CSCI 1523
Spring 2015
Programming Project 3: Systems Administration
Due: April 17, 2016, 11:59P

This programming project contains 5 pages (including this cover page) and includes 2 programming problems.

Before starting to work on this programming project please read it carefully making sure that all of the examples and figures are present and that you have the correct number of pages in the assignment.

In our course we encourage collaboration on programming problems however exchanging SOURCE CODE is expressly prohibited.

To enable your collaboration we have established on the course site a discussion forum for this particular assignment. Students are encouraged to post questions concerning the documentation, logic, and coding of the assignment on this site. however as mentioned sharing source code is strictly prohibited.

This assignment is being distributed in a .zip archive format. All files have been created for you. You only need to unzip the archive after downloading, complete the work within the directory (folder), then when you are ready to submit this assignment, compress the directory (folder) into .zip format and upload it to the Dropbox for the assignment.

The datafiles, *passwd.dat* and *logindata.dat* are also included in this archive. DO NOT DELETE THEM when submitting your completed assignment.

Failure to follow the correct submission process will be treated as an uncompleted assignment until the submission is done in the proper manner.

1. **Processing system log files**
   Networking professionals and systems administrators are frequently asked to develop statistics from system log files and system configuration files which may give IT management some idea of the cause of intermittent system reliability issues. Additionally such statistics can be used in the development of forensic or security information on the system under study.

   In this exercise we are going to analyze system files taken from a Solaris Unix operating system. The Solaris system from which this data was extracted runs the thinclient server in the Computer Science Department at Saint Paul College. We will look at two files.

   The first of these files is the Unix users file, */etc/passwd*. Figure 1 below illustrates the format of the */etc/passwd*. As can be seen the fields it contains are delimited by the ":" character.

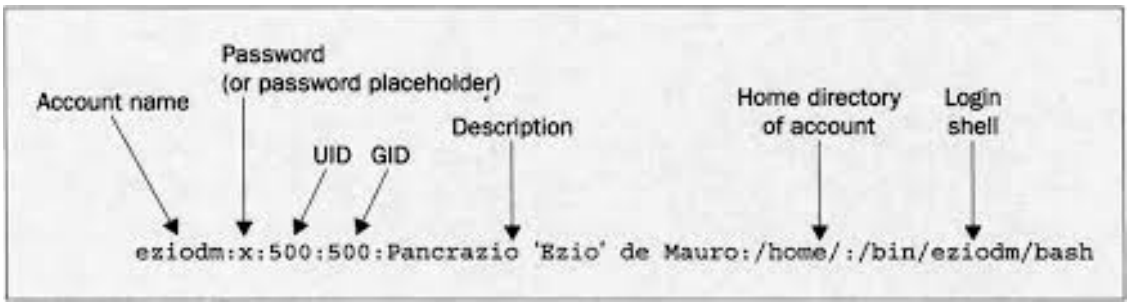

Figure 1: Format of a typical /etc/passwd file.

An excellent discussion of the */etc/passwd* file can be found at: `http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/`.

The second of these files, *logindata.dat*, is a text dump of the system log file which gathers information on the date and time of user logins as well as other information such as reboot and system start up. These records happen to have different data and therefore different formats. Depending on record type, the record length may be well over 100 characters. Our task is simplified by the fact that in each type of record the fields are delimited by a **space** character.

Required:

(a) Analyze passwd file.

   i. Prepare a Python script which reads the */etc/passwd* file which is included in .zip archive which accompanies this assignment assignment. The */etc/passwd* file has been renamed, *passwd.dat*, for the purposes of this assignment. From this file extract the following data and print it to the screen:

      1. The total number of users identified in the file.
      2. The number of users which belong to the systems administrator group, Group Id = 10.

3. The various types of shells specified for the users of the system and a count of users who are configured to use that particular shell type, (bash, sh, ksh). Also keep count of the number of users who do not have a shell prompt specified.

The Python script for this part of the exercise should be submitted in a file:

Csci1523Assignment3A.py

(b) Analyze *logindata.dat* file.

i. Login and system information are kept on all large computer systems. For this particular problem we have extracted login and reboot information for the Computer Science Department thinclient server for the past 3 years. The relevant contents have been placed as reported into a file provided with this assignment: *logindata.dat*.

Prepare a Python script which reads the *logindata.dat* file. This script should separate those records which contain individual user login records from those which have system actions recorded in them. You will need to manually review the file to determine the logic by which this differentiation will be made. *Hint: Look for sub-strings within each record such as "reboot", "system boot", "system reboot" which will allow you to distinguish between those records which are system and those which are generated by login activity.*

NOTE: The records in the file are in reverse chonological order, that is, most recent records are near the top and earliest records are near the end of the file.

In this assignment we will need to use common logic structures, file processing and string object methods to extract relevant data from the log file.

From the file extract the following information:

1. The total number of records in the file.
2. How many times was the system rebooted.
3. How many times was the system brought up.
4. How many times did the user *warren* login.
5. How many times did the user *jwoodcock* login.
6. What was the date of the first login of the user *warren*.
7. What was the date of the last login of the user *warren*.

The Python script for this part of the exercise should be submitted in a file:

Csci1523Assignment3B.py

2. **BONUS Question:** For each entry in the *passwd.dat* file, how many times have each of these users logged in over the period of time *logindata.dat* was collected. Print out the number of login instances for each user along with the total number of login instances for all users.

The Python script for this part of the exercise should be submitted in a file:

Csci1523Assignment3C.py

Basis of grading:

1. **Both scripts should include proper titles which include: *Assignment number, course number, student name, date, a description of the problem being solved.***
2. **Any functions used should include *doc strings* to document *function name, formal parameters, function description, description of return values, error conditions which may occur.***
3. **Access to any file should be done within a *try-except* block and appropriate error handling provided.**
4. **Functions which are used in both Question 1 and 2 should be packaged in a module. This is considered a important part of this assignment and will be evaluated with the assessment of the program logic portion.**
5. **There is no particular format required for the output other than the output should be properly labeled as to what is data is being printed. This will be assessed as part of the program implementation section.**
6. **The accuracy of the data is not of top importance so far as small deviations from actual results are concerned. Large deviations from expected values will result in reductions to both logic and implementation evaluations. Try to be careful here.**

A properly completed assignment will be the programs required and the bonus part if done and the data files in .zip compressed file.