Filters

Filters

Commands that are created to be used with a pipe are often called filters. These filters are very small programs that do one specific thing very efficiently. They can be used as building blocks.

When you pipe two commands, the "filtered " output of the first command is given to the next.

**💡Tips:**

* Pipes '|' send the output of one command as input of another command.
* The Filter takes input from one command, does some processing, and gives output.

cat, tee grep Commands

* cat

When between two pipes, the cat command does nothing (except putting stdin on stdout). Displays the text of the file line by line.

user@clarusway:~$ tac count.txt | cat | cat | cat | cat | cat

five

four

three

two

one

* tee

tee is almost the same as cat, except that it has two identical outputs.

user@clarusway:~$ tac count.txt | tee temp.txt | tac

one

two

three

four

five

user@clarusway:~$ cat temp.txt

five

four

three

two

one

* grep

The most common use of grep is to filter lines of text containing (or not containing) a certain string.

user@clarusway:~$ cat tennis.txt

Amelie Mauresmo, Fra

Justine Henin, BEL

Serena Williams, USA

Venus Williams, USA

user@clarusway:~$ cat tennis.txt | grep Williams

Serena Williams, USA

Venus Williams, USA

Q: Why does the tee command used for?  
A: The tee filter is used to send an output to **more than one destination**. It can send one copy of the output to a file and another to the screen (or some other program) if used with pipe (**|**).

 - Interview Q&A

## grep Command

The grep, which stands for "global regular expression print," is used to search text. It searches the given file for lines containing a match to the given strings or words.

**💡Tips:**

GREP :(Globally search a Regular Expression and Print).

### The syntax:

1

2

grep 'word' filename

This means search any line that contains the word in filename on Linux.

### Examples:

1

2

3

yourname@yourcomp:~$ grep 'understand' quotes.txt

I hear and I forget. I see and I remember. I do and I

    understand.

**💡Tips:**

The Major difference is FIND is for searching files and directories using [filters](https://lms.clarusway.com/mod/lesson/view.php?id=1987) while GREP is for searching a pattern inside a file or searching process(es)

| **Command** | **Description** |
| --- | --- |
| grep -i | Returns the results for case insensitive strings |
| grep -n | Returns the matching strings along with their line number |
| grep -v | Returns the result of lines not matching the search string |
| grep -c | Returns the number of lines in which the results matched the search string |

Q: Why the **grep** command used for?  
A: The grep command is used to search text. It searches the given file for lines containing a match to the given strings or words. It is one of the most useful commands on Linux and Unix-like system.

 - Interview Q&A

cut, tr, wc Commands

* cut

The cut filter can select columns from files, depending on a delimiter or a count of bytes.

clarusway@DESKTOP-UN6T2ES:~$ cat /etc/passwd | tail -3

aaron:x:1001:1001:aaron,,,:/home/aaron:/bin/bash

james:x:1005:1009:james,,,:/home/james:/bin/bash

walter:x:1006:1006:walter clarus:/home/walter:/bin/sh

clarusway@DESKTOP-UN6T2ES:~$ cut -d: -f1-3 /etc/passwd | tail

    -3

aaron:x:1001

james:x:1005

walter:x:1006

clarusway@DESKTOP-UN6T2ES:~$ cut -d: -f1,3 /etc/passwd | tail

    -3

aaron:1001

james:1005

walter:1006

clarusway@DESKTOP-UN6T2ES:~$

* d means delimiter. In the example above, colon (:) is used as a delimiter.
* f means field.
* tr

It is used for translating and deleting characters.

* To delete the characters.

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt

way to reinvent yourself

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt | tr -d e

way to rinvnt yourslf

clarusway@DESKTOUN6T2ES:~$

* To convert lower case to upper case

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt

way to reinvent yourself

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt | tr [a-z] [A

    -Z]

WAY TO REINVENT YOURSELF

clarusway@DESKTOP-UN6T2ES:~$

* To convert upper case to lower case

1

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt

WAY TO REINVENT YOURSELF

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt | tr [A-Z] [a

    -z]

way to reinvent yourself

clarusway@DESKTOP-UN6T2ES:~$

* To translate white-space to tabs

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt

WAY TO REINVENT YOURSELF

clarusway@DESKTOP-UN6T2ES:~$ cat clarusway.txt | tr [:space:]

    '\t'

WAY TO REINVENT YOURSELF

Counting words, lines and characters is easy with wc.

user@clarusway~$ wc tennis.txt

5 15 100 tennis.txt

user@clarusway~$ wc -l tennis.txt

5 tennis.txt

user@clarusway~$ wc -w tennis.txt

15 tennis.txt

user@clarusway~$ wc -c tennis.txt

100 tennis.txt

sort, uniq, comm Command

* sort

The sort filter will default to an alphabetical sort.

user@clarusway:~$ cat music.txt

Queen

Brel

Led Zeppelin

Abba

user@clarusway:~$ sort music.txt

Abba

Brel

Led Zeppelin

Queen

| **Command** | **Description** |
| --- | --- |
| sort -r | the flag returns the results in reverse order |
| sort -f | the flag does case insensitive sorting |
| sort -n | the flag returns the results as per numerical order |

* uniq

With uniq you can remove duplicates from a sorted list.

user@clarusway:~$ cat music.txt

Queen

Brel

Queen

Abba

user@clarusway:~$ sort music.txt

Abba

Brel

Queen

Queen

user@clarusway:~$ sort music.txt |uniq

Abba

Brel

Queen

* comm

Comparing streams (or files) can be done with the comm. By default, comm will output three columns.

* column 1 lines unique to list1
* column 2 lines unique to list2
* column 3 lines that appear in both files

clarusway@DESKTOP-UN6T2ES:~$ cat list1.txt

Abba

Bowie

Cure

Queen

Sweet

clarusway@DESKTOP-UN6T2ES:~$ cat list2.txt

Abba

Cure

Queen

Turner

clarusway@DESKTOP-UN6T2ES:~$ comm list1.txt list2.txt

Abba

Bowie

Cure

Queen

Sweet

Turner

clarusway@DESKTOP-UN6T2ES:~$