SAS® to the Rescue! A UNIX® System Administrator's Tool

Lisa P. Wayte
University of Central Florida, Orlando, Florida

Abstract

As a UNIX System Administrator, you are faced with a variety of problems that require you to write your own programs. These programs are usually script files that contain shell programming commands and may include languages such as C® and Perl®. Even though C programs are usually portable and efficient, they are very hard to understand and maintain. In combining SAS with some basic ksh® shell script files, your applications become easier to understand and maintain. With SAS's data manipulation, graphics and reporting features, you will be able to produce reports on system usage and perform computer account generation with ease.

Introduction

In UNIX System Administration, a variety of programs are needed to produce graphical reports for system usage, and perform computer account generation for new users. This is all accomplished with the use of ksh shell script files and SAS/Base®, SAS/Graph®, and SAS/STAT®.

The script files are used either to extract data from binary files on the UNIX server or to generate the user computer accounts. The SAS programs are used to manipulate data so that reports can be generated or secondary script files can be easily coded.

The binary file on the UNIX server can contain information about a particular user or system. Several of the SAS applications that use the binary files are: extracting expiration dates of user accounts for account deletions; extracting login information to track system usage; and using a SAS application as a preprocessor for generating user accounts and then producing the reports.

User Account Deletions

An AIX® command called "lsuser," lets you obtain information about a particular user's account. This information is stored as a binary file on the system. In combining command options, you are able to create an ASCII file that SAS will use.

The "lsuser" command looks like this:

```
lsuser -a expires ALL > expires.out
```

The file produced looks like this:

```
userid expires=expiration_date
userid expires=expiration_date
```

From this file, you will be able to view user accounts and their expiration dates. This ASCII file will need to be sorted so that all the accounts that have expired will be located at the beginning of the file. See Appendix A for SAS Programs.

The output data file might look like this:

```
51595 aac331
51595 aaz924
.
51595 zf2176
81595 a9130
.
```

Next, delete the observations in the data set that do not meet the selection criteria. Then delete, or remove the date values from the data set.

The output of the data file should look like this:

```
userid
userid
userid
```

From this output, you are able to use a ksh shell script file to delete any user accounts that have expired after a certain date.

The script file might look like this:

```
rmuser -p userid
rm -ir /var/tmp/spool/mail/userid
```

User Group Reports

Another important process that SAS can assist in is in obtaining information about a particular "group" and determine how many users are in that particular "group."
Systems Architecture

This data extraction is also partly done with the use of the “lsuser” command and a SAS program that performs frequency accounts based on the “groups” defined on the UNIX server. See Appendix A for SAS Programs.

The lsuser command looks like this:

```
lsuser -a groups ALL > groups.output
```

The file produced looks like this:

```
userid group=xxx,xxx,xxx
userid group=xxx,xxx,xxx
userid group=xxx
```

This report is very useful when you want to see a breakdown between a college, department, or class. The report will show what groups are utilizing the system the most.

Generating Student Class Accounts

The student database is a file that is used to generate student computer accounts. This file contains information such as the student id number, their name, course and section of classes currently enrolled. Since the student database structure can change frequently, it was decided that SAS would be best to use in managing the format of the data.

A SAS program is used to read in the student database and combine information to produce the following:
- a directory path based on the students first name
- a userid based on the name of the student and their student id number
- a password that is generated
- an expiration date that is determined by the academic calendar year ending date
- the type of shell, or user environment requested
- the group(s) that the student belongs to based on the course number
- the student’s name

The output file produced by the SAS program looks like this:

```
x xxx00000 1234 121895 ksh mar3023 name
```

After the student database is file produced, you can process the information to produce computer user accounts. Also, you can generate a password summary report for the instructor listing all the students in a particular class, and generate the “banner” sheets that are given to the student. These “banner” sheets contain information such as their userid, password, what computer system the user has access to, basic instructions on using the computer account, and the expiration date of the account. See Appendix B for SAS Program.

Only the generation of student class accounts requires that an additional ksh script file be run. The other reports are generated based on the SAS program called class.sas.

Graphing User Loads

Finding out how much your computer system is being utilized can be a puzzle. Every time a user logs onto a computer and logs off a computer a file called wtmp is updated. This information is stored as a binary format on the system.

A ksh script file is first used to extract the userid, the time logged onto the computer, the time logged off the computer and the actual time spent logged onto the system.

Next, a SAS program reads the information that the ksh script file produces and generates a bar graph with frequency counts. The graphs are generated on a weekly basis for each client workstation in the computer labs and the server. See Appendix C for SAS Program and Sample Output.

Conclusions

SAS programs can play a very crucial role in UNIX System Administration when the technical expertise of the support staff is not familiar with programming languages such as C, Perl, and ksh scripts. SAS is far superior in data manipulation than the programming languages mentioned above. Also, SAS, with its graphic capabilities, makes information meaningful by providing visual representation of complex system data.

SAS is a registered trademark or trademark of SAS Institute, Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are registered trademarks of trademarks of their respective companies.

Contact: Lisa P. Wayte
University of Central Florida
Computer Services
4000 Central Florida Boulevard
Orlando, Florida 32812-2500
Phone:(407) 823-5067
e-mail: lisa@pegasus.cc.ucf.edu
Appendix A: SAS Programs For Account Expirations

exp95cnt.sas

This program reads in the user id and their expiration date then sorts the
user id's by expiration date.

options linesize=80;
options pagesize=55;
filename expires '/tmp/count';
libname out 'xxxx';
DATA out.count95;
length exprrdate 8 10;
INFILE expires;
INPUT
    account $
    exprrdate $;
run;
PROC SORT; BY exprrdate;
RUN;
PROC PRINT;
RUN;

asc95cnt.sas

This program uses the above sorted data file, creates an ascii file that
contains the sorted user id's by expiration date. The output data file is
used with a bash script file for further processing.

options linesize=80;
libname ssd01 'xxxx';
FILENAME MYOUT DISK '/xxxx/count95.output';
DATA _NULL_;
SET ssd01.count95;
FILE MYOUT;
PUT exprrdate 10. account;
run;

cnt95p2.sas

This program can be used to make group datasets based on which
group your account belongs to as well as the expiration date datasets.

options linesize=80;
options pagesize=55;
filename part2 '/xxxx/count95.output';
libname out '/xxxx';

DATA out.cnt95p2 out.admin out.system
    out.year95;
INFILE part2;
INPUT
    mo 1-2
    day 3-4
    time 5-8
    year 9-10
    account;
    if mo=0 then output out.admin;
    if year=70 then output out.system;
    if year=95 then output out.year95;
run;

PROC PRINT data=out.admin;
var account mo day year time;
TITLE 'INITIAL ADMIN DATA FILE';
run;

PROC PRINT data=out.system;
var account mo day year time;
TITLE 'INITIAL SYSTEM DATA FILE';
run;

PROC PRINT data=out.year95;
var account mo day year time;
TITLE 'INITIAL YEAR 1995 DATA FILE';
run;

PROC SORT DATA=out.year95;
    by mo day year account;
run;

PROC PRINT data=out.year95;
var account mo day year time;
TITLE1 'SORTED DATA FILE YEAR 1995';
TITLE2 'ACCOUNT, MONTH, DAY, YEAR & TIME.';
run;

proc freq data=out.admin;
tables mo*account / nocol norow nopercent;
TITLE1 'FREQUENCY COUNTS FOR ADMIN';
TITLE2 'BY MONTH AND ACCOUNT.';
run;

proc freq data=out.system;
tables mo*account / nocol norow nopercent;
TITLE1 'FREQUENCY COUNTS FOR SYSTEM';
TITLE2 'BY MONTH AND ACCOUNT.';
run;

proc freq data=out.year95;
tables mo*day / nocol norow nopercent;
TITLE1 'FREQUENCY COUNTS YEAR 1995';
TITLE2 'BY MONTH AND DAY.';
run;
Appendix B: Class Account Generation Program

class.sas

options linesize = 80;
options pagesize = 60;
options nodate;

filename in '/xxxx/xxxxxx.xxxx';

data tempa;
  infile in;
  input
    ssn  $ 1-9 ssn5 $ 4-9 lname $ 10-27 ln $ 10
    fname $ 28-42 fn $ 28
    mi $ 43 enrolled $ 44 pre1 $ 46-48
    numl $ 50-58 pre2 $ 60-62 num2 $ 64-72
    pre3 $ 74-76 num3 $ 78-86 pre4 $ 88-90
    num4 $ 92-100;

dir=lowcase(fn);
expires= '1215170095';
expires = '12/15/95';
group= 'eas4700';
shell = 'ksh';

if (enrolled = ' ') then delete;
run;

proc sort;
  by ssn;
run;

filename in 'xxxx/xxxxxx.xxxx';

data tempb;
  infile in;
  input
    rec 1.9;

  if rec = 3 then
    input
      ssn  $ 2-10 ssn4 $ 2-5 date $ 108-111;
run;

proc sort;
  by ssn;
run;

data temp;merge tempa tempb;
  by ssn;
run;

/***********************************************************/
data temp2 temp3;
  length name $8;
  length class $12;
  length group $8;
  length userid $8;
  length passwd $8;
set temp;

if mi = ' ' then do;
  name = fn||ln||ssn5;
  fn = lowcase(fn);
  name = lowcase(name);
end;

if mi = ' ' then do;
  name = fn||mi||ln||ssn5;
  fn = lowcase(fn);
  name = lowcase(name);
end;

passwd=xxxxxxxxx;
userid=xxxxxxxxx;
class= 'eas4700c0001';
group= class;

class1 = pre1||num1;
class2 = pre2||num2;
class3 = pre3||num3;
class4 = pre4||num4;

class1 = lowcase(class1);
class2 = lowcase(class2);
class3 = lowcase(class3);
class4 = lowcase(class4);

if (class=class1) or (class=class2) or
  (class=class3) or (class=class4)
then output temp3;
run;

proc sort data=temp3;
  by lname;
run;

FILENAME MYOUT DISK '/xxxx/xxxxxxxxxx.gen';
DATA _NULL_;
SET temp3;
FILE MYOUT;
PUT fn userid @12 passwd group expires2
  shell fname mi lname;
run;

FILENAME MYOUT DISK '/xxxx/xxxxxxxxxx.ban';
DATA _NULL_;
SET temp3;
FILE MYOUT;
/* Use "PUT Statements" to place a
description of how to log in to your
computer system here. Also, include the
userid and password for the individual. */
Appendix C: Graphing User Loads

Olympus.sas

options linesize=80;
options pagesize=60;
options nodate;

filename in
'\x/x/xxxxx/xxxxxxx/xxxxxxx/olympus.flat';
libname ssd01 '\x/x/xxxxx/xxx/xxxxxxx';

DATA ssd01.olympus;
INFILE in firstobs=145 obs=1462;
INPUT
   mo $
   day
   date $ 6.
   time1 time5.
   time2 time5.
   time3 time5.
   time4 account $;
run;

/******************************
DATA ssd01.olympus;
set ssd01.olympus;

if (mo='Jul') and (day >= 31) then week=13;
if (mo='Aug') and (day <= 6) then week=13;

if (week=.) then delete;
******************************

hr1 = hour (time1);
min1 = minute (time1);

hr2 = hour (time2);
min2 = minute (time2);

hr3 = hour (time3);
min3 = minute (time3);

/******************************
hr1 = hour (time1);
min1 = minute (time1);

hr2 = hour (time2);
min2 = minute (time2);

hr3 = hour (time3);
min3 = minute (time3);

if (min1 >= 30) then tmin2=30;
output;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
### Systems Architecture

**Olysu95.lst**

The following is a sample user login report for given week on a RiSC/6000 AIX Server.

**Bar Graph For May 15 To June 25, 1995**  
For Server: Olympus.  
Time of Login vs. Number of Logins

<table>
<thead>
<tr>
<th>HOUR</th>
<th>Cum. Freq</th>
<th>Freq</th>
<th>Percent</th>
<th>Cum. Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>22</td>
<td>22</td>
<td>2.44</td>
<td>2.44</td>
</tr>
<tr>
<td>30</td>
<td>21</td>
<td>43</td>
<td>2.33</td>
<td>4.77</td>
</tr>
<tr>
<td>100</td>
<td>12</td>
<td>55</td>
<td>1.33</td>
<td>6.10</td>
</tr>
<tr>
<td>130</td>
<td>14</td>
<td>69</td>
<td>1.55</td>
<td>7.66</td>
</tr>
<tr>
<td>200</td>
<td>8</td>
<td>77</td>
<td>0.89</td>
<td>8.55</td>
</tr>
<tr>
<td>230</td>
<td>9</td>
<td>86</td>
<td>1.00</td>
<td>9.54</td>
</tr>
<tr>
<td>300</td>
<td>5</td>
<td>91</td>
<td>0.55</td>
<td>10.10</td>
</tr>
<tr>
<td>330</td>
<td>5</td>
<td>96</td>
<td>0.55</td>
<td>10.65</td>
</tr>
<tr>
<td>400</td>
<td>7</td>
<td>103</td>
<td>0.78</td>
<td>11.43</td>
</tr>
<tr>
<td>430</td>
<td>4</td>
<td>107</td>
<td>0.44</td>
<td>11.88</td>
</tr>
<tr>
<td>500</td>
<td>5</td>
<td>112</td>
<td>0.55</td>
<td>12.43</td>
</tr>
<tr>
<td>530</td>
<td>4</td>
<td>116</td>
<td>0.44</td>
<td>12.87</td>
</tr>
<tr>
<td>600</td>
<td>4</td>
<td>120</td>
<td>0.44</td>
<td>13.32</td>
</tr>
<tr>
<td>630</td>
<td>5</td>
<td>125</td>
<td>0.55</td>
<td>13.87</td>
</tr>
<tr>
<td>700</td>
<td>6</td>
<td>131</td>
<td>0.67</td>
<td>14.54</td>
</tr>
<tr>
<td>730</td>
<td>8</td>
<td>139</td>
<td>0.89</td>
<td>15.43</td>
</tr>
<tr>
<td>800</td>
<td>9</td>
<td>148</td>
<td>1.00</td>
<td>16.43</td>
</tr>
<tr>
<td>830</td>
<td>9</td>
<td>157</td>
<td>1.00</td>
<td>17.43</td>
</tr>
<tr>
<td>900</td>
<td>17</td>
<td>174</td>
<td>1.89</td>
<td>19.31</td>
</tr>
<tr>
<td>930</td>
<td>18</td>
<td>192</td>
<td>2.00</td>
<td>21.31</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
<td>217</td>
<td>2.27</td>
<td>24.08</td>
</tr>
<tr>
<td>1030</td>
<td>33</td>
<td>250</td>
<td>3.66</td>
<td>27.75</td>
</tr>
<tr>
<td>1100</td>
<td>25</td>
<td>275</td>
<td>2.77</td>
<td>30.52</td>
</tr>
<tr>
<td>1130</td>
<td>16</td>
<td>291</td>
<td>1.78</td>
<td>32.30</td>
</tr>
<tr>
<td>1200</td>
<td>15</td>
<td>306</td>
<td>1.66</td>
<td>33.96</td>
</tr>
<tr>
<td>1230</td>
<td>23</td>
<td>329</td>
<td>2.55</td>
<td>36.51</td>
</tr>
<tr>
<td>1300</td>
<td>25</td>
<td>354</td>
<td>2.77</td>
<td>39.29</td>
</tr>
<tr>
<td>1330</td>
<td>25</td>
<td>379</td>
<td>2.77</td>
<td>42.06</td>
</tr>
<tr>
<td>1400</td>
<td>26</td>
<td>405</td>
<td>2.89</td>
<td>44.95</td>
</tr>
<tr>
<td>1430</td>
<td>32</td>
<td>437</td>
<td>3.55</td>
<td>48.50</td>
</tr>
<tr>
<td>1500</td>
<td>35</td>
<td>472</td>
<td>3.88</td>
<td>52.39</td>
</tr>
<tr>
<td>1530</td>
<td>36</td>
<td>508</td>
<td>4.00</td>
<td>56.38</td>
</tr>
<tr>
<td>1600</td>
<td>42</td>
<td>550</td>
<td>4.66</td>
<td>61.04</td>
</tr>
<tr>
<td>1630</td>
<td>37</td>
<td>587</td>
<td>4.11</td>
<td>65.15</td>
</tr>
<tr>
<td>1700</td>
<td>48</td>
<td>635</td>
<td>5.33</td>
<td>70.48</td>
</tr>
<tr>
<td>1730</td>
<td>33</td>
<td>668</td>
<td>3.66</td>
<td>74.14</td>
</tr>
<tr>
<td>1800</td>
<td>21</td>
<td>689</td>
<td>2.33</td>
<td>76.47</td>
</tr>
<tr>
<td>1830</td>
<td>23</td>
<td>712</td>
<td>2.55</td>
<td>79.02</td>
</tr>
<tr>
<td>1900</td>
<td>20</td>
<td>732</td>
<td>2.22</td>
<td>81.24</td>
</tr>
<tr>
<td>1930</td>
<td>14</td>
<td>746</td>
<td>1.55</td>
<td>82.80</td>
</tr>
<tr>
<td>2000</td>
<td>21</td>
<td>767</td>
<td>2.33</td>
<td>85.13</td>
</tr>
<tr>
<td>2030</td>
<td>21</td>
<td>788</td>
<td>2.33</td>
<td>87.46</td>
</tr>
<tr>
<td>2100</td>
<td>12</td>
<td>800</td>
<td>1.33</td>
<td>88.79</td>
</tr>
<tr>
<td>2130</td>
<td>21</td>
<td>821</td>
<td>2.33</td>
<td>91.12</td>
</tr>
<tr>
<td>2200</td>
<td>20</td>
<td>841</td>
<td>2.22</td>
<td>93.34</td>
</tr>
<tr>
<td>2230</td>
<td>22</td>
<td>863</td>
<td>2.44</td>
<td>95.78</td>
</tr>
<tr>
<td>2300</td>
<td>23</td>
<td>886</td>
<td>2.55</td>
<td>98.34</td>
</tr>
<tr>
<td>2330</td>
<td>15</td>
<td>901</td>
<td>1.66</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

1456