Searching in Bash

How to search files and directories based on various attributes, using different methods. Moreover, this lesson encompasses the ways to search for files based on their contents and patterns.

locate

Definition:

The command `locate` is a quick way to search for the locations of files and directories.

Syntax:

```
locate [options] name(s)
```

Options:

Option	Description
-q	To suppress error messages, such as those that might be returned in the event that the user does not have permission to access designated files or directories.
-n	This option followed by an integer limits the results to a specific number.
-i	To perform a case-insensitive search.
-V	To show which version of locate is

	used, including whether it is locate or slocate.

Example:

- To search for all files named ‘file1’ and all directories named ‘dir1’, for which the user had access permission:

```
locate file1 dir1
```

- To display all the files in the system with **“.jpeg”** extension:

```
locate "*.jpeg"
```

- In the above example, in order to display only **10** results:

```
locate -n 10 "*.jpeg"
```

find

Definition:

`find` is a very powerful command which is used to search for specific patterns of texts within files and directories.

Syntax:

```
find [path...] [expression]
```

Tests (filters):

`find` has the ability to filter which files and folder it “selects”. They are called tests.

Test	Description
-type f	Selects files.
-type d	Selects directories.

-name	True if the base of the file name (the path with the leading directories removed) matches shell pattern pattern.
-iname	Search without regard for text case.
-prune	To ignore a whole directory tree.
-path	This is the exact same as name, except that it doesn't only apply on the filename, but the whole path.
-not	Return only results that do not match the test case.

Examples:

- Searching for file with a particular name:

To find "img.png" file in `runner/projects`:

```
find runner/projects -name "img.png"
```

```
find runner/projects/ -name "img.png"
```

- Searching files with specific pattern:

To find all header files in `runner/projects`:

```
find runner/projects -name "*.h"
```

```
find runner/projects -name "*.h"
```

- Deleting files using find command:

To delete `square.h` file which is in `runner/projects`:

```
find runner/projects -name "square.h" -delete
```

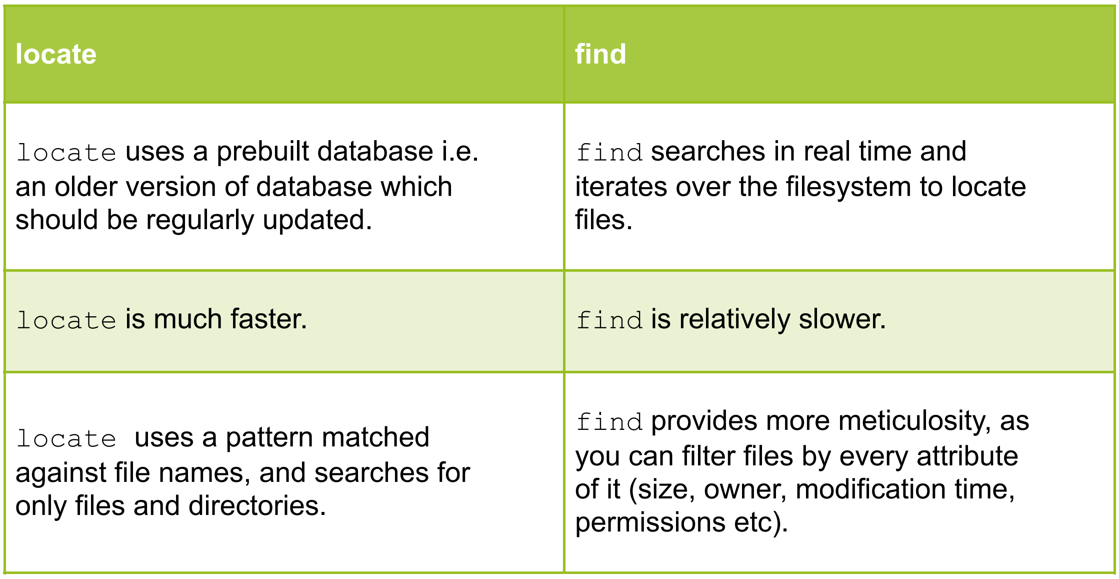
- Excluding files with specific pattern from the search:

If we want to exclude all `.h` files from the search, and display all other files in the "runner/projects" directory:

```
find runner/projects -not -name "*.h"
```

```
find runner/projects -not -name "*.h"
```

Difference between `find` and `locate`



locate	find
`locate` uses a prebuilt database i.e. an older version of database which should be regularly updated.	`find` searches in real time and iterates over the filesystem to locate files.
`locate` is much faster.	`find` is relatively slower.
`locate` uses a pattern matched against file names, and searches for only files and directories.	`find` provides more meticulosity, as you can filter files by every attribute of it (size, owner, modification time, permissions etc).