comm and sort Commands in Linux/Unix

Both comm and sort are powerful tools for working with text data in Linux and Unix-like systems. Here's a breakdown of their usage and common flags:

**sort**

- **Purpose:** Sorts lines in text files based on specified criteria.
- **Flags:**
  - -n (numeric sort): Sorts data according to numerical values (ascending order).
  - -r (reverse): Sorts in descending order.
  - -k POS,LEN (sort by key): Sorts by a specific field/column.
    - POS: Starting position (1-based) of the field within a line.
    - LEN (optional): The length of the field (defaults to entire line).
  - -t DELIM (field separator): Defines the character separating fields (default is whitespace).
  - -u (unique): Removes duplicate lines.
  - -o OUTPUT_FILE (output file): Saves sorted data to a specified file.

**Example:**

Bash

```
sort -n numbers.txt   # Sort numbers.txt numerically (ascending)
sort -r names.txt     # Sort names.txt in reverse alphabetical order
```

**comm**

- **Purpose:** Compares two sorted text files and identifies lines that are **present in only one file**, **present in both files**, or **not present in either file**.
- **Flags:**
  - -1 (suppress first column): Hides lines only in the first file.
  - -2 (suppress second column): Hides lines only in the second file.
  - -3 (suppress third column): Hides lines common to both files.

**Workflow:**

1. **Sort both files**: Since comm requires sorted inputs, use sort beforehand.
2. **Use comm**: Compare the sorted files to identify differences.

**Example:**

1. Sort files file1.txt and file2.txt:
   Bash
   ```
   sort file1.txt > sorted_file1.txt
   sort file2.txt > sorted_file2.txt
   ```

2. Use comm to compare the sorted files:
   Bash
   ```bash
   comm sorted_file1.txt sorted_file2.txt
   ```

   This will display lines:
   ○ In the first column (unless suppressed by -1): Lines only in file1.txt.
   ○ In the second column (unless suppressed by -2): Lines only in file2.txt.
   ○ In the third column (unless suppressed by -3): Lines common to both files.

**Remember:**

● comm works best with pre-sorted files, so use sort before comm.
● Flags like -1, -2, and -3 allow you to focus on specific differences between the files.

By effectively combining sort and comm, you can analyze and compare text data efficiently.