

Notes 6

Command: `awk`

Definition

`awk` is a powerful text processing tool used to manipulate and analyze text files or streams, especially useful for pattern scanning and data extraction.

Usage/Formula

```
bash awk 'pattern { action }' filename
```

Examples

```
Print the first column of a file:
```

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

```
Print lines where the second column equals "apple":
```

```
awk '$2 == "apple" { print $0 }' data.txt
```

```
Calculate the sum of values in the third column:
```

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

Command: `sed`

Definition

`sed` (stream editor) is used to perform basic text transformations on an input stream (a file or input from a pipeline). Usage/Formula

```
sed 's/pattern/replacement/' filename
```

Examples

```
Replace "apple" with "orange" in each line:
```

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

```
Delete the second line:
```

```
sed '2d' file.txt
```

```
Replace all occurrences of "foo" with "bar":
```

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

Command: `less`

Definition

`less` is a pager program used to view (but not change) the contents of a text file one screen at a time.

Usage/Formula

```
less filename
```

Examples

```
View contents of a large file:
```

```
less largefile.txt
```

```
Search for a keyword inside less:
```

```
less log.txt
```

Then type `/error` to search for "error"

```
Pipe output to less for easier viewing:
```

```
ps aux | less
```

Operator: `>`

Definition

Redirects standard output to a file, overwriting the file if it already exists. Usage/Formula

```
command > filename
```

Examples

```
Save the output of ls to a file:
```

```
ls > files.txt
```

```
Overwrite a file with current date:
```

```
date > now.txt
```

```
Save error messages:
```

```
echo "error" > error.log
```

Operator: >>

Definition

Redirects standard output to a file, appending to the end of the file if it exists. Usage/Formula

```
command >> filename
```

Examples

```
Append output of ls to a log file:
```

```
ls >> log.txt
```

```
Add a new line to an existing file:
```

```
echo "New entry" >> notes.txt
```

```
Append date to a log file:
```

```
date >> log.txt
```

Operator: | (Pipe)

Definition

Takes the output of one command and uses it as the input to another command. Usage/Formula

`command1 | command2`

Examples

Count the number of lines in a directory listing:

`ls -l | wc -l`

Find running processes with "python":

`ps aux | grep python`

Display disk usage in human-readable format and sort:

`du -h | sort -h`