

Commands and Redirection

1. `awk`

Definition: `awk` is a programming language designed for text processing and data manipulation.

Usage/Formula: `awk 'pattern {action}' input-file(s)`

Examples:

- Print the first column of a text file:

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

- Print the first and second column of a text file:

```
awk '{print $1,$2}' file.txt
```

- Print the first and second column of a text file with text before it:

```
awk '{print "username is: " $1, "with home drive: " $2}' file.txt
```

2. `sed`

Definition: `sed` (stream editor) is a utility that parses and transforms text.

Usage/Formula: `sed 's/pattern/replacement/flags' file`

Examples:

- Replace "cat" with "dog" in a file:

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

- Delete lines containing "delete_me":

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

- Replace all occurrences of "foo" with "bar" globally:

```
sed 's/foo/bar/g' file.txt
```

3. less

Definition: `less` is a program similar to `more`, but which allows backward movement in the file as well as forward movement.

Usage/Formula: `less file`

Examples:

- View a file:

```
less file.txt
```

- Search for a string "error" within the file:

```
less file.txt  
/error
```

4. alias

Definition: `alias` is used to create shortcuts for long commands or to redefine existing commands.

Usage/Formula: `alias name='command'`

Examples:

- Create an alias for listing files in long format:

```
alias ll='ls -l'
```

- Create an alias for updating the system:

```
alias update='sudo apt update && sudo apt upgrade'
```

Redirection and Piping

1. >

Definition: Redirects the output of a command to a file, overwriting the existing content.

Examples:

- Write the output of `ls` to a file:

```
ls > file.txt
```

- Overwrite the contents of a file with echoed text:

```
echo "This is the last Notes File for CIS-106 Spring 2024" > file.txt
```

2. >>

Definition: Appends the output of a command to a file without overwriting it.

Examples:

- Append directory listing to a file:

```
ls >> file.txt
```

- Append text to a file:

```
echo "Another line" >> file.txt
```

3. |

Definition: Passes the output of one command as input to another command.

Examples:

- List files and pass them to `grep` to find a specific file:

```
ls | grep "specific_file"
```

- Display the number of files in a directory:

```
ls | wc -l
```