Time conversion using Bash

Article Number: 54 | Rating: Unrated | Last Updated: Fri, May 25, 2018 1:02 PM

Time conversion using Bash

This article show how you can obtain the UNIX epoch time (number of seconds since 1970-01-01 00:00:00 UTC) using the Linux bash "date" command. It also shows how you can convert a UNIX epoch time to a human readable time.

Obtain UNIX epoch time using bash

Obtaining the UNIX epoch time using bash is easy. Use the build-in date command and instruct it to output the number of seconds since 1970-01-01 00:00:00 UTC. You can do this by passing a format string as parameter to the date command. The format string for UNIX epoch time is '%s'.

```
lode@srv-debian6:~$ date "+%s"
1234567890
```

To convert a specific date and time into UNIX epoch time, use the -d parameter. The next example shows how to convert the timestamp "February 20th, 2013 at 08:41:15" into UNIX epoch time.

```
lode@srv-debian6:~$ date "+%s" -d "02/20/2013 08:41:15"
1361346075
```

Converting UNIX epoch time to human readable time

Even though I didn't find it in the date manual, it is possible to use the date command to reformat a UNIX epoch time into a human readable time. The syntax is the following:

```
lode@srv-debian6:~$ date -d @1234567890
Sat Feb 14 00:31:30 CET 2009
```

The same thing can also be achieved using a bit of perl programming:

```
lode@srv-debian6:~$ perl -e 'print scalar(localtime(1234567890)), "n"'
Sat Feb 14 00:31:30 2009
```

Please note that the printed time is formatted in the timezone in which your Linux system is configured. My system is configured in UTC+2, you can get another output for the same command.

```
date -d @1234567890
```

```
echo 1234567890 | awk '{print strftime("%c",$1)}'
```

Posted - Fri, May 25, 2018 1:02 PM. This article has been viewed 2827 times.

Online URL: http://kb.ictbanking.net/article.php?id=54