Notes 4

Commands

cat (Concatenate)

The `cat` command is used to concatenate and display the contents of files.

- Usage: `cat+[options]+[file]`
- Examples
  - `cat file.txt`: Displays the contents of "file.txt".
  - `cat todo.lst`: Displays the contents of "todo.lst".
  - `cat ~/Pictures/image.png`: Displays "image.png" from within the Pictures folder.

tac (Reverse cat)

The `tac` command is used to concatenate and display the contents of files in reverse order (line by line).

- Usage: `tac+[option]+[file]`
- Examples
  - `tac file.txt`: Displays the contents of "file.txt" in reverse order.
  - `tac file1.txt file2.txt`: Concatenates and displays the contents of both "file1.txt" and "file2.txt" in reverse order.
  - `tac ~/Documents/todo.md`: Displays the contents of the files using absolute path.

head

The `head` command is used to display the first ten lines of a file

- Usage: `head+[option]+[file]`
- Examples
  - `head ~/Documents/Book/dracula.txt`: Displays the first ten lines of the file.
  - `head -5 ~/Documents/Book/dracula.txt`: Displays the first five lines of the file.
  - `head -c 50 ~/Documents/Book/dracula.txt`: Displays the first 50 bytes of the file.

tail

The `tail` command is used to display the last ten lines of a file.

- Usage: `tail+[options]+[file]`
- Examples
  - `tail ~/Documents/Book/dracula.txt`: Displays the last ten lines of the file.
  - `tail -5 ~/Documents/Book/dracula.txt`: Displays the last five lines of the file.
  - `tail -n 5 dracula.txt bible.txt`: Displays the last five lines of multiple files.

cut

The `cut` command is used to extract specific sections from each line of a file.

- Usage: `cut+[options]+[file]`
- Examples:
  - `cut -d ':' -f1 /etc/passwd`: Displays the list of all the users in the system.
  - `cut -c1-5 file.txt`: Extracts the first 5 characters from each line of a text file.
  - `cut -f2-4 file.txt`: Extracts fields 2 to 4 from a tab-separated file.

sort

The `sort` command is used to sort the lines of a file.

- Usage: `sort+[options]+[file]`
- Examples:
  - `sort file.txt`: Sorts the lines of "file.txt" in ascending order.
  - `sort -r file.txt`: Sorts the lines of "file.txt" in descending order.
  - `sort -n file.txt`: Sorts the lines of "file.txt" numerically.

wc

The `wc` command is used to count the number of lines, words, and characters in a file.

- Usage: `wc+[options]+[file]`
- Examples:
  - `wc file.txt`: Counts the number of lines, words, and characters in "file.txt".
  - `wc -l file.txt`: Counts the number of lines in "file.txt".
  - `wc -w file.txt`: Counts the number of words in "file.txt".

diff

The `diff` command is used to compare files line by line.

- Usage: `diff+[options]+file1+file2`
- Examples:
  - `diff file1.txt file2.txt`: Compares "file1.txt" and "file2.txt" and shows the differences.
  - `diff -u file1.txt file2.txt`: Outputs a unified diff format.
  - `diff -rq dir1 dir2`: Recursively compares directories "dir1" and "dir2".

grep

The `grep` command is used to search for patterns in files.

- Usage: `grep+[options]+pattern+[file]`
- Examples:
  - `grep "keyword" file.txt`: Searches for occurrences of "keyword" in "file.txt".
  - `grep -i "pattern" file.txt`: Performs a case-insensitive search.
  - `grep -r "pattern" directory`: Recursively searches for "pattern" in all files in the "directory".