

Received: 02/01/2016

Completed: 02/13/2016

28/30

CS 3423.01

Assignment 3 : Awk Exploration

Due 12:00 Midnight Monday, February 15, 2016

News Awk Script

You have set up a news server and your feed has just sent you his active file. You need to perform an addgroup on each of three thousand different newsgroups. Rather than typing and executing the commands on an individual basis you decide to write a awk function to create from the active file a file consisting of the necessary addgroup commands. You then plan to execute the file containing the addgroups.

1. The active file is /usr/local/courses/cs3423/assign3/active. In order to save yourself some typing, create a symbolic link in your assign3 directory with the command:
`ln -s /usr/local/courses/cs3423/assign3/active active`
2. The lines in the active file has the following format:
`comp.os.linux.announce 0000002587 02190 m
comp.arch 0000028934 28874 y
utsa.cs.2413 0000000004 0000000001 y`
3. Each line in the active file should be converted to an addgroup command. For example, the previous lines should generate the command:
`addgroup comp.os.linux.announce m
addgroup comp.arch y
addgroup utsa.cs.2413 y`
4. The awk script **MUST** be in a file called news.awk in your assign3 file and you should be able to run the conversion with the command:
`awk -f news.awk active > news.commands`

News Usage I

Your news server is now up and running and you wish to see a summary of the amount and type of news traffic being fed to your news server. In particular, for each feed site, you want to see how many articles were accepted, how many articles were canceled and how many were rejected.

1. Make the following symbolic links in your assign3 directory using the following commands:
`ln -s /usr/local/courses/cs3423/assign3/news news
ln -s /usr/local/courses/cs3423/assign3/news.notice news.notice`
2. The news distribution daemon, innd, will write one line for each article in the news file. This is a text file with a variable number of space-separated fields in one of the following formats:

mon dd hh:mm:ss.mmm + feed <Message-ID> site...
mon dd hh:mm:ss.mmm j feed <Message-ID> site...
mon dd hh:mm:ss.mmm c feed <Message-ID> site...
mon dd hh:mm:ss.mmm - feed <Message-ID> reason...

The first three fields are the date and time to millisecond resolution. The fifth field is the site that sent the article (based on the Path header) and the sixth field is the article's Message-ID; there will be a question mark if the information is not available.

The fourth field indicates whether the article was accepted or not. If it is a plus sign, then the article was accepted. If it is the letter "j" then the article was accepted, but all of newsgroups have an "j" in their active field, so the article was filed into the "junk" newsgroup. If the fourth field is the letter "c", then a cancel message was accepted before the original article arrived. In all three cases, the article has been accepted and the "site..." field contains the space-separated list of sites to which the article is being sent.

If the fourth field is a minus sign, then the article was rejected.

3. Ringer had the following feeds:

swrinde

news.ca.is.net

?

Feed [\\$5] = |
gets unique ss in file

swrinde

news.ca.is.net

not-formail

ringer.cs.utexas.edu

4. Calculate the following information for each feed:

(a) The number of articles accepted from each site.

(b) The number of rejected articles from each site. Same as #3

(c) The number of canceled articles from each site.

5. The date and time of the first entry as well as the last entry should also be printed. The news file represents one day of traffic. $Ne == 1$ Just print at END

StartTime = \$1 \$2 \$3

6. Because the news file is quite large, you may want to use sed to create a smaller file of approximately 1000 lines on which to test your program.

7. The awk script file should be named **newsfeed.awk**.

8. The output should be formatted as closely as possible to the following:

Incoming News Feed Summary

	accepted	rejected	canceled
swrinde:	94755	19683	0
news.cais.net:	14374	240	0
?:	3	0	3

Start Time = Feb 13 21:28:38.409

End Time = Feb 14 20:56:49.066

News Usage II

Now that your users have been reading news for a while, you wish to see a summary of the news usage on the various machines on the campus. In particular you want to see how many articles have been read on lonestar, runner, ringer and the rings over a period of approximately one day. In addition you are curious about the usage of the cs2413 groups, utsa.cs.2413 and utsa.cs.2413.d and want to print out the total number of articles from each newsgroup which were read on each machine.

- When a news session is established, used and finally closed, lines similar to the following lines are written to the news.notice file.

Feb 14 00:18:29 ringer nnrpd[16952]: runner.jpl.utsa.edu connect
Feb 14 00:33:35 ringer nnrpd[16952]: runner.jpl.utsa.edu group utsa.cs.1723-2.d 9
Feb 14 00:33:58 ringer nnrpd[16952]: runner.jpl.utsa.edu group utsa.cs.2413.d 6
Feb 14 00:33:58 ringer nnrpd[16952]: runner.jpl.utsa.edu exit articles 32 groups 9
Feb 14 00:33:58 ringer nnrpd[16952]: runner.jpl.utsa.edu times user 1.060 system 1.060 elapsed 930.059

Notice that field 7 is the key field. If it is "group" then field 8 gives the newsgroup name and field 9 specifies the number of articles read from that group. If it is "exit" then field 9 gives the total number of articles read in that session while field 11 specifies the number of groups read. If field 7 is "times" then the user, system and elapsed times for that session are given.

- Consider the following systems:

lonestar.jpl.utsa.edu

runner.jpl.utsa.edu

ringer.cs.utsa.edu

ring???.cs.utsa.edu Any of the ring machines (ring01 - ring47) not including ringer.

- For each of the above systems calculate

- The number of articles read.
- The number of groups read.
- The number of articles in utsa.cs.2413 read.
- The number of articles in utsa.cs.2413.d read.

- (e) The total elapsed time for all news sessions.
4. The date and time of the first entry as well as the last entry should also be printed.
 5. The awk script file should be named **newsread.awk**.
 6. The output should be formatted as closely as possible to the following:

News Reader Summary

	lonestar	runner	ringer	rings
Articles:	72256	12533	1	4521
Groups:	5282	8344	19	1068
Cs2413:	0	0	0	12
Cs2413.d:	40	25	0	274
User Time:	266197	83377.2	128.554	98294.8

Start Time = Feb 13 21:27:14

End Time = Feb 14 20:56:49

Submission

1. Submit your assignment:

`submit assign3 project3 3423`

2. It is your responsibility to be certain that you turned in the correct assignment and that the assignment was correctly turned in. Please just submit the three awk script files and be certain to delete the active, news and news.notice links. In order to verify what programs you actually turned in type:

`verify -c cs3423 -p project3`

3. Remember, the assignment **must** be turned in by the due time. The assignment will automatically be turned off at that time. If your project is late, then turn it in as **project3-late**. This will only extend for an additional week and there will be a penalty associated with a late project.