For example:

```
$ watch ls
```

Or

```
$ watch 'COMMANDS'
```

For example:

```
$ watch 'ls -l | grep "^d"'
# list only directories
```

This command will update the output at a default interval of two seconds.

We can also specify the time interval at which the output needs to be updated, by using `-n SECONDS`. For example:

```
$ watch -n 5 'ls -l'
#Monitor the output of ls -l at regular intervals of 5 seconds
```

There's more

Let's explore an additional feature of the `watch` command.

Highlighting the differences in the watch output

In `watch`, there is an option for updating the differences that occur during the execution of the command at an update interval to be highlighted using colors. Difference highlighting can be enabled by using the `-d` option as follows:

```
$ watch -d 'COMMANDS'
```

Logging access to files and directories

Logging of file and directory access is very helpful to keep a track of changes that are happening to files and folders. This recipe will describe how to log such accesses.

Getting ready

The `inotifywait` command can be used to gather information about file accesses. It doesn't come by default with every Linux distro. You have to install the `inotify-tools` package by using a package manager. It also requires the Linux kernel to be compiled with `inotify` support. Most of the new GNU/Linux distributions come with `inotify` enabled in the kernel.

How to do it...

Let's walk through the shell script to monitor the directory access:

```
#/bin/bash
#Filename: watchdir.sh
#Description: Watch directory access
path=$1
#Provide path of directory or file as argument to script

inotifywait -m -r -e create,move,delete $path  -q
```

A sample output is as follows:

```
$ ./watchdir.sh .
./ CREATE new
./ MOVED_FROM new
./ MOVED_TO news
./ DELETE news
```

How it works...

The previous script will log events, create, move, and delete files and folders from the given path. The `-m` option is given for monitoring the changes continuously, rather than going to exit after an event happens, and `-r` enables a recursive watch of the directories (symbolic links are ignored). Finally, `-e` specifies the list of events to be watched and `-q` is to reduce the verbose messages and print only the required ones. This output can be redirected to a log file.

We can add or remove the event list. Important events available are as follows:

Event	Description
`access`	When a read happens to a file.
`modify`	When file contents are modified.
`attrib`	When metadata is changed.
`move`	When a file undergoes a move operation.
`create`	When a new file is created.
`open`	When a file undergoes an open operation.
`close`	When a file undergoes a close operation.
`delete`	When a file is removed.