

Homework_1_27.03

March 28, 2020

[1]: *#1. How many options for command "ls"*

[]: *#Answer:
#There are at least 41 options for command "ls". Types of options are given
↪ below.*

[2]: `man ls`

LS(1) User Commands LS(1)

NAME

`ls` - list directory contents

SYNOPSIS

`ls [OPTION]... [FILE]...`

DESCRIPTION

List information about the FILES (the current directory by default). Sort entries alphabetically if none of `-cftuvSUX` nor `--sort` is specified.

Mandatory arguments to long options are mandatory for short options too.

`-a, --all`
do not ignore entries starting with `.`

`-A, --almost-all`
do not list implied `.` and `..`

`--author`
with `-l`, print the author of each file

`-b, --escape`
print C-style escapes for nongraphic characters

`--block-size=SIZE`
scale sizes by SIZE before printing them; e.g., `'--block-size=M'`

prints sizes in units of 1,048,576 bytes; see SIZE format below

-B, --ignore-backups
do not list implied entries ending with ~

-c with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

-C list entries by columns

--color[=WHEN]
colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

-d, --directory
list directories themselves, not their contents

-D, --dired
generate output designed for Emacs' dired mode

-f do not sort, enable -aU, disable -ls --color

-F, --classify
append indicator (one of */=>@|) to entries

--file-type
likewise, except do not append '*'

--format=WORD
across -x, commas -m, horizontal -x, long -l, single-column -l, verbose -l, vertical -C

--full-time
like -l --time-style=full-iso

-g like -l, but do not list owner

--group-directories-first
group directories before files;

can be augmented with a --sort option, but any use of --sort=none (-U) disables grouping

-G, --no-group
in a long listing, don't print group names

-h, --human-readable

with `-l` and/or `-s`, print human readable sizes (e.g., 1K 234M 2G)

`--si` likewise, but use powers of 1000 not 1024

`-H, --dereference-command-line`
follow symbolic links listed on the command line

`--dereference-command-line-symlink-to-dir`
follow each command line symbolic link

that points to a directory

`--hide=PATTERN`
do not list implied entries matching shell `PATTERN` (overridden
by `-a` or `-A`)

`--hyperlink[=WHEN]`
hyperlink file names; `WHEN` can be 'always' (default if omitted),
'auto', or 'never'

`--indicator-style=WORD`
append indicator with style `WORD` to entry names: none (default),
slash (`-p`), file-type (`--file-type`), classify (`-F`)

`-i, --inode`
print the index number of each file

`-I, --ignore=PATTERN`
do not list implied entries matching shell `PATTERN`

`-k, --kibibytes`
default to 1024-byte blocks for disk usage

`-l` use a long listing format

`-L, --dereference`
when showing file information for a symbolic link, show informa-
tion for the file the link references rather than for the link
itself

`-m` fill width with a comma separated list of entries

`-n, --numeric-uid-gid`
like `-l`, but list numeric user and group IDs

`-N, --literal`
print entry names without quoting

- o like -l, but do not list group information

- p, --indicator-style=slash
 append / indicator to directories

- q, --hide-control-chars
 print ? instead of nongraphic characters

- show-control-chars
 show nongraphic characters as-is (the default, unless program is
 'ls' and output is a terminal)

- Q, --quote-name
 enclose entry names in double quotes

- quoting-style=WORD
 use quoting style WORD for entry names: literal, locale, shell,
 shell-always, shell-escape, shell-escape-always, c, escape

- r, --reverse
 reverse order while sorting

- R, --recursive
 list subdirectories recursively

- s, --size
 print the allocated size of each file, in blocks

- S sort by file size, largest first

- sort=WORD
 sort by WORD instead of name: none (-U), size (-S), time (-t),
 version (-v), extension (-X)

- time=WORD
 with -l, show time as WORD instead of default modification time:
 atime or access or use (-u); ctime or status (-c); also use
 specified time as sort key if --sort=time (newest first)

- time-style=STYLE
 with -l, show times using style STYLE: full-iso, long-iso, iso,
 locale, or +FORMAT; FORMAT is interpreted like in 'date'; if
 FORMAT is FORMAT1<newline>FORMAT2, then FORMAT1 applies to
 non-recent files and FORMAT2 to recent files; if STYLE is pre-
 fixed with 'posix-', STYLE takes effect only outside the POSIX
 locale

- t sort by modification time, newest first

-T, --tabsize=COLS
 assume tab stops at each COLS instead of 8

-u with -lt: sort by, and show, access time; with -l: show access
 time and sort by name; otherwise: sort by access time, newest
 first

-U do not sort; list entries in directory order

-v natural sort of (version) numbers within text

-w, --width=COLS
 set output width to COLS. 0 means no limit

-x list entries by lines instead of by columns

-X sort alphabetically by entry extension

-Z, --context
 print any security context of each file

-1 list one file per line. Avoid '\n' with -q or -b

--help display this help and exit

--version
 output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*1024). Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).

Using color to distinguish file types is disabled both by default and with --color=never. With --color=auto, ls emits color codes only when standard output is connected to a terminal. The LS_COLORS environment variable can change the settings. Use the dircolors command to set it.

Exit status:

0 if OK,

1 if minor problems (e.g., cannot access subdirectory),

2 if serious trouble (e.g., cannot access command-line argument).

AUTHOR

Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
Report ls translation bugs to <<http://translationproject.org/team/>>

COPYRIGHT

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU
GPL version 3 or later <<http://gnu.org/licenses/gpl.html>>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

Full documentation at: <<http://www.gnu.org/software/coreutils/ls>>
or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.28

January 2018

LS(1)

```
[3]: #2. Create a folder with your name and txt file inside the folder with your  
↪ resume
```

```
[4]: pwd
```

/home/jupyter-zhannaspace

```
[8]: file zhanna
```

zhanna: directory

```
[10]: cd zhanna
```

```
[12]: file CV_Zhanna.txt
```

CV_Zhanna.txt: UTF-8 Unicode text

```
[61]: mkdir ZHANNA
```

```
[62]: cd ZHANNA
```

```
[67]: echo 'Zhanna Suimenbayeva  
      Birthday 21.03.1984  
      Education:  
          2015-2018 Kazakh National Technical University named after K.I.  
↪ Satpayev  
          (Phd program)  
          2013-2015 Kazakh National Technical University named after K.I.  
↪ Satpayev (Master programme, 1st course)  
      Department: Operation of space means  
      Speciality: Space techniques and technologies
```

Institut Supérieur de l'Aéronautique et de l'Espace (ISAE) Toulouse, France
→ (Exchange program, 2nd course)
Speciality: Space Systems' >resume

```
[68]: file resume
```

```
resume: ASCII text
```

```
[ ]: #3. Play all the commands we've learned on the folder and file you created
```

```
[15]: touch textfile      #create a blank file
```

```
[16]: ls
```

```
CV_Zhanna.txt  textfile
```

```
[17]: file textfile
```

```
textfile: empty
```

```
[18]: echo 'UniSat' >unisat  #Create a textfile "unisat" with a word "UniSat" inside
```

```
[20]: file unisat
```

```
unisat: ASCII text
```

```
[30]: ls
```

```
CV_Zhanna.txt  resume  textfile  unisat
```

```
[26]: file resume
```

```
resume: ASCII text
```

```
[28]: pwd
```

```
/home/jupyter-zhannaspace/zhanna
```

```
[31]: ls -a  #list of files including hidden ones
```

```
.  ..  CV_Zhanna.txt  .ipynb_checkpoints  resume  textfile  unisat
```

```
[29]: ls -l  # Detailed list of files
```

```
total 12
-rw-r--r-- 1 jupyter-zhannaspace jupyter-zhannaspace 1376 Mar 28 02:07
CV_Zhanna.txt
-rw-r--r-- 1 jupyter-zhannaspace jupyter-zhannaspace  532 Mar 28 02:53 resume
```

```
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 0 Mar 28 02:24 textfile
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 7 Mar 28 02:44 unisat
```

[64]: `ls -h`

[33]: `ls`

```
CV_Zhanna.txt  resume  textfile  unisat  Untitled.ipynb
```

[34]: `ls -A`

```
CV_Zhanna.txt  .ipynb_checkpoints  resume  textfile  unisat  Untitled.ipynb
```

[35]: `ls -A -l`

```
CV_Zhanna.txt
.ipynb_checkpoints
resume
textfile
unisat
Untitled.ipynb
```

[36]: `ls -d`

```
.
```

[37]: `cd ..`

[38]: `ls -d`

```
.
```

[39]: `pwd`

```
/home/jupyter-zhannaspac
```

[40]: `cd zhanna`

[41]: `ls -b`

```
CV_Zhanna.txt  resume  textfile  unisat  Untitled.ipynb
```

[45]: `ls -c -lt #sort by last modification`

```
total 24
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 11198 Mar 28 03:00
Untitled.ipynb
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 532 Mar 28 02:53 resume
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 7 Mar 28 02:44 unisat
```



```
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac    0 Mar 28 02:24 textfile
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 1376 Mar 28 02:07
CV_Zhanna.txt
```

[47]: `ls -c -l #sort by name`

```
total 24
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 1376 Mar 28 02:07
CV_Zhanna.txt
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac  532 Mar 28 02:53 resume
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac    0 Mar 28 02:24 textfile
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac    7 Mar 28 02:44 unisat
-rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 1198 Mar 28 03:00
Untitled.ipynb
```

[51]: `ls -f -ls`

```
total 40
 4 drwxr-xr-x 2 jupyter-zhannaspac jupyter-zhannaspac 4096 Mar 28 03:14
'folder 1'
 0 -rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac    0 Mar 28 02:24
textfile
 4 -rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac  532 Mar 28 02:53
resume
 4 drwxr-xr-x 2 jupyter-zhannaspac jupyter-zhannaspac 4096 Mar 28 02:58
.ipynb_checkpoints
 4 -rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac    7 Mar 28 02:44
unisat
 4 drwxr-x--- 9 jupyter-zhannaspac jupyter-zhannaspac 4096 Mar 28 03:12 ..
 4 -rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 1376 Mar 28 02:07
CV_Zhanna.txt
 4 drwxr-xr-x 4 jupyter-zhannaspac jupyter-zhannaspac 4096 Mar 28 03:14 .
12 -rw-r--r-- 1 jupyter-zhannaspac jupyter-zhannaspac 1198 Mar 28 03:00
Untitled.ipynb
```

[52]: `ls -g`

```
total 28
-rw-r--r-- 1 jupyter-zhannaspac 1376 Mar 28 02:07 CV_Zhanna.txt
drwxr-xr-x 2 jupyter-zhannaspac 4096 Mar 28 03:14 'folder 1'
-rw-r--r-- 1 jupyter-zhannaspac  532 Mar 28 02:53 resume
-rw-r--r-- 1 jupyter-zhannaspac    0 Mar 28 02:24 textfile
-rw-r--r-- 1 jupyter-zhannaspac    7 Mar 28 02:44 unisat
-rw-r--r-- 1 jupyter-zhannaspac 1198 Mar 28 03:00 Untitled.ipynb
```

[53]: `ls -i`

```
10098588 CV_Zhanna.txt 10098600 resume 10098597 unisat
```

```
10098608 'folder 1'      10098592  textfile  10098595  Untitled.ipynb
```

```
[54]: ls -k
```

```
CV_Zhanna.txt  'folder 1'  resume  textfile  unisat  Untitled.ipynb
```

```
[55]: ls -n
```

```
total 28
-rw-r--r-- 1 1020 1021  1376 Mar 28 02:07 CV_Zhanna.txt
drwxr-xr-x 2 1020 1021  4096 Mar 28 03:14 'folder 1'
-rw-r--r-- 1 1020 1021   532 Mar 28 02:53 resume
-rw-r--r-- 1 1020 1021     0 Mar 28 02:24 textfile
-rw-r--r-- 1 1020 1021     7 Mar 28 02:44 unisat
-rw-r--r-- 1 1020 1021 11198 Mar 28 03:00 Untitled.ipynb
```

```
[56]: ls -m
```

```
CV_Zhanna.txt, 'folder 1', resume, textfile, unisat, Untitled.ipynb
```

```
[58]: ls -s
```

```
total 28
 4 CV_Zhanna.txt    4 resume      4 unisat
 4 'folder 1'       0 textfile   12 Untitled.ipynb
```

```
[ ]: # Conclusion: there are many options for command "ls", which allow to list
      ↪ directory content
      # and sort it by size, name, time, groups, dates, owners and etc
```