UNIX常用命令简介

# 建立自己的环境

## CSH

在使用C-SHELL时，有两个环境文件需要配置：.cshrc和.login（以“.”开头的文件通常的情况下是隐藏的，可以采用命令“ls -a”查看）。

.cshrc设置说明

umask 027 # 设置创建文件时的缺省权限

set history = 32 # 设置需要记忆的历史命令数目

setenv WORK\_DIR $HOME # setenv 用于csh设置环境变量

setenv TERM vt100 # 设置终端类型

setenv PATH /opt/softbench/bin:/usr/bin/X11:/usr/c++/bin:/usr/contrib/bin: \

                       /usr/local/bin:/usr/contrib/bin/X11:/usr/local/bin/X11: \

                       /opt/ansic/bin:/opt/nettladm/bin:/opt/graphics/common/bin: \

                       /opt/upgrade/bin:/opt/CC/bin:/opt/langtools/bin:/opt/imake/bin: \

                       ${PATH}:.:${INFORMIXDIR}/bin:${INFORMIXDIR}/lib/esql

                          # 设置执行搜索路径

.login设置说明

stty erase "^H" kill "^U" intr "^C" eof "^D" susp "^Z" \

       hupcl ixon ixoff tostop tabs eval `tset -s -Q -m ':?hp' `

# stty 用于设置终端的特殊键位。

backspace是删除，

delete是kill，

ctrl + C是中断，等。

# 很多人回退键没法使用，ctrl+C也没发用等，就是因为没有设置这些属性。

另，在HPUX下经常出现需要自己选择终端类型的提示“TERM = (hp)”，这是因为在.login文件中含有怎么一行“eval `tset -s -Q -m ':?hp' `”，你只需将它注释掉即可，然后直接使用命令“setenv TERM \*\*\*”设置终端类型即可。

## BSH/SH

$ man jobs

BASH\_BUILTINS(1) BASH\_BUILTINS(1)

NAME

bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, compopt, continue, declare, dirs, disown, echo, enable, eval, exec, exit,

export, false, fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt,

source, suspend, test, times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see bash(1)

BASH BUILTIN COMMANDS

Unless otherwise noted, each builtin command documented in this section as accepting options preceded by - accepts -- to signify the end of the options. The :,

true, false, and test builtins do not accept options and do not treat -- specially. The exit, logout, break, continue, let, and shift builtins accept and process

arguments beginning with - without requiring --. Other builtins that accept arguments but are not specified as accepting options interpret arguments beginning with

- as invalid options and require -- to prevent this interpretation.

: [arguments]

No effect; the command does nothing beyond expanding arguments and performing any specified redirections. A zero exit code is returned.

. filename [arguments]

source filename [arguments]

Read and execute commands from filename in the current shell environment and return the exit status of the last command executed from filename. If filename does not contain a slash, file names in PATH are used to find the directory containing filename. The file searched for in PATH need not be executable. When bash is not in posix mode, the current directory is searched if no file is found in PATH. If the sourcepath option to the shopt builtin command is turned off, the PATH is not searched. If any arguments are supplied, they become the positional parameters when filename is executed. Otherwise the positional parameters are unchanged. The return status is the status of the last command exited within the script (0 if no commands are executed), and false if file-name is not found or cannot be read.

alias [-p] [name[=value] ...]

Alias with no arguments or with the -p option prints the list of aliases in the form alias name=value on standard output. When arguments are supplied, an

alias is defined for each name whose value is given. A trailing space in value causes the next word to be checked for alias substitution when the alias is

expanded. For each name in the argument list for which no value is supplied, the name and value of the alias is printed. Alias returns true unless a name

is given for which no alias has been defined.

bg [jobspec ...]

Resume each suspended job jobspec in the background, as if it had been started with &. If jobspec is not present, the shell’s notion of the current job is

used. bg jobspec returns 0 unless run when job control is disabled or, when run with job control enabled, any specified jobspec was not found or was started

without job control.

bind [-m keymap] [-lpsvPSV]

bind [-m keymap] [-q function] [-u function] [-r keyseq]

bind [-m keymap] -f filename

bind [-m keymap] -x keyseq:shell-command

bind [-m keymap] keyseq:function-name

bind readline-command

Display current readline key and function bindings, bind a key sequence to a readline function or macro, or set a readline variable. Each non-option argu-

ment is a command as it would appear in .inputrc, but each binding or command must be passed as a separate argument; e.g., ’"\C-x\C-r": re-read-init-file’.

Options, if supplied, have the following meanings:

-m keymap

Use keymap as the keymap to be affected by the subsequent bindings. Acceptable keymap names are emacs, emacs-standard, emacs-meta, emacs-ctlx, vi,

vi-move, vi-command, and vi-insert. vi is equivalent to vi-command; emacs is equivalent to emacs-standard.

-l List the names of all readline functions.

-p Display readline function names and bindings in such a way that they can be re-read.

-P List current readline function names and bindings.

-s Display readline key sequences bound to macros and the strings they output in such a way that they can be re-read.

-S Display readline key sequences bound to macros and the strings they output.

-v Display readline variable names and values in such a way that they can be re-read.

-V List current readline variable names and values.

-f filename

Read key bindings from filename.

-q function

Query about which keys invoke the named function.

-u function

Unbind all keys bound to the named function.

-r keyseq

Remove any current binding for keyseq.

-x keyseq:shell-command

Cause shell-command to be executed whenever keyseq is entered. When shell-command is executed, the shell sets the READLINE\_LINE variable to the con-

tents of the readline line buffer and the READLINE\_POINT variable to the current location of the insertion point. If the executed command changes the

value of READLINE\_LINE or READLINE\_POINT, those new values will be reflected in the editing state.

The return value is 0 unless an unrecognized option is given or an error occurred.

break [n]

Exit from within a for, while, until, or select loop. If n is specified, break n levels. n must be ≥ 1. If n is greater than the number of enclosing

loops, all enclosing loops are exited. The return value is non-zero when n is ≤ 0; Otherwise, break returns 0 value.

builtin shell-builtin [arguments]

Execute the specified shell builtin, passing it arguments, and return its exit status. This is useful when defining a function whose name is the same as a

shell builtin, retaining the functionality of the builtin within the function. The cd builtin is commonly redefined this way. The return status is false if

shell-builtin is not a shell builtin command.

caller [expr]

Returns the context of any active subroutine call (a shell function or a script executed with the . or source builtins. Without expr, caller displays the

line number and source filename of the current subroutine call. If a non-negative integer is supplied as expr, caller displays the line number, subroutine

name, and source file corresponding to that position in the current execution call stack. This extra information may be used, for example, to print a stack

trace. The current frame is frame 0. The return value is 0 unless the shell is not executing a subroutine call or expr does not correspond to a valid posi-

tion in the call stack.

cd [-L|-P] [dir]

Change the current directory to dir. The variable HOME is the default dir. The variable CDPATH defines the search path for the directory containing dir.

Alternative directory names in CDPATH are separated by a colon (:). A null directory name in CDPATH is the same as the current directory, i.e., ‘‘.’’. If

dir begins with a slash (/), then CDPATH is not used. The -P option says to use the physical directory structure instead of following symbolic links (see

also the -P option to the set builtin command); the -L option forces symbolic links to be followed. An argument of - is equivalent to $OLDPWD. If a non-

empty directory name from CDPATH is used, or if - is the first argument, and the directory change is successful, the absolute pathname of the new working

directory is written to the standard output. The return value is true if the directory was successfully changed; false otherwise.

command [-pVv] command [arg ...]

Run command with args suppressing the normal shell function lookup. Only builtin commands or commands found in the PATH are executed. If the -p option is

given, the search for command is performed using a default value for PATH that is guaranteed to find all of the standard utilities. If either the -V or -v

option is supplied, a description of command is printed. The -v option causes a single word indicating the command or file name used to invoke command to be

displayed; the -V option produces a more verbose description. If the -V or -v option is supplied, the exit status is 0 if command was found, and 1 if not.

If neither option is supplied and an error occurred or command cannot be found, the exit status is 127. Otherwise, the exit status of the command builtin is

the exit status of command.

compgen [option] [word]

Generate possible completion matches for word according to the options, which may be any option accepted by the complete builtin with the exception of -p and

-r, and write the matches to the standard output. When using the -F or -C options, the various shell variables set by the programmable completion facili-

ties, while available, will not have useful values.

The matches will be generated in the same way as if the programmable completion code had generated them directly from a completion specification with the

same flags. If word is specified, only those completions matching word will be displayed.

The return value is true unless an invalid option is supplied, or no matches were generated.

complete [-abcdefgjksuv] [-o comp-option] [-DE] [-A action] [-G globpat] [-W wordlist] [-F function] [-C command]

[-X filterpat] [-P prefix] [-S suffix] name [name ...]

complete -pr [-DE] [name ...]

Specify how arguments to each name should be completed. If the -p option is supplied, or if no options are supplied, existing completion specifications are

printed in a way that allows them to be reused as input. The -r option removes a completion specification for each name, or, if no names are supplied, all

completion specifications. The -D option indicates that the remaining options and actions should apply to the ‘‘default’’ command completion; that is, com-

pletion attempted on a command for which no completion has previously been defined. The -E option indicates that the remaining options and actions should

apply to ‘‘empty’’ command completion; that is, completion attempted on a blank line.

The process of applying these completion specifications when word completion is attempted is described above under Programmable Completion.

Other options, if specified, have the following meanings. The arguments to the -G, -W, and -X options (and, if necessary, the -P and -S options) should be

quoted to protect them from expansion before the complete builtin is invoked.

-o comp-option

The comp-option controls several aspects of the compspec’s behavior beyond the simple generation of completions. comp-option may be one of:

bashdefault

Perform the rest of the default bash completions if the compspec generates no matches.

default Use readline’s default filename completion if the compspec generates no matches.

dirnames

Perform directory name completion if the compspec generates no matches.

filenames

Tell readline that the compspec generates filenames, so it can perform any filename-specific processing (like adding a slash to directory

names, quoting special characters, or suppressing trailing spaces). Intended to be used with shell functions.

nospace Tell readline not to append a space (the default) to words completed at the end of the line.

plusdirs

After any matches defined by the compspec are generated, directory name completion is attempted and any matches are added to the results of

the other actions.

-A action

The action may be one of the following to generate a list of possible completions:

alias Alias names. May also be specified as -a.

arrayvar

Array variable names.

binding Readline key binding names.

builtin Names of shell builtin commands. May also be specified as -b.

command Command names. May also be specified as -c.

directory

Directory names. May also be specified as -d.

disabled

Names of disabled shell builtins.

enabled Names of enabled shell builtins.

export Names of exported shell variables. May also be specified as -e.

file File names. May also be specified as -f.

function

Names of shell functions.

group Group names. May also be specified as -g.

helptopic

Help topics as accepted by the help builtin.

hostname

Hostnames, as taken from the file specified by the HOSTFILE shell variable.

job Job names, if job control is active. May also be specified as -j.

keyword Shell reserved words. May also be specified as -k.

running Names of running jobs, if job control is active.

service Service names. May also be specified as -s.

setopt Valid arguments for the -o option to the set builtin.

shopt Shell option names as accepted by the shopt builtin.

signal Signal names.

stopped Names of stopped jobs, if job control is active.

user User names. May also be specified as -u.

variable

Names of all shell variables. May also be specified as -v.

-G globpat

The pathname expansion pattern globpat is expanded to generate the possible completions.

-W wordlist

The wordlist is split using the characters in the IFS special variable as delimiters, and each resultant word is expanded. The possible completions

are the members of the resultant list which match the word being completed.

-C command

command is executed in a subshell environment, and its output is used as the possible completions.

-F function

The shell function function is executed in the current shell environment. When it finishes, the possible completions are retrieved from the value of

the COMPREPLY array variable.

-X filterpat

filterpat is a pattern as used for pathname expansion. It is applied to the list of possible completions generated by the preceding options and

arguments, and each completion matching filterpat is removed from the list. A leading ! in filterpat negates the pattern; in this case, any comple-

tion not matching filterpat is removed.

-P prefix

prefix is added at the beginning of each possible completion after all other options have been applied.

-S suffix

suffix is appended to each possible completion after all other options have been applied.

The return value is true unless an invalid option is supplied, an option other than -p or -r is supplied without a name argument, an attempt is made to

remove a completion specification for a name for which no specification exists, or an error occurs adding a completion specification.

compopt [-o option] [-DE] [+o option] [name]

Modify completion options for each name according to the options, or for the currently-execution completion if no names are supplied. If no options are

given, display the completion options for each name or the current completion. The possible values of option are those valid for the complete builtin

described above. The -D option indicates that the remaining options should apply to the ‘‘default’’ command completion; that is, completion attempted on a

command for which no completion has previously been defined. The -E option indicates that the remaining options should apply to ‘‘empty’’ command comple-

tion; that is, completion attempted on a blank line.

The return value is true unless an invalid option is supplied, an attempt is made to modify the options for a name for which no completion specification exists, or

an output error occurs.

continue [n]

Resume the next iteration of the enclosing for, while, until, or select loop. If n is specified, resume at the nth enclosing loop. n must be ≥ 1. If n is

greater than the number of enclosing loops, the last enclosing loop (the ‘‘top-level’’ loop) is resumed. When continue is executed inside of loop, the

return value is non-zero when n is ≤ 0; Otherwise, continue returns 0 value. When continue is executed outside of loop, the return value is 0.

declare [-aAfFilrtux] [-p] [name[=value] ...]

typeset [-aAfFilrtux] [-p] [name[=value] ...]

Declare variables and/or give them attributes. If no names are given then display the values of variables. The -p option will display the attributes and

values of each name. When -p is used with name arguments, additional options are ignored. When -p is supplied without name arguments, it will display the

attributes and values of all variables having the attributes specified by the additional options. If no other options are supplied with -p, declare will

display the attributes and values of all shell variables. The -f option will restrict the display to shell functions. The -F option inhibits the display of

function definitions; only the function name and attributes are printed. If the extdebug shell option is enabled using shopt, the source file name and line

number where the function is defined are displayed as well. The -F option implies -f. The following options can be used to restrict output to variables

with the specified attribute or to give variables attributes:

-a Each name is an indexed array variable (see Arrays above).

-A Each name is an associative array variable (see Arrays above).

-f Use function names only.

-i The variable is treated as an integer; arithmetic evaluation (see ARITHMETIC EVALUATION above) is performed when the variable is assigned a value.

-l When the variable is assigned a value, all upper-case characters are converted to lower-case. The upper-case attribute is disabled.

-r Make names readonly. These names cannot then be assigned values by subsequent assignment statements or unset.

-t Give each name the trace attribute. Traced functions inherit the DEBUG and RETURN traps from the calling shell. The trace attribute has no special

meaning for variables.

-u When the variable is assigned a value, all lower-case characters are converted to upper-case. The lower-case attribute is disabled.

-x Mark names for export to subsequent commands via the environment.

Using ‘+’ instead of ‘-’ turns off the attribute instead, with the exceptions that +a may not be used to destroy an array variable and +r will not remove the

readonly attribute. When used in a function, makes each name local, as with the local command. If a variable name is followed by =value, the value of the

variable is set to value. The return value is 0 unless an invalid option is encountered, an attempt is made to define a function using ‘‘-f foo=bar’’, an

attempt is made to assign a value to a readonly variable, an attempt is made to assign a value to an array variable without using the compound assignment

syntax (see Arrays above), one of the names is not a valid shell variable name, an attempt is made to turn off readonly status for a readonly variable, an

attempt is made to turn off array status for an array variable, or an attempt is made to display a non-existent function with -f.

dirs [+n] [-n] [-cplv]

Without options, displays the list of currently remembered directories. The default display is on a single line with directory names separated by spaces.

Directories are added to the list with the pushd command; the popd command removes entries from the list.

+n Displays the nth entry counting from the left of the list shown by dirs when invoked without options, starting with zero.

-n Displays the nth entry counting from the right of the list shown by dirs when invoked without options, starting with zero.

-c Clears the directory stack by deleting all of the entries.

-l Produces a longer listing; the default listing format uses a tilde to denote the home directory.

-p Print the directory stack with one entry per line.

-v Print the directory stack with one entry per line, prefixing each entry with its index in the stack.

The return value is 0 unless an invalid option is supplied or n indexes beyond the end of the directory stack.

disown [-ar] [-h] [jobspec ...]

Without options, each jobspec is removed from the table of active jobs. If jobspec is not present, and neither -a nor -r is supplied, the shell’s notion of

the current job is used. If the -h option is given, each jobspec is not removed from the table, but is marked so that SIGHUP is not sent to the job if the

shell receives a SIGHUP. If no jobspec is present, and neither the -a nor the -r option is supplied, the current job is used. If no jobspec is supplied,

the -a option means to remove or mark all jobs; the -r option without a jobspec argument restricts operation to running jobs. The return value is 0 unless a

jobspec does not specify a valid job.

echo [-neE] [arg ...]

Output the args, separated by spaces, followed by a newline. The return status is always 0. If -n is specified, the trailing newline is suppressed. If the

-e option is given, interpretation of the following backslash-escaped characters is enabled. The -E option disables the interpretation of these escape char-

acters, even on systems where they are interpreted by default. The xpg\_echo shell option may be used to dynamically determine whether or not echo expands

these escape characters by default. echo does not interpret -- to mean the end of options. echo interprets the following escape sequences:

\a alert (bell)

\b backspace

\c suppress further output

\e an escape character

\f form feed

\n new line

\r carriage return

\t horizontal tab

\v vertical tab

\\ backslash

\0nnn the eight-bit character whose value is the octal value nnn (zero to three octal digits)

\xHH the eight-bit character whose value is the hexadecimal value HH (one or two hex digits)

enable [-a] [-dnps] [-f filename] [name ...]

Enable and disable builtin shell commands. Disabling a builtin allows a disk command which has the same name as a shell builtin to be executed without spec-

ifying a full pathname, even though the shell normally searches for builtins before disk commands. If -n is used, each name is disabled; otherwise, names

are enabled. For example, to use the test binary found via the PATH instead of the shell builtin version, run ‘‘enable -n test’’. The -f option means to

load the new builtin command name from shared object filename, on systems that support dynamic loading. The -d option will delete a builtin previously

loaded with -f. If no name arguments are given, or if the -p option is supplied, a list of shell builtins is printed. With no other option arguments, the

list consists of all enabled shell builtins. If -n is supplied, only disabled builtins are printed. If -a is supplied, the list printed includes all

builtins, with an indication of whether or not each is enabled. If -s is supplied, the output is restricted to the POSIX special builtins. The return value

is 0 unless a name is not a shell builtin or there is an error loading a new builtin from a shared object.

eval [arg ...]

The args are read and concatenated together into a single command. This command is then read and executed by the shell, and its exit status is returned as

the value of eval. If there are no args, or only null arguments, eval returns 0.

exec [-cl] [-a name] [command [arguments]]

If command is specified, it replaces the shell. No new process is created. The arguments become the arguments to command. If the -l option is supplied,

the shell places a dash at the beginning of the zeroth argument passed to command. This is what login(1) does. The -c option causes command to be executed

with an empty environment. If -a is supplied, the shell passes name as the zeroth argument to the executed command. If command cannot be executed for some

reason, a non-interactive shell exits, unless the shell option execfail is enabled, in which case it returns failure. An interactive shell returns failure

if the file cannot be executed. If command is not specified, any redirections take effect in the current shell, and the return status is 0. If there is a

redirection error, the return status is 1.

exit [n]

Cause the shell to exit with a status of n. If n is omitted, the exit status is that of the last command executed. A trap on EXIT is executed before the

shell terminates.

export [-fn] [name[=word]] ...

export -p

The supplied names are marked for automatic export to the environment of subsequently executed commands. If the -f option is given, the names refer to func-

tions. If no names are given, or if the -p option is supplied, a list of all names that are exported in this shell is printed. The -n option causes the

export property to be removed from each name. If a variable name is followed by =word, the value of the variable is set to word. export returns an exit

status of 0 unless an invalid option is encountered, one of the names is not a valid shell variable name, or -f is supplied with a name that is not a func-

tion.

fc [-e ename] [-lnr] [first] [last]

fc -s [pat=rep] [cmd]

Fix Command. In the first form, a range of commands from first to last is selected from the history list. First and last may be specified as a string (to

locate the last command beginning with that string) or as a number (an index into the history list, where a negative number is used as an offset from the

current command number). If last is not specified it is set to the current command for listing (so that ‘‘fc -l -10’’ prints the last 10 commands) and to

first otherwise. If first is not specified it is set to the previous command for editing and -16 for listing.

The -n option suppresses the command numbers when listing. The -r option reverses the order of the commands. If the -l option is given, the commands are

listed on standard output. Otherwise, the editor given by ename is invoked on a file containing those commands. If ename is not given, the value of the

FCEDIT variable is used, and the value of EDITOR if FCEDIT is not set. If neither variable is set, is used. When editing is complete, the edited commands

are echoed and executed.

In the second form, command is re-executed after each instance of pat is replaced by rep. A useful alias to use with this is ‘‘r="fc -s"’’, so that typing

‘‘r cc’’ runs the last command beginning with ‘‘cc’’ and typing ‘‘r’’ re-executes the last command.

If the first form is used, the return value is 0 unless an invalid option is encountered or first or last specify history lines out of range. If the -e

option is supplied, the return value is the value of the last command executed or failure if an error occurs with the temporary file of commands. If the

second form is used, the return status is that of the command re-executed, unless cmd does not specify a valid history line, in which case fc returns fail-

ure.

fg [jobspec]

Resume jobspec in the foreground, and make it the current job. If jobspec is not present, the shell’s notion of the current job is used. The return value

is that of the command placed into the foreground, or failure if run when job control is disabled or, when run with job control enabled, if jobspec does not

specify a valid job or jobspec specifies a job that was started without job control.

getopts optstring name [args]

getopts is used by shell procedures to parse positional parameters. optstring contains the option characters to be recognized; if a character is followed by

a colon, the option is expected to have an argument, which should be separated from it by white space. The colon and question mark characters may not be

used as option characters. Each time it is invoked, getopts places the next option in the shell variable name, initializing name if it does not exist, and

the index of the next argument to be processed into the variable OPTIND. OPTIND is initialized to 1 each time the shell or a shell script is invoked. When

an option requires an argument, getopts places that argument into the variable OPTARG. The shell does not reset OPTIND automatically; it must be manually

reset between multiple calls to getopts within the same shell invocation if a new set of parameters is to be used.

When the end of options is encountered, getopts exits with a return value greater than zero. OPTIND is set to the index of the first non-option argument,

and name is set to ?.

getopts normally parses the positional parameters, but if more arguments are given in args, getopts parses those instead.

getopts can report errors in two ways. If the first character of optstring is a colon, silent error reporting is used. In normal operation diagnostic mes-

sages are printed when invalid options or missing option arguments are encountered. If the variable OPTERR is set to 0, no error messages will be displayed,

even if the first character of optstring is not a colon.

If an invalid option is seen, getopts places ? into name and, if not silent, prints an error message and unsets OPTARG. If getopts is silent, the option

character found is placed in OPTARG and no diagnostic message is printed.

If a required argument is not found, and getopts is not silent, a question mark (?) is placed in name, OPTARG is unset, and a diagnostic message is printed.

If getopts is silent, then a colon (:) is placed in name and OPTARG is set to the option character found.

getopts returns true if an option, specified or unspecified, is found. It returns false if the end of options is encountered or an error occurs.

hash [-lr] [-p filename] [-dt] [name]

For each name, the full file name of the command is determined by searching the directories in $PATH and remembered. If the -p option is supplied, no path

search is performed, and filename is used as the full file name of the command. The -r option causes the shell to forget all remembered locations. The -d

option causes the shell to forget the remembered location of each name. If the -t option is supplied, the full pathname to which each name corresponds is

printed. If multiple name arguments are supplied with -t, the name is printed before the hashed full pathname. The -l option causes output to be displayed

in a format that may be reused as input. If no arguments are given, or if only -l is supplied, information about remembered commands is printed. The return

status is true unless a name is not found or an invalid option is supplied.

help [-dms] [pattern]

Display helpful information about builtin commands. If pattern is specified, help gives detailed help on all commands matching pattern; otherwise help for

all the builtins and shell control structures is printed.

-d Display a short description of each pattern

-m Display the description of each pattern in a manpage-like format

-s Display only a short usage synopsis for each pattern

The return status is 0 unless no command matches pattern.

history [n]

history -c

history -d offset

history -anrw [filename]

history -p arg [arg ...]

history -s arg [arg ...]

With no options, display the command history list with line numbers. Lines listed with a \* have been modified. An argument of n lists only the last n

lines. If the shell variable HISTTIMEFORMAT is set and not null, it is used as a format string for strftime(3) to display the time stamp associated with

each displayed history entry. No intervening blank is printed between the formatted time stamp and the history line. If filename is supplied, it is used as

the name of the history file; if not, the value of HISTFILE is used. Options, if supplied, have the following meanings:

-c Clear the history list by deleting all the entries.

-d offset

Delete the history entry at position offset.

-a Append the ‘‘new’’ history lines (history lines entered since the beginning of the current bash session) to the history file.

-n Read the history lines not already read from the history file into the current history list. These are lines appended to the history file since the

beginning of the current bash session.

-r Read the contents of the history file and use them as the current history.

-w Write the current history to the history file, overwriting the history file’s contents.

-p Perform history substitution on the following args and display the result on the standard output. Does not store the results in the history list.

Each arg must be quoted to disable normal history expansion.

-s Store the args in the history list as a single entry. The last command in the history list is removed before the args are added.

If the HISTTIMEFORMAT variable is set, the time stamp information associated with each history entry is written to the history file, marked with the history

comment character. When the history file is read, lines beginning with the history comment character followed immediately by a digit are interpreted as

timestamps for the previous history line. The return value is 0 unless an invalid option is encountered, an error occurs while reading or writing the his-

tory file, an invalid offset is supplied as an argument to -d, or the history expansion supplied as an argument to -p fails.

jobs [-lnprs] [ jobspec ... ]

jobs -x command [ args ... ]

The first form lists the active jobs. The options have the following meanings:

-l List process IDs in addition to the normal information.

-p List only the process ID of the job’s process group leader.

-n Display information only about jobs that have changed status since the user was last notified of their status.

-r Restrict output to running jobs.

-s Restrict output to stopped jobs.

If jobspec is given, output is restricted to information about that job. The return status is 0 unless an invalid option is encountered or an invalid job-

spec is supplied.

If the -x option is supplied, jobs replaces any jobspec found in command or args with the corresponding process group ID, and executes command passing it

args, returning its exit status.

kill [-s sigspec | -n signum | -sigspec] [pid | jobspec] ...

kill -l [sigspec | exit\_status]

Send the signal named by sigspec or signum to the processes named by pid or jobspec. sigspec is either a case-insensitive signal name such as SIGKILL (with

or without the SIG prefix) or a signal number; signum is a signal number. If sigspec is not present, then SIGTERM is assumed. An argument of -l lists the

signal names. If any arguments are supplied when -l is given, the names of the signals corresponding to the arguments are listed, and the return status is

0. The exit\_status argument to -l is a number specifying either a signal number or the exit status of a process terminated by a signal. kill returns true

if at least one signal was successfully sent, or false if an error occurs or an invalid option is encountered.

let arg [arg ...]

Each arg is an arithmetic expression to be evaluated (see ARITHMETIC EVALUATION above). If the last arg evaluates to 0, let returns 1; 0 is returned other-

wise.

local [option] [name[=value] ...]

For each argument, a local variable named name is created, and assigned value. The option can be any of the options accepted by declare. When local is used

within a function, it causes the variable name to have a visible scope restricted to that function and its children. With no operands, local writes a list

of local variables to the standard output. It is an error to use local when not within a function. The return status is 0 unless local is used outside a

function, an invalid name is supplied, or name is a readonly variable.

logout Exit a login shell.

mapfile [-n count] [-O origin] [-s count] [-t] [-u fd] [-C callback] [-c quantum] [array]

readarray [-n count] [-O origin] [-s count] [-t] [-u fd] [-C callback] [-c quantum] [array]

Read lines from the standard input into the indexed array variable array, or from file descriptor fd if the -u option is supplied. The variable MAPFILE is

the default array. Options, if supplied, have the following meanings:

-n Copy at most count lines. If count is 0, all lines are copied.

-O Begin assigning to array at index origin. The default index is 0.

-s Discard the first count lines read.

-t Remove a trailing newline from each line read.

-u Read lines from file descriptor fd instead of the standard input.

-C Evaluate callback each time quantum lines are read. The -c option specifies quantum.

-c Specify the number of lines read between each call to callback.

If -C is specified without -c, the default quantum is 5000. When callback is evaluated, it is supplied the index of the next array element to be assigned as

an additional argument. callback is evaluated after the line is read but before the array element is assigned.

If not supplied with an explicit origin, mapfile will clear array before assigning to it.

mapfile returns successfully unless an invalid option or option argument is supplied, array is invalid or unassignable, or if array is not an indexed array.

popd [-n] [+n] [-n]

Removes entries from the directory stack. With no arguments, removes the top directory from the stack, and performs a cd to the new top directory. Argu-

ments, if supplied, have the following meanings:

-n Suppresses the normal change of directory when removing directories from the stack, so that only the stack is manipulated.

+n Removes the nth entry counting from the left of the list shown by dirs, starting with zero. For example: ‘‘popd +0’’ removes the first directory,

‘‘popd +1’’ the second.

-n Removes the nth entry counting from the right of the list shown by dirs, starting with zero. For example: ‘‘popd -0’’ removes the last directory,

‘‘popd -1’’ the next to last.

If the popd command is successful, a dirs is performed as well, and the return status is 0. popd returns false if an invalid option is encountered, the

directory stack is empty, a non-existent directory stack entry is specified, or the directory change fails.

printf [-v var] format [arguments]

Write the formatted arguments to the standard output under the control of the format. The format is a character string which contains three types of

objects: plain characters, which are simply copied to standard output, character escape sequences, which are converted and copied to the standard output, and

format specifications, each of which causes printing of the next successive argument. In addition to the standard printf(1) formats, %b causes printf to

expand backslash escape sequences in the corresponding argument (except that \c terminates output, backslashes in \', \", and \? are not removed, and octal

escapes beginning with \0 may contain up to four digits), and %q causes printf to output the corresponding argument in a format that can be reused as shell

input.

The -v option causes the output to be assigned to the variable var rather than being printed to the standard output.

The format is reused as necessary to consume all of the arguments. If the format requires more arguments than are supplied, the extra format specifications

behave as if a zero value or null string, as appropriate, had been supplied. The return value is zero on success, non-zero on failure.

pushd [-n] [+n] [-n]

pushd [-n] [dir]

Adds a directory to the top of the directory stack, or rotates the stack, making the new top of the stack the current working directory. With no arguments,

exchanges the top two directories and returns 0, unless the directory stack is empty. Arguments, if supplied, have the following meanings:

-n Suppresses the normal change of directory when adding directories to the stack, so that only the stack is manipulated.

+n Rotates the stack so that the nth directory (counting from the left of the list shown by dirs, starting with zero) is at the top.

-n Rotates the stack so that the nth directory (counting from the right of the list shown by dirs, starting with zero) is at the top.

dir Adds dir to the directory stack at the top, making it the new current working directory.

If the pushd command is successful, a dirs is performed as well. If the first form is used, pushd returns 0 unless the cd to dir fails. With the second

form, pushd returns 0 unless the directory stack is empty, a non-existent directory stack element is specified, or the directory change to the specified new

current directory fails.

pwd [-LP]

Print the absolute pathname of the current working directory. The pathname printed contains no symbolic links if the -P option is supplied or the -o physi-

cal option to the set builtin command is enabled. If the -L option is used, the pathname printed may contain symbolic links. The return status is 0 unless

an error occurs while reading the name of the current directory or an invalid option is supplied.

read [-ers] [-a aname] [-d delim] [-i text] [-n nchars] [-N nchars] [-p prompt] [-t timeout] [-u fd] [name ...]

One line is read from the standard input, or from the file descriptor fd supplied as an argument to the -u option, and the first word is assigned to the

first name, the second word to the second name, and so on, with leftover words and their intervening separators assigned to the last name. If there are

fewer words read from the input stream than names, the remaining names are assigned empty values. The characters in IFS are used to split the line into

words. The backslash character (\) may be used to remove any special meaning for the next character read and for line continuation. Options, if supplied,

have the following meanings:

-a aname

The words are assigned to sequential indices of the array variable aname, starting at 0. aname is unset before any new values are assigned. Other

name arguments are ignored.

-d delim

The first character of delim is used to terminate the input line, rather than newline.

-e If the standard input is coming from a terminal, readline (see READLINE above) is used to obtain the line. Readline uses the current (or default, if

line editing was not previously active) editing settings.

-i text

If readline is being used to read the line, text is placed into the editing buffer before editing begins.

-n nchars

read returns after reading nchars characters rather than waiting for a complete line of input, but honor a delimiter if fewer than nchars characters

are read before the delimiter.

-N nchars

read returns after reading exactly nchars characters rather than waiting for a complete line of input, unless EOF is encountered or read times out.

Delimiter characters encountered in the input are not treated specially and do not cause read to return until nchars characters are read.

-p prompt

Display prompt on standard error, without a trailing newline, before attempting to read any input. The prompt is displayed only if input is coming

from a terminal.

-r Backslash does not act as an escape character. The backslash is considered to be part of the line. In particular, a backslash-newline pair may not

be used as a line continuation.

-s Silent mode. If input is coming from a terminal, characters are not echoed.

-t timeout

Cause read to time out and return failure if a complete line of input is not read within timeout seconds. timeout may be a decimal number with a

fractional portion following the decimal point. This option is only effective if read is reading input from a terminal, pipe, or other special file;

it has no effect when reading from regular files. If timeout is 0, read returns success if input is available on the specified file descriptor, fail-

ure otherwise. The exit status is greater than 128 if the timeout is exceeded.

-u fd Read input from file descriptor fd.

If no names are supplied, the line read is assigned to the variable REPLY. The return code is zero, unless end-of-file is encountered, read times out (in

which case the return code is greater than 128), or an invalid file descriptor is supplied as the argument to -u.

readonly [-aApf] [name[=word] ...]

The given names are marked readonly; the values of these names may not be changed by subsequent assignment. If the -f option is supplied, the functions cor-

responding to the names are so marked. The -a option restricts the variables to indexed arrays; the -A option restricts the variables to associative arrays.

If no name arguments are given, or if the -p option is supplied, a list of all readonly names is printed. The -p option causes output to be displayed in a

format that may be reused as input. If a variable name is followed by =word, the value of the variable is set to word. The return status is 0 unless an

invalid option is encountered, one of the names is not a valid shell variable name, or -f is supplied with a name that is not a function.

return [n]

Causes a function to exit with the return value specified by n. If n is omitted, the return status is that of the last command executed in the function

body. If used outside a function, but during execution of a script by the . (source) command, it causes the shell to stop executing that script and return

either n or the exit status of the last command executed within the script as the exit status of the script. If used outside a function and not during exe-

cution of a script by ., the return status is false. Any command associated with the RETURN trap is executed before execution resumes after the function or

script.

set [--abefhkmnptuvxBCEHPT] [-o option] [arg ...]

set [+abefhkmnptuvxBCEHPT] [+o option] [arg ...]

Without options, the name and value of each shell variable are displayed in a format that can be reused as input for setting or resetting the currently-set

variables. Read-only variables cannot be reset. In posix mode, only shell variables are listed. The output is sorted according to the current locale.

When options are specified, they set or unset shell attributes. Any arguments remaining after option processing are treated as values for the positional

parameters and are assigned, in order, to $1, $2, ... $n. Options, if specified, have the following meanings:

-a Automatically mark variables and functions which are modified or created for export to the environment of subsequent commands.

-b Report the status of terminated background jobs immediately, rather than before the next primary prompt. This is effective only when job control is

enabled.

-e Exit immediately if a pipeline (which may consist of a single simple command), a subshell command enclosed in parentheses, or one of the commands

executed as part of a command list enclosed by braces (see SHELL GRAMMAR above) exits with a non-zero status. The shell does not exit if the command

that fails is part of the command list immediately following a while or until keyword, part of the test following the if or elif reserved words, part

of any command executed in a && or ││ list except the command following the final && or ││, any command in a pipeline but the last, or if the com-

mand’s return value is being inverted with !. A trap on ERR, if set, is executed before the shell exits. This option applies to the shell environ-

ment and each subshell environment separately (see COMMAND EXECUTION ENVIRONMENT above), and may cause subshells to exit before executing all the

commands in the subshell.

-f Disable pathname expansion.

-h Remember the location of commands as they are looked up for execution. This is enabled by default.

-k All arguments in the form of assignment statements are placed in the environment for a command, not just those that precede the command name.

-m Monitor mode. Job control is enabled. This option is on by default for interactive shells on systems that support it (see JOB CONTROL above).

Background processes run in a separate process group and a line containing their exit status is printed upon their completion.

-n Read commands but do not execute them. This may be used to check a shell script for syntax errors. This is ignored by interactive shells.

-o option-name

The option-name can be one of the following:

allexport

Same as -a.

braceexpand

Same as -B.

emacs Use an emacs-style command line editing interface. This is enabled by default when the shell is interactive, unless the shell is started

with the --noediting option. This also affects the editing interface used for read -e.

errexit Same as -e.

errtrace

Same as -E.

functrace

Same as -T.

hashall Same as -h.

histexpand

Same as -H.

history Enable command history, as described above under HISTORY. This option is on by default in interactive shells.

ignoreeof

The effect is as if the shell command ‘‘IGNOREEOF=10’’ had been executed (see Shell Variables above).

keyword Same as -k.

monitor Same as -m.

noclobber

Same as -C.

noexec Same as -n.

noglob Same as -f.

nolog Currently ignored.

notify Same as -b.

nounset Same as -u.

onecmd Same as -t.

physical

Same as -P.

pipefail

If set, the return value of a pipeline is the value of the last (rightmost) command to exit with a non-zero status, or zero if all commands

in the pipeline exit successfully. This option is disabled by default.

posix Change the behavior of bash where the default operation differs from the POSIX standard to match the standard (posix mode).

privileged

Same as -p.

verbose Same as -v.

vi Use a vi-style command line editing interface. This also affects the editing interface used for read -e.

xtrace Same as -x.

If -o is supplied with no option-name, the values of the current options are printed. If +o is supplied with no option-name, a series of set com-

mands to recreate the current option settings is displayed on the standard output.

-p Turn on privileged mode. In this mode, the $ENV and $BASH\_ENV files are not processed, shell functions are not inherited from the environment, and

the SHELLOPTS, BASHOPTS, CDPATH, and GLOBIGNORE variables, if they appear in the environment, are ignored. If the shell is started with the effec-

tive user (group) id not equal to the real user (group) id, and the -p option is not supplied, these actions are taken and the effective user id is

set to the real user id. If the -p option is supplied at startup, the effective user id is not reset. Turning this option off causes the effective

user and group ids to be set to the real user and group ids.

-t Exit after reading and executing one command.

-u Treat unset variables and parameters other than the special parameters "@" and "\*" as an error when performing parameter expansion. If expansion is

attempted on an unset variable or parameter, the shell prints an error message, and, if not interactive, exits with a non-zero status.

-v Print shell input lines as they are read.

-x After expanding each simple command, for command, case command, select command, or arithmetic for command, display the expanded value of PS4, fol-

lowed by the command and its expanded arguments or associated word list.

-B The shell performs brace expansion (see Brace Expansion above). This is on by default.

-C If set, bash does not overwrite an existing file with the >, >&, and <> redirection operators. This may be overridden when creating output files by

using the redirection operator >| instead of >.

-E If set, any trap on ERR is inherited by shell functions, command substitutions, and commands executed in a subshell environment. The ERR trap is

normally not inherited in such cases.

-H Enable ! style history substitution. This option is on by default when the shell is interactive.

-P If set, the shell does not follow symbolic links when executing commands such as cd that change the current working directory. It uses the physical

directory structure instead. By default, bash follows the logical chain of directories when performing commands which change the current directory.

-T If set, any traps on DEBUG and RETURN are inherited by shell functions, command substitutions, and commands executed in a subshell environment. The

DEBUG and RETURN traps are normally not inherited in such cases.

-- If no arguments follow this option, then the positional parameters are unset. Otherwise, the positional parameters are set to the args, even if some

of them begin with a -.

- Signal the end of options, cause all remaining args to be assigned to the positional parameters. The -x and -v options are turned off. If there are

no args, the positional parameters remain unchanged.

The options are off by default unless otherwise noted. Using + rather than - causes these options to be turned off. The options can also be specified as

arguments to an invocation of the shell. The current set of options may be found in $-. The return status is always true unless an invalid option is

encountered.

shift [n]

The positional parameters from n+1 ... are renamed to $1 .... Parameters represented by the numbers $# down to $#-n+1 are unset. n must be a non-negative

number less than or equal to $#. If n is 0, no parameters are changed. If n is not given, it is assumed to be 1. If n is greater than $#, the positional

parameters are not changed. The return status is greater than zero if n is greater than $# or less than zero; otherwise 0.

shopt [-pqsu] [-o] [optname ...]

Toggle the values of variables controlling optional shell behavior. With no options, or with the -p option, a list of all settable options is displayed,

with an indication of whether or not each is set. The -p option causes output to be displayed in a form that may be reused as input. Other options have the

following meanings:

-s Enable (set) each optname.

-u Disable (unset) each optname.

-q Suppresses normal output (quiet mode); the return status indicates whether the optname is set or unset. If multiple optname arguments are given with

-q, the return status is zero if all optnames are enabled; non-zero otherwise.

-o Restricts the values of optname to be those defined for the -o option to the set builtin.

If either -s or -u is used with no optname arguments, the display is limited to those options which are set or unset, respectively. Unless otherwise noted,

the shopt options are disabled (unset) by default.

The return status when listing options is zero if all optnames are enabled, non-zero otherwise. When setting or unsetting options, the return status is zero

unless an optname is not a valid shell option.

The list of shopt options is:

autocd If set, a command name that is the name of a directory is executed as if it were the argument to the cd command. This option is only used by inter-

active shells.

cdable\_vars

If set, an argument to the cd builtin command that is not a directory is assumed to be the name of a variable whose value is the directory to change

to.

cdspell If set, minor errors in the spelling of a directory component in a cd command will be corrected. The errors checked for are transposed characters, a

missing character, and one character too many. If a correction is found, the corrected file name is printed, and the command proceeds. This option

is only used by interactive shells.

checkhash

If set, bash checks that a command found in the hash table exists before trying to execute it. If a hashed command no longer exists, a normal path

search is performed.

checkjobs

If set, bash lists the status of any stopped and running jobs before exiting an interactive shell. If any jobs are running, this causes the exit to

be deferred until a second exit is attempted without an intervening command (see JOB CONTROL above). The shell always postpones exiting if any jobs

are stopped.

checkwinsize

If set, bash checks the window size after each command and, if necessary, updates the values of LINES and COLUMNS.

cmdhist If set, bash attempts to save all lines of a multiple-line command in the same history entry. This allows easy re-editing of multi-line commands.

compat31

If set, bash changes its behavior to that of version 3.1 with respect to quoted arguments to the conditional command’s =~ operator.

compat32

If set, bash changes its behavior to that of version 3.2 with respect to locale-specific string comparison when using the conditional command’s < and

> operators.

compat40

If set, bash changes its behavior to that of version 4.0 with respect to locale-specific string comparison when using the conditional command’s < and

> operators and the effect of interrupting a command list.

dirspell

If set, bash attempts spelling correction on directory names during word completion if the directory name initially supplied does not exist.

dotglob If set, bash includes filenames beginning with a ‘.’ in the results of pathname expansion.

execfail

If set, a non-interactive shell will not exit if it cannot execute the file specified as an argument to the exec builtin command. An interactive

shell does not exit if exec fails.

expand\_aliases

If set, aliases are expanded as described above under ALIASES. This option is enabled by default for interactive shells.

extdebug

If set, behavior intended for use by debuggers is enabled:

1. The -F option to the declare builtin displays the source file name and line number corresponding to each function name supplied as an argu-

ment.

2. If the command run by the DEBUG trap returns a non-zero value, the next command is skipped and not executed.

3. If the command run by the DEBUG trap returns a value of 2, and the shell is executing in a subroutine (a shell function or a shell script exe-

cuted by the . or source builtins), a call to return is simulated.

4. BASH\_ARGC and BASH\_ARGV are updated as described in their descriptions above.

5. Function tracing is enabled: command substitution, shell functions, and subshells invoked with ( command ) inherit the DEBUG and RETURN

traps.

6. Error tracing is enabled: command substitution, shell functions, and subshells invoked with ( command ) inherit the ERROR trap.

extglob If set, the extended pattern matching features described above under Pathname Expansion are enabled.

extquote

If set, $'string' and $"string" quoting is performed within ${parameter} expansions enclosed in double quotes. This option is enabled by default.

failglob

If set, patterns which fail to match filenames during pathname expansion result in an expansion error.

force\_fignore

If set, the suffixes specified by the FIGNORE shell variable cause words to be ignored when performing word completion even if the ignored words are

the only possible completions. See SHELL VARIABLES above for a description of FIGNORE. This option is enabled by default.

globstar

If set, the pattern \*\* used in a pathname expansion context will match a files and zero or more directories and subdirectories. If the pattern is

followed by a /, only directories and subdirectories match.

gnu\_errfmt

If set, shell error messages are written in the standard GNU error message format.

histappend

If set, the history list is appended to the file named by the value of the HISTFILE variable when the shell exits, rather than overwriting the file.

histreedit

If set, and readline is being used, a user is given the opportunity to re-edit a failed history substitution.

histverify

If set, and readline is being used, the results of history substitution are not immediately passed to the shell parser. Instead, the resulting line

is loaded into the readline editing buffer, allowing further modification.

hostcomplete

If set, and readline is being used, bash will attempt to perform hostname completion when a word containing a @ is being completed (see Completing

under READLINE above). This is enabled by default.

huponexit

If set, bash will send SIGHUP to all jobs when an interactive login shell exits.

interactive\_comments

If set, allow a word beginning with # to cause that word and all remaining characters on that line to be ignored in an interactive shell (see COM-

MENTS above). This option is enabled by default.

lithist If set, and the cmdhist option is enabled, multi-line commands are saved to the history with embedded newlines rather than using semicolon separators

where possible.

login\_shell

The shell sets this option if it is started as a login shell (see INVOCATION above). The value may not be changed.

mailwarn

If set, and a file that bash is checking for mail has been accessed since the last time it was checked, the message ‘‘The mail in mailfile has been

read’’ is displayed.

no\_empty\_cmd\_completion

If set, and readline is being used, bash will not attempt to search the PATH for possible completions when completion is attempted on an empty line.

nocaseglob

If set, bash matches filenames in a case-insensitive fashion when performing pathname expansion (see Pathname Expansion above).

nocasematch

If set, bash matches patterns in a case-insensitive fashion when performing matching while executing case or [[ conditional commands.

nullglob

If set, bash allows patterns which match no files (see Pathname Expansion above) to expand to a null string, rather than themselves.

progcomp

If set, the programmable completion facilities (see Programmable Completion above) are enabled. This option is enabled by default.

promptvars

If set, prompt strings undergo parameter expansion, command substitution, arithmetic expansion, and quote removal after being expanded as described

in PROMPTING above. This option is enabled by default.

restricted\_shell

The shell sets this option if it is started in restricted mode (see RESTRICTED SHELL below). The value may not be changed. This is not reset when

the startup files are executed, allowing the startup files to discover whether or not a shell is restricted.

shift\_verbose

If set, the shift builtin prints an error message when the shift count exceeds the number of positional parameters.

sourcepath

If set, the source (.) builtin uses the value of PATH to find the directory containing the file supplied as an argument. This option is enabled by

default.

xpg\_echo

If set, the echo builtin expands backslash-escape sequences by default.

suspend [-f]

Suspend the execution of this shell until it receives a SIGCONT signal. When the suspended shell is a background process, it can be restarted by the fg com-

mand. For more information, read the JOB CONTROL section. The suspend command can not suspend the login shell. However, when -f option is specified, suspend

command can suspend even login shell. The return status is 0 unless the shell is a login shell and -f is not supplied, or if job control is not enabled.

test expr

[ expr ]

Return a status of 0 or 1 depending on the evaluation of the conditional expression expr. Each operator and operand must be a separate argument. Expres-

sions are composed of the primaries described above under CONDITIONAL EXPRESSIONS. test does not accept any options, nor does it accept and ignore an argu-

ment of -- as signifying the end of options.

Expressions may be combined using the following operators, listed in decreasing order of precedence. The evaluation depends on the number of arguments; see

below.

! expr True if expr is false.

( expr )

Returns the value of expr. This may be used to override the normal precedence of operators.

expr1 -a expr2

True if both expr1 and expr2 are true.

expr1 -o expr2

True if either expr1 or expr2 is true.

test and [ evaluate conditional expressions using a set of rules based on the number of arguments.

0 arguments

The expression is false.

1 argument

The expression is true if and only if the argument is not null.

2 arguments

If the first argument is !, the expression is true if and only if the second argument is null. If the first argument is one of the unary conditional

operators listed above under CONDITIONAL EXPRESSIONS, the expression is true if the unary test is true. If the first argument is not a valid unary

conditional operator, the expression is false.

3 arguments

If the second argument is one of the binary conditional operators listed above under CONDITIONAL EXPRESSIONS, the result of the expression is the

result of the binary test using the first and third arguments as operands. The -a and -o operators are considered binary operators when there are

three arguments. If the first argument is !, the value is the negation of the two-argument test using the second and third arguments. If the first

argument is exactly ( and the third argument is exactly ), the result is the one-argument test of the second argument. Otherwise, the expression is

false.

4 arguments

If the first argument is !, the result is the negation of the three-argument expression composed of the remaining arguments. Otherwise, the expres-

sion is parsed and evaluated according to precedence using the rules listed above.

5 or more arguments

The expression is parsed and evaluated according to precedence using the rules listed above.

times Print the accumulated user and system times for the shell and for processes run from the shell. The return status is 0.

trap [-lp] [[arg] sigspec ...]

The command arg is to be read and executed when the shell receives signal(s) sigspec. If arg is absent (and there is a single sigspec) or -, each specified

signal is reset to its original disposition (the value it had upon entrance to the shell). If arg is the null string the signal specified by each sigspec is

ignored by the shell and by the commands it invokes. If arg is not present and -p has been supplied, then the trap commands associated with each sigspec are

displayed. If no arguments are supplied or if only -p is given, trap prints the list of commands associated with each signal. The -l option causes the

shell to print a list of signal names and their corresponding numbers. Each sigspec is either a signal name defined in <signal.h>, or a signal number. Sig-

nal names are case insensitive and the SIG prefix is optional.

If a sigspec is EXIT (0) the command arg is executed on exit from the shell. If a sigspec is DEBUG, the command arg is executed before every simple command,

for command, case command, select command, every arithmetic for command, and before the first command executes in a shell function (see SHELL GRAMMAR above).

Refer to the description of the extdebug option to the shopt builtin for details of its effect on the DEBUG trap. If a sigspec is RETURN, the command arg is

executed each time a shell function or a script executed with the . or source builtins finishes executing.

If a sigspec is ERR, the command arg is executed whenever a simple command has a non-zero exit status, subject to the following conditions. The ERR trap is

not executed if the failed command is part of the command list immediately following a while or until keyword, part of the test in an if statement, part of a

command executed in a && or ││ list, or if the command’s return value is being inverted via !. These are the same conditions obeyed by the errexit option.

Signals ignored upon entry to the shell cannot be trapped, reset or listed. Trapped signals that are not being ignored are reset to their original values in

a subshell or subshell environment when one is created. The return status is false if any sigspec is invalid; otherwise trap returns true.

type [-aftpP] name [name ...]

With no options, indicate how each name would be interpreted if used as a command name. If the -t option is used, type prints a string which is one of

alias, keyword, function, builtin, or file if name is an alias, shell reserved word, function, builtin, or disk file, respectively. If the name is not

found, then nothing is printed, and an exit status of false is returned. If the -p option is used, type either returns the name of the disk file that would

be executed if name were specified as a command name, or nothing if ‘‘type -t name’’ would not return file. The -P option forces a PATH search for each

name, even if ‘‘type -t name’’ would not return file. If a command is hashed, -p and -P print the hashed value, not necessarily the file that appears first

in PATH. If the -a option is used, type prints all of the places that contain an executable named name. This includes aliases and functions, if and only if

the -p option is not also used. The table of hashed commands is not consulted when using -a. The -f option suppresses shell function lookup, as with the

command builtin. type returns true if all of the arguments are found, false if any are not found.

ulimit [-HSTabcdefilmnpqrstuvx [limit]]

Provides control over the resources available to the shell and to processes started by it, on systems that allow such control. The -H and -S options specify

that the hard or soft limit is set for the given resource. A hard limit cannot be increased by a non-root user once it is set; a soft limit may be increased

up to the value of the hard limit. If neither -H nor -S is specified, both the soft and hard limits are set. The value of limit can be a number in the unit

specified for the resource or one of the special values hard, soft, or unlimited, which stand for the current hard limit, the current soft limit, and no

limit, respectively. If limit is omitted, the current value of the soft limit of the resource is printed, unless the -H option is given. When more than one

resource is specified, the limit name and unit are printed before the value. Other options are interpreted as follows:

-a All current limits are reported

-b The maximum socket buffer size

-c The maximum size of core files created

-d The maximum size of a process’s data segment

-e The maximum scheduling priority ("nice")

-f The maximum size of files written by the shell and its children

-i The maximum number of pending signals

-l The maximum size that may be locked into memory

-m The maximum resident set size (many systems do not honor this limit)

-n The maximum number of open file descriptors (most systems do not allow this value to be set)

-p The pipe size in 512-byte blocks (this may not be set)

-q The maximum number of bytes in POSIX message queues

-r The maximum real-time scheduling priority

-s The maximum stack size

-t The maximum amount of cpu time in seconds

-u The maximum number of processes available to a single user

-v The maximum amount of virtual memory available to the shell

-x The maximum number of file locks

-T The maximum number of threads

If limit is given, it is the new value of the specified resource (the -a option is display only). If no option is given, then -f is assumed. Values are in

1024-byte increments, except for -t, which is in seconds, -p, which is in units of 512-byte blocks, and -T, -b, -n, and -u, which are unscaled values. The

return status is 0 unless an invalid option or argument is supplied, or an error occurs while setting a new limit.

umask [-p] [-S] [mode]

The user file-creation mask is set to mode. If mode begins with a digit, it is interpreted as an octal number; otherwise it is interpreted as a symbolic

mode mask similar to that accepted by chmod(1). If mode is omitted, the current value of the mask is printed. The -S option causes the mask to be printed

in symbolic form; the default output is an octal number. If the -p option is supplied, and mode is omitted, the output is in a form that may be reused as

input. The return status is 0 if the mode was successfully changed or if no mode argument was supplied, and false otherwise.

unalias [-a] [name ...]

Remove each name from the list of defined aliases. If -a is supplied, all alias definitions are removed. The return value is true unless a supplied name is

not a defined alias.

unset [-fv] [name ...]

For each name, remove the corresponding variable or function. If no options are supplied, or the -v option is given, each name refers to a shell variable.

Read-only variables may not be unset. If -f is specified, each name refers to a shell function, and the function definition is removed. Each unset variable

or function is removed from the environment passed to subsequent commands. If any of COMP\_WORDBREAKS, RANDOM, SECONDS, LINENO, HISTCMD, FUNCNAME, GROUPS, or

DIRSTACK are unset, they lose their special properties, even if they are subsequently reset. The exit status is true unless a name is readonly.

wait [n ...]

Wait for each specified process and return its termination status. Each n may be a process ID or a job specification; if a job spec is given, all processes

in that job’s pipeline are waited for. If n is not given, all currently active child processes are waited for, and the return status is zero. If n speci-

fies a non-existent process or job, the return status is 127. Otherwise, the return status is the exit status of the last process or job waited for.

SEE ALSO

bash(1), sh(1)

GNU Bash-4.0 2004 Apr 20 BASH\_BUILTINS(1)

(END)

## KSH

待写。

# 目录及文件操作

## ls

语法：ls   [选项] 路径名称

说明：文件、目录列表命令

常用选项说明：

-a 列出所有文件，包括第一个字符为“.”的隐藏文件

-F 目录加/,执行文件加\*,符号连接后加@

-l 长列表输出，显示文件详细信息，每行一个文件，从左至右依次是：

文件存取模式 链接数 文件主 文件组 文件字节数 上次修改时间

其中文件存取模式用10个字母表示，从左至右的意义如下：

第一个字母表示文件种类，可以是以下几种情况：

|  |  |
| --- | --- |
| d | 为目录文件 |
| l | 为链接 |
| b | 为块文件 |
| c | 为字符型文件 |
| p | 为命名管道（FIFO) |
| - | 为普通文件 |

后面9个字母分别表示文件主、同组用户、其他用户对文件的权力，

用r表示可读，w 表示可写，x 表示可执行。

如果是设备文件，则在文件字节数处显示：主设备 从设备。

举例：

ls    查看当前目录下的文件

ls -l  详细查看当前目录下的文件

ls -al/bin 以长列表的形式列出目录 /bin 下的所有文件，包括隐藏文件

ls – l: 查询当前目录下所有文件和文件夹的读写执行权限

drwxrwxrwx或-rwxrwxrwx:d代表目录，-代表文件，r代表读权限，w代表写权限，x代表执行权限。红色三组代表拥有者的权限，绿色三组代表同组人的权限，蓝色三组代表其他组人的权限。其中权限用数字设置：1\*2^2 + 1\*2^1 + 1\*2^0 = 7代表；1\*2^2 + 1\*2^1 + 1\*2^0 = 7; 1\*2^2 + 1\*2^1 + 1\*2^0 = 7;

### $ man ls

LS(1) User Commands LS(1)

NAME

ls - list directory contents

SYNOPSIS

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

DESCRIPTION

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

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 octal escapes for nongraphic characters

--block-size=SIZE

use SIZE-byte blocks. 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

-C list entries by columns

--color[=WHEN]

colorize the output. WHEN defaults to ‘always’ or can be ‘never’ or ‘auto’. More info below

-d, --directory

list directory entries instead of contents, and do not dereference symbolic links

-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.

augment 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, print sizes in human readable format (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 like --block-size=1K

-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 non graphic characters

--show-control-chars

show non graphic characters as-is (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, 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

--sort=WORD

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

--time=WORD

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

--time-style=STYLE

with -l, show times using style STYLE: full-iso, long-iso, iso, locale, +FORMAT. FORMAT is interpreted like ‘date’; if FORMAT is FORMAT1<newline>FORMAT2,

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

-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

-U do not sort; list entries in directory order

-v natural sort of (version) numbers within text

-w, --width=COLS

assume screen width instead of current value

-x list entries by lines instead of by columns

-X sort alphabetically by entry extension

-1 list one file per line

SELinux options:

--lcontext

Display security context. Enable -l. Lines will probably be too wide for most displays.

-Z, --context

Display security context so it fits on most displays. Displays only mode, user, group, security context and file name.

--scontext

Display only security context and file name.

--help display this help and exit

--version

output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: KB 1000, K 1024, MB 1000\*1000, M 1024\*1024, and so on for G, T, P, E, Z, Y.

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 con-

nected 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).

## basename

**$ man basename**

BASENAME(1) User Commands BASENAME(1)

NAME

basename - strip directory and suffix from filenames

SYNOPSIS

basename NAME [SUFFIX]

basename OPTION

DESCRIPTION

Print NAME with any leading directory components removed. If specified, also remove a trailing

SUFFIX.

--help display this help and exit

--version

output version information and exit

EXAMPLES

basename /usr/bin/sort

Output "sort".

basename include/stdio.h .h

Output "stdio".

## pwd

语法：pwd

说明：本命令用于显示当前的工作目录

举例：

**pwd** 显示出当前的工作目录

## cd

使用：cd    路径名称

说明：本命令用于改变当前的工作目录，无参数时使用环境变量$HOME 作为其参数（$HOME一般为注册时进入的路径）。

举例：

**cd**   回到注册进入时的目录

**cd   /tmp**  进入 /tmp 目录

**cd ..** 进入上级目录（“..”代表上一级目录，“.”代码当前目录）

## rmdir

$ man rmdir

RMDIR(1) User Commands RMDIR(1)

NAME

rmdir - remove empty directories

SYNOPSIS

rmdir [OPTION]... DIRECTORY...

DESCRIPTION

Remove the DIRECTORY(ies), if they are empty.

--ignore-fail-on-non-empty

ignore each failure that is solely because a directory is non-empty

-p, --parents

remove DIRECTORY and its ancestors; e.g., ‘rmdir -p a/b/c’ is similar to ‘rmdir a/b/c a/b a’

-v, --verbose

output a diagnostic for every directory processed

--help display this help and exit

--version

output version information and exit

## mkdir

使用：mkdir 目录名称

说明：新建目录

举例：mkdir log 在当前目录下建立log目录

mkdir   log/record    在log目录下建立record目录

巧妙创建父目录

命令mkdir -p /home/adam/make/all/of/these/directories/会根据需要创建不存在的目录。何苦要浪费时间做这种傻事呢：mkdir make ； cd make ； mkdir all ； cd all ； mkdir of ； cd of … 切中要害，使用mkdir -p!

$ mkdir --help

Usage: mkdir [OPTION]... DIRECTORY...

Create the DIRECTORY(ies), if they do not already exist.

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

-m, --mode=MODE set file mode (as in chmod), not a=rwx - umask

-p, --parents no error if existing, make parent directories as needed

-v, --verbose print a message for each created directory

-Z, --context=CTX set the SELinux security context of each created

directory to CTX

--help display this help and exit

--version output version information and exit

$ man mkdir

MKDIR(1) User Commands MKDIR(1)

NAME

mkdir - make directories

SYNOPSIS

mkdir [OPTION]... DIRECTORY...

DESCRIPTION

Create the DIRECTORY(ies), if they do not already exist.

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

-m, --mode=MODE

set file mode (as in chmod), not a=rwx - umask

-p, --parents

no error if existing, make parent directories as needed

-v, --verbose

print a message for each created directory

-Z, --context=CTX

set the SELinux security context of each created directory to CTX

--help display this help and exit

--version

output version information and exit

## more

使用：more   [选项]   文件名

说明： 将文件显示在终端上，每次一屏，在左下部显示 －－more－－，若是从文件读出而非从管道，则在后面显示百分比，表示已显示的部分，按回车键则上滚一行，按空格键则上滚一屏，未显示完时可以使用more 命令中的子命令。

## cp

使用：cp   文件 目标

说明：将文件拷贝到目标上，目标不能与文件同名， 若目标是文件名，则拷贝的文件只能有一个，若目标是目录， 则拷贝的文件可以有多个，若目标文件不存在，则建立这个文件，若存在，则覆盖其以前的内容，若目标是目录，则将文件拷贝到这个目录下。

举例： cp   file1   file2   将文件 file1 拷贝到文件 file2

   cp   file1   file2 /tmp 将文件 file1 和文件 file2 拷贝到目录 /tmp 下

## mv

使用：mv   文件 目标

说明：将文件移动至目标，若目标是文件名，则相当于文件改名

举例： mv file1   file2         将文件 file1 改名为 file2

mv file1 file2 /tmp 将文件 file1 和文件 file2 移动到目录 /tmp 下

MV(1) User Commands MV(1)

NAME

mv - move (rename) files

SYNOPSIS

mv [OPTION]... [-T] SOURCE DEST

mv [OPTION]... SOURCE... DIRECTORY

mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION

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 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

--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

## rename

RENAME(1) Linux Programmer’s Manual RENAME(1)

NAME

rename - Rename files

SYNOPSIS

rename from to file...

rename -V

DESCRIPTION

rename will rename the specified files by replacing the first occurrence of from in their name by to.

-V, --version Display version information and exit.

For example, given the files

foo1, ..., foo9, foo10, ..., foo278, the commands

rename foo foo0 foo?

rename foo foo0 foo??

will turn them into foo001, ..., foo009, foo010, ..., foo278.

And

rename .htm .html \*.htm

will fix the extension of your html files.

SEE ALSO

mmv(1), mv(1)

## rm

使用：rm   文件 / rm -r   目录

说明：删除文件或目录

举例： rm file1    删除文件 file1

   rm –r /mytmp   递归地删除目录 /mytmp，及其子目录

$ man rm

RM(1) User Commands RM(1)

NAME

rm - remove files or directories

SYNOPSIS

rm [OPTION]... FILE...

DESCRIPTION

This manual page documents the GNU version of rm. rm removes each specified file. By default, it

does not remove directories.

If the -I or --interactive=once option is given, and there are more than three files or the -r, -R,

or --recursive are given, then rm prompts the user for whether to proceed with the entire opera-

tion. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the -f or --force option is

not given, or the -i or --interactive=always option is given, rm prompts the user for whether to

remove the file. If the response is not affirmative, the file is skipped.

OPTIONS

Remove (unlink) the FILE(s).

-f, --force

ignore nonexistent files, 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

-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 is usually possible to recover the contents of that

file. If you want more assurance that the contents are truly unrecoverable, consider using shred.

## chmod

使用：chmod [-R] 模式   文件...

  或 chmod [ugoa] {+|-|=} [rwxst] 文件...

说明：改变文件的存取模式，存取模式可表示为数字或符号串，例如：

chmod    nnnn   file ， n为0-7的数字，意义如下:

    4000   运行时可改变UID

    2000   运行时可改变GID

    1000   置粘着位

    0400   文件主可读

    0200   文件主可写

    0100   文件主可执行

    0040   同组用户可读

    0020   同组用户可写

    0010   同组用户可执行

    0004   其他用户可读

    0002   其他用户可写

    0001   其他用户可执行

nnnn 就是上列数字相加得到的，

例如 chmod 0777 file 是指将文件 file 存取权限置为所有用户可读可写可执行。

-R 递归地改变所有子目录下所有文件的存取模式

u 文件主user

g 同组用户group

o 其他用户other

a 所有用户all

+ 增加后列权限

- 取消后列权限

= 置成后列权限

r 可读

w 可写

x 可执行

s 运行时可置UID

t 运行时可置GID

举例：

chmod 0666 file1 file2 将文件 file1 及 file2 置为所有用户可读可写

chmod   u+x   file         对文件 file 增加文件主可执行权限

chmod   o-rwx    对文件file 取消其他用户的所有权限

chmod 777 文件名/用户名 授权

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.

-c, --changes like verbose but report only when a change is made

--no-preserve-root do not treat `/' specially (the default)

--preserve-root fail to operate recursively on `/'

-f, --silent, --quiet suppress most error messages

-v, --verbose output a diagnostic for every file processed

--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]))+'.

　　使用权限:所有使用者

　　使用方式:chmod [-cfvR] [--help] [--version] mode file...

　　说明:Linux/Unix 的档案存取权限分为三级:档案拥有者,群组,其他。利用 chmod 可以藉以控制档案如何被他人所存取。

　　把计:

　　mode:权限设定字串,格式如下:[ugoa...][[+-=][rwxX]...][,...],其中u 表示该档案的拥有者,g 表示与该档案的拥有者属于同一个群体(group)者,o 表示其他以外的人,a 表示这三者皆是。

　　+ 表示增加权限,- 表示取消权限,= 表示唯一设定权限。

　　r 表示可读取,w 表示可写入,x 表示可执行,X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。

　　-c:若该档案权限确实已经更改,才显示其更改动作

　　-f:若该档案权限无法被更改也不要显示错误讯息

　　-v:显示权限变更的详细资料

　　-R:对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)

　　--help:显示辅助说明

　　--version:显示版本

　　范例 :将档案 file1.txt 设为所有人皆可读取:

　　chmod ugo+r file1.txt

　　将档案 file1.txt 设为所有人皆可读取:

　　chmod a+r file1.txt

　　将档案 file1.txt 与 file2.txt 设为该档案拥有者,与其所属同一个群体者可写入,但其他以外的人则不可写入:

　　chmod ug+w,o-w file1.txt file2.txt

　　将 ex1.py 设定为只有该档案拥有者可以执行:

　　chmod u+x ex1.py

　　将目前目录下的所有档案与子目录皆设为任何人可读取:

　　chmod -R a+r \*

　　此外chmod也可以用数字来表示权限如 chmod 777 file

　　语法为：chmod abc file

　　其中a,b,c各为一个数字,分别表示User,Group,及Other的权限。

　　r=4,w=2,x=1

　　若要rwx属性则4+2+1=7；

　　若要rw-属性则4+2=6；

　　若要r-x属性则4+1=7。

　　范例：

　　chmod a=rwx file

　　和

　　chmod 777 file

　　效果相同

　　chmod ug=rwx,o=x file

　　和

　　chmod 771 file

　　效果相同

　　若用chmod 4755 filename可使此程式具有root的权限

## chown

使用：chown [-R] owner:[group] 文件...

说明：修改文件的文件主，文件的UID表示文件的文件主，文件主可用数字表示， 也可用一个有效的用户名表示，此命令改变一个文件的UID，仅当此文件的文件主或超级用户可使用。

-R 递归地改变所有子目录下所有文件的存取模式

owner    用户（使用用户名或ID均可）

group     用户组（使用用户组名或ID均可）

举例：

chown mary file 将文件 file 的文件主改为 mary

chown 150 file 将文件 file 的UID改为150

chown dualbase:informix tdfs 将tdfs文件修改为dualbase用户，informix用户组

## diff

使用：diff   文件1    文件2

说明：比较两个文本文件，将不同的行列出来

举例：

diff scf0.log scf1.log      比较scf0.log和scf1.log的不同

## wc

使用：wc   文件 ...

说明：统计文件的行、字、字符数

举例：

wc scf0.log scf1.log      统计scf0.log和scf1.log两个文件的行、字和字符数

**$ man wc**

WC(1) User Commands WC(1)

NAME

wc - print newline, word, and byte counts for each file

SYNOPSIS

wc [OPTION]... [FILE]...

wc [OPTION]... --files0-from=F

DESCRIPTION

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is spec-

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

|  |  |
| --- | --- |
| **-c, --bytes** | print the byte counts |
| **-m, --chars** | print the character counts |
| **-l, --lines** | print the newline counts |
| **--files0-from=F** | read input from the files specified by NUL-terminated names in file F; If F is - then read names from standard input |
| **-L, --max-line-length** | print the length of the longest line |
| **-w, --words** | print the word counts |
| **--help** | display this help and exit |
| **--version** | output version information and exit |

## touch

使用：touch [-amc] [mmddhhmm[yy]] 文件...

说明：将指定文件的访问时间和修改时间改变，若指定文件不存在则创建之，若无指定时间，则使用当前时间，返回值是未成功改变时间的文件个数，包括不存在而又未能创建的文件。

-a 只改变访问时间

-m 只改变修改时间

-c 若文件不存在，不创建它且不作提示

mmddhhmm[yy]     两位表示 月日时分[年]

举例：

touch file     更新文件file的时间为当前时间

touch 0701000097 HongKong   将文件HongKong的时间改为97年7月1日0时0分

## find

使用：find    路径名...    表达式

说明：find 命令递归地遍历指定路径下的每个文件和子目录，看该文件是否能使表达式值为真，以下 n 代表一个十进制整数，+n 代表大于 n ， -n 代表小于 n ，下面是合法表达式说明：

-name 模式   文件名与模式匹配则为真，(\ 为转意符)

-size n[c]   文件块长度为 n 则真(一块为512字节)，若

     有c 选项，则文件字节长度为 n 则真

-atime n   若文件的最近访问时间为 n 天前则为真，

     find 命令将改变其访问的目录的访问时间

-mtime n   若文件的最近修改时间为 n 天前则为真

-ctime n   若文件状态为 n 天前改变则为真

-exec 命令 { }\; 若命令返回值为0则真，{ }内为命令参数，

     此命令必须以 \; 为结束

-ok 命令    { }\; 与 exec 相同，只是在命令执行前先提示，若

     回答 y 则执行命令

-print   显示输出使表达式为真的文件名

-newer 文件 若文件的访问时间比newer 指定的文件新则真

举例：

find / -name find\* -print

   从根目录开始搜索文件名如 find\* 的文件并显示之

find ./ -exec sleep{1}\; -print

     每秒显示一个当前目录下的文件

find $HOME \(-name a.out -o -name '\*.o' \) -atime +7 -exec   rm {} \;

从$HOME目录开始搜索，删除所有文件名为a.out 或 \*.o 且访问时间在7天前的文件 ls   [选项] 路径名称

## grep

使用：grep [选项] 模式 [文件...]

说明：在指定的文件中搜索模式，并显示所有包含模式的行，模式是一个正规表达式，在使用正规表达式时，最好将其引在单引号(') 中，若指定文件为缺省，则使用标准输入，正规表达式可以是：

|  |  |
| --- | --- |
| . | 匹配任意一个字符 |
| \* | 匹配0个或多个\*前的字符 |
| ^ | 匹配行开头 |
| $ | 匹配行结尾 |
| [] | 匹配[ ]中的任意一个字符，[]中可用 - 表示范围，例如[a-z]表示字母a 至z 中的任意一个 |
| \ | 转意字符 |

命令中的选项为：

|  |  |
| --- | --- |
| -i | 模式中字母不区分大小写 |
| -n | 显示模式所在行的行号 |
| -v | 显示所有不包含模式的行 |

举例：

grep   'good' \*   在所有文件中搜索含有字符串 good 的行

grep   '^myline' mytext 在文件mytext中搜索行首出现myline字符串的行

## cut

cut是以每一行为一个处理对象的，这种机制和sed是一样的。（关于sed的入门文章将在近期发布）

2 cut一般以什么为依据呢? 也就是说，我怎么告诉cut我想定位到的剪切内容呢?

cut命令主要是接受三个定位方法：

第一，字节（bytes），用选项-b

第二，字符（characters），用选项-c

第三，域（fields），用选项-f

3 以“字节”定位，给个最简单的例子?

举个例子吧，当你执行ps命令时，会输出类似如下的内容：

[rocrocket@rocrocket programming]$ who

rocrocket :0 2009-01-08 11:07

rocrocket pts/0 2009-01-08 11:23 (:0.0)

rocrocket pts/1 2009-01-08 14:15 (:0.0)

如果我们想提取每一行的第3个字节，就这样：

[rocrocket@rocrocket programming]$ who|cut -b 3

c

c

c

看明白了吧，-b后面可以设定要提取哪一个字节，其实-b和3之间没有空格也是可以的，但推荐有空格:)

4 如果“字节”定位中，我想提取第3，第4、第5和第8个字节，怎么办?

-b支持形如3-5的写法，而且多个定位之间用逗号隔开就成了。看看例子吧：

[rocrocket@rocrocket programming]$ who|cut -b 3-5,8

croe

croe

croe

但有一点要注意，cut命令如果使用了-b选项，那么执行此命令时，cut会先把-b后面所有的定位进行从小到大排序，然后再提取。可不能颠倒定位的顺序哦。这个例子就可以说明这个问题：

[rocrocket@rocrocket programming]$ who|cut -b 8,3-5

croe

croe

croe

5 还有哪些类似“3-5”这样的小技巧，列举一下吧!

[rocrocket@rocrocket programming]$ who

rocrocket :0 2009-01-08 11:07

rocrocket pts/0 2009-01-08 11:23 (:0.0)

rocrocket pts/1 2009-01-08 14:15 (:0.0)

[rocrocket@rocrocket programming]$ who|cut -b -3

roc

roc

roc

[rocrocket@rocrocket programming]$ who|cut -b 3-

crocket :0 2009-01-08 11:07

crocket pts/0 2009-01-08 11:23 (:0.0)

crocket pts/1 2009-01-08 14:15 (:0.0)

想必你也看到了，-3表示从第一个字节到第三个字节，而3-表示从第三个字节到行尾。如果你细心，你可以看到这两种情况下，都包括了第三个字节“c”。

如果我执行who|cut -b -3,3-，你觉得会如何呢？答案是输出整行，不会出现连续两个重叠的c的。看：

[rocrocket@rocrocket programming]$ who|cut -b -3,3-

rocrocket :0 2009-01-08 11:07

rocrocket pts/0 2009-01-08 11:23 (:0.0)

rocrocket pts/1 2009-01-08 14:15 (:0.0)

6 给个以字符为定位标志的最简单的例子吧!

下面例子你似曾相识，提取第3，第4，第5和第8个字符：

[rocrocket@rocrocket programming]$ who|cut -c 3-5,8

croe

croe

croe

不过，看着怎么和-b没有什么区别啊？莫非-b和-c作用一样? 其实不然，看似相同，只是因为这个例子举的不好，who输出的都是单字节字符，所以用-b和-c没有区别，如果你提取中文，区别就看出来了，来，看看中文提取的情况：

[rocrocket@rocrocket programming]$ cat cut\_ch.txt

星期一

星期二

星期三

星期四

[rocrocket@rocrocket programming]$ cut -b 3 cut\_ch.txt

�

�

�

�

[rocrocket@rocrocket programming]$ cut -c 3 cut\_ch.txt

一

二

三

四

看到了吧，用-c则会以字符为单位，输出正常；而-b只会傻傻的以字节（8位二进制位）来计算，输出就是乱码。

既然提到了这个知识点，就再补充一句，如果你学有余力，就提高一下。

当遇到多字节字符时，可以使用-n选项，-n用于告诉cut不要将多字节字符拆开。例子如下：

[rocrocket@rocrocket programming]$ cat cut\_ch.txt |cut -b 2

�

�

�

�

[rocrocket@rocrocket programming]$ cat cut\_ch.txt |cut -nb 2

[rocrocket@rocrocket programming]$ cat cut\_ch.txt |cut -nb 1,2,3

星

星

星

星

6 域是怎么回事呢？解释解释:)

为什么会有“域”的提取呢，因为刚才提到的-b和-c只能在固定格式的文档中提取信息，而对于非固定格式的信息则束手无策。这时候“域”就派上用场了。

（下面的讲解内容是在假设你对/etc/passwd文件的内容和组织形式比较了解的情况下进行的。）

如果你观察过/etc/passwd文件，你会发现，它并不像who的输出信息那样具有固定格式，而是比较零散的排放。但是，冒号在这个文件的每一行中都起到了非常重要的作用，冒号用来隔开每一个项。

我们很幸运，cut命令提供了这样的提取方式，具体的说就是设置“间隔符”，再设置“提取第几个域”，就OK了！

以/etc/passwd的前五行内容为例：

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1

root

bin

daemon

adm

lp

看到了吧，用-d来设置间隔符为冒号，然后用-f来设置我要取的是第一个域，再按回车，所有的用户名就都列出来了！呵呵有成就感吧！

当然，在设定-f时，也可以使用例如3-5或者4-类似的格式：

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1,3-5

root:0:0:root

bin:1:1:bin

daemon:2:2:daemon

adm:3:4:adm

lp:4:7:lp

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1,3-5,7

root:0:0:root:/bin/bash

bin:1:1:bin:/sbin/nologin

daemon:2:2:daemon:/sbin/nologin

adm:3:4:adm:/sbin/nologin

lp:4:7:lp:/sbin/nologin

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f -2

root:x

bin:x

daemon:x

adm:x

lp:x

7 如果遇到空格和制表符时，怎么分辨呢？我觉得有点乱，怎么办？

有时候制表符确实很难辨认，有一个方法可以看出一段空格到底是由若干个空格组成的还是由一个制表符组成的。

[rocrocket@rocrocket programming]$ cat tab\_space.txt

this is tab    finish.

this is several space finish.

[rocrocket@rocrocket programming]$ sed -n l tab\_space.txt

this is tab\tfinish.$

this is several space finish.$

看到了吧，如果是制表符（TAB），那么会显示为\t符号，如果是空格，就会原样显示。

通过此方法即可以判断制表符和空格了。

注意，上面sed -n后面的字符是L的小写字母哦，不要看错。（字母l、数字1还有或运算｜真是难分辨啊…，看来这三个比制表符还难分辨…）

8 我应该在cut -d中用什么符号来设定制表符或空格呢?

悄悄的告诉你，cut的-d选项的默认间隔符就是制表符，所以当你就是要使用制表符的时候，完全就可以省略-d选项，而直接用－f来取域就可以了！放心，相信我！

如果你设定一个空格为间隔符，那么就这样：

[rocrocket@rocrocket programming]$ cat tab\_space.txt |cut -d ' ' -f 1

this

this

注意，两个单引号之间可确实要有一个空格哦，不能偷懒。

而且，你只能在-d后面设置一个空格，可不许设置多个空格，因为cut只允许间隔符是一个字符。

[rocrocket@rocrocket programming]$ cat tab\_space.txt |cut -d ' ' -f 1

cut: the delimiter must be a single character

Try `cut --help' for more information.

9 我想将ps和cut命令配合使用时，怎么总是在最后两行出现重复现象？

这个问题的具体描述是如下这样的。

当cut和ps配合时：

[rocrocket@rocrocket programming]$ ps

PID TTY TIME CMD

2977 pts/0 00:00:00 bash

5032 pts/0 00:00:00 ps

[rocrocket@rocrocket programming]$ ps|cut -b3

P

9

0

0

看，最后的0重复了两次！！而且，我也试过ps ef或ps aux均有此问题。

而当ps和其他命令配合时，均无此问题，例如cut和who配合则正常：

[rocrocket@rocrocket programming]$ who

rocrocket :0 2009-01-08 11:07

rocrocket pts/0 2009-01-08 11:23 (:0.0)

rocrocket pts/1 2009-01-08 14:15 (:0.0)

[rocrocket@rocrocket programming]$ who|cut -b3

c

c

c

这个看似怪异的令我百思不得其解的问题，得到了sunway的解答，在此非常感谢他。我发问的原帖地址在［此处］。

其实这个问题是这样的，ps|cut会自身创建一个进程，所以当ps时也会提取出这个进程，然后通过管道输出到cut，所以cut截取后，就多出了一行，之所以会重复上一行内容，是由于我们恰巧取到了和上一行内容相同的字符而已。

你测试下执行ps和ps|cat就知道原因了！:)

10 cut有哪些缺陷和不足？

猜出来了吧？对，就是在处理多空格时。

如果文件里面的某些域是由若干个空格来间隔的，那么用cut就有点麻烦了，因为cut只擅长处理“以一个字符间隔”的文本内容

CUT(1) User Commands CUT(1)

NAME

cut - remove sections from each line of files

SYNOPSIS

cut OPTION... [FILE]...

DESCRIPTION

Print selected parts of lines from each FILE to standard output.

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

-b, --bytes=LIST

select only these bytes

-c, --characters=LIST

select only these characters

-d, --delimiter=DELIM

use DELIM instead of TAB for field delimiter

-f, --fields=LIST

select only these fields; also print any line that contains no delimiter character, unless

the -s option is specified

-n with -b: don’t split multibyte characters

--complement

complement the set of selected bytes, characters or fields

-s, --only-delimited

do not print lines not containing delimiters

--output-delimiter=STRING

use STRING as the output delimiter the default is to use the input delimiter

--help display this help and exit

--version

output version information and exit

Use one, and only one of -b, -c or -f. Each LIST is made up of one range, or many ranges separated

by commas. Selected input is written in the same order that it is read, and is written exactly

once. Each range is one of:

N N’th byte, character or field, counted from 1

N- from N’th byte, character or field, to end of line

N-M from N’th to M’th (included) byte, character or field

-M from first to M’th (included) byte, character or field

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

## vi

NAME：vi, view, vedit - screen-oriented (visual) text editor

使用：vi 文件名...

说明：vi 是一个全屏幕编辑器，在命令后跟多个文件时，在第一个文件关闭后，会自动打开第二个文件，依次类推。使用UNIX开发，vi是必须掌握的。

vi 工作模式分为命令模式和输入模式。在输入模式下，输入的一切作为文本插入到当前编辑的缓冲区。在命令模式下，输入的字符都将被翻译成命令。

在输入模式下按“ESC”键进入命令模式。

### 在命令模式下，进入输入方式的命令：

|  |  |
| --- | --- |
| ***a*** | 在当前光标的后面插入文本 |
| ***A*** | 在当前光标的所在的行末插入文本 |
| ***C*** | 修改当前行光标后面的文本 |
| ***cw*** | 修改当前词光标后面的字符 |
| ***i*** | 在当前光标处添加新文本 |
| ***I*** | 在当前行开头插入新文本 |
| ***o*** | 在当前行后插入一行 |
| ***O*** | 在当前行前插入一行 |
| ***s*** | 用一串字符替换当前字符 |

### 光标移动命令：

|  |  |
| --- | --- |
| h, | 左行键, backspace 光标左移 |
| j, | 下行键   光标下移 |
| k, | 上行键   光标上移 |
| l, | 右行键, space   光标右移 |
| 0（数字0） | 移光标到当前行首 |
| $ | 移光标到当前行末 |
| b, B | 光标前移一个词，光标落在第一个字符（B忽略标点） |
| w, W | 光标下移一个词，光标落在第一个字符（C忽略标点） |
| ( | 移光标到上一句开头 |
| ) | 移光标到下一句开头 |
| { | 移光标到上一段（段一般以空行为依据）开头 |
| } | 移光标到下一段（段一般以空行为依据）开头 |
| [[ | 移光标到上一节（节一般以{为依据）开头 |
| ]] | 移光标到下一节（节一般以{为依据）开头 |
| H | 移动光标到当前缓冲区的第一行 |
| L | 移动光标到当缓冲区的最后一行 |
| M | 移动光标到当缓冲区的中间行 |

### 编辑区移动命令（^表示Ctrl键）：

|  |  |
| --- | --- |
| ^B | 上移（回退）一屏 |
| ^U | 上移（回退）半屏 |
| ^F | 下移（前移）一屏 |
| ^D | 下移（前移）半屏 |

### 到一个行号

: n    到第n行

nG    到第n行

1G    到文件第1行

G    到文件尾

### vi编辑命令（包括上面提到的进入输入方式的所以命令）还有下面一些常用的：

#### 修改命令

r 替换一个字符但不进入输入模式

~ 大小写替换

### 取消和重复改变

|  |  |
| --- | --- |
| . | 重复上次命令来修改缓冲区 |
| u | 取消上次最后一条命令 |

### 删除文本：

|  |  |
| --- | --- |
| dd | 删除当前行 |
| d n d | 删除当前下面的n行 |
| dG | 删除到文件尾 |
| D | 删除到行末 |
| x | 删除当前的字符 |
| :line1, line2d | 删除line1到line2之间的行 |
| :line1, $ d | 删除line1到文件末的所有行（$代表文件末） |

### 复制和移动

（拷贝时，一般是存储到缺省缓冲区，可以采用“缓冲区名”的方式指定存储的缓冲区名称，缓冲区名可以是“a~z”的字符，如“"a”指定a缓冲取，“"g”指定g缓冲区；缺省缓冲区不能在文件间使用，但命名缓冲区可以在不同的文件间切换使用，此处的切换是指的不退出vi，而调入新文件的切换；当在不同的文件间进行拷贝时，就可以采用这种命名缓冲区，命名缓冲区的引用采用“"缓冲区名”的方式；可以使用命令“m标记名”方式定义某字符标记当前行，标记名可以是“a~z”的字符，如mj，使用j标记当前行，标记的引用采用“'字符”的方式）：

yy 复制当前行（到缺省缓冲区）

"ayy 复制当前行（到a缓冲区）

yny 复制当前n行（到缺省缓冲区）

"ayny 复制当前n行（到a缓冲区）

y'j 复制当前行到字符j所标记行之间的行到缺省缓冲区

"ay'j 复制当前行到字符j所标记行之间的行到a缓冲区

p 粘贴当前缓冲区内容到光标位置之后

P 粘贴当前缓冲区内容到光标位置之前

"ap 粘贴a缓冲区内容到光标位置之后

"aP 粘贴a缓冲区内容到光标位置之后

### 查找字符串

|  |  |
| --- | --- |
| / 模式 | 向前搜索模式，将光标移动到模式出现处，模式是一个正规表达式 |
| ? 模式 | 向后搜索模式，将光标移动到模式出现处，模式是一个正规表达式 |
| n | 重复上次搜索 |

### 下面是vi的命令集列表

|  |  |
| --- | --- |
| ^B | 退回前一页，前面加数字表示重复次数，每次换页时保留上一页的两行 |
| ^D | 在命令模式下，表示下滚屏幕的一半，在输入模式下，表示回退至左边的自动缩进处 |
| ^E | 显示屏幕底线之下的一行 |
| ^F | 前进一页，前面加数字表示重复次数，每次换页时保留上一页的两行 |
| ^G | 显示当前文件名，当前行号和文件总行数，并用百分号当前行在整个文件中的位置 |
| ^H(退格) | 在命令模式下，光标左移一格；在输入模式下，删去前面的字符 |
| ^I(TAB) | 在输入模式下，产生一串空格 |
| ^J(LF) | 光标下移一行 |
| ^L | 刷新屏幕，即将屏幕重新显示 |
| ^M(回车) | 在命令模式下，光标移动至下行开头 |

### 在“：”提示符下，常用命令如下:

|  |  |
| --- | --- |
| :w | 当前文件存盘 |
| :w! | 强制存盘 |
| :w 文件 | 将内容写入指定文件 |
| :w! 文件 | 强制写入指定文件 |
| :x，y w 文件 | 将 x至 y 行写入指定文件中 |
| :r 文件 | 将文件读到光标位置 |
| :r ! 命令 | 将系统命令的输出读到光标位置 |
| :q | 退出编辑 |
| :q! | 强制退出 |
| :x | 与命令ZZ相同 |
| :e 文件名 | 编辑另一文件 |
| :e ! | 重新编辑文件，放弃任何改变 |
| :sh | 执行sh，结束后回到编辑 |
| :! 命令 | 执行命令后回到编辑 |
| :n | 编辑下一文件 |
| :n 文件表 | 重新定义待编辑文件表 |
| :set | 设置 vi 的选项，例如 set nu 表示每行前显示行号，在选项前加no则表示清除该选项，例如 set nonu 表示每行前不显示行号。下面是一些常用的选项:   |  |  | | --- | --- | | ai | 自动缩进 | | aw | 编辑下一文件前自动存盘 | | ic | 查找字符串时不区分大小写 | | nu | 每行前显示行号 | | sm | 输入)及}时显示与之配对的( 或 { | | slow | 插入时延迟屏幕刷新 | | ws | 使查找能绕过文件尾从头进行 | | wa | 写文件之前不作对文件的检查 | |

### $ vim --help

VIM - Vi IMproved 7.2 (2008 Aug 9, compiled Feb 17 2012 10:23:31)

usage: vim [arguments] [file ..] edit specified file(s)

or: vim [arguments] - read text from stdin

or: vim [arguments] -t tag edit file where tag is defined

or: vim [arguments] -q [errorfile] edit file with first error

Arguments:

-- Only file names after this

-v Vi mode (like "vi")

-e Ex mode (like "ex")

-s Silent (batch) mode (only for "ex")

-d Diff mode (like "vimdiff")

-y Easy mode (like "evim", modeless)

-R Readonly mode (like "view")

-Z Restricted mode (like "rvim")

-m Modifications (writing files) not allowed

-M Modifications in text not allowed

-b Binary mode

-l Lisp mode

-C Compatible with Vi: 'compatible'

-N Not fully Vi compatible: 'nocompatible'

-V[N][fname] Be verbose [level N] [log messages to fname]

-D Debugging mode

-n No swap file, use memory only

-r List swap files and exit

-r (with file name) Recover crashed session

-L Same as -r

-A start in Arabic mode

-H Start in Hebrew mode

-F Start in Farsi mode

-T <terminal> Set terminal type to <terminal>

-u <vimrc> Use <vimrc> instead of any .vimrc

--noplugin Don't load plugin scripts

-p[N] Open N tab pages (default: one for each file)

-o[N] Open N windows (default: one for each file)

-O[N] Like -o but split vertically

+ Start at end of file

+<lnum> Start at line <lnum>

--cmd <command> Execute <command> before loading any vimrc file

-c <command> Execute <command> after loading the first file

-S <session> Source file <session> after loading the first file

-s <scriptin> Read Normal mode commands from file <scriptin>

-w <scriptout> Append all typed commands to file <scriptout>

-W <scriptout> Write all typed commands to file <scriptout>

-x Edit encrypted files

--startuptime <file> Write startup timing messages to <file>

-i <viminfo> Use <viminfo> instead of .viminfo

-h or --help Print Help (this message) and exit

--version Print version information and exit

VIM(1) VIM(1)

NAME

vim - Vi IMproved, a programmers text editor

SYNOPSIS

vim [options] [file ..]

vim [options] -

vim [options] -t tag

vim [options] -q [errorfile]

ex

view

gvim gview evim eview

rvim rview rgvim rgview

DESCRIPTION

Vim is a text editor that is upwards compatible to Vi. It can be used to edit all kinds of plain

text. It is especially useful for editing programs.

There are a lot of enhancements above Vi: multi level undo, multi windows and buffers, syntax high-

lighting, command line editing, filename completion, on-line help, visual selection, etc.. See

":help vi\_diff.txt" for a summary of the differences between Vim and Vi.

While running Vim a lot of help can be obtained from the on-line help system, with the ":help" com-

mand. See the ON-LINE HELP section below.

Most often Vim is started to edit a single file with the command

vim file

More generally Vim is started with:

vim [options] [filelist]

If the filelist is missing, the editor will start with an empty buffer. Otherwise exactly one out

of the following four may be used to choose one or more files to be edited.

file .. A list of filenames. The first one will be the current file and read into the buffer.

The cursor will be positioned on the first line of the buffer. You can get to the

other files with the ":next" command. To edit a file that starts with a dash, precede

the filelist with "--".

- The file to edit is read from stdin. Commands are read from stderr, which should be a

tty.

-t {tag} The file to edit and the initial cursor position depends on a "tag", a sort of goto

label. {tag} is looked up in the tags file, the associated file becomes the current

file and the associated command is executed. Mostly this is used for C programs, in

which case {tag} could be a function name. The effect is that the file containing that

function becomes the current file and the cursor is positioned on the start of the

function. See ":help tag-commands".

-q [errorfile]

Start in quickFix mode. The file [errorfile] is read and the first error is displayed.

If [errorfile] is omitted, the filename is obtained from the ’errorfile’ option

(defaults to "AztecC.Err" for the Amiga, "errors.err" on other systems). Further

errors can be jumped to with the ":cn" command. See ":help quickfix".

Vim behaves differently, depending on the name of the command (the executable may still be the same

file).

vim The "normal" way, everything is default.

ex Start in Ex mode. Go to Normal mode with the ":vi" command. Can also be done with the

"-e" argument.

view Start in read-only mode. You will be protected from writing the files. Can also be done

with the "-R" argument.

gvim gview

The GUI version. Starts a new window. Can also be done with the "-g" argument.

evim eview

The GUI version in easy mode. Starts a new window. Can also be done with the "-y" argu-

ment.

rvim rview rgvim rgview

Like the above, but with restrictions. It will not be possible to start shell commands,

or suspend Vim. Can also be done with the "-Z" argument.

OPTIONS

The options may be given in any order, before or after filenames. Options without an argument can be combined after a single dash.

+[num] For the first file the cursor will be positioned on line "num". If "num" is missing,

the cursor will be positioned on the last line.

+/{pat} For the first file the cursor will be positioned on the first occurrence of {pat}. See

":help search-pattern" for the available search patterns.

+{command}

-c {command}

{command} will be executed after the first file has been read. {command} is inter-

preted as an Ex command. If the {command} contains spaces it must be enclosed in dou-

ble quotes (this depends on the shell that is used). Example: Vim "+set si" main.c

Note: You can use up to 10 "+" or "-c" commands.

-S {file} {file} will be sourced after the first file has been read. This is equivalent to -c

"source {file}". {file} cannot start with ’-’. If {file} is omitted "Session.vim" is

used (only works when -S is the last argument).

--cmd {command}

Like using "-c", but the command is executed just before processing any vimrc file.

You can use up to 10 of these commands, independently from "-c" commands.

-A If Vim has been compiled with ARABIC support for editing right-to-left oriented files

and Arabic keyboard mapping, this option starts Vim in Arabic mode, i.e. ’arabic’ is

set. Otherwise an error message is given and Vim aborts.

-b Binary mode. A few options will be set that makes it possible to edit a binary or exe-

cutable file.

-C Compatible. Set the ’compatible’ option. This will make Vim behave mostly like Vi,

even though a .vimrc file exists.

-d Start in diff mode. There should be two, three or four file name arguments. Vim will

open all the files and show differences between them. Works like vimdiff(1).

-d {device} Open {device} for use as a terminal. Only on the Amiga. Example: "-d

con:20/30/600/150".

-D Debugging. Go to debugging mode when executing the first command from a script.

-e Start Vim in Ex mode, just like the executable was called "ex".

-E Start Vim in improved Ex mode, just like the executable was called "exim".

-f Foreground. For the GUI version, Vim will not fork and detach from the shell it was

started in. On the Amiga, Vim is not restarted to open a new window. This option

should be used when Vim is executed by a program that will wait for the edit session to

finish (e.g. mail). On the Amiga the ":sh" and ":!" commands will not work.

--nofork Foreground. For the GUI version, Vim will not fork and detach from the shell it was

started in.

-F If Vim has been compiled with FKMAP support for editing right-to-left oriented files

and Farsi keyboard mapping, this option starts Vim in Farsi mode, i.e. ’fkmap’ and

’rightleft’ are set. Otherwise an error message is given and Vim aborts.

-g If Vim has been compiled with GUI support, this option enables the GUI. If no GUI sup-

port was compiled in, an error message is given and Vim aborts.

-h Give a bit of help about the command line arguments and options. After this Vim exits.

-H If Vim has been compiled with RIGHTLEFT support for editing right-to-left oriented

files and Hebrew keyboard mapping, this option starts Vim in Hebrew mode, i.e. ’hkmap’

and ’rightleft’ are set. Otherwise an error message is given and Vim aborts.

-i {viminfo}

When using the viminfo file is enabled, this option sets the filename to use, instead

of the default "~/.viminfo". This can also be used to skip the use of the .viminfo

file, by giving the name "NONE".

-L Same as -r.

-l Lisp mode. Sets the ’lisp’ and ’showmatch’ options on.

-m Modifying files is disabled. Resets the ’write’ option. You can still modify the

buffer, but writing a file is not possible.

-M Modifications not allowed. The ’modifiable’ and ’write’ options will be unset, so that

changes are not allowed and files can not be written. Note that these options can be

set to enable making modifications.

-N No-compatible mode. Reset the ’compatible’ option. This will make Vim behave a bit

better, but less Vi compatible, even though a .vimrc file does not exist.

-n No swap file will be used. Recovery after a crash will be impossible. Handy if you

want to edit a file on a very slow medium (e.g. floppy). Can also be done with ":set

uc=0". Can be undone with ":set uc=200".

-nb Become an editor server for NetBeans. See the docs for details.

-o[N] Open N windows stacked. When N is omitted, open one window for each file.

-O[N] Open N windows side by side. When N is omitted, open one window for each file.

-p[N] Open N tab pages. When N is omitted, open one tab page for each file.

-R Read-only mode. The ’readonly’ option will be set. You can still edit the buffer, but

will be prevented from accidently overwriting a file. If you do want to overwrite a

file, add an exclamation mark to the Ex command, as in ":w!". The -R option also

implies the -n option (see below). The ’readonly’ option can be reset with ":set

noro". See ":help ’readonly’".

-r List swap files, with information about using them for recovery.

-r {file} Recovery mode. The swap file is used to recover a crashed editing session. The swap

file is a file with the same filename as the text file with ".swp" appended. See

":help recovery".

-s Silent mode. Only when started as "Ex" or when the "-e" option was given before the

"-s" option.

-s {scriptin}

The script file {scriptin} is read. The characters in the file are interpreted as if

you had typed them. The same can be done with the command ":source! {scriptin}". If

the end of the file is reached before the editor exits, further characters are read

from the keyboard.

-T {terminal}

Tells Vim the name of the terminal you are using. Only required when the automatic way

doesn’t work. Should be a terminal known to Vim (builtin) or defined in the termcap or

terminfo file.

-u {vimrc} Use the commands in the file {vimrc} for initializations. All the other initializa-

tions are skipped. Use this to edit a special kind of files. It can also be used to

skip all initializations by giving the name "NONE". See ":help initialization" within

vim for more details.

-U {gvimrc} Use the commands in the file {gvimrc} for GUI initializations. All the other GUI ini-

tializations are skipped. It can also be used to skip all GUI initializations by giv-

ing the name "NONE". See ":help gui-init" within vim for more details.

-V[N] Verbose. Give messages about which files are sourced and for reading and writing a

viminfo file. The optional number N is the value for ’verbose’. Default is 10.

-v Start Vim in Vi mode, just like the executable was called "vi". This only has effect

when the executable is called "ex".

-w {scriptout}

All the characters that you type are recorded in the file {scriptout}, until you exit

Vim. This is useful if you want to create a script file to be used with "vim -s" or

":source!". If the {scriptout} file exists, characters are appended.

-W {scriptout}

Like -w, but an existing file is overwritten.

-x Use encryption when writing files. Will prompt for a crypt key.

-X Don’t connect to the X server. Shortens startup time in a terminal, but the window

title and clipboard will not be used.

-y Start Vim in easy mode, just like the executable was called "evim" or "eview". Makes

Vim behave like a click-and-type editor.

-Z Restricted mode. Works like the executable starts with "r".

-- Denotes the end of the options. Arguments after this will be handled as a file name.

This can be used to edit a filename that starts with a ’-’.

--echo-wid GTK GUI only: Echo the Window ID on stdout.

--help Give a help message and exit, just like "-h".

--literal Take file name arguments literally, do not expand wildcards. This has no effect on

Unix where the shell expands wildcards.

--noplugin Skip loading plugins. Implied by -u NONE.

--remote Connect to a Vim server and make it edit the files given in the rest of the arguments.

If no server is found a warning is given and the files are edited in the current Vim.

--remote-expr {expr}

Connect to a Vim server, evaluate {expr} in it and print the result on stdout.

--remote-send {keys}

Connect to a Vim server and send {keys} to it.

--remote-silent

As --remote, but without the warning when no server is found.

--remote-wait

As --remote, but Vim does not exit until the files have been edited.

--remote-wait-silent

As --remote-wait, but without the warning when no server is found.

--serverlist

List the names of all Vim servers that can be found.

--servername {name}

Use {name} as the server name. Used for the current Vim, unless used with a --remote

argument, then it’s the name of the server to connect to.

--socketid {id}

GTK GUI only: Use the GtkPlug mechanism to run gvim in another window.

--version Print version information and exit.

## sed

### ****简介****

sed是非交互式的编辑器。它不会修改文件，除非使用shell重定向来保存结果。默认情况下，所有的输出行都被打印到屏幕上。

sed编辑器逐行处理文件（或输入），并将结果发送到屏幕。具体过程如下：首先sed把当前正在处理的行保存在一个临时缓存区中（也称为模式空间），然后处理临时缓冲区中的行，完成后把该行发送到屏幕上。sed每处理完一行就将其从临时缓冲区删除，然后将下一行读入，进行处理和显示。处理完输入文件的最后一行后，sed便结束运行。sed把每一行都存在临时缓冲区中，对这个副本进行编辑，所以不会修改原文件。

### ****地址****

地址用于决定对哪些行进行编辑。地址的形式可以是数字、正则表达式、或二者的结合。如果没有指定地址，sed将处理输入文件的所有行。

地址是一个数字，则表示行号； “$"符号，则表示最后一行。例如：

|  |
| --- |
| sed -n'3p' datafile 只打印第三行 |

只显示指定行范围的文件内容，例如：

**# 只查看文件的第100行到第200行  
sed -n '100,200p' mysql\_slow\_query.log**

地址是逗号分隔的，那么需要处理的地址是这两行之间的范围（包括这两行在内）。范围可以用数字、正则表达式、或二者的组合表示。例如：

|  |
| --- |
| sed '2,5d' datafile #删除第二到第五行  sed '/My/,/You/d' datafile #删除包含"My"的行到包含"You"的行之间的行  sed '/My/,10d' datafile #删除包含"My"的行到第十行的内容 |

### ****命令与选项****

sed命令告诉sed如何处理**由地址指定的各输入行，如果没有指定地址则处理所有的输入行**。

**3.1 sed命令**

|  |  |
| --- | --- |
| 命令 | 功能 |
| **a\** | 在**当前行后**添加一行或多行。**多行时除最后一行外，每行末尾需用“\”续行** |
| c\ | 用此符号后的新文本替换当前行中的文本。多行时除最后一行外，每行末尾需用"\"续行 |
| i\ | 在当前行之前插入文本。多行时除最后一行外，每行末尾需用"\"续行 |
| **d** | **删除行** |
| h | 把模式空间里的内容复制到暂存缓冲区 |
| H | 把模式空间里的内容追加到暂存缓冲区 |
| g | 把暂存缓冲区里的内容复制到模式空间，覆盖原有的内容 |
| G | 把暂存缓冲区的内容追加到模式空间里，追加在原有内容的后面 |
| l | 列出非打印字符 |
| p | 打印行 |
| n | 读入下一输入行，并从下一条命令而不是第一条命令开始对其的处理 |
| q | 结束或退出sed |
| r | 从文件中读取输入行 |
| ! | 对所选行以外的所有行应用命令 |
| s | **用一个字符串替换另一个** |
| g | **在行内进行全局替换** |
| w | 将所选的行写入文件 |
| x | 交换暂存缓冲区与模式空间的内容 |
| y | 将字符替换为另一字符（不能对正则表达式使用y命令） |

**3.2 sed选项**

|  |  |
| --- | --- |
| 选项 | 功能 |
| -e | 进行多项编辑，即对输入行应用多条sed命令时使用 |
| **-n** | **取消 默认的输出** |
| -f | 指定sed脚本的文件名 |
| **-i** | 决定是不是将sed的操作作用在源文件上 |
| **-r** | [想少用\  那你就用-r](http://asmboy001.blog.51cto.com/340398/182290/" \t "http://blog.csdn.net/wh_19910525/article/details/_blank) |

### ****退出状态****

sed不像grep一样，不管是否找到指定的模式，它的退出状态都是0。只有当命令存在语法错误时，sed的退出状态才不是0。

### ****正则表达式元字符****

 与grep一样，sed也支持特殊元字符，来进行模式查找、替换。不同的是，sed使用的正则表达式是括在斜杠线"/"之间的模式。

如果要把正则表达式分隔符"/"改为另一个字符，比如o，只要在这个字符前加一个反斜线，在字符后跟上正则表达式，再跟上这个字符即可。例如：sed -n '\o^Myop' datafile

|  |  |  |
| --- | --- | --- |
| 元字符 | 功能 | 示例 |
| ^ | 行首定位符 | **/^my/  匹配所有以my开头的行** |
| $ | 行尾定位符 | **/my$/  匹配所有以my结尾的行** |
| . | 匹配除换行符以外的单个字符 | /m..y/  匹配包含字母m，后跟两个任意字符，再跟字母y的行 |
| \* | 匹配零个或多个前导字符 | **/my\*/  匹配包含字母m,后跟零个或多个y字母的行** |
| [] | 匹配指定字符组内的任一字符 | /[Mm]y/  匹配包含My或my的行 |
| [^] | 匹配不在指定字符组内的任一字符 | /[^Mm]y/  匹配包含y，但y之前的那个字符不是M或m的行 |
| .. | 保存已匹配的字符 | 1,20s/youself/\1r/  标记元字符之间的模式，并将其保存为标签1，之后可以使用\1来引用它。最多可以定义9个标签，从左边开始编号，最左边的是第一个。此例中，对第1到第20行进行处理，you被保存为标签1，如果发现youself，则替换为your。 |
| & | 保存查找串以便在替换串中引用 | s/my/\*\*&\*\*/  符号&代表查找串。my将被替换为\*\*my\*\* |
| \< | 词首定位符 | /\<my/  匹配包含**以my开头**的单词的行 |
| \> | 词尾定位符 | /my\>/  匹配包含**以my结尾**的单词的行 |
| x\{m\} | 连续m个x | /9\{5\}/ 匹配包含连续5个9的行 |
| x\{m,\} | 至少m个x | /9\{5,\}/  匹配包含至少连续5个9的行 |
| x\{m,n\} | 至少m个，但不超过n个x | /9\{5,7\}/  匹配包含连续5到7个9的行 |

### ****范例****

在DOS文件格式中使用CR/LF换行，在Unix下仅使用LF换行，sed替换命令如下：

         DOS转UNIX：$ sed ‘s/.$//’dosfile.txt > unixfile.txt

         UNIX转DOS：$ sed ’s/$/\r/’unixfile.txt > dosfile.txt

#### ****p命令****

命令p用于显示模式空间的内容。默认情况下，sed把输入行打印在屏幕上，选项-n用于取消默认的打印操作。当选项-n和命令p同时出现时,sed可打印选定的内容。

|  |
| --- |
| sed '/my/p' datafile  #默认情况下，sed把所有输入行都打印在标准输出上。如果某行匹配模式my，p命令将把该行另外打印一遍。  **sed -n '/my/p' datafile**  **#选项-n取消sed默认的打印，p命令把匹配模式my的行打印一遍。**  定行的范围：逗号  $ sed -n '/test/,/check/p' example  所有在模板test和check所确定的范围内的行都被打印。  $ sed -n '5,/^test/p' example  打印从第五行开始到第一个包含以test开始的行之间的所有行。  $ sed '/test/,/check/s/$/sed test/' example-----对于模板test和west之间的行，每行的末尾用字符串sed test替换。 |

#### ****d命令****

命令d用于删除输入行。sed先将输入行从文件复制到模式空间里，然后对该行执行sed命令，最后将模式空间里的内容显示在屏幕上。如果发出的是命令d，当前模式空间里的输入行会被删除，不被显示。

|  |
| --- |
| 删除：d命令  $ sed '2d' example-----删除example文件的第二行。  $ sed '2,$d' example-----删除example文件的第二行到末尾所有行。  $ sed '$d' example-----删除example文件的最后一行，其余的都被显示。  $ sed '/test/'d example-----删除example文件所有包含test的行。  **sed'/my/d'** **datafile**  **#删除包含my的行，其余的都被显示，注意：d可以在引号里，也可以在引号外。**  **sed'/my$/d'** **datafile**  **#删除以my结尾 的行，**  **sed'/^my/d'** **datafile**  **#删除以my开头 的行，** |

#### ****s命令****

|  |
| --- |
| **替换：s命令**  **$ sed 's/test/mytest/g' example**  #在example 所有行范围内把test替换为mytest。如果没有g标记，则只有每行第一个匹配的test被替换成mytest。  **$ sed -n 's/^test/mytest/p' example**  #(-n)选项和p标志一起使用表示只打印那些发生替换的行。也就是说，如果某一行开头的test被替换成mytest，就打印它。  **$ sed 's/^192.168.0.1/&localhost/' example**  # &符号表示替换字符串中被找到的部份。所有以192.168.0.1开头的行都会被替换成它自已加localhost，变成192.168.0.1localhost。  **$ sed 's#10#100#g' example**  #不论什么字符，紧跟着s命令的都被认为是新的分隔符，所以，“#”在这里是分隔符，代替了默认的“/”分隔符。表示把所有10替换成100。  **$ sed-n'1,20s/My$/You/gp' datafile** #取消默认输出，处理1到20行里匹配以My结尾的行，把行内所有的My替换为You，并打印到屏幕上。 |

#### ****e选项****

-e是编辑命令，用于sed执行多个编辑任务的情况下。在下一行开始编辑前，所有的编辑动作将应用到模式缓冲区中的行上。

|  |
| --- |
| sed -e'1,3d' -e's/My/Your/g' datafile  #**选项-e用于进行多重编辑**。第一重编辑删除第1-3行。第二重编辑将出现的所有My替换为Your。**因为是逐行进行这两项编辑（即这两个命令都在模式空间的当前行上执行），所以编辑命令的顺序会影响结果。** |

#### ****r命令****

r命令是读命令。sed使用该命令将一个文本文件中的内容加到当前文件的特定位置上。

|  |
| --- |
| sed'/My/r introduce.txt' datafile #如果在文件datafile的某一行匹配到模式My，就在该行后读入文件introduce.txt的内容。如果出现My的行不止一行，则在出现My的各行后都读入introduce.txt文件的内容。 |

#### ****w命令****

|  |
| --- |
| sed-n'/hrwang/w me.txt' datafile |

#### ****a\ 命令****

a\ 命令是追加命令，追加将添加新文本到文件中当前行（即读入模式缓冲区中的行）的后面。所追加的文本行位于sed命令的下方另起一行。如果要追加的内容超过一行，则每一行都必须以反斜线结束，最后一行除外。最后一行将以引号和文件名结束。

|  |
| --- |
| sed'/**^hrwang**/**a\ >**hrwang and mjfan are husband**\** and wife' datafile  #如果在datafile文件中发现匹配**以hrwang开头**的**行**，则在该行下面追加 如下 两行  hrwang and mjfan are husband  and wife |

#### ****i\ 命令****

i\ 命令是在当前行的前面插入新的文本。

$ sed '/test/i\

new line' example

如果test被匹配，则把反斜杠后面的文本newline 插入到匹配行的前面。

#### ****c\ 命令****

sed使用该命令将已有文本修改成新的文本。

#### ****n命令****

sed使用该命令获取输入文件的下一行，并将其读入到模式缓冲区中，任何sed命令都将应用到匹配行紧接着的下一行上。

|  |
| --- |
| sed'/hrwang/{n;s/My/Your/;}' datafile |

注：如果需要使用 多条命令，或者需要 在某个地址范围内嵌套地址，就必须用花括号将命令括起来，每行只写一条命令，或者用分号分割同一行中的多条命令。

#### ****y命令****

该命令与UNIX/Linux中的tr命令类似，字符按照一对一的方式从左到右进行转换。例如，y/abc/ABC/将把所有小写的a转换成A，小写的b转换成B，小写的c转换成C。

|  |
| --- |
| **sed'1,20y/hrwang12/HRWANG^$/'** **datafile #将1到20行内，所有的小写hrwang转换成大写，将1转换成^,将2转换成$。** #正则表达式元字符对y命令不起作用。与s命令的分隔符一样，斜线可以被替换成其它的字符。 |

#### ****q命令****

q命令将导致sed程序退出，不再进行其它的处理。

|  |
| --- |
| sed'/hrwang/{s/hrwang/HRWANG/;q;}' datafile |

#### ****h命令和g命令****

|  |
| --- |
| #cat datafile  My name is hrwang.  Your name is mjfan.  hrwang is mjfan's husband.  mjfan is hrwang's wife.    sed-e'/hrwang/h'-e'$G' datafile  sed -e '/hrwang/H' -e '$G' datafile  #通过上面两条命令，你会发现h会把原来暂存缓冲区的内容清除，只保存最近一次执行h时保存进去的模式空间的内容。而H命令则把每次匹配hrwnag的行都追加保存在暂存缓冲区。  sed -e '/hrwang/H' -e '$g' datafile  sed -e '/hrwang/H' -e '$G' datafile  #通过上面两条命令，你会发现g把暂存缓冲区中的内容替换掉了模式空间中当前行的内容，此处即替换了最后一行。而G命令则把暂存缓冲区的内容追加到了模式空间的当前行后。此处即追加到了末尾。 |

### ****sed脚本****

sed脚本就是写在文件中的一列sed命令。脚本中，要求命令的末尾不能有任何多余的空格或文本。如果在一行中有多个命令，要用分号分隔。执行脚本时，sed先将输入文件中第一行复制到模式缓冲区，然后对其执行脚本中所有的命令。每一行处理完毕后，sed再复制文件中下一行到模式缓冲区，对其执行脚本中所有命令。使用sed脚本时，不再用引号来确保sed命令不被shell解释。

==================================

下一个：n命令

$ sed '/test/{ n; s/aa/bb/; }' example-----如果test被匹配，则移动到匹配行的下一行，替换这一行的aa，变为bb，并打印该行，然后继续。

变形：y命令

$ sed '1,10y/abcde/ABCDE/' example-----把1--10行内所有abcde转变为大写，注意，正则表达式元字符不能使用这个命令。

退出：q命令

$ sed '10q' example-----打印完第10行后，退出sed。

保持和获取：h命令和G命令

$ sed -e '/test/h' -e '$G example-----在sed处理文件的时候，每一行都被保存在一个叫模式空间的临时缓冲区中，除非行被删除或者输出被取消，否则所有被处理的行都将打印在屏幕上。接着模式空间被清空，并存入新的一行等待处理。在这个例子里，匹配test的行被找到后，将存入模式空间，h命令将其复制并存入一个称为保持缓存区的特殊缓冲区内。第二条语句的意思是，当到达最后一行后，G命令取出保持缓冲区的行，然后把它放回模式空间中，且追加到现在已经存在于模式空间中的行的末尾。在这个例子中就是追加到最后一行。简单来说，任何包含test的行都被复制并追加到该文件的末尾。

保持和互换：h命令和x命令

$ sed -e '/test/h' -e '/check/x' example -----互换模式空间和保持缓冲区的内容。也就是把包含test与check的行互换。

### $ man sed

SED(1) User Commands SED(1)

NAME

sed - stream editor for filtering and transforming text

SYNOPSIS

sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and is consequently more efficient. But it is sed’s ability to filter text in a pipeline which particularly distin-guishes it from other types of editors.

-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; hard links will still be broken.

-i[SUFFIX], --in-place[=SUFFIX]

edit files in place (makes backup if extension supplied). The default operation mode is to break symbolic and hard links.

This can be changed with --follow-symlinks and --copy.

-c, --copy

use copy instead of rename when shuffling files in -i mode. While this will avoid breaking links (symbolic or hard), the resulting editing operation is not atomic. This is rarely the desired mode; --follow-symlinks is usually enough, and it is both faster and more secure.

-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

--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.

COMMAND SYNOPSIS

This is just a brief synopsis of sed commands to serve as a reminder to those who already know sed; other documentation (such as the

texinfo document) must be consulted for fuller descriptions.

Zero-address ‘‘commands’’

: label

Label for b and t commands.

#comment

The comment extends until the next newline (or the end of a -e script fragment).

} The closing bracket of a { } block.

Zero- or One- address commands

= Print the current line number.

a \

text Append text, which has each embedded newline preceded by a backslash.

i \

text Insert text, which has each embedded newline preceded by a backslash.

q [exit-code]

Immediately quit the sed script without processing any more input, except that if auto-print is not disabled the current pat-

tern space will be printed. The exit code argument is a GNU extension.

Q [exit-code]

Immediately quit the sed script without processing any more input. This is a GNU extension.

r filename

Append text read from filename.

R filename

Append a line read from filename. Each invocation of the command reads a line from the file. This is a GNU extension.

Commands which accept address ranges

{ Begin a block of commands (end with a }).

b label

Branch to label; if label is omitted, branch to end of script.

t label

If a s/// has done a successful substitution since the last input line was read and since the last t or T command, then branch

to label; if label is omitted, branch to end of script.

T label

If no s/// has done a successful substitution since the last input line was read and since the last t or T command, then

branch to label; if label is omitted, branch to end of script. This is a GNU extension.

c \

text Replace the selected lines with text, which has each embedded newline preceded by a backslash.

d Delete pattern space. Start next cycle.

D Delete up to the first embedded newline in the pattern space. Start next cycle, but skip reading from the input if there is

still data in the pattern space.

h H Copy/append pattern space to hold space.

g G Copy/append hold space to pattern space.

x Exchange the contents of the hold and pattern spaces.

l List out the current line in a ‘‘visually unambiguous’’ form.

l width

List out the current line in a ‘‘visually unambiguous’’ form, breaking it at width characters. This is a GNU extension.

n N Read/append the next line of input into the pattern space.

p Print the current pattern space.

P Print up to the first embedded newline of the current pattern space.

s/regexp/replacement/

Attempt to match regexp against the pattern space. If successful, replace that portion matched with replacement. The

replacement may contain the special character & to refer to that portion of the pattern space which matched, and the special

escapes \1 through \9 to refer to the corresponding matching sub-expressions in the regexp.

w filename

Write the current pattern space to filename.

W filename

Write the first line of the current pattern space to filename. This is a GNU extension.

y/source/dest/

Transliterate the characters in the pattern space which appear in source to the corresponding character in dest.

**Addresses**

Sed commands can be given with no addresses, in which case the command will be executed for all input lines; with one address, in which case the command will only be executed for input lines which match that address; or with two addresses, in which case the command will be executed for all input lines which match the inclusive range of lines starting from the first address and continuing to the second address. Three things to note about address ranges: the syntax is addr1,addr2 (i.e., the addresses are separated by a comma); the line which addr1 matched will always be accepted, even if addr2 selects an earlier line; and if addr2 is a regexp, it will not be tested against the line that addr1 matched.

After the address (or address-range), and before the command, a ! may be inserted, which specifies that the command shall only be executed if the address (or address-range) does not match.

The following address types are supported:

number Match only the specified line number.

first~step

Match every step’th line starting with line first. For example, ‘‘sed -n 1~2p’’ will print all the odd-numbered lines in the input stream, and the address 2~5 will match every fifth line, starting with the second. first can be zero; in this case, sed operates as if it were equal to step. (This is an extension.)

**$**  Match the last line.

/regexp/

Match lines matching the regular expression regexp.

\cregexpc

Match lines matching the regular expression regexp. The c may be any character.

GNU sed also supports some special 2-address forms:

0,addr2

Start out in "matched first address" state, until addr2 is found. This is similar to 1,addr2, except that if addr2 matches

the very first line of input the 0,addr2 form will be at the end of its range, whereas the 1,addr2 form will still be at the

beginning of its range. This works only when addr2 is a regular expression.

addr1,+N

Will match addr1 and the N lines following addr1.

addr1,~N

Will match addr1 and the lines following addr1 until the next line whose input line number is a multiple of N.

REGULAR EXPRESSIONS

POSIX.2 BREs should be supported, but they aren’t completely because of performance problems. The \n sequence in a regular expres-

sion matches the newline character, and similarly for \a, \t, and other sequences.

SEE ALSO

awk(1), ed(1), grep(1), tr(1), perlre(1), sed.info, any of various books on sed, the sed FAQ (http://sed.sf.net/grabbag/tutori-

als/sedfaq.txt), http://sed.sf.net/grabbag/.

The full documentation for sed is maintained as a Texinfo manual. If the info and sed programs are properly installed at your site,

the command

info sed

should give you access to the complete manual.

sed version 4.2.1 March 2012 SED(1)

## tr

### 简介

tr是translate的简写，亦即翻译，但是遗憾的是，它不能翻译句子，只能翻译单个字符。

****1 tr的工作原理是什么?****

先记住一点，tr命令不接受指定的文件参数，而只是对标准输入进行翻译。好了，记住这点后，我们继续。

tr的命令格式是tr SET1 SET2，凡是在SET1中的字符，都会被替换为SET2中相应位置上的字符，简单吧!

****2 举个例子吧!****

tr的例子其实都大同小异，看一个最简单的例子：

我们有一个包含了四个人身高的数据文件，名字叫height.txt，它的内容是这样的：

1. $ cat height.txt
2. 1.79 1.83
3. 1.65 1.59

我们想搞一个恶作剧，将所有人的身高从1米级别都提高到2米级别，呵呵，一个tr就可以搞定。

1. $ tr 1 2 < height.txt
2. 2.79 2.83
3. 2.65 2.59

希望你没有忘记，tr只处理标准输入，所以我们需要将height.txt通过重定向指到tr的标准输入才可以。

****3 我们可以用tr来修改文件中的间隔符么？****

Good idea!

当使用cut的时候，通常会被间隔符问题所困扰，我们这个时候可以考虑到tr这个命令。

还是以height.txt文件为例，其中的间隔符是空格，我们把它修改为制表符吧！

1. $ tr ' ' '\11' < height.txt
2. 1.79 1.83
3. 1.65 1.59

这里有一个小知识点，像制表符、换行符这些字符不好表示，我们可以考虑使用ASCII的八进制形式来表示，制表符的八进制形式是11，回车是15，换行是12。而在tr命令中，可以使用\nnn形式表示八进制形式的字符。如果你实在记不住这些编号，那么用\t表示制表符你总该可以记住吧！（\n代表新行，\r代表回车）

这下，你应该可以理解上面那条命令的作用了吧。如果你仍然怀疑，那么，看看下面这条命令，你就该心服口服了：

1. $ tr ' ' '\11' < height.txt |sed -n l
2. 1.79\t1.83$
3. 1.65\t1.59$

看！空格的的确确被替换成了制表符喽！

****4 使用tr能把文章中的小写都转换成大写么？****

这个正是tr所擅长的地方。

加入我们拥有一个文件叫做word.txt，里面包含的内容是AbcdE。

那么最简单的替换大小写的方法是：

1. $ cat word.txt
2. AbcdE
3. $ tr 'a-z' 'A-Z' < word.txt
4. ABCDE

****5 [CHAR\*]怎么用？****

这是 SET2 专用的设定，功能是重复指定的字符到与 SET1 相同长度为止

例子：

1. $ cat number.txt
2. 1234567890
3. $ tr '1-5' '[A\*]' < number.txt
4. AAAAA67890

****6 [CHAR\*REPEAT]怎么用？****

这也是SET2专用的设定，功能是将CHAR重复REPEAT次数。其中REPEAT次数可以用八进制数表示，但记得要以0开头表示八进制数。

例子：

1. $ tr '1-9' '[A\*5]BCDE' < number.txt
2. AAAAABCDE0

****7 在tr中还有哪些表示集合的符号呢?****

[:alnum:] ：所有字母字符与数字  
[:alpha:] ：所有字母字符  
[:blank:] ：所有水平空格  
[:cntrl:] ：所有控制字符  
[:digit:] ：所有数字  
[:graph:] ：所有可打印的字符(不包含空格符)  
[:lower:] ：所有小写字母  
[:print:] ：所有可打印的字符(包含空格符)  
[:punct:] ：所有标点字符  
[:space:] ：所有水平与垂直空格符  
[:upper:] ：所有大写字母  
[:xdigit:] ：所有 16 进位制的数字

例子：

将所有的数字都转换为字符x。

1. $ tr [:alnum:] '[x\*]' < number.txt
2. xxxxxxxxxx

****8 tr里面包含SET1和SET2，那如果出现两个集合的大小不同的情况，tr如何处理呢？****

这个问题，最好的解决办法就是做实验啊。

第一种情况是SET1>SET2：

1. $ tr 1-5 AB < number.txt
2. ABBBB67890

结论一下子就出来了，SET1中多出来的字符都会和SET2中最后一个字符相对应。

第二种情况SET1<SET2：

1. $ tr 1-3 ABCDE < number.txt
2. ABC4567890

很明了，SET2中多余的部分将被抛弃。

****9 讲一讲tr命令的-s选项吧！****

这个-s选项，是专门针对SET1起作用的，意思是如果发现有连续的SET1里的字符，就把它们缩减为1个。

一个很经典的应用就是把不规律的空格缩减为一个空格：

1. $ cat spaces.txt
2. How     are                you?
3. Fine          Thank   you
4. $ tr -s ' ' ' ' < spaces.txt
5. How are you?
6. Fine Thank you

效果很明显，用户很满意。恩！

****10 -d选项咋用？****

-d选项是用来删除字符用的。格式是这样的：tr -d charset

1. $ tr -d ' ' < spaces.txt
2. How  are  you?
3. FineThankyou

看，该有的空格都没了…这就是-d的作用，把空格都删除了!

如果你想把文章中的数字都删除，就tr -d [0-9] < filename就可以了。

****11 -c选项的神奇****

这个参数能删除指定字符外的补集。

1. root@ubuntu:/test# cat file
2. a b c 1 2 3 a b c
3. root@ubuntu:/test# tr -c 'a-z \n' 'x' <file
4. a b c x x x a b c

这把把除小写字母，空格，换行符以外的字符替换成x。

注：都可以用sed 来完成。

eval </dev/urandom tr -dc A-Za-z0-9|(head -c $1 > /dev/null 2>&1||head -c 10)

随机生成10个字符串

### tr命令详解

通过使用 tr，您可以非常容易地实现 sed 的许多最基本功能。您可以将 tr 看作为 sed 的（极其）简化的变体：它可以用一个字符来替换另一个字符，或者可以完全除去一些字符。您也可以用它来除去重复字符。这就是所有 tr 所能够做的。   
    tr用来从标准输入中通过替换或删除操作进行字符转换。tr主要用于删除文件中控制字符或进行字符转换。使用tr时要转换两个字符串：字符串1用于查询，字符串2用于处理各种转换。tr刚执行时，字符串1中的字符被映射到字符串2中的字符，然后转换操作开始。  
    带有最常用选项的tr命令格式为：  
tr -c -d -s ["string1\_to\_translate\_from"] ["string2\_to\_translate\_to"] < input-file  
这里：  
-c 用字符串1中字符集的补集替换此字符集，要求字符集为ASCII。  
-d 删除字符串1中所有输入字符。  
-s 删除所有重复出现字符序列，只保留第一个；即将重复出现字符串压缩为一个字符串。  
input-file是转换文件名。虽然可以使用其他格式输入，但这种格式最常用。

2、字符范围  
指定字符串1或字符串2的内容时，只能使用单字符或字符串范围或列表。  
[a-z] a-z内的字符组成的字符串。  
[A-Z] A-Z内的字符组成的字符串。  
[0-9] 数字串。  
\octal 一个三位的八进制数，对应有效的ASCII字符。  
[O\*n] 表示字符O重复出现指定次数n。因此[O\*2]匹配OO的字符串。  
tr中特定控制字符的不同表达方式  
速记符含义八进制方式  
\a Ctrl-G  铃声\007  
\b Ctrl-H  退格符\010  
\f Ctrl-L  走行换页\014  
\n Ctrl-J  新行\012  
\r Ctrl-M  回车\015  
\t Ctrl-I  tab键\011  
\v Ctrl-X  \030

实例：

、将文件file中出现的"abc"替换为"xyz"

# cat file | tr "abc" "xyz" > new\_file

【注意】这里，凡是在file中出现的"a"字母，都替换成"x"字母，"b"字母替换为"y"字母，"c"字母替换为"z"字母。而不是将字符串"abc"替换为字符串"xyz"。

2、使用tr命令“统一”字母大小写

（小写 --> 大写）  
# cat file | tr [a-z] [A-Z] > new\_file

（大写 --> 小写）  
# cat file | tr [A-Z] [a-z] > new\_file

3、把文件中的数字0-9替换为a-j

# cat file | tr [0-9] [a-j] > new\_file

4、删除文件file中出现的"Snail"字符

# cat file | tr -d "Snail" > new\_file

【注意】这里，凡是在file文件中出现的'S','n','a','i','l'字符都会被删除！而不是紧紧删除出现的"Snail”字符串。

5、删除文件file中出现的换行'\n'、制表'\t'字符

# cat file | tr -d "\n\t" > new\_file

不可见字符都得用转义字符来表示的，这个都是统一的。

6、删除“连续着的”重复字母，只保留第一个

# cat file | tr -s [a-zA-Z] > new\_file

7、删除空行

# cat file | tr -s "\n" > new\_file

8、删除Windows文件“造成”的'^M'字符

# cat file | tr -d "\r" > new\_file  
或者  
# cat file | tr -s "\r" "\n" > new\_file

【注意】这里-s后面是两个参数"\r"和"\n"，用后者替换前者

9、用空格符\040替换制表符\011

# cat file | tr -s "\011" "\040" > new\_file

10、把路径变量中的冒号":"，替换成换行符"\n"

# echo $PATH | tr -s ":" "\n"

### # man tr

TR(1) User Commands TR(1)

NAME

tr - translate or delete characters

SYNOPSIS

tr [OPTION]... SET1 [SET2]

DESCRIPTION

Translate, squeeze, and/or delete characters from standard input, writing to standard output.

-c, -C, --complement

use the complement of SET1

-d, --delete

delete characters in SET1, do not translate

-s, --squeeze-repeats

replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

-t, --truncate-set1

first truncate SET1 to length of SET2

--help display this help and exit

--version

output version information and exit

SETs are specified as strings of characters. Most represent themselves. Interpreted sequences are:

\NNN character with octal value NNN (1 to 3 octal digits)

\\ backslash

\a audible BEL

\b backspace

\f form feed

\n new line

\r return

\t horizontal tab

\v vertical tab

CHAR1-CHAR2

all characters from CHAR1 to CHAR2 in ascending order

[CHAR\*]

in SET2, copies of CHAR until length of SET1

[CHAR\*REPEAT]

REPEAT copies of CHAR, REPEAT octal if starting with 0

[:alnum:]

all letters and digits

[:alpha:]

all letters

[:blank:]

all horizontal whitespace

[:cntrl:]

all control characters

[:digit:]

all digits

[:graph:]

all printable characters, not including space

[:lower:]

all lower case letters

[:print:]

all printable characters, including space

[:punct:]

all punctuation characters

[:space:]

all horizontal or vertical whitespace

[:upper:]

all upper case letters

[:xdigit:]

all hexadecimal digits

[=CHAR=]

all characters which are equivalent to CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear. -t may be used only when translating. SET2 is extended to

length of SET1 by repeating its last character as necessary. Excess characters of SET2 are ignored. Only [:lower:] and [:upper:]

are guaranteed to expand in ascending order; used in SET2 while translating, they may only be used in pairs to specify case conver-

sion. -s uses SET1 if not translating nor deleting; else squeezing uses SET2 and occurs after translation or deletion.

AUTHOR

Written by Jim Meyering.

REPORTING BUGS

Report tr bugs to bug-coreutils@gnu.org

GNU coreutils home page: <http://www.gnu.org/software/coreutils/>

General help using GNU software: <http://www.gnu.org/gethelp/>

Report tr translation bugs to <http://translationproject.org/team/>

COPYRIGHT

Copyright © 2010 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

The full documentation for tr is maintained as a Texinfo manual. If the info and tr programs are properly installed at your site,

the command

info coreutils 'tr invocation'

should give you access to the complete manual.

GNU coreutils 8.4 April 2012 TR(1)

# 设备管理

## mount

使用：mount [-r] 设备 目录

说明：将设备安装到目录下

-r 以只读方式安装

举例：

将CDROM（路径/dev/dsk/c0t2d0，可以使用命令ioscan -fnC disk检查）安装到/cdrom目录下：mount /dev/dsk/c0t2d0   /cdrom

## umount

使用：umount 设备

说明：将安装的文件系统卸下

举例：umount /cdrom 将安装在/cdrom下的文件系统卸载。

## gunzip

Linux压缩保留源文件的方法：

gzip –c filename > filename.gz

Linux解压缩保留源文件的方法：

gunzip –c filename.gz > filename

gunzip的用法

  1.作用

gunzip命令作用是解压文件，使用权限是所有用户。

2.格式

gunzip [-acfhlLnNqrtvV][-s ][文件...]

或者gunzip [-acfhlLnNqrtvV][-s ][目录]

3.主要参数

-a或--ascii：使用ASCII文字模式。

-c或--stdout或--to-stdout：把解压后的文件输出到标准输出设备。

-f或-force：强行解开压缩文件，不理会文件名称或硬连接是否存在，以及该文件是否为符号连接。

-h或--help：在线帮助。

-l或--list：列出压缩文件的相关信息。

-L或--license：显示版本与版权信息。

-n或--no-name：解压缩时，若压缩文件内含有原来的文件名称及时间戳记，则将其忽略不予处理。

-N或--name：解压缩时，若压缩文件内含有原来的文件名称及时间戳记，则将其回存到解开的文件上。

-q或--quiet：不显示警告信息。

-r或--recursive：递归处理，将指定目录下的所有文件及子目录一并处理。

-S或--suffix：更改压缩字尾字符串。

-t或--test：测试压缩文件是否正确无误。

-v或--verbose：显示指令执行过程。

-V或--version：显示版本信息。

4.说明

gunzip是个使用广泛的解压缩程序，它用于解开被gzip压缩过的文件，这些压缩文件预设最后的扩展名为“.gz”。事实上，gunzip就是gzip的硬连接，因此不论是压缩或解压缩，都可通过gzip指令单独完成。

## tar

使用：tar -c [vwfbL] [设备] [块] 文件...

tar -r[vwfbL] [设备] [块] 文件...

tar -t[vfL]   [设备] [文件...]

tar -u[vwfbL] [设备] [块] 文件...

tar -x[lmovwfL] [设备] [文件...]

说明：将多个文件归档到一个文件中，命令中各参数的意义为：

|  |  |
| --- | --- |
| x | 抽取文件 |
| t | 显示文件 |
| u | 附加方式归档，同时删除旧版文件 |
| c | 建立新档案文件 |
| v | 显示所处理的文件名 |
| f | 处理文件，f不带缺省为磁带 |

举例：

tar tvf file.tar 查看file.tar中有那些文件

tar tv       查看磁带中有那些文件

tar xvf file.tar 抽取file.tar中的文件

tar xv    抽取磁带中的文件

tar cvf file.tar log bin    创建新文件file.tar，包含log，bin两个文件或目录

tar cv   log bin   创建新的磁带文档，包含log，bin两个文件或目录

1.压缩命令：

　　命令格式：tar -zcvf 压缩文件名.tar.gz 被压缩文件名

可先切换到当前目录下。压缩文件名和被压缩文件名都可加入路径。

2.解压缩命令：

　　命令格式：tar -zxvf 压缩文件名.tar.gz

　　解压缩后的文件只能放在当前的目录。

tar -zxvf 压缩文件名.tar.gz -C /data/bb/

tar --help

用法: tar [选项...] [FILE]...

GNU ‘tar’

将许多文件一起保存至一个单独的磁带或磁盘归档，并能从归档中单独还原所需文件。

示例

tar -cf archive.tar foo bar # 从文件 foo 和 bar 创建归档文件

archive.tar。

tar -tvf archive.tar # 详细列举归档文件 archive.tar

中的所有文件。

tar -xf archive.tar # 展开归档文件 archive.tar

中的所有文件。

主操作模式:

-A, --catenate, --concatenate 追加 tar 文件至归档

-c, --create 创建一个新归档

-d, --diff, --compare 找出归档和文件系统的差异

--delete 从归档(非磁带！)中删除

-r, --append 追加文件至归档结尾

-t, --list 列出归档内容

--test-label 测试归档卷标并退出

-u, --update 仅追加比归档中副本更新的文件

-x, --extract, --get 从归档中解出文件

操作修饰符:

--check-device 当创建增量归档时检查设备号(默认)

-g, --listed-incremental=FILE 处理新式的 GNU 格式的增量备份

-G, --incremental 处理老式的 GNU 格式的增量备份

--ignore-failed-read

当遇上不可读文件时不要以非零值退出

--level=NUMBER 所创建的增量列表归档的输出级别

-n, --seek 归档可检索

--no-check-device 当创建增量归档时不要检查设备号

--no-seek 归档不可检索

--occurrence[=NUMBER] 仅处理归档中每个文件的第 NUMBER

个事件；仅当与以下子命令 --delete,

--diff, --extract 或是 --list

中的一个联合使用时，此选项才有效。而且不管文件列表是以命令行形式给出或是通过

-T 选项指定的；NUMBER 值默认为 1

--sparse-version=MAJOR[.MINOR]

设置所用的离散格式版本(隐含

--sparse)

-S, --sparse 高效处理离散文件

重写控制:

-k, --keep-old-files don't replace existing files when extracting,

treat them as errors

--keep-newer-files

不要替换比归档中副本更新的已存在的文件

--no-overwrite-dir 保留已存在目录的元数据

--overwrite 解压时重写存在的文件

--overwrite-dir

解压时重写已存在目录的元数据(默认)

--recursive-unlink 解压目录之前先清除目录层次

--remove-files 在添加文件至归档后删除它们

--skip-old-files don't replace existing files when extracting,

silently skip over them

-U, --unlink-first 在解压要重写的文件之前先删除它们

-W, --verify 在写入以后尝试校验归档

选择输出流:

--ignore-command-error 忽略子进程的退出代码

--no-ignore-command-error

将子进程的非零退出代码认为发生错误

-O, --to-stdout 解压文件至标准输出

--to-command=COMMAND

将解压的文件通过管道传送至另一个程序

操作文件属性:

--acls Save the ACLs to the archive

--atime-preserve[=METHOD]

在输出的文件上保留访问时间，要么通过在读取(默认

METHOD=‘replace’)后还原时间，要不就不要在第一次(METHOD=‘system’)设置时间

--delay-directory-restore

直到解压结束才设置修改时间和所解目录的权限

--group=名称 强制将 NAME

作为所添加的文件的组所有者

--mode=CHANGES 强制将所添加的文件(符号)更改为权限

CHANGES

--mtime=DATE-OR-FILE 从 DATE-OR-FILE 中为添加的文件设置 mtime

-m, --touch 不要解压文件的修改时间

--no-acls Don't extract the ACLs from the archive

--no-delay-directory-restore

取消 --delay-directory-restore 选项的效果

--no-same-owner

将文件解压为您所有(普通用户默认此项)

--no-same-permissions

从归档中解压权限时使用用户的掩码位(默认为普通用户服务)

--no-selinux Don't extract the SELinux context from the

archive

--no-xattrs Don't extract the user/root xattrs from the

archive

--numeric-owner 总是以数字代表用户/组的名称

--owner=名称 强制将 NAME

作为所添加的文件的所有者

-p, --preserve-permissions, --same-permissions

解压文件权限信息(默认只为超级用户服务)

--preserve 与 -p 和 -s 一样

--same-owner

尝试解压时保持所有者关系一致(超级用户默认此项)

-s, --preserve-order, --same-order

为解压至匹配归档排序名称

--selinux Save the SELinux context to the archive

--xattrs Save the user/root xattrs to the archive

设备选择和切换:

-f, --file=ARCHIVE 使用归档文件或 ARCHIVE 设备

--force-local

即使归档文件存在副本还是把它认为是本地归档

-F, --info-script=名称, --new-volume-script=名称

在每卷磁带最后运行脚本(隐含 -M)

-L, --tape-length=NUMBER 写入 NUMBER × 1024 字节后更换磁带

-M, --multi-volume 创建/列出/解压多卷归档文件

--rmt-command=COMMAND 使用指定的 rmt COMMAND 代替 rmt

--rsh-command=COMMAND 使用远程 COMMAND 代替 rsh

--volno-file=FILE 使用/更新 FILE 中的卷数

设备分块:

-b, --blocking-factor=BLOCKS 每个记录 BLOCKS x 512 字节

-B, --read-full-records 读取时重新分块(只对 4.2BSD 管道有效)

-i, --ignore-zeros 忽略归档中的零字节块(即文件结尾)

--record-size=NUMBER 每个记录的字节数 NUMBER，乘以 512

选择归档格式:

-H, --format=FORMAT 创建指定格式的归档

FORMAT 是以下格式中的一种:

gnu GNU tar 1.13.x 格式

oldgnu GNU 格式 as per tar <= 1.12

pax POSIX 1003.1-2001 (pax) 格式

posix 等同于 pax

ustar POSIX 1003.1-1988 (ustar) 格式

v7 old V7 tar 格式

--old-archive, --portability

等同于 --format=v7

--pax-option=关键字[[:]=值][,关键字[[:]=值]]...

控制 pax 关键字

--posix 等同于 --format=posix

-V, --label=TEXT 创建带有卷名 TEXT

的归档；在列出/解压时，使用 TEXT

作为卷名的模式串

压缩选项:

-a, --auto-compress 使用归档后缀名来决定压缩程序

-I, --use-compress-program=PROG

通过 PROG 过滤(必须是能接受 -d

选项的程序)

-j, --bzip2 通过 bzip2 过滤归档

-J, --xz 通过 xz 过滤归档

--lzip 通过 lzip 过滤归档

--lzma 通过 lzma 过滤归档

--lzop

--no-auto-compress 不使用归档后缀名来决定压缩程序

-z, --gzip, --gunzip, --ungzip 通过 gzip 过滤归档

-Z, --compress, --uncompress 通过 compress 过滤归档

本地文件选择:

--add-file=FILE 添加指定的 FILE 至归档(如果名字以 -

开始会很有用的)

--backup[=CONTROL] 在删除前备份，选择 CONTROL 版本

-C, --directory=DIR 改变至目录 DIR

--exclude=PATTERN 排除以 PATTERN 指定的文件

--exclude-backups 排除备份和锁文件

--exclude-caches 除标识文件本身外，排除包含

CACHEDIR.TAG 的目录中的内容

--exclude-caches-all 排除包含 CACHEDIR.TAG 的目录

--exclude-caches-under 排除包含 CACHEDIR.TAG 的目录中所有内容

--exclude-tag=FILE 除 FILE 自身外，排除包含 FILE

的目录中的内容

--exclude-tag-all=FILE 排除包含 FILE 的目录

--exclude-tag-under=FILE 排除包含 FILE 的目录中的所有内容

--exclude-vcs 排除版本控制系统目录

-h, --dereference

跟踪符号链接；将它们所指向的文件归档并输出

--hard-dereference

跟踪硬链接；将它们所指向的文件归档并输出

-K, --starting-file=MEMBER-NAME

从归档中的 MEMBER-NAME 成员处开始

--newer-mtime=DATE 当只有数据改变时比较数据和时间

--no-null 禁用上一次的效果 --null 选项

--no-recursion 避免目录中的自动降级

--no-unquote 不以 -T 读取的文件名作为引用结束

--null -T 读取以空终止的名字，-C 禁用

-N, --newer=DATE-OR-FILE, --after-date=DATE-OR-FILE

只保存比 DATE-OR-FILE 更新的文件

--one-file-system 创建归档时保存在本地文件系统中

-P, --absolute-names 不要从文件名中清除引导符‘/’

--recursion 目录递归(默认)

--suffix=STRING 在删除前备份，除非被环境变量

SIMPLE\_BACKUP\_SUFFIX

覆盖，否则覆盖常用后缀(‘’)

-T, --files-from=FILE 从 FILE

中获取文件名来解压或创建文件

--unquote 以 -T

读取的文件名作为引用结束(默认)

-X, --exclude-from=FILE 排除 FILE 中列出的模式串

文件名变换:

--strip-components=NUMBER 解压时从文件名中清除 NUMBER

个引导部分

--transform=EXPRESSION, --xform=EXPRESSION

使用 sed 代替 EXPRESSION

来进行文件名变换

文件名匹配选项(同时影响排除和包括模式串):

--anchored 模式串匹配文件名头部

--ignore-case 忽略大小写

--no-anchored 模式串匹配任意‘/’后字符(默认对

exclusion 有效)

--no-ignore-case 匹配大小写(默认)

--no-wildcards 逐字匹配字符串

--no-wildcards-match-slash 通配符不匹配‘/’

--wildcards use wildcards (default)

--wildcards-match-slash

通配符匹配‘/’(默认对排除操作有效)

提示性输出:

--checkpoint[=NUMBER] 每隔 NUMBER

个记录显示进度信息(默认为 10 个)

--checkpoint-action=ACTION 在每个检查点上执行 ACTION

--index-file=FILE 将详细输出发送至 FILE

-l, --check-links

只要不是所有链接都被输出就打印信息

--no-quote-chars=STRING 禁用来自 STRING 的字符引用

--quote-chars=STRING 来自 STRING 的额外的引用字符

--quoting-style=STYLE 设置名称引用风格；有效的 STYLE

值请参阅以下说明

-R, --block-number 每个信息都显示归档内的块数

--show-defaults 显示 tar 默认选项

--show-omitted-dirs

列表或解压时，列出每个不匹配查找标准的目录

--show-transformed-names, --show-stored-names

显示变换后的文件名或归档名

--totals[=SIGNAL] 处理归档后打印出总字节数；当此

SIGNAL 被触发时带参数 -

打印总字节数；允许的信号为:

SIGHUP，SIGQUIT，SIGINT，SIGUSR1 和

SIGUSR2；同时也接受不带 SIG

前缀的信号名称

--utc 以 UTC 格式打印文件修改信息

-v, --verbose 详细地列出处理的文件

--warning=KEYWORD 警告控制:

-w, --interactive, --confirmation

每次操作都要求确认

兼容性选项:

-o 创建归档时，相当于

--old-archive；展开归档时，相当于

--no-same-owner

其它选项:

-?, --help 显示此帮助列表

--restrict 禁用某些潜在的有危险的选项

--usage 显示简短的用法说明

--version 打印程序版本

长选项和相应短选项具有相同的强制参数或可选参数。

除非以 --suffix 或 SIMPLE\_BACKUP\_SUFFIX

设置备份后缀，否则备份后缀就是“~”。

可以用 --backup 或 VERSION\_CONTROL 设置版本控制，可能的值为：

none, off 从不做备份

t, numbered 进行编号备份

nil, existing

如果编号备份存在则进行编号备份，否则进行简单备份

never, simple 总是使用简单备份

--quoting-style 选项的有效参数为:

literal

shell

shell-always

c

c-maybe

escape

locale

clocale

此 tar 默认为:

--format=gnu -f- -b20 --quoting-style=escape --rmt-command=/sbin/rmt

--rsh-command=/usr/bin/rsh

## du

使用：du [-sk] 目录/文件

说明：显示磁盘文件的使用情况，s选项表示只显示指定目录或文件的大小，k选项表示以KBYTE为单位。

举例：du -sk   log    显示log文件或目录的大小

du –sk Linux统计磁盘空间大小

## ioscan

使用：ioscan [选项]

说明：查看IO系统，如磁盘，网卡等

举例：

ioscan -fnC disk/lan   详细查看磁盘/网卡设备

ioscan     查看所有IO设备

## cpio

使用：cpio   [选项] 文件或目录

说明：同tar类似

举例：

cpio -icvdumB < IDS.CPI 从文件IDS.CPI中抽取文件

# 进程管理

## ps

使用：ps [ -efl] [ -t 终端表] [ -u 用户表] [ -g 组表]

说明：显示出有关进程的状态

|  |  |
| --- | --- |
| -e | 显示出现在正在运行的所有进程 |
| -f | 显示所有信息 |
| -l | 产生一个长列表 |
| -u | 显示指定用户进程 |

举例：

ps -lu scp 详细查看scp用户的各进程状态

ps –ef|grep ftsapp:查看fts进程

ps –ef|grep cdpmgr:查看C:D进程

## kill

使用：kill   -信号   进程号

说明：将信号传递给指定进程，信号共有几十个，但常用的是以下几个：

|  |  |  |
| --- | --- | --- |
| # define SIGNULL | 0 | /\* Null, Check access to pid \*/ |
| # define SIGINT | 2 | /\* Interrupt \*/ |
| # define SIGQUIT | 3 | /\* quit \*/ |
| # define SIGKILL | 9 | /\* kill (cannot be caught or ignored) \*/ |
| # define SIGBUS | 10 | /\* bus error \*/ |
| # define SIGSEGV | 11 | /\* Segmentation violation \*/ |
| # define SIGSYS | 12 | /\* bad argument to system call \*/ |
| # define SIGPIPE | 13 | /\* write on a pipe with no one to read it \*/ |
| # define SIGALRM | 14 | /\* alarm clock \*/ |
| # define SIGTERM | 15 | /\* Software termination signal from kill \*/ |
| # define SIGUSR1 | 16 | /\* user defined signal 1 \*/ |
| # define SIGUSR2 | 17 | /\* user defined signal 2 \*/ |
| # define SIGCHLD | 18 | /\* Child process terminated or stopped \*/ |
| # define SIGCLD | SIGCHLD | /\* death of a child \*/ |
| # define \_SIGSTOP | 24 | /\* Stop signal (cannot be caught or ignored) \*/ |

SIGBUG和SIGSEGV信号通常产生我们常见的core文件，SIGUSR1和SIGUSR2是留给用于自定义唯有的两个。

SIGNULL信号需要特殊注意，它主要用来测试一个进程是否存在，实际上并不会发出这么一个信号。相对于ps和grep组合来检查进程来说，使用SIGNULL来检查进程性能非常的好，不过它需要预先检查出进程ID。

举例： 给某进程ID为1068的进程发送一个SIGTERM信号终止其运行，

kill -s SIGTERM    1068

kill -s   15   1068

kill   -SIGTERM   1068

kill   -15     1068

## nohup

简单而有用的nohup命令在UNIX/LINUX中，普通进程用&符号放到后台运行，如果启动该程序的控制台logout，则该进程随即终止。 要实现守护进程，一种方法是按守护进程的规则去编程（本站有文章介绍过），比较麻烦；另一种方法是仍然用普通方法编程，然后用nohup命令启动程序：

nohup <程序名> &

则控制台logout后，进程仍然继续运行，起到守护进程的作用（虽然它不是严格意义上的守护进程）。

使用nohup命令后，原程序的的标准输出被自动改向到当前目录下的nohup.out文件，起到了log的作用，实现了完整的守护进程功能。

**For example：**

**如何远程启动WebLogic服务?**

用telnet远程控制服务器，远程启动WEBLOGIC服务，启动后关闭telnet，WebLogic服务也跟着停止，这是因为使用telnet启动的进程会随着telnet进程的关闭而关闭。所以我们可以使用一些UNIX下的命令来做到不关闭。

使用如下命令：

nohup startWeblogic.sh&

如果想要监控标准输出可以使用：

tail -f nohup.out

FreeBSD可以同时运行多个进程，在shell下直接输入命令后，shell将进程放到前台执行。如果要将进程放到后台执行，需要在命令行的结尾加上一个 “&” 符号。下面的命令从后台执行，从ftp.isc.org下载文件。

$ fetch ftp://ftp.isc.org/pub/inn/inn-1.7.2.tar.gz &

当程序已经在前台执行的时候，可以使用^Z将这个程序挂起，暂停执行。然后可以使用bg命令将这个挂起的程序放到后台执行，或者使用fg将某个在后台或挂起的进程放到前台执行。

当在后台运行了程序的时候，可以用jobs命令来查看后台作业的状态。在有多个后台程序时，要使用来参数的fg命令将不同序号的后台作业切换到前台上运行。

$ jobs

[1]+ Running fetch ftp://ftp.isc.org/pub/inn/inn-1.7.2.tar.gz &

$ fg %1

fetch ftp://ftp.isc.org/pub/inn/inn-1.7.2.tar.gz

在启动了多个程序之后，可以使用ps命令来查看这些进程及其状态。

$ ps

PID TT STAT TIME COMMAND

501 p2 Ss 0:00.24 -bash (bash)

988 p2 R+ 0:00.00 ps

765 p3 Is+ 0:00.28 -bash (bash)

230 v0 Is+ 0:00.14 -bash (bash)

显示的结果包括进程的标识号PID，控制终端TT（p0表示控制终端为ttyp0），进程的状态STAT，进程使用的处理器时间TIME和具体的命令。

可以给ps命令加上参数，来获得更多的输出内容，以下命令将输出系统中所有的进程：

$ ps waux

USER PID %CPU %MEM VSZ RSS TT STAT STARTED TIME COMMAND

wb 989 0.0 0.4 400 236 p2 R+ 5:48PM 0:00.00 ps -aux

root 1 0.0 0.1 496 72 ?? Is 10:12PM 0:00.02 /sbin/init --

root 2 0.0 0.0 0 0 ?? DL 10:12PM 0:07.05 (pagedaemon)

root 3 0.0 0.0 0 0 ?? DL 10:12PM 0:00.20 (vmdaemon)

root 4 0.0 0.0 0 0 ?? DL 10:12PM 0:04.27 (syncer)

root 27 0.0 0.0 204 0 ?? IWs - 0:00.00 (adjkerntz)

root 91 0.0 0.5 820 328 ?? Is 2:12PM 0:00.82 syslogd

daemon 100 0.0 0.0 792 0 ?? IWs - 0:00.00 (portmap)

root 131 0.0 0.3 864 164 ?? Is 2:12PM 0:00.06 inetd

root 134 0.0 0.3 980 192 ?? Is 2:12PM 0:00.11 cron

root 138 0.0 0.6 1252 380 ?? Is 2:12PM 0:00.11 sendmail: accepti

wb 230 0.0 1.1 1540 668 v0 Is+ 2:12PM 0:00.14 -bash (bash)

root 231 0.0 0.0 824 0 v1 IWs+ - 0:00.00 (getty)

root 232 0.0 0.0 824 0 v2 IWs+ - 0:00.00 (getty)

root 500 0.0 0.9 876 524 ?? Ss 4:19PM 0:01.78 telnetd

wb 501 0.0 1.4 1540 888 p2 Ss 4:19PM 0:00.24 -bash (bash)

root 698 0.0 1.5 1644 900 ?? Is 4:49PM 0:00.02 /usr/local/sbin/s

root 700 0.0 1.2 1308 748 ?? Ss 4:49PM 0:00.22 /usr/local/sbin/n

root 702 0.0 3.4 2900 2112 ?? S 4:49PM 0:00.32 /usr/local/sbin/s

root 764 0.0 0.9 880 540 ?? Is 5:10PM 0:00.22 telnetd

wb 765 0.0 1.7 1536 1052 p3 Is+ 5:10PM 0:00.28 -bash (bash)

root 0 0.0 0.0 0 0 ?? DLs 10:12PM 0:00.02 (swapper)

当用户启动一个进程的时候，这个进程是运行在前台，使用与相应控制终端相联系的标准输入、输出进行输入和输出。即使将进程的输入输出重定向，并将进程放在后台执行，进程仍然和当前终端设备有关系。正因为如此，在当前的登录会话结束时，控制终端设备将和登录进程相脱离，那么系统就向所有与这个终端相联系的进程发送SIGHUP的信号，通知进程线路已经挂起了，如果程序没有接管这个信号的处理，那么缺省的反应是进程结束。因此普通的程序并不能真正脱离登录会话而运行进程，为了使得在系统登录后还可以正常执行，只有使用命令nohup来启动相应程序。

从上面的ps的输出结果可以看出，有些程序没有控制终端，这些程序通常是一些后台进程。使用命令nohup当然可以启动这样的程序，但nohup启动的程序在进程执行完毕就退出，而常见的一些服务进程通常永久的运行在后台，不向屏幕输出结果。在Unix中这些永久的后台进程称为守护进程（daemon）。守护进程通常从系统启动时自动开始执行，系统关闭时才停止。如果偶然某个守护进程消失了，那么它提供的服务将不再能被使用。

在守护进程中，最重要的一个是超级守护进程inetd，这个进程接管了大部分网络服务，但并不是对每个服务都自己进行处理，而是依据连接请求，启动不同的服务程序与客户机打交道。inetd支持网络服务种类在它的设置文件/etc/inet.conf中定义。inet.conf文件中的每一行就对应一个端口地址，当inetd接受到连接这个端口的连接请求时，就启动相应的进程进行处理。使用inetd的好处是系统不必启动很多守护进程，从而节约了系统资源，然而使用inetd启动守护进程相应反应会迟缓一些，不适合用于被密集访问的服务进程

### **nohup 命令**

　　用途：不挂断地运行命令。

　　语法：nohup Command [ Arg ... ] [　& ]

　 　描述：nohup 命令运行由 Command 参数和任何相关的 Arg 参数指定的命令，忽略所有挂断（SIGHUP）信号。在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令，添加 & （ 表示“and”的符号）到命令的尾部。

　　 无论是否将 nohup 命令的输出重定向到终端，输出都将附加到当前目录的 nohup.out 文件中。如果当前目录的 nohup.out 文件不可写，输出重定向到 $HOME/nohup.out 文件中。如果没有文件能创建或打开以用于追加，那么 Command 参数指定的命令不可调用。如果标准错误是一个终端，那么把指定的命令写给标准错误的所有输出作为标准输出重定向到相同的文件描述符。

　　退出状态：该命令返回下列出口值：

　　126 可以查找但不能调用 Command 参数指定的命令。

　　127 nohup 命令发生错误或不能查找由 Command 参数指定的命令。

　　否则，nohup 命令的退出状态是 Command 参数指定命令的退出状态。

**nohup命令及其输出文件**

　　nohup命令：如果你正在运行一个进程，而且你觉得在退出帐户时该进程还不会结束，那么可以使用nohup命令。该命令可以在你退出帐户/关闭终端之后继续运行相应的进程。nohup就是不挂起的意思( n ohang up)。

该命令的一般形式为**：nohup command &**

使用nohup命令提交作业

如果使用nohup命令提交作业，那么在缺省情况下该作业的所有输出都被重定向到一个名为nohup.out的文件中，除非另外指定了输出文件：

nohup command > myout.file 2>&1 &

在上面的例子中，输出被重定向到myout.file文件中。

使用 jobs 查看任务。

使用 fg %n　关闭。

另外有两个常用的ftp工具ncftpget和ncftpput，可以实现后台的ftp上传和下载，这样我就可以利用这些命令在后台上传和下载文件了。

nohup /weblogic/domains/mac\_domain/bin/startWebLogic.sh >> /weblogic/domains/mac\_domain/bin/nohup.log&

nohup /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/startWebLogic.sh >> /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/nohup.log&

nohup /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/startManagedWebLogic.sh mac3\_server >> /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/nohup.log&

nohup /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/stopWebLogic.sh mac3\_server >> /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/bin/nohup.log&

### $ man nohup

NOHUP(1) User Commands NOHUP(1)

NAME

nohup - run a command immune to hangups, with output to a non-tty

SYNOPSIS

nohup COMMAND [ARG]...

nohup OPTION

DESCRIPTION

Run COMMAND, ignoring hangup signals.

--help display this help and exit

--version

output version information and exit

If standard input is a terminal, redirect it from /dev/null. If standard output is a terminal, append output to ‘nohup.out’ if possible, ‘$HOME/nohup.out’ other- wise. If standard error is a terminal, redirect it to standard output. To save output to FILE, use ‘nohup COMMAND > FILE’.

# 系统管理和用户管理

## 用户管理

1、Linux里查看所有用户

linux里,并没有像windows的net user,net localgroup这些方便的命令来管理用户.

（1）在终端里.其实只需要查看 /etc/passwd文件就行了.

（2）看第三个参数:500以上的,就是后面建的用户了.其它则为系统的用户.

或者用cat /etc/passwd |cut -f 1 -d :

2、用户管理命令

useradd 注：添加用户

adduser 注：添加用户

passwd 注：为用户设置密码

usermod 注：修改用户命令，可以通过usermod 来修改登录名、用户的家目录等等；

pwcov 注：同步用户从/etc/passwd 到/etc/shadow

pwck 注：pwck是校验用户配置文件/etc/passwd 和/etc/shadow 文件内容是否合法或完整；

pwunconv 注：是pwcov 的立逆向操作，是从/etc/shadow和 /etc/passwd 创建/etc/passwd ，然后会删除 /etc/shadow 文件；

finger 注：查看用户信息工具

id 注：查看用户的UID、GID及所归属的用户组

chfn 注：更改用户信息工具

su 注：用户切换工具

sudo 注：sudo 是通过另一个用户来执行命令（execute a

command as another user），su 是用来切换用户，然后通过切换到的用户来完成相应的任务，但sudo

能后面直接执行命令，比如sudo 不需要root 密码就可以执行root 赋与的执行只有root才能执行相应的命令；但得通过visudo

来编辑/etc/sudoers来实现；

visudo 注：visodo 是编辑 /etc/sudoers 的命令；也可以不用这个命令，直接用vi 来编辑 /etc/sudoers 的效果是一样的；

sudoedit 注：和sudo 功能差不多；

3、管理用户组（group）的工具或命令；

groupadd 注：添加用户组；

groupdel 注：删除用户组；

groupmod 注：修改用户组信息

groups 注：显示用户所属的用户组

grpck

grpconv 注：通过/etc/group和/etc/gshadow 的文件内容来同步或创建/etc/gshadow ，如果/etc/gshadow 不存在则创建；

grpunconv 注：通过/etc/group 和/etc/gshadow 文件内容来同步或创建/etc/group ，然后删除gshadow文件。

## man

man(1) man(1)

NAME

man - format and display the on-line manual pages

SYNOPSIS

man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config\_file] [-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S section\_list] [section] name ...

DESCRIPTION

man formats and displays the on-line manual pages. If you specify section, man only looks in that section of the manual. name is normally the name of the manual

page, which is typically the name of a command, function, or file. However, if name contains a slash (/) then man interprets it as a file specification, so that

you can do man ./foo.5 or even man /cd/foo/bar.1.gz.

See below for a description of where man looks for the manual page files.

MANUAL SECTIONS

The standard sections of the manual include:

1 User Commands

2 System Calls

3 C Library Functions

4 Devices and Special Files

5 File Formats and Conventions

6 Games et. Al.

7 Miscellanea

8 System Administration tools and Deamons

Distributions customize the manual section to their specifics, which often include additional sections.

OPTIONS

-C config\_file

Specify the configuration file to use; the default is /etc/man.config. (See man.config(5).)

-M path

Specify the list of directories to search for man pages. Separate the directories with colons. An empty list is the same as not specifying -M at all. See

SEARCH PATH FOR MANUAL PAGES.

-P pager

Specify which pager to use. This option overrides the MANPAGER environment variable, which in turn overrides the PAGER variable. By default, man uses

/usr/bin/less -is.

-B Specify which browser to use on HTML files. This option overrides the BROWSER environment variable. By default, man uses /usr/bin/less-is,

-H Specify a command that renders HTML files as text. This option overrides the HTMLPAGER environment variable. By default, man uses /bin/cat,

-S section\_list

List is a colon separated list of manual sections to search. This option overrides the MANSECT environment variable.

-a By default, man will exit after displaying the first manual page it finds. Using this option forces man to display all the manual pages that match name, not

just the first.

-c Reformat the source man page, even when an up-to-date cat page exists. This can be meaningful if the cat page was formatted for a screen with a different

number of columns, or if the preformatted page is corrupted.

-d Don?. actually display the man pages, but do print gobs of debugging information.

-D Both display and print debugging info.

-f Equivalent to whatis.

-F or --preformat

Format only - do not display.

-h Print a help message and exit.

-k Equivalent to apropos.

-K Search for the specified string in \*all\* man pages. Warning: this is probably very slow! It helps to specify a section. (Just to give a rough idea, on my

machine this takes about a minute per 500 man pages.)

-m system

Specify an alternate set of man pages to search based on the system name given.

-p string

Specify the sequence of preprocessors to run before nroff or troff. Not all installations will have a full set of preprocessors. Some of the preprocessors

and the letters used to designate them are: eqn (e), grap (g), pic (p), tbl (t), vgrind (v), refer (r). This option overrides the MANROFFSEQ environment

variable.

-t Use /usr/bin/groff -Tps -mandoc to format the manual page, passing the output to stdout. The default output format of /usr/bin/groff -Tps -mandoc is

Postscript, refer to the manual page of /usr/bin/groff -Tps -mandoc for ways to pick an alternate format.

Depending on the selected format and the availability of printing devices, the output may need to be passed through some filter or another before being printed.

-w or --path

Don?. actually display the man pages, but do print the location(s) of the files that would be formatted or displayed. If no argument is given: display (on

stdout) the list of directories that is searched by man for man pages. If manpath is a link to man, then "manpath" is equivalent to "man --path".

-W Like -w, but print file names one per line, without additional information. This is useful in shell commands like man -aW man | xargs ls -l

CAT PAGES

Man will try to save the formatted man pages, in order to save formatting time the next time these pages are needed. Traditionally, formatted versions of pages in

DIR/manX are saved in DIR/catX, but other mappings from man dir to cat dir can be specified in /etc/man.config. No cat pages are saved when the required cat direc-

tory does not exist. No cat pages are saved when they are formatted for a line length different from 80. No cat pages are saved when man.config contains the line

NOCACHE.

It is possible to make man suid to a user man. Then, if a cat directory has owner man and mode 0755 (only writable by man), and the cat files have owner man and

mode 0644 or 0444 (only writable by man, or not writable at all), no ordinary user can change the cat pages or put other files in the cat directory. If man is not

made suid, then a cat directory should have mode 0777 if all users should be able to leave cat pages there.

The option -c forces reformatting a page, even if a recent cat page exists.

HTML PAGES

Man will find HTML pages if they live in directories named as expected to be ".html", thus a valid name for an HTML version of the ls(1) man page would be

/usr/share/man/htmlman1/ls.1.html.

SEARCH PATH FOR MANUAL PAGES

man uses a sophisticated method of finding manual page files, based on the invocation options and environment variables, the /etc/man.config configuration file, and

some built in conventions and heuristics.

First of all, when the name argument to man contains a slash (/), man assumes it is a file specification itself, and there is no searching involved.

But in the normal case where name doesn?. contain a slash, man searches a variety of directories for a file that could be a manual page for the topic named.

If you specify the -M pathlist option, pathlist is a colon-separated list of the directories that man searches.

If you don?. specify -M but set the MANPATH environment variable, the value of that variable is the list of the directories that man searches.

If you don?. specify an explicit path list with -M or MANPATH, man develops its own path list based on the contents of the configuration file /etc/man.config. The

MANPATH statements in the configuration file identify particular directories to include in the search path.

Furthermore, the MANPATH\_MAP statements add to the search path depending on your command search path (i.e. your PATH environment variable). For each directory that

may be in the command search path, a MANPATH\_MAP statement specifies a directory that should be added to the search path for manual page files. man looks at the

PATH variable and adds the corresponding directories to the manual page file search path. Thus, with the proper use of MANPATH\_MAP, when you issue the command man

xyz, you get a manual page for the program that would run if you issued the command xyz.

In addition, for each directory in the command search path (we?.l call it a "command directory") for which you do not have a MANPATH\_MAP statement, man automati-

cally looks for a manual page directory "nearby" namely as a subdirectory in the command directory itself or in the parent directory of the command directory.

You can disable the automatic "nearby" searches by including a NOAUTOPATH statement in /etc/man.config.

In each directory in the search path as described above, man searches for a file named topic.section, with an optional suffix on the section number and possibly a

compression suffix. If it doesn?. find such a file, it then looks in any subdirectories named manN or catN where N is the manual section number. If the file is in

a catN subdirectory, man assumes it is a formatted manual page file (cat page). Otherwise, man assumes it is unformatted. In either case, if the filename has a

known compression suffix (like .gz), man assumes it is gzipped.

If you want to see where (or if) man would find the manual page for a particular topic, use the --path (-w) option.

ENVIRONMENT

MANPATH

If MANPATH is set, man uses it as the path to search for manual page files. It overrides the configuration file and the automatic search path, but is over-

ridden by the -M invocation option. See SEARCH PATH FOR MANUAL PAGES.

MANPL If MANPL is set, its value is used as the display page length. Otherwise, the entire man page will occupy one (long) page.

MANROFFSEQ

If MANROFFSEQ is set, its value is used to determine the set of preprocessors run before running nroff or troff. By default, pages are passed through the

tbl preprocessor before nroff.

MANSECT

If MANSECT is set, its value is used to determine which manual sections to search.

MANWIDTH

If MANWIDTH is set, its value is used as the width manpages should be displayed. Otherwise the pages may be displayed over the whole width of your screen.

MANPAGER

If MANPAGER is set, its value is used as the name of the program to use to display the man page. If not, then PAGER is used. If that has no value either,

/usr/bin/less -is is used.

BROWSER

The name of a browser to use for displaying HTML manual pages. If it is not set, /usr/bin/less -is is used.

HTMLPAGER

The command to use for rendering HTML manual pages as text. If it is not set, /bin/cat is used.

LANG If LANG is set, its value defines the name of the subdirectory where man first looks for man pages. Thus, the command ?.ANG=dk man 1 foo?.will cause man to

look for the foo man page in .../dk/man1/foo.1, and if it cannot find such a file, then in .../man1/foo.1, where ... is a directory on the search path.

NLSPATH, LC\_MESSAGES, LANG

The environment variables NLSPATH and LC\_MESSAGES (or LANG when the latter does not exist) play a role in locating the message catalog. (But the English

messages are compiled in, and for English no catalog is required.) Note that programs like col(1) called by man also use e.g. LC\_CTYPE.

PATH PATH helps determine the search path for manual page files. See SEARCH PATH FOR MANUAL PAGES.

SYSTEM SYSTEM is used to get the default alternate system name (for use with the -m option).

BUGS

The -t option only works if a troff-like program is installed.

If you see blinking \255 or <AD> instead of hyphens, put ?.ESSCHARSET=latin1?.in your environment.

TIPS

If you add the line

(global-set-key [(f1)] (lambda () (interactive) (manual-entry (current-word))))

to your .emacs file, then hitting F1 will give you the man page for the library call at the current cursor position.

To get a plain text version of a man page, without backspaces and underscores, try

# man foo | col -b > foo.mantxt

## who

使用：who

说明：显示本系统的登录人员

举例：whoami 显示自己

## passwd

使用：passwd [用户]

说明：修改密码，指定用户则修改指定用户密码

## su

使用：su [- ]   [用户名]

说明：su 命令使当前用户成为指定用户，若无指定，则成为超级用户，但必须输入该用户的密码，-选项表示用该用户的注册环境

## date

使用： date

date mmddhhmm[yy]

说明： date 无参数时用于显示系统时间，修改时间时参数形式为：月日时分[年]

**$ date -I**

**2015-11-11**

**$ date +%Y%m%d**

**20151214**

**date -d '-10 days' #显示10天前日期**

如果只减去一天的话，直接写就可以了。

|  |  |
| --- | --- |
| 1 | #date -d"yesterday 20150401" +%Y%m%d |

如果要减去几天，还可以这样写，如果用负数是往前数，

|  |  |
| --- | --- |
| 1 | #date -d"10 day ago 2015-04-01" +%Y-%m-%d |

$ date --help

用法：date [选项]... [+格式]

　或：date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

以给定的格式显示当前时间，或是设置系统日期。

-d,--date=字符串 显示指定字符串所描述的时间，而非当前时间

-f,--file=日期文件 类似--date，从日期文件中按行读入时间描述

-r, --reference=文件 显示文件指定文件的最后修改时间

-R, --rfc-2822 以RFC 2822格式输出日期和时间

例如：2006年8月7日，星期一 12:34:56 -0600

--rfc-3339=TIMESPEC 以RFC 3339 格式输出日期和时间。

TIMESPEC=`date'，`seconds'，或 `ns'

表示日期和时间的显示精度。

日期和时间单元由单个的空格分开：

2006-08-07 12:34:56-06:00

-s, --set=字符串 设置指定字符串来分开时间

-u, --utc, --universal 输出或者设置协调的通用时间

--help 显示此帮助信息并退出

--version 显示版本信息并退出

给定的格式FORMAT 控制着输出，解释序列如下：

|  |  |
| --- | --- |
| %% | 一个文字的 % |
| %a | 当前locale 的星期名缩写(例如： 日，代表星期日) |
| %A | 当前locale 的星期名全称 (如：星期日) |
| %b | 当前locale 的月名缩写 (如：一，代表一月) |
| %B | 当前locale 的月名全称 (如：一月) |
| %c | 当前locale 的日期和时间 (如：2005年3月3日 星期四 23:05:25) |
| %C | 世纪；比如 %Y，通常为省略当前年份的后两位数字(例如：20) |
| %d | 按月计的日期(例如：01) |
| %D | 按月计的日期；等于%m/%d/%y |
| %e | 按月计的日期，添加空格，等于%\_d |
| %F | 完整日期格式，等价于 %Y-%m-%d |
| %g | ISO-8601 格式年份的最后两位 (参见%G) |
| %G | ISO-8601 格式年份 (参见%V)，一般只和 %V 结合使用 |
| %h | 等于%b |
| %H | 小时(00-23) |
| %I | 小时(00-12) |
| %j | 按年计的日期(001-366) |
| %k | 时(0-23) |
| %l | 时(1-12) |
| %m | 月份(01-12) |
| %M | 分(00-59) |
| %n | 换行 |
| %N | 纳秒(000000000-999999999) |
| %p | 当前locale 下的"上午"或者"下午"，未知时输出为空 |
| %P | 与%p 类似，但是输出小写字母 |
| %r | 当前locale 下的 12 小时时钟时间 (如：11:11:04 下午) |
| %R | 24 小时时间的时和分，等价于 %H:%M |
| %s | 自UTC 时间 1970-01-01 00:00:00 以来所经过的秒数 |
| %S | 秒(00-60) |
| %t | 输出制表符 Tab |
| %T | 时间，等于%H:%M:%S |
| %u | 星期，1 代表星期一 |
| %U | 一年中的第几周，以周日为每星期第一天(00-53) |
| %V | ISO-8601 格式规范下的一年中第几周，以周一为每星期第一天(01-53) |
| %w | 一星期中的第几日(0-6)，0 代表周一 |
| %W | 一年中的第几周，以周一为每星期第一天(00-53) |
| %x | 当前locale 下的日期描述 (如：12/31/99) |
| %X | 当前locale 下的时间描述 (如：23:13:48) |
| %y | 年份最后两位数位 (00-99) |
| %Y | 年份 |
| %z +hhmm | 数字时区(例如，-0400) |
| %:z +hh:mm | 数字时区(例如，-04:00) |
| %::z +hh:mm:ss | 数字时区(例如，-04:00:00) |
| %:::z | 数字时区带有必要的精度 (例如，-04，+05:30) |
| %Z | 按字母表排序的时区缩写 (例如，EDT) |

默认情况下，日期的数字区域以0 填充。

以下可选标记可以跟在"%"后:

- (连字符)不填充该域

\_ (下划线)以空格填充

0 (数字0)以0 填充

^ 如果可能，使用大写字母

# 如果可能，使用相反的大小写

在任何标记之后还允许一个可选的域宽度指定，它是一个十进制数字。

作为一个可选的修饰声明，它可以是E，在可能的情况下使用本地环境关联的

表示方式；或者是O，在可能的情况下使用本地环境关联的数字符号。

$ man date

DATE(1) User Commands DATE(1)

NAME

date - print or set the system date and time

SYNOPSIS

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION

Display the current time in the given FORMAT, or set the system date.

-d, --date=STRING

display time described by STRING, not ‘now’

-f, --file=DATEFILE

like --date once for each line of DATEFILE

-r, --reference=FILE

display the last modification time of FILE

-R, --rfc-2822

output date and time in RFC 2822 format. Example: Mon, 07 Aug 2006 12:34:56 -0600

--rfc-3339=TIMESPEC

output date and time in RFC 3339 format. TIMESPEC=‘date’, ‘seconds’, or ‘ns’ for date and

time to the indicated precision. Date and time components are separated by a single space:

2006-08-07 12:34:56-06:00

-s, --set=STRING

set time described by STRING

-u, --utc, --universal

print or set Coordinated Universal Time

--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)

%C century; like %Y, except omit last two digits (e.g., 20)

%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 ( 0..23)

%l hour ( 1..12)

%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 timezone (e.g., -0400)

%:z +hh:mm numeric timezone (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.

DATE STRING

The --date=STRING is a mostly free format human readable date string such as "Sun, 29 Feb 2004

16:21:42 -0800" or "2004-02-29 16:21:42" or even "next Thursday". A date string may contain items

indicating calendar date, time of day, time zone, day of week, relative time, relative date, and

numbers. An empty string indicates the beginning of the day. The date string format is more com-

plex than is easily documented here but is fully described in the info documentation.

ENVIRONMENT

TZ Specifies the timezone, unless overridden by command line parameters. If neither is speci-

fied, the setting from /etc/localtime is used.

## shutdown

使用：shutdown [-h] [-y]

说明：关机命令，不同UNIX操行系统，可能会有所不同。

HPUX立即关机命令：shutdown -h -y 0（0表示等待时间为0秒）

进入单用户状态：shutdown

重新启动机器：reboot -r

## echo

使用：echo 环境变量

说明：显示环境变量内容

## env

使用：env   [选项] 路径名称

说明：显示所有环境变量

env | grep HOME

HOME的环境变量设置 是 /etc/passwd

## uname

使用：uname [-a]

说明：显示系统信息，-a显示所有信息

## netstat

使用：netstat -in/an

说明：查看端口状态、IP地址

举例：netstat -an | grep 5600    查看5600端口的状态

netstat -in 查看网卡的IP地址

### AIX查看某个端口被哪个进程占用

问题描述：

在系统管理过程中经常遇到的情况就是在启动某个进程时，会提示端口被占用。如启动WebSphere管理控制台时经常碰到9090端口被占用。用 netstat -an | grep 9090 可以看到该端口正被监听，但却不知道是哪个进程占用了该端口，当然也可以改变端口但比较麻烦。如果能找到是哪个进程占用了该端口，把这个进程kill掉就 可以了。

问题解决：

1. netstat -Aan|grep <portnumber>

找到该端口连接对应的PCB/ADDR和连接的协议类型。

注：PCB ----Protocol Control Block

2. 如果是tcp连接，则rmsock <PCB/ADDR> tcpcb

如果是udp连接，则rmsock <PCB/ADDR> inpcb

下面我们以telnet服务所使用的23号端口为例，说明该方法：

#netstat -Aan|grep 23

f1000200019ce398 tcp 0 0 \*.23 \*.\* LISTEN

可以看到PCB/ADDR为f1000200019ce398，且协议类型为tcp。

#rmsock f1000200019ce398 tcpcb

The socket 0x19ce008 is being held by proccess 185006 (inetd).

命令报告该端口正在被inetd进程使用，PID为185006。

注意：rmsock命令需要root权限执行。

(更新)您要看到PID对应的是哪个程序的端口，只需要在任务管理器中，选择查看＼列，在”列”选项卡中，把PID号勾上,就可以在任务管理器中看到进程的PID号了...希望对您有用

在CMD里面输入netstat -ano

可以找到端口和应用程序对应的PID号

再打开任务管理器，选到进程，选择查看\列，把PID号钩选，就可以看到响应进程的PID号了 ``再去对比就行了

希望对你有帮助

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

可能是命令没有输入正确哦~~`

注意netstat 跟-ano之间有一个空格哦~`

是netstat -ano

另外:不知道您是什么系统,怎么会出现英文的解释

(!!参数分开打也可以,比如netstat -o就会显示出进程ID)

对应的中文解释是:

显示协议统计信息和当前 TCP/IP 网络连接。

NETSTAT [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [interval]

|  |  |
| --- | --- |
| -a | 显示所有连接和监听端口。 |
| -b | 显示包含于创建每个连接或监听端口的可执行组件。在某些情况下已知可执行组件拥有多个独立组件，并且在这些情况下包含于创建连接或监听端口的组件序列被显示。这种情况下，可执行组件名在底部的 [] 中，顶部是其调用的组件，等等，直到 TCP/IP 部分。注意此选项可能需要很长时间，如果没有足够权限可能失败。 |
| -e | 显示以太网统计信息。此选项可以与 -s选项组合使用。 |
| -n | 以数字形式显示地址和端口号。 |
| -o | 显示与每个连接相关的所属进程 ID。 |
| -p proto | 显示 proto 指定的协议的连接；proto 可以是下列协议之一: TCP、UDP、TCPv6 或 UDPv6。 如果与 -s 选项一起使用以显示按协议统计信息，proto 可以是下列协议之一:IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 或 UDPv6。 |
| -r | 显示路由表。 |
| -s | 显示按协议统计信息。默认地，显示 IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 和 UDPv6 的统计信息； |
| -p | 选项用于指定默认情况的子集。 |
| -v | 与 -b 选项一起使用时将显示包含于为所有可执行组件创建连接或监听端口的组件。 |
| -interval | 重新显示选定统计信息，每次显示之间暂停时间间隔(以秒计)。按 CTRL+C 停止重新显示统计信息。如果省略，netstat 显示当前配置信息(只显示一次) |

网络状态说明：

|  |  |
| --- | --- |
| estab | 已建立连接 |
| listen | 监听中，等待连接 |
| syn\_sent | 传送syn信号，请求建立连接 |
| syn\_recv | 收到syn信号，表示收到连接请求 |
| fin\_wait1 | 套接字关闭，正终止连接 |
| fin\_wait2 | 连接关闭，套接字等待远端的终止请求 |
| time\_wait | 套接字在关闭后等待处理packets |
| closed | 套接字关闭 |
| close\_wait | 远端已终止，等待套接字关闭 |
| last\_ack | 远端已终止，套接字已关闭，等待响应 |
| closing | 套接字均已终止，但数据没有全部送出 |
| unknown | 未知的状态 |

ps命令：收集、查看应用进程活动信息

vmstat命令：用于获取CPU、页面调度和内存使用情况的总体图形描述

iostat命令：用于确定磁盘和CPU的使用情况

netstat命令：用于确定发送的接收的信息包数

Top命令：Top程序以2秒的缺省时间间隔从系统上抽取并显示统计信息

##grep某进程本身的进程，而不是grep出来的进程

ps -ef | grep ftsapp |grep -v grep

##根据进程名字获取进程号，并将该进程杀死

kill `ps -ef|grep ftsapp|grep -v grep|awk '{print $2}'`

##查看当前用户是什么

whoami

##查看当前用户的登录的详尽时间

who am I

##显示当前操作系统的名称

uname

为：Linux

##获取操作系统名称

hostname

如为：fedora.vm

#在文件中搜索字符串

grep 'sum' test.sh

如为在test.sh文件中搜索sum关键字

echo &date "+%B%d%A"

### $ man netstat

NETSTAT(8) Linux Programmer’s Manual NETSTAT(8)

NAME

netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

SYNOPSIS

netstat [address\_family\_options] [--tcp|-t] [--udp|-u] [--raw|-w] [--listening|-l] [--all|-a] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports]

[--symbolic|-N] [--extend|-e[--extend|-e]] [--timers|-o] [--program|-p] [--verbose|-v] [--continuous|-c] [delay]

netstat {--route|-r} [address\_family\_options] [--extend|-e[--extend|-e]] [--verbose|-v] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports] [--con-

tinuous|-c] [delay]

netstat {--interfaces|-I|-i} [iface] [--all|-a] [--extend|-e] [--verbose|-v] [--program|-p] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports]

[--continuous|-c] [delay]

netstat {--groups|-g} [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports] [--continuous|-c] [delay]

netstat {--masquerade|-M} [--extend|-e] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports] [--continuous|-c] [delay]

netstat {--statistics|-s} [--tcp|-t] [--udp|-u] [--raw|-w] [delay]

netstat {--version|-V}

netstat {--help|-h}

address\_family\_options:

[--protocol={inet,unix,ipx,ax25,netrom,ddp}[,...]] [--unix|-x] [--inet|--ip] [--ax25] [--ipx] [--netrom] [--ddp]

NOTE

This program is obsolete. Replacement for netstat is ss. Replacement for netstat -r is ip route. Replacement for netstat -i is ip -s link. Replacement for net-

stat -g is ip maddr.

DESCRIPTION

Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument, as follows:

(none)

By default, netstat displays a list of open sockets. If you don’t specify any address families, then the active sockets of all configured address families will be printed.

--route , -r

Display the kernel routing tables.

--groups , -g

Display multicast group membership information for IPv4 and IPv6.

--interfaces=iface , -I=iface , -i

Display a table of all network interfaces, or the specified iface.

--masquerade , -M

Display a list of masqueraded connections.

--statistics , -s

Display summary statistics for each protocol.

OPTIONS

--verbose , -v

Tell the user what is going on by being verbose. Especially print some useful information about unconfigured address families.

--numeric , -n

Show numerical addresses instead of trying to determine symbolic host, port or user names.

--numeric-hosts

shows numerical host addresses but does not affect the resolution of port or user names.

--numeric-ports

shows numerical port numbers but does not affect the resolution of host or user names.

--numeric-users

shows numerical user IDs but does not affect the resolution of host or port names.

--protocol=family , -A

Specifies the address families (perhaps better described as low level protocols) for which connections are to be shown. family is a comma (’,’) separated list of

address family keywords like inet, unix, ipx, ax25, netrom, and ddp. This has the same effect as using the --inet, --unix (-x), --ipx, --ax25, --netrom, and --ddp

options.

The address family inet includes raw, udp and tcp protocol sockets.

-c, --continuous

This will cause netstat to print the selected information every second continuously.

-e, --extend

Display additional information. Use this option twice for maximum detail.

-o, --timers

Include information related to networking timers.

-p, --program

Show the PID and name of the program to which each socket belongs.

-l, --listening

Show only listening sockets. (These are omitted by default.)

-a, --all

Show both listening and non-listening (for TCP this means established connections) sockets. With the --interfaces option, show interfaces that are not marked

-F

Print routing information from the FIB. (This is the default.)

-C

Print routing information from the route cache.

-Z --context

If SELinux enabled print SELinux context.

-T --notrim

Stop trimming long addresses.

delay

Netstat will cycle printing through statistics every delay seconds. UP.

OUTPUT

Active Internet connections (TCP, UDP, raw)

Proto

The protocol (tcp, udp, raw) used by the socket.

Recv-Q

The count of bytes not copied by the user program connected to this socket.

Send-Q

The count of bytes not acknowledged by the remote host.

Local Address

Address and port number of the local end of the socket. Unless the --numeric (-n) option is specified, the socket address is resolved to its canonical host name

(FQDN), and the port number is translated into the corresponding service name.

Foreign Address

Address and port number of the remote end of the socket. Analogous to "Local Address."

State

The state of the socket. Since there are no states in raw mode and usually no states used in UDP, this column may be left blank. Normally this can be one of several

values:

ESTABLISHED

The socket has an established connection.

SYN\_SENT

The socket is actively attempting to establish a connection.

SYN\_RECV

A connection request has been received from the network.

FIN\_WAIT1

The socket is closed, and the connection is shutting down.

FIN\_WAIT2

Connection is closed, and the socket is waiting for a shutdown from the remote end.

TIME\_WAIT

The socket is waiting after close to handle packets still in the network.

CLOSED The socket is not being used.

CLOSE\_WAIT

The remote end has shut down, waiting for the socket to close.

LAST\_ACK

The remote end has shut down, and the socket is closed. Waiting for acknowledgement.

LISTEN The socket is listening for incoming connections. Such sockets are not included in the output unless you specify the --listening (-l) or --all (-a) option.

CLOSING

Both sockets are shut down but we still don’t have all our data sent.

UNKNOWN

The state of the socket is unknown.

User

The username or the user id (UID) of the owner of the socket.

PID/Program name

Slash-separated pair of the process id (PID) and process name of the process that owns the socket. --program causes this column to be included. You will also need

superuser privileges to see this information on sockets you don’t own. This identification information is not yet available for IPX sockets.

Timer

(this needs to be written)

Active UNIX domain Sockets

Proto

The protocol (usually unix) used by the socket.

RefCnt

The reference count (i.e. attached processes via this socket).

Flags

The flags displayed is SO\_ACCEPTON (displayed as ACC), SO\_WAITDATA (W) or SO\_NOSPACE (N). SO\_ACCECPTON is used on unconnected sockets if their corresponding pro-

cesses are waiting for a connect request. The other flags are not of normal interest.

Type

There are several types of socket access:

SOCK\_DGRAM

The socket is used in Datagram (connectionless) mode.

SOCK\_STREAM

This is a stream (connection) socket.

SOCK\_RAW

The socket is used as a raw socket.

SOCK\_RDM

This one serves reliably-delivered messages.

SOCK\_SEQPACKET

This is a sequential packet socket.

SOCK\_PACKET

Raw interface access socket.

UNKNOWN

Who ever knows what the future will bring us - just fill in here :-)

State

This field will contain one of the following Keywords:

FREE The socket is not allocated

LISTENING

The socket is listening for a connection request. Such sockets are only included in the output if you specify the --listening (-l) or --all (-a) option.

CONNECTING

The socket is about to establish a connection.

CONNECTED

The socket is connected.

DISCONNECTING

The socket is disconnecting.

(empty)

The socket is not connected to another one.

UNKNOWN

This state should never happen.

PID/Program name

Process ID (PID) and process name of the process that has the socket open. More info available in Active Internet connections section written above.

Path

This is the path name as which the corresponding processes attached to the socket.

Active IPX sockets

(this needs to be done by somebody who knows it)

Active NET/ROM sockets

(this needs to be done by somebody who knows it)

Active AX.25 sockets

(this needs to be done by somebody who knows it)

NOTES

Starting with Linux release 2.2 netstat -i does not show interface statistics for alias interfaces. To get per alias interface counters you need to setup explicit

rules using the ipchains(8) command.

FILES

/etc/services -- The services translation file

/proc -- Mount point for the proc filesystem, which gives access to kernel status information via the following files.

/proc/net/dev -- device information

/proc/net/raw -- raw socket information

/proc/net/tcp -- TCP socket information

/proc/net/udp -- UDP socket information

/proc/net/igmp -- IGMP multicast information

/proc/net/unix -- Unix domain socket information

/proc/net/ipx -- IPX socket information

/proc/net/ax25 -- AX25 socket information

/proc/net/appletalk -- DDP (appletalk) socket information

/proc/net/nr -- NET/ROM socket information

/proc/net/route -- IP routing information

/proc/net/ax25\_route -- AX25 routing information

/proc/net/ipx\_route -- IPX routing information

/proc/net/nr\_nodes -- NET/ROM nodelist

/proc/net/nr\_neigh -- NET/ROM neighbours

/proc/net/ip\_masquerade -- masqueraded connections

/proc/net/snmp -- statistics

SEE ALSO

ss(8),ip(8)

## ulimit

ulimit 命令。ulimit用于shell启动进程所占用的资源.

shell内建命令

语法格式:

ulimit [-acdfHlmnpsStvw] [size]

参数

-H 设置硬资源限制.

-S 设置软资源限制.

-a 显示当前所有的资源限制.

-c size:设置core文件的最大值.单位:blocks

-d size:设置[数据段](http://baike.baidu.com/view/1005328.htm" \t "http://baike.baidu.com/_blank)的最大值.单位:kbytes

-f size:设置创建文件的最大值.单位:blocks

-l size:设置在内存中锁定进程的最大值.单位:kbytes

-m size:设置可以使用的常驻内存的最大值.单位:kbytes

-n size:设置[内核](http://baike.baidu.com/view/1366.htm" \t "http://baike.baidu.com/_blank)可以同时打开的[文件描述符](http://baike.baidu.com/view/1303430.htm" \t "http://baike.baidu.com/_blank)的最大值.单位:n

-p size:设置管道[缓冲区](http://baike.baidu.com/view/266782.htm" \t "http://baike.baidu.com/_blank)的最大值.单位:kbytes

-s size:设置[堆栈](http://baike.baidu.com/view/93201.htm" \t "http://baike.baidu.com/_blank)的最大值.单位:kbytes

-t size:设置CPU使用时间的最大上限.单位:seconds

-v size:设置[虚拟内存](http://baike.baidu.com/view/976.htm" \t "http://baike.baidu.com/_blank)的最大值.单位:kbytes

-u <程序数目> 　用户最多可开启的程序数目

简单实例

1]在RH8的环境文件/etc/profile中,我们可以看到系统是如何配置ulimit的:

#grep ulimit /etc/profile

ulimit -S -c 0 > /dev/null 2>&1

这条语句设置了对软件资源和对core文件大小的设置

2]如果我们想要对由shell创建的文件大小作些限制,如:

#ll h

-rw-r--r-- 1 lee lee 150062 7月 22 02:39 h

#ulimit -f 100 #设置创建文件的最大块(一块=512字节)

#cat h>newh

File size limit exceeded

#ll newh

-rw-r--r-- 1 lee lee 51200 11月 8 11:47 newh

文件h的大小是150062字节,而我们设定的创建文件的大小是512字节x100块=51200字节

当然系统就会根据你的设置生成了51200字节的newh文件.

3]可以像实例1]一样,把你要设置的ulimit放在/etc/profile这个环境文件中.

## file

FILE(1) BSD General Commands Manual FILE(1)

NAME

file - determine file type

SYNOPSIS

file [-bchikLNnprsvz0] [--apple] [--mime-encoding] [--mime-type] [-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...

file -C [-m magicfiles]

file [--help]

DESCRIPTION

file tests each argument in an attempt to classify it. There are three sets of tests, performed in this order: filesystem tests, magic tests, and language tests.

The first test that succeeds causes the file type to be printed.

The type printed will usually contain one of the words text (the file contains only printing characters and a few common control characters and is probably safe to

read on an ASCII terminal), executable (the file contains the result of compiling a program in a form understandable to some UNIX kernel or another), or data meaning

anything else (data is usually ‘binary’ or non-printable). Exceptions are well-known file formats (core files, tar archives) that are known to contain binary data.

When modifying magic files or the program itself, make sure to preserve these keywords. Users depend on knowing that all the readable files in a directory have the

word ‘text’ printed. Don’t do as Berkeley did and change ‘shell commands text’ to ‘shell script’.

The filesystem tests are based on examining the return from a stat(2) system call. The program checks to see if the file is empty, or if it’s some sort of special

file. Any known file types appropriate to the system you are running on (sockets, symbolic links, or named pipes (FIFOs) on those systems that implement them) are

intuited if they are defined in the system header file

The magic tests are used to check for files with data in particular fixed formats. The canonical example of this is a binary executable (compiled program) a.out

file, whose format is defined in #include <a.out.h>

and possibly #include <exec.h>

in the standard include directory. These files have a ‘magic number’ stored in a particular place near the beginning of the file that tells the UNIX operating system

that the file is a binary executable, and which of several types thereof. The concept of a ‘magic’ has been applied by extension to data files. Any file with some

invariant identifier at a small fixed offset into the file can usually be described in this way. The information identifying these files is read from the compiled

magic file /usr/share/misc/magic.mgc, or the files in the directory /usr/share/misc/magic if the compiled file does not exist. In addition, if $HOME/.magic.mgc or

$HOME/.magic exists, it will be used in preference to the system magic files. If /etc/magic exists, it will be used together with other magic files.

If a file does not match any of the entries in the magic file, it is examined to see if it seems to be a text file. ASCII, ISO-8859-x, non-ISO 8-bit extended-ASCII

character sets (such as those used on Macintosh and IBM PC systems), UTF-8-encoded Unicode, UTF-16-encoded Unicode, and EBCDIC character sets can be distinguished by

the different ranges and sequences of bytes that constitute printable text in each set. If a file passes any of these tests, its character set is reported. ASCII,

ISO-8859-x, UTF-8, and extended-ASCII files are identified as ‘text’ because they will be mostly readable on nearly any terminal; UTF-16 and EBCDIC are only

‘character data’ because, while they contain text, it is text that will require translation before it can be read. In addition, file will attempt to determine other

characteristics of text-type files. If the lines of a file are terminated by CR, CRLF, or NEL, instead of the Unix-standard LF, this will be reported. Files that

contain embedded escape sequences or overstriking will also be identified.

Once file has determined the character set used in a text-type file, it will attempt to determine in what language the file is written. The language tests look for

particular strings (cf. #include <names.h>

) that can appear anywhere in the first few blocks of a file. For example, the keyword .br indicates that the file is most likely a troff(1) input file, just as the

keyword struct indicates a C program. These tests are less reliable than the previous two groups, so they are performed last. The language test routines also test

for some miscellany (such as tar(1) archives).

Any file that cannot be identified as having been written in any of the character sets listed above is simply said to be ‘data’.

OPTIONS

-b, --brief

Do not prepend filenames to output lines (brief mode).

-C, --compile

Write a magic.mgc output file that contains a pre-parsed version of the magic file or directory.

-c, --checking-printout

Cause a checking printout of the parsed form of the magic file. This is usually used in conjunction with the -m flag to debug a new magic file before

installing it.

-e, --exclude testname

Exclude the test named in testname from the list of tests made to determine the file type. Valid test names are:

apptype EMX application type (only on EMX).

text Various types of text files (this test will try to guess the text encoding, irrespective of the setting of the ‘encoding’ option).

encoding Different text encodings for soft magic tests.

tokens Looks for known tokens inside text files.

cdf Prints details of Compound Document Files.

compress Checks for, and looks inside, compressed files.

elf Prints ELF file details.

soft Consults magic files.

tar Examines tar files.

-F, --separator separator

Use the specified string as the separator between the filename and the file result returned. Defaults to ‘:’.

-f, --files-from namefile

Read the names of the files to be examined from namefile (one per line) before the argument list. Either namefile or at least one filename argument must be

present; to test the standard input, use ‘-’ as a filename argument.

-h, --no-dereference

option causes symlinks not to be followed (on systems that support symbolic links). This is the default if the environment variable POSIXLY\_CORRECT is not

defined.

-i, --mime

Causes the file command to output mime type strings rather than the more traditional human readable ones. Thus it may say ‘text/plain; charset=us-ascii’

rather than ‘ASCII text’. In order for this option to work, file changes the way it handles files recognized by the command itself (such as many of the text

file types, directories etc), and makes use of an alternative ‘magic’ file. (See the FILES section, below).

--mime-type, --mime-encoding

Like -i, but print only the specified element(s).

-k, --keep-going

Don’t stop at the first match, keep going. Subsequent matches will be have the string ‘\012- ’ prepended. (If you want a newline, see the ‘-r’ option.)

-L, --dereference

option causes symlinks to be followed, as the like-named option in ls(1) (on systems that support symbolic links). This is the default if the environment

variable POSIXLY\_CORRECT is defined.

-m, --magic-file magicfiles

Specify an alternate list of files and directories containing magic. This can be a single item, or a colon-separated list. If a compiled magic file is found

alongside a file or directory, it will be used instead.

-N, --no-pad

Don’t pad filenames so that they align in the output.

-n, --no-buffer

Force stdout to be flushed after checking each file. This is only useful if checking a list of files. It is intended to be used by programs that want file-

type output from a pipe.

-p, --preserve-date

On systems that support utime(2) or utimes(2), attempt to preserve the access time of files analyzed, to pretend that file never read them.

-r, --raw

Don’t translate unprintable characters to \ooo. Normally file translates unprintable characters to their octal representation.

-s, --special-files

Normally, file only attempts to read and determine the type of argument files which stat(2) reports are ordinary files. This prevents problems, because read-

ing special files may have peculiar consequences. Specifying the -s option causes file to also read argument files which are block or character special

files. This is useful for determining the filesystem types of the data in raw disk partitions, which are block special files. This option also causes file

to disregard the file size as reported by stat(2) since on some systems it reports a zero size for raw disk partitions.

-v, --version

Print the version of the program and exit.

-z, --uncompress

Try to look inside compressed files.

-0, --print0

Output a null character ‘\0’ after the end of the filename. Nice to cut(1) the output. This does not affect the separator which is still printed.

--help Print a help message and exit.

FILES

/usr/share/misc/magic.mgc Default compiled list of magic.

/usr/share/misc/magic Directory containing default magic files.

ENVIRONMENT

The environment variable MAGIC can be used to set the default magic file name. If that variable is set, then file will not attempt to open $HOME/.magic. file adds

‘.mgc’ to the value of this variable as appropriate. The environment variable POSIXLY\_CORRECT controls (on systems that support symbolic links), whether file will

attempt to follow symlinks or not. If set, then file follows symlink, otherwise it does not. This is also controlled by the -L and -h options.

SEE ALSO

magic(5), strings(1), od(1), hexdump(1,) file(1posix)

STANDARDS CONFORMANCE

This program is believed to exceed the System V Interface Definition of FILE(CMD), as near as one can determine from the vague language contained therein. Its behav-

ior is mostly compatible with the System V program of the same name. This version knows more magic, however, so it will produce different (albeit more accurate) out-

put in many cases.

The one significant difference between this version and System V is that this version treats any white space as a delimiter, so that spaces in pattern strings must be

escaped. For example,

>10 string language impress (imPRESS data)

in an existing magic file would have to be changed to

>10 string language\ impress (imPRESS data)

In addition, in this version, if a pattern string contains a backslash, it must be escaped. For example

0 string \begindata Andrew Toolkit document

in an existing magic file would have to be changed to

0 string \\begindata Andrew Toolkit document

SunOS releases 3.2 and later from Sun Microsystems include a file command derived from the System V one, but with some extensions. My version differs from Sun’s only

in minor ways. It includes the extension of the ‘&’ operator, used as, for example,

>16 long&0x7fffffff >0 not stripped

MAGIC DIRECTORY

The magic file entries have been collected from various sources, mainly USENET, and contributed by various authors. Christos Zoulas (address below) will collect

additional or corrected magic file entries. A consolidation of magic file entries will be distributed periodically.

The order of entries in the magic file is significant. Depending on what system you are using, the order that they are put together may be incorrect. If your old

file command uses a magic file, keep the old magic file around for comparison purposes (rename it to /usr/share/misc/magic.orig ).

EXAMPLES

$ file file.c file /dev/{wd0a,hda}

file.c: C program text

file: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),

dynamically linked (uses shared libs), stripped

/dev/wd0a: block special (0/0)

/dev/hda: block special (3/0)

$ file -s /dev/wd0{b,d}

/dev/wd0b: data

/dev/wd0d: x86 boot sector

$ file -s /dev/hda{,1,2,3,4,5,6,7,8,9,10}

/dev/hda: x86 boot sector

/dev/hda1: Linux/i386 ext2 filesystem

/dev/hda2: x86 boot sector

/dev/hda3: x86 boot sector, extended partition table

/dev/hda4: Linux/i386 ext2 filesystem

/dev/hda5: Linux/i386 swap file

/dev/hda6: Linux/i386 swap file

/dev/hda7: Linux/i386 swap file

/dev/hda8: Linux/i386 swap file

/dev/hda9: empty

/dev/hda10: empty

$ file -i file.c file /dev/{wd0a,hda}

file.c: text/x-c

file: application/x-executable

/dev/hda: application/x-not-regular-file

/dev/wd0a: application/x-not-regular-file

HISTORY

There has been a file command in every UNIX since at least Research Version 4 (man page dated November, 1973). The System V version introduced one significant major

change: the external list of magic types. This slowed the program down slightly but made it a lot more flexible.

This program, based on the System V version, was written by Ian Darwin <ian@darwinsys.com> without looking at anybody else’s source code.

John Gilmore revised the code extensively, making it better than the first version. Geoff Collyer found several inadequacies and provided some magic file entries.

Contributions by the ‘&’ operator by Rob McMahon, cudcv@warwick.ac.uk, 1989.

Guy Harris, guy@netapp.com, made many changes from 1993 to the present.

Primary development and maintenance from 1990 to the present by Christos Zoulas (christos@astron.com).

Altered by Chris Lowth, chris@lowth.com, 2000: Handle the -i option to output mime type strings, using an alternative magic file and internal logic.

Altered by Eric Fischer (enf@pobox.com), July, 2000, to identify character codes and attempt to identify the languages of non-ASCII files.

Altered by Reuben Thomas (rrt@sc3d.org), 2007 to 2008, to improve MIME support and merge MIME and non-MIME magic, support directories as well as files of magic, apply

many bug fixes and improve the build system.

The list of contributors to the ‘magic’ directory (magic files) is too long to include here. You know who you are; thank you. Many contributors are listed in the

source files.

LEGAL NOTICE

Copyright (c) Ian F. Darwin, Toronto, Canada, 1986-1999. Covered by the standard Berkeley Software Distribution copyright; see the file LEGAL.NOTICE in the source

distribution.

The files tar.h and is\_tar.c were written by John Gilmore from his public-domain tar(1) program, and are not covered by the above license.

BUGS

There must be a better way to automate the construction of the Magic file from all the glop in Magdir. What is it?

file uses several algorithms that favor speed over accuracy, thus it can be misled about the contents of text files.

The support for text files (primarily for programming languages) is simplistic, inefficient and requires recompilation to update.

The list of keywords in ascmagic probably belongs in the Magic file. This could be done by using some keyword like ‘\*’ for the offset value.

Complain about conflicts in the magic file entries. Make a rule that the magic entries sort based on file offset rather than position within the magic file?

The program should provide a way to give an estimate of ‘how good’ a guess is. We end up removing guesses (e.g. ‘Fromas first 5 chars of file) because’ they are not

as good as other guesses (e.g. ‘Newsgroups:’ versus ‘Return-Path:’ ). Still, if the others don’t pan out, it should be possible to use the first guess.

This manual page, and particularly this section, is too long.

RETURN CODE

file returns 0 on success, and non-zero on error.

If the file named by the file operand does not exist, cannot be read, or the type of the file named by the file operand cannot be determined, this is not be consid-

ered an error that affects the exit status.

AVAILABILITY

You can obtain the original author’s latest version by anonymous FTP on ftp.astron.com in the directory /pub/file/file-X.YZ.tar.gz

BSD October 9, 2008 BSD

## ifconfig

ifconfig是[linux](http://baike.baidu.com/view/1634.htm" \t "http://baike.baidu.com/_blank)中用于显示或配置网络设备（[网络接口卡](http://baike.baidu.com/view/547393.htm" \t "http://baike.baidu.com/_blank)）的命令，英文全称是network interfaces configuring。

配置网卡的IP地址语法：ifconfig eth0 192.168.0.1 netmask 255.255.255.0

语法

ifconfig [[网络设备](http://baike.baidu.com/view/1158081.htm" \t "http://baike.baidu.com/_blank)][down up -allmulti -arp -promisc][add<地址>][del<地址>][<硬件地址>] [media<网络媒介类型>][mem\_start<内存地址>][metric<数目>][mtu<字节>][netmask<子网掩码>][tunnel<地址>][-broadcast<地址>] [-pointopoint<地址>]

补充说明：ifconfig可设置网络设备的状态，或是显示当前的设置。

参　数：

|  |  |
| --- | --- |
| [[网络设备](http://baike.baidu.com/view/1158081.htm" \t "http://baike.baidu.com/_blank)] | 网络设备的名称。 |
| down | 关闭指定的网络设备。 |
| up | 启动指定的网络设备。 |
| -arp | 打开或关闭指定接口上使用的ARP协议。前面加上一个负号用于关闭该选项。 |
| -allmuti | 关闭或启动指定接口的无区别模式。前面加上一个负号用于关闭该选项。 |
| -promisc | 关闭或启动指定网络设备的promiscuous模式。前面加上一个负号用于关闭该选项。 |
| add<地址> | 设置网络设备IPv6的IP地址。 |
| del<地址> | 删除网络设备IPv6的IP地址。 |
| media<网络媒介类型> | 设置网络设备的媒介类型。 |
| mem\_start<内存地址> | 设置网络设备在主内存所占用的起始地址。 |
| metric<数目> | 指定在计算[数据包](http://baike.baidu.com/view/25880.htm" \t "http://baike.baidu.com/_blank)的转送次数时，所要加上的数目。 |
| mtu<字节> | 设置网络设备的MTU。 |
| netmask<子网掩码> | 设置网络设备的子网掩码。 |
| tunnel<地址> | 建立IPv4与IPv6之间的隧道通信地址。 |
| -broadcast<地址> | 将要送往指定地址的数据包当成[广播数据包](http://baike.baidu.com/view/567597.htm" \t "http://baike.baidu.com/_blank)来处理。 |
| -pointopoint<地址> | 与指定地址的网络设备建立直接连线，此模式具有[保密功能](http://baike.baidu.com/view/649905.htm" \t "http://baike.baidu.com/_blank)。 |

### 网卡命令

用于配置网卡的基本命令就是ifconfig。

在执行ifconfig 命令后，系统将在[内核](http://baike.baidu.com/view/1366.htm" \t "http://baike.baidu.com/_blank)表中设置必要的参数，这样Linux 就知道如何与网络上的网卡通信。ifconfig 命令有以下两种格式：

※ifconfig [interface]

※ifconfig interface [aftype] option | address …

ifconfig 的第一种格式（或使用不带任何参数的ifconfig 命令）可以用来查看当前系统的网络配置情况。

在刚刚安装完系统之后，实际上是在没有网卡或者网络连接的情况下使用Linux，但通过ifconfig 可以使用回绕方式工作，使计算机认为自己工作在网络上。

下面我们运行一下ifconfig 命令，不带参数的ifconfig 命令可以显示当前启动的网络接口，其输出结果为：

[root@machine1 /sbin]#ifconfig

eth0 Link encap:Ethernet HWaddr 52:54:AB:DD:6F:61

inet addr:210.34.6.89 Bcast:210.34.6.127 Mask:255.255.255.128

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:46299 errors:0 dropped:0 overruns:0 frame:189

TX packets:3057 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100

Interrupt:5 Base address:0xece0

lo Link encap:Local Loopback

inet addr:[127.0.0.1](http://baike.baidu.com/view/971216.htm" \t "http://baike.baidu.com/_blank)Mask:255.0.0.0

UP LOOPBACK RUNNING MTU:3924 Metric:1

RX packets:44 errors:0 dropped:0 overruns:0 frame:0

TX packets:44 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

其中以eth0 为首的部分是本机的[以太网](http://baike.baidu.com/view/848.htm" \t "http://baike.baidu.com/_blank)卡配置参数，这里显示了网卡的设备名/dev/eth0 和硬件的MAC 地址52:54:AB:DD:6F:61，MAC 地址是生产厂家定的，每个网卡拥有的唯一地址。

不过我们可以手工改动网卡的MAC 地址，只要我们在/etc/rc.d/init.d/中的network 中加入：

ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx

eth0，eth1,eth2,代表网卡一，网卡二，网卡三

hw 代表hardware 硬件意思

ether 代表ethernet[以太网](http://baike.baidu.com/view/848.htm" \t "http://baike.baidu.com/_blank)的意思

然后重启，此时再用ifconfig 命令查看一下，我们就会发现网卡的MAC 地址已经变成xx:xx:xx:xx:xx:xx了。

### 配置网卡

配置网卡的IP地址

ifconfig eth0 192.168.0.1 netmask 255.255.255.0

在eth0上配置上192.168.0.1 的IP地址及24位掩码。若想再在eth0上在配置一个192.168.1.1/24 的IP地址怎么办？用下面的命令

ifconfig eth0：0 192.168.1.1 netmask 255.255.255.0

这时再用ifconifg命令查看，就可以看到两个网卡的信息了，分别为：eth0和eth0：0.若还想再增加IP，那网卡的命名就接着是：eth0：1、eth0：2……想要几个就填几个。ok！

配置网卡的硬件地址

ifconfig eth0 hw ether xx：xx：xx：xx：xx：xx

就将网卡的硬件地址更改了，此时你就可以骗过局域网内的IP地址绑定了。

将网卡禁用

ifconfig eth0 down

将网卡启用

ifconfig eth0 up

ifconfig 命令的功能很强大，还可以设置网卡的MTU，混杂模式等。

## crontab

基本格式 :

\*　　\*　　\*　　\*　　\*　　command

分　时　日　月　周　命令

第1列表示分钟1～59 每分钟用\*或者 \*/1表示

第2列表示小时1～23（0表示0点）

第3列表示日期1～31

第4列表示月份1～12

第5列标识号星期0～6（0表示星期天）

第6列要运行的命令

crontab文件的一些例子：

30 21 \* \* \* /usr/local/etc/rc.d/lighttpd restart

上面的例子表示每晚的21:30重启apache。

45 4 1,10,22 \* \* /usr/local/etc/rc.d/lighttpd restart

上面的例子表示每月1、10、22日的4 : 45重启apache。

10 1 \* \* 6,0 /usr/local/etc/rc.d/lighttpd restart

上面的例子表示每周六、周日的1 : 10重启apache。

0,30 18-23 \* \* \* /usr/local/etc/rc.d/lighttpd restart

上面的例子表示在每天18 : 00至23 : 00之间每隔30分钟重启apache。

0 23 \* \* 6 /usr/local/etc/rc.d/lighttpd restart

上面的例子表示每星期六的11 : 00 pm重启apache。

\* \*/1 \* \* \* /usr/local/etc/rc.d/lighttpd restart

每一小时重启apache

\* 23-7/1 \* \* \* /usr/local/etc/rc.d/lighttpd restart

晚上11点到早上7点之间，每隔一小时重启apache

0 11 4 \* mon-wed /usr/local/etc/rc.d/lighttpd restart

每月的4号与每周一到周三的11点重启apache

0 4 1 jan \* /usr/local/etc/rc.d/lighttpd restart

一月一号的4点重启apache

名称 : crontab

使用权限 : 所有使用者

使用方式 :

crontab file [-u user]-用指定的文件替代目前的crontab。

crontab-[-u user]-用标准输入替代目前的crontab.

crontab-1[user]-列出用户目前的crontab.

crontab-e[user]-编辑用户目前的crontab.

crontab-d[user]-删除用户目前的crontab.

crontab-c dir- 指定crontab的目录。

crontab文件的格式：M H D m d cmd.

M: 分钟（0-59）。

H：小时（0-23）。

D：天（1-31）。

m: 月（1-12）。

d: 一星期内的天（0~6，0为星期天）。

cmd要运行的程序，程序被送入sh执行，这个shell只有USER,HOME,SHELL这三个环境变量

说明 :

crontab 是用来让使用者在固定时间或固定间隔执行程序之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定

user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设

定自己的时程表。

参数 :

crontab -e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数

来指定使用那个文字编辑器(比如说 setenv VISUAL joe)

crontab -r : 删除目前的时程表

crontab -l : 列出目前的时程表

crontab file [-u user]-用指定的文件替代目前的crontab。

时程表的格式如下 :

f1 f2 f3 f4 f5 program

其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执

行的程序。

当 f1 为 \* 时表示每分钟都要执行 program，f2 为 \* 时表示每小时都要执行程序，其馀类推

当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其馀类推

当 f1 为 \*/n 时表示每 n 分钟个时间间隔执行一次，f2 为 \*/n 表示每 n 小时个时间间隔执行一次，其馀类推

当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其馀类推

使用者也可以将所有的设定先存放在档案 file 中，用 crontab file 的方式来设定时程表。

例子 :

#每天早上7点执行一次 /bin/ls :

0 7 \* \* \* /bin/ls

在 12 月内, 每天的早上 6 点到 12 点中，每隔3个小时执行一次 /usr/bin/backup :

0 6-12/3 \* 12 \* /usr/bin/backup

周一到周五每天下午 5:00 寄一封信给 alex@domain.name :

0 17 \* \* 1-5 mail -s "hi" alex@domain.name < /tmp/maildata

每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"

20 0-23/2 \* \* \* echo "haha"

注意 :

当程序在你所指定的时间执行后，系统会寄一封信给你，显示该程序执行的内容，若是你不希望收到这样的信，请在每一行空一格之

后加上 > /dev/null 2>&1 即可

例子2 :

#每天早上6点10分

10 6 \* \* \* date

#每两个小时

0 \*/2 \* \* \* date

#晚上11点到早上8点之间每两个小时，早上8点

0 23-7/2，8 \* \* \* date

#每个月的4号和每个礼拜的礼拜一到礼拜三的早上11点

0 11 4 \* mon-wed date

#1月份日早上4点

0 4 1 jan \* date

范例

$crontab -l 列出用户目前的crontab.

crontab命令的功能是在一定的时间间隔调度一些命令的执行。在/etc目录下有一个crontab文件，这里存放有系统运行的一些调度程序。每个用户可以建立自己的调度crontab。

cron 的主配置文件是 /etc/crontab，它包括下面几行：

[SHELL](https://www.baidu.com/s?wd=SHELL&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)=/[bin](https://www.baidu.com/s?wd=bin&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)/bash

[PATH](https://www.baidu.com/s?wd=PATH&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)=/s[bin](https://www.baidu.com/s?wd=bin&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank):/[bin](https://www.baidu.com/s?wd=bin&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank):/usr/sbin:/usr/bin

[MAILTO](https://www.baidu.com/s?wd=MAILTO&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)=root

[HOME](https://www.baidu.com/s?wd=HOME&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)=/

# run-parts

01 \* \* \* \* root run-parts /etc/cron.hourly

02 4 \* \* \* root run-parts /etc/cron.daily

22 4 \* \* 0 root run-parts /etc/cron.weekly

42 4 1 \* \* root run-parts /etc/cron.monthly

前四行是用来配置 cron 任务运行环境的变量。[SHELL](https://www.baidu.com/s?wd=SHELL&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 变量的值告诉系统要使用哪个 [shell](https://www.baidu.com/s?wd=shell&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 环境（在这个例子里是 bash shell）；[PATH](https://www.baidu.com/s?wd=PATH&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 变量定义用来执行命令的路径。cron 任务的输出被邮寄给 [MAILTO](https://www.baidu.com/s?wd=MAILTO&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 变量定义的用户名。如果 [MAILTO](https://www.baidu.com/s?wd=MAILTO&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 变量被定义为空白字符串（MAILTO=""），电子邮件就不会被寄出。[HOME](https://www.baidu.com/s?wd=HOME&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank) 变量可以用来设置在执行命令或脚本时使用的主目录。

/etc/crontab 文件中的每一行都代表一项任务，它的格式是：

**minute hour day month dayofweek command**

minute — 分钟，从 0 到 59 之间的任何整数

hour — 小时，从 0 到 23 之间的任何整数

day — 日期，从 1 到 31 之间的任何整数（如果指定了月份，必须是该月份的有效日期）

month — 月份，从 1 到 12 之间的任何整数（或使用月份的英文简写如 jan、feb 等等）

dayofweek — 星期，从 0 到 7 之间的任何整数，这里的 0 或 7 代表星期日（或使用星期的英文简写如 sun、mon 等等）

command — 要执行的命令（命令可以是 ls /proc >> /tmp/proc 之类的命令，也可以是执行你自行编写的脚本的命令。）

在以上任何值中，星号（\*）可以用来代表所有有效的值。譬如，月份值中的星号意味着在满足其它制约条件后每月都执行该命令。

整数间的短线（-）指定一个整数范围。譬如，1-4 意味着整数 1、2、3、4。

用逗号（,）隔开的一系列值指定一个列表。譬如，3, 4, 6, 8 标明这四个指定的整数。

正斜线（/）可以用来指定间隔频率。在范围后加上 / 意味着在范围内可以跳过 integer。譬如，0-59/2 可以用来在分钟字段定义每两分钟。间隔频率值还可以和星号一起使用。例如，\*/3 的值可以用在月份字段中表示每三个月运行一次任务。

开头为井号（#）的行是注释，不会被处理。

如你在 /etc/crontab 文件中所见，它使用 run-parts 脚本来执行 /etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly 和 /etc/cron.monthly 目录中的脚本，这些脚本被相应地每小时、每日、每周、或每月执行。这些目录中的文件应该是 shell 脚本。

如果某 cron 任务需要根据调度来执行，而不是每小时、每日、每周、或每月地执行，它可以被添加到 /etc/cron.d 目录中。该目录中的所有文件使用和 /etc/crontab 中一样的语法。

# record the memory usage of the system every monday

# at 3:30AM in the file /tmp/meminfo 30 3 \* \* mon cat /proc/meminfo >> /tmp/meminfo

# run custom script the first day of every month at 4:10AM 10 4 1 \* \* /root/scripts/backup.sh

同时在写crontab执行的.sh脚本时要注意添加权限和路径。例如：

crontab语句：30 14 \* \* \* /opt/apps/ems\_query/bin/start.sh

相应的脚本：

source /[home](https://www.baidu.com/s?wd=home&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHfzPHTLuH6zPWT1uywb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6K1TL0qnfK1TL0z5HD0IgF_5y9YIZ0lQzqlpA-bmyt8mh7GuZR8mvqVQL7dugPYpyq8Q16vPWRznHcL" \t "http://zhidao.baidu.com/_blank)/tvgame/.bash\_profile

cd /opt/apps/ems\_query/bin/

java -Xmx64M -jar ../lib/ems\_query.jar &

## mount

mount准确来说应该是挂载的意思.

就是把相应的设备挂载到相关的文件系统下进行访问和控制.

例如mount /abc /mnt

就是把/abc目录挂载到/mnt目录下进行访问和控制

mount 后可以接一些参数来进行额外的调整.

比如mount -t iso9660 -o ro /dev/cdrom /mnt

这条指令是把光盘介质以只读的状态挂载到/mnt下来.

-t 挂载介质类型,具体有iso9660,nfs等.

-o 挂载的选项,例如ro(Read Only),rw(Read & Write),remount(重新挂载,一般用于挂载需要进行读写的只读的目录或者设备)等等.

## gedit

gedit是一个GNOME桌面环境下兼容UTF-8的文本编辑器。

命令 gedit

激活环境变量，使配置文件生效

source /etc/profile

新增组：sudo addgroup groupName

新增用户：sudo adduser -ingroup groupName userName

增加root权限：sudo gedit /etc/sudoers

# User privilege specification

root ALL=(ALL:ALL) ALL

hadoop ALL=(ALL:ALL) ALL

## apt-get

apt-get是一条linux命令，适用于deb包管理式的操作系统，主要用于自动从互联网的软件仓库中搜索、安装、升级、卸载软件或操作系统。

Advanced Package Tool，又名apt-get，是一款适用于Unix和Linux系统的应用程序管理器。

## RPM

RPM[1]  是RedHat Package Manager（RPM软件包管理器）的缩写，这一文件格式名称虽然打上了RedHat的标志，但是其原始设计理念是开放式的，现在包括OpenLinux、S.u.S.E.以及Turbo Linux等Linux的分发版本都有采用，可以算是公认的行业标准了。

RPM软件包管理器

英文原义：RPM Package Manager（原Red Hat Package Manager，现在是一个[递归缩写](http://baike.baidu.com/view/5034635.htm" \t "http://baike.baidu.com/_blank)）

在Terminal中，基本的安装指令如下：

rpm －i xv－3.10a－13.i386.rpm

如果你的连网速度足够快，也可以直接从网络上安装应用软件，只需要在软件的文件名前加上适当的[URL](http://baike.baidu.com/view/1496.htm" \t "http://baike.baidu.com/_blank)路径。

作为一个软件包管理工具，RPM管理着系统已安装的所有RPM程序组件的资料。我们也可以使用RPM来卸载相关的应用程序。

rpm －e xv

RPM的常用参数还包括：

－vh：显示安装进度；

－U：升级软件包；

－qpl：列出RPM软件包内的文件信息；

－qpi：列出RPM软件包的描述信息；

－qf：查找指定文件属于哪个RPM软件包；

－Va：校验所有的RPM软件包，查找丢失的文件；

－qa: 查找相应文件，如 rpm -qa mysql

### RPM主要功能

安装、[卸载](http://baike.baidu.com/view/386432.htm" \t "http://baike.baidu.com/_blank)、升级和管理软件

组件查询功能

验证功能

软件包GPG和MD5数字签名的导入、验证和发布

软件包依赖处理

选择安装

网络远程安装功能

rpm 命令：遵循GPL协议且功能强大的包管理，它可以建立、安装、请求、确认、和卸载软件包。间接的提升了Linux 的易用性

-e 卸载rpm包

-q 查询已安装的软件信息

-i 安装rpm包

-u 升级rpm包

--replacepkgs 重新安装rpm包

--justdb 升级数据库，不修改文件系统

--percent 在软件包安装时输出百分比

--help 帮助

--version 显示版本信息

-c 显示所有配置文件

-d 显示所有文档文件

-h 显示安装进度

-l 列出软件包中的文件

-a 显示出文件状态

-p 查询/校验一个软件包文件

-v 显示详细的处理信息

--dump 显示基本文件信息

--nomd5 不验证文件的md5支持

--nofiles 不验证软件包中的文件

--nodeps 不验证软件包的依赖关系

--whatrequires 查询/验证需要一个依赖性的软件包

--whatprovides 查询/验证提供一个依赖性的软件包

# SSH

## [在Ubuntu 12.04安装和设置SSH服务](http://blog.chinaunix.net/uid-20791108-id-3761681.html)

### 安装

更新apt-get

sudo apt-get update

安装ssh-server

sudo apt-get install openssh-server

安装ssh-client

sudo apt-get install openssh-client

### 确认sshserver是否安装好

ps -e | grep sshd

  450 ?        00:00:00 sshd

如果看到sshd那说明ssh-server已经启动了。

如果只有ssh-agent说明ssh-server还没有启动，需要执行命令启动ssh服务：

/etc/init.d/ssh start；

注：在ubuntu-12.04-server-i386.iso安装中只显示sshd这一项：

### 扩展配置

1. SSH默认服务端口为22，用户可以自已定义成其他端口，如222，需要修改的配置文件为：
2. /etc/ssh/sshd\_config
3. 把里面的Port参数修改成222即可
4. 然后重启SSH服务：
5. sudo/etc/init.d/ssh restart

## Linux ssh设置免密码登录

执行：**ssh-keygen -t rsa**

这时.ssh目录下会生成私钥和公钥的键值对id\_rsa，id\_rsa.pub

再其他服务器上执行同样的命令生成文件

将所有服务器上的公钥合并到一个文件authorized\_keys中

**cat id\_rsa.pub >> authorized\_keys**

**ssh root@s205 cat /root/.ssh/id\_rsa.pub >> authorized\_keys**

查看authorized\_keys里面有多个server的公钥信息，如果你有多个server同样往里面加就好了

将authorized\_keys、known\_hosts分别复制到其他服务器的.ssh目录下

**scp authorized\_keys root@s205:/root/.ssh/**

**scp known\_hosts root@s205:/root/.ssh/**

这时候再ssh登录，这时候就不用再输入密码啦。

$ ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa $ cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys $ export HADOOP\\_PREFIX=/usr/local/hadoop

或

#ssh-keygen -t rsa

#cat /home/hadoop/.ssh/id\_rsa.pub >> /home/hadoop/.ssh/authorized\_keys

#ssh-copy-id -i id\_rsa.pub hadoop@hyname01（将公匙copy到hyname01上）

#ssh hyname01（测试SSH无密码登入是否成功）

## SCP

scp [-1246BCpqrv] [-c cipher] [-F ssh\_config] [-i identity\_file] [-l limit] [-o ssh\_option] [-P port] [-S program] [[user@]host1:]file1 [...] [[user@]host2:]file2

最简单的应用如下 :

scp localUserName@local\_IP:file\_name1 remote\_username@remote\_IP:file\_name2

[ 本地用户名 @IP 地址 :] 可以不输入 , 可能需要输入远程用户名所对应的密码 .

可能有用的几个参数 :

-v 和大多数 linux 命令中的 -v 意思一样 , 用来显示进度 . 可以用来查看连接 , 认证 , 或是配置错误 .

-C 使能[压缩](http://www.linuxso.com/linuxpeixun/12737.html" \t "http://blog.csdn.net/liangxanhai/article/details/_blank)选项 .

-P 选择端口 . 注意 -p 已经被 [rcp](http://www.linuxso.com/command/rcp.html" \t "http://blog.csdn.net/liangxanhai/article/details/_blank) 使用 .

-4 强行使用 IPV4 地址 .

-6 强行使用 IPV6 地址 .

Linux scp命令用于Linux之间复制文件和目录，具体如何使用这里好好介绍一下，从本地复制到远程、从远程复制到本地是两种使用方式。这里有具体举例：

==================

**Linux scp 命令**

==================

scp 可以在 2个 linux 主机间复制文件；

命令基本格式：

scp [可选参数] [file](http://www.linuxso.com/command/file.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank)\_source file\_target

======

**从 本地 复制到 远程**

======

\* 复制文件：

\* 命令格式：

scp local\_file remote\_username@remote\_ip:remote\_[fold](http://www.linuxso.com/command/fold.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank)er

或者

scp local\_file remote\_username@remote\_ip:remote\_file

或者

scp local\_file remote\_ip:remote\_folder

或者

scp local\_file remote\_ip:remote\_file

第1,2个指定了用户名，命令执行后需要再输入密码，第1个仅指定了远程的目录，文件名字不变，第2个指定了文件名；

第3,4个没有指定用户名，命令执行后需要输入用户名和密码，第3个仅指定了远程的目录，文件名字不变，第4个指定了文件名；

\* 例子：

scp /home/space/music/1.mp3 root@www.cumt.e[du](http://www.linuxso.com/command/du.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank).cn:/home/root/others/music

scp /home/space/music/1.mp3 root@www.cumt.edu.cn:/home/root/others/music/001.mp3

scp /home/space/music/1.mp3 www.cumt.edu.cn:/home/root/others/music

scp /home/space/music/1.mp3 www.cumt.edu.cn:/home/root/others/music/001.mp3

\* 复制目录：

\* 命令格式：

scp -r local\_folder remote\_username@remote\_ip:remote\_folder

或者

scp -r local\_folder remote\_ip:remote\_folder

第1个指定了用户名，命令执行后需要再输入密码；

第2个没有指定用户名，命令执行后需要输入用户名和密码；

\* 例子：

scp -r /home/space/music/ root@www.cumt.edu.cn:/home/root/others/

scp -r /home/space/music/ www.cumt.edu.cn:/home/root/others/

上面 命令 将 本地 music 目录 复制 到 远程 others 目录下，即复制后有 远程 有 ../others/music/ 目录

======

**从 远程 复制到 本地**

======

从 远程 复制到 本地，只要将 从 本地 复制到 远程 的命令 的 后2个参数 调换顺序 即可；

例如：

scp root@www.cumt.edu.cn:/home/root/others/music /home/space/music/1.mp3

scp -r www.cumt.edu.cn:/home/root/others/ /home/space/music/

最简单的应用如下 :

scp 本地用户名 @IP 地址 : 文件名 1 远程用户名 @IP 地址 : 文件名 2

[ 本地用户名 @IP 地址 :] 可以不输入 , 可能需要输入远程用户名所对应的密码 .

可能有用的几个参数 :

-v 和大多数 linux 命令中的 -v 意思一样 , 用来显示进度 . 可以用来查看连接 , 认证 , 或是配置错误 .

-C 使能[压缩](http://www.linuxso.com/linuxpeixun/12737.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank)选项 .

-P 选择端口 . 注意 -p 已经被 [rcp](http://www.linuxso.com/command/rcp.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank) 使用 .

-4 强行使用 IPV4 地址 .

-6 强行使用 IPV6 地址 .

Linux scp命令的使用方法应该可以满足大家对Linux文件和目录的复制使用了。

**关于scp的安全方面**

copy 本地的档案到远程的机器上

scp /etc/[lilo](http://www.linuxso.com/command/lilo.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank).conf k@net67.ee.oit.edu.tw:/home/k

会将本地的 /etc/lilo.conf 这个档案 copy 到 net67.ee.oit.edu.tw，使用者 k 的家目录下。

copy远程机器上的档案到本地来

scp k@net67.ee.oit.edu.tw:/etc/lilo.conf /etc

会将 net67.ee.oitdu.tw 中 /etc/lilo.conf 档案 copy 到本地的 /etc 目录下。

保持从来源 host 档案的属性

scp –p k@net67.ee.tw:/etc/lilo.conf /etc

如果想使用特定端口 使用 scp –p（大写） 如 scp –p 1234 k@net67.ee.tw:/etc/lilo.conf /etc

在此必须注意使用者的权限是否可读取远程上的档案，若想知道更多关于 scp 的使用方法，可去看看 scp 的使用手册。

ssh-keygen

产生公开钥 (pulib key) 和私人钥 (private key)，以保障 ssh 联机的安性， 当 ssh 连 shd 服务器，会交换公开钥上，系统会检查 /etc/ssh\_know\_hosts 内储存的 key，如果找到客户端就用这个 key 产生一个随机产生的session key 传给服务器，两端都用这个 key 来继续完成 ssh 剩下来的阶段。

它会产生 [id](http://www.linuxso.com/command/id.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank)entity.pub、identity 两个档案，私人钥存放于identity，公开钥 存放于 identity.pub 中，接下来使用 scp 将 identity.pub copy 到远程机器的家目录下.ssh下的authorized\_keys。 .ssh/authorized\_keys(这个 authorized\_keys 档案相当于协议的 rhosts 档案)， 之后使用者能够不用密码去登入。RSA的认证绝对是比 rhosts 认证更来的安全可靠。

执行：

scp identity.pub k@linux1.ee.oit.edu.tw:.ssh/authorized\_keys

若在使用 ssh-keygen 产生钥匙对时没有输入密码，则如上所示不需输入密码即可从 net67.ee.oit.edu.tw 去登入 linux1.ee.oit.edu.tw。在此，这里输入的密码可以跟帐号的密码不同，也可以不输入密码。

**scp命令的一个实例**

Linux下scp的用法

scp就是secure copy，一个在linux下用来进行远程拷贝文件的命令。

有时我们需要获得远程服务器上的某个文件，该服务器既没有配置ftp服务器，也没有做共享，无法通过常规途径获得文件时，只需要通过简单的scp命令便可达到目的。

一、将本机文件复制到远程服务器上

#scp /home/adminis[tr](http://www.linuxso.com/command/tr.html" \t "http://blog.csdn.net/jiangkai_nju/article/details/_blank)ator/news.txt root@192.168.6.129:/etc/squid

/home/administrator/ 本地文件的绝对路径

news.txt 要复制到服务器上的本地文件

root 通过root用户登录到远程服务器（也可以使用其他拥有同等权限的用户）

192.168.6.129 远程服务器的ip地址（也可以使用域名或机器名）

/etc/squid 将本地文件复制到位于远程服务器上的路径

二、将远程服务器上的文件复制到本机

#scp remote@www.abc.com:/usr/local/sin.sh /home/administrator

remote 通过remote用户登录到远程服务器（也可以使用其他拥有同等权限的用户）

www.abc.com 远程服务器的域名（当然也可以使用该服务器ip地址）

/usr/local/sin.sh 欲复制到本机的位于远程服务器上的文件

/home/administrator 将远程文件复制到本地的绝对路径

注意两点：

1.如果远程服务器防火墙有特殊限制，scp便要走特殊端口，具体用什么端口视情况而定，命令格式如下：

#scp -p 4588 remote@www.abc.com:/usr/local/sin.sh /home/administrator

2.使用scp要注意所使用的用户是否具有可读取远程服务器相应文件的权限。

# 主机名

查看主机名

在Ubuntu系统中，快速查看主机名有多种方法：

其一，打开一个GNOME终端窗口，在命令提示符中可以看到主机名，主机名通常位于“@”符号后；

其二，在终端窗口中输入命令：hostname或uname –n，均可以查看到当前主机的主机名。

2、临时修改主机名

命令行下运行命令：“hostname 新主机名”

其中“新主机名”可以用任何合法字符串来表示。不过采用这种方式，新主机名并不保存在系统中，重启系统后主机名将恢复为原先的主机名称。

例子：hostname ubuntu-temp

这样主机名字就临时被修改为ubuntu-temp，但是终端下不会立即显示生效后的主机名，重开一个终端窗口(通过ssh连接的终端需要重新连接才可以);

3、永久修改主机名

在Ubuntu系统中永久修改主机名也比较简单。主机名存放在/etc/hostname文件中，修改主机名时，编辑hostname文件，在文件中输入新的主机名并保存该文件即可。重启系统后，参照上面介绍的快速查看主机名的办法来确认主机名有没有修改成功。

值的指出的是，在其它Linux发行版中，并非都存在/etc/hostname文件。如Fedora发行版将主机名存放在/etc/sysconfig/network文件中。所以，修改主机名时应注意区分是哪种Linux发行版。

/etc/hostname与/etc/hosts的区别

/etc/hostname中存放的是主机名，hostname文件的一个例子：

v-jiwan-ubuntu-temp

/etc/hosts存放的是域名与ip的对应关系，域名与主机名没有任何关系，你可以为任何一个IP指定任意一个名字，hostname文件的一个例子：

127.0.0.1       localhost

127.0.1.1       v-jiwan-ubuntu

1. 使用pgrep快速查找PID

　　pgrep会遍历当前运行的进程，列出符合查找条件的进程ID。

pgrep ssh

　　这条命令会列出所有与ssh有关的进程ID。

　　2. 执行上次执行过的命令

　　这个标题有些绕口，但下面这条命令的确能做到这一点。

!!

　　该命令会执行你在命令行中执行过的上一条命令。

　　3. 执行最近一次以特定字母开头的命令

　　如欲执行命令行历史中一个s开头的命令，可以使用如下命令：

!s

　　该命令会执行最近一次在命令行中用过的且以字母s开头的命令。

　　4. 反复执行一个命令并在屏幕上输出

　　watch会反复执行一个命令，并在屏幕上打印输出。你可以借此观察一段时间内程序的输出变化。程序（即示例中的ls -l）默认每2秒运行一次.watch命令与tail命令非常相似。

watch -d ls -l

　　这条命令会监视当前目录，如有文件增删或修改，就会高亮显示目录变化。

　　5. 在VI/VIM中快速保存

　　要忙着办其他事情时，可以使用Shift + zz（即按住换档键，敲两下z键）快速退出vi插入模式。

　　6. 快速退出终端会话

　　使用CTRL+D可以快速退出终端会话。

　　7. 返回上一次所在的目录

　　使用命令cd -可以返回上一次所在的目录：

　　8.

　　9. 删除一整行

　　输入了一长串命令但又不想要了，可以使用CTRL+U删除这一整行。（译注：光标不在行尾时，可以先用CTRL+E定位到行尾。）

　　10. 设定文件的时间戳

　　命令touch -c -t 0801010800 filename.c会将文件的时间戳设定为2008-01-01 8:00，格式为(YYMMDDhhmm)。

　　你还能想到哪些不为人熟知的Linux命令？

# Ubuntu相关

Ubuntu打开终端的方法: Ctrl+Alt+T，F11切换终端的全屏和窗口

## 目录说明

|  |  |
| --- | --- |
| / | 目录属于根目录，是所有目录的绝对路径的起始点，Ubuntu 中的所有文件和目录都在跟目录下。 |
| /etc | 此目录非常重要，绝大多数系统和相关服务的配置文件都保存在这里，这个目录的内容一般只能由管理员进行修改。像密码文件、设置网卡信息、环境变量的设置等都在此目录中。此目录的 rcn.d 目录中存放不同启动级别所启动的服务，network 目录放置网卡的配置信息等。 |
| /home | 此目录是所有普通用户的宿主目录所在地，在一般情况下，如果想要对用户进行磁盘限额功能，最好将此目录单独分区。 |
| /bin | 此目录中放置了所有用户能够执行的命令。 |
| /sbin | 此目录中放置了一般是只有系统管理有才能执行的命令。 |
| /dev | 此目录中保存了所有设备文件，例如，使用的分区：/dev/hda\1，/dev/cdrom 等。 |
| /mnt | 此目录主要是作为挂载点使用。 |
| /usr | 此目录包含了所有的命令、说明文件、程序库等，此目录下有很多重要的目录，常见的有：/usr/local 这个目录包含管理员自己安装的程序；/usr/share 包含文件的帮助文件；/usr/bin 和/usr/sbin 包含了所有的命令。 |
| /var | 包含了日志文件、计划性任务和邮件等内容。 |
| /lib | 包含了系统的函数库文件。 |
| /lost+found | 包含了系统修复时的回复文件。 |
| /tmp | 包含了临时的文件。 |
| /boot | 系统的内核所在地，也是启动分区。 |
| /media | 主要用于挂载多媒体设备。 |
| /root | 系统管理员的宿主目录。 |

# 杂记

## $ man jps

jps(1) jps(1)

Name

jps - Java Virtual Machine Process Status Tool

SYNOPSIS

jps [ options ] [ hostid ]

PARAMETERS

options

Command-line options.

hostid

The host identifier of the host for which the process report should be generated. The hostid may include optional components that indicate the communications

protocol, port number, and other implementation specific data.

DESCRIPTION

The jps tool lists the instrumented HotSpot Java Virtual Machines (JVMs) on the target system. The tool is limited to reporting information on JVMs for which it has

the access permissions.

If jps is run without specifying a hostid, it will look for instrumented JVMs on the local host. If started with a hostid, it will look for JVMs on the indicated

host, using the specified protocol and port. A jstatd process is assumed to be running on the target host.

The jps command will report the local VM identifier, or lvmid, for each instrumented JVM found on the target system. The lvmid is typically, but not necessarily,

the operating system’s process identifier for the JVM process. With no options, jps will list each Java application’s lvmid followed by the short form of the appli-

cation’s class name or jar file name. The short form of the class name or JAR file name omits the class’s package information or the JAR files path information.

The jps command uses the java launcher to find the class name and arguments passed to the main method. If the target JVM is started with a custom launcher, the

class name (or JAR file name) and the arguments to the main method will not be available. In this case, the jps command will output the string Unknown for the class

name or JAR file name and for the arguments to the main method.

The list of JVMs produced by the jps command may be limited by the permissions granted to the principal running the command. The command will only list the JVMs for

which the principle has access rights as determined by operating system specific access control mechanisms.

NOTE: This utility is unsupported and may not be available in future versions of the JDK. It is not currently available on Windows 98 and Windows ME platforms.

OPTIONS

The jps command supports a number of options that modify the output of the command. These options are subject to change or removal in the future.

-q Suppress the output of the class name, JAR file name, and arguments passed to the main method, producing only a list of local VM identifiers.

-m Output the arguments passed to the main method. The output may be null for embedded JVMs.

-l Output the full package name for the application’s main class or the full path name to the application’s JAR file.

-v Output the arguments passed to the JVM.

-V Output the arguments passed to the JVM through the flags file (the .hotspotrc file or the file specified by the -XX:Flags=<filename> argument).

-Joption

Pass option to the java launcher called by jps. For example, -J-Xms48m sets the startup memory to 48 megabytes. It is a common convention for -J to pass

options to the underlying VM executing applications written in Java.

HOST IDENTIFIER

The host identifier, or hostid is a string that indicates the target system. The syntax of the hostid string largely corresponds to the syntax of a URI:

[protocol:][[//]hostname][:port][/servername]

protocol

The communications protocol. If the protocol is omitted and a hostname is not specified, the default protocol is a platform specific, optimized, local proto-

col. If the protocol is omitted and a hostname is specified, then the default protocol is rmi.

hostname

A hostname or IP address indicating the target host. If hostname is omitted, then the target host is the local host.

port

The default port for communicating with the remote server. If the hostname is omitted or the protocol specifies an optimized, local protocol, then port is

ignored. Otherwise, treatment of the port parameter is implementation specific. For the default rmi protocol the port indicates the port number for the

rmiregistry on the remote host. If port is omitted, and protocol indicates rmi, then the default rmiregistry port (1099) is used.

servername

The treatment of this parameter depends on the implementation. For the optimized, local protocol, this field is ignored. For the rmi protocol, this parameter

is a string representing the name of the RMI remote object on the remote host. See the -n option for the jstatd(1) command.

OUTPUT FORMAT

The output of the jps command follows the following pattern:

lvmid [ [ classname | JARfilename | "Unknown"] [ arg\* ] [ jvmarg\* ] ]

Where all output tokens are separated by white space. An arg that includes embedded white space will introduce ambiguity when attempting to map arguments to their

actual positional parameters.

NOTE: You are advised not to write scripts to parse jps output since the format may change in future releases. If you choose to write scripts that parse jps output,

expect to modify them for future releases of this tool.

EXAMPLES

This section provides examples of the jps command.

Listing the instrumented JVMs on the local host:

jps

18027 Java2Demo.JAR

18032 jps

18005 jstat

Listing the instrumented JVMs on a remote host:

This example assumes that the jstat server and either the its internal RMI registry or a separate external rmiregistry process are running on the remote host on the

default port (port 1099). It also assumes that the local host has appropriate permissions to access the remote host. This example also includes the -l option to

output the long form of the class names or JAR file names.

jps -l remote.domain

3002 /opt/jdk1.7.0/demo/jfc/Java2D/Java2Demo.JAR

2857 sun.tools.jstatd.jstatd

Listing the instrumented JVMs on a remote host with a non-default port for the RMI registry

This example assumes that the jstatd server, with an internal RMI registry bound to port 2002, is running on the remote host. This example also uses the -m option

to include the arguments passed to the main method of each of the listed Java applications.

jps -m remote.domain:2002

3002 /opt/jdk1.7.0/demo/jfc/Java2D/Java2Demo.JAR

3102 sun.tools.jstatd.jstatd -p 2002

SEE ALSO

o java(1) - the Java Application Launcher

o jstat(1) - the Java virtual machine Statistics Monitoring Tool

o jstatd(1) - the jstat daemon

o rmiregistry(1) - the Java Remote Object Registry

18 Jul 2013 jps(1)

Java环境变量，编辑/etc/profile，末尾加如下配置：

空间不足

chfs -a size=+1G /tmp

chfs -a size=+2G /

lsvg rootvg

ps -ealf | head -1 ; ps -ealf | sort -rn +9 | head

ps aux | head -1 ; ps aux | sort -rn +3 | head

以进程占用内存量从大到小排列，SIZE单位为k。

nmon c、m、D CPU/内存/硬盘

prtconf 查看系统配置

vmstat 1 每隔一秒监控内存

topas 查看cpu

cp /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/servers/AdminServer/security/boot.properties /crm/app/Oracle/Middleware/user\_projects/domains/espdomain/servers/mac3\_Server/security/boot.properties

du /crm/ | sort -nr | more 查看文件夹占用空间情况

/crm/app/Oracle/Middleware/user\_projects/domains/espdomain/servers/Admim/stage/mac3

1. ps -ef|grep XXX 查看进程

7. groupadd XXX 建立用户组

8. useradd username 建立用户

useradd -g 用户组 -d /fitp/bank（该用户登录后默认进入的路径） 路径名

9. passwd 密码 设置密码

10.unzip xxxx.zip 解压缩zip文件

kill -9 进程号

tvf 查看内容

jar -xvf resource.jar(解压);jar -cvf resource.jar(压缩)

12.groups 查看当前登录用户的组内成员

groups gliethttp 查看gliethttp用户所在的组,以及组内成员

whoami 查看当前登录用户名

/etc/group 文件包含所有组

/etc/shadow和/etc/passwd 系统存在的所有用户名

13. df -g (字节) 查看当前资源分配

df -m (兆) 查看当前资源分配

14.chown -R 777 weblogic:dba /crm/app 赋权限给用户(macetl用户名 etl用户组)

chmod -R 744 /weblogic

15.whoami 查看当前用户名

chown -R weblogic:dba /crm/app

chmod -R 744 /weblogic

nohup /weblogic/domains/base\_domain/bin/startWebLogic.sh >> /weblogic/domains/base\_domain/nohup.log&

nohup /weblogic/domains/base\_domain/bin/startManagedWebLogic.sh MACServer >> /weblogic/domains/base\_domain/bin/nohup.log&

nohup /weblogic/domains/mac\_domain/bin/startWebLogic.sh >> /weblogic/domains/mac\_domain/nohup.log&

df -k 查看空间

topas 查看进程

lsvg rootvg

更改时区

利用smitty工具

smit date 更改时间

查看SWAP，使用lsps –a命令查看,默认安装SWAP是512M,例如：

# lsps -a

Page Space Physical Volume Volume Group Size %Used Active Auto Type

hd6 hdisk0 rootvg 512MB 1 yes yes lv

增加SWAP，chps -s number hd6 ,例如：

# chps -s 28 hd6 ----20表示以上面pp\*28，计算如下：pp=128M PP\*28=3584M

/usr/java6/bin/java -Xms512m -Xmx512m

单独增加JVM 方法： 登入console->服务器->点击bam\_server01->配置->服务器启动(start)

拉到最下面有 Arguments（参数）项， 需要点击导航部分的“锁定并编辑”才能增加，

可以写入参数部分的比如：

-Xms2048m -Xmx4096m -XX:PermSize=512m -XX:MaxPermSize=1024m

clear命令

clear命令的功能是清除屏幕上的信息，它类似于DOS中的 cls命令。清屏后，提示符移动到屏幕左上角。

unzip \*.zip 解压zip包

topas(AIX) ：查看AIX系统的CPU

top(HP) ：查看HP系统的CPU

df –sg :查看文件所占大小

history 200 >> ./a.txt ：将最近的200条命令保存到a.txt

ps auxww|grep ftsapp :查进程的CPU以及内存等信息

uname –a/i :获取机器码

ulimit –a : 写文件帐号的文件大小限制

lslpp -lc|grep -i "aix.\*c++" :

lslpp -Lc|grep libpthreads :

tail -f 日志文件 看到动态的输出

id cdadmin :查看cdadmin用户信息

du –s:

env :是显示环境变量，

export :是定义了一个环境变量之后向系统提交。

export单独用和env一样

exports是设置NFS共享文件的

Linux export命令

　　功能说明：设置或显示环境变量。

　　语　法：export [-fnp][变量名称]=[变量设置值]

　　补充说明：在shell中执行程序时，shell会提供一组环境变量。export可新增，修改或删除环境变量，供后续执行的程序使用。export的效力仅及于该此登录操作

-f 代表[变量名称]中为函数名称。

　　-n 删除指定的变量。变量实际上并未删除，只是不会输出到后续指令的执行环境中。

　　-p 列出所有的shell赋予程序的环境变量。

　　一个变量创建时，它不会自动地为在它之后创建的shell进程所知。而命令export可以向后面的shell传递变量的值。当一个shell脚本调用并执行时，它不会自动得到原为脚本（调用者）里定义的变量的访问权，除非这些变量已经被显式地设置为可用。export命令可以用于传递一个或多个变量的值到任何后继脚本。

netstat –an | grep 327,确认32790,32792端口已经打开

lslpp -lc|grep -i "aix.\*c++"

lslpp -Lc|grep libpthreads

shutdown -r 服务器重启

. ./.profile