Notes 8 - Handling Text Files 1

CAT

- Definition: Concatenates and displays file contents.
- Usage: cat [OPTION]... [FILE]...
- Common Options:

- -n : Number all output lines

- -b : Number non-empty output lines

- -s : Squeeze multiple blank lines into one

Examples

- Display a file cat file.txt

- Display multiple files cat file1.txt file2.txt

- Display with line numbers cat -n file.txt

- Create a new file cat > newfile.txt Hello World Ctrl+D

- Append to a file cat file1.txt >> file2.txt

TAC

- Definition: Concatenates and displays file contents in reverse order (last line first).
- Usage: tac [OPTION]... [FILE]...
- Common Options:

- -b : Attach separator before instead of after

- -s : Use specified string as separator instead of newline

Examples

- Display file in reverse order tac file.txt

- Reverse multiple files tac file1.txt file2.txt

- Practical example - reversing log entries echo -e "Line 1\nLine 2\nLine 3" | tac

Output: Line 3, Line 2, Line 1

HEAD

- Definition: Outputs the first part of files.
- Usage: head [OPTION]... [FILE]...
- Common Options:

- -n [NUM] : Print first NUM lines (default: 10)

- -c [NUM] : Print first NUM bytes

- -q : Quiet mode (never print headers)

- -v : Verbose mode (always print headers)

Examples:

- Show first 10 lines (default) head file.txt

- Show first 5 lines head -n 5 file.txt

- Show first 100 bytes head -c 100 file.txt

- Show first 3 lines of multiple files head -n 3 file1.txt file2.txt

TAIL

- Definition: Outputs the last part of files.
- Usage: tail [OPTION]... [FILE]...
- Common Options:

- -n [NUM] : Print last NUM lines (default: 10)

- -c [NUM] : Print last NUM bytes

- -f : Follow (output appended data as file grows)

- -F : Same as -f but retries if file is inaccessible

Examples:

- Show last 10 lines (default) tail file.txt

- Show last 5 lines tail -n 5 file.txt

- Follow a log file in real-time tail -f /var/log/syslog

- Show last 100 bytes tail -c 100 file.txt

- Show and follow multiple files tail -f file1.txt file2.txt

CUT

- Definition: Removes sections from each line of files.
- Usage: cut OPTION... [FILE]...
- Common Options:

- -c [LIST] : Select only these characters

- -f [LIST] : Select only these fields

- -d [DELIM] : Use DELIM instead of TAB for field delimiter

- --complement : Complement the set of selected items

Examples:

- Extract first 10 characters from each line cut -c 1-10 file.txt

- Extract specific columns (1,3,5) cut -c 1,3,5 file.txt

- Extract first field from CSV cut -d ',' -f 1 data.csv

- Extract fields 2-4 from tab-delimited file cut -f 2-4 data.tsv

- Extract all except first field cut -d ' ' -f 2- users.txt

SORT

- Definition: Sorts lines of text files.
- Usage: sort [OPTION]... [FILE]...
- Common Options:

- -r : Reverse the result of comparisons

- -n : Compare according to string numerical value

- -u : Output only the first of an equal run

- -k [POS] : Sort by a key (field position)

- -t [SEP] : Use SEP as field separator

- -o [FILE] : Write output to FILE instead of stdout

Examples:

- Sort alphabetically sort file.txt

- Sort numerically sort -n numbers.txt

- Sort in reverse order sort -r file.txt

- Sort by second field (tab-delimited) sort -t $'\t' -k 2 data.tsv

- Sort by third field numerically sort -t ',' -k 3n data.csv

- Sort and remove duplicates sort -u file.txt

- Sort and save to file sort input.txt -o sorted.txt

WC

- Definition: Prints newline, word, and byte counts for each file.
- Usage: wc [OPTION]... [FILE]...

- Common Options:
- -l : Print the newline count
- -w : Print the word count
- -c : Print the byte count
- -m : Print the character count
- -L : Print the length of the longest line

Examples:

- Get all counts (lines, words, bytes) wc file.txt `Output: 10 50 300 file.txt`
- Count lines only wc -l file.txt
- Count words only wc -w file.txt
- Count characters wc -m file.txt
- Find longest line length wc -L file.txt
- Count lines in multiple files wc -l *.txt
- Count lines from piped input echo "Hello World" | wc -w