What Are Filters?

- A filter is a UNIX program that reads input (usually stdin), performs some transformation on it and writes it (usually to stdout).
- This follows the UNIX/Linux model of building simple components and then combining them to create more powerful applications.
  - We might use grep or tail to select some of our input, sort to sort it, wc to count characters and/or lines, etc.
Examples of Filters

• UNIX filters include:
  – grep – selects lines from standard input based on whether they contain a specified pattern. There are also egrep and fgrep.
  – sort – places lines of input in order
  – sed – "stream editor" – allows the user to perform certain specified transformation on the input.
  – awk – named for Alfred Aho, Peter Weinberger and Brian Kernighan, it offers much more power in transforming input than sed.

sort

• sort sorts lines of input in ASCII order.
• The user has a certain amount of control over which column is used as the sort key:
  – sort -f - fold upper and lower case together
  – sort -d - sorts by "dictionary order", ignores everything except blanks and alphanumerics
  – sort - n - sorts by numeric order
  – sort -o filename - places sorted output in filename
  – sort -k number - skip the first number columns
sort – Some Examples

ls | sort -f
   sort files in alphabetic order
ls -s | sort -n
   sort small files first
ls -s | sort -nr
   sort large files first
ls -l | sort -nrk5
   sort by byte count largest first
who | sort + 4n
   sort by login, oldest first
sort -f -u fleas
   sort by first field (ignore case)
   sort by first field (consider case)
   don't print duplicates

uniq

- uniq can do one of 4 different things:
  - Retain only duplicate lines
  - Retain only unique lines
  - Eliminate duplicate lines
  - Count how many duplicate lines there are.
- Syntax
  uniq [-cdu] [infile [outfile]]
  -c prefixes line with number of occurrences
  -d only print duplicate lines
  -u only print unique lines
uniq – An Example

[SIEGFRIE@panther ~]$ cat data
Barbara
Al
Al
Kathy
Barbara
[SIEGFRIE@panther ~]$ uniq -d data
Al
[SIEGFRIE@panther ~]$ uniq -u data
Barbara
Kathy
Barbara

[SIEGFRIE@panther ~]$ uniq data
Barbara
Al
Kathy
Barbara
[SIEGFRIE@panther ~]$ uniq -c data
 1 Barbara
 2 Al
 1 Kathy
 1 Barbara
[SIEGFRIE@panther ~]$


uniq and sort Together

[SIEGFRIE@panther ~]$ cat CS270
Dan
George
Alice
Roger
Stuart
Abigail
Steven

[SIEGFRIE@panther ~]$ cat CS271
Dan
Alice
Steven
Polly
Molly
Solly
Abigail

[SIEGFRIE@panther ~]$ sort CS270 CS271 | uniq -d
Abigail
Alice
Dan
Steven

[SIEGFRIE@panther ~]$ sort CS270 CS271 | uniq -u
George
Molly
Polly
Roger
Solly
Stuart


```
[SIEGFRIE@panther ~]$ sort CS270 CS271 | uniq
Abigail
Alice
Dan
George
Molly
Polly
Roger
Solly
Steven
Stuart
```

```
[SIEGFRIE@panther ~]$ sort CS270 CS271 | uniq -c
  2 Abigail
  2 Alice
  2 Dan
  1 George
  1 Molly
  1 Polly
  1 Roger
  1 Solly
  2 Steven
  1 Stuart
```

```
[SIEGFRIE@panther ~]$ sort CS270 CS271 | uniq -c
```
**comm**

- `comm file1 file2` – compare `file1` and `file2` line by line and prints the output in 3 columns:
  - lines appearing in `file1` only
  - lines appearing in `file2` only
  - lines appearing in both files
- `comm -1` suppress column 1
- `comm -2` suppress column 2
- `comm -3` suppress column 3

**comm – An Example**

```
[SIEGFRIE@panther ~]$ cat CS270
Dan
George
Alice
Roger
Stuart
Abigail
Steven

[SIEGFRIE@panther ~]$ cat CS271
Dan
Alice
Steven
Polly
Molly
Solly
Abigail
```
[SIEGFRIE@panther ~]$ comm -1 CS270 CS271
Dan
Alice
Steven
Polly
Molly
Solly
Abigail

[SIEGFRIE@panther ~]$ comm -2 CS270 CS271
Dan
George
Alice
Roger
Stuart
Abigail
Steven

[SIEGFRIE@panther ~]$ comm -3 CS270 CS271
Alice
George
Alice
Roger
    Steven
    Polly
    Molly
    Solly
    Abigail
Stuart
Abigail
Steven
[SIEGFRIE@panther ~]$
**tr**

- **tr** translates characters in a file
  - **tr** a-z A-Z maps lower-case letters into upper case.
  - **tr** A-Z a-z maps upper-case letters into lower case.
  - **tr** –c complements

---

**tr – An Example**

[SIEGFRIE@panther ~]$

`cat bin/sq`

`cat $* |`

`tr -sc A-Za-z '\012' |`

# Compress nonletters into newlines

`sort |`  # sort them

`uniq -c |`  # give a count

`sort -n |`  # sort by that count

`tail |`  # print the last term

`pr -5`  # in 5 columns

[SIEGFRIE@panther ~]$