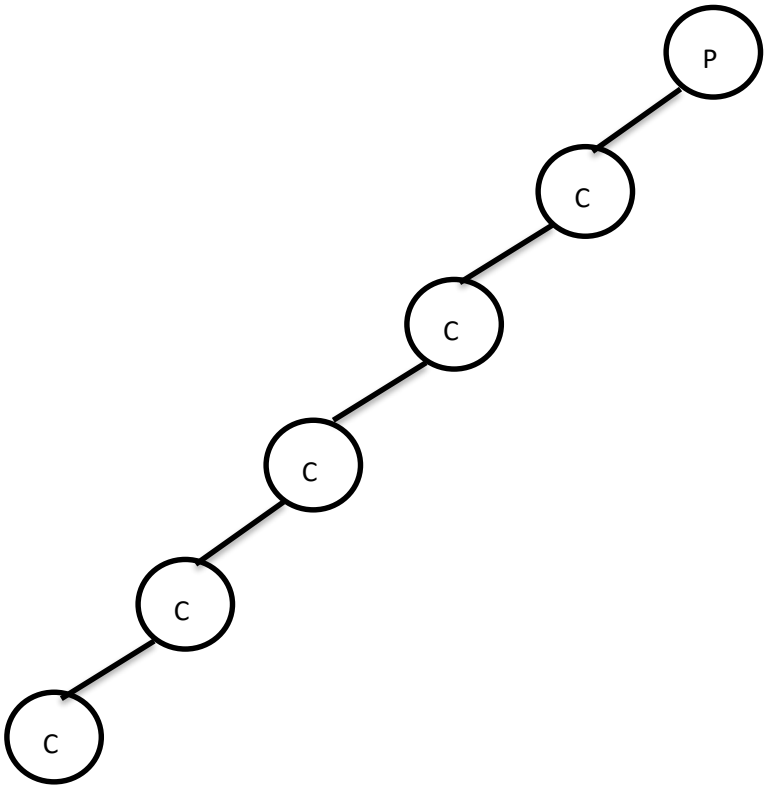
Bonus Challenge: Integrate an option to **block** apps after exceeding a set time limit.

QUESTION 4

Write a program which take string from user and you have to do following tasks with the string.

1. Reverse
2. Find Length
3. Add 2 in the ASCII of each character
4. Sort
5. Capitalize

You have to create child of child's as given below process tree for each task and each child exec with the image of program of particular task. Print the string after each operation. After creating first child process you will ask user that what task should be performed. After completion of first task create further child and ask user again and so on. Create one file where all processes will save their task detail (assigned task), output and its identity info and after completion of all tasks, show that file content on screen.



Question 5

Scenario:
You want to track your **daily expenses and income** without using any external software. The script should:

1. **Allow users to enter expenses** (category, amount, date).
2. **Automatically categorize transactions** (e.g., Food, Bills, Shopping).
3. **Provide a summary of spending habits**, showing:
    - Total spent per category.
    - Monthly savings vs expenses.
    - Alert when spending crosses a set budget.
4. **Export data to CSV** so you can analyze it in Excel.

Bonus Challenge: Add an option to **visualize spending trends** using ASCII graphs (gnuplot or awk)

---------- Best of Luck 😊 ----------