gramma and template

kevinluo

Contents

contents

1 网络资源地址

Linux 基础知识——Linux 常用命令大全

Linux 命令大全

在 Linux 下查看文件三种时间

Linux 系统下 date 常用命令的参数以及获取时间戳的方法

如何使用 Linux sed 命令进行字符串替换

Linux xargs 命令 www.runoob.com

2 Linux 命令--help

GNU coreutils online help:

GNU findutils

GNU gawk

GNU sed

Decoded: GNU coreutils

cp

touch

rm

ls

mv

mkdir

cat

GNU find

2.1 rm

```
$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).
  -f, --force           ignore nonexistent files and arguments, never    prompt
  -i                    prompt before every removal
  -I                    prompt once before removing more than three    files, or
                          when removing recursively; less intrusive than   -i,
                          while still giving protection against most    mistakes
      --interactive[=WHEN]  prompt according to WHEN: never, once (-I),   or
                          always (-i); without WHEN, prompt always
      --one-file-system  when removing a hierarchy recursively, skip any
                          directory that is on a file system different    from
                          that of the corresponding command line argument
      --no-preserve-root  do not treat '/' specially
      --preserve-root   do not remove '/' (default)
  -r, -R, --recursive   remove directories and their contents recursively
  -d, --dir             remove empty directories
  -v, --verbose         explain what is being done
      --help      display this help and exit
      --version   output version information and exit
By default, rm does not remove directories.  Use the --recursive (-r or    -R)
option to remove each listed directory, too, along with all of its    contents.
To remove a file whose name starts with a '-', for example '-foo',
use one of these commands:
  rm -- -foo
  rm ./-foo
Note that if you use rm to remove a file, it might be possible to recover
some of its contents, given sufficient expertise and/or time.  For    greater
assurance that the contents are truly unrecoverable, consider using    shred.
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/rm>
or available locally via: info '(coreutils) rm invocation'
The command "rm --help" exited with 0.
```

3

2.2 cp

```
$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                 option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
                                 a previous -i option)
  -P, --no-dereference         never follow symbolic links in SOURCE
  -p                           same as --preserve=mode,ownership,timestamps
      --preserve[=ATTR_LIST]   preserve the specified attributes (default:
                                 mode,ownership,timestamps), if possible
                                 additional attributes: context, links, xattr,
                                 all
      --no-preserve=ATTR_LIST  don't preserve the specified attributes
      --parents                use full source file name under DIRECTORY
  -R, -r, --recursive          copy directories recursively
      --reflink[=WHEN]         control clone/CoW copies. See below
      --remove-destination     remove each existing destination file before
                                 attempting to open it (contrast with --force)
      --sparse=WHEN            control creation of sparse files. See below
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
                                 argument
  -s, --symbolic-link          make symbolic links instead of copying
  -S, --suffix=SUFFIX          override the usual backup suffix
  -t, --target-directory=DIRECTORY  copy all SOURCE arguments into DIRECTORY
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 copy only when the SOURCE file is newer
                                 than the destination file or when the
                                 destination file is missing
  -v, --verbose                explain what is being done
  -x, --one-file-system        stay on this file system
  -Z                           set SELinux security context of destination
                                 file to default type
      --context[=CTX]          like -Z, or if CTX is specified then set the
                                 SELinux or SMACK security context to CTX
      --help     display this help and exit
```

```
         --version  output version information and exit
By default, sparse SOURCE files are detected by a crude heuristic and the
corresponding DEST file is made sparse as well.  That is the behavior
selected by --sparse=auto.  Specify --sparse=always to create a sparse DEST
file whenever the SOURCE file contains a long enough sequence of zero bytes.
Use --sparse=never to inhibit creation of sparse files.
When --reflink[=always] is specified, perform a lightweight copy, where the
data blocks are copied only when modified.  If this is not possible the copy
fails, or if --reflink=auto is specified, fall back to a standard copy.
The backup suffix is '~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX.
The version control method may be selected via the --backup option or through
the VERSION_CONTROL environment variable.  Here are the values:
  none, off       never make backups (even if --backup is given)
  numbered, t     make numbered backups
  existing, nil   numbered if numbered backups exist, simple otherwise
  simple, never   always make simple backups
As a special case, cp makes a backup of SOURCE when the force and backup
options are given and SOURCE and DEST are the same name for an existing,
regular file.
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/cp>
or available locally via: info '(coreutils) cp invocation'
The command "cp --help" exited with 0.
0.01s$ \cp -RT $TRAVIS_BUILD_DIR/output/sphinx/build-memo/* /tmp/klgit/gp-memo
cp: extra operand '/home/travis/build/kevinluolog/kdoc/output/sphinx/build-memo/002plan'
Try 'cp --help' for more information.
The command "\cp -RT $TRAVIS_BUILD_DIR/output/sphinx/build-memo/* /tmp/klgit/gp-memo" exited w:
0.00s$ pwd
/tmp/klgit/gp-memo
```

上面cp命令,错在:
- 不能用大写T, 这是表示 DEST是文件, 不是目录, 报错的原因
更正:
cp -rt /tmp/klgit/gp-memo $TRAVIS_BUILD_DIR/output/sphinx/build-memo/*
注意: -rt指定目标目录时要紧跟, 所以如果参数写在前面, 则目标目录也到前面了。
source目录后面带星通配和-r配合使用, 则表示只copy文件和子目录。

2.3 touch

```
0.02s$ touch --help
Usage: touch [OPTION]... FILE...
Update the access and modification times of each FILE to the current time.
A FILE argument that does not exist is created empty, unless -c or -h
is supplied.
A FILE argument string of - is handled specially and causes touch to
change the times of the file associated with standard output.
Mandatory arguments to long options are mandatory for short options too.
  -a                     change only the access time
  -c, --no-create        do not create any files
  -d, --date=STRING      parse STRING and use it instead of current time
  -f                     (ignored)
  -h, --no-dereference   affect each symbolic link instead of any   referenced
```

```
                            file (useful only on systems that can change the
                            timestamps of a symlink)
  -m                        change only the modification time
  -r, --reference=FILE      use this file's times instead of current time
  -t STAMP                  use [[CC]YY]MMDDhhmm[.ss] instead of current time
      --time=WORD           change the specified time:
                               WORD is access, atime, or use: equivalent to -a
                               WORD is modify or mtime: equivalent to -m
      --help     display this help and exit
      --version  output version information and exit
Note that the -d and -t options accept different time-date formats.
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/touch>
or available locally via: info '(coreutils) touch invocation'
The command "touch --help" exited with 0.
```

2.4   ls

```
$ ls --help
Usage: ls [OPTION]... [FILE]...
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 -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
```

6

```
-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)
    --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 information 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 raw entry names (don't treat e.g.    control
                            characters specially)
-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 prefixed 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).
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
The command "ls --help" exited with 0.
```

2.5   find

```
0.01s$ find --help
Usage: find [-H] [-L] [-P] [-Olevel] [-D  help|tree|search|stat|rates|opt|exec|time] [path...]
default path is the current directory; default expression is -print
expression may consist of: operators, options, tests, and actions:
operators (decreasing precedence; -and is implicit where no others are    given):
      ( EXPR )   ! EXPR   -not EXPR   EXPR1 -a EXPR2   EXPR1 -and EXPR2
      EXPR1 -o EXPR2   EXPR1 -or EXPR2   EXPR1 , EXPR2
positional options (always true): -daystart -follow -regextype
normal options (always true, specified before other expressions):
      -depth --help -maxdepth LEVELS -mindepth LEVELS -mount -noleaf
      --version -xdev -ignore_readdir_race -noignore_readdir_race
tests (N can be +N or -N or N): -amin N -anewer FILE -atime N -cmin N
      -cnewer FILE -ctime N -empty -false -fstype TYPE -gid N -group NAME
      -ilname PATTERN -iname PATTERN -inum N -iwholename PATTERN -iregex   PATTERN
      -links N -lname PATTERN -mmin N -mtime N -name PATTERN -newer FILE
      -nouser -nogroup -path PATTERN -perm [-/]MODE -regex PATTERN
      -readable -writable -executable
      -wholename PATTERN -size N[bcwkMG] -true -type [bcdpflsD] -uid N
      -used N -user NAME -xtype [bcdpfls]
      -context CONTEXT
actions: -delete -print0 -printf FORMAT -fprintf FILE FORMAT -print
      -fprint0 FILE -fprint FILE -ls -fls FILE -prune -quit
      -exec COMMAND ; -exec COMMAND {} + -ok COMMAND ;
```

```
      -execdir COMMAND ; -execdir COMMAND {} + -okdir COMMAND ;
Please see also the documentation at http://www.gnu.org/software/  findutils/.
You can report (and track progress on fixing) bugs in the "find"
program via the GNU findutils bug-reporting page at
https://savannah.gnu.org/bugs/?group=findutils or, if
you have no web access, by sending email to <bug-findutils@gnu.org>.
The command "find --help" exited with 0.
```

2.6   sed

```
0.01s$ sed --help
Usage: sed [OPTION]... {script-only-if-no-other-script} [input-file]...
  -n, --quiet, --silent
                 suppress automatic printing of pattern space
  -e script, --expression=script
                 add the script to the commands to be executed
  -f script-file, --file=script-file
                 add the contents of script-file to the commands to be executed
  --follow-symlinks
                 follow symlinks when processing in place
  -i[SUFFIX], --in-place[=SUFFIX]
                 edit files in place (makes backup if SUFFIX supplied)
  -l N, --line-length=N
                 specify the desired line-wrap length for the `l' command
  --posix
                 disable all GNU extensions.
  -r, --regexp-extended
                 use extended regular expressions in the script.
  -s, --separate
                 consider files as separate rather than as a single continuous
                 long stream.
  -u, --unbuffered
                 load minimal amounts of data from the input files and flush
                 the output buffers more often
  -z, --null-data
                 separate lines by NUL characters
      --help     display this help and exit
      --version  output version information and exit
If no -e, --expression, -f, or --file option is given, then the first
non-option argument is taken as the sed script to interpret.  All
remaining arguments are names of input files; if no input files are
specified, then the standard input is read.
GNU sed home page: <http://www.gnu.org/software/sed/>.
General help using GNU software: <http://www.gnu.org/gethelp/>.
E-mail bug reports to: <bug-sed@gnu.org>.
Be sure to include the word ``sed'' somewhere in the ``Subject:'' field.
The command "sed --help" exited with 0.
```

2.7   gawk

```
0.01s$ gawk --help
```

```
Usage: gawk [POSIX or GNU style options] -f progfile [--] file ...
Usage: gawk [POSIX or GNU style options] [--] 'program' file ...
POSIX options:     GNU long options: (standard)
  -f progfile    --file=progfile
  -F fs      --field-separator=fs
  -v var=val     --assign=var=val
Short options:     GNU long options: (extensions)
  -b       --characters-as-bytes
  -c       --traditional
  -C       --copyright
  -d[file]     --dump-variables[=file]
  -D[file]     --debug[=file]
  -e 'program-text' --source='program-text'
  -E file      --exec=file
  -g       --gen-pot
  -h       --help
  -i includefile     --include=includefile
  -l library     --load=library
  -L[fatal|invalid] --lint[=fatal|invalid]
  -M       --bignum
  -N       --use-lc-numeric
  -n       --non-decimal-data
  -o[file]     --pretty-print[=file]
  -O       --optimize
  -p[file]     --profile[=file]
  -P       --posix
  -r       --re-interval
  -S       --sandbox
  -t       --lint-old
  -V       --version
To report bugs, see node `Bugs' in `gawk.info', which is
section `Reporting Problems and Bugs' in the printed version.
gawk is a pattern scanning and processing language.
By default it reads standard input and writes standard output.
Examples:
  gawk '{ sum += $1 }; END { print sum }' file
  gawk -F: '{ print $1 }' /etc/passwd
The command "gawk --help" exited with 0.
```

2.8 awk

```
$ awk --help
Usage: awk [POSIX or GNU style options] -f progfile [--] file ...
Usage: awk [POSIX or GNU style options] [--] 'program' file ...
POSIX options:     GNU long options: (standard)
  -f progfile    --file=progfile
  -F fs      --field-separator=fs
  -v var=val     --assign=var=val
Short options:     GNU long options: (extensions)
  -b       --characters-as-bytes
  -c       --traditional
  -C       --copyright
```

```
-d[file]      --dump-variables[=file]
-D[file]      --debug[=file]
-e 'program-text' --source='program-text'
-E file       --exec=file
-g            --gen-pot
-h            --help
-i includefile    --include=includefile
-l library    --load=library
-L[fatal|invalid] --lint[=fatal|invalid]
-M            --bignum
-N            --use-lc-numeric
-n            --non-decimal-data
-o[file]      --pretty-print[=file]
-O            --optimize
-p[file]      --profile[=file]
-P            --posix
-r            --re-interval
-S            --sandbox
-t            --lint-old
-V            --version
```

To report bugs, see node `Bugs' in `gawk.info', which is
section `Reporting Problems and Bugs' in the printed version.
gawk is a pattern scanning and processing language.
By default it reads standard input and writes standard output.
Examples:
```
  gawk '{ sum += $1 }; END { print sum }' file
  gawk -F: '{ print $1 }' /etc/passwd
```
The command "awk --help" exited with 0.

2.9 grep

:

```
$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERN in each FILE or standard input.
PATTERN is, by default, a basic regular expression (BRE).
Example: grep -i 'hello world' menu.h main.c
Regexp selection and interpretation:
  -E, --extended-regexp     PATTERN is an extended regular expression (   ERE)
  -F, --fixed-strings       PATTERN is a set of newline-separated strings
  -G, --basic-regexp        PATTERN is a basic regular expression (BRE)
  -P, --perl-regexp         PATTERN is a Perl regular expression
  -e, --regexp=PATTERN      use PATTERN for matching
  -f, --file=FILE           obtain PATTERN from FILE
  -i, --ignore-case         ignore case distinctions
  -w, --word-regexp         force PATTERN to match only whole words
  -x, --line-regexp         force PATTERN to 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 matches
  -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 the file name for each match
  -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 the part of a line matching PATTERN
  -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=FILE_PATTERN  search only files that match FILE_PATTERN
      --exclude=FILE_PATTERN  skip files and directories matching    FILE_PATTERN
      --exclude-from=FILE    skip files matching any file pattern from    FILE
      --exclude-dir=PATTERN  directories that match PATTERN will be    skipped.
  -L, --files-without-match  print only names of FILEs containing no    match
  -l, --files-with-matches  print only names of FILEs containing matches
  -c, --count                print only a count of matching 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)
  -u, --unix-byte-offsets    report offsets as if CRs were not there
                             (MSDOS/Windows)
'egrep' means 'grep -E'.  'fgrep' means 'grep -F'.
Direct invocation as either 'egrep' or 'fgrep' is deprecated.
When FILE is -, read standard input.  With no FILE, read . if a    command-line
-r is given, - otherwise.  If fewer than two FILEs are given, assume -h.
Exit status is 0 if any line is selected, 1 otherwise;
if any error occurs and -q is not given, the exit status is 2.
Report bugs to: bug-grep@gnu.org
GNU grep home page: <http://www.gnu.org/software/grep/>
General help using GNU software: <http://www.gnu.org/gethelp/>
The command "grep --help" exited with 0.
```

2.10   date

```
0.02s$ date --help
Usage: date [OPTION]... [+FORMAT]
  or:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.
Mandatory arguments to long options are mandatory for short options too.
  -d, --date=STRING          display time described by STRING, not 'now'
  -f, --file=DATEFILE        like --date; once for each line of DATEFILE
  -I[FMT], --iso-8601[=FMT]  output date/time in ISO 8601 format.
                               FMT='date' for date only (the default),
                               'hours', 'minutes', 'seconds', or 'ns'
                               for date and time to the indicated precision.
                               Example: 2006-08-14T02:34:56-0600
  -R, --rfc-2822             output date and time in RFC 2822 format.
                               Example: Mon, 14 Aug 2006 02:34:56 -0600
      --rfc-3339=FMT         output date/time in RFC 3339 format.
                               FMT='date', 'seconds', or 'ns'
                               for date and time to the indicated precision.
                               Example: 2006-08-14 02:34:56-06:00
  -r, --reference=FILE       display the last modification time of FILE
  -s, --set=STRING           set time described by STRING
  -u, --utc, --universal     print or set Coordinated Universal Time (UTC)
      --help     display this help and exit
      --version  output version information and exit
FORMAT controls the output.  Interpreted sequences are:
  %%   a literal %
  %a   locale's abbreviated weekday name (e.g., Sun)
  %A   locale's full weekday name (e.g., Sunday)
  %b   locale's abbreviated month name (e.g., Jan)
  %B   locale's full month name (e.g., January)
  %c   locale's date and time (e.g., Thu Mar  3 23:05:25 2005)
  %d   day of month (e.g., 01)
  %D   date; same as %m/%d/%y
  %e   day of month, space padded; same as %_d
  %F   full date; same as %Y-%m-%d
  %g   last two digits of year of ISO week number (see %G)
  %G   year of ISO week number (see %V); normally useful only with %V
  %h   same as %b
  %H   hour (00..23)
  %I   hour (01..12)
  %j   day of year (001..366)
  %k   hour, space padded ( 0..23); same as %_H
  %l   hour, space padded ( 1..12); same as %_I
  %m   month (01..12)
  %M   minute (00..59)
  %n   a newline
  %N   nanoseconds (000000000..999999999)
  %p   locale's equivalent of either AM or PM; blank if not known
  %P   like %p, but lower case
  %r   locale's 12-hour clock time (e.g., 11:11:04 PM)
  %R   24-hour hour and minute; same as %H:%M
  %s   seconds since 1970-01-01 00:00:00 UTC
```

```
%S    second (00..60)
%t    a tab
%T    time; same as %H:%M:%S
%u    day of week (1..7); 1 is Monday
%U    week number of year, with Sunday as first day of week (00..53)
%V    ISO week number, with Monday as first day of week (01..53)
%w    day of week (0..6); 0 is Sunday
%W    week number of year, with Monday as first day of week (00..53)
%x    locale's date representation (e.g., 12/31/99)
%X    locale's time representation (e.g., 23:13:48)
%y    last two digits of year (00..99)
%Y    year
%z    +hhmm numeric time zone (e.g., -0400)
%:z   +hh:mm numeric time zone (e.g., -04:00)
%::z  +hh:mm:ss numeric time zone (e.g., -04:00:00)
%:::z  numeric time zone with : to necessary precision (e.g., -04, +05:30)
%Z    alphabetic time zone abbreviation (e.g., EDT)
By default, date pads numeric fields with zeroes.
The following optional flags may follow '%':
  -   (hyphen) do not pad the field
  _   (underscore) pad with spaces
  0   (zero) pad with zeros
  ^   use upper case if possible
  #   use opposite case if possible
After any flags comes an optional field width, as a decimal number;
then an optional modifier, which is either
E to use the locale's alternate representations if available, or
O to use the locale's alternate numeric symbols if available.
Examples:
Convert seconds since the epoch (1970-01-01 UTC) to a date
  $ date --date='@2147483647'
Show the time on the west coast of the US (use tzselect(1) to find TZ)
  $ TZ='America/Los_Angeles' date
Show the local time for 9AM next Friday on the west coast of the US
  $ date --date='TZ="America/Los_Angeles" 09:00 next Fri'
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/date>
or available locally via: info '(coreutils) date invocation'
The command "date --help" exited with 0.
```

2.11   stat

```
0.02s$ stat --help
Usage: stat [OPTION]... FILE...
Display file or file system status.
Mandatory arguments to long options are mandatory for short options too.
  -L, --dereference     follow links
  -f, --file-system     display file system status instead of file status
  -c  --format=FORMAT   use the specified FORMAT instead of the default;
                          output a newline after each use of FORMAT
      --printf=FORMAT   like --format, but interpret backslash escapes,
                          and do not output a mandatory trailing newline;
```

```
                         if you want a newline, include \n in FORMAT
  -t, --terse            print the information in terse form
      --help     display this help and exit
      --version  output version information and exit
The valid format sequences for files (without --file-system):
  %a   access rights in octal (note '#' and '0' printf flags)
  %A   access rights in human readable form
  %b   number of blocks allocated (see %B)
  %B   the size in bytes of each block reported by %b
  %C   SELinux security context string
  %d   device number in decimal
  %D   device number in hex
  %f   raw mode in hex
  %F   file type
  %g   group ID of owner
  %G   group name of owner
  %h   number of hard links
  %i   inode number
  %m   mount point
  %n   file name
  %N   quoted file name with dereference if symbolic link
  %o   optimal I/O transfer size hint
  %s   total size, in bytes
  %t   major device type in hex, for character/block device special files
  %T   minor device type in hex, for character/block device special files
  %u   user ID of owner
  %U   user name of owner
  %w   time of file birth, human-readable; - if unknown
  %W   time of file birth, seconds since Epoch; 0 if unknown
  %x   time of last access, human-readable
  %X   time of last access, seconds since Epoch
  %y   time of last data modification, human-readable
  %Y   time of last data modification, seconds since Epoch
  %z   time of last status change, human-readable
  %Z   time of last status change, seconds since Epoch
Valid format sequences for file systems:
  %a   free blocks available to non-superuser
  %b   total data blocks in file system
  %c   total file nodes in file system
  %d   free file nodes in file system
  %f   free blocks in file system
  %i   file system ID in hex
  %l   maximum length of filenames
  %n   file name
  %s   block size (for faster transfers)
  %S   fundamental block size (for block counts)
  %t   file system type in hex
  %T   file system type in human readable form
NOTE: your shell may have its own version of stat, which usually supersedes
the version described here.  Please refer to your shell's documentation
for details about the options it supports.
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/stat>
```

or available locally via: info '(coreutils) stat invocation'
The command "stat --help" exited with 0.

0.03s$ bash --help
GNU bash, version 4.3.48(1)-release-(x86_64-pc-linux-gnu)
Usage: bash [GNU long option] [option] ...
 bash [GNU long option] [option] script-file ...
GNU long options:
 --debug
 --debugger
 --dump-po-strings
 --dump-strings
 --help
 --init-file
 --login
 --noediting
 --noprofile
 --norc
 --posix
 --rcfile
 --restricted
 --verbose
 --version
Shell options:
 -ilrsD or -c command or -O shopt_option   (invocation only)
 -abefhkmnptuvxBCHP or -o option
Type `bash -c "help set"' for more information about shell options.
Type `bash -c help' for more information about shell builtin commands.
Use the `bashbug' command to report bugs.
The command "bash --help" exited with 0.

2.12   bash -c "help set"

$ bash -c "help set"
set: set [-abefhkmnptuvxBCHP] [-o option-name] [--] [arg ...]
    Set or unset values of shell options and positional parameters.

    Change the value of shell attributes and positional parameters, or
    display the names and values of shell variables.

    Options:
      -a Mark variables which are modified or created for export.
      -b Notify of job termination immediately.
      -e Exit immediately if a command exits with a non-zero status.
      -f Disable file name generation (globbing).
      -h Remember the location of commands as they are looked up.
      -k All assignment arguments are placed in the environment for a
          command, not just those that precede the command name.
      -m Job control is enabled.
      -n Read commands but do not execute them.
      -o option-name

16

```
     Set the variable corresponding to option-name:
         allexport    same as -a
         braceexpand  same as -B
         emacs        use an emacs-style line editing interface
         errexit      same as -e
         errtrace     same as -E
         functrace    same as -T
         hashall      same as -h
         histexpand   same as -H
         history      enable command history
         ignoreeof    the shell will not exit upon reading EOF
         interactive-comments
                      allow comments to appear in interactive commands
         keyword      same as -k
         monitor      same as -m
         noclobber    same as -C
         noexec       same as -n
         noglob       same as -f
         nolog        currently accepted but ignored
         notify       same as -b
         nounset      same as -u
         onecmd       same as -t
         physical     same as -P
         pipefail     the return value of a pipeline is the status of
                      the last command to exit with a non-zero status,
                      or zero if no command exited with a non-zero status
         posix        change the behavior of bash where the default
                      operation differs from the Posix standard to
                      match the standard
         privileged   same as -p
         verbose      same as -v
         vi           use a vi-style line editing interface
         xtrace       same as -x
-p  Turned on whenever the real and effective user ids do not match.
    Disables processing of the $ENV file and importing of shell
    functions.  Turning this option off causes the effective uid and
    gid to be set to the real uid and gid.
-t  Exit after reading and executing one command.
-u  Treat unset variables as an error when substituting.
-v  Print shell input lines as they are read.
-x  Print commands and their arguments as they are executed.
-B  the shell will perform brace expansion
-C  If set, disallow existing regular files to be overwritten
    by redirection of output.
-E  If set, the ERR trap is inherited by shell functions.
-H  Enable ! style history substitution.  This flag is on
    by default when the shell is interactive.
-P  If set, do not resolve symbolic links when executing commands
    such as cd which change the current directory.
-T  If set, the DEBUG trap is inherited by shell functions.
--  Assign any remaining arguments to the positional parameters.
    If there are no remaining arguments, the positional parameters
    are unset.
```

-      Assign any remaining arguments to the positional parameters.
              The -x and -v options are turned off.

      Using + rather than - causes these flags to be turned off. The
      flags can also be used upon invocation of the shell. The current
      set of flags may be found in $-. The remaining n ARGs are positional
      parameters and are assigned, in order, to $1, $2, .. $n. If no
      ARGs are given, all shell variables are printed.

      Exit Status:
      Returns success unless an invalid option is given.
The command "bash -c "help set"" exited with 0.

2.13   bash -c help

```
0.01s$ bash -c help
GNU bash, version 4.3.48(1)-release (x86_64-pc-linux-gnu)
These shell commands are defined internally.  Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.
A star (*) next to a name means that the command is disabled.
 job_spec [&]                            history [-c] [-d offset] [n] or hist>
 (( expression ))                        if COMMANDS; then COMMANDS; [ elif C>
 . filename [arguments]                  jobs [-lnprs] [jobspec ...] or jobs >
 :                                       kill [-s sigspec | -n signum | -sigs>
 [ arg... ]                              let arg [arg ...]
 [[ expression ]]                        local [option] name[=value] ...
 alias [-p] [name[=value] ... ]          logout [n]
 bg [job_spec ...]                       mapfile [-n count] [-O origin] [-s c>
 bind [-lpsvPSVX] [-m keymap] [-f file>  popd [-n] [+N | -N]
 break [n]                               printf [-v var] format [arguments]
 builtin [shell-builtin [arg ...]]       pushd [-n] [+N | -N | dir]
 caller [expr]                           pwd [-LP]
 case WORD in [PATTERN [| PATTERN]...)>  read [-ers] [-a array] [-d delim] [->
 cd [-L|[-P [-e]] [-@]] [dir]            readarray [-n count] [-O origin] [-s>
 command [-pVv] command [arg ...]        readonly [-aAf] [name[=value] ...] o>
 compgen [-abcdefgjksuv] [-o option]  >  return [n]
 complete [-abcdefgjksuv] [-pr] [-DE] >  select NAME [in WORDS ... ;] do COMM>
 compopt [-o|+o option] [-DE] [name ..>  set [-abefhkmnptuvxBCHP] [-o option->
 continue [n]                            shift [n]
 coproc [NAME] command [redirections]    shopt [-pqsu] [-o] [optname ...]
 declare [-aAfFgilnrtux] [-p] [name[=v>  source filename [arguments]
 dirs [-clpv] [+N] [-N]                  suspend [-f]
 disown [-h] [-ar] [jobspec ...]         test [expr]
 echo [-neE] [arg ...]                   time [-p] pipeline
 enable [-a] [-dnps] [-f filename] [na>  times
 eval [arg ...]                          trap [-lp] [[arg] signal_spec ...]
 exec [-cl] [-a name] [command [argume>  true
 exit [n]                                type [-afptP] name [name ...]
 export [-fn] [name[=value] ...] or ex>  typeset [-aAfFgilrtux] [-p] name[=va>
 false                                   ulimit [-SHabcdefilmnpqrstuvxT] [lim>
```

```
fc [-e ename] [-lnr] [first] [last] o>   umask [-p] [-S] [mode]
fg [job_spec]                            unalias [-a] name [name ...]
for NAME [in WORDS ... ] ; do COMMAND>   unset [-f] [-v] [-n] [name ...]
for (( exp1; exp2; exp3 )); do COMMAN>   until COMMANDS; do COMMANDS; done
function name { COMMANDS ; } or name >   variables - Names and meanings of so>
getopts optstring name [arg]             wait [-n] [id ...]
hash [-lr] [-p pathname] [-dt] [name >   while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...]                { COMMANDS ; }
The command "bash -c help" exited with 0.
```

2.14   xargs

```
0.03s$ xargs --help
Usage: xargs [OPTION]... COMMAND [INITIAL-ARGS]...
Run COMMAND with arguments INITIAL-ARGS and more arguments read from input.
Mandatory and optional arguments to long options are also
mandatory or optional for the corresponding short option.
  -0, --null                  items are separated by a null, not whitespace;
                                disables quote and backslash processing and
                                logical EOF processing
  -a, --arg-file=FILE         read arguments from FILE, not standard input
  -d, --delimiter=CHARACTER   items in input stream are separated by CHARACTER,
                                not by whitespace; disables quote and backslash
                                processing and logical EOF processing
  -E END                      set logical EOF string; if END occurs as a line
                                of input, the rest of the input is ignored
                                (ignored if -0 or -d was specified)
  -e, --eof[=END]             equivalent to -E END if END is specified;
                                otherwise, there is no end-of-file string
  -I R                        same as --replace=R
  -i, --replace[=R]           replace R in INITIAL-ARGS with names read
                                from standard input; if R is unspecified,
                                assume {}
  -L, --max-lines=MAX-LINES   use at most MAX-LINES non-blank input lines per
                                command line
  -l[MAX-LINES]               similar to -L but defaults to at most one non-
                                blank input line if MAX-LINES is not specified
  -n, --max-args=MAX-ARGS     use at most MAX-ARGS arguments per command line
  -P, --max-procs=MAX-PROCS   run at most MAX-PROCS processes at a time
  -p, --interactive           prompt before running commands
      --process-slot-var=VAR  set environment variable VAR in child processes
  -r, --no-run-if-empty       if there are no arguments, then do not run COMMAND;
                                if this option is not given, COMMAND will be
                                run at least once
  -s, --max-chars=MAX-CHARS   limit length of command line to MAX-CHARS
      --show-limits           show limits on command-line length
  -t, --verbose               print commands before executing them
  -x, --exit                  exit if the size (see -s) is exceeded
      --help                  display this help and exit
      --version               output version information and exit
Please see also the documentation at http://www.gnu.org/software/findutils/.
You can report (and track progress on fixing) bugs in the "xargs"
```

program via the GNU findutils bug-reporting page at
https://savannah.gnu.org/bugs/?group=findutils or, if
you have no web access, by sending email to <bug-findutils@gnu.org>.
trThe command "xargs --help" exited with 0.

2.15   mv

```
$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
  or:  mv [OPTION]... SOURCE... DIRECTORY
  or:  mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.
Mandatory arguments to long options are mandatory for short options too.
      --backup[=CONTROL]       make a backup of each existing    destination file
  -b                           like --backup but does not accept an    argument
  -f, --force                  do not prompt before overwriting
  -i, --interactive            prompt before overwrite
  -n, --no-clobber             do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes    effect.
      --strip-trailing-slashes  remove any trailing slashes from each    SOURCE
                                 argument
  -S, --suffix=SUFFIX          override the usual backup suffix
  -t, --target-directory=DIRECTORY  move all SOURCE arguments into    DIRECTORY
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 move only when the SOURCE file is newer
                                 than the destination file or when the
                                 destination file is missing
  -v, --verbose                explain what is being done
  -Z, --context                set SELinux security context of    destination
                                 file to default type
      --help     display this help and exit
      --version  output version information and exit
The backup suffix is '~', unless set with --suffix or    SIMPLE_BACKUP_SUFFIX.
The version control method may be selected via the --backup option or    through
the VERSION_CONTROL environment variable.  Here are the values:
  none, off       never make backups (even if --backup is given)
  numbered, t     make numbered backups
  existing, nil   numbered if numbered backups exist, simple otherwise
  simple, never   always make simple backups
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/mv>
or available locally via: info '(coreutils) mv invocation'
The command "mv --help" exited with 0.
```


2.16   chmod --help

$ sudo chmod --help Usage: chmod [OPTION]... MODE[,MODE]... FILE... or: chmod [OPTION]...
OCTAL-MODE FILE... or: chmod [OPTION]... --reference=RFILE FILE... Change the mode of
each FILE to MODE. With --reference, change the mode of each FILE to that of RFILE. -c, --changes
like verbose but report only when a change is made -f, --silent, --quiet suppress most error messages
-v, --verbose output a diagnostic for every file processed --no-preserve-root do not treat '/' specially

(the default) --preserve-root fail to operate recursively on '/' --reference=RFILE use RFILE's mode instead of MODE values -R, --recursive change files and directories recursively --help display this help and exit --version output version information and exit Each MODE is of the form '[ugoa]*([-+=]([rwxXst]*|[ugo]))+'. GNU coreutils online help: <http://www.gnu.org/software/coreutils/> Full documentation at: <http://www.gnu.org/software/coreutils/chmod> or available locally via: info '(coreutils) chmod invocation'

3 Linux 常用命令大全

Linux 基础知识——Linux 常用命令大全

3.1 创建目录 mkdir

作用：在当前目录下创建下一级目录，无法跨级创建

常用参数
-p 创建多级目录（跨级创建）
-v 查看目录创建的过程（创建目录可视化）

3.2 删除文件 rmdir

仅可以删除空白目录（不可以删除包含内容的目录）

3.3 创建文件 touch

作用：创建空白文件

3.4 删除文件或目录 rm

1、删除文件
rm 文件名（删除时会询问是否删除）
rm -f 文件名（强制删除）
rm -v 文件名（可视化删除）

2、删除目录
rm -r 目录名（删除时会询问是否删除）
rm -rf 目录名（强制删除，若目录不存在，此命令依旧可以执行，不报错）
rm -rv 目录名（可视化强制）
删除目录和文件时，先删除文件在删除目录

rm的用法如下：
1、删除文件夹以及文件夹中的所有文件命令：
rm -rf 目录名字
其中：
-r：向下递归删除
-f：直接强行删除，且没有任何提示
2、删除文件命令

rm -f 文件名
将会强行删除文件，且无提示
注意：
使用rm -rf要格外注意，linux中没有回收站，慎重删除

如果空目录就可以用rmdir
如果是有文件的目录就用 rm -f
一般文件用 rm

3.5 复制文件或目录（可以对目标文件或目录重命名）cp

源文件始终不变，仅仅是对目标文件进行改变。

1、复制文件
格式：cp 源文件 目标文件

2、拷贝目录（目录需要加/）注意区分绝对路径和相对路径
格式：cp -r 源目录 目标目录

3.6 移动（类似于 Windows 中的剪切）mv

注意与复制命令cp的区别。mv命令使源文件的状态发生改变。

1、移动目录时：
若果目录存在，则会将原目录移动到目标目录下；如果目录不存在，则相当于移动并重命名

3.7 查看文件内容 cat tac more less head tail

4 Linux 命令

Linux 命令

4.1 wget

Linux wget 命令详解

Linux 命令

wget 是一个下载文件的工具，它用在命令行下。

使用 wget -O 下载并以不同的文件名保存 (-O：下载文件到对应目录，并且修改文件名称)

wget -O wordpress.zip http://www.minjieren.com/download.aspx?id=1080
wget https://github.com/jgm/pandoc/releases/download/1.17.1/pandoc-1.17.1-2-amd64.deb

使用 wget -b 后台下载

wget -b <a href="http://www.minjieren.com/wordpress-3.1-zh_CN.zip">http://www.minjieren.com/w

备注： 你可以使用以下命令来察看下载进度：tail -f wget-log

利用-spider: 模拟下载，不会下载，只是会检查是否网站是否好着

```
wget --spider  www.baidu.com #不下载任何文件
```

4.2 gsub 函数

gsub 函数则使得在所有正则表达式被匹配的时候都发生替换

```
gsub(regular expression, subsitution string, target string);
```
简称 gsub（r,s,t）

4.3 sub 和 gsub 的区别

sub 匹配第一次出现的符合模式的字符串，相当于 sed 's//'。gsub 匹配所有的符合模式的字符串，相当于 sed 's//g'。例如：

```
awk '{sub(/Mac/,"Macintosh");print}' urfile 用Macintosh替换Mac
awk '{sub(/Mac/,"MacIntosh",$1); print}' file 第一个域内用
```

Macintosh 替换 Mac 把上面 sub 换成 gsub 就表示在满足条件得域里面替换所有的字符。

awk 的 sub 函数用法：

sub 函数匹配指定域/记录中最大、最靠左边的子字符串的正则表达式，并用替换字符串替换这些字符串。如果没有指定目标字符串就默认使用整个记录。替换只发生在第一次匹配的时候。格式如下：

```
sub (regular expression, substitution string):
sub (regular expression, substitution string, target string)
```

实例：

```
$ awk '{ sub(/test/, "mytest"); print }' testfile
$ awk '{ sub(/test/, "mytest", $1); print }' testfile
```

第一个例子在整个记录中匹配，替换只发生在第一次匹配发生的时候。第二个例子在整个记录的第一个域中进行匹配，替换只发生在第一次匹配发生的时候。

如要在整个文件中进行匹配需要用到 gsub

4.4 awk gawk

Linux awk 命令详解

linux gawk 命令

LinuxShell 编程之 gawk 详解

awk 是一个强大的文本分析工具，相对于 grep 的查找，sed 的编辑，awk 在其对数据分析并生成报告时，显得尤为强大。简单来说 awk 就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

使用方法：awk '{pattern + action}' {filenames}

尽管操作可能会很复杂，但语法总是这样，其中 pattern 表示 AWK 在数据中查找的内容，而 action 是在找到匹配内容时所执行的一系列命令。花括号（{}）不需要在程序中始终出现，但它们用于根据特定的模式对一系列指令进行分组。pattern 就是要表示的正则表达式，用斜杠括起来。

awk 语言的最基本功能是在文件或者字符串中基于指定规则浏览和抽取信息，awk 抽取信息后，才能进行其他文本操作。完整的 awk 脚本通常用来格式化文本文件中的信息。通常，awk 是以文件的一行为处理单位的。awk 每接收文件的一行，然后执行相应的命令，来处理文本。

gawk **命令格式**

Usage: gawk [POSIX or GNU styleoptions] -f progfile [--] file ...

Usage: gawk [POSIX or GNU styleoptions] [--] 'program' file ...

gawk **选项**

-F fs	指定描绘一行中数据字段的文件分隔符
-f file	指定读取程序的文件名
-v var=value	定义 gawk 程序中使用的变量和默认值
-mf N	指定数据文件中要处理的字段的最大数目
-mr N	指定数据文件中的最大记录大小
-W keyword	指定 gawk 的兼容模式或警告级别

gawk **的主要功能之一是其处理文本文件中数据的能力。它通过自动将变量分配给每行中的每个数据元素实现这一功能。默认情况下，** gawk **将下面的变量分配给在文本行中检测到的每个数据字段：**

$0	表示整行文本
$1	表示文本行中的第一个数据字段
$2	表示文本行中的第二个数据字段
$n	表示文本行中的第 n 个数据字段

各数据字段依据文本行中的字段分隔符确定。 gawk **读取一行文本时，使用定义的字段分隔符描述各数据字段。** gawk **的默认字段分隔符是任意空白字符（如制表符或空格符）**

4.5 find

Linux-find **命令详解**

在目录结构中搜索文件，并执行指定的操作。 Linux **下** find **命令提供了相当多的查找条件，功能很强大**

find **命令格式：**

find path -option 【-print】【-exec -ok |xargs |grep】【command {} \; 】

Linux **下** find **命令在目录结构中搜索文件，并执行指定的操作。** Linux **下** find **命令提供了相当多的查找条件，功能很强大** find **常见命令参数**

4.5.1 **命令选项：**

```
-name    按照文件名查找文件。
-perm    按照文件权限来查找文件。
-user    按照文件属主来查找文件。
-group   按照文件所属的组来查找文件。
-mtime -n +n 按照文件的更改时间来查找文件 【-7 7天之内 +7 7天前】
-nogroup  查找无效属组的文件，即该文件所属的组在/etc/groups中不存在。
-nouser  查找无效属主的文件，即该文件的属主在/etc/passwd中不存在。
-newer file1 ! file2 查找更改时间比文件file1新但比文件file2旧的文件。
-type  查找某一类型的文件，诸如：
        b - 块设备文件。
        d - 目录。
```

```
          c - 字符设备文件。
          p - 管道文件。
          l - 符号链接文件。
          f - 普通文件。
```
-size n：[c] 查找文件长度为n块的文件，带有c表示文件长度以字节计。
-depth：在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找。
-follow：如果find命令遇到符号链接文件，就跟踪至链接所指向的文件。
另外，下面三个的区别：
-amin n   查找系统中最后N分钟访问的文件
-atime n   查找系统中最后n*24小时访问的文件
-cmin n   查找系统中最后N分钟被改变文件状态的文件
-ctime n   查找系统中最后n*24小时被改变文件状态的文件
-mmin n   查找系统中最后N分钟被改变文件数据的文件
-mtime n   查找系统中最后n*24小时被改变文件数据的文件

4.5.2 常用的命令展示

4.5.3 查找普通文件/目录

```
find /home/omd -type f  （普通文件）
find /home/omd -type d  （查询目录）
```

4.5.4 只显示 1 级目录文件且过滤自身

```
find ./ -maxdepth 1  -type d  ! -name "hhh"
```

4.5.5 查找一天内被访问过 (access) 的文件

```
find /home/omd/ -atime -1 -type f
```

4.5.6 查询 inode 相同的文件

:: find / -inum inode 数字

4.5.7 除了某个文件以为，其余的均删除

```
find /home/omd/ -type f ! -name h.txt  | xargs  rm   f
ls | grep -v "h.txt" |xargs rm -rf （与上面类似，删除除了某个文件外的所有文件）
```

4.5.8 删除目录下所有文件

```
find /tmp/ -type f -exec rm -rf {} \;
find /tmp/ -type f | xargs rm -rf
```

4.5.9 查看当前路径下所有文件的信息：

```
find /tmp/ -type f ! -name a |xargs rm    rf
find ./ -type f -exec file {} \;
```

查找指定时间内修改过的文件 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

```
# 当前路径下访问文件超过2分钟文件
find ./ -amin +2
# 当前路径下访问文件刚好2分钟的文件
find ./ -amin 2
find ./ -cmin +2
find ./ -mmin +2
find ./ -mtime +2
find ./ -ctime +2
find ./ -mtime +2
find ./ -ctime +2
find / -ctime  +20    最近修改文件时间20分钟以前
find / -mtime  +7     修改文件为7天之前的(最重要)
find / -mtime  7      修改文件为第7天，就是往前推7天
find / -mtime  -7     修改文件为7天之内的
```

4.5.10 按照目录或文件的权限来查找文件

```
find /opt -perm 777
```

4.5.11 按大小查找文件

```
find / -size +10M  |sort 【查找大于10M的文件】
find / -size -10M  |sort 【查找小于10M的文件】
find / -size 10M   |sort  【查找10M的文件】
```

在 test 目录下查找不在 test4 子目录之内的所有文件 ^^^

```
find ./test -path "test/test4" -prune -o -print
```
【可以使用-prune选项来指出需要忽略的目录。在使用-prune选项时要当心，因为如果你同时使用了-dep

4.5.12 查找比 yum.log 但不比 hhh.txt 新的文件

```
[root@localhost ftl]# find / newer /var/log/yum.log ! -newer ./hhh.txt
```

查找更改时间在比 log2012.log 文件新的文件 ^^

```
find ./ -newer log2012.log
```

4.5.13 在当前目录下查找文件长度大于 1 M 字节的文件

```
find ./ -size +1000000c   print
find ./   size +1M -print
```

4.5.14 在/home/apache 目录下查找文件长度恰好为 100 字节的文件

find /home/apache -size 100c -print

4.5.15 在当前目录下查找长度超过 10 块的文件

```
find . -size 10   print
```

4.5.16 其他命令：

```
find /home/omd/ -name *.txt | while read line; do cp $line /home/omd/h;done
for name in `chkconfig | grep 3:on |awk '{print $1}'` ; do echo $name >> h.txt; done;
find /home/omd/ -name *.txt | xargs -i cp {} /home/omd/h
cat /home/omd/h/he.txt | while read line; do echo $line >> /home/omd/h.txt ; done;
cat /home/omd/h.txt | awk 'BEGIN{print "Name "} {print $1}'
cat /home/omd/h.txt | xargs -I {} cat {}
find . -name  "*.txt" |xargs   sed -i 's/hhhh/\hHHh/g'
```

4.5.17 find 命令之 execokprint

ls -l 命令放在 find 命令的-exec 选项中

```
find . -type f -exec ls -l {} \; 【 {}   花括号代表前面find查找出来的文件名】
```

4.5.18 在目录中查找更改时间在 n 日以前的文件并删除它们

```
find ./ -mtime +10 -exec rm {} \;
```

4.5.19 在目录中查找更改时间在 n 日以前的文件并删除它们，在删除之前先给出提示

```
find / -mtime +1 -a -name "*.log" -type f -ok cp {} /tmp/ftl \; 【 -ok是安全模式，根exec效果同】
```

4.5.20 exec 中使用 grep 命令

```
find /etc -name "passwd*" -exec grep "root" {} \; 【过滤文件内容用】
```

4.5.21 查找文件移动到指定目录

```
find . -name "*.log" -exec mv {} .. \;
```

4.5.22 用 exec 选项执行 cp 命令

```
find . -name "*.log" -exec cp {} test3 \;
```

4.6 linux-xargs-命令

xargs 是给命令传递参数的一个过滤器，也是组合多个命令的一个工具。

xargs 可以将管道或标准输入（stdin）数据转换成命令行参数，也能够从文件的输出中读取数据。

xargs 也可以将单行或多行文本输入转换为其他格式，例如多行变单行，单行变多行。

xargs 默认的命令是 echo，这意味着通过管道传递给 xargs 的输入将会包含换行和空白，不过通过 xargs 的处理，换行和空白将被空格取代。

xargs 是一个强有力的命令，它能够捕获一个命令的输出，然后传递给另外一个命令。

之所以能用到这个命令，关键是由于很多命令不支持 | 管道来传递参数，而日常工作中有有这个必要，所以就有了

xargs 命令，例如：

```
find /sbin -perm +700 |ls -l        #这个命令是错误的
find /sbin -perm +700 |xargs ls -l    #这样才是正确的
```

xargs 一般是和管道一起使用。

4.6.1 命令格式：

```
somecommand |xargs -item  command
```

4.6.2 参数：

- -a file 从文件中读入作为 sdtin
- -e flag ，注意有的时候可能会是-E，flag 必须是一个以空格分隔的标志，当 xargs 分析到含有 flag 这个标志的时候就停止。
- -p 当每次执行一个 argument 的时候询问一次用户。
- -n num 后面加次数，表示命令在执行的时候一次用的 argument 的个数，默认是用所有的。
- -t 表示先打印命令，然后再执行。
- -i 或者是-I，这得看 linux 支持了，将 xargs 的每项名称，一般是一行一行赋值给 {}，可以用 {} 代替。
- -r no-run-if-empty 当 xargs 的输入为空的时候则停止 xargs，不用再去执行了。
- -s num 命令行的最大字符数，指的是 xargs 后面那个命令的最大命令行字符数。
- -L num 从标准输入一次读取 num 行送给 command 命令。
- -l 同 -L。
- -d delim 分隔符，默认的 xargs 分隔符是回车，argument 的分隔符是空格，这里修改的是 xargs 的分隔符。
- -x exit 的意思，主要是配合-s 使用。。
- -P 修改最大的进程数，默认是 1，为 0 时候为 as many as it can，这个例子我没有想到，应该平时都用不到的吧。

4.6.3 实例

4.6.3.1 xargs 用作替换工具，读取输入数据重新格式化后输出。   定义一个测试文件，内有多行文本数据：

```
# cat test.txt

a b c d e f g
```

```
h i j k l m n
o p q
r s t
u v w x y z
```

多行输入单行输出：

```
# cat test.txt | xargs
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

-n 选项多行输出：

```
# cat test.txt | xargs -n3

a b c
d e f
g h i
j k l
m n o
p q r
s t u
v w x
y z
```

-d 选项可以自定义一个定界符：

```
# echo "nameXnameXnameXname" | xargs -dX

name name name name
```

结合 -n 选项使用：

```
# echo "nameXnameXnameXname" | xargs -dX -n2

name name
name name
```

读取 stdin，将格式化后的参数传递给命令

假设一个命令为 sk.sh 和一个保存参数的文件 arg.txt：

```
#!/bin/bash
#sk.sh命令内容，打印出所有参数。

echo $*
```

arg.txt 文件内容：

```
# cat arg.txt

aaa
bbb
ccc
```

4.6.3.2 xargs 的一个选项 -I {} xargs 的一个选项 -I，使用 -I 指定一个替换字符串 {}，这个字符串在 xargs 扩展时会被替换掉，当 -I 与 xargs 结合使用，每一个参数命令都会被执行一次：

```
# cat arg.txt | xargs -I {} ./sk.sh -p {} -l
```

```
-p aaa -l
-p bbb -l
-p ccc -l
```

复制所有图片文件到 /data/images **目录下**：

```
ls *.jpg | xargs -n1 -I {} cp {} /data/images
```

4.6.3.3 xargs 结合 find 使用 用 rm 删除太多的文件时候，可能得到一个错误信息：/bin/rm Argument list too long. 用 xargs 去避免这个问题：

```
find . -type f -name "*.log" -print0 | xargs -0 rm -f
```

xargs -0 **将** \0 **作为定界符。**

统计一个源代码目录中所有 php **文件的行数：**

```
find . -type f -name "*.php" -print0 | xargs -0 wc -l
```

查找所有的 jpg **文件，并且压缩它们：**

```
find . -type f -name "*.jpg" -print | xargs tar -czvf images.tar.gz
```

4.6.3.4 xargs 其他应用 假如你有一个文件包含了很多你希望下载的 URL，**你能够使用** xargs **下载所有链接：**

```
# cat url-list.txt | xargs wget -c
```

4.7 Linux 系统下 date 常用命令的参数以及获取时间戳的方法

date: **用于显示/设置系统的时间或者日期：** date **选项 + 指定的格式：**

+：**进行格式化输出**
%Y:**表示年份**
%m:**表示月份**
%d:**表示第几天**
%H:**表示小时**
%M:**表示分钟**
%S:**表示秒钟**
查看当前的系统时间： date
设置系统时间为:date -s "20180316 16:53:10"
查看本地系统时间： date "+%Z"
查看星期几： date "+%A"
输入当前是上午还是下午： date "+%p"
判断今天是一年中的第几天： date "+%j"
ctrl+l:**清屏操作，相当于**clear
等价一： date + %Y-%m-%d=date + %F
等价二： date + %H :%M :%S=date + %T
等价三： date + "%F %T" =date + '%F %T' （**注意：有空格需要用到双引号或单引号**）

时间戳：时间戳是指格林威治时间自1970**年**1**月**1**日（**00:00:00 GMT**）至当前时间的总秒数。它也被称为U**

时间->时间戳： date +%s
时间戳->时间： date +%Y:%m:%d -d @1425384141
Unix**时间戳（英文为**Unix epoch, Unix time, POSIXme 或 Unix timestamp）**是从**1970**年**1**月**1**日（**UTC/GMT
misc

4.8 cp 命令详解

Linux-cp 命令详解

默认情况下，如果目标文件存在，它将被覆盖。-n 选项告诉 cp 不要覆盖现有文件。要提示确认，请使用该 -i 选项。

```
cp -i file.txt file_backup.txt
```

如果要仅在文件比目标更新时复制文件，请使用以下 -u 选项：

```
cp -u file.txt file_backup.txt
```

另一个可能有用的选项是 -v，他告诉 cp 打印详细输出：

```
cp -v file.txt file_backup.txt
'file.txt' -> 'file_backup.txt'
```

使用 cp 命令复制目录要复制目录 (包括其所有文件和子目录)，请使用 -R 或 -r 选项。在以下示例中，我们将目录复制 Pictures 到 Pictures_backup：

```
cp -R 源目录 目标目录
```

要仅复制文件和子目录，而不复制目标目录，请使用以下 -t 选项 (原版有错，不能用-T)：

```
cp -Rt 目标目录 源目录
```

另一种只复制目录内容而不是目录本身的方法是使用通配符 (*)。以下命令的缺点是它不会复制隐藏文件和目录 (以点. 开头的文件和目录)：

```
cp -Rt 目标目录 源目录/*
```

4.9 拷贝命令比较，XCOPY(win) VS cp(linux)

windows 下 XCOPY 命令，目标目录的父目录可以不存在，命令自己会创建

Linux 下 cp 不会自动创建目标目录的父目录，如果目标目录不在在会直接报错。

4.10 gnumake-wildcard(win) VS cp(linux)

windows 下 gnumake 命令 wildcard 返回匹配文件名带目录（待确认）

Linux 下 gnumake 命令 wildcard 返回匹配文件名带目录（已确认）

4.11 touch 命令直接创建空白文件

Linux Touch 命令的 8 种常见使用方法

touch test.txt

命令为："touch [选项] [文件]"。

```
-a     只更改访问时间
-c, --no-create 不创建任何文件
-d, --date=字符串 使用指定字符串表示时间而非当前时间
-f     (忽略)
-h, --no-dereference   会影响符号链接本身，而非符号链接所指示的目的地
   (当系统支持更改符号链接的所有者时，此选项才有用)
-m     只更改修改时间
```

```
-r, --reference=FILE  use this file's times instead of current time
-t STAMP              use [[CC]YY]MMDDhhmm[.ss] instead of current time
    --time=WORD           change the specified time:
                              WORD is access, atime, or use: equivalent to -a
                              WORD is modify or mtime: equivalent to -m
    --help    显示此帮助信息并退出
    --version   显示版本信息并退出
```

4.12 Linux 文件三种时间属性 atime/mtime/ctime：

atime(access time): 最近访问文件内容时间（Last Access Time）。

mtime(modify time): 最近修改文件内容时间（Last Modification Time）。

ctime(change time): 最近更改文件属性（Inode 内容更改）的时间，包括文件名、大小、内容、权限、属主、属组等（Last Change Time）。

1. 输入 "touch filetime.txt" 创建新文件，输入 "stat filetime.txt" 即可查看文件 filetime.txt 的时间属性。

   备注：新创建文件的三种时间抓取当前时间，本例中为 2019-01-05 19:42:36。

   Birth 时间为空，Linux 需要内核提供 xstat() 接口才可获取 Birth 时间。

2. 使用 cat，less，more 等命令查看文件后 atime 已更新（2019-01-05 19:44:13）。

   备注：ls，stat 命令不会修改 atime。

3. 输入 "echo "add test"»filetime.txt" 给文件增加内容 "add test" 后，输入 "stat filetime.txt" 查看时间属性，发现 mtime 和 ctime 均已更新（2019-01-05 19:55:05）。

4. 输入 "mv filetime.txt new.txt" 修改文件名为 new.txt，输入 "stat new.txt" 查看时间属性，发现只有 ctime 更新（2019-01-05 19:57:05）。

   备注：chown 和 chmod 命令均修改 ctime，ln（不包括 ln -s）亦修改 ctime。

5. 输入 "ls -lc new.txt" 可查看文件 new.txt 的 ctime。

6. 输入 "ls -lu new.txt" 可查看文件 new.txt 的 atime。

7. 输入 "ls -l new.txt" 可查看文件 new.txt 的 mtime。

4.13 利用 date 时间戳 <-> 时间

时间戳：时间戳是指格林威治时间自 1970 年 1 月 1 日（00:00:00 GMT）至当前时间的总秒数。它也被称为 Unix 时间戳（Unix Timestamp）。通俗的讲，时间戳是一份能够表示一份数据在一个特定时间点已经存在的完整的可验证的数据。

时间-> 时间戳：date +%s

时间戳-> 时间：date +%Y:%m:%d -d @1425384141

Unix 时间戳（英文为 Unix epoch, Unix time, POSIXme 或 Unix timestamp）是从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数，不考虑闰秒。

4.14 sed 命令功能强大替换

一、基本的替换：

命令格式1：sed 's/原字符串/新字符串/' 文件
命令格式2：sed 's/原字符串/新字符串/g' 文件

这两种命令格式的区别在于是否有个 "g"。没有 "g" 表示只替换第一个匹配到的字符串，有 "g" 表示替换所有能匹配到的字符串

二、替换某行内容:

命令格式1：sed '行号c 新字符串' 文件
命令格式2：sed '起始行号，终止行号c 新字符串' 文件

第一个命令表示用新的字符串替换指定这一行的内容，第二个命令表示用新字符串替换指定几行的内容。如下图，第一个命令将第 2 行内容替换成了 "new test!"，第二个命令将第 2 到 6 行替换成了 "new test!"。

三、多条件替换

命令格式：sed -e 命令1 -e 命令2 -e 命令3

有些时候有多个替换条件，那就可以使用 "-e" 参数将这些替换条件连接起来，一次性完成所有的替换操作。例如，可以将上述的两种命令连接起来："sed -e 's/原字符串/新字符串/' '行号 c 新字符串' 文件"。如下图，不仅将小写 "a" 替换成了大写 "A"，还将第 2 行内容替换成了 "new test!"。

四、保存替换结果到文件中

命令格式：sed -i 命令

上述这些命令都只是将替换结果打印到屏幕上，如果想保存结果到文件中，就需要加上 "-i" 参数。