**Lab Practical #01:**

Study of basic LINUX commands.

**Practical Assignment #01:**

1. Perform and explain various Linux commands listed below:
   1. **WHOAMI**
   2. **LS**
   3. **MAN**
   4. **CLEAR**
   5. **PWD**
   6. **CD**
   7. **MKDIR**
   8. **TOUCH**
   9. **RMDIR**
   10. **RM**
   11. **MV**
   12. **CP**
   13. **HEAD**
   14. **TAIL**
   15. **UNIQ**
   16. **FIND**
   17. **GREP**
   18. **NANO**
   19. **SU**

## WHOAMI

### Description: Thewhoamicommand is used to display the username associated with the current effective user ID. It is often used in shell scripts or command-line environments to determine the current user's identity.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | --help | It gives the help message. |
| 2 | --version | It gives the version information. |

### Implementation:



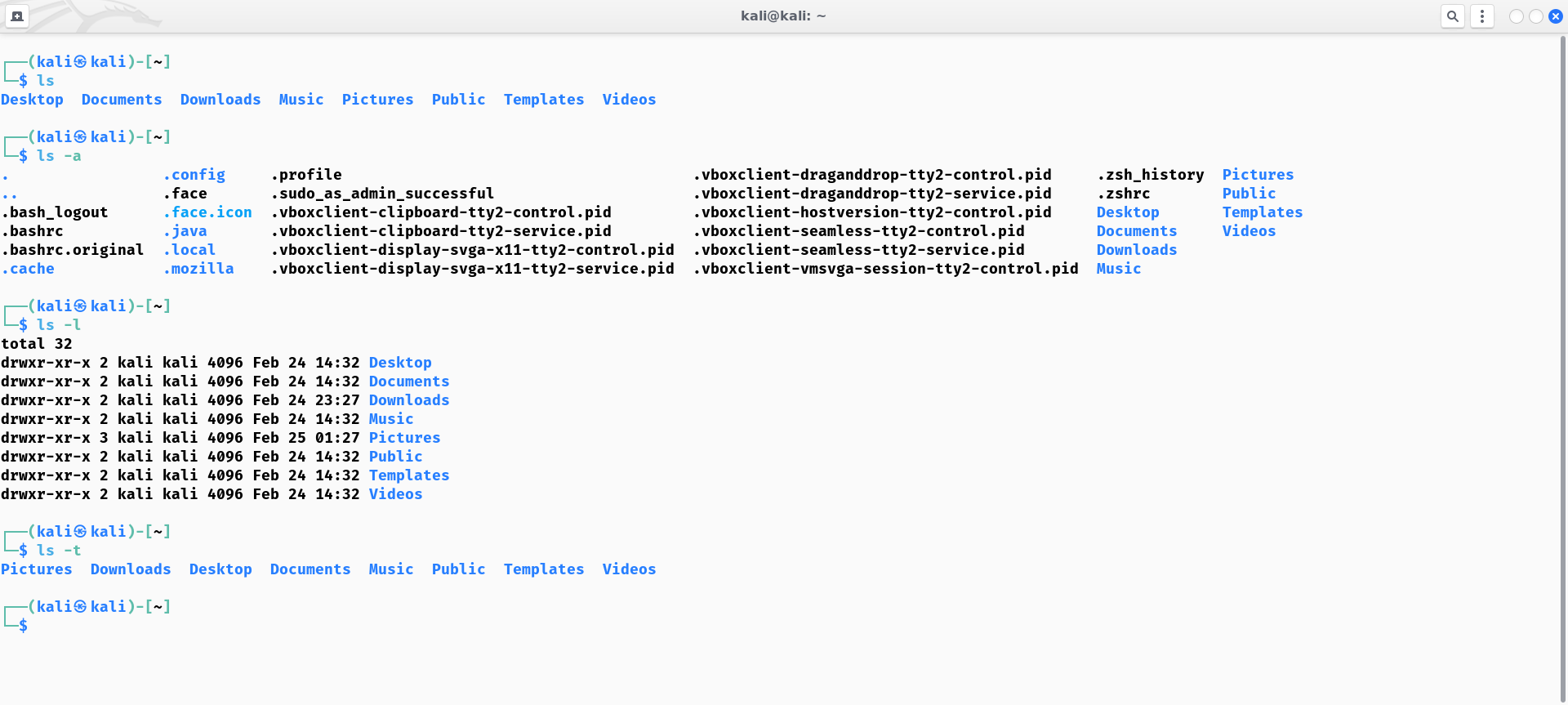
## LS

### Description:

The **ls** command is used to list directory contents. It has various flags (options) that modify its behavior.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -l | Displays detailed information about files and directories, including permissions, number of links, owner, group, file size, and modification date. |
| 2 | -a | Includes hidden files and directories in the listing. Hidden files are those whose names start with a dot (.) |
| 3 | -t | Sorts the list of files and directories by modification time, with the newest files first. |

### Implementation:



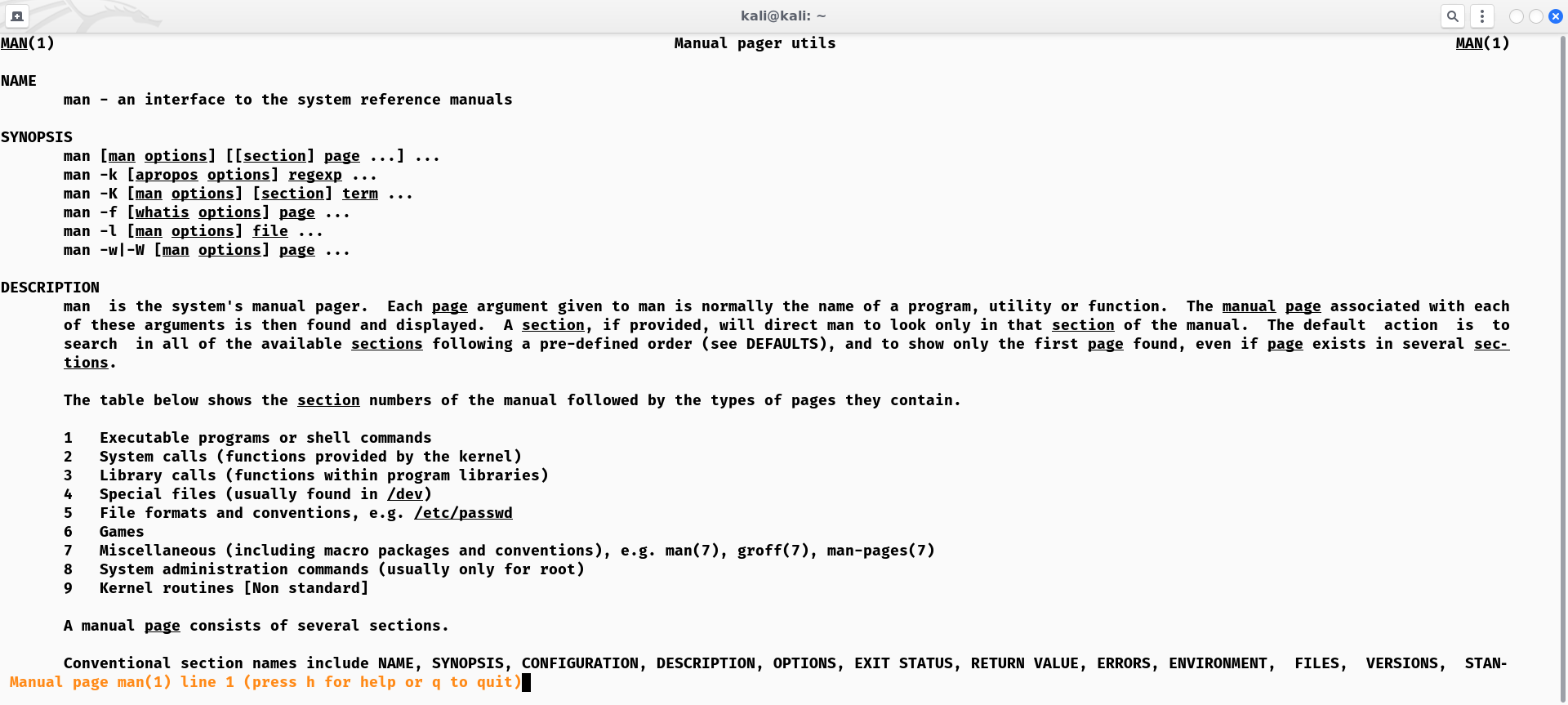
## MAN

### Description:

The **man** command used to display the manual pages for various commands, utilities, and system calls. Manual pages provide detailed information about the usage, options, and functionality of these commands.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -k | Searches the manual page names and descriptions for the specified keywords and displays a list of matches. |
| 2 | -f | Displays concise descriptions of the specified commands or topics. |
| 3 | -c | Uses the specified configuration file instead of the default. |

### Implementation:

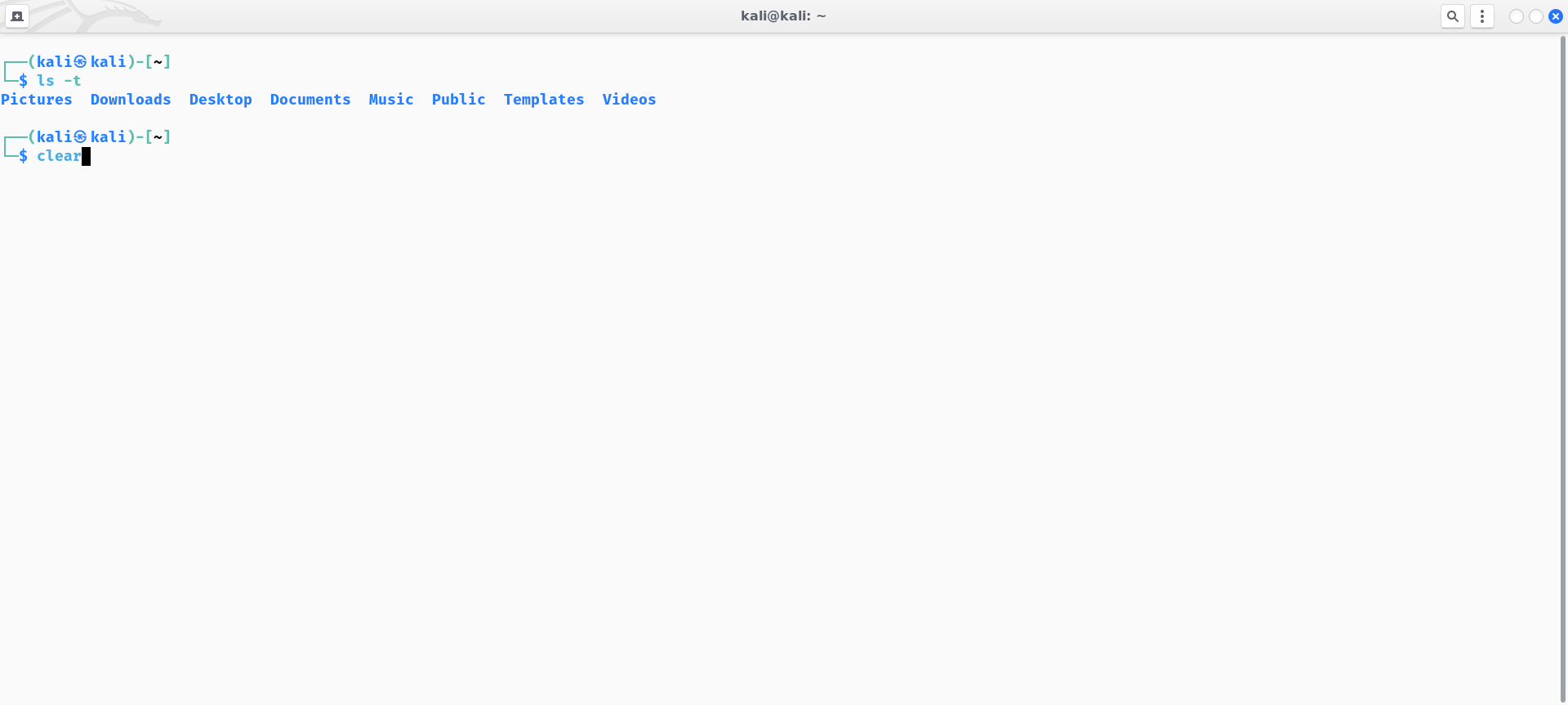


## CLEAR

### Description:

The **clear** command is used to clear the terminal screen by removing all previous output and commands. It doesn't have any flags or options. When you execute **clear**, it simply clears the terminal window, providing you with a clean slate to work with.

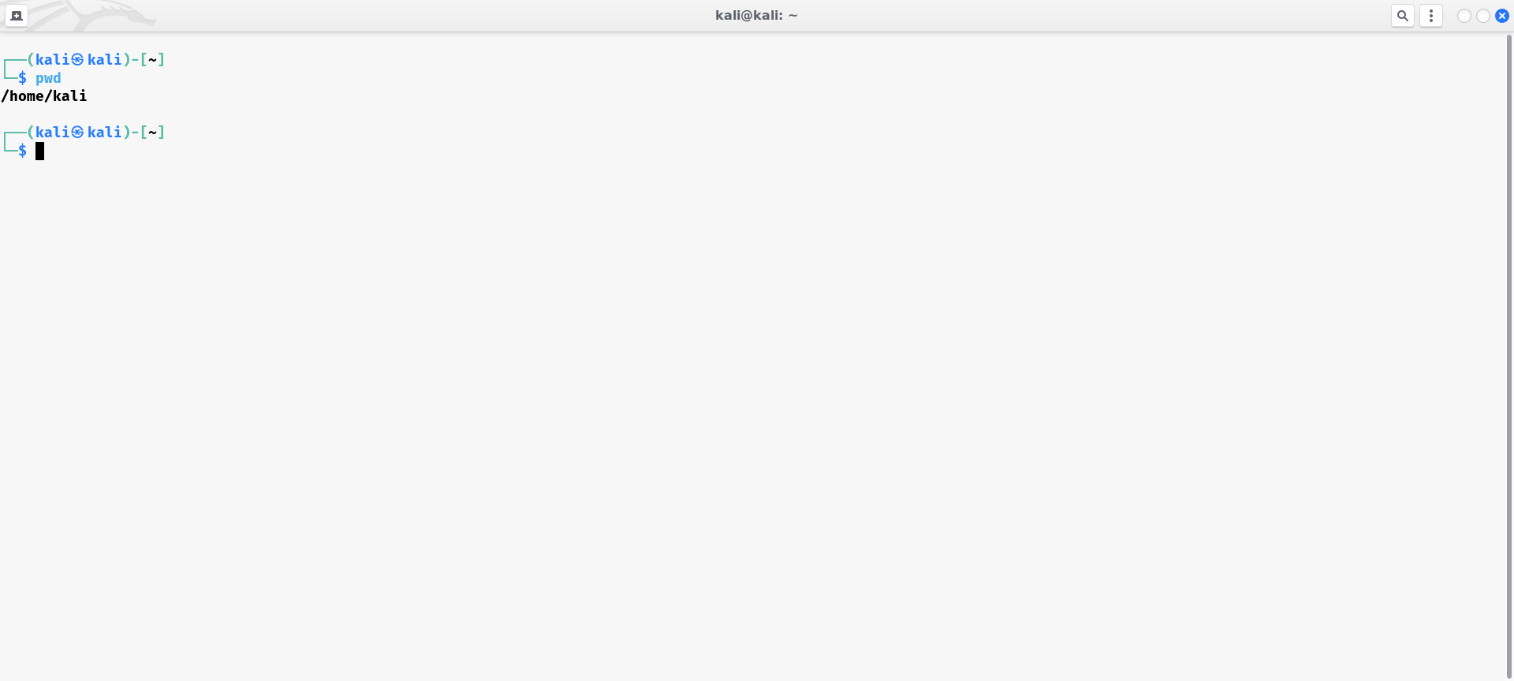
### Implementation:



## PWD

### Description:

The **pwd** command stands for "print working directory" and is used to display the current working directory. It doesn't typically have any flags or options. When you execute **pwd**, it simply prints the absolute path of the current directory you are working in.Implementation:

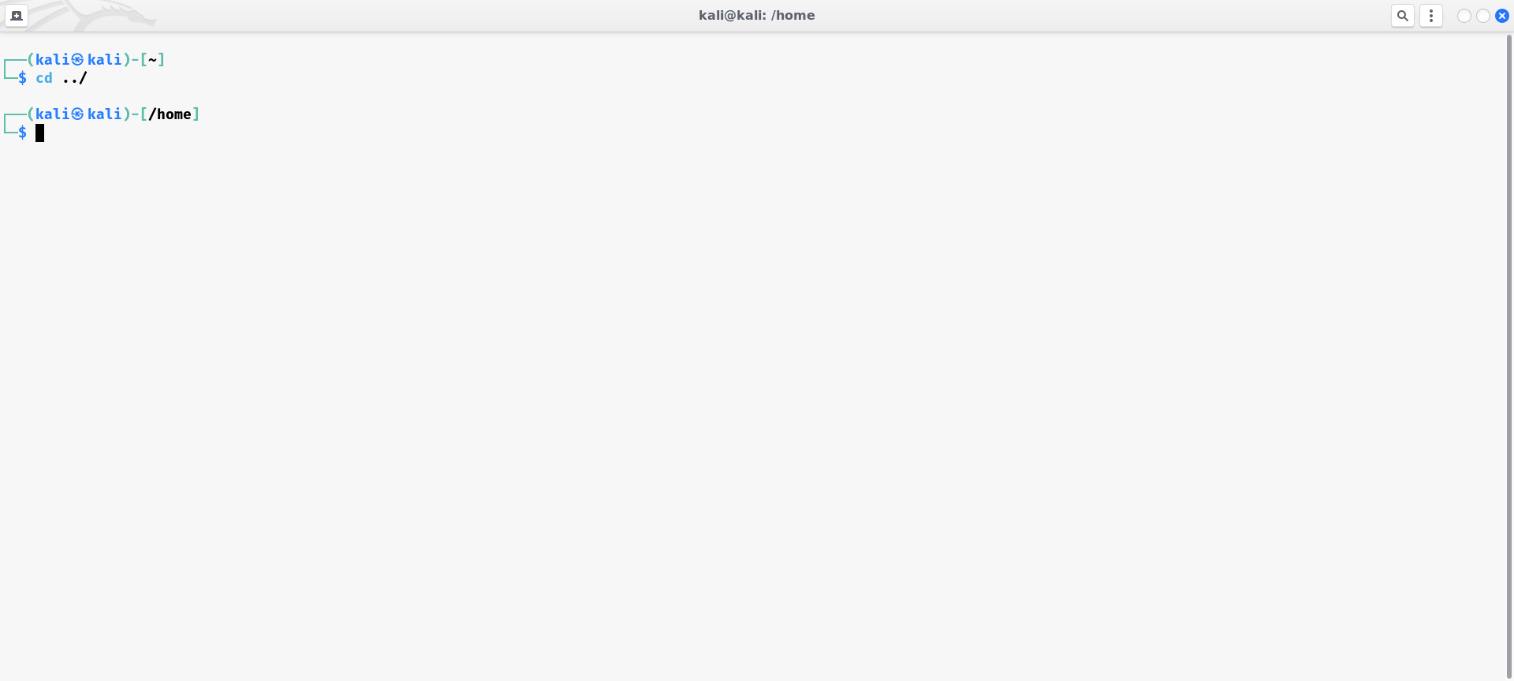


## CD

### Description:

The **cd** command, short for "change directory," is used to change the current working directory. It doesn't typically have any flags or options. When you execute **cd**, you provide it with a directory path as an argument, and it changes the current directory to the specified one.

### Implementation:



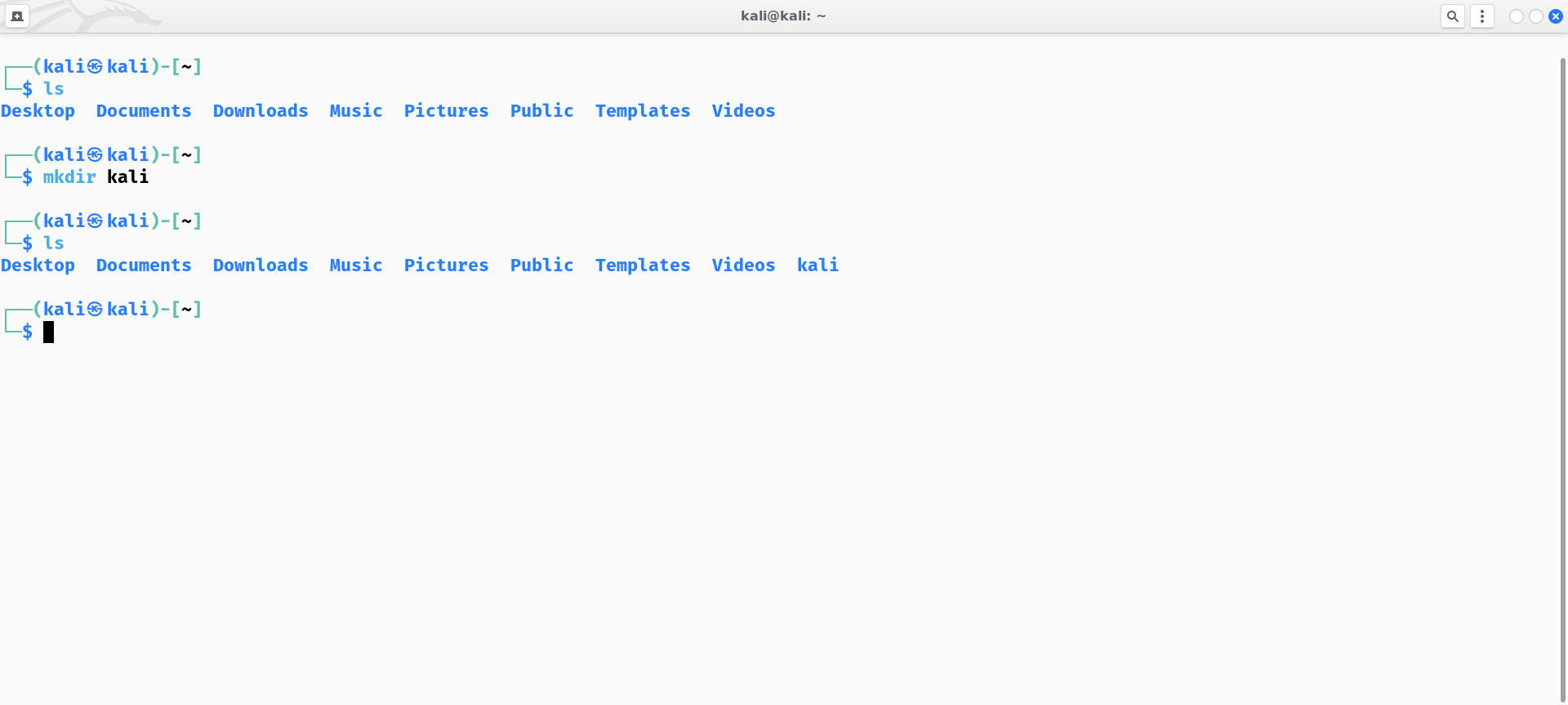
## MKDIR

### Description:

The **mkdir** command is used to create directories (folders). It has several flags or options that can modify its behavior.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -p | This flag allows you to create not only the specified directory but also any necessary parent directories. If the parent directories already exist, **mkdir** will not raise an error. |
| 2 | -m | This flag allows you to specify the permissions mode for the newly created directory using octal notation. For example, **-m 755** sets read, write, and execute permissions for the owner, and read and execute permissions for group and others. |
| 3 | -v | This flag displays the version of the **mkdir** command. |

### Implementation:



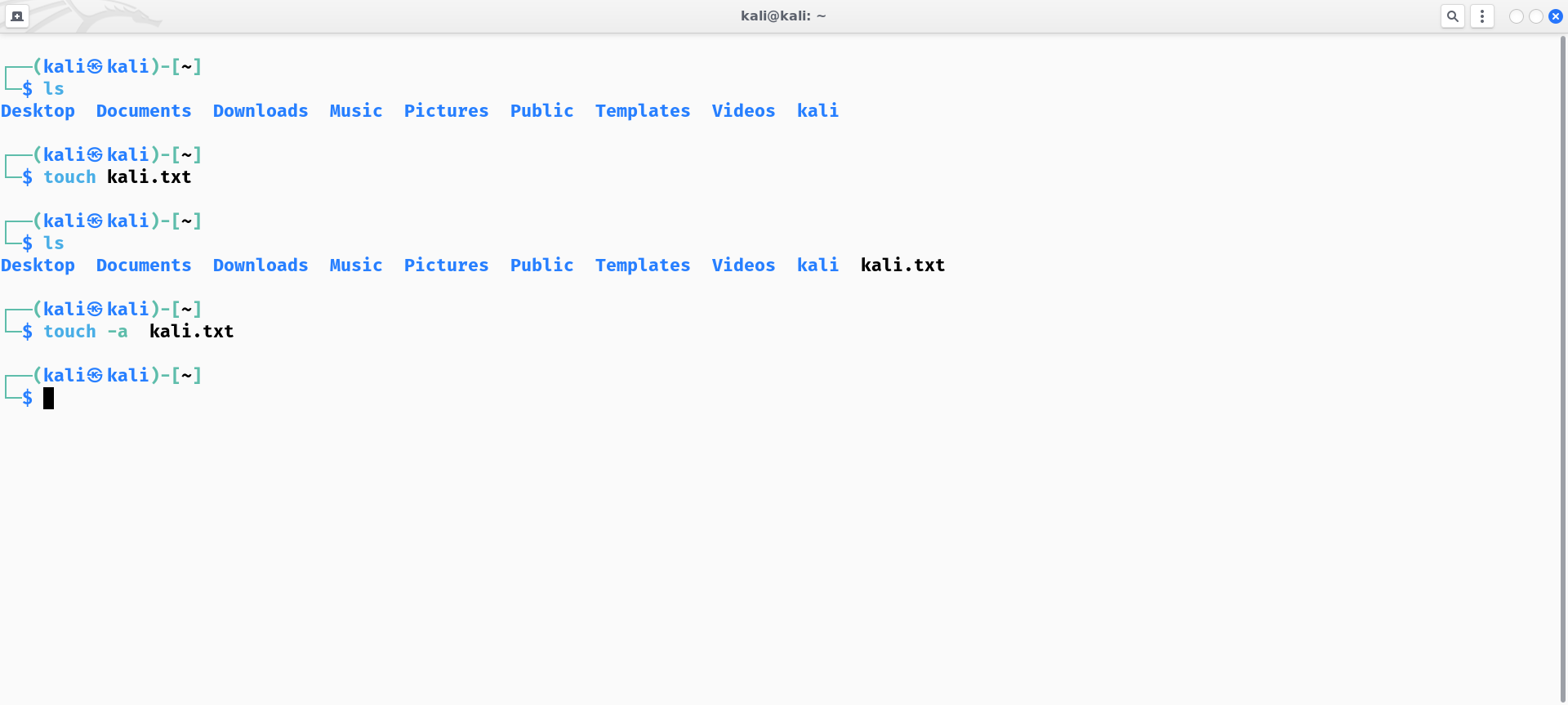
## TOUCH

### Description:

The **touch** command is primarily used to update the access and modification timestamps of a file. If the file specified doesn't exist, **touch** creates an empty file.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -a | This flag updates only the access time of the specified file(s) without modifying their content or modification time. |
| 2 | -c | This flag prevents **touch** from creating a new file if the specified file doesn't exist. It won't report an error if the file is not found. |
| 3 | -m | This flag updates only the modification time of the specified file(s) without changing their content or access time. |

### Implementation:



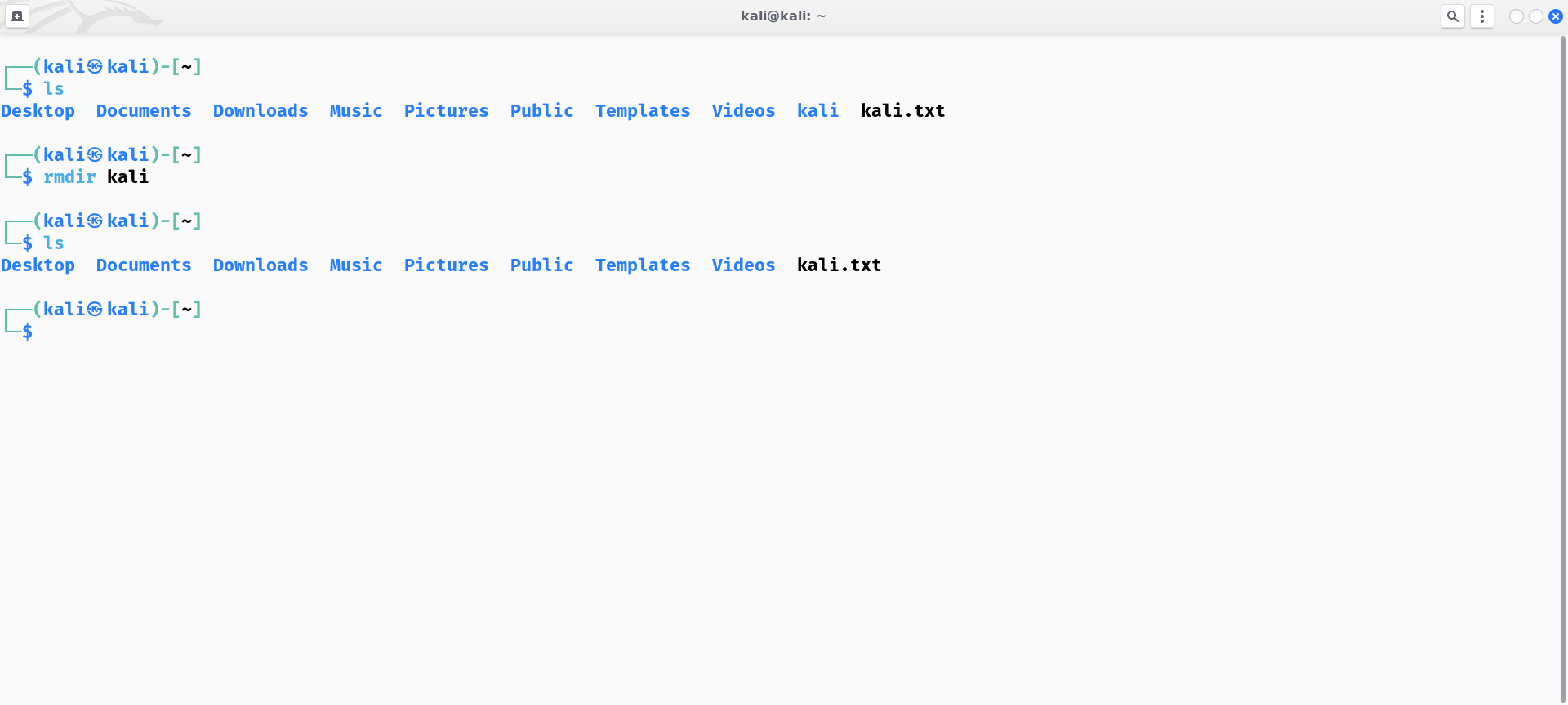
## RMDIR

### Description:

The **rmdir** command is used to remove empty directories (folders).

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -p | This flag allows you to remove not only the specified directory but also any necessary parent directories if they become empty after the removal. |
| 2 | --ignore-fail-on-non-empty | By default, **rmdir** fails if the specified directory is not empty. This flag makes it ignore the failure and exit with a success status even if the directory is not empty. |

### Implementation:



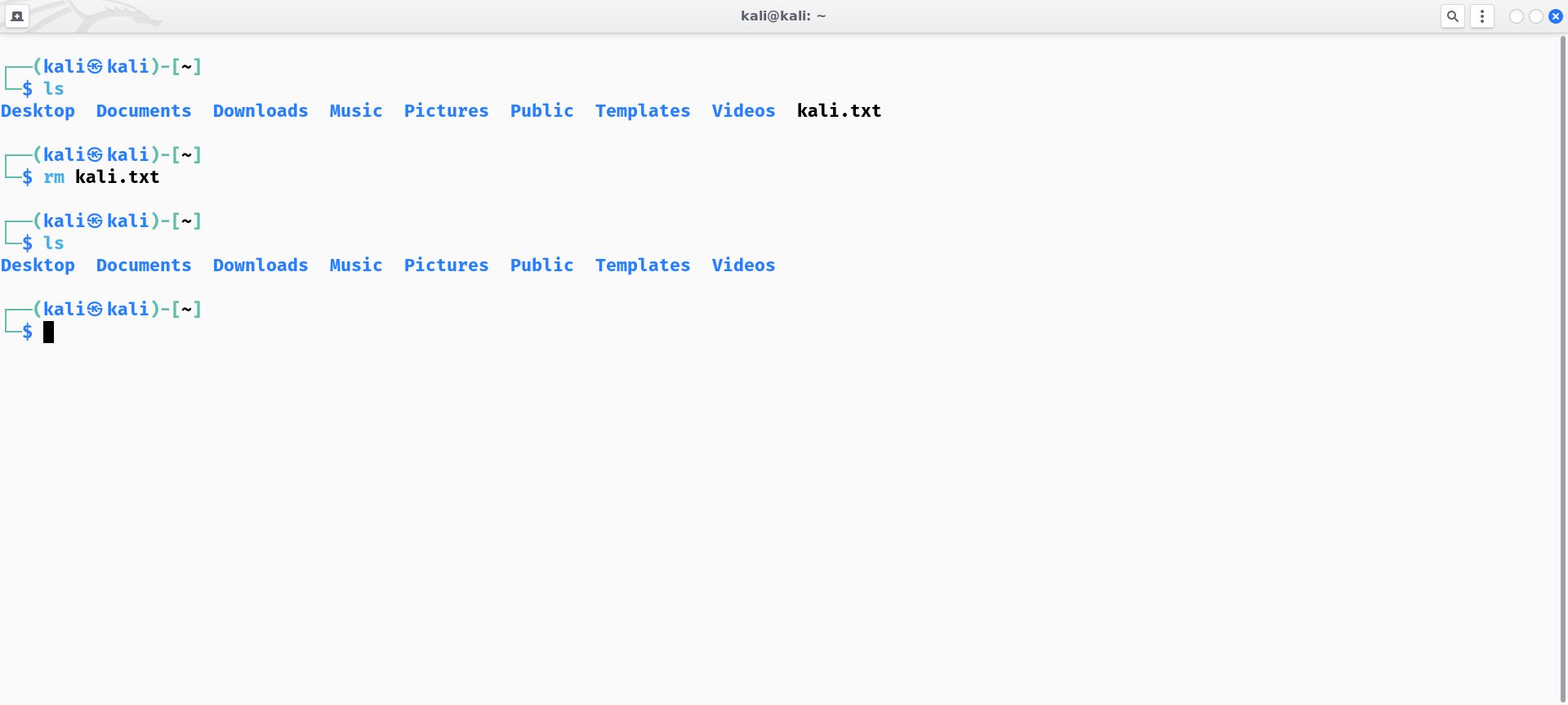
## RM

### Description:

The **rm** command is used to remove files or directories. It has several flags or options that modify its behavior.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -f | This flag suppresses most error messages and prompts, allowing **rm** to continue removing files without asking for confirmation, even if files are write-protected. |
| 2 | -i | This flag prompts the user for confirmation before removing each file or directory, allowing you to verify before proceeding. |
| 3 | -r | This flag enables **rm** to remove directories and their contents recursively. It's used when deleting directories and their contents instead of individual files. |

### Implementation:



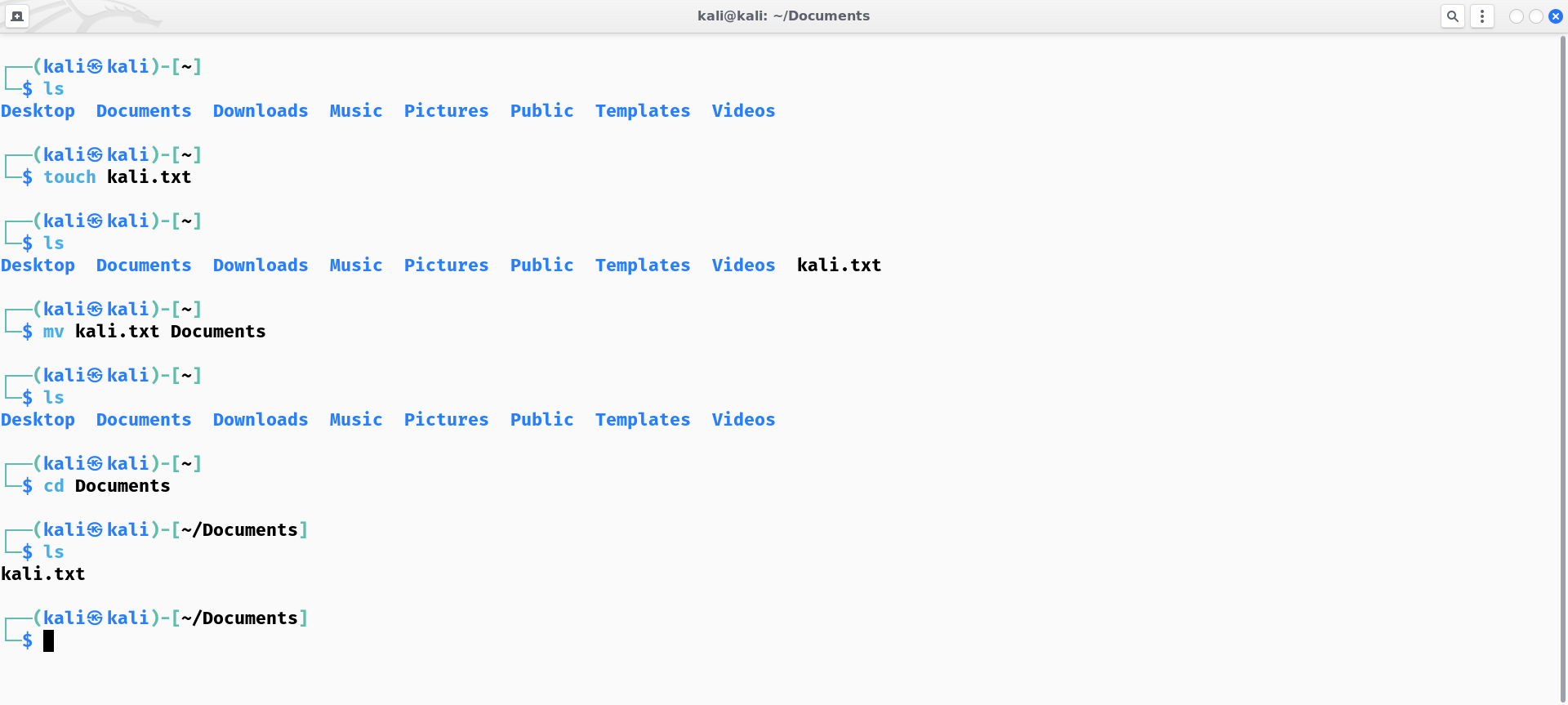
## MV

### Description:

The **mv** command is used to move or rename files and directories.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -i | This flag prompts the user for confirmation before overwriting existing destination files. |
| 2 | -f | This flag suppresses warnings and prompts, allowing **mv** to overwrite existing destination files without asking for confirmation. |
| 3 | -n | This flag prevents **mv** from overwriting existing destination files. If a file with the same name already exists, **mv** will not overwrite it and will instead display an error message. |

### Implementation:



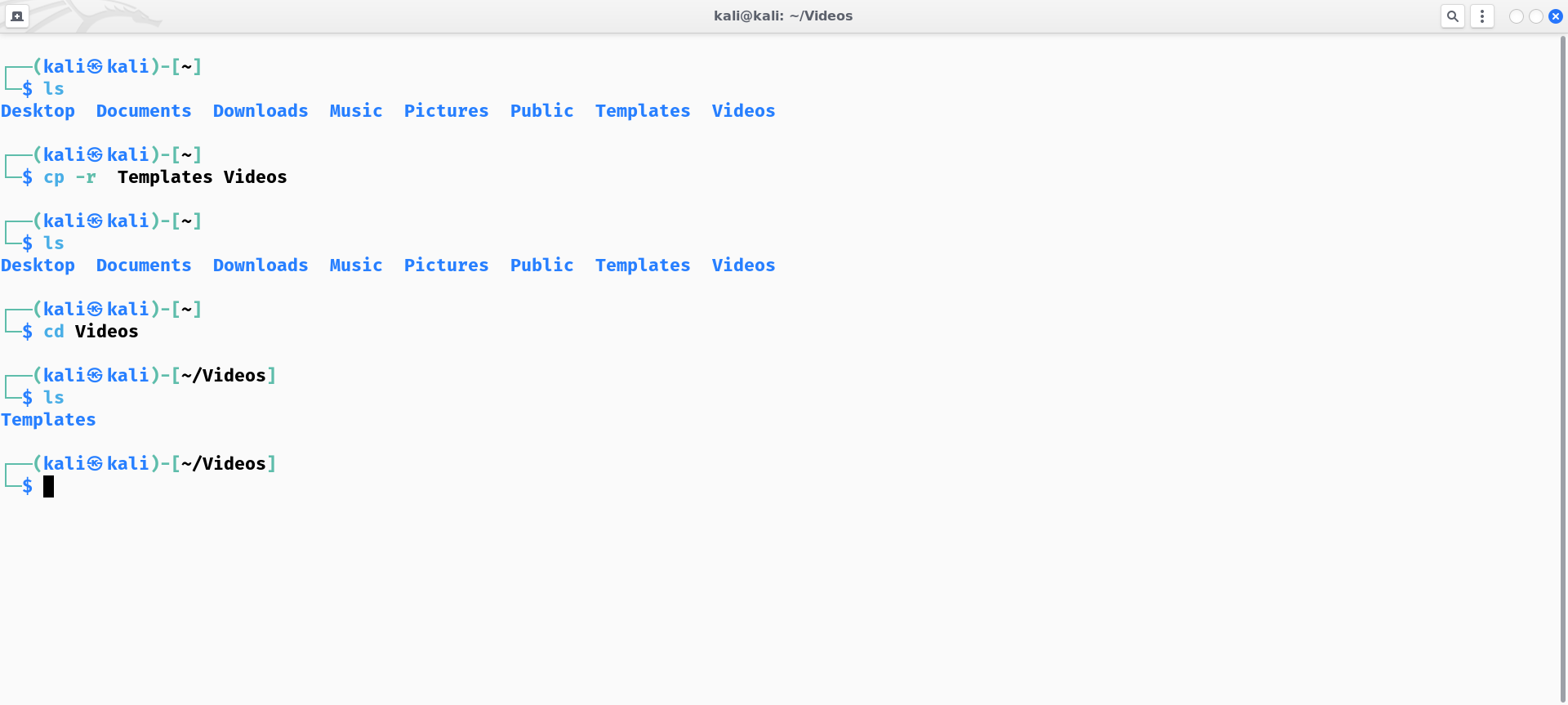
## CP

### Description:

The **cp** command is used to copy files and directories.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -i | This flag prompts the user for confirmation before overwriting existing destination files. |
| 2 | -f | This flag suppresses warnings and prompts, allowing **cp** to overwrite existing destination files without asking for confirmation. |
| 3 | -n | This flag prevents **cp** from overwriting existing destination files. If a file with the same name already exists, **cp** will not overwrite it and will instead display an error message. |

### Implementation:



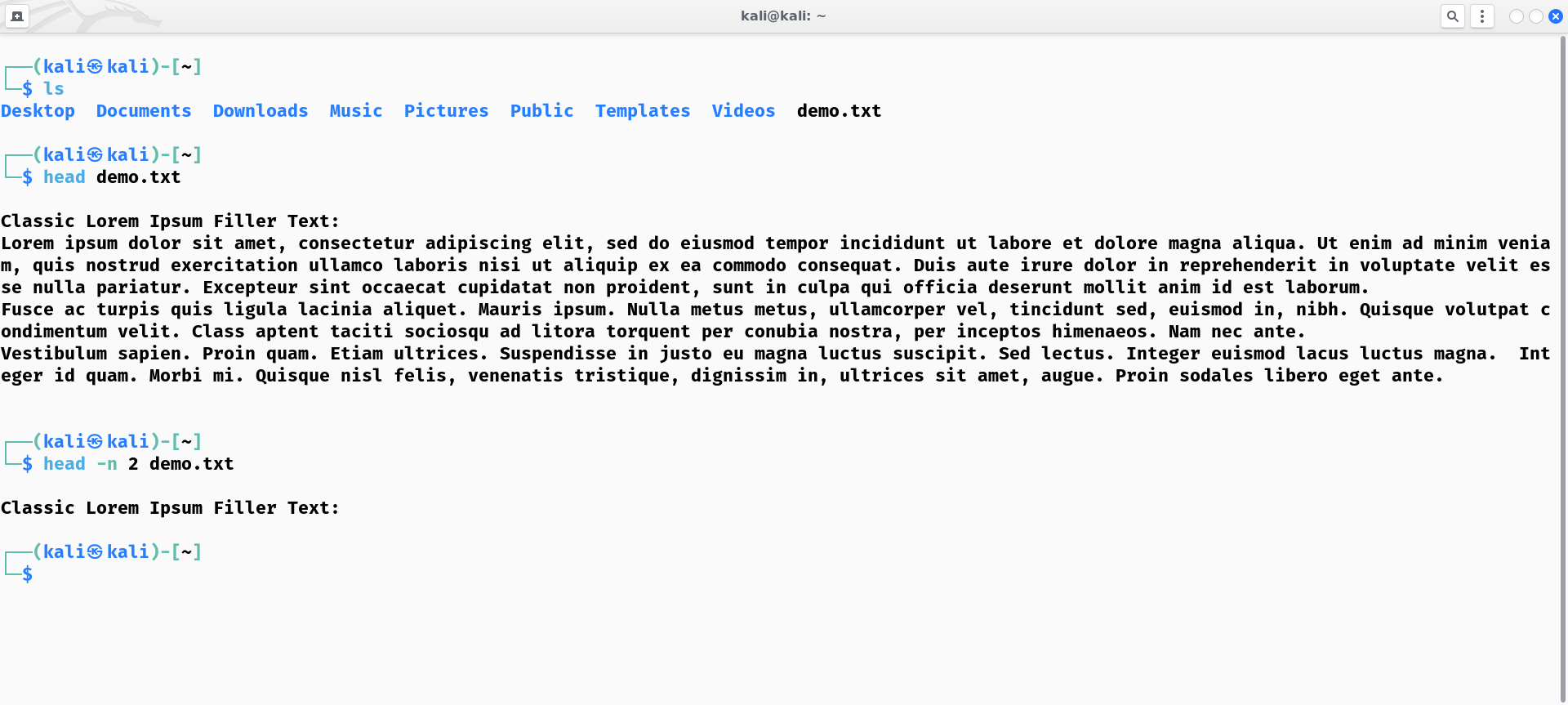
## HEAD

### Description:

The **head** command is used to display the first few lines of a file. It is particularly useful for viewing the beginning of large files without having to load the entire file into memory.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -n num | This flag specifies the number of lines to be displayed from the beginning of the file. By default, **head** displays the first 10 lines. |
| 2 | -c num | This flag specifies the number of bytes to be displayed from the beginning of the file. It overrides the **-n** flag. |
| 3 | -q | This flag suppresses the display of file names when multiple files are specified as arguments. |

### Implementation:



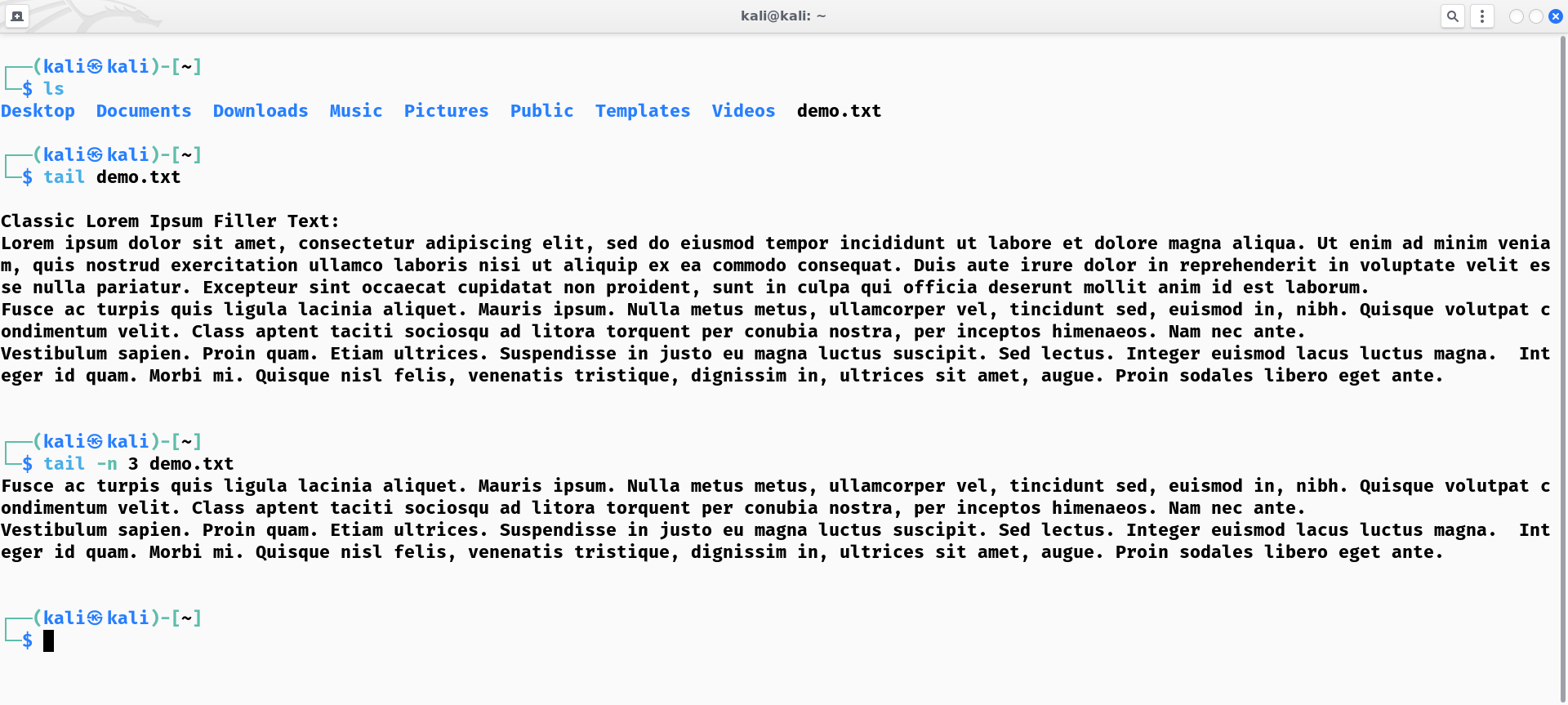
## TAIL

### Description:

The **tail** command is used to display the last few lines of a file. It is particularly useful for viewing the end of log files or continuously updated files.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -n num | This flag specifies the number of lines to be displayed from the end of the file. By default, **tail** displays the last 10 lines. |
| 2 | -c num | This flag specifies the number of bytes to be displayed from the end of the file. It overrides the **-n** flag. |
| 3 | -f | This flag causes **tail** to output appended data in real-time as the file grows. It is commonly used to monitor log files or other continuously updated files. |

### Implementation:



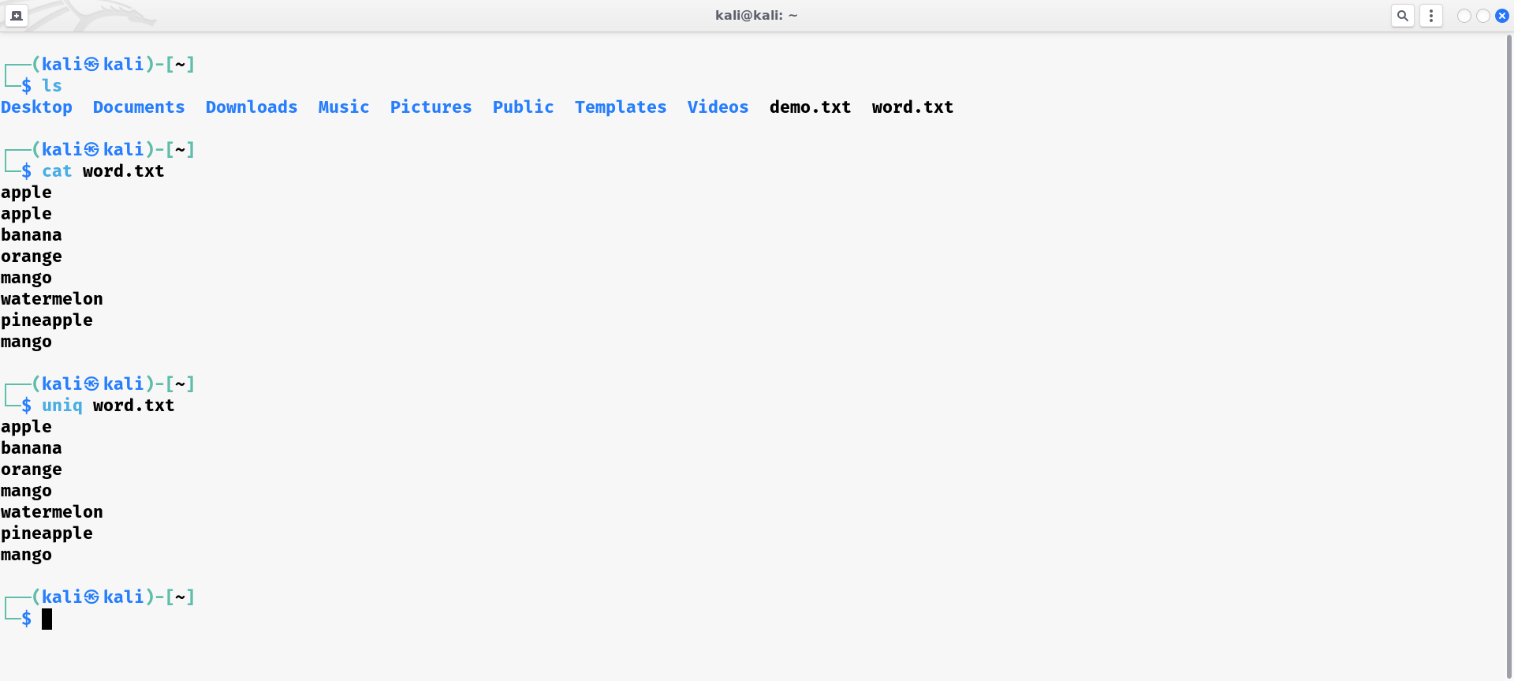
## UNIQ

### Description:

The **uniq** command is used to filter out adjacent duplicate lines in a file. It's often used in conjunction with other commands in shell scripts or pipelines to process text data.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -c | This flag precedes each line with a count of the number of times it occurred in the input. It's useful for counting duplicate lines. |
| 2 | -d | This flag suppresses unique lines and displays only those lines that appear more than once in the input. |
| 3 | -i | This flag makes **uniq** treat uppercase and lowercase letters as equivalent, effectively ignoring differences in case when determining uniqueness. |

### Implementation:



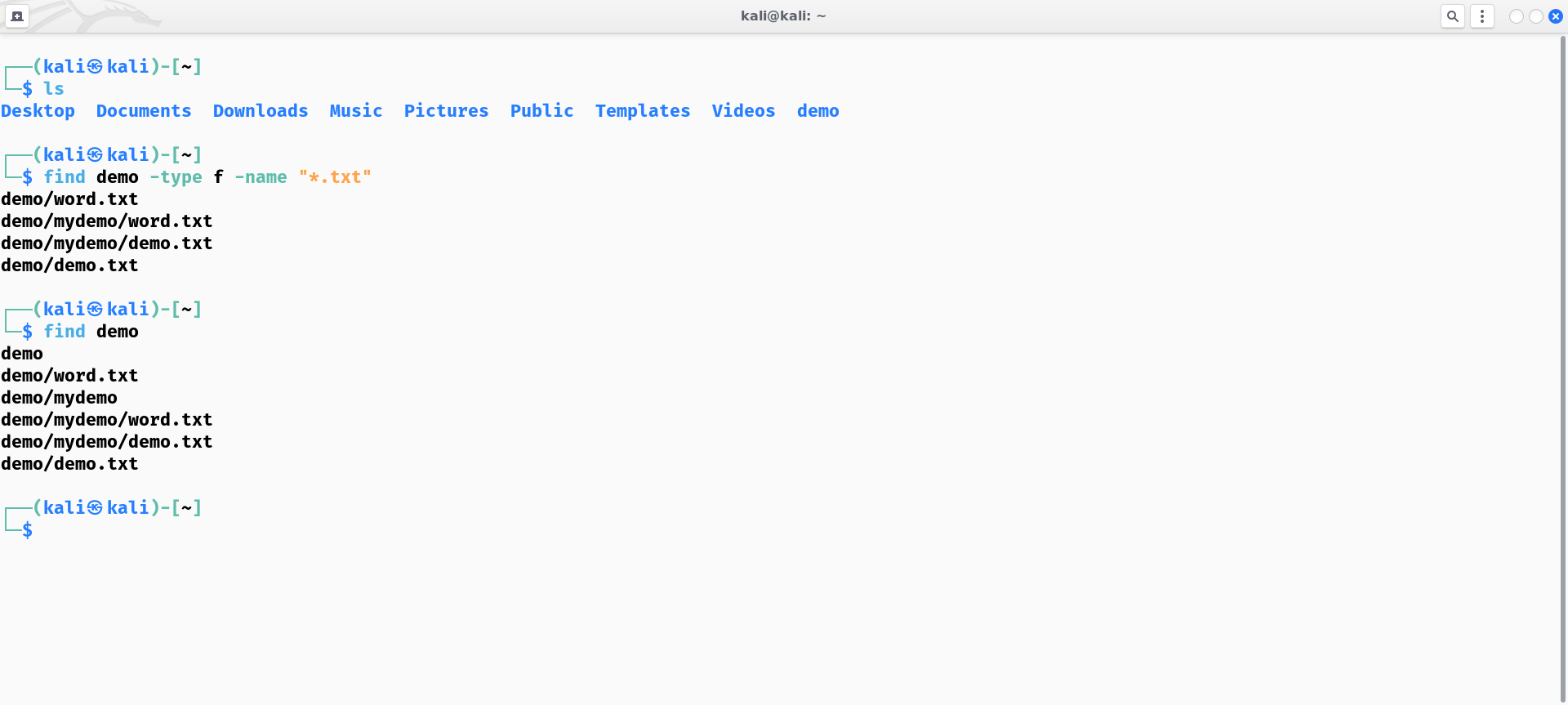
## FIND

### Description:

The **find** command is used to search for files and directories within a specified directory hierarchy. It offers a wide range of options to customize the search criteria.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -name pattern | This flag specifies the name pattern to search for. It supports wildcard characters like **\*** and **?** for matching. |
| 2 | -type type | This flag specifies whether to search for files (**f**) or directories (**d**). |
| 3 | -mtime n | This flag searches for files that were modified approximately n days ago. |

### Implementation:



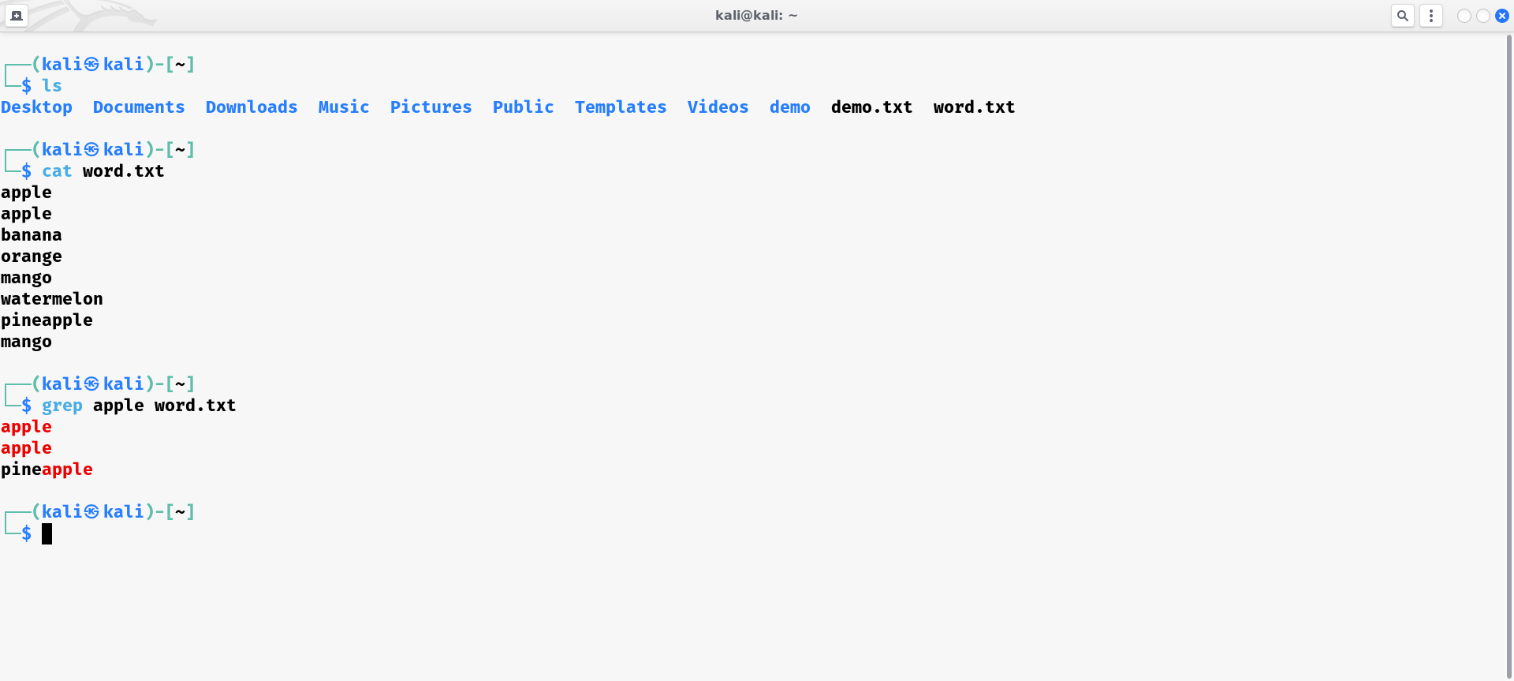
## GREP

### Description:

The **grep** command is used to search for patterns in text files. It's a powerful tool for extracting information from files based on specific criteria.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -i | This flag makes **grep** ignore case distinctions when searching for patterns. It treats uppercase and lowercase letters as equivalent. |
| 2 | -v | This flag selects lines that do not match the specified pattern. |
| 3 | -n | Prefix each line of output with the 1-based line number within its input file |

### Implementation:



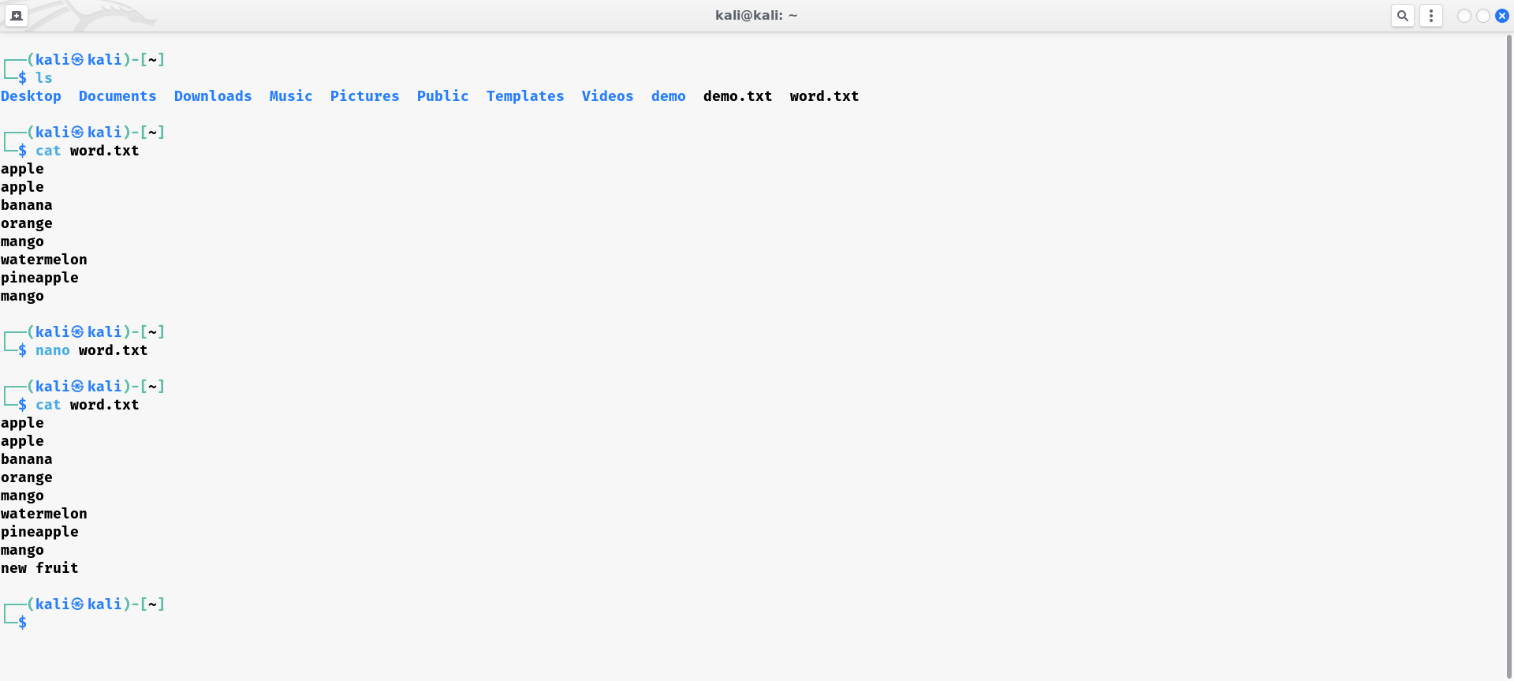
## NANO

### Description:

The **nano** command is a simple text editor, commonly used for editing configuration files and other plain text files. It's particularly popular for its user-friendly interface, making it accessible even to users with minimal experience in command-line text editing.

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -h | This flag prints a brief usage summary and a list of available options for the **nano** command. |
| 2 | -V | This flag prints the version number of the **nano** command. |
| 3 | -c | This flag makes **nano** always display the current cursor position in the status bar. |

### Implementation:



## SU

### Description:

The **su** command is used to switch to another user account. It stands for "substitute user" or "switch user".

|  |  |  |
| --- | --- | --- |
| No. | Option | Description |
| 1 | -l | This flag starts a login shell with the target user's environment, similar to the **-** flag. It's equivalent to **su - username**. |
| 2 | -m | This flag preserves the current environment variables when switching to the target user's account. It prevents the target user's environment variables from overriding the current ones. |
| 3 | -p | This flag uses the target user's login prompt when switching to their account, instead of the default prompt. |

### Implementation:

