UNIX Commands:

Unix commands are entered at the shell prompt. They should be followed by the RETURN key. There are many more unix commands than are listed here. If you intend to work extensively with Unix, you should seek more information.

There is on-line documentation for most unix commands. To read these manuals, type: man [section number] command This will provide information on the man command itself. man man Some man files have been stored in the directory for the language or system. Use “man -M path command”.

Korn shell.

The Korn shell (see ksh(1)) our standard login shell is “upwardly compatible” with the Bourne shell (see sh(1)). Most Bourne shell scripts will work under ksh.

Bash and tcsh are additional shells which can be used after login.

It is the Korn shell command programming language that executes commands read from a terminal or a file.

A pipeline is a sequence of one or more commands separated by ‘|’ The standard output of each command excluding the last is connected to the standard input of the next command. Each command is run as a separate process; the shell waits for the last command to terminate. The exit status of a pipeline is the exit status of the last command.

time pipeline returns the real, user and system time used by the pipeline.

exit or ‘ ^D ‘ Terminates the shell.

User may enter and edit commands using one of the two following modes:

Parameter expansion

${parameter#pattern} Remove small left pattern
${parameter##pattern} Remove large left pattern
${parameter%pattern} Remove small right pattern
${parameter%%pattern} Remove large right pattern

The result is unspecified when parameter is @, *, or an array variable with subscript @.

To rename all files in the current directory ending in ".htm" to end in .html

```
for i in $( ls *\.htm); do
        mv  ${i} ${i%\.htm}.html
done
```

Vi Editing Mode

There are two typing modes. Initially, when you enter a command you are in the input mode. To edit, the user enters control mode by typing Escape (ESC, (octal 033)) and moves the cursor (**Not with the mouse !**) to the point needing correction and then inserts or deletes characters or words as needed. Most control commands accept an optional repeat count prior to the command.

Input Edit Commands

By default the editor is command mode. Commands used are the same as those for VI Editor (section below).

These commands access your command history.

[count]**k** Fetch previous command. Each time k is entered the previous command back in time is accessed.

[count]**-** Equivalent to k.

[count]**j** Fetch next command. Each time j is entered the next command forward in time is accessed.

[count]**+** Equivalent to j.

[count]**G** The command number count is fetched. The default is the earliest history command.

/string Search backward through history for a previous command containing string. String is terminated by a RETURN or NEW LINE . If string is null the previous string will be used.

?string Same as / except that search will be in the forward direction.

n Search for next match of the last pattern to / or ? commands.

N Search for next match of the last pattern to / or ? , but in reverse direction. Search history for the string entered by the previous / command.

Text Modification Edit Commands

These commands are the same as those for VI Editor.

Miscellaneous commands.

u Undo the last text modifying command.

U Undo all the text modifying commands performed on the line.

Return executes the edited command line, ^C aborts and displays a new prompt.

Emacs Editing Mode

This mode is entered by enabling either the emacs or gmacs option. The only difference between these two modes is the way they handle ^T. To edit, the user moves the cursor to the point needing correction and then inserts or deletes characters or words as needed. All the editing commands are control characters or escape sequences. The notation for control characters is caret (^) followed by the character. For example, ^F is the notation for control F. This is entered by pressing CONTROL and f ,the SHIFT key is not pressed. (The notation ^? indicates the DEL (delete) key.) The notation for escape sequences is M- followed by a character. For example, M-f (pronounced Meta f) is entered by depressing ESC (octal 033) followed by ‘ f M-F would be the notation for ESC followed by ‘ SHIFT’ (capital) ‘ F All edit commands operate from any place on the line (not just at the beginning). Neither the “ RETURN “ nor the “ LINE FEED” key is entered after edit commands except when noted.

^F Move cursor forward (right) one character.

M-f Move cursor forward one word. (The editor’s idea of a word is a string of characters consisting of only letters, digits and underscores.)

^B Move cursor backward (left) one character.

M-b Move cursor backward one word.
^A Move cursor to start of line.
^E Move cursor to end of line.
^]char Move cursor to character char on current line.

Commonly used commands:

ls

ls - list the contents of a directory

Syntax ls [-(options)] filename ...

Also our local aliases l == /bin/ls -l and lf == /bin/ls -F

Description For each filename which is a directory, ls lists the contents of the directory; for each filename which is a file, ls repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.

Permissions Field

The mode printed under the -l option contains 10 characters interpreted as follows. If the first character is:

d entry is a directory;
b entry is a block-type special file;
c entry is a character-type special file;
l entry is a symbolic link;
p entry is a FIFO (also known as “named pipe”) special file;
s entry is an UNIX address family socket
- entry is a plain file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to owner permissions; the next refers to permissions to others in the same user-group; and the last refers to all others. Within each set the three characters indicate permission respectively to read, to write, or to execute the file as a program. For a directory, “execute” permission is interpreted to mean permission to search the directory. The permissions are indicated as follows:

r the file is readable;
w the file is writable;
x the file is executable;
- the indicated permission is not granted.

The group-execute permission character is given as s if the file has the set-group-id bit set; likewise the owner-execute permission character is given as s if the file has the set-user-id bit set.

The last character of the mode (normally x or ‘-’) is t if the 1000 bit of the mode is on. See chmod(1V) for the meaning of this mode. The indications of set-ID and 1000 bits of the mode are capitalized (S and T respectively) if the corresponding execute permission is not set.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks is printed.

Options

-a List all entries; in the absence of this option, entries whose names begin with a '.' are not listed (except for the super-user, for whom ls, but not /usr/5bin/ls, normally prints even files that begin with a '.').

-A Same as -a, except that '.' and '..' are not listed.

-l List in long format, giving mode, number of links, owner, size in bytes, and time of last modification for each file. If the file is a special file the size field will instead contain the major and minor device numbers. If the time of last modification is greater than six months ago, it is shown in the format 'month date year'; files modified within six months show 'month date time'. If the file is a symbolic link the pathname of the linked-to file is printed preceded by '->'.

-F uses suffix * for executible , / for directory , @ for a symbolic link

-g removes the group.

-L If argument is a symbolic link, list the file or directory the link references rather than the link itself.

-q Display non-graphic characters in filenames as the character ?; for ls, this is the default when output is to a terminal.

-r Reverse the order of sort to get reverse alphabetic or oldest first as appropriate.

-R Recursively list subdirectories encountered.

-s Give size of each file, including any indirect blocks used to map the file, in kilobytes (ls) or 512 byte blocks (/usr/5bin/ls).

-t Sort by time modified (latest first) instead of by name.

-u Use time of last access instead of last modification for sorting (with the -t option) and/or printing (with the -l option).

cat

cat - concatenate and display

Syntax

cat [-] [-benstuv] [filename...]

Description

cat reads each filename in sequence and displays it on the standard output. Thus:

"cat goodies" displays the contents of goodies on the standard output, and

"cat goodies1 > > goodies4" appends goodies1 on the end of goodies4.

If no filename argument is given, or if the argument '**-**' is given, cat reads from the standard input. If the standard input is a terminal, input is terminated by an EOF condition (**^D**).

Options

-b Number the lines, as -n, but omit the line numbers from blank lines.

-e Display non-printing characters, as -v, and in addition display a $ character at the end of each line.

-n Precede each line output with its line number.

-s Substitute a single blank line for multiple adja-

cent blank lines.

-t Display non-printing characters, as -v, and in addition display TAB characters as ^I (CTRL-I).

cd

cd - change working directory

Syntax: cd
cd -
cd [directory]

"cd" alone changes to the user's home directory ($HOME).
"cd -" changes to the previous directory.
"cd arg" arg becomes the new working directory. The process must have execute (search) permission in directory. In ksh(1) you may specify a list of directories in which directory is to be sought as a subdirectory if it is not a subdirectory of the current directory; see the Description of the CDPATH variable in ksh(1).

df -k

Display the amount of available disk space in kilobytes

du -k

Display information on disk usage.
du [directory list] reports the space used by the directory and its subdirectories,

default report is of the current directory
-s display summary information
-a reports on files also

diff

diff file1 file2
Display differences between text files

-b ignore blanks and tabs

cmp

cmp file1 file2
Report first difference location , no return if the files match

cal

Display a calendar
cal [month 1-12] year

cal 1999

displays a calendar for all of 1999

mkdir

mkdir - make a directory

Syntax: mkdir [-p] dirname...

Description: mkdir creates directories. Standard entries, '.', for the directory itself, and '..' for its parent, are made automatically. The -p flag generates missing parent directories as needed.

With the exception of the set-gid bit, the current umask(2V) setting determines the mode in which directories are created. The new directory inherits the set-gid bit of the parent directory. Modes may be modified after creation by using chmod(1V).
mkdir requires write permission in the parent directory.

mv

Move (rename) a file or directory
mv existing_file new_name or directory
mv directory directory

rm

rm, rmdir - remove (unlink) files or directories

Syntax rm [-fir] [-] filename... or directory
rmdir directory...

Description rm removes (directory entries for) one or more files. If an entry was the last link to the file, the contents of that file are lost. See ln(1V) for more information about multiple links to files.

To remove a file, you must have write permission on it. There is an exception, if you have write permission in its directory, and if the "sticky bit is not set on the directory, then removal is possible.
rmdir removes each named directory. rmdir only removes empty directories.

Options

- Treat the following arguments as filenames '-' so that you can specify filenames starting with a minus.

-f Force files to be removed without displaying permissions, asking questions or reporting errors.

-i Ask whether to delete each file, and, under -r, whether to examine each directory. Sometimes called the interactive option.

-r Recursively delete the contents of a directory, its subdirectories, and the directory itself.

find

find - find files by name, or by other characteristics

Syntax find pathname-list expression
find component

Description find recursively descends the directory hierarchy for each pathname in the pathname-list, seeking files that match a logical expression written using the operators listed below.

find does not follow symbolic links to other files or directories; it applies the selection criteria to the symbolic links themselves, as if they were ordinary files.

Usage

Operators

n In the Descriptions, the argument n is used as a decimal integer where +n means more than n, -n means less than n, and n means exactly n.

-name filename True if the filename argument matches the current file name. Shell argument Syntax can be used if escaped (watch out for [, ? and *).

-type c True if the type of the file is c, where c is one of:
b for block special file c

c for character special file
d for directory
f for plain file
p for named pipe (FIFO)
l for symbolic link
s for socket

-size n True if the file is n blocks long (512 bytes per block). If n is followed by a c, the size is in characters.

-print Always true; the current pathname is printed.

-ls Always true; prints current pathname together with its associated statistics. These include (respectively) inode number, size in kilobytes (1024 bytes), protection mode, number of hard links, user, group, size in bytes, and modification time. If the file is a special file the size field will instead contain the major and minor device numbers. If the file is a symbolic link the pathname of the linked-to file is printed preceded by '->'. The format is identical to that of ls -gilds (see ls(1V)). Note: formatting is done internally, without executing the ls program.

Example

In a development system, a file called TIMESTAMP is kept in all the manual page directories. Here is how to find all entries that have been updated since TIMESTAMP was created:

"find /usr/share/man/man2 -newer /usr/share/man/man2/TIMESTAMP -print "

/usr/share/man/man2 /usr/share/man/man2/socket.2 /usr/share/man/man2/mmap.2

To find all the files called intro.ms starting from the current directory:

"find . -name intro.ms -print ./manuals/assembler/intro.ms " ./manuals/sun.core/intro.ms ./manuals/driver.tut/intro.ms ./manuals/sys.manager/uucp.impl/intro.ms /supplements/general.works/unix.introduction/intro.ms ./supplements/programming.tools/sccs/intro.ms

To recursively print all files names in the current directory and below, but skipping SCCS directories:

find . -name SCCS -prune -o -print

To recursively print all files names in the current directory and below, skipping the contents of SCCS directories, but printing out the SCCS directory name:

find . -print -name SCCS -prune

To remove files beneath your home directory named a.out or *.o that have not been accessed for a week and that are not mounted using NFS:

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

Backslashes are needed to prevent the shell from interpreting special characters.

chmod

chmod - change the permissions mode of a file

Syntax chmod [-fR] mode filename ...

Description Change the permissions (mode) of a file or files. Only the owner of a file (or the super-user) may change its mode.

The mode of each named file is changed according to mode, which may be absolute or symbolic, as follows.

Absolute Modes

An absolute mode is an octal number constructed from the OR of the following modes:

400 Read by owner.
200 Write by owner.
100 Execute (search in directory) by owner.
040 Read by group.
020 Write by group.
010 Execute (search) by group.
004 Read by others.
002 Write by others.
001 Execute (search) by others.
4000 Set user ID on execution.
2000 Set group ID on execution (this bit is ignored if the file is a directory; it may be set or cleared only using symbolic mode).
1000 Sticky bit, (see chmod(2V) for more information).

Symbolic Modes

A symbolic mode has the form: [who] op permission [op permission] ...

who is a combination of:

u User's permissions.
g Group permissions.
o Others.
a All, or ugo.

If who is omitted, the default is a, but the setting of the file creation mask (see umask in sh(1) or csh(1) for more information) is taken into account. When who is omitted, chmod will not override the restrictions of your user mask.

op is one of:

+ To add the permission.
- To remove the permission.
= To assign the permission explicitly (all other bits for that category, owner, group, or others, will be reset).

Permission is any combination of:

r Read.
w Write.
x Execute.
X Give execute permission if the file is a directory or if there is execute permission for one of the other user classes.
s Set owner or group ID. This is only useful with u or g. Also, the set group ID bit of a directory may only be modified with '+' or '-'.
t Set the sticky bit on directories to restrict file deletions to their owners.

The letters u, g, or o indicate that permission is to be taken from the current mode for the user-class.

Omitting permission is only useful with '=', to take away all permissions.

Multiple symbolic modes, separated by commas, may

be given. Operations are performed in the order specified.
OPTIONS
-f Force. chmod will not complain if it fails to change the mode of a file.
-R Recursively descend through directory arguments setting the mode for each file as described above. When symbolic links are encountered, the mode of the target file is changed, but no recursion takes place.

Example

The first example denies write permission to others, the second makes a file executable by all if it is executable by anyone:

chmod o-w file
chmod +X file

ps

In changing to Solaris2 Sun switched the "ps" commands, the BSD ps is now found in /usr/ucb, while the systemV ps was moved from /usr/5bin to /usr/bin. /usr/local/bin/ps is a link to /usr/ucb/ps on the ITS network.
Syntax (BSD)

ps [[-]acCegjklnrSuUvwx][-tx][num] [kernelname] [c-dump-file] [swap-file]

ps displays information about processes. Normally, only those processes that are running with your effective user ID and are attached to a controlling terminal are shown. Additional categories of processes can be added to the display using various options. In particular, the -a option allows you to include processes that are not owned by you (that do not have your user ID), and the -x option allows you to include processes without control terminals. When you specify both -a and -x, you get processes owned by anyone, with or without a control terminal. The -r option restricts the list of processes printed to “running” processes: runnable processes, those in page wait, or those in short-term non-interruptible waits.

ps displays the process ID, under PID; the control terminal (if any), under TT; the cpu time used by the process so far, including both user and system time), under TIME; the state of the process, under STAT; and finally, an indication of the COMMAND that is running.

The state is given by a sequence of four letters, for example, ‘RWNA’.

First letter indicates the runnability of the process:

R Runnable processes.
T Stopped processes.
P Processes in page wait.
D Processes in non-interruptible waits; typically short-term waits for disk or NFS I/O.
S Processes sleeping for less than about 20 seconds.
I Processes that are idle (sleeping longer than about 20 seconds).
Z Processes that have terminated and that are waiting for their parent process to do a wait(2V) (“zombie” processes).

Second letter indicates whether a process is swapped out;

blank Represented as a SPACE character, in this position indicates that the process is loaded (in memory).
W Process is swapped out.
> Process has specified a soft limit on memory requirements and has exceeded that limit; such a process is (necessarily) not swapped.

Third letter indicates whether a process is running with altered CPU scheduling priority (nice(1)):

blank Represented as a SPACE character, in this position indicates that the process is running without special treatment.
N The process priority is reduced,
< The process priority has been raised artificially.

Fourth letter indicates any special treatment of the process for virtual memory replacement. Currently the possibilities are:

blank Represented as a SPACE character, in this position stands for VANORM.
A Stands for VAANOM. An A typically represents a program which is doing garbage collection.
S Stands for VASEQL. An S is typical of large image processing programs that are using virtual memory to sequentially address voluminous data.

SYNOPSIS (SystemV, /usr/5bin/ps, **Solaris2**)

ps [-acdefjl] [-g grplist] [-p proclist] [-s sidlist] [-t term] [-u uidlist]

OPTIONS

-a Print information about all processes most frequently requested: all those except process group leaders and processes not associated with a terminal.
-c Print information in a format that reflects scheduler properties as described in priocntl(1). The -c option affects the output of the -f and -l options.
-d Print information about all processes except session leaders.
-e Print information about every process now running.
-f Generate a full listing.
-j Print session ID and process group ID.
-l Generate a long listing.
-p proclist List only process data whose process ID numbers are given in proclist.
-t term List only process data associated with term.

Terminal identifiers are specified as a device file name, and an identifier. For example, term/a, or pts/0.

-u uidlist List only process data whose user ID number or login name is given in uidlist. In the listing, the numerical user ID will be printed unless you give the -f option, which prints the login name.

rlogin

rlogin establishes a remote login session from your terminal to the remote machine named hostname.

Syntax rsh or rlogin [-L] [-8] [-ec] [-l username] hostname

Hostnames are listed in the hosts database. Rlogin/rsh from hosts in the /etc/hosts.equiv file (within our local network) do not require passwords and (except for delays) the remote login is transparent. Flow control using ^S (CTRL-S) and ^Q (CTRL-Q) and flushing of input and output on interrupts are handled properly.

The remote terminal type is the same as your local terminal type (as given in your environment TERM variable). The terminal or window size is also copied to the remote system if the server supports the option, and changes in size are reflected as well.

Escapes

Lines that you type which start with the tilde character are “escape sequences” (the escape character can be changed using the -e options):

~. Disconnect from the remote host - this is not the same as a logout, because the local host breaks the connection with no warning to the remote end.

~susp Suspend the login session (only if you are using the C shell). susp is your “suspend” character, usually ^Z, (CTRL-Z), see tty(1).

~dsusp Suspend the input half of the login, but output will still be seen (only if you are using the C shell). dsusp is your “deferred suspend” character, usually ^Y, (CTRL-Y), see tty(1).

Options

-L Allow the rlogin session to be run in “litout” mode.

-8 Pass eight-bit data across the net instead of seven-bit data.

-ec Specify a different escape character, c, for the line used to disconnect from the remote host.

-l username Specify a different username for the remote login. If you do not use this option, the remote username used is the same as your local username.

rsh

rsh connects to the specified hostname and executes the specified command. rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. Interrupt, quit and terminate signals are propagated to the remote command; rsh normally terminates when the remote command does.

Syntax

rsh [-l username] [-n] hostname [command]

If you omit command, instead of executing a single command, rsh logs you in on the remote host using rlogin(1).

Hostnames are given in the hosts database (ypcat hosts).
rsh will not prompt for a password if access is denied on the remote machine unless the command argument is omitted.

Options

-l username Use username as the remote username instead of your local username. In the absence of this option, the remote username is the same as your local username.

-n Redirect the input of rsh to /dev/null. You sometimes need this option to avoid unfortunate interactions between rsh and the shell which invokes it. For example, if you are running rsh and start a rsh in the background without redirecting its input away from the terminal, it will block even if no reads are posted by the remote command. The -n option will prevent this.

The type of remote shell (sh, ksh, or other) is determined by the user’s entry in the password file on the remote system.

Examples

The following command appends the remote file lizard.file from the machine called lizard to the file called example.file on the local machine .

rsh lizard cat lizard.file >> example.file

Commands in double quotes " are carried out on the remote machine.
The following command appends the remote file lizard.file from the machine called lizard to the file called example.file on the remote machine .

rsh lizard "cat lizard.file >> example.file "

NOTE: You cannot run an interactive command (such as vi(1)); use rlogin if you wish to do so.

In Addition...

Commands outside the standard Sun distribution are usually accessed through links in **/usr/local/bin/** or **/sw1/bin** , which are early in our standard **$PATH (**the list of directories which are searched for commands).
Additional information maybe kept in the source directory of the command. "ls -l /usr/local/bin/command" probably will return a link to " /users?/command_dir/bin/command ". Look for information in /users?/command_dir/ ; READMEs, FAQs, white papers , doc directory , *.doc.
i.e.> l /usr/local/bin/scheme
lrwxrwxrwx 1 root other 25 Jun 26 17:13 /usr/local/bin/scheme -> /users7/scheme/bin/scheme
i.e.> lf /users7/scheme/
bin/ doc/ etc/ lib/

i.e.> lf /users7/scheme/doc
macros.txt refman.ps user.ps
/users7/scheme/doc/refman.ps and user.ps are postscript files which can be viewed with pageview or ghostview or printed.
/users13/sa_info contains copies of many FAQs (Frequently Answered Questions) , white papers and articles on various computer topics.
/users13/src_code/ contains program listings from books.

We have archive tools in /usersa/arctools, including: zip & unzip (similar to Phil Katz's PKZIP & PKUNZIP) ; arc ;Gnu Zip utilities (gunzip, gzexe, gzip, zcat, zdiff, zforce, zgrep, zmore & znew); Lharc as xlharc; zoo and fiz. Man pages are provided for most.

REGULAR EXPRESSIONS

Shell supports an extensive and complicated (comparing to DOS) set of matching rules called 'regular expressions.' They may be used to enhance some unix utilities (for instance, rm to remove files matching certain pattern, or fgrep to search a fixed string, or command line editor to substitute one fixed string in a file for another).

Take care when using the characters '$', '*', [, '^', '|', '(', ')', and '\' in the expression, as these characters are also meaningful to the shell. It is safest to enclose the entire expression argument in single quotes '...'.

The following one-character regular expressions match a single character:

c An ordinary character (not one of the special characters discussed below) is a one-character regular expression that matches that character.

\c A backslash (\) followed by any special character is a one-character regular expression that matches the special character itself. The special characters are:

'.', '*', '[', and '\'

(period, asterisk, left square bracket, and backslash, respectively), which are always special, except when they appear within square brackets ([]).

'^' (caret or circumflex), which is special at the beginning of an entire regular expression, or when it immediately follows the left of a pair of square brackets ([]).

$ (currency symbol), which is special at the end of an entire regular expression.

A backslash followed by one of '<', '>', '(', ')', '{', or '}', represents a special operator in the regular expression; see below.

. A '.' (period) is a one-character regular expression that matches any character except NEWLINE.

[string] A non-empty string of characters enclosed in square brackets is a one-character regular expression that matches any one character in that string. If, however, the first character of the string is a '^' (a circumflex or caret), the one-character regular expression matches any character except NEWLINE and the remaining characters in the string. The '^' has this special meaning only if it occurs first in the string. The '-' (minus) may be used to indicate a range of consecutive ASCII characters; for example, [0-9] is equivalent to [0123456789]. The '-' loses this special meaning if it occurs first (after an initial '^', if any) or last in the string. The ']' (right square bracket) does not terminate such a string when it is the first character within it (after an initial '^', if any); that is, []a-f] matches either ']' (a right square bracket) or one of the letters a through f inclusive. The four characters '.', '*', '[', and '\' stand for themselves within such a string of characters.

The following rules may be used to construct regular expressions:

* A one-character regular expression followed by '*' (an asterisk) is a regular expression that matches zero or more occurrences of the one-character regular expression. If there is any choice, the longest leftmost string that permits a match is chosen.

\(and\) A regular expression enclosed between the character sequences \(and \) matches whatever the unadorned regular expression matches. This applies only to grep.

\n The expression \n matches the same string of characters as was matched by an expression enclosed between \(and \) earlier in the same regular expression. Here n is a digit; the sub-expression specified is that beginning with the nth occurrence of \(counting from the left. For example, the expression ^\(.*\)\1$ matches a line consisting of two repeated appearances of the same string.

Concatenation

The concatenation of regular expressions is a regular expression that matches the concatenation of the strings matched by each component of the regular expression.

\< The sequence \< in a regular expression constrains the one-character regular expression immediately following it only to match something at the beginning of a "word"; that is, either at the beginning of a line, or just before a letter, digit, or underline and after a character not one of these.

\> The sequence \> in a regular expression constrains the one-character regular expression immediately following it only to match something at the end of a "word"; that is, either at the end of a line, or just before a character which is neither a letter, digit, nor underline.

\{m\} \{m,\} \{m,n\}

A regular expression followed by \{m\}, \{m,\}, or \{m,n\} matches a range of occurrences of the regular expression. The values of m and n must be non-negative integers less than 256; \{m\} matches exactly m occurrences; \{m,\} matches at least m occurrences; \{m,n\} matches any number of occurrences between m and n inclusive. Whenever a choice exists, the regular expression matches as many occurrences as possible.

^ A circumflex or caret (^) at the beginning of an entire regular expression constrains that regular expression to match an initial segment of a line.

$ A currency symbol ($) at the end of an entire regular expression constrains that regular expression to match a final segment of a line.

The construction. ^entire regular expression $ constrains the entire regular expression to match the entire line.

? A regular expression followed by '?' (a question mark) is a regular expression that matches zero or one occurrences of the one-character regular expression.
If there is any choice, the longest leftmost string that permits a match is chosen.

The order of precedence of operators at the same parenthesis level is '[ ]' (character classes), then '*' '+' '?' (closures), then concatenation, then '|' (alternation) and NEWLINE.
Examples
Search a file for a fixed string using fgrep: fgrep intro /usr/share/man/man3/*.3*
Look for character classes using grep: grep '[1-8]([CJMSNX])' /usr/share/man/man1/*.1
Look for alternative patterns using egrep: egrep '(Sally|Fred) (Smith|Jones|Parker)' telephone.list
To get the filename displayed when only processing a single file, use /dev/null as the second file in the list: grep 'Sally Parker' telephone.list /dev/null

GNU's grep is called as Grep (capital **G**).
man -a grep Shows both grep man pages.
Gnu grep has incorporated egrep and fgrep as -E and -F options and added -num, -A [num], -B [num] and -C options.
-num Matches will be printed with num lines of leading and trailing context.
-A num Print num lines of trailing context after matched lines.
-B num Print num lines of leading context before matched lines.
-C Equivalent to -A2 -B2 .