1 Visualizing Email

Write an AWK program that builds a graph that shows who emails whom on our system. The idea is to search through /var/log/syslog on lectura and count the number of times each user emails another user. From this information we generate a file which can be read by the dot program which will generate a graph that we can examine.

You should build an AWK program mail2dot which should be run like this:

```
> mail2dot /var/log/syslog >! graph.dot
> /cs/linux/graphviz-1.8.5/bin/dot -Tps graph.dot >! graph.ps
```

Note that on our installations dot only runs under linux.

The format of the /var/log/syslog mail entries is as follows:

```
Feb 14 09:05:35 lectura.CS.Arizona.EDU sm-mta[3305]: [ID 801593 mail.info] h1EG5ZtS003305: \
from=<tapas@CS.Arizona.EDU>, size=1947, class=0, nrcpts=2, \
msgid=<Pine.GSO.4.33.0302140901060.2005-100000@lectura.CS.Arizona.EDU>, \
proto=ESMTP, daemon=Daemon0, relay=localhost [127.0.0.1]
```

```
Feb 14 09:05:35 lectura.CS.Arizona.EDU sendmail[3304]: [ID 801593 mail.info] h1EG5YhD003302: \
to=<collberg@CS.Arizona.EDU>,<jennifer@atc-nycorp.com>, delay=00:00:00, xdelay=00:00:00, \
mailer=relay, pri=150412, relay=localhost [127.0.0.1], dsn=2.0.0, \
stat=Sent (h1EG5ZtS003305 Message accepted for delivery)
```

Note that all the information is on one line. The first entry specifies that tapas@CS.Arizona.EDU sent a message with message-identifier h1EG5YhD003302. The second entry says that this message was sent to collberg@CS.Arizona.EDU and jennifer@atc-nycorp.com.

From this information we would generate a dot-file like this:

```
digraph G {
    page="8.5,11";
    margin=0;
    ratio=auto;
    pagedir=TL;
    n1 [label="tapas"]; 
    n2 [label="collberg"]; 
}
```
The first five lines are declarations that describe how the graph should be laid out. The next three lines declare three nodes, labeled by email addresses. The next two lines add edges to the graph, labeled by the number of messages that the sender has sent to each receiver (only one, in this example). The graph generated by `dot` will look like this:

A larger example is shown in Section 6.

2 Hints

My highly-unoptimized solution is 81 lines long and looks roughly like this:

```bash
#!/usr/local/bin/gawk -f
BEGIN {
  MINCOUNT = 4  # Minimum edge weight
}
# Delete crud around email addressed and simplify by deleting
# "@*.cs.arizona.edu" and ".arizona.edu".
function strip(A) {
  ...
}
# Map an email address to a unique number.
function insertUser(A) {
  ...
}
function buildGraph() {
  print "digraph G {
  print "  page="8.5,11";"
  print "  margin=0;"
  print "  ratio=auto;"
  print "  pagedir=TL;"
  ......  
  print "}"
```
}  
# Deal with the sender entry in /var/log/syslog  
pattern { 
   ....  
}  
# Deal with the receiver entry in /var/log/syslog  
pattern {  
   ....  
}  
END {  
   buildGraph()  
}

You can learn more about dot here: http://www.research.att.com/sw/tools/graphviz/. There are  
many facilities for tweaking the layout of the graph, adding color, etc.

dot is an extremely useful research tool since graphs occur in almost all Computer Science applications. It  
is well worth the time to learn how to use it.

3 Extension (No extra credits)

To avoid the graph being too huge, strip out edges with small weights and the resulting disconnected nodes.

To make the nodes smaller it makes sense to strip out “obvious” parts of email addresses. For example,  
collberg@cs.arizona.edu should convert to collberg, collberg@email.arizona.edu should convert to  
collberg@email, etc.

Add an option to mail2dot that labels edges with the dates the messages were sent rather than just the  
number of messages sent.

Fiddle with dot’s graph layout primitives to get as pretty and readable graphs as possible. You can, for  
example, make edges with large weights thicker. Can you make good use of color to improve the effectiveness  
of the visualization?

4 Extension (Some extra credits)

Write a 10-page research paper on the privacy issues involved in the decision to make /var/log/syslog  
world readable.

5 Submission and Assessment

The deadline for this assignment is 10:30AM, Wednesday, February 26. You should submit the assignment  
electronically using the Unix command `turnin cs520.5 <files>`. This assignment is worth 2% of your  
final grade.
Don’t show your code to anyone, don’t read anyone else’s code, don’t discuss the details of your code with anyone. If you need help with the assignment see the instructor.

6 Example Output