

Searching in Files

Using Pipes

What You Will Learn

- How to search the contents of files.
- What pipes are and how to use them.

The `grep` Command

`grep` Display lines matching a pattern.

```
grep pattern file
```

grep Options

- i Perform a search, ignoring case.
- c Count the number of occurrences in a file.
- n Precede output with line numbers.
- v Invert Match. Print lines that don't match.

Demo - grep

The `file` Command

`file file_name` Display the file type.

```
$ file sales.data
```

```
sales.data: ASCII text
```

```
$ file *
```

```
bin: directory
```

```
jason.tar: POSIX tar archive
```

Searching for Text in Binary Files

`strings` Display printable strings.

Pipes

| Pipe symbol

`command-output | command-input`

Pipes

```
grep pattern file
```

```
cat file | grep pattern
```

The `cut` Command

`cut [file]`

Cut out selected portions of file. If file is omitted, use standard input.

cut Options

- `-d delimiter` Use delimiter as the field separator.
- `-f N` Display the Nth field.

Demo - strings/grep

Searching and Pipe Example

- Find all users named "bob" in /etc/passwd.
- Print account name and real name.
- Print in alphabetical order by account name.
- Print in a tabular format.

Piping Output to a Pager

- more
- less

Demo - more / less

Summary

- `grep`
- `file`
- `cut`
- `tr`
- `column`
- `more`
- `less`
- **Pipes**