Processing Text Streams

Fermilab Linux Users Group
May 30, 2012

Filters

Pipes redirect the output of one command to the input of another.

By "piping" simple commands together, we can read and modify text files on the commandline.

These commands mostly take filenames, or if no filename given (or "-" as filename), read from stdin.

File Redirection

In addition to redirecting between programs, we can redirect output to and from files using <, > and >>.

```
cmd < file receive stdin from file
cmd >> file append std output to
file
cmd > send std output to file
```

cat

Short for "concatenate", print files to standard output.

Syntax

```
   cat options [files]
```

Options

```
   -s (never print more than one
blank line)
```

tac

...the opposite of cat!

Syntax

```
    tac options [files]
```

Options

```
    -s, --separator=STRING use
STRING as separator instead of
newline
```

grep, egrep, fgrep

Print lines matching a pattern.

-Different executables, but all can be used with flags to grep

- default uses Basic Regular Expressions
- egrep or grep -E uses Extended Regex
- fgrep or grep -F uses fixed strings

grep, egrep, fgrep

-c print a count of matching lines

-s silent; output goes to /dev/null, you can use the return code in scripts

-l print the name of each output file with matches

-L print the name of each output file without matches

cut

Print selected columns or fields

Syntax

```
cut options [files]
```

Options

```
  -ddelim (define delimeter
                  [default ;])
-ffield (print this field)
```

paste

Print corresponding lines of one or more files into columns

Syntax

```
   paste options [files]
```

Options

```
      -ddelim (define delimeter
                             [default Tab])
```

head

Print the first few lines of a file

Syntax

```
    head options [files]
```

Options

```
     -nn (number of lines
                     [default 10])
```

tail

Print the last few lines of a file

Syntax

```
tail options [files]
```

Options

```
-nn (number of lines
                  [default 10])
```

-f ("follow", or append data as file grows)

tr

Translate or delete characters.

Syntax

```
    tr options SET1 [SET2]
```

Options

```
     -s suppress repeated chars
    -c use the complement of set
```

tr

Example

All uppercase:

```
    cat file1 | tr a-z A-Z
```

sort

Sorts lines of text files.

Syntax

```
    sort options [file]
```

Options

```
    -b ignore leading blanks
    -f ignore case
    -r reverse order
```

uniq

Print (or omit) repeated lines. Text must be sorted first!

Syntax

```
   uniq options [file[output]]
```

Options

```
    -d only print repeated lines
    -i ignore case
```

nl

Number lines of files; cat to stdout.

Syntax

```
    nl options [files]
```

Options

```
    -b p REGEXP (number only lines
matching REGEXP)
```

expand/unexpand

Tabs to spaces, spaces to tabs.

Syntax

```
    [un]expand options [files]
```

wc

Count lines, words, or characters in a file

Syntax

```
   uniq options [file[output]]
```

Options

```
    -c characters

    -w words

   -l lines
```

BONUS: File Descriptors

stdin, stdout and stderr have special file descriptors; 0, 1 and 2 respectively.

```
cmd 2> file send stderr to file
cmd > file 2>&1 send stderr and
stdout to file
```

.....etc

awk, sed

...For another day!