



LEIDEN UNIVERSITY MEDICAL CENTER

LUMC Basic Linux Course

Ownership, Installing and Updating Software

Jeroen F. J. Laros

Department of Human Genetics

Center for Human and Clinical Genetics



- Ownership
 - When do we have to deal with it?
 - Users and groups
 - File permissions
- Installing software
 - Updating
 - Simple installation
 - Searching for packages
 - Manual installation
 - * Archives
 - * The make command

Ownership: When do we have to deal with it?

Permission denied

```
> cat /var/log/syslog  
cat: /var/log/syslog: Permission denied
```

```
> whoami  
jlaros
```

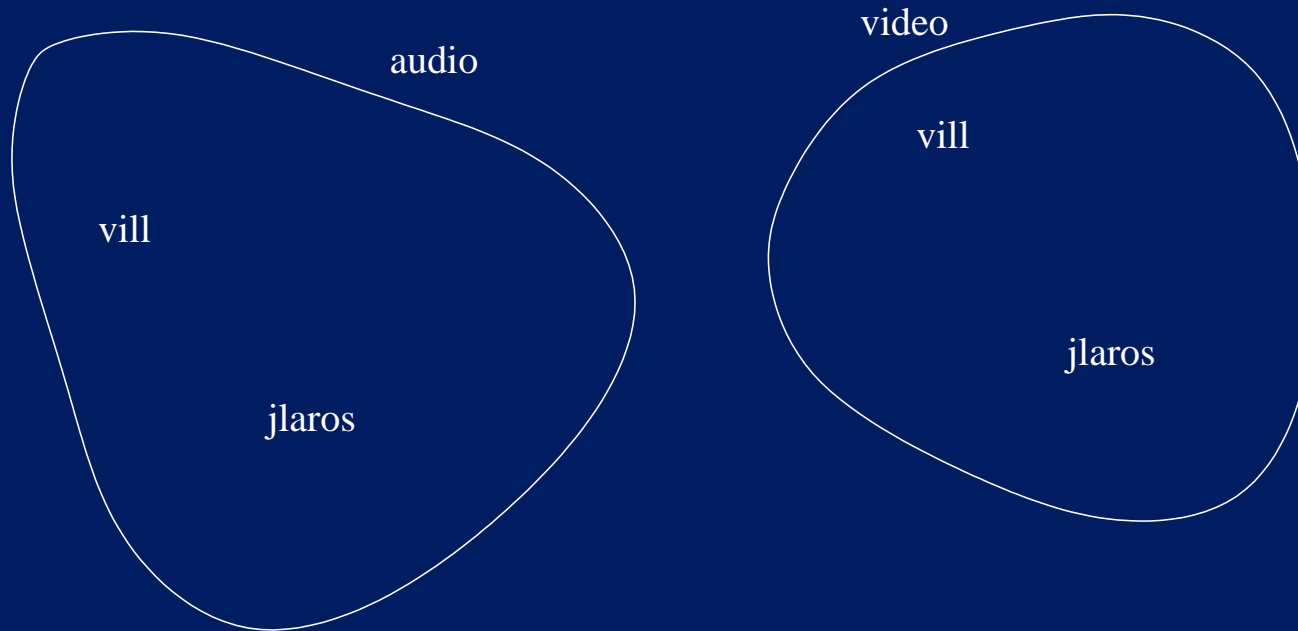
Apparently, the user `jlaros` is not allowed to read `syslog`.

Ownership: Users and groups

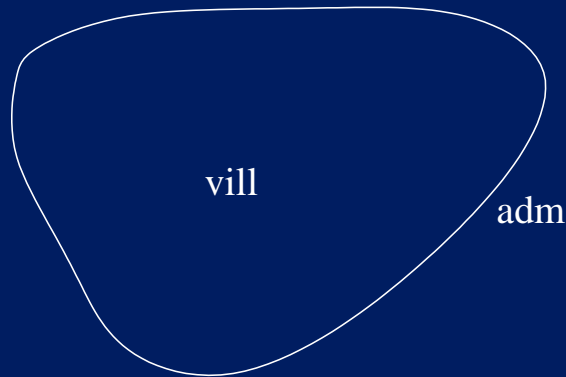
- Linux uses *users* and *groups* for access control.
- Files and directories are owned by one *user*.
- A user can be part of a *group*.

By using *group permissions* on files and putting users in a group, access to files and directories can be controlled.

Ownership: Users and groups



group	users
audio	vill, jlaros
video	vill, jlaros
adm	vill



Ownership: Users and groups

```
> groups jlaros
```

```
jlaros : jlaros audio video
```

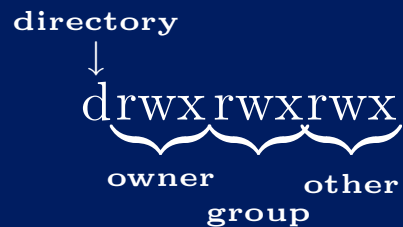
```
> groups vill
```

```
vill : vill audio video adm
```

Ownership: File permissions

```
> ls -al
```

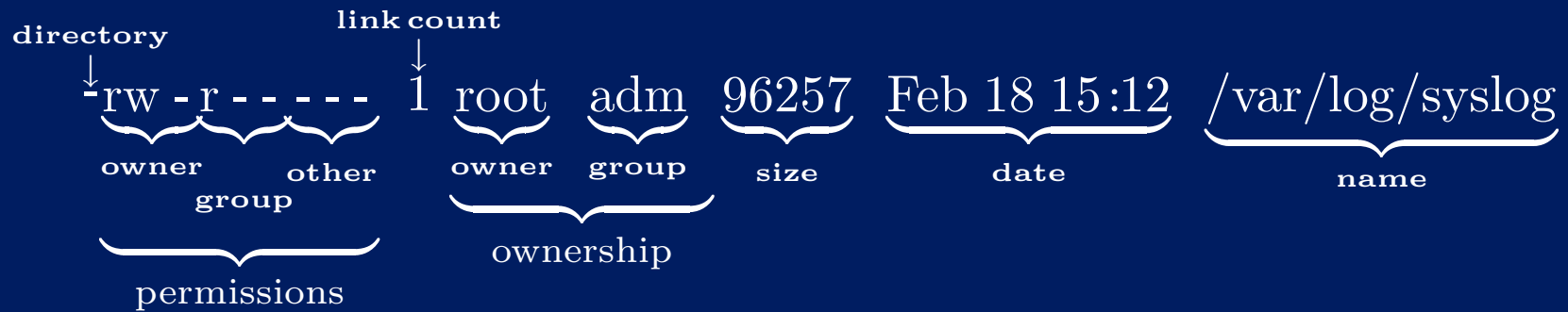
option	meaning
-a	show all files (including hidden files)
-l	long listing format



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

Ownership: File permissions

```
> ls -al /var/log/syslog
```



user	status	permissions
root	owner	rw-
vill	member of the "adm" group	r--
jlaros	normal user	---

The execute bit for files.

```
> ls -al
drwx-----  2 jlaros  jlaros  1024 Mar 24 13:50 .
drwxrwxrwt 27 root    root   10240 Mar 24 13:50 ..
-rwx-----  1 jlaros  jlaros   29 Mar 24 13:50 hello
-rw-----  1 jlaros  jlaros   20 Mar 24 13:50 textfile.txt
```

```
> ./hello
Hello world!
```

```
> ./textfile.txt
bash: ./textfile.txt: Permission denied
```

```
> cat textfile.txt
This is a textfile.
```

The execute bit for directories.

```
> ls -al
drwx-----  2 jlaros  jlaros   1024 Mar 24 13:50 .
drwxrwxrwt 27 root    root    10240 Mar 24 13:50 ..
-rwx-----  1 jlaros  jlaros    29 Mar 24 13:50 hello
-rw-----  1 jlaros  jlaros    20 Mar 24 13:50 textfile.txt
```

```
> chmod -r .
> ls -al
ls: cannot open directory .: Permission denied
```

```
> ./hello
Hello world!
```

```
> chmod -x .
> ./hello
bash: ./hello: Permission denied
```

Summary of valid combinations.

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

First of all, run the update command to make sure we download the latest versions.

```
> sudo apt-get update
```

If we want to apply all updates, run:

```
> sudo apt-get upgrade
```

In many cases, the package name is equal to the command we want to use:

First we install a program called “sl”.

```
> sudo apt-get install sl  
> sl
```

If we want to search for all packages that have anything to do with “alignment”:

```
> apt-cache search alignment
bwa - Burrows-Wheeler Aligner
samtools - processing sequence alignments in SAM
          and BAM formats
seaview - Multiple sequence alignment editor
sigma-align - Simple greedy multiple alignment of
             non-coding DNA sequences
sim4 - tool for aligning cDNA and genomic DNA
wise - comparison of biopolymers, commonly DNA and
       protein sequences
etc...
```

Suppose we know a command, but we don't know which package to install:

```
> apt-cache search extract_genes.pl
```

This will return nothing (unlike our “sl” example).

```
> apt-file update
```

```
> apt-file search extract_genes.pl
```

```
bioperl:
```

```
/usr/share/doc/bioperl/examples/tools/extract_genes.pl
```

We now get a list of packages that provide the file “extract_genes.pl”.

```
> sudo apt-get install bioperl
```


Reasons for manual installation:

- There is no package available.
- We want the latest (development) version.
- We don't get permission to install something system wide.

Installing software: Manual installation

Commonly used archiving programs:

- `zip`
- `tar` in combination with `gzip` or `bzip2`

`zip` is commonly used in Windows and easy to use in Linux.

To unpack a `.zip` archive, we use the command:

```
unzip archivename.zip
```

To create a `.zip` archive, use:

```
zip -r archivename.zip directoryname
```

The option `-r` stands for *recursive*.

`tar` is an old program used to pack lots of files into one archive to put it on tape (hence the name Tape ARchive).

A `tar` archive is *not* compressed, but separate compression utilities are used.

- `gzip` is fast.
- `bzip2` is slow, but compresses better than `gzip`.

Commonly used options for tar:

option	meaning
-x	extract
-c	create
-z	Use the gzip compression utility
-j	Use the bzip2 compression utility
-v	Be verbose
-f <filename>	The name of the archive

Extraction of a `tar.gz` file (tar compressed with gzip):

```
tar -xzvf myarchive.tar.gz
```

Creation of a `tar.gz` file (tar compressed with gzip):

```
tar -czvf myarchive.tar.gz directoryname
```

Extraction of a `tar.bz2` file (tar compressed with bzip2):

```
tar -xjvf myarchive.tar.bz2
```

Creation of a `tar.bz2` file (tar compressed with bzip2):

```
tar -cjvf myarchive.tar.bz2 directoryname
```

- Go to the Bowtie website.
- Click on Bowtie 0.12.3 in the “latest release” section
- Click on `bowtie-0.12.3-src.zip`
- Choose “save file”
- Open a terminal

```
> unzip bowtie-0.12.3-src.zip  
> cd bowtie-0.12.3  
> make  
> ./bowtie -h
```

<http://eu.liacs.nl/LinuxCourse/>