

## Notes 8

# AWK Command

Definition: is a powerful programming language and command-line utility for text processing and data extraction. It is used primarily for pattern scanning and processing. Usage/Formula: `awk 'pattern {action}' input_file` Examples:

- 1. `awk -F, '{print $1}' file.csv`
- 2. `awk '{sum += $1} END {print sum}' numbers.txt`
- 3. `awk '/pattern/' file.txt`

# SED Command

Definition: sed(stream editor) is a non-interactive command-line utility for parsing and transforming text. It is commonly used for text substitution, deletion, and insertion. Usage/Formula: `sed 's/pattern/replacement/flags' input_file` Examples:

- 1. `sed 's/foo/bar/g' file.txt`
- 2. `sed '/pattern/d' file.txt`
- 3. `sed '/pattern/a\New line of text' file.txt`

# Less Command

Definition: is a command-line utility for viewing the content of a file one screen at a time. It allows for forward and backward navigation through the file. Usage/Formula: `less file` Examples:

- 1. `less file.txt`
- 2. `less file.txt`
- 3. `less file1.txt file2.txt`

# Redirection and Pipes

Definition: The `>` operator redirects the output of a command to a file, overwriting the file if it already exists. Usage/Formula: `command > file` Examples:

- 1. `echo "Hello,world!" > greeting.txt`
- 2. `ls > file_list.txt`

# Append

Definition: The `>>` operator appends the output of a command to the end of a file without overwriting the existing content. Usage/Formula: `command >> file` Examples:

- 1. `echo "Additional line" >> file.txt`
- 2. `ls >> file_list.txt`

# Pipe

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

Usage/Formula: command1 | command2 Examples:

- 1. ls | grep 'pattern'
- 2. ps aux | grep 'process\_name'
- 3. ls | wc -l