ToolMan Edits Pipes – Interactively!

This issue I'll describe a handy little tool that I use in many situations in my day-to-day work. This one should be a welcome addition to the toolbox of any UNIX shell hackers out there. It's called `vip` (VI Pipe), and it enables you to interactively edit any point in a pipeline, whether at the beginning, middle, or end.

Toolz R Us

Normally, in a pipeline, when you need to edit some phase of the data stream, you use a standard tool such as `sed`, `grep`, or `awk` to alter, filter, or otherwise manipulate the stream. One potential problem with this approach is that the manipulations have to be very well thought out in advance. Another is that the manipulations will probably need to be applied uniformly. And third, the data must be very well understood in advance. Not all situations and data easily conform to these constraints.

Alternatively, when the changes needed for the data are more than trivial, or perhaps you just don't feel like expending the mental energy needed to work out all the expressions in advance, a typical approach might be to run some process or pipeline, dump output to a file, edit the file with `vi`, `pico`, or `emacs`, then push the data along to the next phase by using the file as input to some additional process or pipeline. The catch here – other than the sheer awkwardness of this process – is that you have to remember to come back later and clean up all of those little and not-so-little "temporary" files.

So, wouldn't you just like to be able to tap in an edit session at any arbitrary point in the pipeline, do your magic on the data, then have it automagically continue on its merry way? The `vip` program provides this functionality, and operates syntactically just like any other filter.

Example 1

Here's a little example. Suppose you want to send Xena a zephyrgram. (Perhaps you're working from a Telnet session, and `xzwrite` is not an option.)

Option 1: `zwrite xena`. You have to type in lines of text, and once you hit "<return>" on a line, you can't go back and fix things. Goof up badly enough, and all you can do is hit "<control-C>" and start all over.

Option 2: Edit a file, then `zwrite xena < tfile`. Then you have to remove the file. This is tedious.

Option 3: `vip -n | zwrite xena`. Edit a message, exit the editor, and away goes the zgram, no temp files to worry about. (The `-n` tells `vip` not to read standard input before invoking the editor.)

Example 2

Let's make it a little more complicated. Let's say that part of what you want to zgram Xena about is a report of disk utilization produced by the `df` command. In this case, you could type something like

```
df -lk | vip | zwrite xena
```

When the editor starts up, `df`'s output is already there in the edit buffer. Add your embellishments and exit.

The "ToolMan" articles, seen in recent issues of *;login:*, will continue on a regular basis. ToolMan needs your help! If you have written a software tool that is unique, useful, or really cool, please send ToolMan <des@cs.duke.edu> a description, and he'll try to feature it in an upcoming issue.

Example 3

One more example, this time with `vip` at the end of a pipeline. Suppose we have some subdirectories and want to copy a new version of a file into a few of them. The subdirectories look like this:

```
% ls -F
Distfile                sgi-IRIX64_6.1/         sparc-SunOS_5.3@
dec-OSF1_V3.2/          sparc-SunOS_4.1.3@      sparc-SunOS_5.4@
i386-SunOS_5.4@         sparc-SunOS_4.1.3C@     sparc-SunOS_5.5@
i386-SunOS_5.5@         sparc-SunOS_4.1.3_U1@   sparc-SunOS_5.5.1/
i386-SunOS_5.5.1/       sparc-SunOS_4.1.4/
```

Run the command `ls | vip -o`. (The -o tells `vip` not to send the buffer to standard output after editing.) In the edit buffer, we edit the list and add some shell syntax to get this:

```
for DIR in \
  sgi-IRIX64_6.1 \
  dec-OSF1_V3.2 \
  sparc-SunOS_5.5.1 \
  i386-SunOS_5.5.1 \
  sparc-SunOS_4.1.4
do
  cp /cs/puma/src/bin/zippy $DIR/bin
done
```

Then we send it to a shell with ":w !sh<enter>" to execute the commands. If we decide to perform some additional operations on these files, such as to change their permissions, it's easy enough to just alter the `cp` line in the edit buffer and send the commands to a shell again. (This could have been done with `ls | vip | sh`, but that wouldn't have allowed for iterations of command executions.)

Smoke and Mirrors

I have to admit that temporary files are at work here. But the cleanup is automagic and transparent to the user. Also, just to prove that I'm not a `vi` nazi, `vip` is not restricted to just that editor. It will check your VISUAL and EDITOR environment variables and behave accordingly.

`vip` is a Bourne shell script that does some redirection, uses `umask` to protect privacy, and uses `trap` to clean up if a common signal is caught. It is, in fact, rather elementary.

Cop It

Even if you thought `duf` didn't have the stuff (August 1997), or you said to heck with `check` (June, '97), I think you'll like `vip`. The adventurous – and even the timid – can pick up a copy at <http://www.cs.duke.edu/~des/scripts.html> or <ftp://ftp.cs.duke.edu/pub/des/scripts/>. Don't forget the `man` page! Happy plumbing!

Note: A *much* faster version of `duf` (1.16 or later) is now available (and no more temporary sort files!).