2. In order to run the script at the fifth, sixth, and seventh hours on all days, use:

```
00 5,6,7 * * /home/slynux/test.sh
```

3. Execute `script.sh` at every hour on Sundays as follows:

```
00 */12 * * 0 /home/slynux/script.sh
```

4. Shutdown the computer at 2 A.M. everyday as follows:

```
00 02 * * * /sbin/shutdown -h
```

5. Now, let us see how to schedule a cron job. You can execute the `crontab` command in multiple ways to schedule the scripts.

   Use the `-e` option to `crontab` to start editing the cron table:

```
$ crontab -e
02 02 * * * /home/slynux/script.sh
```

   When `crontab -e` is entered, the default text editor (usually vi) is opened up and the user can type the cron jobs and save it. The cron jobs will be scheduled and executed at specified time intervals.

6. There are two other methods we usually use when we invoke the crontab command inside a script for scheduling tasks:

   1. Create a text file (for example, `task.cron`) with the cron job in it, and then run the `crontab` with this filename as the command argument:

```
$ crontab task.cron
```

   2. Or, specify the cron job inline without creating a separate file. For example:

```
crontab<<EOF
02 * * * * /home/slynux/script.sh
EOF
```

   The cron job needs to be written in between `crontab<<EOF` and `EOF`.

How it works...

Each cron table consists of six sections in the following order:

- `Minute` (0 - 59)
- `Hour` (0 - 23)
- `Day` (1 - 31)
- `Month` (1 - 12)
- `Weekday` (0 - 6)
- `COMMAND` (the script or command to be executed at the specified time)

The first five sections specify the time at which an instance of the command is to be executed. There are a few additional options to specify the time schedule.

An asterisk (`*`) is used to specify that the command should be executed at every instance of time. That is, if `*` is written in the `Hour` field in the cron job, the command will be executed for every hour. Similarly, if you would like to execute the command at multiple instances of a particular time period, specify the time period separated by a comma in the corresponding time field (for example, for running the command at the fifth minute and tenth minute, enter `5,10` in the `Minute` field). We also have another nice option to run the command at particular divisions of time. Use `*/5` in the minutes field for running the command at every five minutes. We can apply this to any time field. A cron table can consist of one or more lines of cron jobs and each line in the cron table is a single job.

Cron jobs are executed with privileges with which the `crontab` command was executed. If you need to execute commands that require higher privileges, such as a command for shutting down the computer, run the `crontab` command as root.

The commands specified in a cron job are written with the full path to the command. This is because the environment in which a cron job is executed is different from the one that we execute on a terminal. Hence, the `PATH` environment variable may not be set. If your command requires certain environment variables to be set for running, you should explicitly set the environment variables.

There's more...

The `crontab` command has more options. Let's see a few of them.

Specifying environment variables

Many of the commands require environment variables to be set properly for execution. We can set environment variables by inserting a line with a variable assignment statement in the cron table of the user.

For example, if you are using a proxy server for connecting to the Internet, to schedule a command that uses the Internet you have to set the HTTP proxy environment variable `http_proxy`. It can be done as follows:

```
crontab<<EOF
http_proxy=http://192.168.0.3:3128
00 * * * * /home/slynux/download.sh
EOF
```