cubefiction

March 28, 2020

```
[29]: pwd       #currentway panic mode was activated
```

```
/home/jupyter-cubefiction
```

```
[30]: cd cubefiction   #openparticularfolder
```

```
[31]: pwd #you are here :0
```

```
/home/jupyter-cubefiction/cubefiction
```

```
[32]: ls            #opensoderzhanie
```

```
 giphy.gif  'Untitled Folder'
```

```
[34]: file giphy.gif  #soderzhanie one way
```

```
giphy.gif: GIF image data, version 89a, 478 x 472
```

```
[36]: file ./giphy.gif #soderzhanie  another way
```

```
./giphy.gif: GIF image data, version 89a, 478 x 472
```

```
[43]: pwd       #WHERE AM I????
```

```
/home/jupyter-cubefiction/cubefiction
```

```
[52]: man ls    #DO NOT WORRY PLZ
```

```
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  speci-
```

```
fied.

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  -1,
       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)
```

[51]:
```
pwd
```

```
/home/jupyter-cubefiction/cubefiction
```

[53]:
```
ls -a
```

```
 .   ..   giphy.gif   .ipynb_checkpoints  'where is my algorithm?'
```

[54]:
```
cd 'where is my algorithm?'   #where am i? help pls :O
```

[55]:
```
pwd      #you are here -_-
```

```
/home/jupyter-cubefiction/cubefiction/where is my algorithm?
```

[58]:
```
ls           #i want to know my existence :C
```

```
'algorithm of love.txt'
```

[60]:
```
ls -1         #here!!!
```

```
'algorithm of love.txt'
```

[64]:
```
touch 'algorithm of love.txt'
```

[67]:
```
ls -a          #if u want to know MOREEEEEEE -_-
```

```
 .   ..  'algorithm of love.txt'   .ipynb_checkpoints
```

[70]:
```
file 'algorithm of love.txt' #soderzhanietxt
```

```
algorithm of love.txt: C++ source, ASCII text, with CRLF line terminators
```

[71]:
```
cat 'algorithm of love.txt'
```

```
#include <iostream>

using namespace std;

struct Wi_Fi{
        char name[20];
        int x;
        int y;
        int r;
};

int main(){
        int n;
        cin >> n;

        Wifi *A = new Wifi[n];

        for(int i = 0; i<n; i++){
                cin >> A[i].name >> A[i].x >> A[i].y >> A[i].r;
        }

        cout << "Done coordinates: ";
        int x1, y1;
        cin >> x1 >> y1;

        for(int i = 0; i<n; i++){
                if(x1<=A[i].x+A[i].r/2 && x1>=A[i].x-A[i].r/2 &&
y1<=A[i].y+A[i].r/2 && y1>=A[i].y-A[i].r/2) cout << A[i].name << endl;
```

```
            }
    }
```

```
[ ]: #CONGRATS, a ru happy now ^-^???
```