**NAME**

ls – list contents of directory

**SYNOPSIS**

**/usr/bin/ls** [ **–aAbcCdfFgilLmnopqrRstux1** ] [ *file* . . . ]
**/usr/xpg4/bin/ls** [ **–aAbcCdfFgilLmnopqrRstux1** ] [ *file* . . . ]

**AVAILABILITY**

**/usr/bin/ls** SUNWcsu

**/usr/xpg4/bin/ls** SUNWxcu4

**DESCRIPTION**

For each *file* that is a directory, **ls** lists the contents of the directory; for each *file* that is an ordinary file, **ls** repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format for output directed to a terminal is multi–column with entries sorted down the columns. The –**1** option allows single column output and –**m** enables stream output format. In order to determine output formats for the –**C**, –**x**, and –**m** options, **ls** uses an environment variable, **COLUMNS**, to determine the number of character positions available on one output line. If this variable is not set, the **terminfo**(4) database is used to determine the number of columns, based on the environment variable **TERM**. If this information cannot be obtained, 80 columns are assumed.

The mode printed under the –**l** option consists of ten characters. The first character may be one of the following:

- **d** the entry is a directory;
- **l** the entry is a symbolic link;
- **b** the entry is a block special file;
- **c** the entry is a character special file;
- **p** the entry is a fifo (or “named pipe”) special file;
- – the entry is an ordinary file;
- **|** the entry is a FIFO.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner’s permissions; the next to permissions of others in the user-group of the file; and the last to all others. Within each set, the three characters indicate permission to read, to write, and to execute the file as a program, respectively. For a directory, ‘‘execute’’ permission is interpreted to mean permission to search the directory for a specified file. The character after permissions is ACL indication. A plus sign is displayed if there is an ACL associated with the file. Nothing is displayed if there are just permissions.

**ls** –**l** (the **l**ong list) prints its output as follows:

**–rwxrwxrwx+ 1 smith dev 10876 May 16 9:42 part2**

Reading from right to left, you see that the current directory holds one file, named **part2**. Next, the last time that file's contents were modified was 9:42 A.M. on May 16. The file contains 10,876 characters, or bytes. The owner of the file, or the user, belongs to the group **dev** (perhaps indicating ''development''), and his or her login name is **smith**. The number, in this case **1**, indicates the number of links to file **part2**; see **cp**(1). The plus sign indicates that there is an ACL associated with the file. Finally, the dash and letters tell you that user, group, and others have permissions to read, write, and execute **part2**.

The execute (**x**) symbol here occupies the third position of the three-character sequence. A – in the third position would have indicated a denial of execution permissions.

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 |
| **/usr/bin/ls** **l** | mandatory locking occurs during access (the set-group-ID bit is on and the group execution bit is off) |
| **/usr/xpg4/bin/ls** **L** | mandatory locking occurs during access (the set-group-ID bit is on and the group execution bit is off) |
| **s** | the **s**et-user-ID or **s**et-group-ID bit is on, and the corresponding user or group execution bit is also on |
| **S** | undefined bit-state (the set-user-ID bit is on and the user execution bit is off) |
| **t** | the 1000 (octal) bit, or sticky bit, is on (see **chmod**(1)), and execution is on |
| **T** | the 1000 bit is turned on, and execution is off (undefined bit-state) |

For user and group permissions, the third position is sometimes occupied by a character other than **x** or –. **s** also may occupy this position, referring to the state of the **s**et-ID bit, whether it be the user's or the group's. The ability to assume the same ID as the user during execution is, for example, used during login when you begin as root but need to assume the identity of the user you login as.

In the case of the sequence of group permissions, **l** may occupy the third position. **l** refers to mandatory file and record locking. This permission describes a file's ability to allow other files to lock its reading or writing permissions during access.

For others permissions, the third position may be occupied by **t** or **T**. These refer to the state of the sticky bit and execution permissions.

## OPTIONS

**–a** List all entries, including those that begin with a dot (**.**), which are normally not listed.

**–A** List all entries, including those that begin with a dot (**.**), with the exception of the working directory (**.**) and the parent directory (**..**).

**–b** Force printing of non-printable characters to be in the octal *****ddd* notation.

**–c** Use time of last modification of the i-node (file created, mode changed, and so forth) for sorting (**–t**) or printing (**–l** or **–n**).

**–C** Multi-column output with entries sorted down the columns. This is the default output format.

**−d** If an argument is a directory, list only its name (not its contents); often used with **−l** to get the status of a directory.

**−f** Force each argument to be interpreted as a directory and list the name found in each slot. This option turns off **−l**, **−t**, **−s**, and **−r**, and turns on **−a**; the order is the order in which entries appear in the directory.

**−F** Put a slash (**/**) after each filename if the file is a directory, an asterisk (*) if the file is an executable, and an at-sign (**@**) if the file is a symbolic link.

**−g** The same as **−l**, except that the owner is not printed.

**−i** For each file, print the i-node number in the first column of the report.

**−l** List in long format, giving mode, ACL indication, number of links, owner, group, size in bytes, and time of last modification for each file (see above). If the file is a special file, the size field instead contains 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 filename is printed followed by "→" and the path name of the referenced file.

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

**−m** Stream output format; files are listed across the page, separated by commas.

**−n** The same as **−l**, except that the owner's UID and group's GID numbers are printed, rather than the associated character strings.

**−o** The same as **−l**, except that the group is not printed.

**−p** Put a slash (**/**) after each filename if the file is a directory.

**−q** Force printing of non-printable characters in file names as the character question mark (**?**).

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

**−R** Recursively list subdirectories encountered.

**−s** Give size in blocks, including indirect blocks, for each entry.

**−t** Sort by time stamp (latest first) instead of by name. The default is the last modification time. (See **−u** and **−c**.)

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

**−x** Multi-column output with entries sorted across rather than down the page.

**−1** Print one entry per line of output.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: **−C** and **−1** (one), **−c** and **−u**. The last option specified in each pair determines the output format.

**/usr/bin/ls**

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: −**C** and −**l** (ell), −**m** and −**l** (ell), −**x** and −**l** (ell). The −**l** option overrides the other option specified in each pair.

**/usr/xpg4/bin/ls**

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: −**C** and −**l** (ell), −**m** and −**l** (ell), −**x** and −**l** (ell). The last option specified in each pair determines the output format.

## OPERANDS

The following operand is supported:

*file* A path name of a file to be written. If the file specified is not found, a diagnostic message will be output on standard error.

## EXAMPLES

An example of a file's permissions is:

**−rwxr−−r−−**

This describes a file that is readable, writable, and executable by the user and readable by the group and others.

Another example of a file's permissions is:

**−rwsr−xr−x**

This describes a file that is readable, writable, and executable by the user, readable and executable by the group and others, and allows its user-ID to be assumed, during execution, by the user presently executing it.

Another example of a file's permissions is:

**−rw−rwl−−−**

This describes a file that is readable and writable only by the user and the group and can be locked during access.

An example of a command line:

**example% ls −a**

This command prints the names of all files in the current directory, including those that begin with a dot (**.**), which normally do not print.

Another example of a command line:

**example% ls −aisn**

This command provides information on **a**ll files, including those that begin with a dot (**a**), the **i**-number—the memory address of the i-node associated with the file—printed in the left-hand column (**i**); the **s**ize (in blocks) of the files, printed in the column to the right of the i-numbers (**s**); finally, the report is displayed in the **n**umeric version of the long list, printing the **UID** (instead of user name) and **GID** (instead of group name) numbers associated with the files.

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

**ENVIRONMENT**

See **environ**(5) for descriptions of the following environment variables that affect the execution of **ls**: **LC_COLLATE**, **LC_CTYPE**, **LC_TIME**, **LC_MESSAGES**, **NLSPATH**, and **TZ**.

**COLUMNS** Determine the user's preferred column position width for writing multiple text-column output. If this variable contains a string representing a decimal integer, the **ls** utility calculates how many path name text columns to write (see −**C**) based on the width provided. If **COLUMNS** is not set or invalid, 80 is used. The column width chosen to write the names of files in any given directory will be constant. File names will not be truncated to fit into the multiple text-column output.

**EXIT STATUS**

**0** All information was written successfully.

**>0** An error occurred.

**FILES**

| | |
|---|---|
| **/etc/group** | group IDs for **ls** −**l** and **ls** −**g** |
| **/etc/passwd** | user IDs for **ls** −**l** and **ls** −**o** |
| **/usr/share/lib/terminfo/?/*** | terminal information database |

**SEE ALSO**

**chmod**(1), **cp**(1), **setfacl**(1), **terminfo**(4), **environ**(5)

**NOTES**

Unprintable characters in file names may confuse the columnar output options.

The total block count will be incorrect if if there are hard links among the files.