The ls -l command output format, filesystem and SELinux security context

Output format example	-rw-r--r--@ 1 jdoe staff ␣ 5111 9 Jun 14:30 readme.rst.txt							Last updated on: 2025-09-17		

Fields	-	rw-	r--	r--	@	1	jdoe	staff	5111	9 Jun 14:30	readme.rst.txt
	Device Type:	*Owner*	*Group*	*Word*	*Optional extra field*						
								ownership			

Description

Note: use the **info ls** command to see more information related to your system.

See Also: **ls @ wikipedia** with all the identified external links.

- **-** Regular file.
- **b** Block special file.
- **c** Character special file.
- **C** High performance (*contiguous data*) file.
- **d** Directory.
- **D** Door (Solaris).
- **l** (letter l) Symbolic link.
- **M** Off-line (migrated) file (Cray DMF).
- **n** Network special file (HP-UX).
- **p** FIFO (named pipe).
- **P** Port (Solaris).
- **s** Socket.
- **?** Some other file type.

☝ sys/stat.h or bits/stat.h define flags for ➡

Discretionary Access Control (DAC) Permissions:

- **read**: Allow opening/reading a *file*.
  - Allow listing *directory's* content if 'x' attribute is also set.
- **write**: Allow writing to *file*. Ability to rename or delete file is controlled by the directory attribute.
  - Allow files in a *directory* to be created, renamed, deleted if the 'x' attribute is also set.
- **other**:
  - **s** : If set-user-ID (S_ISUID: 04000) or set-group-ID (S_ISGID: 02000) and corresponding executable bit are both set.
  - **S** : If the set-user-ID (04000) or set-group-ID (02000) is set but the corresponding executable bit is not set.
  - **t** : If the restricted deletion flag or sticky bit (S_ISVTX: 01000), and the other-executable bit, are both set. The restricted deletion flag is another name of the sticky bit.
  - **T** : If the restricted deletion flag or sticky bit (01000) is set but the other-executable bit is not set.
  - **x** : Allows a *file* to be treated as a program and executed. Script files must be also set as readable to be executable.
    - Allows a *directory* to be entered (eg. via a **cd** command).
  - **–** : otherwise.

Optional extra field

 macOS only:

- **@** has **extended attributes**.
- **%** dataless file or directory.

 Linux only:

- **.** Flag that file has **SELinux security context**

  The SELinux context is shown with ls -Z option.

Number of links or directories

User *ownership:* user that owns the file or directory

Group ownership

Size in bytes.

With **ls -lh**, size format is human readable with units:
- **k** : kilo
- **M** : mega
- **G** : giga

Date of last modification.

Date format might be affected by the LANG environment variable.

On Linux, you can change the date format with the **—time-style** option.

For example:
ls -l --time-style="long-iso"

Name of the file, or the symlink.

Extra Notes:

- **POSIX File System Permissions**

•	**s**	The s and S bits identify whether the set user ID or set group ID permissions are active.
•	**S**	These are special permissions bits that allow a program, when run by any user, to be run with the effective UID of the owner (identified by the ownership fields).
		• For example, if the user ownership is root and the s bit is set, another user will be able to run the program as if it was root.
		This permission is therefore a security risk and should be restricted to the programs that absolutely require this (as sudo does for example).

 SELinux: With **-Z** option:

References:
- **SELinux intro @ Gentoo wiki**
- **SELinux for mere mortals**

SELinux security context
- Shown with the -Z option between ownership & size for the **ls -l** output: in place of ␣ above.
- **SELinux Notebook** (the authors)
  - Table of Contents
- **Red Hat SELinux**
- **SELinux @ Gentoo wiki**
- **SELinux @ Fedora wiki**
- **SELinux @ ArchLinux wiki**
- **Rocky Linux 8 @ server-world**
- **Alma Linux 9 @ server-world**

•	**?**	**?** is displayed when the file has no associated **SELinux security context** (see also this and this) which implements a *Mandatory Access Control* for Linux. Otherwise it shows:

SELinux security context : as string of **user:role:type:level** syntax with the following fields (as described in the SELinux RedHat web page):

- user (…_u) — The **SELinux user** identity. This can be associated to one or more roles that the SELinux user is allowed to use.
- role (…_r) — The **SELinux role**. This can be associated to one or more types the SELinux user is allowed to access.
- type (…_t) — The **SELinux type** of the file (the **SELinux object**). It defines what access permissions the SELinux user has to that object.
- level/range — **SELinux security level** field (or range). It is only present if the policy supports MCS or MLS. The entry can consist of:
  - A single security level that contains a **sensitivity** level and zero or more **categories** (e.g. s0, s1:c0, s7:c10.c15).
  - A range that consists of two security levels (a low and high) separated by a hyphen (e.g. s0 - s15:c0.c1023).

☝ *On SELinux:* The **-Z** switch is available on several utilities to show or manage SELinux security contexts information. For example:

For files:	**ls -lZ**		*For users:*	**id -Z**
For processes:	**ps -efZ** or **ps axZ**		*For sockets:*	**ss -z** or **ss -Z** or **netstat -WeeZ**

☝ **On Emacs** Use the **selinux-policy** external package to edit SELinux policy files. Under PEL, set **pel-use-selinux-policy** user-option to **t** to get PEL to install and use it for the .te, .if, .fc and .spt files.

Filesystem	To list all filesystems used:	On 🐧: `df –hT` On 🍎: `df –hY`	In both cases, the -h option provides simpler, human readable, size values.	🐧 Some file systems used in Linux: • xfs • tmpfs, temporary file system • devtmpfs, temp fs for dynamically created devices • Ext4, a journaling file system • prl_fs : Parallels Desktop VM file system • fuse.sshfs	🍎 Some file systems on macOS: • apfs • hfs+ • devs • autofs • nulls • smbfs
	On 🐧, to list block devices and their related file systems:	On 🐧: `lsblk –f`	This shows the block device tree and their file systems; their type, label, UUID and mount point.		
	On 🐧, to list the file system of a directory or file:	On 🐧: `stat –f –c %T /path/to-dir/or-file`			
Manipulating files extended attributes	The following commands allow listing, reading and writing the extended attributes of files and directories. • Extended attributes are name:value pairs. • The attribute name is a fully.qualified.name. Something like `security.selinux` or `system.posix_acl_access`.			• **attr** • **getfattr** • **setfattr**	
HowTo See: **ls man pages**	List attribute of a directory DIR (not the files it holds)	• `ls –ld DIR`	List attributes of a directory (not its content)		
		• `ls –lda DIR`	List attributes of a hidden directory (not its content)		
		• `ls –ldaZ DIR`	List all attributes of a hidden directory (not its content)		