A Thesis Report

On

# COMPARE AND CONTRAST THE CMP COMMAND WITH THE COMM COMMAND

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENT FOR THE

DEGREE

OF

**BACHELOR IN COMPUTER APPLICATION**

PREPARED BY

JEET MAITY

Roll no.- 33701222034

Paper Name – UNIX AND SHELL PROGRAMMING

Paper Code – BCAC 601

Name of Institute – Global Group of Institutions, Haldia, Dist. – Purba Medinipur

Under the Guidance of

**SUDIPTA PRAMANIK**

Assistant Professor, Global Group of Institutions, Haldia

SUBMITTED ON

March, 2025

## Abstract

This paper presents a comparative analysis of the cmp and comm commands, two essential utilities in Unix/Linux systems for file comparison. While both commands serve the purpose of comparing files, they do so in fundamentally different ways and are suited for different use cases. This study aims to elucidate the functionalities, advantages, and limitations of each command, providing users with a clearer understanding of when to use cmp versus comm.

## Introduction

File comparison is a common task in programming and system administration, often necessary for debugging, version control, and data integrity verification. Unix/Linux systems provide various tools for this purpose, among which cmp and comm are widely used. Understanding the differences between these commands is crucial for users who need to efficiently compare files and interpret the results.

## Objective

The objective of this paper is to compare and contrast the cmp and comm commands in terms of their functionality, output, use cases, and performance. By analyzing these aspects, the paper aims to guide users in selecting the appropriate command for their specific file comparison needs.

## 1. Overview of `cmp`

The cmp command is primarily used to compare two files byte by byte. It reports the first byte and line number where the files differ. If the files are identical, cmp produces no output and returns an exit status of 0. If they differ, it returns an exit status of 1, and if an error occurs, it returns an exit status of 2.

**Usage Example:**

```
cmp file1.txt file2.txt
```

## 2. Overview of `comm`

The comm command compares two sorted files line by line. It produces three columns of output: lines unique to the first file, lines unique to the second file, and lines common to both files. The files must be sorted beforehand for comm to function correctly.

**Usage Example:**

```
comm file1.txt file2.txt
```

## 3. Key Differences

- **Functionality**: cmp compares files byte by byte, while comm compares files line by line.
- **Output**: cmp provides minimal output (the first differing byte), whereas comm provides a detailed comparison in three columns.
- **Sorting Requirement**: comm requires sorted input files, while cmp does not.

- **Use Cases**: cmp is suitable for binary files and quick checks, while comm is ideal for text files where line-by-line comparison is needed.

## Review of Literature

Previous studies and documentation on Unix/Linux commands highlight the importance of file comparison tools in software development and system administration. The GNU Coreutils documentation provides detailed descriptions of both commands, emphasizing their respective use cases and functionalities. Various online forums and tutorials also discuss practical applications and examples, aiding users in understanding when to use each command effectively.

## Methodology

This comparative analysis was conducted through a review of official documentation, user manuals, and practical experimentation with both commands. Various file types (text and binary) were used to evaluate the performance and output of cmp and comm. The results were analyzed to identify strengths and weaknesses in different scenarios.

## Conclusion

In conclusion, both cmp and comm serve essential roles in file comparison within Unix/Linux systems. cmp is best suited for quick, byte-level comparisons, particularly for binary files, while comm excels in line-by-line comparisons of sorted text files. Understanding the differences between these commands allows users to choose the most appropriate tool for their specific needs, enhancing efficiency and accuracy in file comparison tasks.

## References

1. https://www.quora.com/What-are-the-differences-between-the-diff-and-cmp-commands-on-Linux-Unix
2. https://tldp.org/LDP/abs/html/
3. https://www.gnu.org/software/coreutils/