

## Notes #8

- **cat:**

cat (concatenate) is used to read, display, combine, and create text files.

Usage / Formula:

```
cat [options] [file ...]
```

Examples:

```
Display contents of file.txt:  
cat file.txt  
  
Combine contents of file.txt:  
cat file1.txt file2.txt > combined.txt
```

- **tac:**

tac is cat in reverse — it prints files line-by-line from bottom to top.

Examples:

```
Displays the file starting with its last line"  
tac file.txt  
  
Displays the both files starting with its last line:  
tac a.txt b.txt
```

- **head:**

head prints the first lines of a file (default = 10 lines).

Command line;

```
head [options] [file]
```

Examples:

```
It shows the first 10 lines:  
head file.txt  
  
Show first 5 lines:
```

```
head -n 5 file.txt

Show first 20 bytes:
head -c 20 file.txt
```

- **tail:**

tail prints the last lines of a file (default = 10 lines).

Command line:

```
tail [options] [file]
```

Examples:

```
Show last 10 lines:
tail file.txt

Show last 3 lines:
tail -n 3 file.txt
```

- **cut:**

cut extracts columns, fields, or character ranges from lines of a text file.

Command line:

```
cut OPTION... [FILE]
```

Examples:

```
Extract fields 1 and 3 from a CSV file:
cut -f1,3 -d', ' data.csv

Extract first five characters of each line:
cut -c1-5 file.txt
```

- **sort:**

sort arranges lines of text in alphabetical or numeric order.

Command line:

```
sort [options] [file]
```

Common options:

-n → numeric sort

-r → reverse

-k → key (column)

Example:

```
Sort alphabetically:  
sort names.txt
```

```
Sort numbers in ascending numeric order:  
sort -n numbers.txt
```

- **wc:**

wc (word count) prints the number of lines, words, and bytes in a file.

Command line:

```
wc [options] [file]
```

Common options:

-l → lines -w → words -c → bytes

Examples:

```
Prints lines, words, and bytes:  
wc file.txt
```

```
Count line only:  
wc -l file.txt
```

```
Word count of redirected input:  
wc -w < file.txt
```