UNIX/Linux environment

Filters

Stefano Quer – Pietro Laface
Dipartimento di Automatica e Informatica
Politecnico di Torino

Filters

- In UNIX/Linux a **filter** is a command that
  - Takes its input from standard input
  - Process (filters) it according to some parameters and options
  - Produces its output on standard output
- Commands that are useful tools for text file processing

Filters

- Most popular filters
  - awk, cat, cut, compress, grep, head, perl, sed, sort, tail, tr, uniq, wc
- Some of these commands
  - Are quite complex
    - grep, sort
  - Are scripting languages
    - sed, awk
  - Often
    - Use **regular expressions**
    - Are used in **pipe** (by means of the **|** operator) with other commands

cut

- Selects and outputs sections from each line of files
  - Format
    - cut [options] file
  - Main options
    - --characters=LIST, -c LIST
      - Select only the characters at the positions in LIST
    - --fields=LIST, -f LIST
      - Selects the (comma separated) list of bytes (characters, fields)
    - --delimiter=DELIM, -d DELIM
      - Uses DELIM to separate fields rather than the default **TAB** delimiter

Examples

```
# Selects fields 1 and 3 of all file lines

cut –f 1,3 file.txt

# Selects fields 1 to 3 (1, 2 and 3)
# and 5 to 6 of file foo.txt
# fields are delimited by space rather than by TAB

cut –f 1-3,5-6 –d “ ” foo.txt
```

tr

- Translate, squeeze, and/or delete characters from standard input, writing to standard output.
- Must be used by redirecting the output of other commands to its input
  - Format tr [options] set1 [set2]
  - Main options
    - --delete, -d
      - Delete the characters in set_1
    - --squeeze-repeats, **-**s
      - replace each input sequence of a repeated character that is listed in set_1 with a single occurrence of that character
    - --complement, -c, -C
      - Uses the complement of set_1

Examples

```
# Outputs file.txt eliminating characters a, b, c, d

tr –d abcd < file.txt


# Outputs cIAo
echo ciao | tr ia IA


# squeezes the sequence of spaces to a single space
# Outputs a b c

echo a b    c  | tr –s ’ ’
```

uniq

- Report or omit repeated lines of the input file
  - Format
    - uniq [options] [inFile] [outFile]
  - The file must be sorted
  - Without options eliminates the repeated lines

uniq

- Main options
  - --count, -c
    - prefix lines by the number of occurrences
  - -repeated, -d
    - only print duplicate lines, one for each group
  - --skip-fields=N, -f N
    - avoid comparing the first N fields
  - --ignore-case, -I
    - Case insensitive

Examples

```
# Eliminates duplicate lines of a sorted text file
# outputs the others and
# inserts the number of occurrences
# of the duplicated lines

uniq –-count file.txt
uniq –c file.txt

# Outputs the duplicated lines only

uniq –d a.x
```

basename

- Eliminates the directories form a pathname, and possibly its extension
  - Format
    - **`basename pathname [extension]`**

Examples

```
> basename /home/user1/current/file.txt
file.txt
> basename /home/user1/current/file.txt ”.txt”
file
> basename /home/user1/current/file.txt .txt
file
> basename /home/user1/current/file.txt txt
file.
```

sort

- Sort the input file in alphabetic order
  - Format
    - **`sort [options] [file]`**
  - Main options
    - **`--ignore-leading-blanks, -b`**
    - **`--dictionary-order, -d`**
      - Considers spaces and alphabetic characters only
    - **`--ignore-case, -f`**

sort

- **`--numeric-sort, -n`**
  - Sort in numeric order
- **`--reverse, -r`**
  - Sort in reverse order
- **`--key=c1,[,c2], -k c1[,c2]`**
  - Sort on the basis of the selected fields
- **`--merge, -m`**
  - Merges sorted files, no sort without other options
- **`--output=f, -o=f`**
  - Writes its output on file f rather than on standard output

grep

- Global Regular Expression Print
  - Searches the input files for lines containing a match to the given pattern. If no files are specified, or if the file "-" is given, grep searches standard input. By default, grep prints the matching lines.
- Versions
  - grep
  - egrep, fgrep, rgrep
    - Egrep equivalent to ”grep –E”
    - Uses Extended RE for matching the pattern

grep

- Format
  - **`grep [options] pattern [file]`**
- Main options
  - **`--line-number, -n`**
    - Outputs the matching line number
  - **`--recursive, -r, -R`**
    - Search recursively the sub-trees
  - **`--inverse-match, -v`**
    - Outputs only the lines that do not match
  - **`--ignore-case, -I`**
    - Case insensitive

grep

- **`--regexp=PATTERN, -e PATTERN`**
  - Specifies the search patterns
- **`--after-context=N, -A N`**
  - Outputs N lines after each match line
- **`--with-filename, -H`**
  - Outputs the filename for each matching line

grep

```
# Outputs the file lines that include the string ”abc”

grep abc file.txt

# Outputs the file lines that include character ’l’
# followed by any other character,
# or include character ’a’

grep –e ”l.” –e a file.txt

# Outputs the file lines that include string ”abc”,
# and the next 4 lines, preceded by the filename

grep –H –A 4 abc file.txt
```