Josh Kunz
xargs Analysis
January 28, 2014

`xargs` is a standard UNIX utility (from the findutils utility package) for converting file contents into arguments for other programs. Typically `xargs` is employed when the user wishes to run a command over a set of input files. For example, if a user had a directory full of `*.wav` files that they wanted to convert to `*.mp3` files, they might use xargs like so:

```
find . -name "*.wav" -print0 | xargs -0 -n1 lame -V0
```

The `find` utility is used to build a list of `wav` files which is then passed to `xargs`. `xargs` converts this input to a list of arguments (in this case each files is an argument) and then passes those arguments to utility program, in this case a popular wav to mp3 conversion tool `lame`. Here we use `xarg`'s `-n` argument to ensure that only one file is passed to `lame` at a time. Without this option `xargs` will try to pass as many arguments to the utility program as it can.

From this command we can already see some peculiarities of the `xargs` command, for example `xargs`'s `-0` option. Under normal operation, input to `xargs` is *parsed* according to the conventions normally used by shells. Input lines are split into arguments on whitespace (so the line `abc def` would be parsed a two inputs `abc` and `def`), whitespace can be escaped with quotes (`"abc def"` becomes one argument) and quote characters can be escaped with backslashes (`"abc \" def"` is parsed as `abc " def`).

What this means, is that if you want run a command over a set of files, those files have to be properly escaped before they are given as input to `xargs` . This can be very confusing for users of `xargs` who might expect it to simply use each line as argument instead of parsing each line into arguments. This confusing behaviour should be considered before `xargs` is relied upon.

As can be seen above `xargs` has added a work around for this problem, the `-0` option. This option causes `xargs` to treat everything between two null-terminators (or a null-terminator and an `EOF`) as an argument, no parsing is done on the input.