

SAT 4650 – Applied Computing with Python

Lab 3: File system analysis for chemical process data

Objective

Apply python file system operations and file input/output techniques to analyze and manage chemical engineering-related data files stored across multiple directories.

Background

In chemical engineering workflows, simulation outputs, logs, and experiment notes are often distributed across multiple folders. Engineers frequently need to locate specific files, remove invalid or outdated outputs, and extract key information from text-based reports. In this lab, you will work with a directory structure that mimics a simplified chemical process data archive.

Dataset

You are provided with a compressed file called **ChemicalProcessData.zip**. After extracting the zip file, you will see a folder structure similar to the following:

```
ChemicalProcessData/
|
└── bin/
    ├── reactor_output.bin
    └── outdated_output.bin
|
└── logs/
    └── process_log.txt
|
└── reports/
    └── safety_report.txt
```

Problem 1 – File discovery and cleanup (Guided steps) – **1.75 marks**

In chemical process simulations, outdated binary outputs must be removed before new runs are performed.

Task

Write a Python program that does the following:

1. Automatically access all subfolders inside **ChemicalProcessData**

- a. You may not hard-code file paths
 - b. Use the `os` module to navigate directories
2. Locate the file named `outdated_output.bin`
 3. Once found:
 - a. Print the full path of the file
 - b. Remove/Delete the file using python
 4. After deletion, print a confirmation message indicating that the file was successfully removed.
 5. Locate the file named `reactor_output.bin` and rename it to `reactor_output_20260123.bin` using `os`.

Requirements

- Use the `os` module
- Use a loop to traverse folders
- Your program should still work if folder names change but the target filename stays the same

Problem 2 – Chemical process log analysis (Unguided) – **1.75 marks**

Chemical engineers often examine process logs to identify unsafe conditions or abnormal behavior. The file `process_log.txt` contains time-ordered messages related to a simulated chemical reactor.

Task

Write a python program that:

1. Opens and reads the contents of `process_log.txt`
2. Identifies all lines that indicate unsafe or abnormal conditions
3. Prints those lines clearly to the screen.
4. Writes those lines to a new file named `flagged_events.txt`

Constraints

1. You must decide:
 - a. Whether to use `read()`, `readline()`, or `readlines()`
 - b. How to store and process the lines (list, set, tuple, dictionary)
2. You must use:
 - a. At least one loop
 - b. At least one python collection (list, set, tuple, or dictionary)
3. You may use list comprehensions if helpful
4. Hint: unsafe conditions may be indicated by keywords such as “ALERT”, “WARNING”, “FAILURE”, or “EXCEEDED”. Decide what words you want to use in addition (if you see something else as well). Think like a field-agnostic engineer for this project.

Problem 3 – Use AI to review my knowledge – 0.5 marks

Upload `safety_report.txt` to Google's Gemini and ask it to identify any content that appears unrealistic, incorrect, or impractical from a chemical engineering perspective. After reviewing this feedback, write a 5-7 sentence critical reflection evaluating what this reveals about limitations or gaps in *my* understanding of chemical process safety as reflected in the report. Approach this as if you were reviewing my work as a chemical engineering instructor and explaining what important safety considerations I may be missing or oversimplifying.

Submission Requirements

Please submit the following on Canvas:

1. Your Python source code
2. Screenshots of program output, specifically:
 - a. File removal confirmation
 - b. Printed unsafe log lines
3. `flagged_events.txt` generated by your program
4. Upload `my_views.txt` containing your solution to problem 3