|  |  |
| --- | --- |
| **EXP NO** | 01 |
| **DATE** | 10/01/2025 |

**BASIC OF UNIX COMMANDS**

**This experiment maps to the following COs and POs.**

|  |  |
| --- | --- |
| **CO1** | Understand and implement basic services and functionalities of the operating system using system calls. |
| **PO1** | Engineering knowledge |
| **PO5** | Modern tool usage |
| **PO9** | Individual and team work |

**AIM:**

To study and implement the **basic unix** commands and the variantions.

**COMMANDS:**

1. **mkdir**

|  |  |
| --- | --- |
| **Description** | Make directories; create directories if they don’t exist. |
| **Syntax** | mkdir [OPTION] DIRECTORY |
| **Options** | **-m:** set file mode (as in chmod)  **-p:** make parent directories as needed  **-v:** print a message for each created directory |

**Commands:**

mkdir os

mkdir –v os1

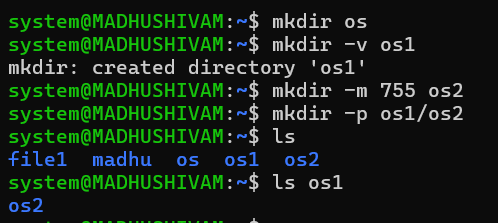
mkdir –m 755 os2

mkdir –p os1|os2

ls

ls os1

**OUTPUT**



1. **cd**

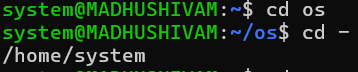
|  |  |
| --- | --- |
| **Description** | Change the shell working directory. |
| **Syntax** | cd [OPTIONS] DIRECTORY |
| **Options** | **~:** change to $HOME  **-:** change to the previous directory |

**Commands:**

cd os

cd-

**OUTPUT**



1. **touch**

|  |  |
| --- | --- |
| **Description** | Update the access and modification times of each file to the current time. A file argument that doesn’t exist is created empty. |
| **Syntax** | touch [OPTION] FILE |
| **Options** | **-a:** change only the access time  **-c:** do not create any files  **-m:** change only the modification time |

**Commands**

touch process.txt

ls

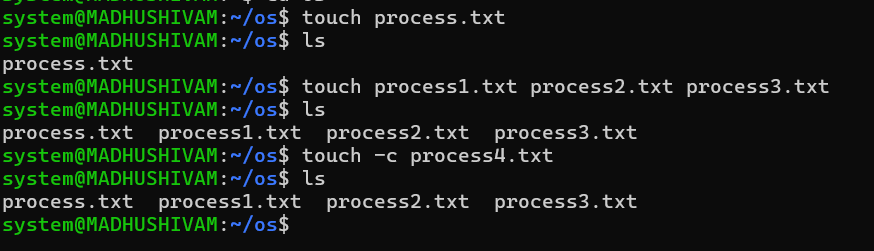
touch process1.txt process2.txt process3.txt

ls

touch –c process4.txt

ls

**OUTPUT**



1. **cat**

|  |  |
| --- | --- |
| **Description** | The cat command is used to concatenate and display the contents of files. It can also be used to create files, view file content, and combine multiple files into one. |
| **Syntax** | cat [OPTIONS] [FILE]... |
| **Options** | -n: Number all output lines.  -b: Number non-empty output lines only. |

**Commands:**

cat > process1.txt

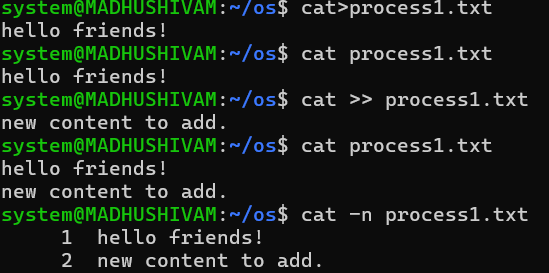
cat process1.txt

cat >>process.txt

cat process1.txt

cat –n process1.txt

**OUTPUT**



1. **ls**

|  |  |
| --- | --- |
| **Description** | List files and directories in the current directory. |
| **Syntax** | ls [OPTION] [DIRECTORY] |
| **Options** | **-a:** do not ignore entries starting with **.**; show all files and directories, including hidden files  **-l:** use long listing format |

**Commands**

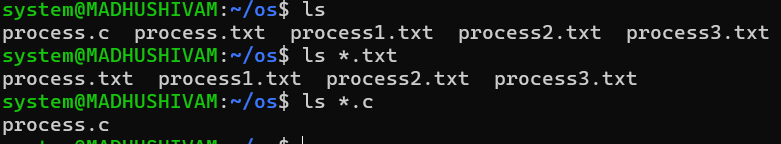
ls

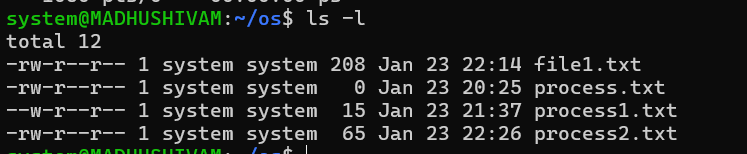
ls \*.txt

ls \*.c

ls –l

**OUTPUT**





1. **pwd**

|  |  |
| --- | --- |
| **Description** | Print the full filename of the current working directory. |
| **Syntax** | pwd [OPTION] |
| **Options** | **-L:** use PWD from environment, even if it contains symlinks  **-P:** avoid all symlinks |

**Commands**

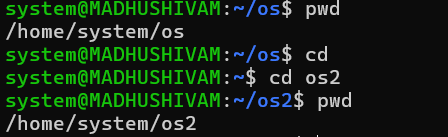
Pwd

Cd

Os2

Pwd

**OUTPUT:**



1. **cp**

|  |  |
| --- | --- |
| **Description** | Copy source to destination, or to directory; copy files and directories. |
| **Syntax** | cp [OPTION] SOURCE DEST  cp [OPTION] SOURCE DIRECTORY cp [OPTION] -t DIRECTORY SOURCE |
| **Options** | **-i:** prompt before overwrite  **-l:** hard link files instead of copying  **-s:** make symbolic links instead of copying  **-t:** copy all source arguments into directory  **-T:** treat DEST as a normal file |

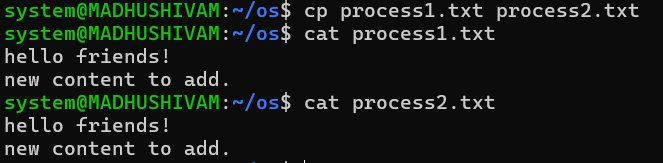
**Commands**

cp process1.txt process2.txt

cat process1.txt

cat process2.txt

**OUTPUT**



1. **mv**

|  |  |
| --- | --- |
| **Description** | Move/rename files; rename source to destination, or move source(s) to directory |
| **Syntax** | mv [OPTION] SOURCE DEST  mv [OPTION] SOURCE DIRECTORY mv [OPTION] -t DIRECTORY SOURCE |
| **Options** | **-f:** don’t prompt before overwriting  **-i:** prompt before overwrite  **-t:** move all source arguments into directory  **-T:** treat DEST as a normal file |

**Commands**

cd

ls

mv process1.txt/home/system/os2

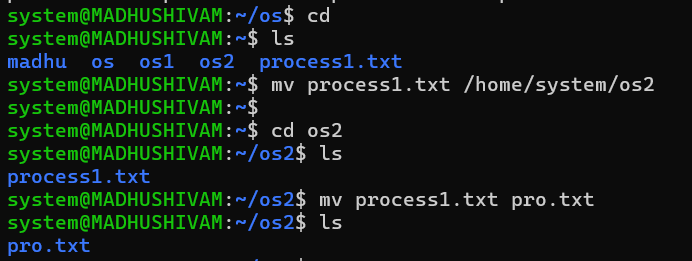
cd os2

ls

mv process1.txt pro.txt

ls

**OUTPUT**



1. **head**

|  |  |
| --- | --- |
| **Description** | Print the first 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. |
| **Syntax** | head [OPTION] FILE |
| **Options** | **-n:** print the given no. of lines instead of first 10.  **-v:** always print headers giving file names |

**Commands**

vi file.txt

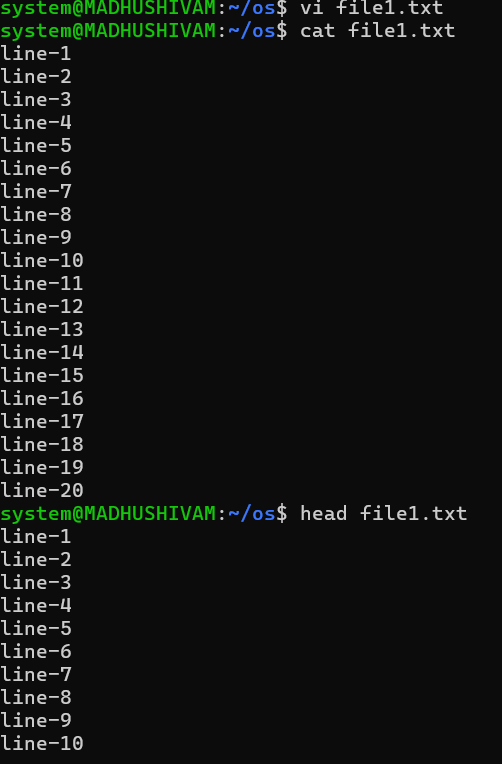
cat file.txt

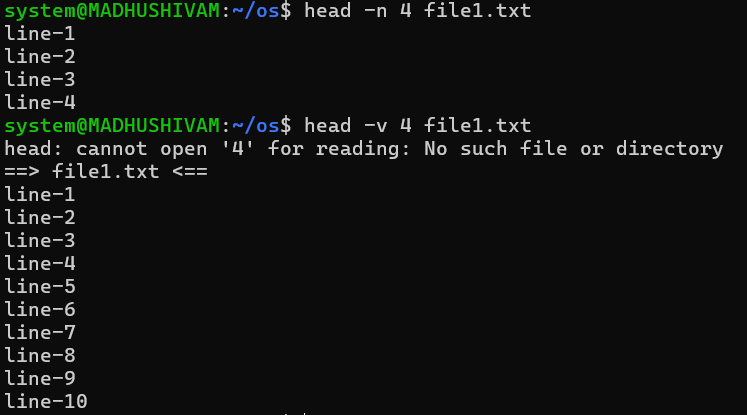
head file1.txt

head –n 4 file1.txt

head –v 4 file1.txt

**OUTPUT**





1. **tail**

|  |  |
| --- | --- |
| **Description** | Print the last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. |
| **Syntax** | tail [OPTION] FILE |
| **Options** | **-n:** print the given no. of lines instead of first 10.  **-v:** always print headers giving file names |

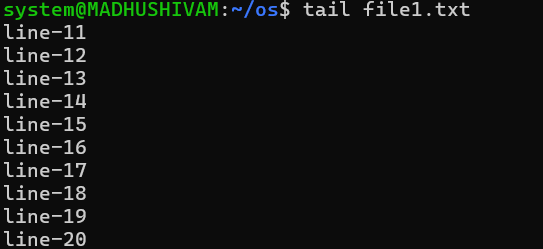
**Commands**

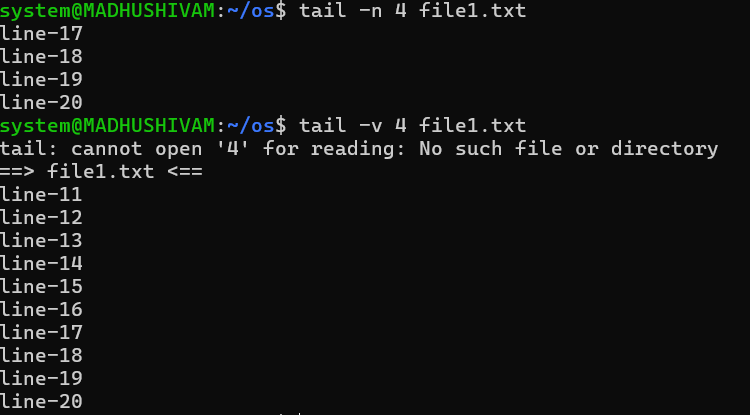
tail file1.txt

tail –n 4 file1.txt

tail –v 4 file1.txt

**OUTPUT**





1. **tac**

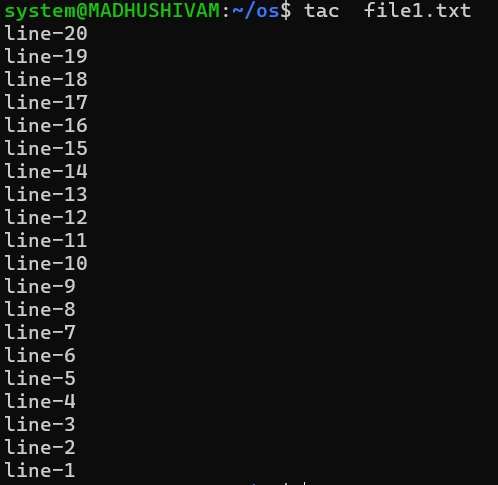
|  |  |
| --- | --- |
| **Description** | The tac command is used to display the contents of a file in reverse order, line by line. It is essentially the reverse of the cat command. |
| **Syntax** | tac [OPTIONS] [FILE]... |
| **Options** | **-b**: Attach the separator before each line. |

**Commands**

cat file1.txt

tac file1.txt

**OUTPUT**



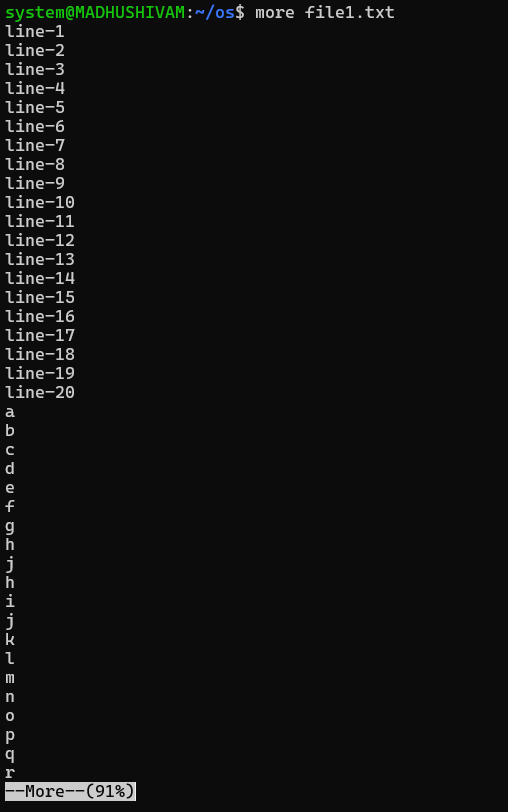
1. **more**

|  |  |
| --- | --- |
| **Description** | A filter for paging through text on screenful at a time. This version is especially primitive. |
| **Syntax** | more [OPTION] FILE |
| **Options** | **-s:** squeeze multiple blank lines into one. |

**Commands**

more file.txt

**OUTPUT**



1. **vi**

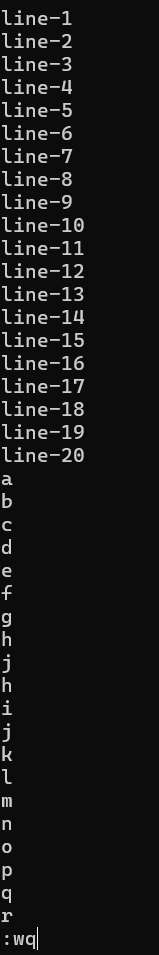
|  |  |
| --- | --- |
| **Description** | A text editor used to edit all kinds of plain text. It is especially useful for editing programs. |
| **Syntax** | vim [OPTION] FILE vi [OPTION] FILE |

**Commands**

Vi file.txt

**OUTPUT**





1. **id**

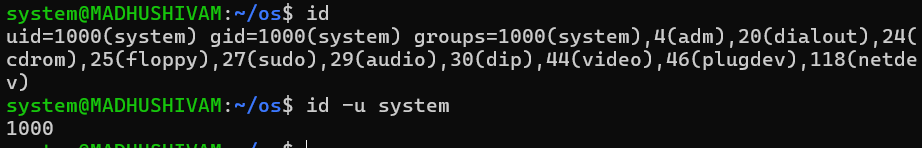
|  |  |
| --- | --- |
| **Description** | The id command shows information about the current user, such as their user ID (UID) and group ID (GID). |
| **Syntax** | id [OPTION] [USERNAME] |
| **Options** | **-u: Show user ID (UID).**  **-g: Show group ID (GID).-** |

**Commands**

id

id -u system

**OUTPUT**



1. **grep**

|  |  |
| --- | --- |
| **Description** | Searches for patterns in each file. |
| **Syntax** | grep [OPTIONS] PATTERN FILE |
| **Options** | **-E:** interpret patterns as extended regular expressions  **-i:** ignore case distinctions in patterns and input data, so that characters that differ only in case match each other  **-v:** invert the sense of matching, to select non-matching lines  **-c:** print a count of matching lines  **-o:** print only the matched (non-empty) parts of a matching line, with each such part on a separate output line |

**Commands**

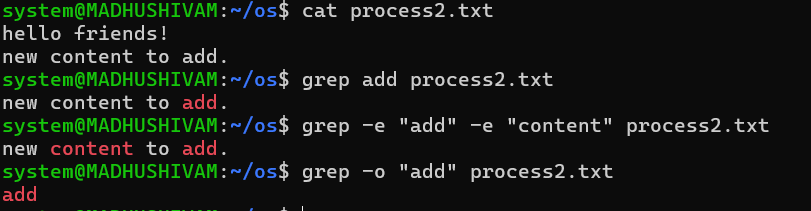
cat process2.txt

grep add process2.txt

grep -e “add” -e “content” process2.txt

grep -o “add” process2.txt

**OUTPUT**



1. **diff**

|  |  |
| --- | --- |
| **Description** | Compare files line by line. |
| **Syntax** | diff [OPTION] FILE1 FILE2 |
| **Options** | **-q:** report only when files differ |

**Commands**

cat process1.txt

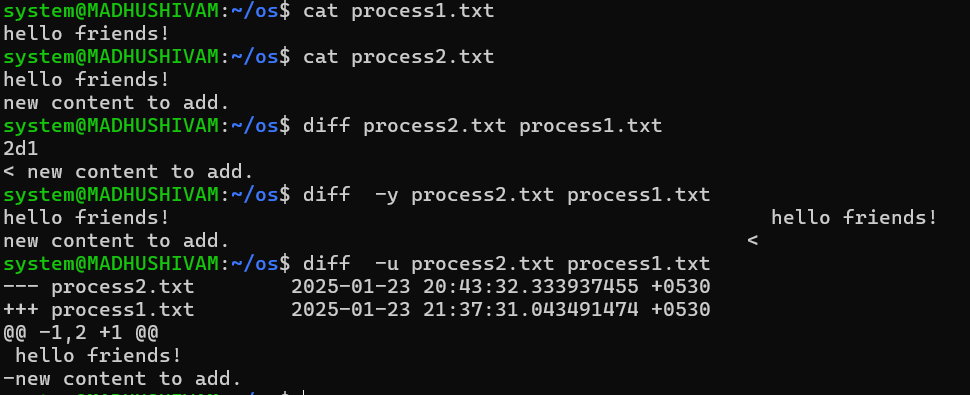
cat process2.txt

diff process2.txt process1.txt

diff -y process2.txt process1.txt

diff -u process2.txt process1.txt

**OUTPUT**



1. **ping**

|  |  |
| --- | --- |
| **Description** | The ping command is used to test the connection to another computer or network device (like a server) by sending "echo request" packets and waiting for a response. |
| **Syntax** | ping [OPTION] DESTINATION |
| **Options** | **-c COUNT**: Send a specific number of ping requests. |

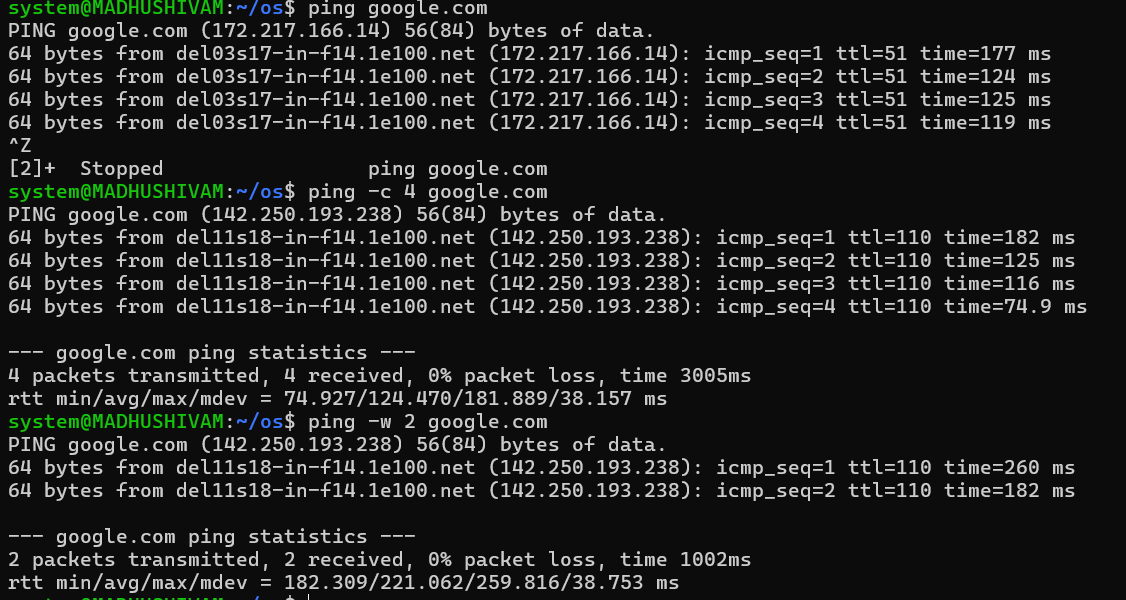
**Commands**

ping google.com

ping -c 4 google.com

ping -w 2 googlr.com

**OUTPUT**



1. **history**

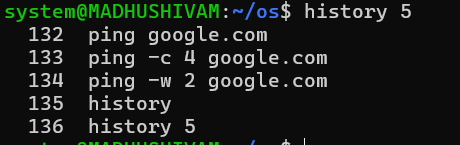
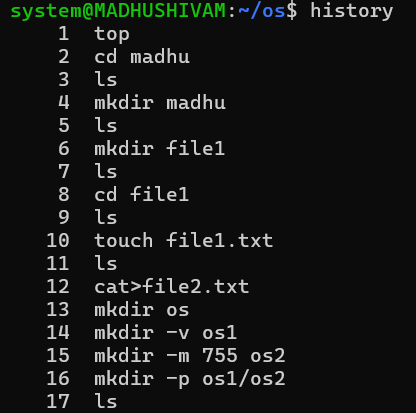
|  |  |
| --- | --- |
| **Description** | Keeps track of all inputs/commands read from the user. |
| **Syntax** | history |

**Commands**

history

history 5

**OUTPUT**



1. **hostname**

|  |  |
| --- | --- |
| **Description** | Used to display the system’s DNS name, and to display or set its hostname or NIS domain name |
| **Syntax** | hostname [OPTIONS] |
| **Options** | **-d:** display the name of the DNS domain  **-f:** display the Fully Qualified Domain Name  **-i:** show the IP address of the system |

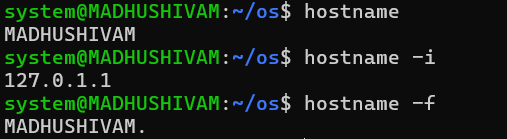
**Commands**

hostname

hostname -i

hostname -f

**OUTPUT**



1. **chmod**

|  |  |
| --- | --- |
| **Description** | Change the access permissions of the named files; can’t change the permissions of symbolic links. |
| **Syntax** | chmod [OPTIONS] MODE FILENAME |
| **Options** | **-v:** output a diagnostic for every file processed  **-f:** suppress most error messages |

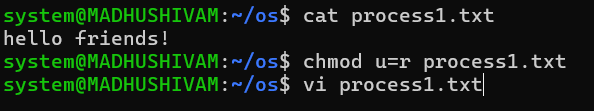
**Commands**

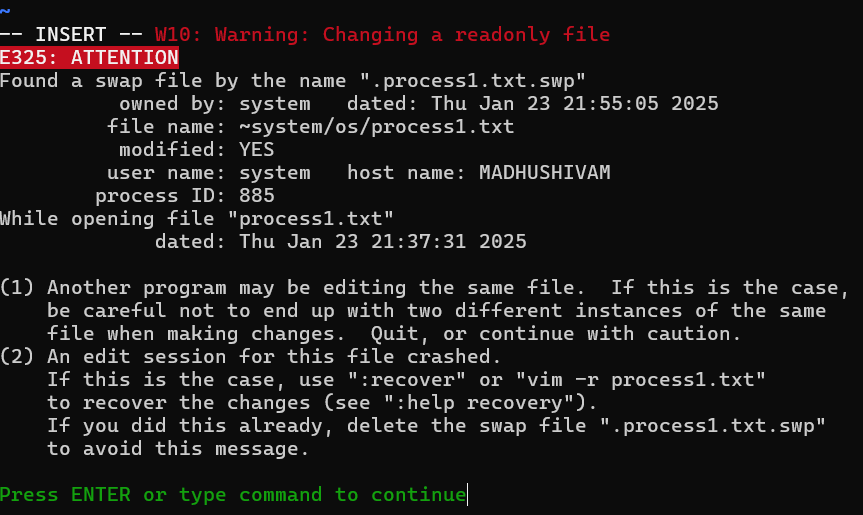
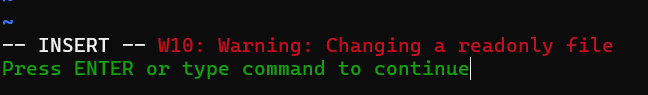
cat process1.txt

chmod u=r process1.txt

vi process1.txt

**OUTPUT**







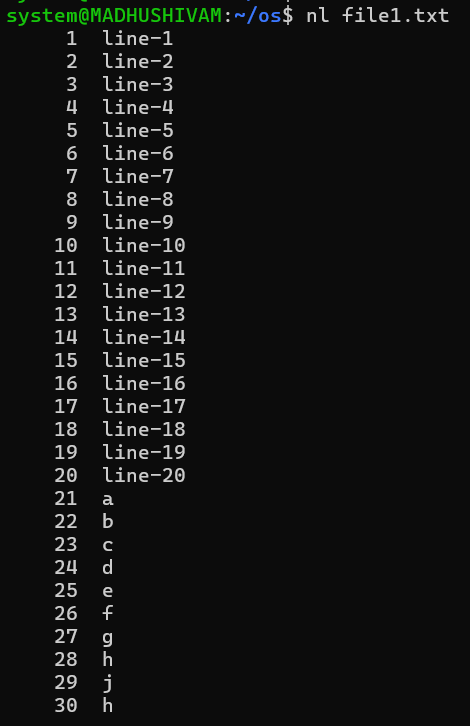
1. **nl**

|  |  |
| --- | --- |
| **Description** | The nl command is used to number the lines of a file or standard input. It outputs the file with line numbers added, which can be helpful for viewing or editing text. |
| **Syntax** | nl [OPTION] FILE |
| **Options** | -**s SEP**: Specify a separator (default is a tab) between the line number and the content. |

**Commands**

nl file1.txt

**OUTPUT**



1. **wc**

|  |  |
| --- | --- |
| **Description** | Print newline, word, and byte counts for each FILE, and a total line if more than one file is specified. |
| **Syntax** | wc [OPTIONS] FILE |
| **Options** | **-c:** print the byte count  **-m:** print the character count  **-l:** print the newline count  **-w:** print the word count |

**Commands**

cat process2.txt

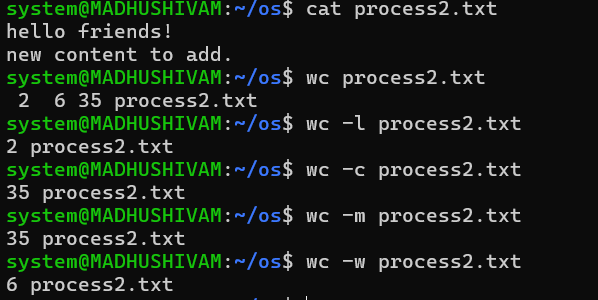
wc process2.txt

wc -l process2.txt

wc -m process2.txt

wc -w process2.txt

**OUTPUT**



1. **uniq**

|  |  |
| --- | --- |
| **Description** | The uniq command is used to filter out or report repeated lines in a file or input. It’s typically used after sorting data, as uniq only identifies consecutive duplicate lines. |
| **Syntax** | uniq [OPTION] [INPUT] [OUTPUT] |
| **Options** | -c: Prefix each line with the number of occurrences.  -d: Show only duplicate lines (lines that appear more than once).  -u: Show only unique lines (lines that appear only once).  -i: Ignore case when comparing lines. |

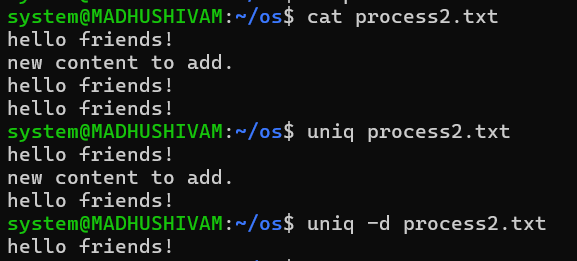
**Commands**

cat process2.txt

uniq process2.txt

uniq -d process2.txt

**OUTPUT**



1. **rmdir**

|  |  |
| --- | --- |
| **Description** | Remove empty directories. |
| **Syntax** | rmdir [OPTION] DIRECTORY |
| **Options** | --**ignore-fail-on-non-empty:** ignore each failure that’s solely because a directory is non-empty  **-p:** remove DIRECTORY and its ancestors; eg:  *rmdir -p a/b/c* is the same as *rmdir a/b/c a/b a*  **-v:** output a diagnostic for every directory processed |

**Commands**

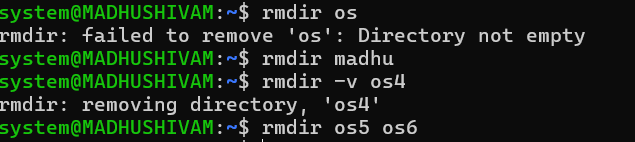
rmdir os

rmdir madhu

rmdir -v os4

rmdir os5 os6

**OUTPUT**



1. **rm**

|  |  |
| --- | --- |
| **Description** | The rm (remove) command is used to delete files or directories in Unix/Linux systems. |
| **Syntax** | rm [OPTION] FILE |
| **Options** | -f: Force remove files without prompting for confirmation, even if the file is write-protected.  -r: Remove directories and their contents recursively.  -i: Prompt for confirmation before deleting each file. |

**Commands**

rm os2

rm process3.txt

rm -i process.txt

ls

**OUTPUT**

