

More Using the spooler to print to email: email attachments

by Dirk Hart

We already know how to modify a spooler interface script to email plain text to people, but it bothered me that print jobs with embedded PCL commands or other escape codes might get mangled in the emailing process.

Normally when we want to email stuff with non-text characters in it we have to uuencode it on one end and uudecode it on the other end just so we can be sure the whole thing gets transmitted properly. This is no guarantee that the file won't look like trash, it just makes sure the whole file arrives unmangled. What uuencode does is convert all the characters in a file to ascii (ascii *pairs* of characters actually) and uudecode is our friend when we want to undo that process.

The unix mail command isn't smart enough to uuencode stuff. You could write a script that places attachment boundaries (markers) around your uuencoded file and then email the entire mess away, but that's too much like work. Fortunately we have another mail program called mutt that does the job for us.

I found mutt at <ftp://ftp2.sco.com/pub/skunkware/osr5/vols/> in a file called mutt-1.2.5.1-VOLS.tar which I installed on my ancient OpenServer 5.0.4 system with custom. This part was easy but I soon ran into trouble when I tried to execute mutt. I got an error that said dynamic linker : mutt : error opening /usr/local/lib/libncurses.so.4 . This was not the same result as having my email attachments bombard the internet, so I cast about for a solution. I searched high and low for libncurses, ncurses and curses to no avail, until someone pointed out that the ncurses package was on the skunkware ftp site (thanks, JPR). Off I flew with my browser to <ftp://ftp2.sco.com/pub/skunkware/osr5/vols/> and just a few lines below the mutt package was the ncurses package I needed, which I downloaded and installed with custom.

This time when I tried mutt I got a 'character-graphical' screen with a mini-menu and such that made it easy to send a test message. So I did.

Since that seemed to work just fine I typed in mutt -help and discovered how to attach a file to an email. From the command line point of view seems to be a lot like the familiar mail command. I typed in echo test|mutt -a /etc/hosts -s test boopy and sure enough boopy received an email with the contents of /etc/hosts. Pretty painless, actually.

Next, I set about modifying the spooler interface script for my all-too-cleverly named print-to-email spooler destination, called email. Previously we had modified the FILTER or modified the script like this:

```
{
while [ $i -le $copies ]
do
  for file in $files
  do
    0<${file} eval ${FILTER} 2>&1
    echo "\014\c"
  done
  i=`expr $i + 1`
done
} | mail -s report boopy
```

But it wasn't clear to me if this would email **all** files in \$files or the first or the last (not that I tested this, though), so I changed the script to the less ambiguous

```
while [ $i -le $copies ]
do
  for file in $files
  do
    {
      0<${file} eval ${FILTER} 2>&1
      echo "\014\c"
    } | /usr/local/bin/mutt -a $file -s test boopy
  done
  i=`expr $i + 1`
done
```

incorporating the appropriate changes for mutt. This change will send each \$file in \$files as a separate email to boopy. When boopy joyously receives our email our attachment will show up, and if (in mozilla) we have check View-->Display Attachments Inline the uuencoded (and decoded) document is displayed including all the non-ascii characters we might have sent. But at least we have the option of printing the attachment along with all its PCL codes.