Assignment 1 - Linux/Unix shell fundamentals

Write a single-lined command per question to produce the information listed below (including extra credit). Single-lined command defined as follows:

- No semicolons or multiline commands (except within command arguments).
- Pipes and redirects are allowed.
- Command and process substitutions allowed, e.g. **touch $(cat files.txt)**
- You may read and write to auxiliary files in **/tmp/**

Commands allowed (you may use any options):

- sed
- cut
- paste
- join
- tr
- tee
- xargs
- sort
- uniq
- wc
- split
- fmt
- fold
- echo
- printf
- cp
- mv
- rm
- pwd
- ls
- touch
- cat
- head
- tail
- grep
- find
- bc
- date
- watch
- kill

Part 1:
The following scenario applies to questions a, b, and c.
Assume a program named **ecs198flog** exists on your system and is contained in your **$PATH**. Running this program will generate a onelined log message, print to stdout, and terminate. This program generates a new log message every one second.

a. Generate the log file **ecs198flogs** in the **/var/log** directory if and only if it does not already exist. [5%]
b. Append the output of **ecs198flog** continuously to the above generated file. If this command is not terminated, it should run for eternity, assuming enough disk space. [5%]
c. Search the log file for any lines containing **ethan** or **akhil** but not **jason**, then return the number of unique lines found printed to stdout. [20%]

Part 2:

d. Given the output to the command **ps aux** (note the BSD style) on a Linux/Unix system, return the top 20% (rounded down) of lines with a percentage CPU less than 2% and percentage RAM less than 1%. This should be displayed to the terminal via stdout. [40%]
e. Using the command/programming language AWK, rewrite the solution to the above problem (question d). You may only use the command awk, i.e. your solution should start with **ps aux | awk** … [30%]

Extra credit:

f. Implement the command **pkill firefox** with the list of commands above. You should use the output of **ps aux** and you may use the **awk** command. Note this line in the pkill manual page: a running pgrep or pkill process will never consider itself as a potential match. [5%]