Running commands at system start up/boot

Running specific commands when the system starts (or, boots) is a common requirement at times. There are a lot of ways to achieve this, and using `cron` is one of them (the others being adding your commands to `/etc/rc.d` but that's not guaranteed to be the same across distros).

To run a command at boot, add the following line to your `crontab`:

```
@reboot command
```

This will run the command as your user at runtime. To run the command as root, edit root's `crontab`.

Viewing the cron table

We can list these existing cron jobs using the `-l` option:

```
$ crontab -l
02 05 * * * /home/user/disklog.sh
```

The `crontab -l` lists the existing entries in the cron table for the current user.

We can also view the cron table for other users by specifying a username with the `-u` option as follows:

```
$ crontab -l -u slynux
09 10 * * * /home/slynux/test.sh
```

You should run as root when you use the `-u` option to gain higher privilege.

Removing the cron table

We can remove the cron table for the current user using the `-r` option:

```
$ crontab -r
```

In order to remove `crontab` for another user, use:

```
# crontab -u slynux -r
```

Run as root to get higher privilege.

Writing and reading the MySQL database from Bash

MySQL is a widely used database management system used to manage databases for the storage systems for applications that are written in languages, such as PHP, Python, C++, and so on. Accessing and manipulating MySQL databases from shell scripts is also interesting, as we can write scripts to store contents from a text file or **Comma Separated Values** (**CSV**) into tables and interact with the MySQL database to read and manipulate data. For example, we can read all the e-mail addresses stored in a guestbook program's database by running a query from the shell script. In this recipe, we will see how to read and write to the MySQL database from Bash. Let's take this example problem:

I have a CSV file containing details of students. I need to insert the contents of the file to a database table. From this data, I need to generate a separate rank list for each department.

Getting ready

In order to handle MySQL databases, you should have `mysql-server` and `mysql-client` packages installed on your system. These tools do not come with a Linux distribution by default. As MySQL comes with a username and password for authentication, you should also set a username and password while installing the MySQL server.

How to do it...

The preceding problem can be solved using Bash utilities `sort`, `awk`, and so on. Alternately, we can also solve it by using an SQL database table. We will write three scripts for the purpose of creating a database and table, inserting student data into the table, and reading and displaying processed data from the table.

Create the database and table script as follows:

```
#!/bin/bash
#Filename: create_db.sh
#Description: Create MySQL database and table

USER="user"
PASS="user"

mysql -u $USER -p$PASS <<EOF 2> /dev/null
CREATE DATABASE students;
EOF

[ $? -eq 0 ] && echo Created DB || echo DB already exist
```