# Note 8

## awk

***Definition***: `is a powerful text-processing command used to pattern-match and manipulate text, especially structured data like columns in a file.`

***Usage/Formula***: `awk 'pattern { action }' filename`

***Examples***:

1. Print the first column of a file

   ```
   awk '{ print $1 }' file.txt
   ```

2. Print lines where the second column is greater than 50

   ```
   awk '$2 > 50' file.txt
   ```

3. Print line number and line content

   ```
   awk '{ print NR, $0 }' file.txt
   ```

## sed

***Definition***: `(Stream Editor) is used to perform basic text transformations on an input stream or file.`

***Usage/Formula***: `sed 's/pattern/replacement/' filename`

***Examples***:

1. Replace the first instance of "apple" with "orange" in each line

   ```
   sed 's/apple/orange/' file.txt
   ```

2. Replace all instances of "dog" with "cat" in each line

   ```
   sed 's/dog/cat/g' file.txt
   ```

3. Delete lines containing the word "error"

   ```
   sed '/error/d' file.txt
   ```

## less

***Definition***: `is a pager program used to view the contents of a file one screen at a time, allowing backward and forward navigation.`

***Usage/Formula***: `less filename`

***Examples***:

1. View a file one screen at a time

   ```
   less file.txt
   ```

2. View the output of a command using a pipe

   ```
   ps aux | less
   ```

3. Search for a keyword inside less (e.g., "user") Inside less, type:

   ```
   /user
   ```

## >

---

***Definition***: `Redirects standard output to a file, overwriting the file if it exists.`

Usage/Formula: `command > file`

Examples:

1. Redirect output to a new file

   echo "Hello World" > hello.txt

2. Save a list of files into a file

   ls > file_list.txt

## >>

---

***Definition***: `Appends standard output to the end of a file without overwriting existing content.`

Usage/Formula: `command >> file`

Examples:

1. Append a line to a log file

   echo "New log entry" >> log.txt

2. Append disk usage info to a report

   du -h >> disk_report.txt

# | (Pipe)

***Definition***: `The pipe (|) passes the output of one command as input to another.`

***Usage/Formula***: `command1 | command2`

***Examples***:

1. Find the number of lines containing "error"

   grep "error" log.txt | wc -l

2. Sort the list of running processes

   ps aux | sort -k 3 -nr | head

3. Display contents of a file with paging

   cat bigfile.txt | less