gramma and template

kevinluo

Contents

contents

1 網絡資源地址

Linux 基礎知識——Linux 常用命令大全

Linux 命令大全

在 Linux 下查看文件三種時間

Linux 系統下 date 常用命令的參數以及獲取時間戳的方法

如何使用 Linux sed 命令進行字符串替換

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.
```

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 $TRAVIS_BUILD_DIR/output/sphinx/build-memo/* /tmp/klgit/gp-memo" exited wi
0.00s$ pwd
/tmp/klgit/gp-memo
```

上面cp命令,錯在：
- 不能用大寫T，這是表示 DEST是文件，不是目錄，報錯的原因
更正：

```
cp -rt /tmp/klgit/gp-memo $TRAVIS_BUILD_DIR/output/sphinx/build-memo/*
```

注意： -rt指定目標目錄時要緊跟，所以如果參數寫在前面，則目標目錄也到前面了。
source目錄後面帶星通配和-r配合使用，則表示衹copy文件和子目錄。

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
```

```
-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
          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]*[-+=][0-7]+’. 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 基礎知識——Linux 常用命令大全

3.1 創建目錄 mkdir

作用：在當前目錄下創建下一級目錄，無法跨級創建

常用參數
-p 創建多級目錄（跨級創建）
-v 查看目錄創建的過程（創建目錄可視化）

3.2 刪除文件 rmdir

僅可以刪除空白目錄（不可以刪除包含內容的目錄）

3.3 創建文件 touch

作用：創建空白文件

3.4 刪除文件或目錄 rm

1、刪除文件
rm 文件名（刪除時會詢問是否刪除）
rm -f 文件名（强制刪除）
rm -v 文件名（可視化刪除）

2、刪除目錄
rm -r 目錄名（刪除時會詢問是否刪除）
rm -rf 目錄名（强制刪除，若目錄不存在，此命令依舊可以執行，不報錯）
rm -rv 目錄名（可視化强制）
刪除目錄和文件時，先刪除文件在刪除目錄

rm的用法如下：
1、刪除文件夾以及文件夾中的所有文件命令：
rm -rf 目錄名字
其中：
-r：向下遞歸刪除
-f：直接强行刪除，且沒有任何提示
2、刪除文件命令

rm -f 文件名
將會强行刪除文件，且無提示
注意：
使用rm -rf要格外注意，linux中沒有回收站，慎重刪除

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

3.5 復制文件或目錄（可以對目標文件或目錄重命名）cp

源文件始終不變，僅僅是對目標文件進行改變。

1、復制文件
格式：cp 源文件 目標文件

2、拷貝目錄（目錄需要加/）注意區分絕對路徑和相對路徑
格式：cp -r 源目錄 目標目錄

3.6 移動（類似于 Windows 中的剪切）mv

注意與復制命令cp的區別。mv命令使源文件的狀態發生改變。

1、移動目錄時：
若果目錄存在，則會將原目錄移動到目標目錄下；如果目錄不存在，則相當于移動并重命名

3.7 查看文件內容 cat tac more less head tail

4 Linux 命令

Linux 命令

4.1 wget

Linux wget 命令詳解

Linux 命令

wget 是一個下載文件的工具，它用在命令行下。

使用 wget -O 下載并以不同的文件名保存 (-O：下載文件到對應目錄，并且修改文件名稱)

```
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 後臺下載

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

備注： 你可以使用以下命令來察看下載進度：tail -f wget-log

利用-spider: 模擬下載，不會下載，祇是會檢查是否網站是否好着

```
wget --spider  www.baidu.com #不下載任何文件
```

4.2 gsub 函數

gsub 函數則使得在所有正則表達式被匹配的時候都發生替換

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

4.3 sub 和 gsub 的區別

sub 匹配第一次出現的符合模式的字符串，相當于 sed 's//' 。gsub 匹配所有的符合模式的字符串，相當于 sed 's//g' 。例如：

```
awk '{sub(/Mac/,"Macintosh");print}' urfile 用Macintosh替換Mac
awk '{sub(/Mac/,"MacIntosh",$1); print}' file 第一個域內用
```

Macintosh 替換 Mac 把上面 sub 換成 gsub 就表示在滿足條件得域裏面替換所有的字符。

awk 的 sub 函數用法：

sub 函數匹配指定域/記錄中最大、最靠左邊的子字符串的正則表達式，并用替換字符串替換這些字符串。如果沒有指定目標字符串就默認使用整個記錄。替換衹發生在第一次匹配的時候。格式如下：

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

實例：

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

第一個例子在整個記錄中匹配，替換衹發生在第一次匹配發生的時候。第二個例子在整個記錄的第一個域中進行匹配，替換衹發生在第一次匹配發生的時候。

如要在整個文件中進行匹配需要用到 gsub

4.4 awk gawk

Linux awk 命令詳解

linux gawk 命令

LinuxShell 編程之 gawk 詳解

awk 是一個强大的文本分析工具，相對于 grep 的查找，sed 的編輯，awk 在其對數據分析并生成報告時，顯得尤爲强大。簡單來說 awk 就是把文件逐行的讀入，以空格爲默認分隔符將每行切片，切開的部分再進行各種分析處理。

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

盡管操作可能會很復雜，但語法總是這樣，其中 pattern 表示 AWK 在數據中查找的內容，而 action 是在找到匹配內容時所執行的一系列命令。花括號（{}）不需要在程序中始終出現，但它們用于根據特定的模式對一系列指令進行分組。pattern 就是要表示的正則表達式，用斜杠括起來。

awk 語言的最基本功能是在文件或者字符串中基于指定規則瀏覽和抽取信息，awk 抽取信息後，才能進行其他文本操作。完整的 awk 脚本通常用來格式化文本文件中的信息。通常，awk 是以文件的一行爲處理單位的。awk 每接收文件的一行，然後執行相應的命令，來處理文本。

gawk 命令格式

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

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

gawk 選項

-F fs	指定描繪一行中數據字段的文件分隔符
-f file	指定讀取程序的文件名
-v var=value	定義 gawk 程序中使用的變量和默認值
-mf N	指定數據文件中要處理的字段的最大數目
-mr N	指定數據文件中的最大記錄大小
-W keyword	指定 gawk 的兼容模式或警告級別

gawk 的主要功能之一是其處理文本文件中數據的能力。它通過自動將變量分配給每行中的每個數據元素實現這一功能。默認情況下，gawk 將下面的變量分配給在文本行中檢測到的每個數據字段：

$0	表示整行文本
$1	表示文本行中的第一個數據字段
$2	表示文本行中的第二個數據字段
$n	表示文本行中的第 n 個數據字段

各數據字段依據文本行中的字段分隔符確定。gawk 讀取一行文本時，使用定義的字段分隔符描述各數據字段。gawk 的默認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 命令詳解

在目錄結構中搜索文件，并執行指定的操作。Linux 下 find 命令提供了相當多的查找條件，功能很強大

find 命令格式：

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

Linux 下 find 命令在目錄結構中搜索文件，并執行指定的操作。Linux 下 find 命令提供了相當多的查找條件，功能很強大 find 常見命令參數

4.5.1 命令選項：

```
-name   按照文件名查找文件。
-perm   按照文件權限來查找文件。
-user   按照文件屬主來查找文件。
-group  按照文件所屬的組來查找文件。
-mtime -n +n 按照文件的更改時間來查找文件 【-7 7天之內 +7 7天前】
-nogroup  查找無效屬組的文件，即該文件所屬的組在/etc/groups中不存在。
-nouser  查找無效屬主的文件，即該文件的屬主在/etc/passwd中不存在。
-newer file1 ! file2 查找更改時間比文件file1新但比文件file2舊的文件。
-type  查找某一類型的文件，諸如：
         b - 塊設備文件。
         d - 目錄。
```

c - 字符設備文件。
p - 管道文件。
l - 符號鏈接文件。
f - 普通文件。

-size n：[c] 查找文件長度爲n塊的文件，帶有c表示文件長度以字節計。
-depth：在查找文件時，首先查找當前目錄中的文件，然後再在其子目錄中查找。
-follow：如果find命令遇到符號鏈接文件，就跟踪至鏈接所指向的文件。
另外,下面三個的區別:

-amin n　查找系統中最後N分鐘訪問的文件
-atime n　查找系統中最後n*24小時訪問的文件
-cmin n　查找系統中最後N分鐘被改變文件狀態的文件
-ctime n　查找系統中最後n*24小時被改變文件狀態的文件
-mmin n　查找系統中最後N分鐘被改變文件數據的文件
-mtime n　查找系統中最後n*24小時被改變文件數據的文件

4.5.2 常用的命令展示

4.5.3 查找普通文件/目錄

```
find /home/omd -type f （普通文件）
find /home/omd -type d （查詢目錄）
```

4.5.4 祇顯示 1 級目錄文件且過濾自身

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

4.5.5 查找一天內被訪問過 (access) 的文件

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

4.5.6 查詢 inode 相同的文件

:: find / -inum inode 數字

4.5.7 除了某個文件以爲，其餘的均刪除

```
find /home/omd/ -type f ! -name h.txt  | xargs  rm   f
ls | grep -v "h.txt" |xargs rm -rf（與上面類似，刪除除了某個文件外的所有文件）
```

4.5.8 刪除目錄下所有文件

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

4.5.9 查看當前路徑下所有文件的信息：

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

查找指定時間內修改過的文件 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

```
# 當前路徑下訪問文件超過2分鐘文件
find ./ -amin +2
# 當前路徑下訪問文件剛好2分鐘的文件
find ./ -amin 2
find ./ -cmin +2
find ./ -mmin +2
find ./ -mtime +2
find ./ -ctime +2
find ./ -mtime +2
find ./ -ctime +2
find / -ctime  +20  最近修改文件時間20分鐘以前
find / -mtime  +7   修改文件為7天之前的(最重要)
find / -mtime  7    修改文件為第7天，就是往前推7天
find / -mtime  -7   修改文件為7天之內的
```

4.5.10 按照目錄或文件的權限來查找文件

```
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 目錄下查找不在 test4 子目錄之內的所有文件 ^^

```
find ./test -path "test/test4" -prune -o -print
【可以使用-prune選項來指出需要忽略的目錄。在使用-prune選項時要當心，因為如果你同時使用了-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
```

查找更改時間在比 log2012.log 文件新的文件 ^^

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

4.5.13 在當前目錄下查找文件長度大于 1 M 字節的文件

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

4.5.14 在/home/apache 目錄下查找文件長度恰好爲 100 字節的文件

```
find /home/apache -size 100c -print
```

4.5.15 在當前目錄下查找長度超過 10 塊的文件

```
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 選項中

```
find . -type f -exec ls -l {} \; 【{}   花括號代表前面find查找出來的文件名】
```

4.5.18 在目錄中查找更改時間在 n 日以前的文件并刪除它們

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

4.5.19 在目錄中查找更改時間在 n 日以前的文件并刪除它們，在刪除之前先給出提示

```
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" {} \; 【過濾文件內容用】
```

4.5.21 查找文件移動到指定目錄

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

4.5.22 用 exec 選項執行 cp 命令

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

4.6 linux-xargs-命令

xargs 是給命令傳遞參數的一個過濾器，也是組合多個命令的一個工具。

xargs 可以將管道或標準輸入（stdin）數據轉換成命令行參數，也能夠從文件的輸出中讀取數據。

xargs 也可以將單行或多行文本輸入轉換爲其他格式，例如多行變單行，單行變多行。

xargs 默認的命令是 echo，這意味着通過管道傳遞給 xargs 的輸入將會包含換行和空白，不過通過 xargs 的處理，換行和空白將被空格取代。

xargs 是一個强有力的命令，它能夠捕獲一個命令的輸出，然後傳遞給另外一個命令。

之所以能用到這個命令，關鍵是由于很多命令不支持 | 管道來傳遞參數，而日常工作中有有這個必要，所以就有了

xargs 命令，例如：

```
find /sbin -perm +700 |ls -l        #這個命令是錯誤的
find /sbin -perm +700 |xargs ls -l   #這樣才是正確的
```

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 參數：

- -a file 從文件中讀入作爲 sdtin
- -e flag，注意有的時候可能會是-E，flag 必須是一個以空格分隔的標志，當 xargs 分析到含有 flag 這個標志的時候就停止。
- -p 當每次執行一個 argument 的時候詢問一次用户。
- -n num 後面加次數，表示命令在執行的時候一次用的 argument 的個數，默認是用所有的。
- -t 表示先打印命令，然後再執行。
- -i 或者是-I，這得看 linux 支持了，將 xargs 的每項名稱，一般是一行一行賦值給 {}，可以用 {} 代替。
- -r no-run-if-empty 當 xargs 的輸入爲空的時候則停止 xargs，不用再去執行了。
- -s num 命令行的最大字符數，指的是 xargs 後面那個命令的最大命令行字符數。
- -L num 從標準輸入一次讀取 num 行送給 command 命令。
- -l 同 -L。
- -d delim 分隔符，默認的 xargs 分隔符是回車，argument 的分隔符是空格，這裏修改的是 xargs 的分隔符。
- -x exit 的意思，主要是配合-s 使用。。
- -P 修改最大的進程數，默認是 1，爲 0 時候爲 as many as it can，這個例子我没有想到，應該平時都用不到的吧。

4.6.3 實例

4.6.3.1 xargs 用作替换工具，讀取輸入數據重新格式化後輸出。 定義一個測試文件，内有多行文本數據：

```
# 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
```

多行輸入單行輸出：

```
# 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 選項多行輸出：

```
# 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 選項可以自定義一個定界符：

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

name name name name
```

結合 -n 選項使用：

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

name name
name name
```

讀取 stdin，將格式化後的參數傳遞給命令

假設一個命令爲 sk.sh 和一個保存參數的文件 arg.txt：

```
#!/bin/bash
#sk.sh命令內容，打印出所有參數。

echo $*
```

arg.txt 文件內容：

```
# cat arg.txt

aaa
bbb
ccc
```

4.6.3.2 xargs 的一個選項 -I {}

xargs 的一個選項 -I，使用 -I 指定一個替換字符串 {}，這個字符串在 xargs 擴展時會被替換掉，當 -I 與 xargs 結合使用，每一個參數命令都會被執行一次：

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

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

```
-p ccc -l
```

復制所有圖片文件到 /data/images 目錄下：

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

4.6.3.3 xargs 結合 find 使用 用 rm 刪除太多的文件時候，可能得到一個錯誤信息：**/bin/rm Argument list too long.** 用 xargs 去避免這個問題：

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

xargs -0 將 \0 作爲定界符。

統計一個源代碼目錄中所有 php 文件的行數：

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

查找所有的 jpg 文件，并且壓縮它們：

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

4.6.3.4 xargs 其他應用 假如你有一個文件包含了很多你希望下載的 URL，你能够使用 xargs 下載所有鏈接：

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

4.7 Linux 系統下 date 常用命令的參數以及獲取時間戳的方法

date: 用于顯示/設置系統的時間或者日期：date 選項 + 指定的格式：

+：進行格式化輸出
%Y:表示年份
%m:表示月份
%d:表示第幾天
%H:表示小時
%M:表示分鐘
%S:表示秒鐘
查看當前的系統時間：date
設置系統時間爲:date -s “20180316 16:53:10”
查看本地系統時間：date “+%Z”
查看星期幾：date “+%A”
輸入當前是上午還是下午：date “+%p”
判斷今天是一年中的第幾天：date “+%j”
ctrl+l:清屏操作，相當于clear
等價一：date + %Y-%m-%d=date + %F
等價二：date + %H :%M :%S=date + %T
等價三：date + “%F %T”=date + ‘%F %T’（注意：有空格需要用到雙引號或單引號）

時間戳：時間戳是指格林威治時間自1970年1月1日（00:00:00 GMT）至當前時間的總秒數。它也被稱爲U

時間->時間戳： date +%s
時間戳->時間： date +%Y:%m:%d -d @1425384141
Unix時間戳（英文爲Unix epoch, Unix time, POSIXme 或 Unix timestamp）是從1970年1月1日（UTC/GMT
misc

4.8 cp 命令詳解

Linux-cp 命令詳解

默認情況下，如果目標文件存在，它將被覆蓋。-n 選項告訴 cp 不要覆蓋現有文件。要提示確認，請使用該 -i 選項。

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

如果要僅在文件比目標更新時復制文件，請使用以下 -u 選項：

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

另一個可能有用的選項是 -v，他告訴 cp 打印詳細輸出：

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

使用 cp 命令復制目録要復制目録 (包括其所有文件和子目録)，請使用 -R 或 -r 選項。在以下示例中，我們將目録復制 Pictures 到 Pictures_backup :

```
cp -R 源目録 目標目録
```

要僅復制文件和子目録，而不復制目標目録，請使用以下 -t 選項 (原版有錯，不能用-T):

```
cp -Rt 目標目録 源目録
```

另一種祇復制目録內容而不是目録本身的方法是使用通配符 (*) 。以下命令的缺點是它不會復制隱藏文件和目録 (以點. 開頭的文件和目録) :

```
cp -Rt 目標目録 源目録/*
```

4.9 拷貝命令比較，XCOPY(win) VS cp(linux)

windows 下 XCOPY 命令，目標目録的父目録可以不存在，命令自己會創建

Linux 下 cp 不會自動創建目標目録的父目録，如果目標目録不在在會直接報錯。

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

windows 下 gnumake 命令 wildcard 返回匹配文件名帶目録（待確認）

Linux 下 gnumake 命令 wildcard 返回匹配文件名帶目録（已確認）

4.11 touch 命令直接創建空白文件

Linux Touch 命令的 8 種常見使用方法

touch test.txt

命令為：“touch [選項] [文件]”。

```
-a    祇更改訪問時間
-c，--no-create 不創建任何文件
-d，--date=字符串 使用指定字符串表示時間而非當前時間
-f    （忽略）
-h，--no-dereference  會影響符號鏈接本身，而非符號鏈接所指示的目的地
  （當系統支持更改符號鏈接的所有者時，此選項才有用）
-m    祇更改修改時間
```

```
-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  顯示此幫助信息并退出
    --version  顯示版本信息并退出
```

4.12 Linux 文件三種時間屬性 atime/mtime/ctime：

atime(access time): 最近訪問文件內容時間（Last Access Time）。

mtime(modify time): 最近修改文件內容時間（Last Modification Time）。

ctime(change time): 最近更改文件屬性（Inode 內容更改）的時間，包括文件名、大小、內容、權限、屬主、屬組等（Last Change Time）。

1. 輸入 “touch filetime.txt” 創建新文件，輸入 “stat filetime.txt” 即可查看文件 filetime.txt 的時間屬性。

   備注：新創建文件的三種時間抓取當前時間，本例中爲 2019-01-05 19:42:36。

   Birth 時間爲空，Linux 需要內核提供 xstat() 接口才可獲取 Birth 時間。

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

   備注：ls，stat 命令不會修改 atime。

3. 輸入 “echo ”add test”»filetime.txt” 給文件增加內容 “add test” 後，輸入 “stat filetime.txt” 查看時間屬性，發現 mtime 和 ctime 均已更新（2019-01-05 19:55:05）。

4. 輸入 “mv filetime.txt new.txt” 修改文件名爲 new.txt，輸入 “stat new.txt” 查看時間屬性，發現衹有 ctime 更新（2019-01-05 19:57:05）。

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

5. 輸入 “ls -lc new.txt” 可查看文件 new.txt 的 ctime。

6. 輸入 “ls -lu new.txt” 可查看文件 new.txt 的 atime。

7. 輸入 “ls -l new.txt” 可查看文件 new.txt 的 mtime。

4.13 利用 date 時間戳 <-> 時間

時間戳：時間戳是指格林威治時間自 1970 年 1 月 1 日（00:00:00 GMT）至當前時間的總秒數。它也被稱爲 Unix 時間戳（Unix Timestamp）。通俗的講，時間戳是一份能夠表示一份數據在一個特定時間點已經存在的完整的可驗證的數據。

時間-> 時間戳：date +%s

時間戳-> 時間：date +%Y:%m:%d -d @1425384141

Unix 時間戳（英文爲 Unix epoch, Unix time, POSIXme 或 Unix timestamp）是從 1970 年 1 月 1 日（UTC/GMT 的午夜）開始所經過的秒數，不考慮閏秒。

4.14 sed 命令功能強大替換

一、基本的替換:

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

這兩種命令格式的區別在于是否有個 “g”。沒有 “g” 表示祇替換第一個匹配到的字符串，有 “g” 表示替換所有能匹配到的字符串

二、替換某行內容:

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

第一個命令表示用新的字符串替換指定這一行的內容，第二個命令表示用新字符串替換指定幾行的內容。如下圖，第一個命令將第 2 行內容替換成了 “new test!”，第二個命令將第 2 到 6 行替換成了 “new test!”。

三、多條件替換

命令格式：sed -e 命令1 -e 命令2 -e 命令3

有些時候有多個替換條件，那就可以使用 “-e” 參數將這些替換條件連接起來，一次性完成所有的替換操作。例如，可以將上述的兩種命令連接起來：“sed -e ’s/原字符串/新字符串/’ ’ 行號 c 新字符串’ 文件”。如下圖，不僅將小寫 “a” 替換成了大寫 “A”，還將第 2 行內容替換成了 “new test!”。

四、保存替換結果到文件中

命令格式：sed -i 命令

上述這些命令都祇是將替換結果打印到屏幕上，如果想保存結果到文件中，就需要加上 “-i” 參數。