Leiden University
Medical Center

User Environment

Users, Groups, File permissions and Ownership

Department of Human Genetics
Center for Human and Clinical Genetics

Outline

When do we have to deal with these things?

Not all files and folders are readable for everyone.

```
1   $ cat /var/log/syslog
2   cat: /var/log/syslog: Permission denied
3   $ whoami
4   zorro
```

Listing 1: Permission denied.

Apparently, the user `zorro` is not allowed to read `syslog`.

Users

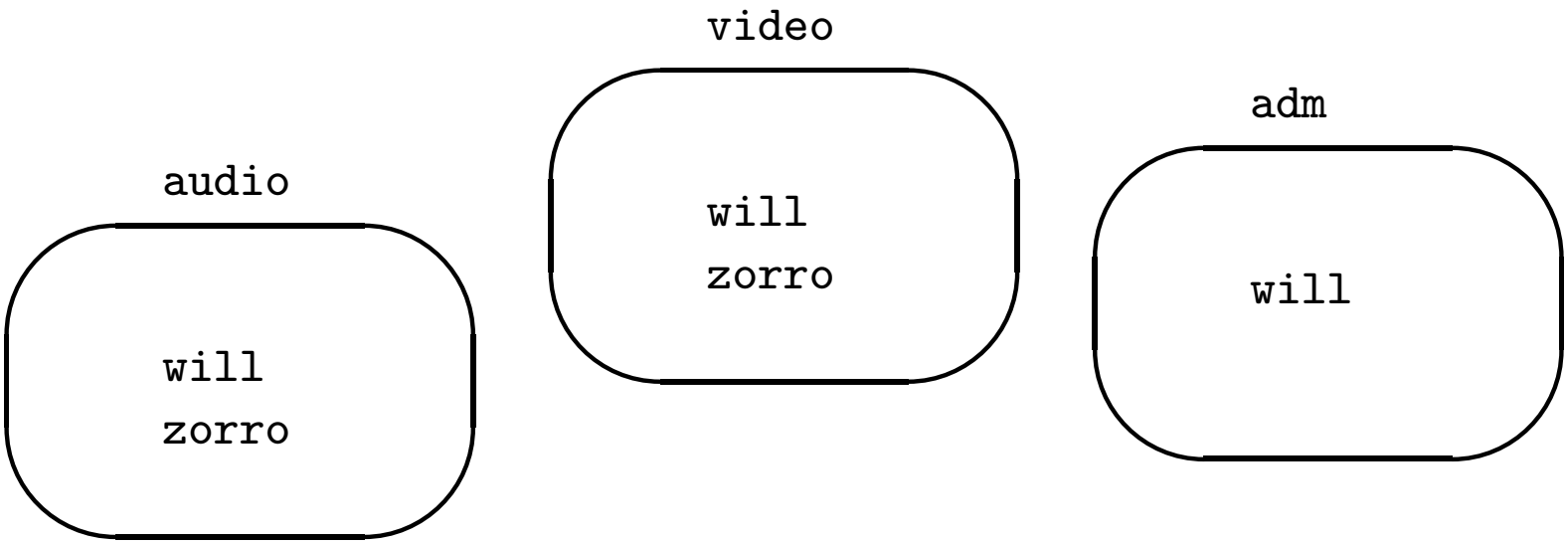
Every Linux / Unix system has multiple users:

- `zorro`
- `will`
- `root`

The users `zorro` and `will` are specific to this lecture.

The user `root` is present on *all* systems. This account is only used for administration purposes.

Group membership



group	users
audio	will, zorro
video	will, zorro
adm	will

Table 1: Group membership.

Group membership

Use `groups` to see a user's group memberships.

```
1   $ groups zorro
2   zorro : audio video
3   $ groups will
4   will : audio video adm
```

Listing 2: See group memberships.

The user `will` is member of the group `adm`.

Find out who owns files

We use the familiar `ls` to find out who owns a file and which permissions are set.

```
1   $ ls -al
```

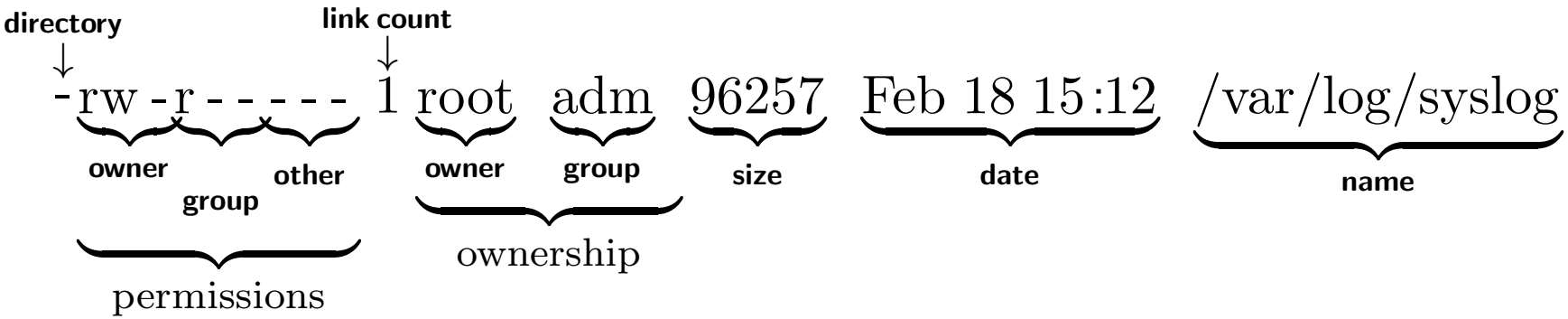
Listing 3: Long listing format.

option	meaning
`-a`	show all files (including hidden files)
`-l`	long listing format

Table 2: Some important options.

File permissions

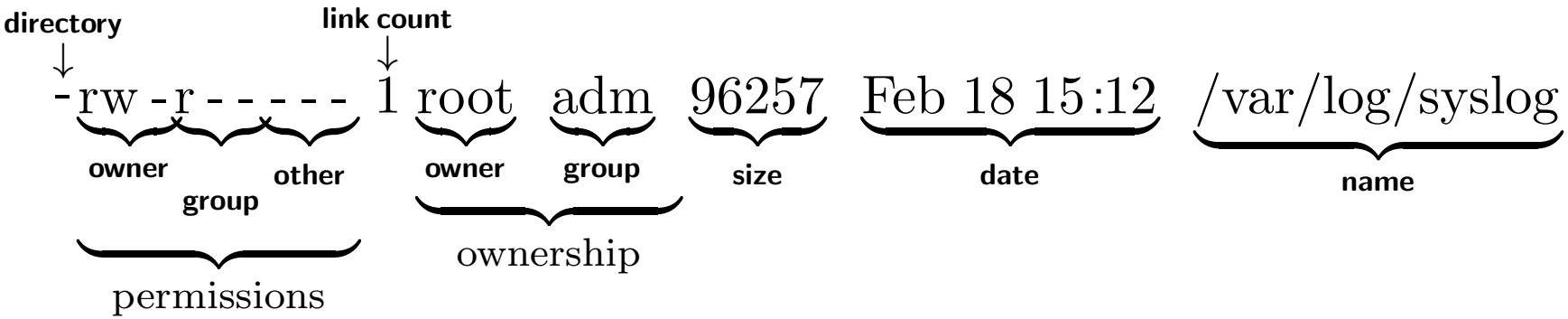
Find out who owns files



This file is owned by the user `root` and the group `adm`.

File permissions

Find out who owns files

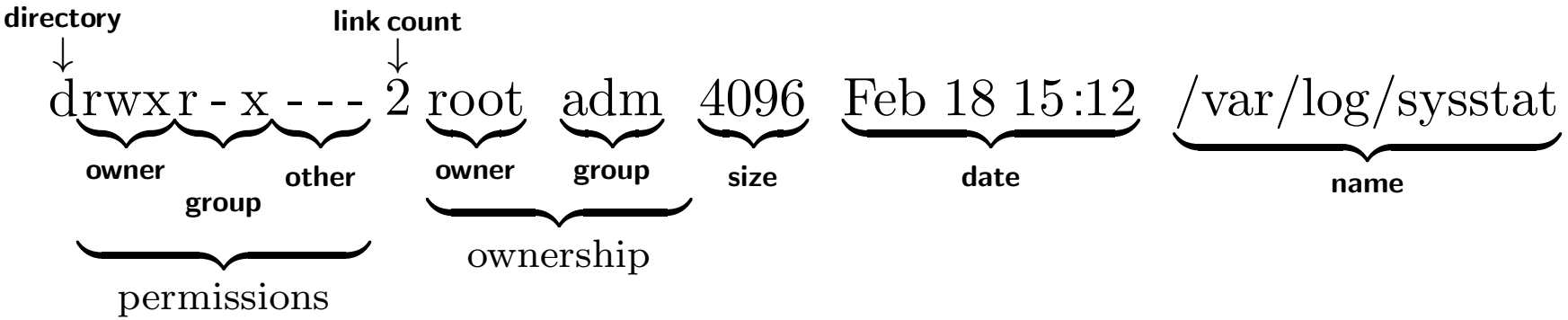


This file is owned by the user `root` and the group `adm`.

Additional information in this output:

- Permissions (next slides).
- Link count.
- File size.
- Last modification date.
- File or directory name.

File permission flags



abbreviation	meaning
`d`	directory
`r`	read permission
`w`	write permission
`x`	execute permission

Table 3: Permissions.

File permission flags

```
1   $ ls -al syslog
2   -rw-r----- 1 root adm 9257 Feb 18 15:12 syslog
```

Listing 4: Long listing of a file.

user	status	permissions
`root`	owner	`rw-`
`will`	member of the `adm` group	`r--`
`zorro`	normal user	`---`

Table 4: How permissions are applied in Listing 4.

File permission flags

```
1   $ ls -ald sysstat
2   drwxr-x--- 2 root adm 4096 Feb 18 15:12 sysstat
```

Listing 5: Long listing of a directory.

user	status	permissions
`root`	owner	`rwx`
`will`	member of the `adm` group	`r-x`
`zorro`	normal user	`---`

Table 5: How permissions are applied in Listing 5.

The execute flag

The execute flag for files.

```
1  $ ls -al
2  drwx------  2 zorro user  1024 Mar 24 13:50 .
3  drwxrwxrwx 27 root  root 10240 Mar 24 13:50 ..
4  -rwx------  1 zorro user    29 Mar 24 13:50 hello
5  -rw-------  1 zorro user    20 Mar 24 13:50 text
```

Listing 6: Long listing of two files.

```
1  $ ./hello
2  Hello world!
3  $ ./text
4  bash: ./text: Permission denied
```

Listing 7: We can execute only one file.

Changing permissions

Setting permissions is done with `chmod`.

```
1  $ ls -al hello
2  -rw-------  1 zorro user    29 Mar 24 13:50 hello
3  $ chmod u+x hello
4  -rwx------  1 zorro user    29 Mar 24 13:50 hello
```

Listing 8: Set the execute flag.

abbreviation	meaning	example
`u`	user	`chmod u+w hello`
`g`	group	`chmod g+rx hello`
`o`	other	`chmod o-rwx hello`

Table 6: Setting permissions.

Changing permissions

The execute flag for directories.

```
1   $ ls -al
2   drwx------  2 zorro user  1024 Mar 24 13:50 .
3   drwxrwxrwx 27 root  root 10240 Mar 24 13:50 ..
4   -rwx------  1 zorro user    29 Mar 24 13:50 hello
```

Listing 9: Long listing of a directory.

```
1   $ chmod u-r .
2   $ ls -al
3   ls: cannot open directory .: Permission denied
4   $ ./hello
5   Hello world!
```

Listing 10: Make a directory unlistable.

Changing permissions

To remove access to a directory completely, remove the execute flag.

```
1   $ chmod u-x .
2   $ ./hello
3   bash: ./hello: Permission denied
```

Listing 11: Close a directory.

These manipulations are useful when you want to give people access, but you do not want them to browse.

Changing permissions

Summary of valid combinations.

type	flags	result
file	`+x +r`	Executable and readable.
file	`-x +r`	Readable.
directory	`+x +r`	Usable and readable.
directory	`+x -r`	Usable but not readable.

Table 7: Combinations of read and execute flags.

Summary

Users, groups and file permissions

Linux uses *users* and *groups* for access control.

Users and groups:

- A user can be a member of multiple groups.

Files and directories:

- Are owned by a user.
- Are owned by a group.

Access control is done by setting permissions for users and groups on files or directories.

Users, groups and file permissions

```
1   -rwxr-x---  1 zorro user    29 Mar 24 13:50 hello
```

Listing 12: Example.

Questions:

- Which user owns this file?
- Which group owns this file?

Users, groups and file permissions

```
1   -rwxr-x---  1 zorro user    29 Mar 24 13:50 hello
```

Listing 12: Example.

Questions:

- Which user owns this file?
- Which group owns this file?
- Which permissions does `zorro` have?
- Which permissions do people in the `user` group have?
- Which permissions do other people have?

Users, groups and file permissions

```
1   drwxr-x--x  1 zorro user    29 Mar 24 13:50 hello
```

Listing 13: Example.

Questions:

- Who can write to this folder?
- Who can see the contents of this folder?
- Who can access this folder?

Leiden University
Medical Center

Acknowledgements

Mihai Lefter
Jeroen Laros