

NOTES 8

AWK

- **Definition:**
 - A text-processing tool and programming language used for pattern scanning and text manipulation. It processes input line by line and allows you to extract and transform data based on patterns.
- **Usage:**
 - `awk 'pattern {action}' file`
- **Examples:**
 - **Print the first column of a file:**

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

- **Print rows where the second column equals 50:**

```
awk '$2 == 50 {print}' data.txt
```

- **Calculate the total of the third column:**

```
awk '{sum += $3} END {print sum}' data.txt
```

SED

- **Definition:**
 - A stream editor for parsing and transforming text. It is commonly used for text substitution, insertion, deletion, and pattern matching in files.
- **Usage:**
 - `sed [options] 'command' file`
- **Examples:**
 - **Replace "hello" with "hi" in a file:**

```
sed 's/hello/hi/' file.txt
```

- **Delete lines containing the word "error":**

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

- Insert a line after the second line:

```
sed '2a\New Line' file.txt
```

LESS

- **Definition:**
 - A pager utility for viewing large files or outputs line by line. It is faster than opening the entire file and supports forward/backward navigation.
- **Usage:**
 - `less file`
- **Examples:**
 - Open a log file for navigation:

```
less largefile.log
```

- Search for a keyword while in **less**:
 - Type `/keyword` and press `Enter`.
- Scroll forward or backward:
 - Forward: Press `Space`
 - Backward: Press `b`

>

- **Definition:**
 - Redirects the output of a command to a file. If the file exists, it will be overwritten.
- **Usage:**
 - `command > file`
- **Examples:**
 - Save the output of **ls** to a file:

```
ls > directory_list.txt
```

- Save the first 10 lines of a file:

```
head file.txt > top10.txt
```

>>

- **Definition:**
 - Redirects the output of a command to a file, appending it to the end of the file without overwriting.
- **Usage:**
 - `command >> file`
- **Examples:**
 - **Append the output of `echo` to a log file:**

```
echo "Log entry" >> logfile.txt
```

- **Append the last 5 lines of a file:**

```
tail -5 file.txt >> summary.txt
```

|

- **Definition:**
 - Pipes the output of one command as input into another command, allowing command chaining.
- **Usage:**
 - `command1 | command2`
- **Examples:**
 - **Count the number of lines in the output of `ls`:**

```
ls | wc -l
```

- **Filter and sort log entries containing "error":**

```
grep "error" logfile.txt | sort
```

- **Combine multiple commands to get disk usage summary:**

```
du -sh * | sort -h
```