How it works...

When the `-F` parameter is used with `ls`, all entries are appended with some type of file characters such as `@`, `*`, `|`, and so on. For directories, entries are appended with the `/` character. We use `grep` to filter only entries ending with the `/$` end-of-line indicator.

The first character of any line in the `ls -d` output is the type of file character. For a directory, the type of file character is `"d"`. Hence, we use `grep` to filter lines starting with `"d"`. `^` is a start-of-line indicator.

The `find` command can take the parameter type as directory and `maxdepth` is set to `1` since we don't want it to search inside the subdirectories.

Fast command-line navigation using pushd and popd

When dealing with multiple locations on a terminal or shell prompt, our common practice is to copy and paste the paths. However, this is ineffective when there is only command-line access without a GUI. For example, if we are dealing with locations `/var/www`, `/home/slynux`, and `/usr/src`, when we need to navigate to these locations one by one, it is really difficult to type the path every time when we need to switch between the paths. Hence, the **command-line interface** (**CLI**) based navigation techniques such as `pushd` and `popd` are used. Let us see how to use them in this recipe.

Getting ready

`pushd` and `popd` are used to switch between multiple directories without the copying and pasting of directory paths. `pushd` and `popd` operate on a stack. We know that a stack is a **last in first out** (**LIFO**) data structure. It will store the directory paths in a stack and switch between them using the push and pop operations.

How to do it...

We omit the use of the `cd` command while using `pushd` and `popd`:

1. To push and change a directory to a path, use the following command:

   ```
   ~ $ pushd /var/www
   ```

   Now the stack contains `/var/www ~` and the current directory is changed to `/var/www`.

2. Now, again push the next directory path as follows:

   ```
   /var/www $ pushd /usr/src
   ```

Now the stack contains `/usr/src /var/www ~` and the current directory is `/usr/src`.

You can similarly push as many directory paths as needed.

3. To view the stack contents, use the following command:

```
$ dirs
/usr/src /var/www ~ /usr/share /etc
0            1               2  3               4
```

4. Now when you want to switch to any path in the list, number each path from `0` to `n`, then use the path number for which we need to switch, for example:

```
$ pushd +3
```

Now it will rotate the stack and switch to the `/usr/share` directory.

`pushd` will always add paths to the stack, to remove paths from the stack use `popd`.

5. To remove a last pushed path and change directory to the next directory, use the following command:

```
$ popd
```

Suppose the stack is `/usr/src /var/www ~ /usr/share /etc` such that the current directory is `/usr/src`, then `popd` will change the stack to `/var/www ~ /usr/share /etc` and change the directory to `/var/www`.

6. To remove a specific path from the list, use `popd +num`:

`num` is counted as `0` to `n` from left to right.

There's more...

Let's go through the essential directory navigation practices.

Most frequently used directory switching

`pushd` and `popd` can be used when there are more than three directory paths used. But when you use only two locations, there is an alternative and easier way. That is `cd -`.

The current path is `/var/www`.

```
/var/www $  cd /usr/src
/usr/src $ # do something
```

Now, to switch back to `/var/www`; you don't have to type again, but just execute:

```
/usr/src $ cd -
```