GREP-COMMANDS

`grep` is a powerful Unix command used for searching and matching patterns in text. Here are some common `grep` commands and their explanations:

1. *Basic Pattern Search:*

   - `grep 'pattern' filename`: Search for the specified `pattern` in the `filename`. It will display all lines containing the pattern.

2. *Case Insensitive Search:*

   - `grep -i 'pattern' filename`: Perform a case-insensitive search for the pattern.

3. *Invert Match:*

   - `grep -v 'pattern' filename`: Invert the match, i.e., display lines that do not contain the pattern.

4. *Count Matches:*

   - `grep -c 'pattern' filename`: Count the number of lines that contain the pattern.

5. *Display Line Numbers:*

   - `grep -n 'pattern' filename`: Display line numbers along with matching lines.

6. *Search for Whole Words:*

   - `grep -w 'word' filename`: Search for whole words only, not partial matches.

7. *Recursive Search (in directories):*

   - `grep -r 'pattern' directory`: Recursively search for the pattern in all files within the specified directory and its subdirectories.

8. *Regular Expressions:*

   - `grep -E 'regex' filename`: Use extended regular expressions for pattern matching.

- **`grep -P 'regex' filename`:** Use Perl-compatible regular expressions for pattern matching.

9. *Search for Multiple Patterns:*

- **`grep 'pattern1\|pattern2' filename`:** Search for lines containing either `pattern1` or `pattern2`.

10. *Output Matching Portion:*

- **`grep -o 'pattern' filename`:** Display only the matching portion of the line, rather than the entire line.

11. *Quiet Mode:*

- **`grep -q 'pattern' filename`:** Use this in scripts; it returns a status code (0 if a match is found, 1 if not) without displaying output.

12. *Display Context:*

- **`grep -A num 'pattern' filename`:** Display `num` lines of trailing context after the match.

- **`grep -B num 'pattern' filename`:** Display `num` lines of leading context before the match.

- **`grep -C num 'pattern' filename`:** Display `num` lines of context around the match (both leading and trailing).

13. *Multiple Files:*

- **`grep 'pattern' file1 file2`:** Search for the pattern in multiple files.

14. *Use a File List:*

- **`grep 'pattern' -f file_list.txt`:** Search for the pattern in files listed in `file_list.txt`.

15. *Piping Input:*

- `cat file.txt | grep 'pattern'`: Use `grep` with the output of another command (e.g., `cat`) through a pipe.

Here are some of the most commonly used and useful `grep` commands in Unix for various tasks:

1. Search for a string in a file:

```bash
grep "search_string" file_name
```

2. Search for a string case-insensitively:

```bash
grep -i "search_string" file_name
```

3. Search for a string recursively in a directory and its subdirectories:

```bash
grep -r "search_string" directory_name
```

4. Display line numbers along with matching lines:

```bash
grep -n "search_string" file_name
```

5. Invert the match (display lines that do not contain the search string):

```bash

grep -v "search_string" file_name

```

6. Display only the matching part of the line (using Perl-compatible regular expressions):

~~~bash

grep -oP "pattern" file_name

``

**7. Search for lines that start with a specific pattern:**

```bash

grep "^pattern" file_name

~~~

8. Search for lines that end with a specific pattern:

```bash

grep "pattern$" file_name

```

9. Search for lines that match multiple patterns (AND condition):

```bash

grep "pattern1" file_name | grep "pattern2"

```

10. Count the number of matching lines:

   ```bash

   grep -c "search_string" file_name

   ```

11. Search for lines that match a pattern in a compressed (gzip) file:

   ```bash

   zcat file_name.gz | grep "search_string"

  ```

12. Search for lines that match a pattern, ignoring binary files:

   ```bash

   grep -I "search_string" *

   ```

13. Search for a pattern in a specific file type (e.g., all `.log` files):

   ```bash

   grep "search_string" *.log

   ```

14. Search for lines that do not match a pattern:

   ~~~bash

   grep -v "unwanted_string" file_name

   ``

**15. Search for lines that match a pattern and also show surrounding context:**

   ```bash
   ~~~

grep -C 2 "search_string" file_name

16. Search for lines that match a pattern and highlight the matches:

```bash
grep --color=auto "search_string" file_name
```

17. Search for lines that match a pattern and display the line number and file name:

```bash
grep -Hn "search_string" file_name
```

These `grep` commands cover a wide range of scenarios and are commonly used for text search and analysis in Unix environments. You can adapt them to suit your specific needs and use cases.