Text tools

cat **-** concatenate files and print

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ cat < test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

Create a new file named new_test_file and add text

```
eureka@ubuntu:~/Desktop$ cat > new_test_file
hello I'm a test file
```

```
Ctrl + d
```

```
eureka@ubuntu:~/Desktop$ cat new_test_file
hello I'm a test file
```

tac- concatenate and print files in reverse

```
eureka@ubuntu:~$ tac test_file
4,SAO,a
2,LON,b
3,CHI,a
2,CHI*,c
3,TOK,a
2,LON,c
1,CHI*,b
day,city,host
```

echo - display output

-n no trailing newline

-e interpret backslash escapes

```
eureka@ubuntu:~$ echo "hello\nworld"
hello\nworld
```

```
eureka@ubuntu:~$ echo -e "hello\nworld"
hello
world
```

```
eureka@ubuntu:~$ echo -en "hello\nworld"
hello
worldeureka@ubuntu:~$
```

add newline to end of file -> echo >> filename

```
eureka@ubuntu:~$ cat new_test_file
hello I'm a test file
```

```
eureka@ubuntu:~$ echo "I'm a new line" >>  new_test_file
```

```
eureka@ubuntu:~$ cat new_test_file
hello I'm a test file
I'm a new line
```

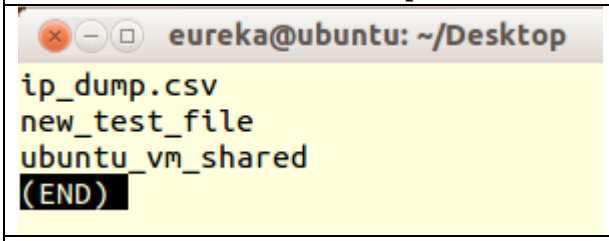
printf – print and format

less – view but not change the contents of a file. Nice when you want to read an important file's contents without accidentally fat fingering an edit on the file.

ex: less FILE

pipe commands to less

```
eureka@ubuntu:~/Desktop$ ls | less
```



'q' to exit

Sort – sort line of text files

-u unique

-t file separator

-k specifies sort field

-s disables 'last-resort' sorting, which sorts on everything that wasn't part of a specified key

-n is for sorting numerically

Sort comma separated file by column 6 and how only unique lines

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
```

```
eureka@ubuntu:~$ sort -u -t, -k1
-s test_file
1,CHI,b
2,CHI,c
2,LON,c
```

```
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,a
```

```
3,CHI,a
3,TOK,a
4,SAO,a
day,city,host
```

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,a
```

```
eureka@ubuntu:~$ sort -u -t, -k1
-s -n test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
4,SAO,a
```

find duplicate rows in file

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,a
```

```
eureka@ubuntu:~$ sort test_file|
uniq -d
2,LON,c
```

cut - display only specific columns from a text file or other command outputs

-d delimiter

-f fields

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,      a
```

```
eureka@ubuntu:~$ cut -d "," -f 1
test_file
day
1
2
3
2
3
2
4
```

Use tab as a delimiter

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,      a
```

```
eureka@ubuntu:~$ cut -d "       " -f 1 test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,
```

```
To input tab -> cut -d "<CTR>v <TAB>"
```

paste – join files horizontally

```
eureka@ubuntu:~$ cat number_file
1
2
3
4,a
```

```
eureka@ubuntu:~$ cat name_file
Bjarne
Linus
Guido
```

```
eureka@ubuntu:~$ paste number_file name_file
1     Bjarne
2     Linus
3     Guido
4
```

colrm – column remove

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ colrm 3 < test_file
da
1,
2,
3,
2,
3,
2,
4,
```

join - merges the lines of two sorted text files based on the presence of a common field

```
eureka@ubuntu:~$ cat test_file
1 CHI*,b
2 LON,c
3 TOK,a
4 SAO,a
```

```
eureka@ubuntu:~$ cat other_test_file
1 yes
2 no
3 yes
4 no
```

```
eureka@ubuntu:~$ join test_file other_test_file
1 CHI*,b yes
2 LON,c no
3 TOK,a yes
4 SAO,a no
```

expand - convert tab to spaces.

-t set tab stops

```
eureka@ubuntu:~$ cat test_file
1	CHI*,b
2	LON,c
3	TOK,a
4	SAO,a
```

```
eureka@ubuntu:~$ expand -t 1 test_file
1 CHI*,b
2 LON,c
3 TOK,a
4 SAO,a
```

unexpand - convert spaces into tabs. Must have at least 2 spaces

-t set tab stops

```
eureka@ubuntu:~$ cat test_file
1  CHI*,b
2  LON,c
3  TOK,a
4  SAO,a
```

```
eureka@ubuntu:~$ unexpand -t 2 test_file
1	 CHI*,b
2	 LON,c
3	 TOK,a
4	 SAO,a
```

grep - searches plain-text data sets for lines matching a regular expression

grep can return exit code 1 even if it ran successfully. 0 means it found something, 1 means it found nothing

-F disable regex

```
eureka@ubuntu:~$ cat test_file
day,city,host
```

```
eureka@ubuntu:~$ grep -F "CHI*"
test_file
```

1,CHI*,b 2,LON,c 3,TOK,a 2,CHI*,c 3,CHI,a 2,LON,b 4,SAO,a	1,CHI*,b 2,CHI*,c

eureka@ubuntu:~$ cat test_file day,city,host 1,CHI*,b 2,LON,c 3,TOK,a 2,CHI*,c 3,CHI,a 2,LON,b 4,SAO,a	eureka@ubuntu:~$ grep "CHI*" test_file 1,CHI*,b 2,CHI*,c

eureka@ubuntu:~$ cat test_file day,city,host 1,CHI*,b 2,LON,c 3,TOK,a 2,CHI*,c 3,CHI,a 2,LON,b 4,SAO,a	eureka@ubuntu:~$ grep "CHI*" test_file 1,CHI*,b 2,CHI*,c 3,CHI,a

-H show file name

-n show line number

eureka@ubuntu:~$ cat test_file day,city,host 1,CHI*,b 2,LON,c 3,TOK,a 2,CHI*,c 3,CHI,a 2,LON,b 4,SAO,a	eureka@ubuntu:~$ grep -Hn '[[:blank:]]' test_file test_file:6: 2,CHI*,c test_file:7: 3,CHI,a

-w matches whole word only

eureka@ubuntu:~$ cat test_file day,city,host 1,CHI*,b 2,LON,c 3,TOK,a	eureka@ubuntu:~$ grep -w CH test_file eureka@ubuntu:~$	eureka@ubuntu:~$ grep -w CHI test_file 1,CHI*,b 2,CHI*,c 3,CHI,a

2,CHI*,c 3,CHI,a 2,LON,b 4,SAO,a		

-m stop searching after first one is found

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ grep -m 1 CHI
test_file
1,CHI*,b
```

wc - Print Line, Word, and Byte Counts

-l count lines

```
eureka@ubuntu:~$ wc new_test_file
  6   60 426 new_test_file
```

```
eureka@ubuntu:~$ wc -l new_test_file
6 new_test_file
```

tr - Translate/transliterate characters from standard input, writing to standard output. use quoting and/or brackets, as appropriate.

Great article on tr http://www.linuxjournal.com/article/2563

-d delete

-c complement

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI,b
2,LON,c
3,TOK,a
2,CHI,c
3,CHI,a
2,LON,c
4,SAO,a
```

```
eureka@ubuntu:~$
tr -d '3' <
test_file >
new_test
```

```
eureka@ubuntu:~$
cat new_test
day,city,host
1,CHI,b
2,LON,c
,TOK,a
2,CHI,c
,CHI,a
2,LON,c
4,SAO,a
```

Count how many times * appears in a text file – notice the escaped '*'. tr deletes all characters which are not '*' in the character set

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ tr -d -c \* <
test_file |wc -c
2
```

tail – shows last part of file

-n number of lines to display

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ tail -5
test_file
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

head - displays beginning of file – default 10

-n number of lines to display

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ head -5
test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
```

diff- Displays two files and prints the lines that are different

-y side-by-side

-w ignore white space

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
```

```
eureka@ubuntu:~$ cat
other_test_file
day,city,host
```

```
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
I'm an extra line
```

```
eureka@ubuntu:~$ diff test_file other_test_file
8a9
> I'm an extra line
```

```
eureka@ubuntu:~$ diff -y test_file other_test_file
day,city,host                                           day,city,host
1,CHI*,b                                              1,CHI*,b
2,LON,c                                                     2,LON,c
3,TOK,a                                                     3,TOK,a
2,CHI*,c                                              2,CHI*,c
3,CHI,a                                                     3,CHI,a
2,LON,b                                                     2,LON,b
4,SAO,a                                                     4,SAO,a
                                                       >    I'm an extra line
```

comm - compare two sorted files line by line. produces three columns of output - lines only in file1 (column 1), lines only in file2 (column 2), and lines common to both files (column 3).

```
eureka@ubuntu:~$ cat test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
```

```
eureka@ubuntu:~$ cat
other_test_file
day,city,host
1,CHI*,b
2,LON,c
3,TOK,a
2,CHI*,c
3,CHI,a
2,LON,b
4,SAO,a
I'm an extra line
```

```
eureka@ubuntu:~$ comm test_file other_test_file
          day,city,host
          1,CHI*,b
          2,LON,c
          3,TOK,a
          2,CHI*,c
          3,CHI,a
          2,LON,b
          4,SAO,a
    I'm an extra line
```