COSMOGIC

LINUX

Redirection

The coolest feature of command line is input/output redirection .
With this we can redirect input and output of command from and to a file .
We can also connect multiple command to make powerful command pipelines .

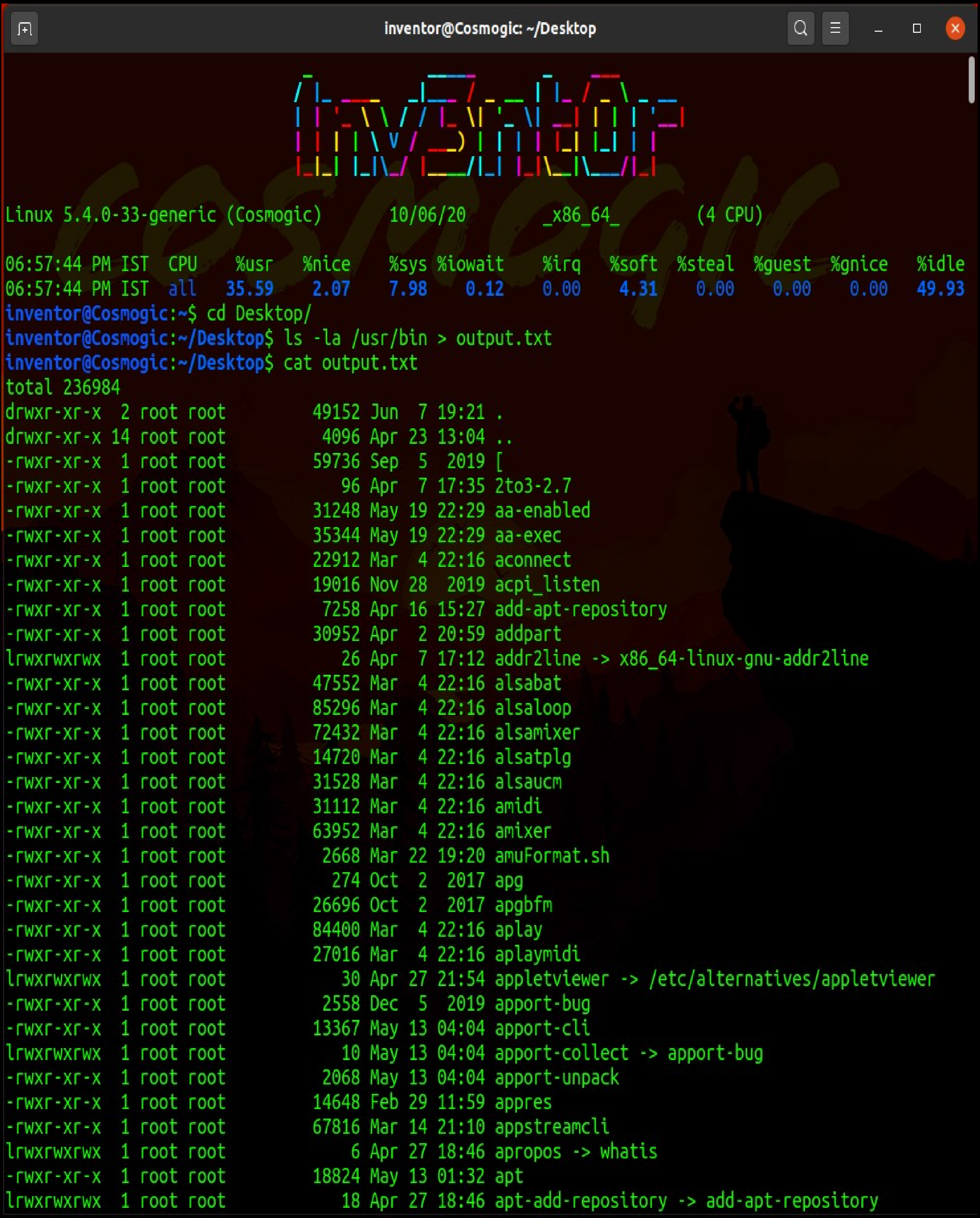
Some basic command are :

- echo
- cat
- sort
- uniq
- wc
- grep
- tee

From previous pdf's we know how to create a file and how to use ls command .
Using redirection we can store any output in file for example

```
ls -la /usr/bin > output.txt
```

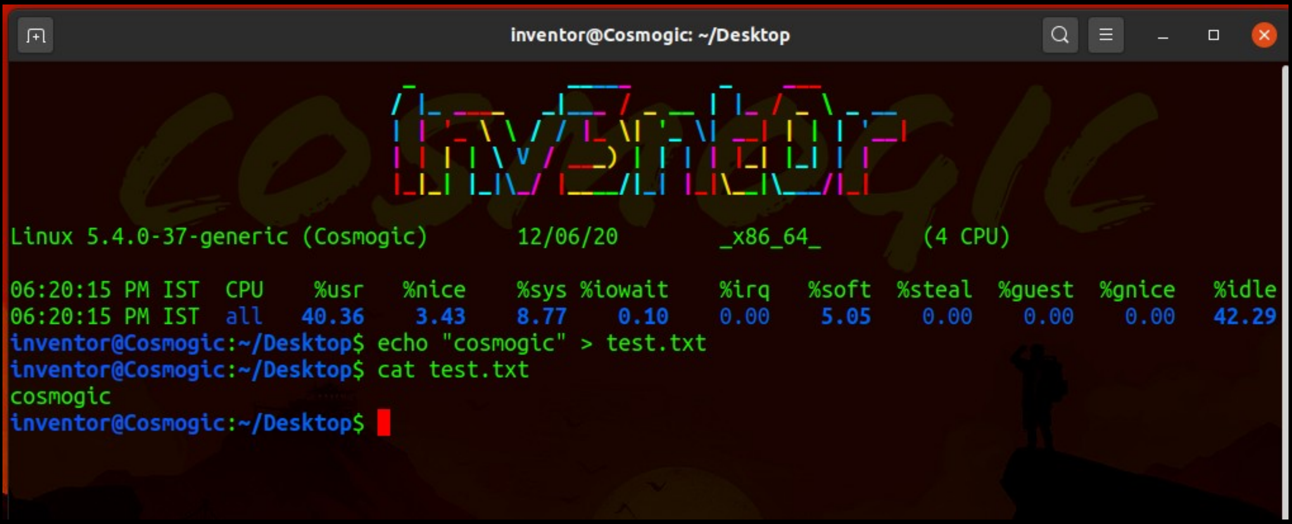
this will show all the output in a text file
and we can do this with any command



```
Linux 5.4.0-33-generic (Cosmogic)       10/06/20        _x86_64_        (4 CPU)

06:57:44 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:57:44 PM IST  all   35.59    2.07    7.98    0.12    0.00    4.31    0.00    0.00    0.00   49.93
inventor@Cosmogic:~$ cd Desktop/
inventor@Cosmogic:~/Desktop$ ls -la /usr/bin > output.txt
inventor@Cosmogic:~/Desktop$ cat output.txt
total 236984
drwxr-xr-x  2 root root         49152 Jun  7 19:21 .
drwxr-xr-x 14 root root          4096 Apr 23 13:04 ..
-rwxr-xr-x  1 root root         59736 Sep  5  2019 [
-rwxr-xr-x  1 root root            96 Apr  7 17:35 2to3-2.7
-rwxr-xr-x  1 root root         31248 May 19 22:29 aa-enabled
-rwxr-xr-x  1 root root         35344 May 19 22:29 aa-exec
-rwxr-xr-x  1 root root         22912 Mar  4 22:16 aconnect
-rwxr-xr-x  1 root root         19016 Nov 28  2019 acpi_listen
-rwxr-xr-x  1 root root          7258 Apr 16 15:27 add-apt-repository
-rwxr-xr-x  1 root root         30952 Apr  2 20:59 addpart
lrwxrwxrwx  1 root root            26 Apr  7 17:12 addr2line -> x86_64-linux-gnu-addr2line
-rwxr-xr-x  1 root root         47552 Mar  4 22:16 alsabat
-rwxr-xr-x  1 root root         85296 Mar  4 22:16 alsaloop
-rwxr-xr-x  1 root root         72432 Mar  4 22:16 alsamixer
-rwxr-xr-x  1 root root         14720 Mar  4 22:16 alsatplg
-rwxr-xr-x  1 root root         31528 Mar  4 22:16 alsaucm
-rwxr-xr-x  1 root root         31112 Mar  4 22:16 amidi
-rwxr-xr-x  1 root root         63952 Mar  4 22:16 amixer
-rwxr-xr-x  1 root root          2668 Mar 22 19:20 amuFormat.sh
-rwxr-xr-x  1 root root           274 Oct  2  2017 apg
-rwxr-xr-x  1 root root         26696 Oct  2  2017 apgbfm
-rwxr-xr-x  1 root root         84400 Mar  4 22:16 aplay
-rwxr-xr-x  1 root root         27016 Mar  4 22:16 aplaymidi
lrwxrwxrwx  1 root root            30 Apr 27 21:54 appletviewer -> /etc/alternatives/appletviewer
-rwxr-xr-x  1 root root          2558 Dec  5  2019 apport-bug
-rwxr-xr-x  1 root root         13367 May 13 04:04 apport-cli
lrwxrwxrwx  1 root root            10 May 13 04:04 apport-collect -> apport-bug
-rwxr-xr-x  1 root root          2068 May 13 04:04 apport-unpack
-rwxr-xr-x  1 root root         14648 Feb 29 11:59 appres
-rwxr-xr-x  1 root root         67816 Mar 14 21:10 appstreamcli
lrwxrwxrwx  1 root root             6 Apr 27 18:46 apropos -> whatis
-rwxr-xr-x  1 root root         18824 May 13 01:32 apt
lrwxrwxrwx  1 root root            18 Apr 27 18:46 apt-add-repository -> add-apt-repository
```

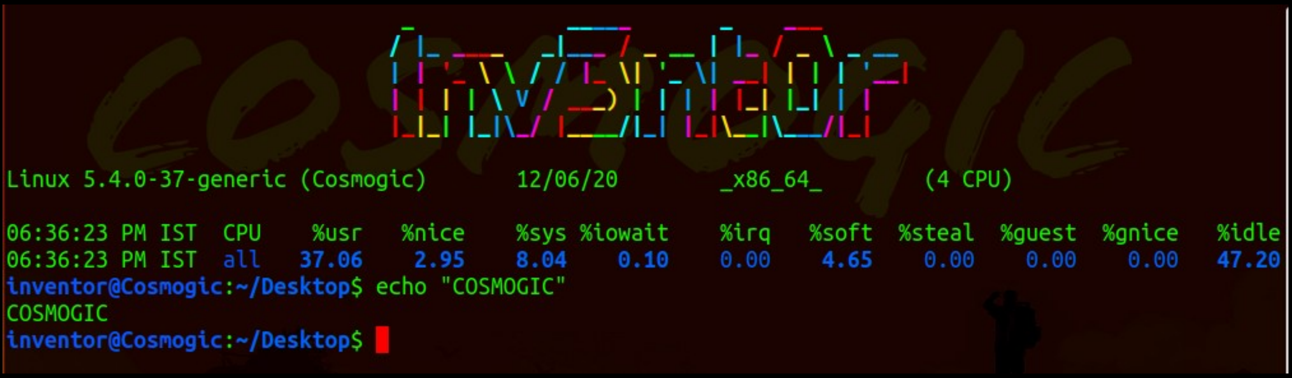
In the same way if we want to create a txt file by giving some input we can use echo command (with redirection) .

echo "Cosmogic" > test.txt



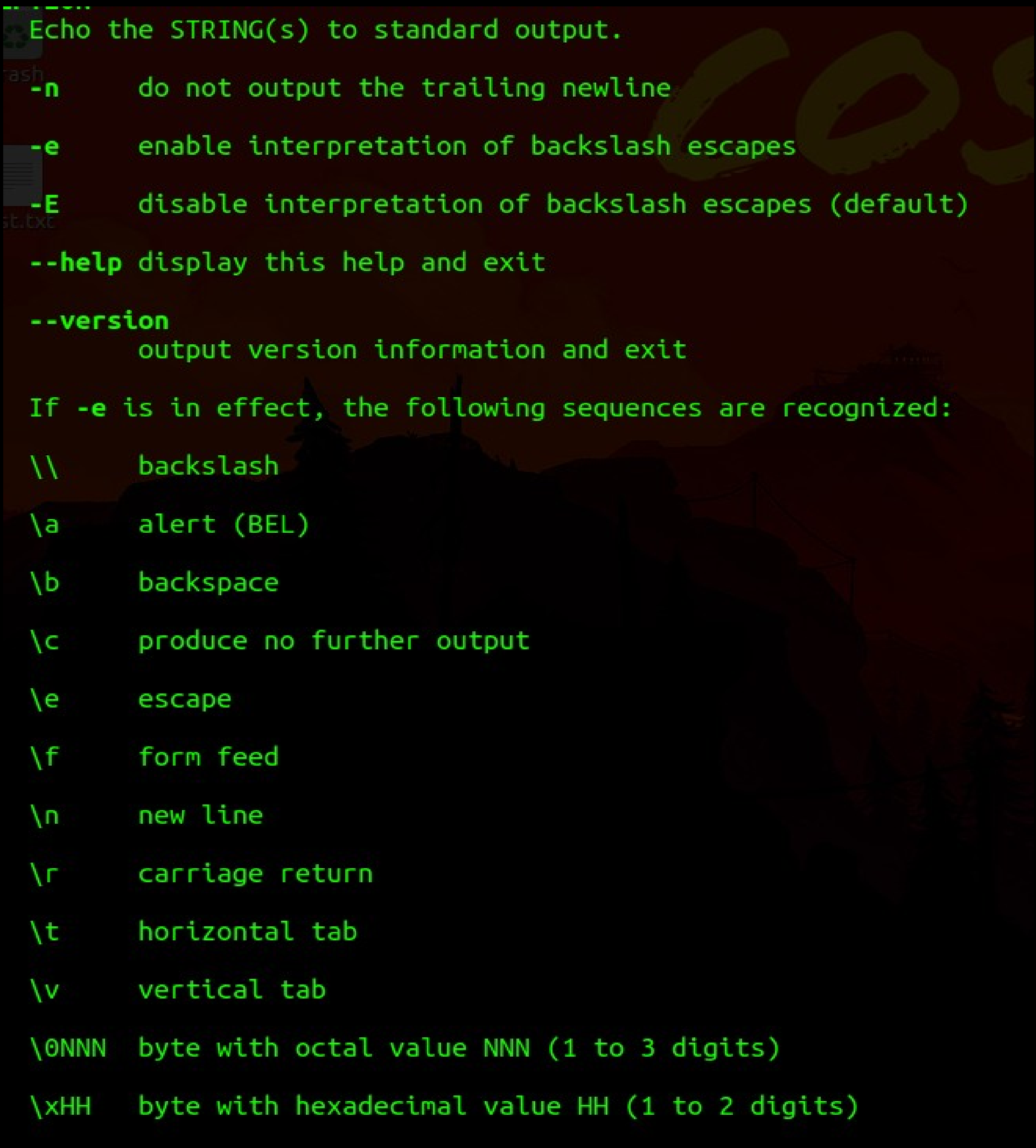
```
Linux 5.4.0-37-generic (Cosmogic)       12/06/20        _x86_64_        (4 CPU)

06:20:15 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:20:15 PM IST  all   40.36    3.43    8.77    0.10    0.00    5.05    0.00    0.00    0.00   42.29
inventor@Cosmogic:~/Desktop$ echo "cosmogic" > test.txt
inventor@Cosmogic:~/Desktop$ cat test.txt
cosmogic
inventor@Cosmogic:~/Desktop$ 
```

talking about echo its a builtin shell command used to display text that are passed as an argument.
If we just write echo with any text it will just display the same text



```
Linux 5.4.0-37-generic (Cosmogic)       12/06/20        _x86_64_        (4 CPU)

06:36:23 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:36:23 PM IST  all   37.06    2.95    8.04    0.10    0.00    4.65    0.00    0.00    0.00   47.20
inventor@Cosmogic:~/Desktop$ echo "COSMOGIC"
COSMOGIC
inventor@Cosmogic:~/Desktop$ 
```

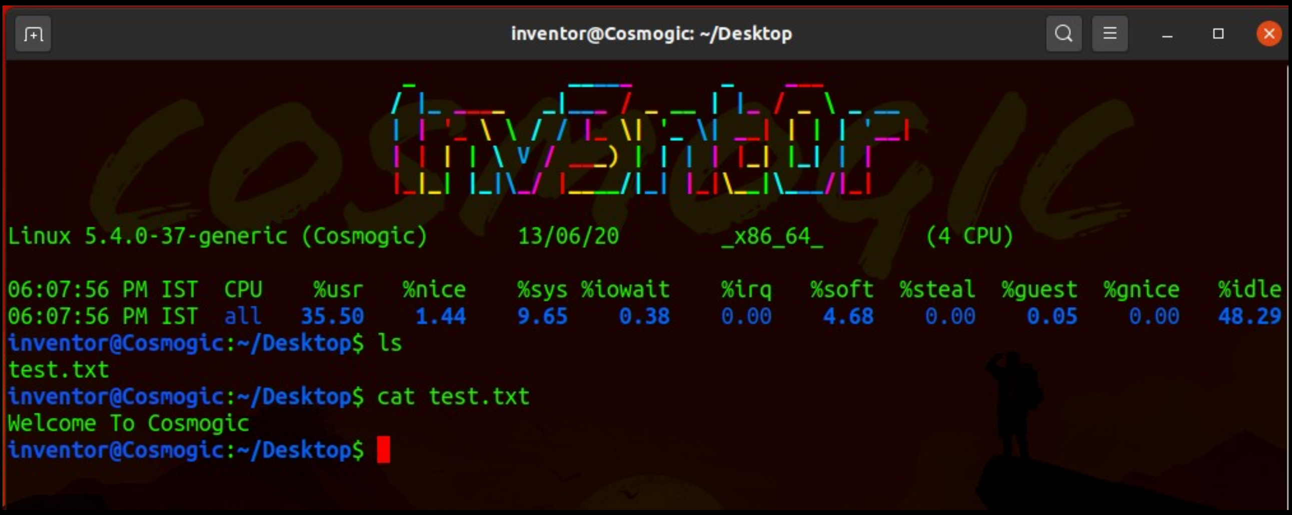
Options available with echo command are



```
Echo the STRING(s) to standard output.

-n     do not output the trailing newline

-e     enable interpretation of backslash escapes

-E     disable interpretation of backslash escapes (default)

--help display this help and exit

--version
       output version information and exit

If -e is in effect, the following sequences are recognized:

\\     backslash

\a     alert (BEL)

\b     backspace

\c     produce no further output

\e     escape

\f     form feed

\n     new line

\r     carriage return

\t     horizontal tab

\v     vertical tab

\0NNN  byte with octal value NNN (1 to 3 digits)

\xHH   byte with hexadecimal value HH (1 to 2 digits)
```

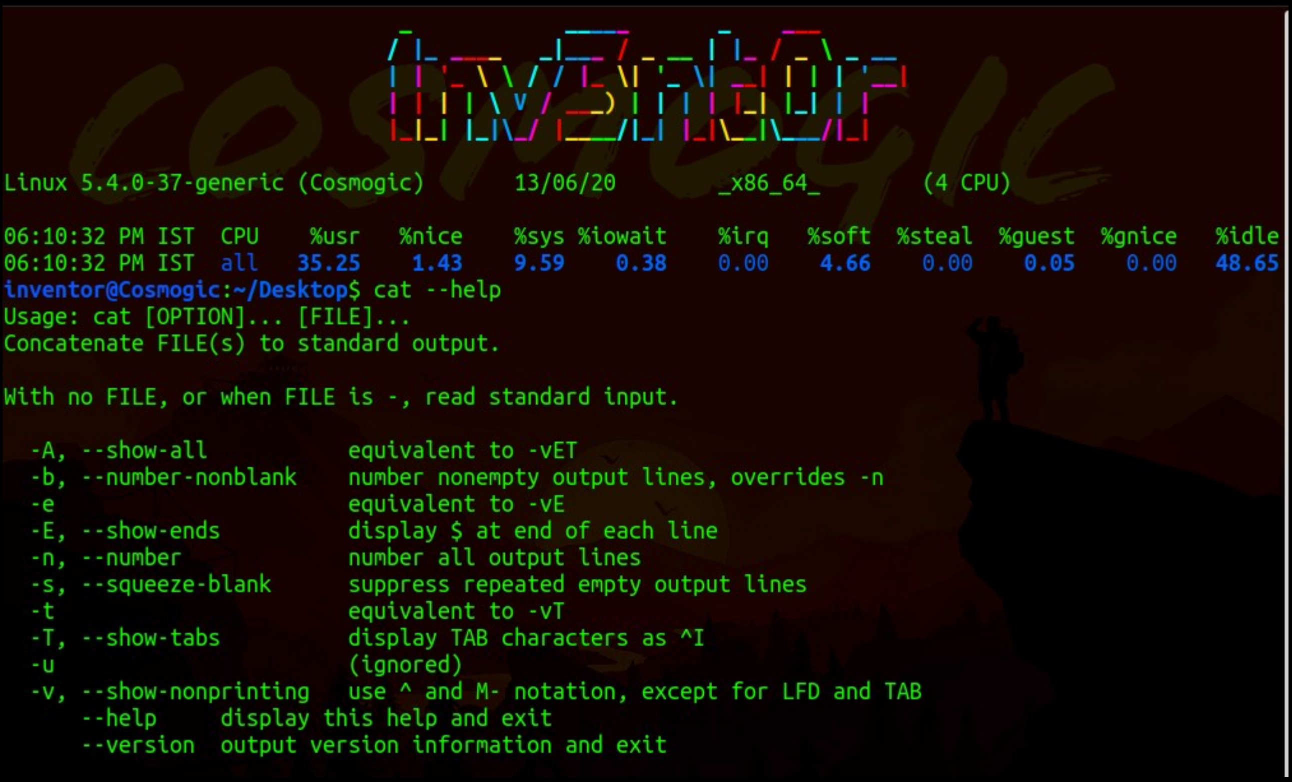
you can try all different combination to get results for ex :- echo -e "Cosmogic \bBeyond \bLimits " This will remove space from the string and CosmogicBeyondLimits will be printed .

CAT :- cat is used to concatenate files . If we wish to see the content of a file we write
cat (name of the file)



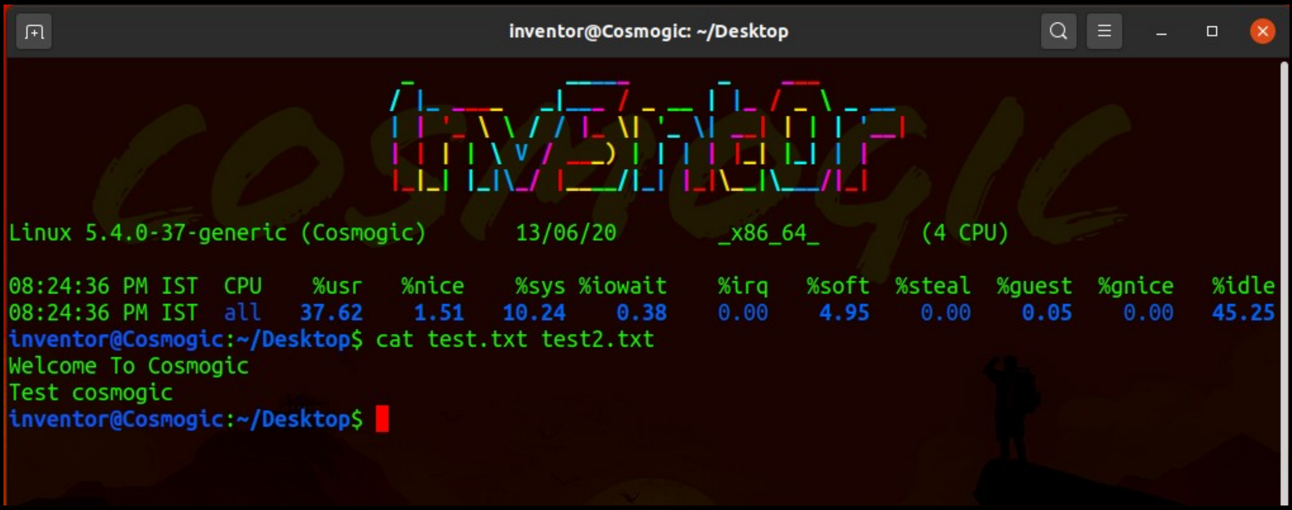
```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

06:07:56 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:07:56 PM IST  all   35.50    1.44    9.65    0.38    0.00    4.68    0.00    0.05    0.00   48.29
inventor@Cosmogic:~/Desktop$ ls
test.txt
inventor@Cosmogic:~/Desktop$ cat test.txt
Welcome To Cosmogic
inventor@Cosmogic:~/Desktop$
```

Options available in cat command are :



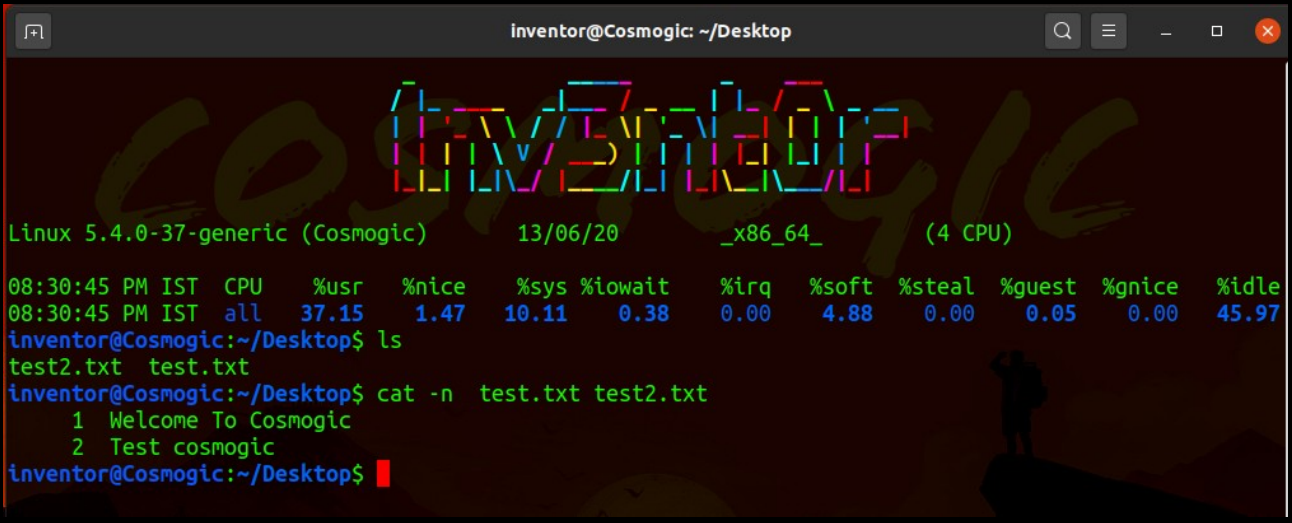
```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

06:10:32 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:10:32 PM IST  all   35.25    1.43    9.59    0.38    0.00    4.66    0.00    0.05    0.00   48.65
inventor@Cosmogic:~/Desktop$ cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit
```

we can also cat two file to get the combined results
cat first file second file



```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

08:24:36 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:24:36 PM IST  all   37.62    1.51   10.24    0.38    0.00    4.95    0.00    0.05    0.00   45.25
inventor@Cosmogic:~/Desktop$ cat test.txt test2.txt
Welcome To Cosmogic
Test cosmogic
inventor@Cosmogic:~/Desktop$
```

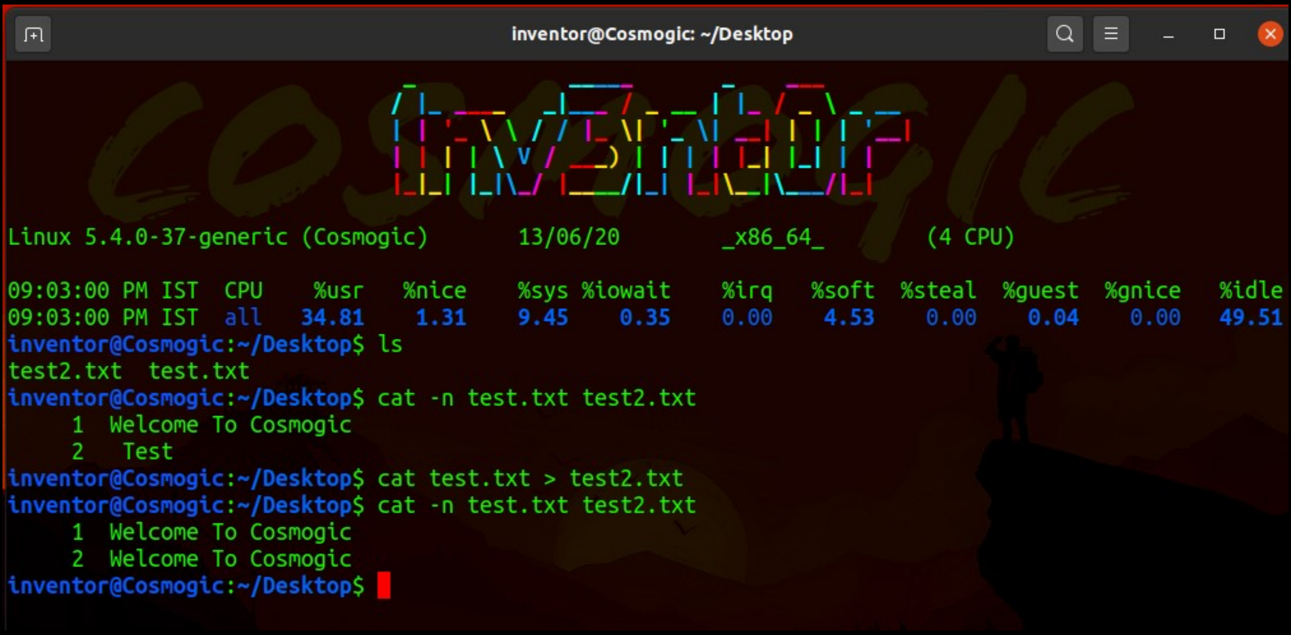
to get the linear output we can use -n in the command
cat -n file1 file2



```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

08:30:45 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:30:45 PM IST  all   37.15    1.47   10.11    0.38    0.00    4.88    0.00    0.05    0.00   45.97
inventor@Cosmogic:~/Desktop$ ls
test2.txt  test.txt
inventor@Cosmogic:~/Desktop$ cat -n  test.txt test2.txt
     1  Welcome To Cosmogic
     2  Test cosmogic
inventor@Cosmogic:~/Desktop$
```

we can also copy content of one file to another with help of cat and redirection

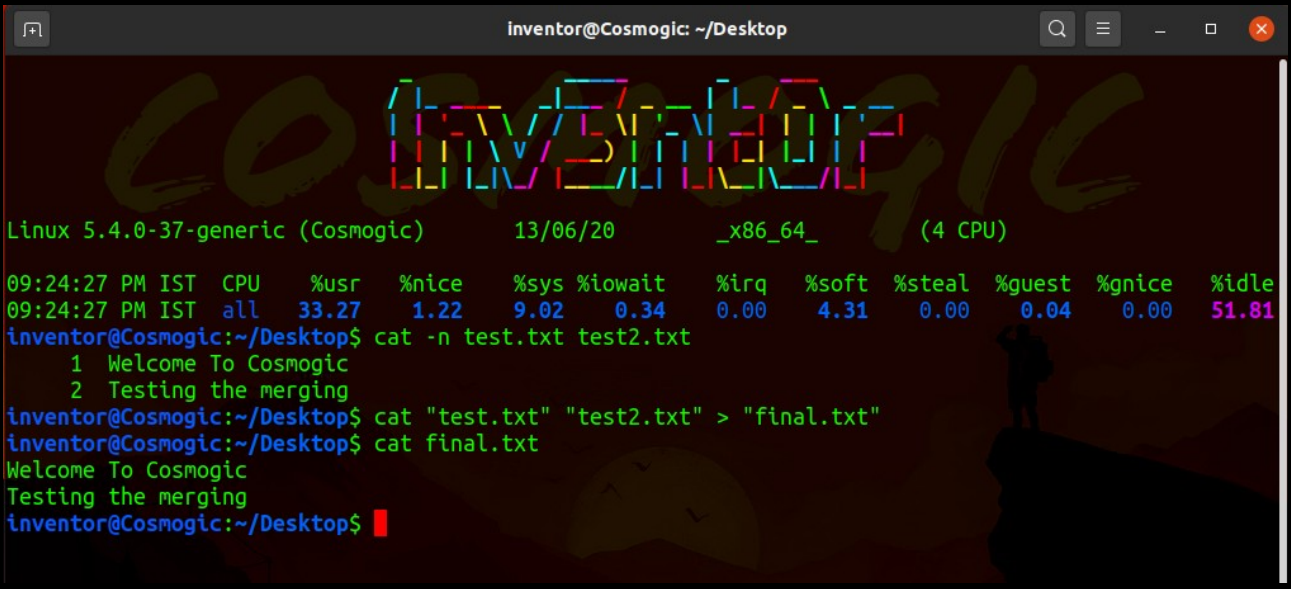
cat source file > destination file



```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

09:03:00 PM IST  CPU     %usr    %nice     %sys %iowait    %irq    %soft  %steal  %guest  %gnice   %idle
09:03:00 PM IST  all    34.81     1.31     9.45    0.35    0.00     4.53    0.00    0.04    0.00   49.51
inventor@Cosmogic:~/Desktop$ ls
test2.txt  test.txt
inventor@Cosmogic:~/Desktop$ cat -n test.txt test2.txt
     1  Welcome To Cosmogic
     2   Test
inventor@Cosmogic:~/Desktop$ cat test.txt > test2.txt
inventor@Cosmogic:~/Desktop$ cat -n test.txt test2.txt
     1  Welcome To Cosmogic
     2  Welcome To Cosmogic
inventor@Cosmogic:~/Desktop$
```

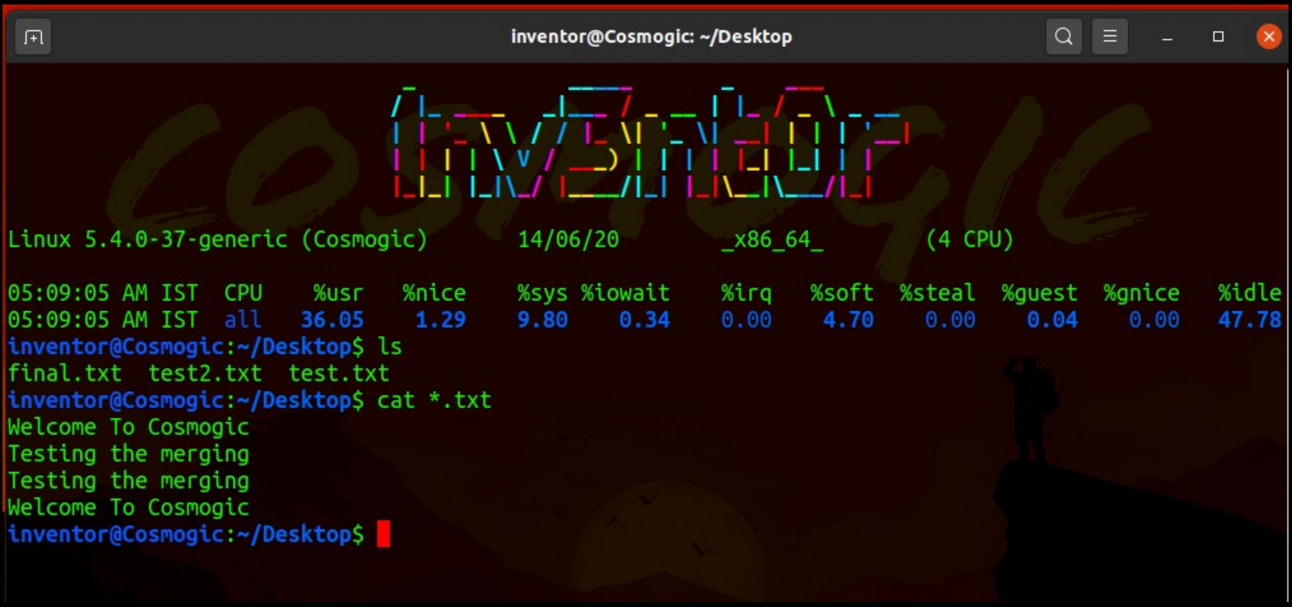
In the same way if we want to merge data of multiple files we can do :

cat "file 1" "file 2" "file 3" > "final file"



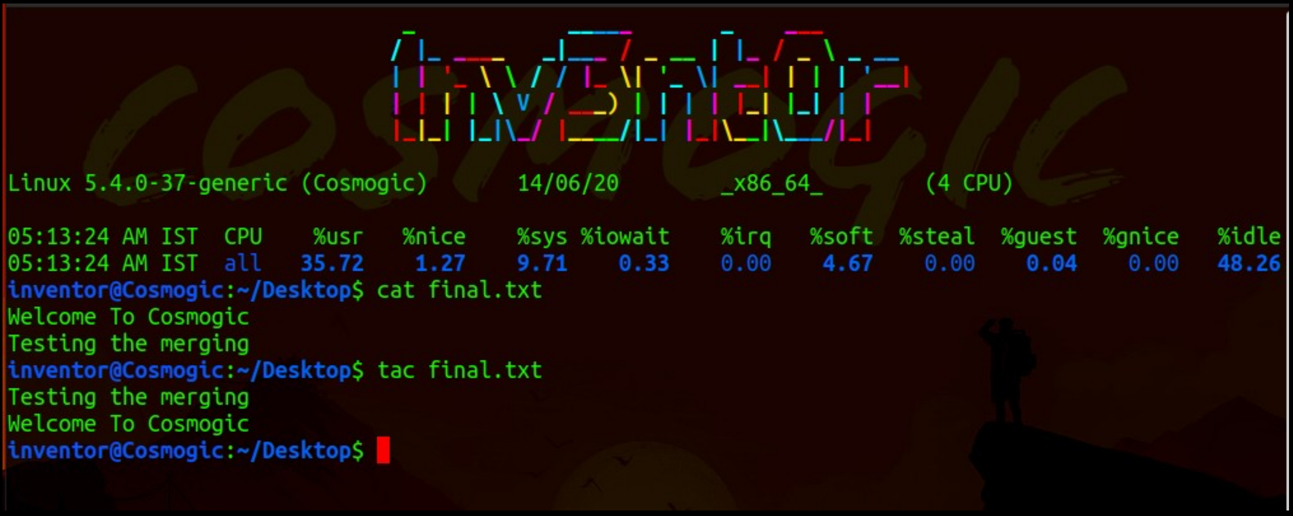
```
Linux 5.4.0-37-generic (Cosmogic)       13/06/20        _x86_64_        (4 CPU)

09:24:27 PM IST  CPU     %usr    %nice     %sys %iowait    %irq    %soft  %steal  %guest  %gnice   %idle
09:24:27 PM IST  all    33.27     1.22     9.02    0.34    0.00     4.31    0.00    0.04    0.00   51.81
inventor@Cosmogic:~/Desktop$ cat -n test.txt test2.txt
     1  Welcome To Cosmogic
     2  Testing the merging
inventor@Cosmogic:~/Desktop$ cat "test.txt" "test2.txt" > "final.txt"
inventor@Cosmogic:~/Desktop$ cat final.txt
Welcome To Cosmogic
Testing the merging
inventor@Cosmogic:~/Desktop$
```

If we want to view the content of some type of files for ex .txt we can write
cat *.txt   It will display all the content of files having .txt extension



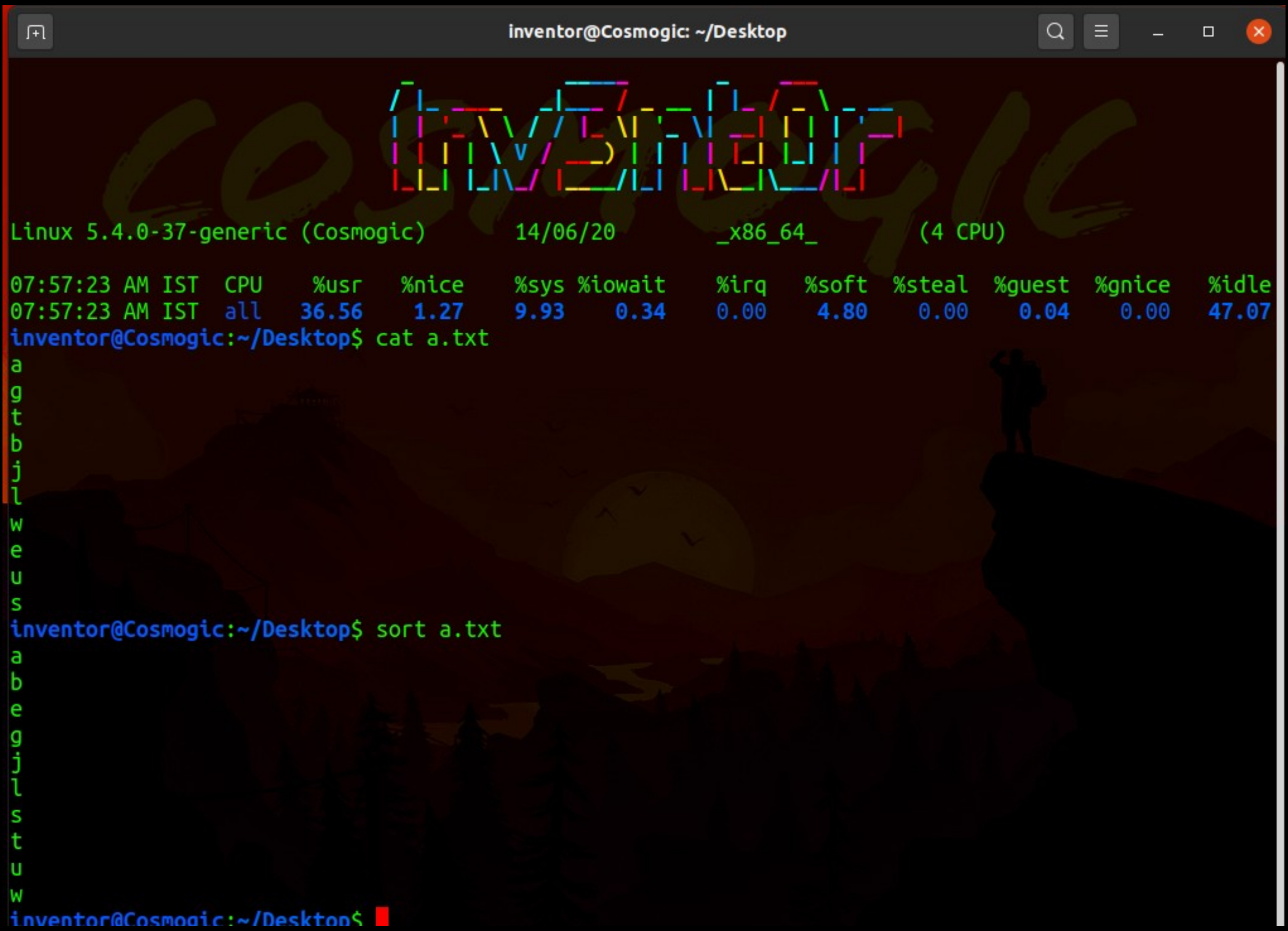
```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

05:09:05 AM IST  CPU     %usr    %nice     %sys %iowait    %irq    %soft   %steal   %guest   %gnice    %idle
05:09:05 AM IST  all    36.05     1.29     9.80    0.34    0.00     4.70     0.00     0.04     0.00    47.78
inventor@Cosmogic:~/Desktop$ ls
final.txt  test2.txt  test.txt
inventor@Cosmogic:~/Desktop$ cat *.txt
Welcome To Cosmogic
Testing the merging
Testing the merging
Welcome To Cosmogic
inventor@Cosmogic:~/Desktop$ 
```

we can also get output in reverse order using tac command
tac filename



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

05:13:24 AM IST  CPU     %usr    %nice     %sys %iowait    %irq    %soft   %steal   %guest   %gnice    %idle
05:13:24 AM IST  all    35.72     1.27     9.71    0.33    0.00     4.67     0.00     0.04     0.00    48.26
inventor@Cosmogic:~/Desktop$ cat final.txt
Welcome To Cosmogic
Testing the merging
inventor@Cosmogic:~/Desktop$ tac final.txt
Testing the merging
Welcome To Cosmogic
inventor@Cosmogic:~/Desktop$ 
```

Sort :- As it's name suggest it's used for shorting
for example :- sort filename



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

07:57:23 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
07:57:23 AM IST  all   36.56    1.27    9.93    0.34    0.00    4.80    0.00    0.04    0.00   47.07
inventor@Cosmogic:~/Desktop$ cat a.txt
a
g
t
b
j
l
w
e
u
s
inventor@Cosmogic:~/Desktop$ sort a.txt
a
b
e
g
j
l
s
t
u
w
inventor@Cosmogic:~/Desktop$
```

but this is not permanent if we will cat the file again we will get the unsorted result .

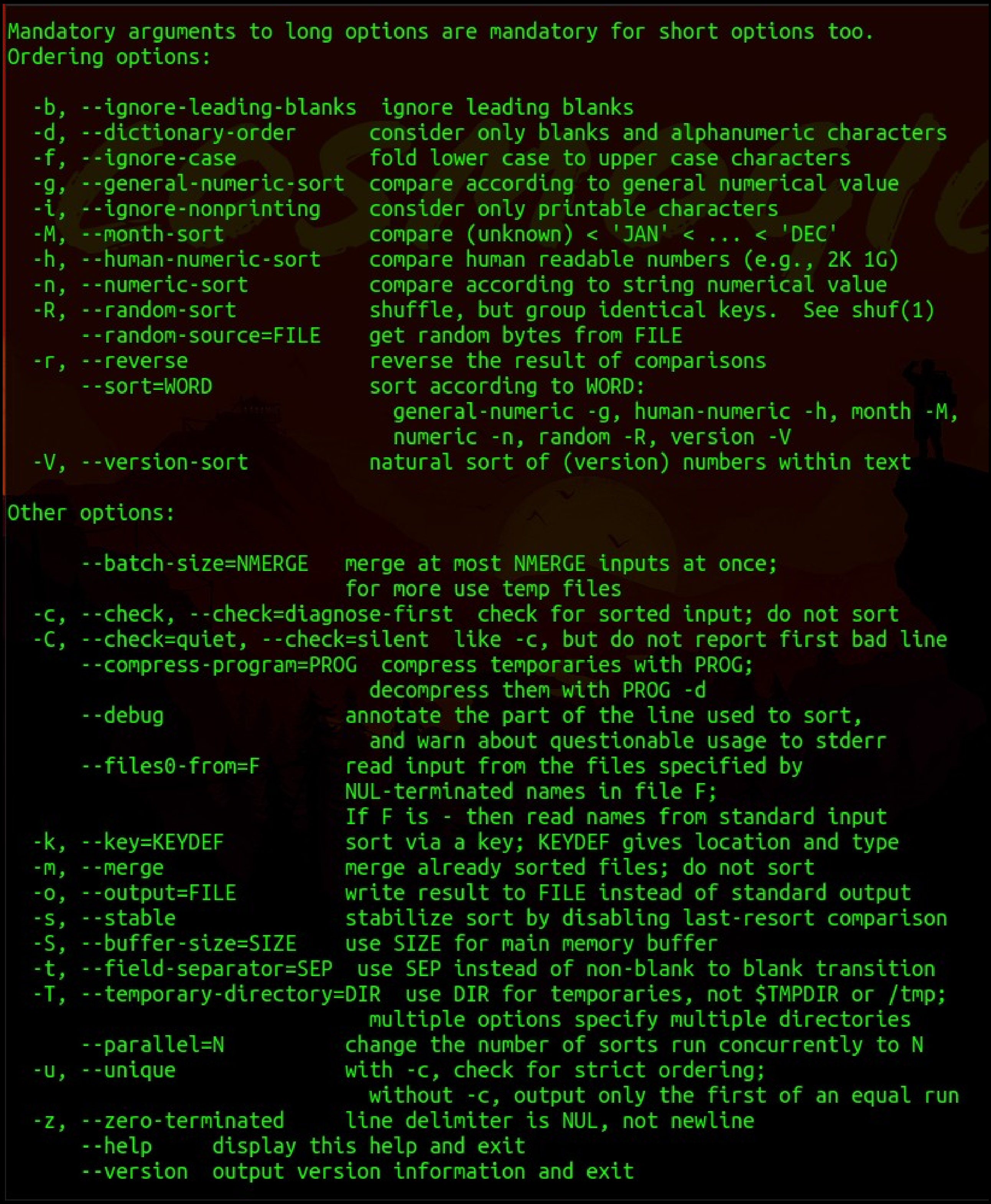
We can use redirection hear to save the sorted result in another file or -o can be used to achieve same results in case of -o command like :

(sort -o result file input file)

To get the sorted result in reverse order we can use -r with sort command like :

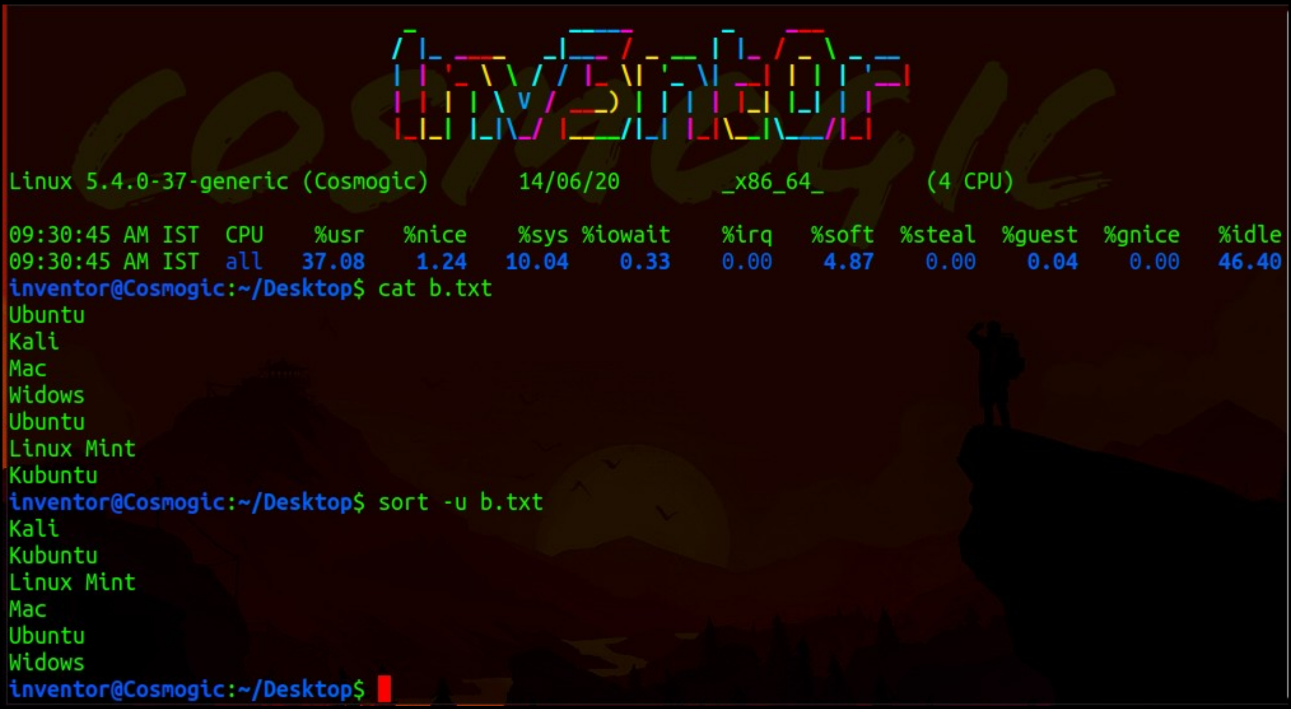
sort -r filename

Option available with sort command are :



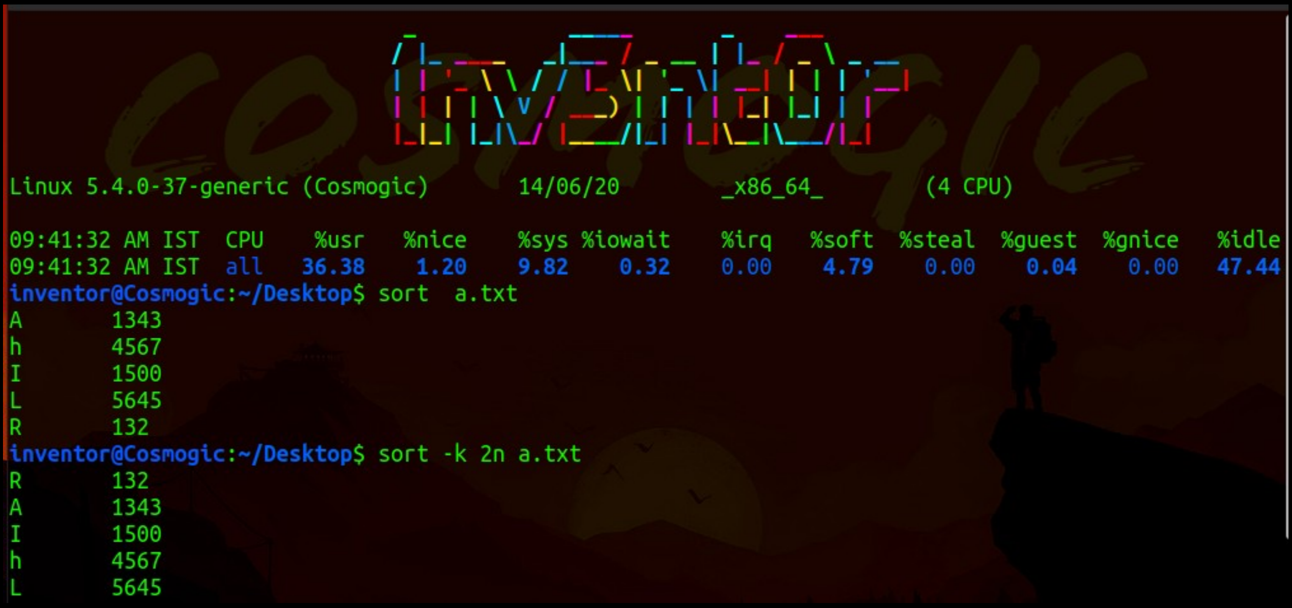
```
Mandatory arguments to long options are mandatory for short options too.
Ordering options:

  -b, --ignore-leading-blanks  ignore leading blanks
  -d, --dictionary-order      consider only blanks and alphanumeric characters
  -f, --ignore-case           fold lower case to upper case characters
  -g, --general-numeric-sort  compare according to general numerical value
  -i, --ignore-nonprinting    consider only printable characters
  -M, --month-sort            compare (unknown) < 'JAN' < ... < 'DEC'
  -h, --human-numeric-sort    compare human readable numbers (e.g., 2K 1G)
  -n, --numeric-sort          compare according to string numerical value
  -R, --random-sort           shuffle, but group identical keys.  See shuf(1)
      --random-source=FILE    get random bytes from FILE
  -r, --reverse               reverse the result of comparisons
      --sort=WORD             sort according to WORD:
                                general-numeric -g, human-numeric -h, month -M,
                                numeric -n, random -R, version -V
  -V, --version-sort          natural sort of (version) numbers within text

Other options:

      --batch-size=NMERGE   merge at most NMERGE inputs at once;
                            for more use temp files
  -c, --check, --check=diagnose-first  check for sorted input; do not sort
  -C, --check=quiet, --check=silent  like -c, but do not report first bad line
      --compress-program=PROG  compress temporaries with PROG;
                              decompress them with PROG -d
      --debug               annotate the part of the line used to sort,
                              and warn about questionable usage to stderr
      --files0-from=F       read input from the files specified by
                            NUL-terminated names in file F;
                            If F is - then read names from standard input
  -k, --key=KEYDEF          sort via a key; KEYDEF gives location and type
  -m, --merge               merge already sorted files; do not sort
  -o, --output=FILE         write result to FILE instead of standard output
  -s, --stable              stabilize sort by disabling last-resort comparison
  -S, --buffer-size=SIZE    use SIZE for main memory buffer
  -t, --field-separator=SEP  use SEP instead of non-blank to blank transition
  -T, --temporary-directory=DIR  use DIR for temporaries, not $TMPDIR or /tmp;
                              multiple options specify multiple directories
      --parallel=N          change the number of sorts run concurrently to N
  -u, --unique              with -c, check for strict ordering;
                              without -c, output only the first of an equal run
  -z, --zero-terminated     line delimiter is NUL, not newline
      --help     display this help and exit
      --version  output version information and exit
```

-u is used to remove duplicate from the list



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

09:30:45 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
09:30:45 AM IST  all   37.08    1.24   10.04    0.33    0.00    4.87    0.00    0.04    0.00   46.40
inventor@Cosmogic:~/Desktop$ cat b.txt
Ubuntu
Kali
Mac
Widows
Ubuntu
Linux Mint
Kubuntu
inventor@Cosmogic:~/Desktop$ sort -u b.txt
Kali
Kubuntu
Linux Mint
Mac
Ubuntu
Widows
inventor@Cosmogic:~/Desktop$ 
```

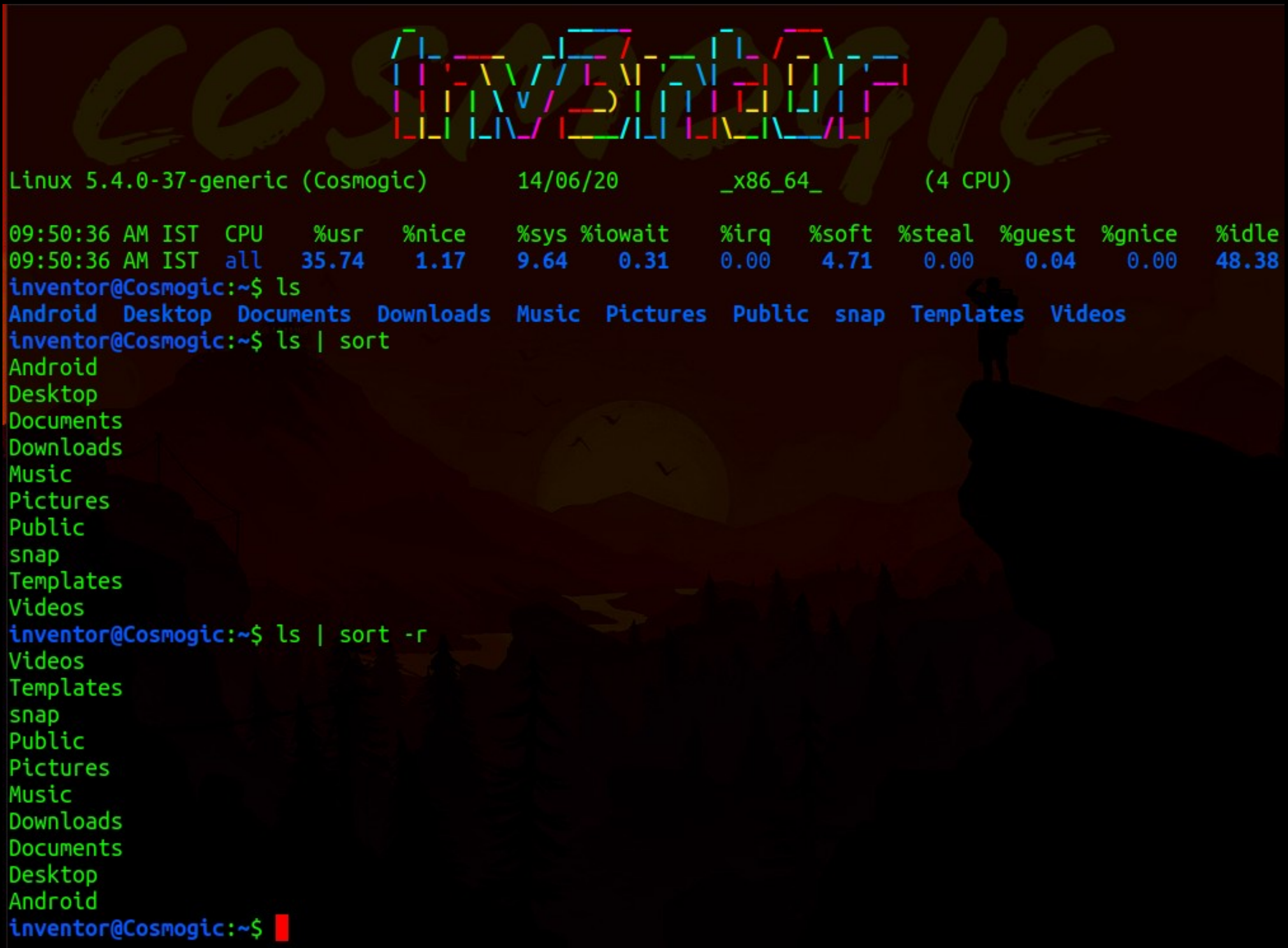
If we have two column in a file and we want to sort by any specific column we can use -k



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

09:41:32 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
09:41:32 AM IST  all   36.38    1.20    9.82    0.32    0.00    4.79    0.00    0.04    0.00   47.44
inventor@Cosmogic:~/Desktop$ sort  a.txt
A       1343
h       4567
I       1500
L       5645
R       132
inventor@Cosmogic:~/Desktop$ sort -k 2n a.txt
R       132
A       1343
I       1500
h       4567
L       5645
```

Pipelines

The ability of commands to read data from standard input and send to
standard output is utilized by a shell feature called pipelines. Using the pipe
operator | (vertical bar), the standard output of one command can be piped
into the standard input of another.
For example

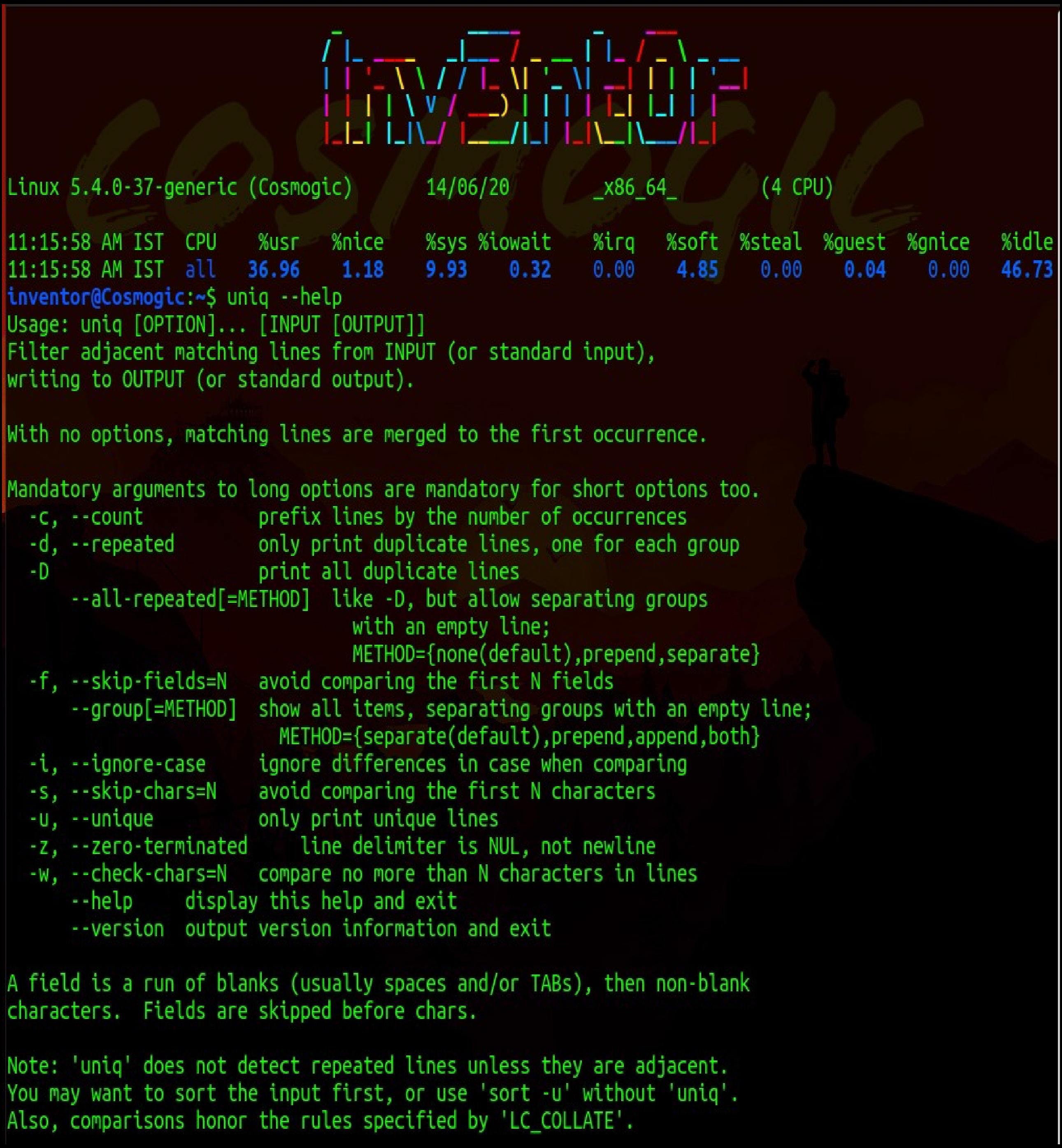


```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

09:50:36 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
09:50:36 AM IST  all   35.74    1.17    9.64    0.31    0.00    4.71    0.00    0.04    0.00   48.38
inventor@Cosmogic:~$ ls
Android  Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
inventor@Cosmogic:~$ ls | sort
Android
Desktop
Documents
Downloads
Music
Pictures
Public
snap
Templates
Videos
inventor@Cosmogic:~$ ls | sort -r
Videos
Templates
snap
Public
Pictures
Music
Downloads
Documents
Desktop
Android
inventor@Cosmogic:~$ 
```

using this we can merge different command to get desired results

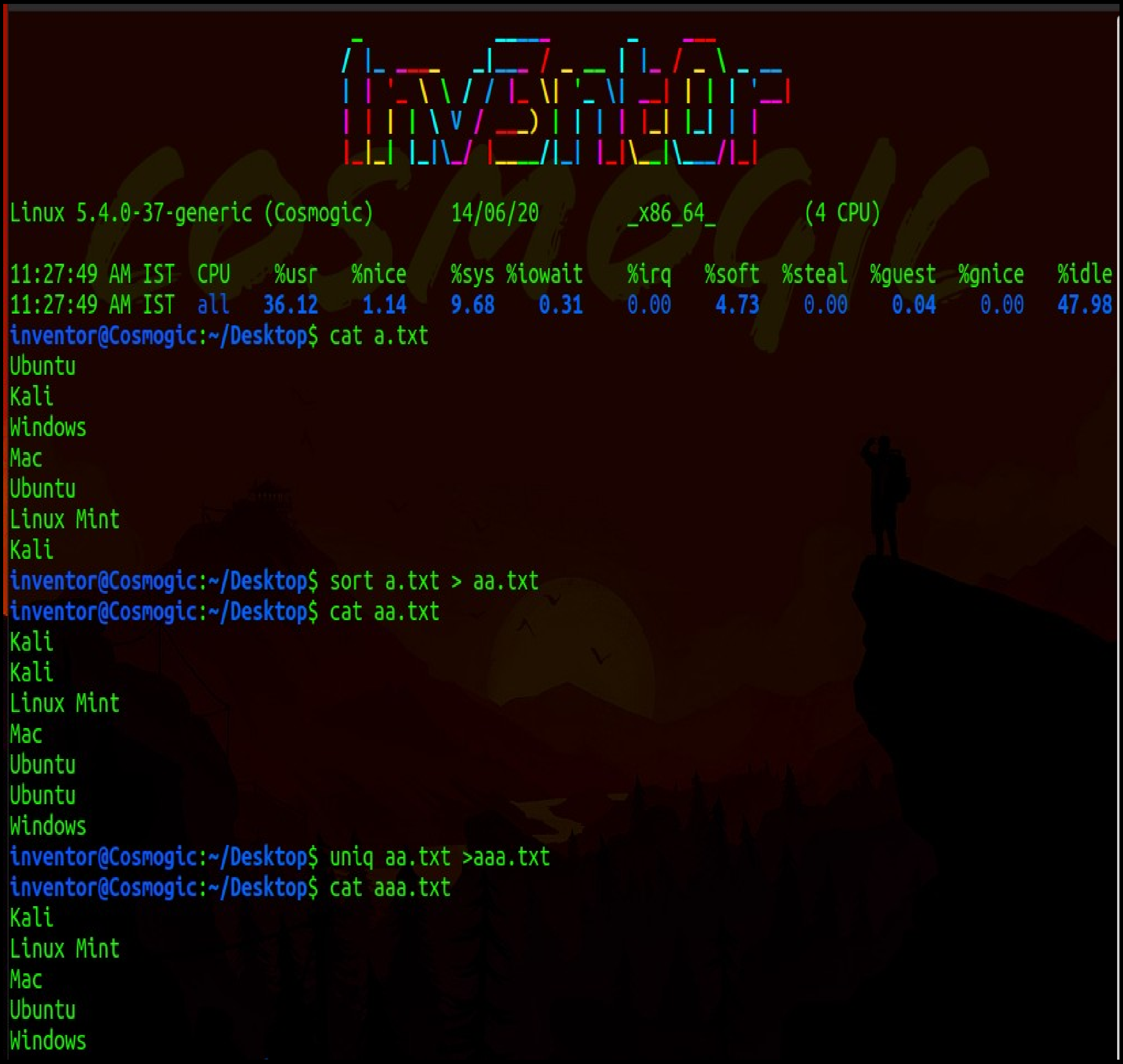
UNIQ

This command is used to remove all repeated lines from a file and it is designed to work with sort command options available with uniq command are :



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

11:15:58 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:15:58 AM IST  all   36.96    1.18    9.93    0.32    0.00    4.85    0.00    0.04    0.00   46.73
inventor@Cosmogic:~$ uniq --help
Usage: uniq [OPTION]... [INPUT [OUTPUT]]
Filter adjacent matching lines from INPUT (or standard input),
writing to OUTPUT (or standard output).

With no options, matching lines are merged to the first occurrence.

Mandatory arguments to long options are mandatory for short options too.
  -c, --count           prefix lines by the number of occurrences
  -d, --repeated        only print duplicate lines, one for each group
  -D                    print all duplicate lines
      --all-repeated[=METHOD]  like -D, but allow separating groups
                                 with an empty line;
                                 METHOD={none(default),prepend,separate}
  -f, --skip-fields=N   avoid comparing the first N fields
      --group[=METHOD]  show all items, separating groups with an empty line;
                          METHOD={separate(default),prepend,append,both}
  -i, --ignore-case     ignore differences in case when comparing
  -s, --skip-chars=N    avoid comparing the first N characters
  -u, --unique          only print unique lines
  -z, --zero-terminated     line delimiter is NUL, not newline
  -w, --check-chars=N   compare no more than N characters in lines
      --help     display this help and exit
      --version  output version information and exit

A field is a run of blanks (usually spaces and/or TABs), then non-blank
characters.  Fields are skipped before chars.

Note: 'uniq' does not detect repeated lines unless they are adjacent.
You may want to sort the input first, or use 'sort -u' without 'uniq'.
Also, comparisons honor the rules specified by 'LC_COLLATE'.
```

to remove the same lines from the file first we need to sort them in a file then we can remove the same line as uniq does not detect repeated lines unless they are adjacent .



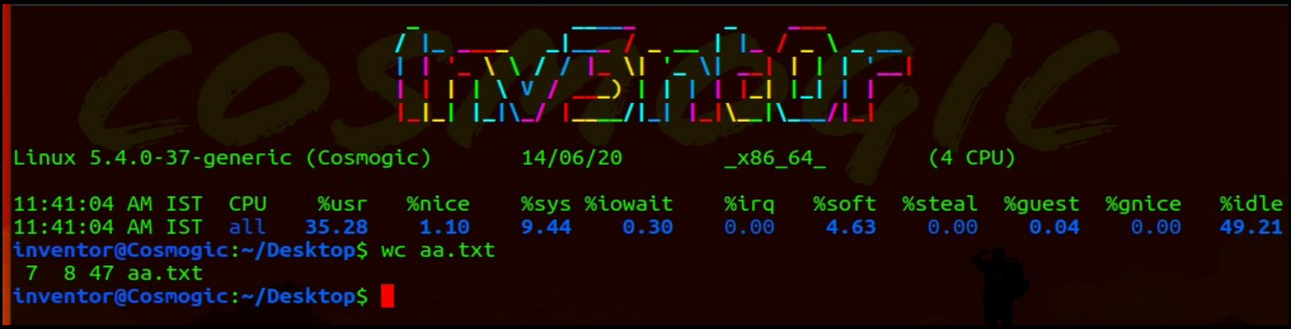
```
Linux 5.4.0-37-generic (Cosmogic)      14/06/20        _x86_64_        (4 CPU)

11:27:49 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:27:49 AM IST  all   36.12    1.14    9.68    0.31    0.00    4.73    0.00    0.04    0.00   47.98
inventor@Cosmogic:~/Desktop$ cat a.txt
Ubuntu
Kali
Windows
Mac
Ubuntu
Linux Mint
Kali
inventor@Cosmogic:~/Desktop$ sort a.txt > aa.txt
inventor@Cosmogic:~/Desktop$ cat aa.txt
Kali
Kali
Linux Mint
Mac
Ubuntu
Ubuntu
Windows
inventor@Cosmogic:~/Desktop$ uniq aa.txt >aaa.txt
inventor@Cosmogic:~/Desktop$ cat aaa.txt
Kali
Linux Mint
Mac
Ubuntu
Windows
```

WC :- wc stands for word count ans as we can understand by the name it's used for counting purpose

It is used to find out number of lines, word count, byte and characters count in the files

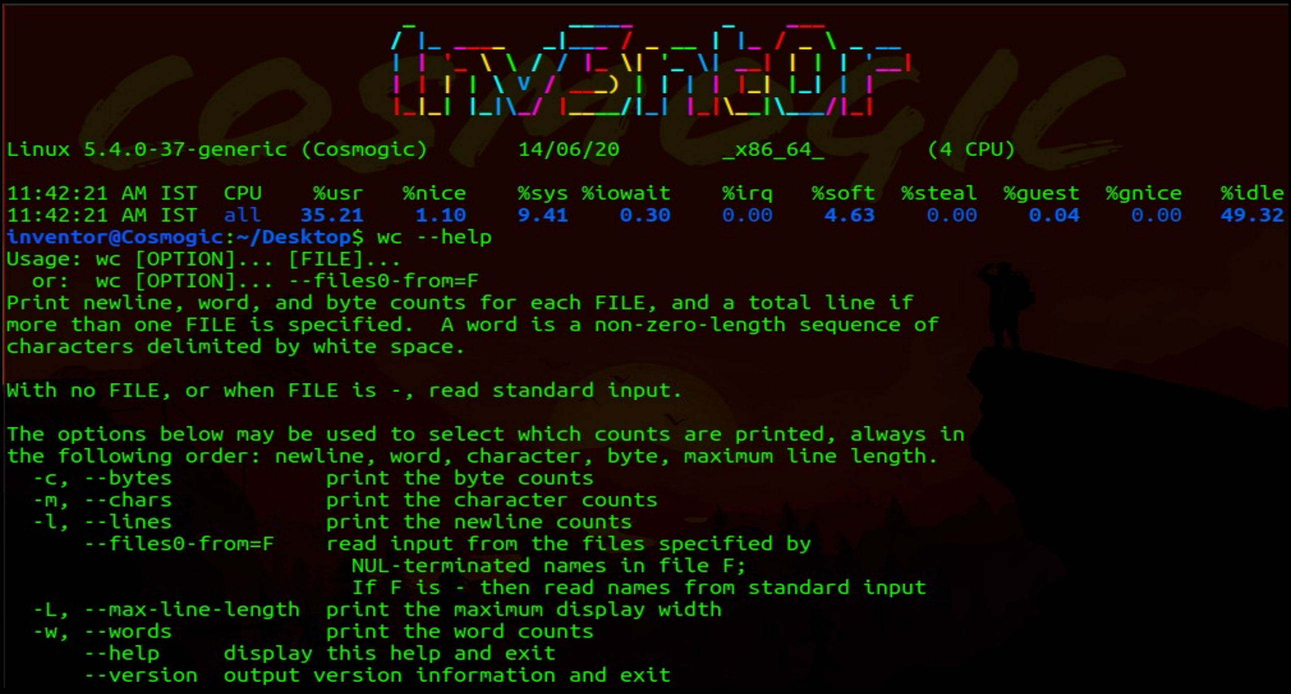
By default four columns are displayed in the output

First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

11:41:04 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:41:04 AM IST  all   35.28    1.10    9.44    0.30    0.00    4.63    0.00    0.04    0.00   49.21
inventor@Cosmogic:~/Desktop$ wc aa.txt
 7  8 47 aa.txt
inventor@Cosmogic:~/Desktop$ 
```

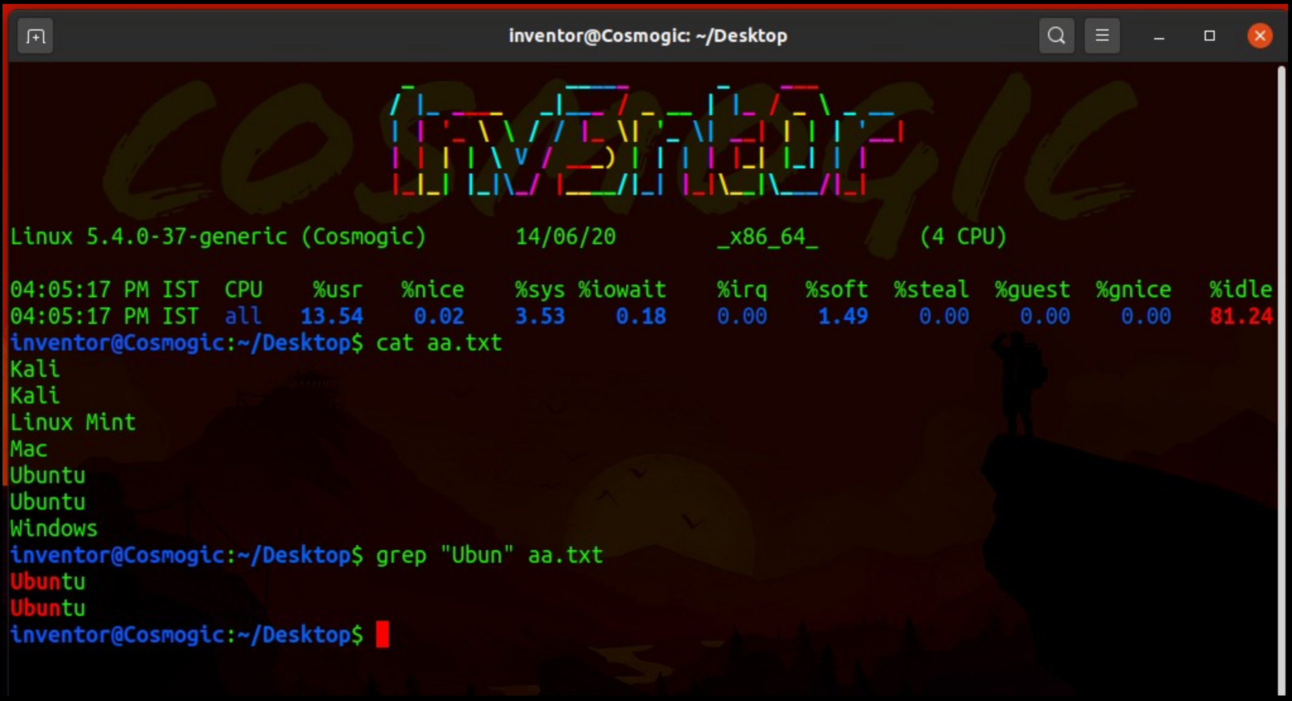
Different options available in wc command are :



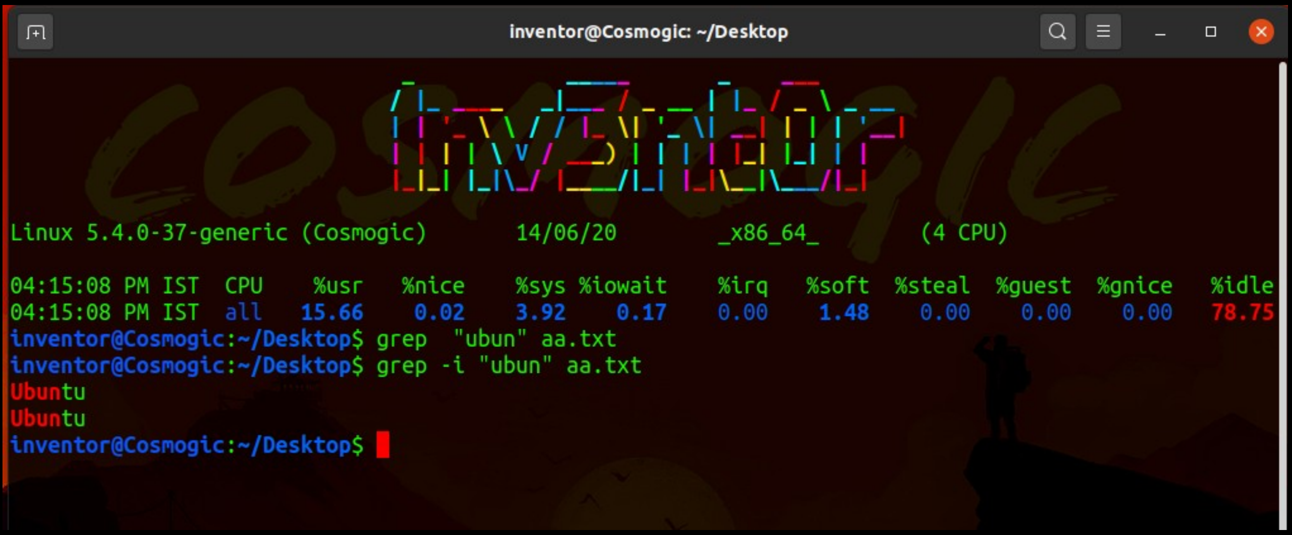
```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

11:42:21 AM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:42:21 AM IST  all   35.21    1.10    9.41    0.30    0.00    4.63    0.00    0.04    0.00   49.32
inventor@Cosmogic:~/Desktop$ wc --help
Usage: wc [OPTION]... [FILE]...
  or:  wc [OPTION]... --files0-from=F
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
  -c, --bytes            print the byte counts
  -m, --chars            print the character counts
  -l, --lines            print the newline counts
      --files0-from=F    read input from the files specified by
                           NUL-terminated names in file F;
                           If F is - then read names from standard input
  -L, --max-line-length  print the maximum display width
  -w, --words            print the word counts
      --help     display this help and exit
      --version  output version information and exit
```

GREP

It is one of the most powerful and widely used command of Linux operating system . It is used to find a pattern specified by us in the file
to find something we write
grep "what ever we want to find" filename

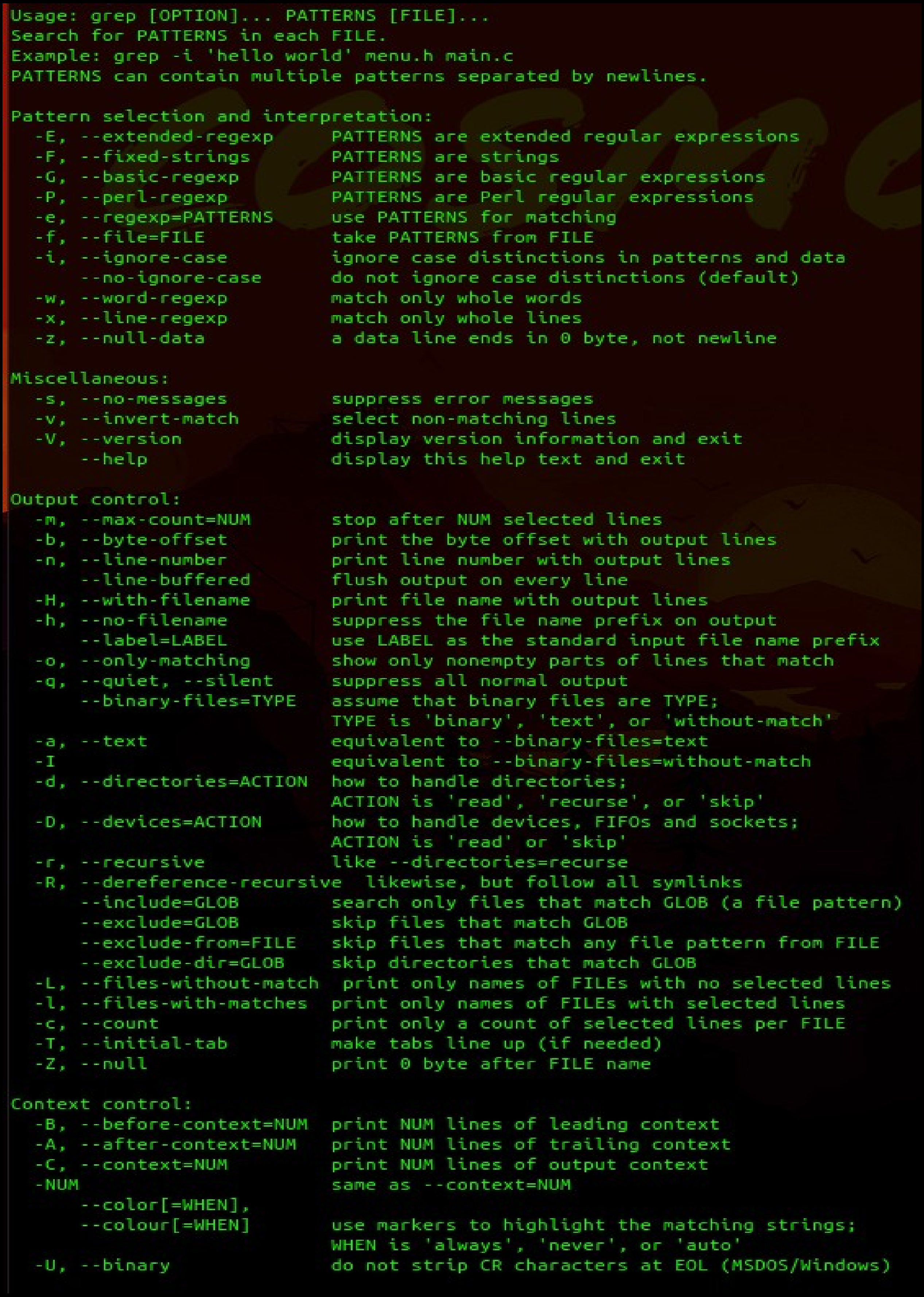


```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

04:05:17 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:05:17 PM IST  all   13.54    0.02    3.53    0.18    0.00    1.49    0.00    0.00    0.00   81.24
inventor@Cosmogic:~/Desktop$ cat aa.txt
Kali
Kali
Linux Mint
Mac
Ubuntu
Ubuntu
Windows
inventor@Cosmogic:~/Desktop$ grep "Ubun" aa.txt
Ubuntu
Ubuntu
inventor@Cosmogic:~/Desktop$
```



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

04:15:08 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:15:08 PM IST  all   15.66    0.02    3.92    0.17    0.00    1.48    0.00    0.00    0.00   78.75
inventor@Cosmogic:~/Desktop$ grep  "ubun" aa.txt
inventor@Cosmogic:~/Desktop$ grep -i "ubun" aa.txt
Ubuntu
Ubuntu
inventor@Cosmogic:~/Desktop$
```

Options present in grep command are :



```
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
  -E, --extended-regexp     PATTERNS are extended regular expressions
  -F, --fixed-strings       PATTERNS are strings
  -G, --basic-regexp        PATTERNS are basic regular expressions
  -P, --perl-regexp         PATTERNS are Perl regular expressions
  -e, --regexp=PATTERNS     use PATTERNS for matching
  -f, --file=FILE           take PATTERNS from FILE
  -i, --ignore-case         ignore case distinctions in patterns and data
      --no-ignore-case      do not ignore case distinctions (default)
  -w, --word-regexp         match only whole words
  -x, --line-regexp         match only whole lines
  -z, --null-data           a data line ends in 0 byte, not newline

Miscellaneous:
  -s, --no-messages         suppress error messages
  -v, --invert-match        select non-matching lines
  -V, --version             display version information and exit
      --help                display this help text and exit

Output control:
  -m, --max-count=NUM       stop after NUM selected lines
  -b, --byte-offset         print the byte offset with output lines
  -n, --line-number         print line number with output lines
      --line-buffered       flush output on every line
  -H, --with-filename       print file name with output lines
  -h, --no-filename         suppress the file name prefix on output
      --label=LABEL         use LABEL as the standard input file name prefix
  -o, --only-matching       show only nonempty parts of lines that match
  -q, --quiet, --silent     suppress all normal output
      --binary-files=TYPE   assume that binary files are TYPE;
                            TYPE is 'binary', 'text', or 'without-match'
  -a, --text                equivalent to --binary-files=text
  -I                        equivalent to --binary-files=without-match
  -d, --directories=ACTION  how to handle directories;
                            ACTION is 'read', 'recurse', or 'skip'
  -D, --devices=ACTION      how to handle devices, FIFOs and sockets;
                            ACTION is 'read' or 'skip'
  -r, --recursive           like --directories=recurse
  -R, --dereference-recursive  likewise, but follow all symlinks
      --include=GLOB        search only files that match GLOB (a file pattern)
      --exclude=GLOB        skip files that match GLOB
      --exclude-from=FILE   skip files that match any file pattern from FILE
      --exclude-dir=GLOB    skip directories that match GLOB
  -L, --files-without-match  print only names of FILEs with no selected lines
  -l, --files-with-matches  print only names of FILEs with selected lines
  -c, --count               print only a count of selected lines per FILE
  -T, --initial-tab         make tabs line up (if needed)
  -Z, --null                print 0 byte after FILE name

Context control:
  -B, --before-context=NUM  print NUM lines of leading context
  -A, --after-context=NUM   print NUM lines of trailing context
  -C, --context=NUM         print NUM lines of output context
  -NUM                      same as --context=NUM
      --color[=WHEN],
      --colour[=WHEN]       use markers to highlight the matching strings;
                            WHEN is 'always', 'never', or 'auto'
  -U, --binary              do not strip CR characters at EOL (MSDOS/Windows)
```

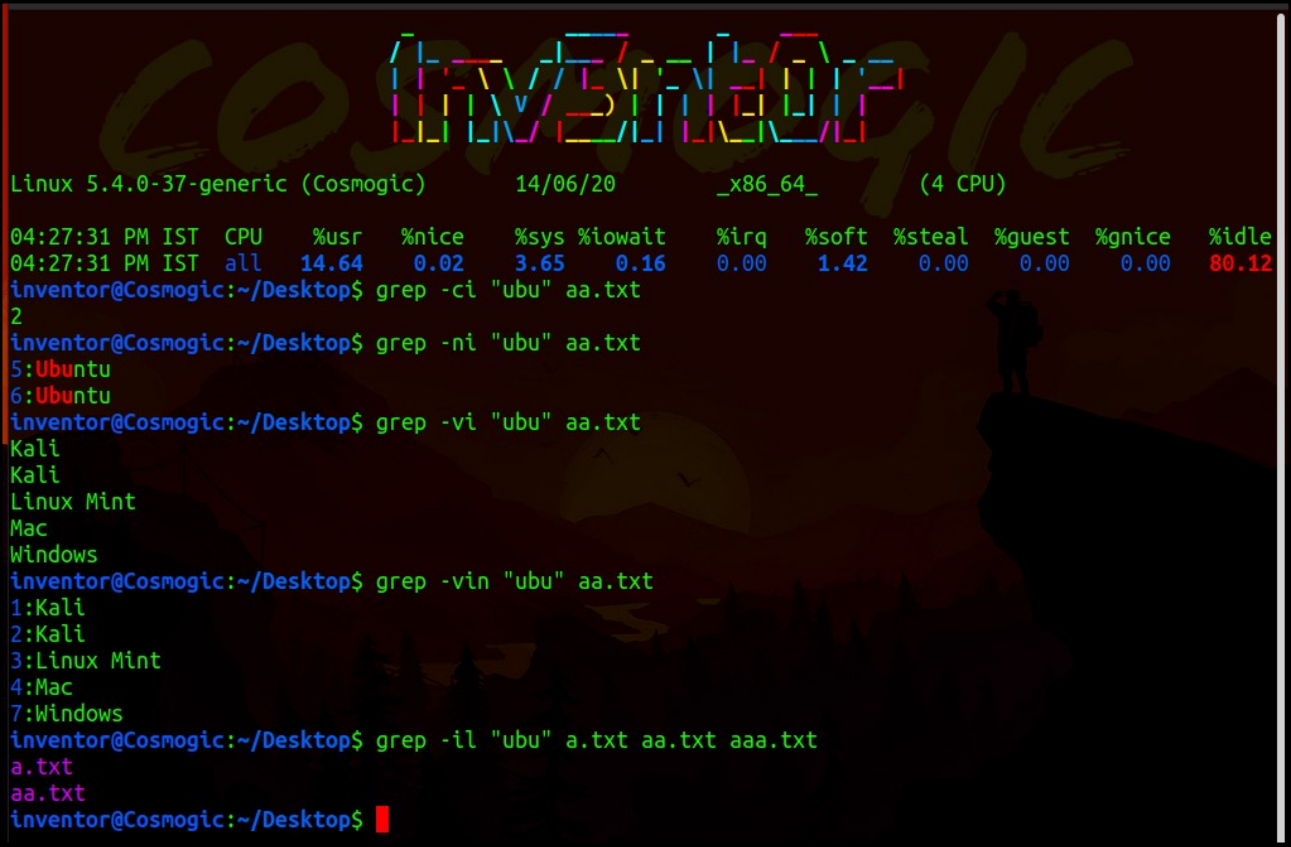
We use

-c to show how many times the pattern matched

-l to show how many files have that pattern

-n to print which line have that pattern

-v to print that doesn't match



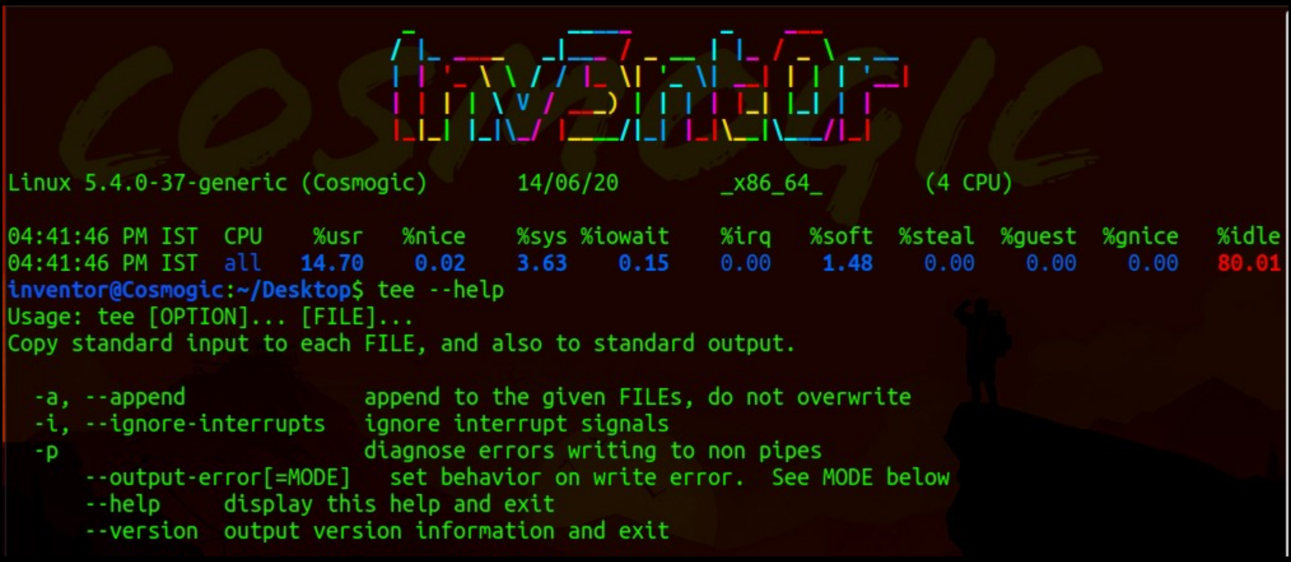
```
 _              _____       _    ___
/ |_ __ __   __|___ / _ __ | |_ / _ \ _ __
| | '_ \\ \ / /  |_ \| '_ \| __| | | | '__|
| | | | |\ V /  ___) | | | | |_| |_| | |
|_|_| |_| \_/  |____/|_| |_|\__|\___/|_|

Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

04:27:31 PM IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:27:31 PM IST  all   14.64    0.02    3.65    0.16    0.00    1.42    0.00    0.00    0.00   80.12
inventor@Cosmogic:~/Desktop$ grep -ci "ubu" aa.txt
2
inventor@Cosmogic:~/Desktop$ grep -ni "ubu" aa.txt
5:Ubuntu
6:Ubuntu
inventor@Cosmogic:~/Desktop$ grep -vi "ubu" aa.txt
Kali
Kali
Linux Mint
Mac
Windows
inventor@Cosmogic:~/Desktop$ grep -vin "ubu" aa.txt
1:Kali
2:Kali
3:Linux Mint
4:Mac
7:Windows
inventor@Cosmogic:~/Desktop$ grep -il "ubu" a.txt aa.txt aaa.txt
a.txt
aa.txt
inventor@Cosmogic:~/Desktop$ 
```

try to explore grep command more with the help of the list given in page 18 .

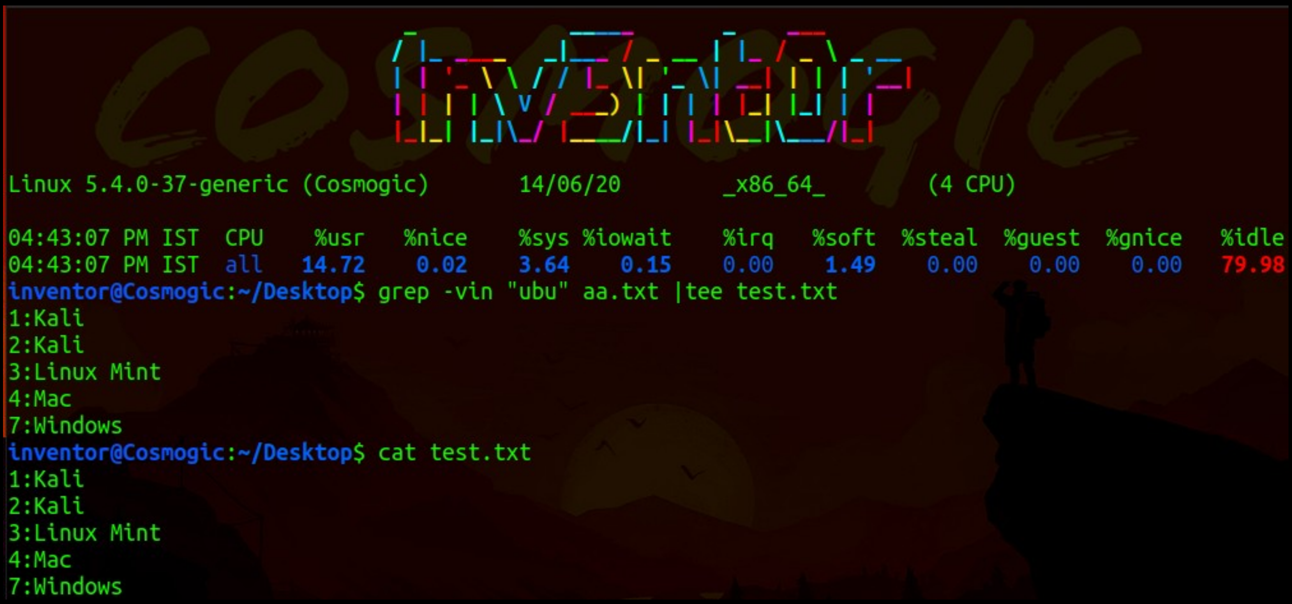
you can also try combining different command to get some specific results .

Tee

It reads the input and writs to both the output and one more file .
We can understand it as the T splitter we use in water connection .
We use it to store the output and display at the same time . Option available are :



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

04:41:46 PM IST  CPU     %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:41:46 PM IST  all    14.70    0.02    3.63    0.15    0.00    1.48    0.00    0.00    0.00   80.01
inventor@Cosmogic:~/Desktop$ tee --help
Usage: tee [OPTION]... [FILE]...
Copy standard input to each FILE, and also to standard output.

  -a, --append              append to the given FILEs, do not overwrite
  -i, --ignore-interrupts   ignore interrupt signals
  -p                        diagnose errors writing to non pipes
      --output-error[=MODE]   set behavior on write error.  See MODE below
      --help     display this help and exit
      --version  output version information and exit
```



```
Linux 5.4.0-37-generic (Cosmogic)       14/06/20        _x86_64_        (4 CPU)

04:43:07 PM IST  CPU     %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:43:07 PM IST  all    14.72    0.02    3.64    0.15    0.00    1.49    0.00    0.00    0.00   79.98
inventor@Cosmogic:~/Desktop$ grep -vin "ubu" aa.txt |tee test.txt
1:Kali
2:Kali
3:Linux Mint
4:Mac
7:Windows
inventor@Cosmogic:~/Desktop$ cat test.txt
1:Kali
2:Kali
3:Linux Mint
4:Mac
7:Windows
```

Source code of commands :

echo

https://github.com/coreutils/coreutils/blob/master/src/echo.c

cat

https://github.com/coreutils/coreutils/blob/master/src/cat.c

sort

https://github.com/coreutils/coreutils/blob/master/src/sort.c

uniq

https://github.com/coreutils/coreutils/blob/master/src/uniq.c

wc

https://github.com/coreutils/coreutils/blob/master/src/wc.c

grep

https://www.gnu.org/software/grep/

tee

https://github.com/coreutils/coreutils/blob/master/src/tee.c

Thank you

-1nv3ntor