Documentation: Filtering Valid Phone Numbers from a Text File Using Bash

Overview:

The task is to read a text file containing a list of phone numbers and filter out only the valid ones. A valid phone number must conform to one of the following formats:

- (xxx) xxx-xxxx
- xxx-xxx-xxxx

Here, x represents any digit between 0 and 9. The solution is to utilize a Bash script that reads through each line of the file and identifies valid phone numbers using regular expressions.

Input Format:

- The input is a text file named file.txt.
- Each line in the file contains one phone number.
- *The phone numbers can be in various formats, but only two specific formats are considered valid:*
    1. Parentheses format: (xxx) xxx-xxxx
    2. Hyphen format: xxx-xxx-xxxx
- Each line should not have any leading or trailing white spaces.

Output Format:

- The script should print only the valid phone numbers to stdout.
- Invalid phone numbers (those that do not match the required formats) should be ignored.

Valid Phone Number Formats:

1. **(xxx) xxx-xxxx:**

- The area code xxx is enclosed in parentheses.
- A space separates the area code from the next part of the number.
- The number part follows the pattern xxx-xxxx, where each x is a digit.

2. **xxx-xxx-xxxx:**

- The number is split into three parts by hyphens: xxx, xxx, and xxxx.
- Each x is a digit.

Assumptions:

- Every line in the file contains only one phone number.
- There are no extra spaces or characters in the file except for the valid formats mentioned.
- If a line contains an invalid phone number, it should not be printed.

Regular Expression Breakdown:

To identify valid phone numbers, a regular expression (regex) is used. The regex must account for both of the valid formats. Here is a breakdown of how the regex works:

1. **Matching the Parentheses Format ((xxx) xxx-xxxx):**

- The area code is represented as ([0-9]{3}), which matches three digits enclosed in parentheses.
- This is followed by a space: [ ].
- Next comes the number part: [0-9]{3}-[0-9]{4}. This matches three digits, a hyphen, and then four digits.

2. **Matching the Hyphen Format (xxx-xxx-xxxx):**
   - The number is represented as [0-9]{3}-[0-9]{3}-[0-9]{4}, which matches three digits, a hyphen, three more digits, another hyphen, and finally four digits.

3. **Combining Both Formats:**
   - The regex uses the OR operator | to combine the two valid formats.

Steps to Solution:

1. **Read the File:**
   - The script will read file.txt line by line.

2. **Apply Regular Expressions:**
   - For each line, the script applies the regex that matches the valid phone number formats.

3. **Filter and Print Valid Numbers:**
   - If a line matches the regex, the valid phone number is printed to the standard output (stdout).

4. **Ignore Invalid Numbers:**
   - If a phone number does not match the regex, it is ignored, and nothing is printed for that line.

Key Features of the Script:

- **Simplicity:** The script is short and efficient, leveraging grep with extended regular expressions to perform the matching.

- **Performance:** It processes the file line by line, making it memory efficient, especially for large files.

- **Flexibility:** The regex can be modified to accommodate additional valid formats, if necessary.

Edge Cases:

- **Invalid Phone Number:** If a phone number contains spaces, additional characters, or incorrect separators (e.g., dots or slashes), it will not be matched by the regex and will be ignored.

- **No Phone Number:** If the file is empty or contains no valid phone numbers, the script will simply output nothing.

- **Extra Characters:** Phone numbers with extra characters such as letters or symbols mixed with digits will not be considered valid.

Usage:

- The script can be run in a Unix-like environment (e.g., Linux or macOS). Ensure that the text file file.txt is available in the same directory as the script.

By following these guidelines, the script will accurately filter and output valid phone numbers from the text file based on the specified format.