UNIX AND SHELL PROGRAMMING

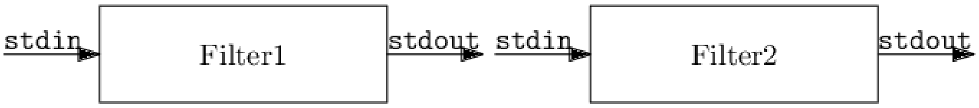
Filters

S Milton Rajendram

31 July 2019

1 Filters and pipes

- *Filter*: a command that reads its input from `stdin`, processes it, and writes its output to `stdout`.



2 Important filters

- Light
  - `cat`
  - `head`, `tail`
  - `cut`, `paste`
  - `sort`
  - `uniq`
  - `tr`
  - `cmp`, `comm`, `diff`
- Powerful
  - `grep`
  - `sed`
  - `awk`

3 Concatenate

- `cat file1 file2 file3`
- Concatenates `file1`, `file2`, and `file3` to `stdout`, one after another.

- Combines multiples files to a single file. `cat file1 file2 file3 > file4`
- Display a single file. `cat file`
- Create a file `cat > file` `C-d` signals the end of file – not a character.
- Usage:
  - `cat`
  - `cat < file1`
  - `cat > file2`
  - `cat < file1 > file2`
  - `cat file` : not used as filter

4 Display beginning and end of file

- `head -m` Default: `m = 10`
- `head -m file`
- `head -m file1 file2`
- `tail -m`: last `m` lines
- `tail -n+m`: from `m` th line

option	meaning
`-c`	count by characters

- Combine `head` and `tail` to select any sublist of lines

5 Cut, paste

- `cut` selects fields of input (files or `stdin`)
- Input file is in table or csv (comma separated values) format:
  - View it as a table: rows and colums
  - View it as a CSV file: lines and fields
    - Sequence of lines
    - Each line is a sequence of fields
    - Field separator
- Specify character positions `cut -cn1,n2-n3,n4-n5`
- Field specification – default delimiter (field separator): single tab `cut -fn1,n2-n3,n4-n5`

```
cat distros.txt | tr -s [:blank:] \\t > distros-tab.txt
```

```
cut -f3 distros.txt
```

```
cut -f3 distros.txt | cut -c7-10
```

- Specify delimiter (quote the delimiter, safe) `cut -d"c" -fn1,n2-n3,n4-n5`

  ```
  cut -d":" -f1 /etc/passwd | head
  ```

- `head` and `tail` selects lines (horizontal); `cut` selects fields (vertical)
- `-s` suppresses lines that do not have the delimiter.

-paste

- `paste files` combines lines from files horizontally, separate them tab (`cat` combines files vertically)
- `paste -` reads from `stdin`
- `paste file1 file2` of different sizes

  ```
  cut -f1,2 distros-tab.txt > distros-versions.txt
  cut -f3 distros-tab.txt > distros-dates.txt
  paste distros-dates.txt distros-versions.txt
  ```

- `-d"cc"` delimiter characters

6 Sort

6.1 Sort by lines

- ASCII code
- control characters (00-1F)
- digits (30-39)
- uppercase letters (41-5A)
- lowercase letters (61-7A)
- printable characters, in between
- `sort`
- `sort f1 f2 f3 > f4`

6.2 Sort by fields

- lines – terminated by newline
- fields – separated by a blank or a tab
- sort by (field1, field2, . . .) – sort keys

6.3 Sort by number

- `sort -n`

  ```
  du -s /usr/share/* | sort -n | head
  ```

6.4 Sort reverse

- `sort -r`

  ```
  du -s /usr/share/* | sort -nr | head
  ```

6.5 Sort by fields

- `sort -k m` sort by field `m, m+1, m+2, ...`
- `sort -k m,n` sort by fields `m` to `n`

  ```
  sort -k3,3 phone.txt
  ```

  ```
  sort -k1,1 -k2n phone.txt
  ```

  ```
  ls -l /usr/bin | sort -nrk 5 | head
  ```

  ```
  sort --key=1,1 --key=2n distros.txt
  ```

6.6 Ignore leading blanks

- Every blank/tab is counted as a separator.
- `sort -b` ignores leading blanks

  ```
  sort -k 3.7nbr -k 3.1nbr -k 3.4nbr distros.txt
  ```

6.7 Specify field separator

- `sort -t"c" options files`

  ```
  sort -t ':' -k 7 /etc/passwd | head
  ```

6.8 Merge

- `sort -m files`
- Input `files` are already sorted

6.9 Fold case

- `sort -f files`
- Convert upper case to lower case for sorting purpose

7 Uniqe lines

- `uniq files` removes duplicate lines – sorted lines input
- `sort -u files` removes duplicate lines

option	meaning
`-c`	count duplicate lines
`-d`	repeated lines
`-f n`	ignore `n` leading fields
`-i`	ignore case
`-u`	unqie lines

8 Transliterate from one set to another

- `stdin` to `stdout`, does not read file
- `tr options string1 string2`
  `tr "aeiou" "AEIOU"`
- Each character in set1 (`string1`) is converted to the corresponding character in set2 (`string2`)
- If set2 is smaller than set1, unmatched characters in set1 are converted to the last chracter in set2.

```
echo "not to be contentious, gentle," | tr aeiou AEIOU

echo "not to be contentious, gentle," | tr aeiou AE?

echo "Shout for Joy!" | tr a-z A-Z

echo "Shout for Joy!" | tr [:lower:] A
```

8.1 Delete characters

- `tr -d set`
  delete characters in `set`, do not translate.

```
echo "not to be contentious, gentle," | tr -d aeiou
```

8.2 Complement

```
echo "not to be contentious, gentle," | tr -c aeiou ?
```

8.3 Squeeze (delete) repeated instances of characters

- `tr -s set1 set2`
  replace each sequence of a repeated character listed in `set2`, with a single occurrence of that character

  ```
  echo "aaabbbccc" | tr -s ab

  echo "abcabcabc" | tr -s ab

  cat phone.txt | tr -cs a-zA-Z "\n"
  ```

  ```
  Robert
  M
  Johnson
  Lyndon
  B
  Johnson
  Samuel
  H
  Johnson
  Michael
  K
  Loukides
  Jerry
  O
  Peek
  Timothy
  F
  O
  Reilly
  ```

9 Comparing files

`cmp, diff, comm`

9.1 Compare (cmp)

- `cmp file1 file2`
  Displays the line number and byte number of the first differing byte
- `cmp -l file1 file2`
  Displays the line numbers and byte numbers of all differing bytes
- `cmp -s file1 file2`
  Suppress all output; used for exit status.

exit status	meaning
0	same
1	different

9.2 Difference (**`diff`**)

- `diff` always works on files (two files, two versions of a file)

  ```
  diff file1 file2
  ```

- Change command: `rage1 operation range2`

Operation	Action
r1ar2	At position r1 in file1, append lines at r2 in file2
r1cr2	Change (replace) lines at position r1 with the lines at position r2 in file2.
r1dr2	Delete lines at position r1 in file1, which would have appeared at range r2 in file2

  range is comma separated list of starting line and ending line

- Context format

  ```
  diff -c file1 file2
  ```

Character	Meaning
blank	This line is shared by both files
-	This line was removed from the first file.
+	This line was added to the first file.

- Unified format

  ```
  diff -u file1 file2
  ```

9.3 Patch (**`patch`**)

- Create a diff file

```
diff -Naur file1.txt file2.txt > patchfile.txt
```

```
patch < patchfile.txt
```

9.4 Common (**`comm`**)

- `comm file1 file2` compares two text files and displays 3 columns:
  - lines unique to `file1`
  - lines unique to `file2`
  - lines they have in common
- `comm -n file1 file2` supress column `n`
- `comm -n1n2 file1 file2` supress columns `n1,n2`