Ruby one-liner to count logins in Linux last command

count logins in last command

If you want to take the output of the Linux `last` command, and get a count of how many times each user connected to your server, this script will do.

sample input

```
tommytrojan      sshd         abc.def Fri May 15 13:30 - 13:30  (00:00)joebruin      sshd
```

sample output

```
Total entries: 4joebruin = 2janebruin = 1tommytrojan = 1
```

script and explanation

```
last | ruby -ane 'BEGIN{rows=0; a={}}; next if $F[0].nil?; a[$F[0]]||=0;a[$F[0]]+= 1; rows += 1;
END{puts "Total entries: #{rows}" ; a.sort_by{|k,v|v}.reverse.each{|k,v| puts "#{k} = #{v}"} }'
```

- ruby -e executes the following script; -n loops; and -a splits input and outputs into $F array
- BEGIN block is necessary to initialize row count and a hash, where we'll store counts
- next if $F0.nil?; is necessary, because at least one of our input lines had no user field.
- $F0 gives us first field in input line.
- a[$F0]||=0; initializes hash location to 0 if it hasn't been used yet
- a[$F0]+= 1; increments user access count in hash
- END block lets us print our report.
- a.sort_by{|k,v|v}does a sort by the counts, not by the user names. a.sort would have been enough if we just wanted it sorted by the key, ie: usernames.
- reverse.each{|k,v| puts "#{k} = #{v}"} reverses the sort to get descending order, and outputs the key (k) and value(v), e.g. username = access count.

useful links

- http://benoithamelin.tumblr.com/ruby1line
- http://www.tutorialspoint.com/ruby/ruby_predefined_variables.htm

NOTE: My colleague did the same thing in this shell script. And his script includes the line with the missing user field.

```
last | cut -d" " -f1 | sort | uniq -c | awk '{print $1" " $2}' | sort -nr
```

Revision #3
Created 2015-05-15 21:05:31 UTC by Franks, Mike
Updated 2015-05-15 21:15:03 UTC by Franks, Mike