How to do it...

Let's go through the script to find out active users and generate the report:

```
#!/bin/bash
#Filename: active_users.sh
#Description: Reporting tool to find out active users

log=/var/log/wtmp

if [[ -n $1 ]];
then
  log=$1
fi

printf "%-4s %-10s %-10s %-6s %-8s\n" "Rank" "User" "Start" "Logins"
"Usage hours"

last -f $log | head -n -2   > /tmp/ulog.$$

cat /tmp/ulog.$$ |  cut -d' ' -f1 | sort | uniq> /tmp/users.$$

(
while read user;
do
  grep ^$user /tmp/ulog.$$ > /tmp/user.$$
  minutes=0

  while read t
  do
    s=$(echo $t | awk -F: '{ print ($1 * 60) + $2 }')
    let minutes=minutes+s
  done< <(cat /tmp/user.$$ | awk '{ print $NF }' | tr -d ')(')

  firstlog=$(tail -n 1 /tmp/user.$$ | awk '{ print $5,$6 }')
  nlogins=$(cat /tmp/user.$$ | wc -l)
  hours=$(echo "$minutes / 60.0" | bc)

  printf "%-10s %-10s %-6s %-8s\n"  $user "$firstlog" $nlogins $hours
done< /tmp/users.$$

) | sort -nrk 4 | awk '{ printf("%-4s %s\n", NR, $0) }'
rm /tmp/users.$$ /tmp/user.$$ /tmp/ulog.$$
```

A sample output is as follows:

```
$ ./active_users.sh
Rank User        Start      Logins Usage hours
1    easyibaa    Dec 11     531    349
2    demoproj    Dec 10     350    230
3    kjayaram    Dec 9      213    55
4    cinenews    Dec 11     85     139
5    thebenga    Dec 10     54     35
6    gateway2    Dec 11     52     34
7    soft132     Dec 12     49     25
8    sarathla    Nov 1      45     29
9    gtsminis    Dec 11     41     26
10   agentcde    Dec 13     39     32
```

How it works...

In the `active_users.sh` script, we can either provide the `wtmp` logfile as a command-line argument or it will use the default `wtmp` log file. The `last -f` command is used to print the logfile contents. The first column in the logfile is the username. By using `cut`, we extract the first column from the logfile. Then, the unique users are found out by using the `sort` and `uniq` commands. Now for each user, the log lines corresponding to their login sessions are found out using `grep` and are written to a temporary file. The last column in the last log is the duration for which the user logged in to the session. Hence, in order to find out the total usage hours for a user, the session duration is to be added. The usage duration is in (`HOUR:SEC`) format and it is converted into minutes using a simple awk script.

In order to extract the session hours for the users, we have used the `awk` command. For removing the parenthesis, `tr -d` is used. The list of the usage hour string is passed to the standard input for the `while` loop using the `< ( COMMANDS )` operator, which acts as a file input. Each hour string is converted into seconds by using the `date` command and added to the variable `seconds`. The first login time for a user is in the last line and it is extracted. The number of login attempts is the number of log lines. In order to calculate the rank of each user according to the total usage hours, the data record is to be sorted in the descending order with usage hours as the key. For specifying the number reverse sort, the `-nr` option is used along with `sort` command. `-k4` is used to specify the key column (usage hour). Finally, the output of the sort is passed to `awk`. The `awk` command prefixes a line number to each of the lines, which becomes the rank for each user.