Parameter	Description
`%Z`	This shows the system's page size in bytes. This is a per-system constant, but varies between systems.
`%P`	This shows the percentage of the CPU that this job got. This is just user + system times divided by the total running time. It also prints a percentage sign.
`%K`	This shows the average total (data + stack + text) memory usage of the process, in Kilobytes.
`%w`	This shows the number of times that the program was context-switched voluntarily, for instance, while waiting for an I/O operation to complete.
`%c`	This shows the number of times the process was context-switched involuntarily (because the time slice expired).

Collecting information about logged in users, boot logs, and boot failures

Linux supports commands to report aspects of the runtime system including logged in users, how long the computer has been powered on, and boot failures. This data is used to allocate resources and diagnose problems.

Getting ready

This recipe introduces the who, w, users, uptime, last, and lastb commands.

How to do it...

1. The `who` command reports information about the current users:

```
$ who
slynux   pts/0   2010-09-29 05:24 (slynuxs-macbook-pro.local)
slynux   tty7    2010-09-29 07:08 (:0)
```

   This output lists the login name, the TTY used by the users, login time, and remote hostname (or X display information) about logged in users.

TTY (the term comes from **TeleTYpewriter**) is the device file associated with a text terminal that is created in `/dev` when a terminal is newly spawned by the user (for example, `/dev/pts/3`). The device path for the current terminal can be found out by executing the `tty` command.

2. The `w` command provides more detailed information:

```
$ w
07:09:05 up  1:45,  2 users,  load average: 0.12, 0.06, 0.02
USER     TTY      FROM    LOGIN@   IDLE  JCPU PCPU WHAT
slynux   pts/0    slynuxs 05:24  0.00s  0.65s 0.11s sshd: slynux
slynux   tty7     :0      07:08  1:45m  3.28s 0.26s bash
```

This first line lists the current time, system uptime, number of users currently logged on, and the system load averages for the past 1, 5, and 15 minutes. Following this, the details about each login session are displayed with each line containing the login name, the TTY name, the remote host, login time, idle time, total CPU time used by the user since login, CPU time of the currently running process, and the command line of their current process.

Load average in the `uptime` command's output indicates system load. This is explained in more detail in `Chapter 10`, *Administration Calls*.

3. The users command lists only the name of logged-in users:

```
$ users
slynux slynux slynux hacker
```

If a user has multiple sessions open, either by logging in remotely several times or opening several terminal windows, there will be an entry for each session. In the preceding output, the `slynux` user has opened three terminals sessions. The easiest way to print unique users is to filter the output through `sort` and `uniq`:

```
$ users | tr ' ' '\n' | sort | uniq
slynux
hacker
```

The `tr` command replaces each `' '` character with `'\n'`. Then a combination of `sort` and `uniq` reduces the list to a unique entry for each user.

4. The `uptime` command reports how long the system has been powered on:

```
$ uptime
21:44:33 up 6 days, 11:53, 8 users, load average: 0.09, 0.14,
0.09
```

   The time that follows the `up` word is how long the system has been powered on. We can write a one-liner to extract the uptime only:

```
$ uptime | sed 's/.*up \(.*\),.*users.*/\1/'
```

   This uses `sed` to replace the line of output with only the string between the word up and the comma before users.

5. The `last` command provides a list of users who have logged onto the system since the `/var/log/wtmp` file was created. This may go back a year or more:

```
$ last
aku1  pts/3   10.2.1.3   Tue May 16 08:23 - 16:14  (07:51)
cfly  pts/0   cflynt.com Tue May 16 07:49   still logged in
dgpx  pts/0   10.0.0.5   Tue May 16 06:19 - 06:27  (00:07)
stvl  pts/0   10.2.1.4   Mon May 15 18:38 - 19:07  (00:29)
```

   The `last` command reports who logged in, what `tty` they were assigned, where they logged in from (IP address or local terminal), the login, logout, and session time. Reboots are marked as a login by a pseudo-user named `reboot`.

6. The `last` command allows you to define a user to get only information about that user:

```
$ last USER
```

7. USER can be a real user or the pseudo-user `reboot`:

```
$ last reboot
reboot   system boot  2.6.32-21-generi Tue Sep 28 18:10 - 21:48
(03:37)
reboot   system boot  2.6.32-21-generi Tue Sep 28 05:14 - 21:48
(16:33)
```

8. The `lastb` command will give you a list of the failed login attempts:

```
# lastb
test     tty8         :0               Wed Dec 15 03:56 - 03:56
(00:00)
slynux   tty8         :0               Wed Dec 15 03:55 - 03:55
(00:00)
```

The `lastb` command must be run as the root user.

Both `last` and `lastb` report the contents of `/var/log/wtmp`. The default is to report month, day, and time of the event. However, there may be multiple years of data in that file, and the month/day can be confusing.

The `-F` flag will report the full date:

```
# lastb -F
hacker   tty0       1.2.3.4          Sat Jan 7 11:50:53 2017 -
Sat Jan 7 11:50:53 2017 (00:00)
```

Listing the top ten CPU– consuming processes in an hour

The CPU is another resource that can be exhausted by a misbehaving process. Linux supports commands to identify and control the processes hogging the CPU.

Getting ready

The `ps` command displays details about the processes running on the system. It reports details such as CPU usage, running commands, memory usage, and process status. The `ps` command can be used in a script to identify who consumed the most CPU resource over an hour. For more details on the `ps` command, refer to `Chapter 10`, *Administration Calls*.