How to use the last command in Linux

Want to know who, what time and which device to access your Linux computer? Please read the following article.

Want to know who, what time and which device to access your Linux computer? Please read the following article.

1. Basic Linux commands everyone needs to know
2. How to take a screenshot of the login screen in Linux
3. How to remotely control Linux using a Windows computer

Wtmp file

Linux and other Unix-like operating systems such as MacOS manage very well in login. On the system, you can find a log of everything related to login, logout on the computer. This log file is called wtmp. W stands for When (when) or who (who). The tmp section may be short for temporary but may also be short for timestamp (timestamp).

We only need to know wtmp is a log that records all logon and logoff events on the computer. Viewing data in the wtmp log is the basic step in security. For a regular home computer, security is not an important issue, but it is also interesting to review the use of your computer.

Unlike other text-based log files in Linux, wtmp is a binary file. To access data in this file, you need to use the last command.

1. Basic file system in Unix / Linux

Last order

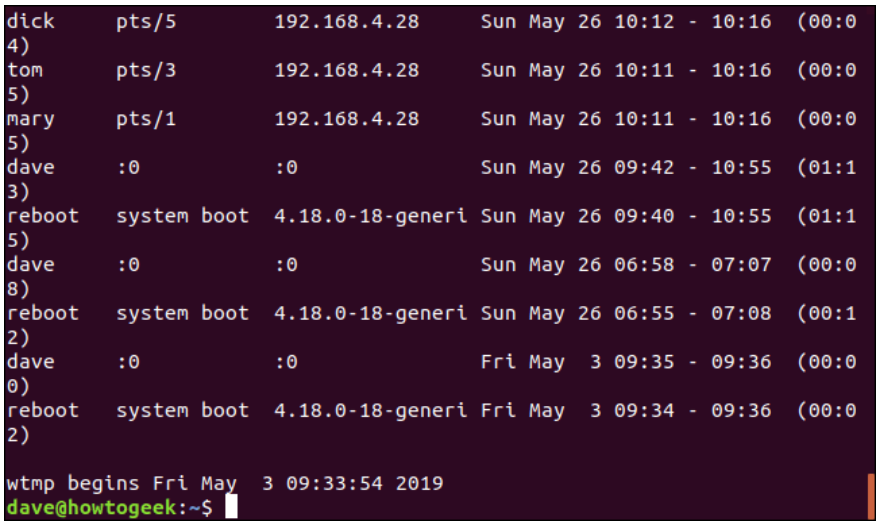
The last command reads data from the wtmp record and displays it in the terminal window. If typing last and pressing **Enter** , it will display all the records from the log file.

```
last
```

Each record from wtmp is displayed in the terminal window.

From left to right, each line contains:

1. Login username.
2. The device they logged in. Device item: 0 means login on the Linux computer itself.
3. IP address of login machine.
4. Login time and date stamp.
5. Time of the session



The last line tells us the date and time of the earliest recorded session in the log.

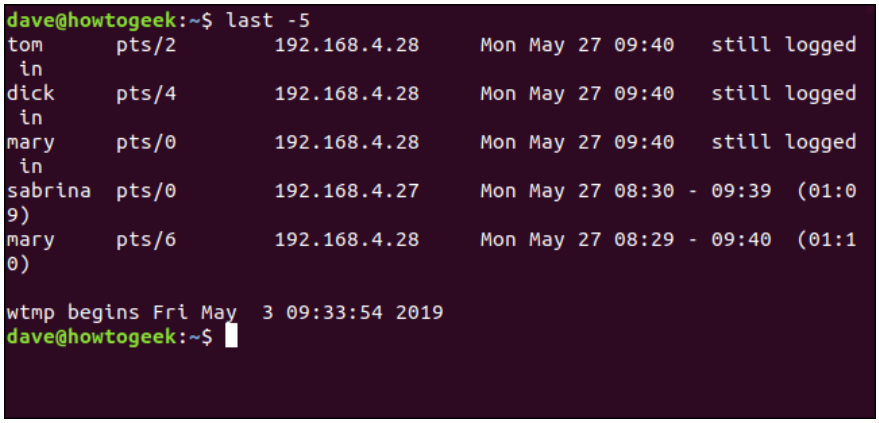
The login for fictitious user "reboots" is entered into the log every time the computer is started. The device field is replaced with the kernel version. The duration of the login session for these items represents the computer's uptime.

Show some specific lines

Using only the last command will display all the above information. If you want specific information, you can ask for last to provide the specific line number of the output. For example, if you want to see 5 lines, you need to type **-5** with the last command.

```
last -5
```

The above command will show the first five lines of the log, this is the most recent data.

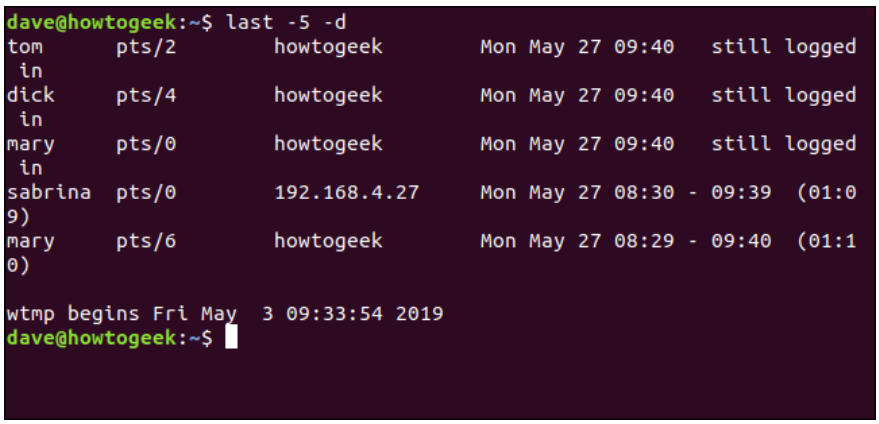


Display network name for remote users

The **-d** (Domain Name System) option requires last resolving the remote user's IP address into a host name or network name.

```
last -d
```

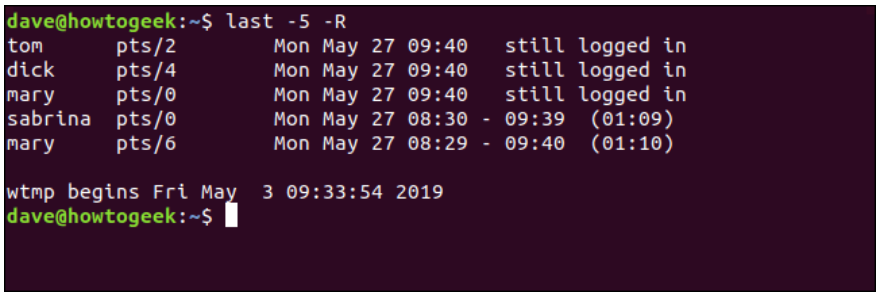
However, it is not always possible to convert IP addresses to network names, but the command will execute when possible.



Hide IP address and network name

If you are not interested in the IP address or network name, use the **-R** option (without the server name) to block this field.

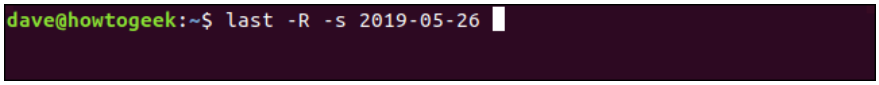
This option will give you a more neat output, so it will be used in the examples below. If using last to determine unusual activity, you should not block this field.



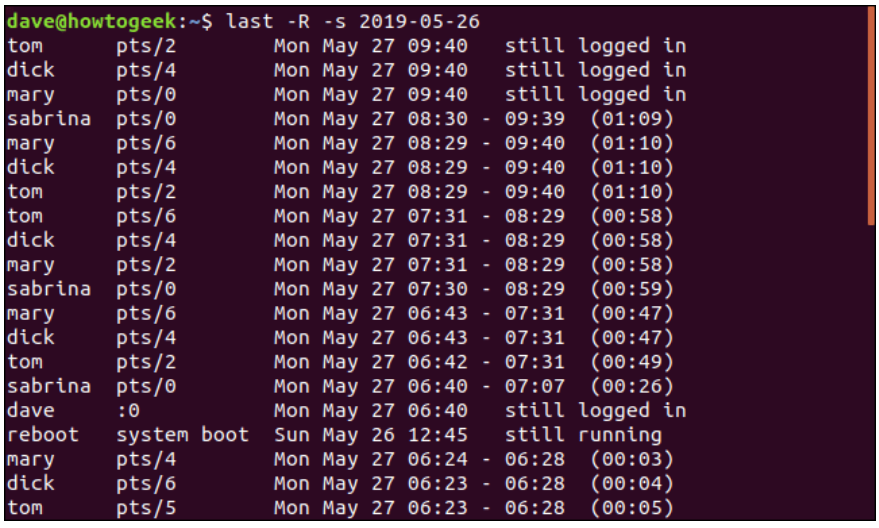
Select the record by date

You can use the **-s** option (since) to restrict the output to only show log events that take place since a particular date.

If you only want to see the login events taking place from 26/5/2019, you will use the following command:

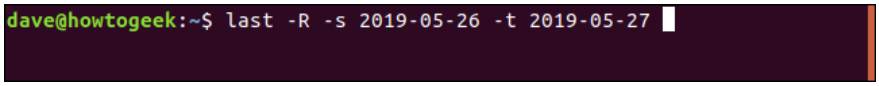


The output that displays the log with log events takes place from 00:00 on the day specified to the latest record on the log file.

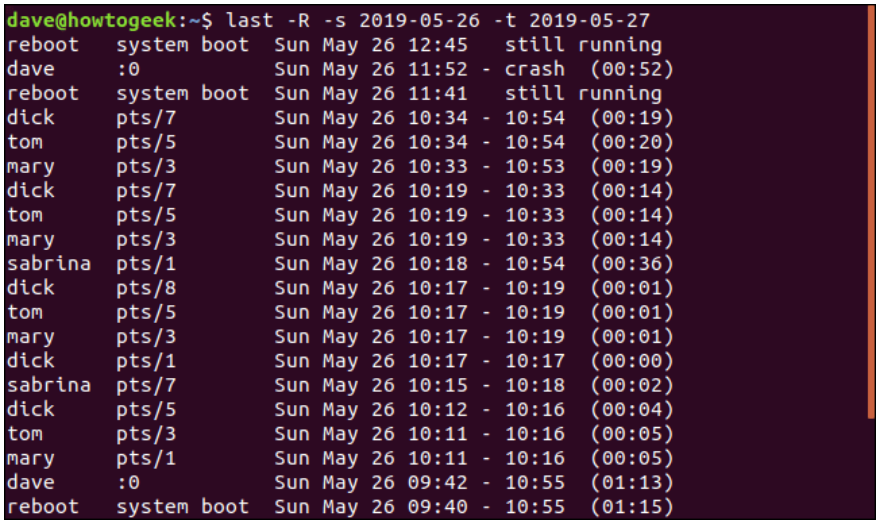


Search within a specific time period

You can use **-t** (until) to specify an end date. This allows you to select a log file set that takes place at a specific time.



This command requires **last** retrieval and displays log log from 00:00 (dawn) day 26 to 00:00 (dawn) day 27. It limits the login session to take place only on the 26th



Format time and date

You can use time as well as dates with **-s** and **-t options.** Different time formats can be used with **last** options for dates and times:

1. YYYYMMDDhhmmss
2. YYYY-MM-DD hh: mm: ss
3. YYYY-MM-DD hh: mm - seconds are set to 00
4. YYYY-MM-DD - time is set to 00:00:00
5. hh: mm: ss - date is set to today
6. hh: mm - date will be set to today, seconds to 00
7. now
8. yesterday - time is set to 00:00:00
9. today - time is set to 00:00:00
10. tomorrow - time is set to 00:00:00
11. + 5min
12. -5days

These commands are tested on Ubuntu, Fedora and Manjaro distributions. These are derivatives of the Debian, RedHat and Arch distributions respectively.

```
last -R -s 2019-05-26 11:00 -t 2019-05-27 13:00
```

```
dave@howtogeek:~$ last -R -s 2019-05-26 11:00 -t 2019-05-27 13:00

wtmp begins Fri May  3 09:33:54 2019
dave@howtogeek:~$
```

As you can see the above command does not return any records. Use the first date and time format from the list as the previous command returns records:

```
last -R -s 20190526110000 -t 20190527130000
```

```
dave@howtogeek:~$ last -R -s 20190526110000 -t 20190526130000
reboot   system boot  Sun May 26 12:45   still running
dave     :0           Sun May 26 11:52 - crash  (00:52)
reboot   system boot  Sun May 26 11:41   still running

wtmp begins Fri May  3 09:33:54 2019
dave@howtogeek:~$
```

Search in relative units

You also specify the time interval in minutes or days, relative to the current date and time. The following command we require a record from the previous 2 days until the previous day.

```
last -R -s -2days -t -1days
```

```
dave@howtogeek:~$ last -R -s -2days -t -1days
dave     :0           Sun May 26 09:42    gone - no logout
reboot   system boot  Sun May 26 09:40   still running
dave     :0           Sun May 26 06:58 - 07:07  (00:08)
reboot   system boot  Sun May 26 06:55 - 07:08  (00:12)

wtmp begins Fri May  3 09:33:54 2019
dave@howtogeek:~$
```

Yesterday, today and now

You can use yesterday and tomorrow to abbreviate for yesterday's date and today's date.

```
last -R -s yesterday -t today
```

```
dave@howtogeek:~$ last -R -s yesterday  -t today
```

Orders require records from the start date to the end date. It does not include records for the end date.

```
dave@howtogeek:~$ last -R -s yesterday -t today
reboot   system boot  Sun May 26 12:45   still running
dave     :0           Sun May 26 11:52 - crash  (00:52)
reboot   system boot  Sun May 26 11:41   still running
dick     pts/7        Sun May 26 10:34 - 10:54  (00:19)
tom      pts/5        Sun May 26 10:34 - 10:54  (00:20)
mary     pts/3        Sun May 26 10:33 - 10:53  (00:19)
dick     pts/7        Sun May 26 10:19 - 10:33  (00:14)
tom      pts/5        Sun May 26 10:19 - 10:33  (00:14)
mary     pts/3        Sun May 26 10:19 - 10:33  (00:14)
sabrina  pts/1        Sun May 26 10:18 - 10:54  (00:36)
dick     pts/8        Sun May 26 10:17 - 10:19  (00:01)
tom      pts/5        Sun May 26 10:17 - 10:19  (00:01)
mary     pts/3        Sun May 26 10:17 - 10:19  (00:01)
dick     pts/1        Sun May 26 10:17 - 10:17  (00:00)
sabrina  pts/7        Sun May 26 10:15 - 10:18  (00:02)
dick     pts/5        Sun May 26 10:12 - 10:16  (00:04)
tom      pts/3        Sun May 26 10:11 - 10:16  (00:05)
mary     pts/1        Sun May 26 10:11 - 10:16  (00:05)
dave     :0           Sun May 26 09:42 - 10:55  (01:13)
reboot   system boot  Sun May 26 09:40 - 10:55  (01:15)
```

Now option is an abbreviation for 'today at the current time'. To see a login event that takes place from 00:00 (dawn) until the time when you use this command:

```
last -R -s today -t now
```

```
dave@howtogeek:~$ last -R -s today -t now
```

The above command will display all current login events, including those that are still logged in.

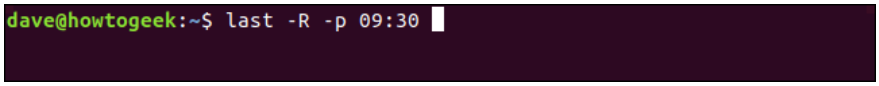
```
dave@howtogeek:~$ last -R -s today -t now
tom      pts/2        Mon May 27 09:40   still logged in
dick     pts/4        Mon May 27 09:40   still logged in
mary     pts/0        Mon May 27 09:40   still logged in
sabrina  pts/0        Mon May 27 08:30 - 09:39  (01:09)
mary     pts/6        Mon May 27 08:29 - 09:40  (01:10)
dick     pts/4        Mon May 27 08:29 - 09:40  (01:10)
tom      pts/2        Mon May 27 08:29 - 09:40  (01:10)
tom      pts/6        Mon May 27 07:31 - 08:29  (00:58)
dick     pts/4        Mon May 27 07:31 - 08:29  (00:58)
mary     pts/2        Mon May 27 07:31 - 08:29  (00:58)
sabrina  pts/0        Mon May 27 07:30 - 08:29  (00:59)
mary     pts/6        Mon May 27 06:43 - 07:31  (00:47)
dick     pts/4        Mon May 27 06:43 - 07:31  (00:47)
tom      pts/2        Mon May 27 06:42 - 07:31  (00:49)
sabrina  pts/0        Mon May 27 06:40 - 07:07  (00:26)
dave     :0           Mon May 27 06:40   still logged in
mary     pts/4        Mon May 27 06:24 - 06:28  (00:03)
dick     pts/6        Mon May 27 06:23 - 06:28  (00:04)
tom      pts/5        Mon May 27 06:23 - 06:28  (00:05)
mary     pts/4        Mon May 27 06:23 - 06:24  (00:01)
```

-P option

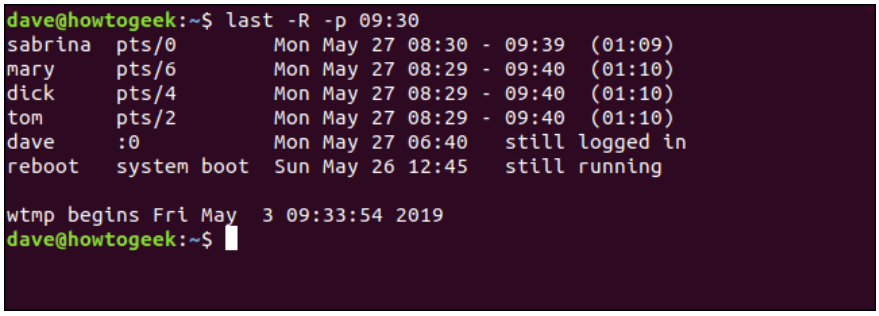
The **-p** (current) option allows you to find out who has logged in at a point in a specific time. If someone signs in to the computer at the time you specify, they will be listed.

If you specify a time without a date, the last command will default to you today (ie the date of using this command).

```
last -R -p 9:30
```



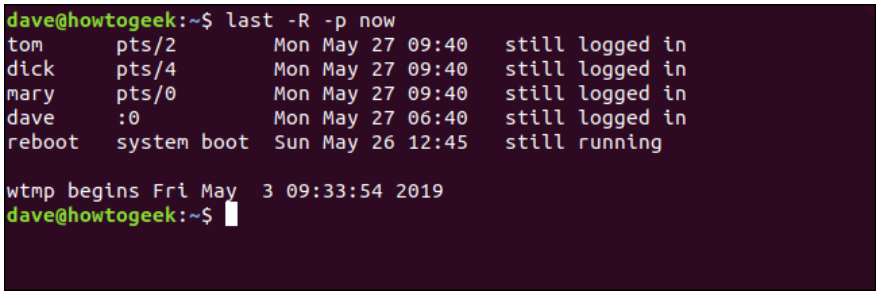
People still logged in without logout time; They are described as **still logged in** . If the computer has not been restarted since the time you specified it, it will be listed as **still running** .



If you use **now** with the **-p** option, you can detect who is logged in at the time you use the command.

```
last -R -p now
```

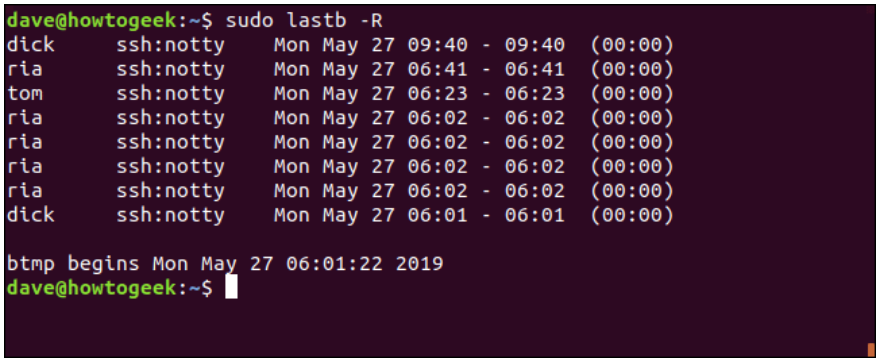
This is a slightly longer way when using the who command.



Lastb command

The lastb command reads data from the record called btmp. Lastb lists the failed login times. You can also use the options of last with this command. Because login failed, the time will be 00:00.

You must use **sudo** with **lastb** .



Knowing who has logged in to your Linux computer is very useful and incorporating information about unsuccessful logins will help you take the first steps in investigating computer intrusions. doubt.

You finished reading the article "**How to use the last command in Linux**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

© 2019 TipsMake.com