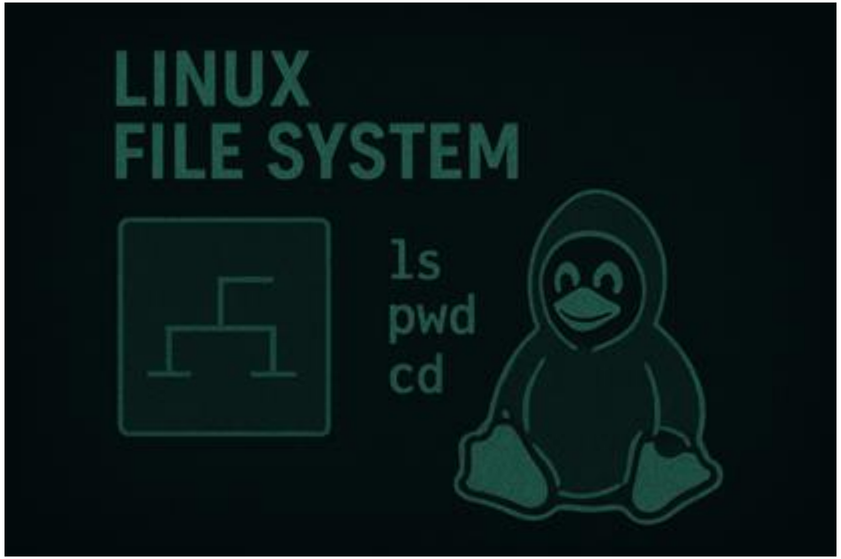
Operating System fundamentals

Navigating the file system



Karel de Grote Hogeschool

Contents

Karel de Grote Hogeschool

Course text

- chapter 3: Manage Files from the Command Line

Karel de Grote Hogeschool

The file system

1

Karel de Grote Hogeschool

The Windows file system

In Windows a file can be specified by its path consisting of:

- drive letter
- directories (folders)
- filename

Backslashes are used to separate the different parts

A path in Windows is not case sensitive

Example:

```
C:\Users\username\Documents\text.txt
```

Karel de Grote Hogeschool

The Linux file system

In Linux (and MacOS) there are no drive letters. A path consists of:

- directories (folders)
- filename

Forward slashes are used to separate the different parts

A path in Linux is always case sensitive!

Example:

```
/home/username/Documents/textfile
```

The first slash is called the “root” of the file system

Linux filesystem

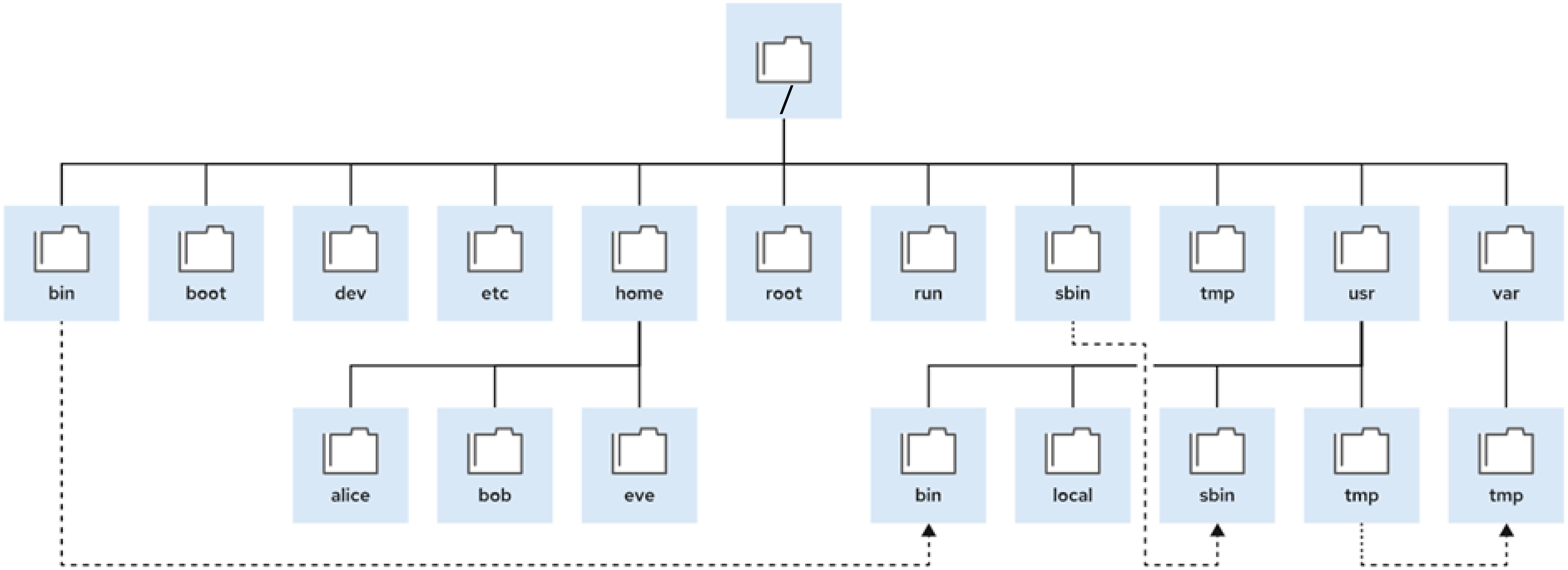


Figure 3.1: Significant file-system directories in Red Hat Enterprise Linux 9

Location	Purpose
`/boot`	Files to start the boot process.
`/dev`	Special device files that the system uses to access hardware.
`/etc`	System-specific configuration files.
`/home`	Home directory, where regular users store their data and configuration files.
`/root`	Home directory for the administrative superuser, `root`.
`/run`	Runtime data for processes that started since the last boot. This data includes process ID files and lock files. The contents of this directory are re-created on reboot.
`/tmp`	A world-writable space for temporary files. Files that are not accessed, changed, or modified for 10 days are deleted from this directory automatically.
`/usr`	Installed software, shared libraries, including files, and read-only program data. Significant subdirectories include: ● `/usr/bin`: User commands ● `/usr/sbin`: System administration commands ● `/usr/local`: Locally customized software
`/var`	System-specific variable data should persist between boots. Files that dynamically change, such as databases, cache directories, log files, printer-spooled documents, mail boxes, and website content, might be found under `/var`.

The Linux directory hierarchy

- You can see the directory structure with the “tree” command
- Examples:
    - **tree** /home
    - tree /usr/local
    - tree /

Karel de Grote Hogeschool

The working directory

2

Karel de Grote Hogeschool

The working directory

- When you interact with a computer with a command line interface you are always in a certain directory on the system
- This is called the “working directory”
- You can always find out the working directory with the “**pwd**” command (Print Working Directory)
- You can change the working directory with the “cd” command
  cd /tmp -> go to the tmp directory under root
  cd /home/kris -> go to the home directory of user kris

Karel de Grote Hogeschool

Listing files in a directory

- Use the “ls” (list) command to see all files in a directory
- **ls** -> shows all files and directories in the current working directory
- **ls /etc** -> shows all files and directories in the /etc directory
- **ls -l** -> shows more information about the files and directories
- **ls -a** -> also shows the “hidden” files. Their file names start with a dot
- **ls -la** -> combine the options l and a
- the files are shown in alphabetical order

Karel de Grote Hogeschool

Absolute and relative paths

Karel de Grote Hogeschool

Absolute and relative paths

- A path can be “absolute” or “relative”
- An **absolute** path does not depend on the current working directory
  - it starts from the root directory (/)
  - `ls /var/log/messages`
- A **relative** path is relative to the current working directory
  - it never starts with a /
  - `ls log/messages`
- Avoid using paths with white spaces (use \ -esc character)

Karel de Grote Hogeschool

Special directories

In order to be able to address any folder using a relative path, two special directories always exist:

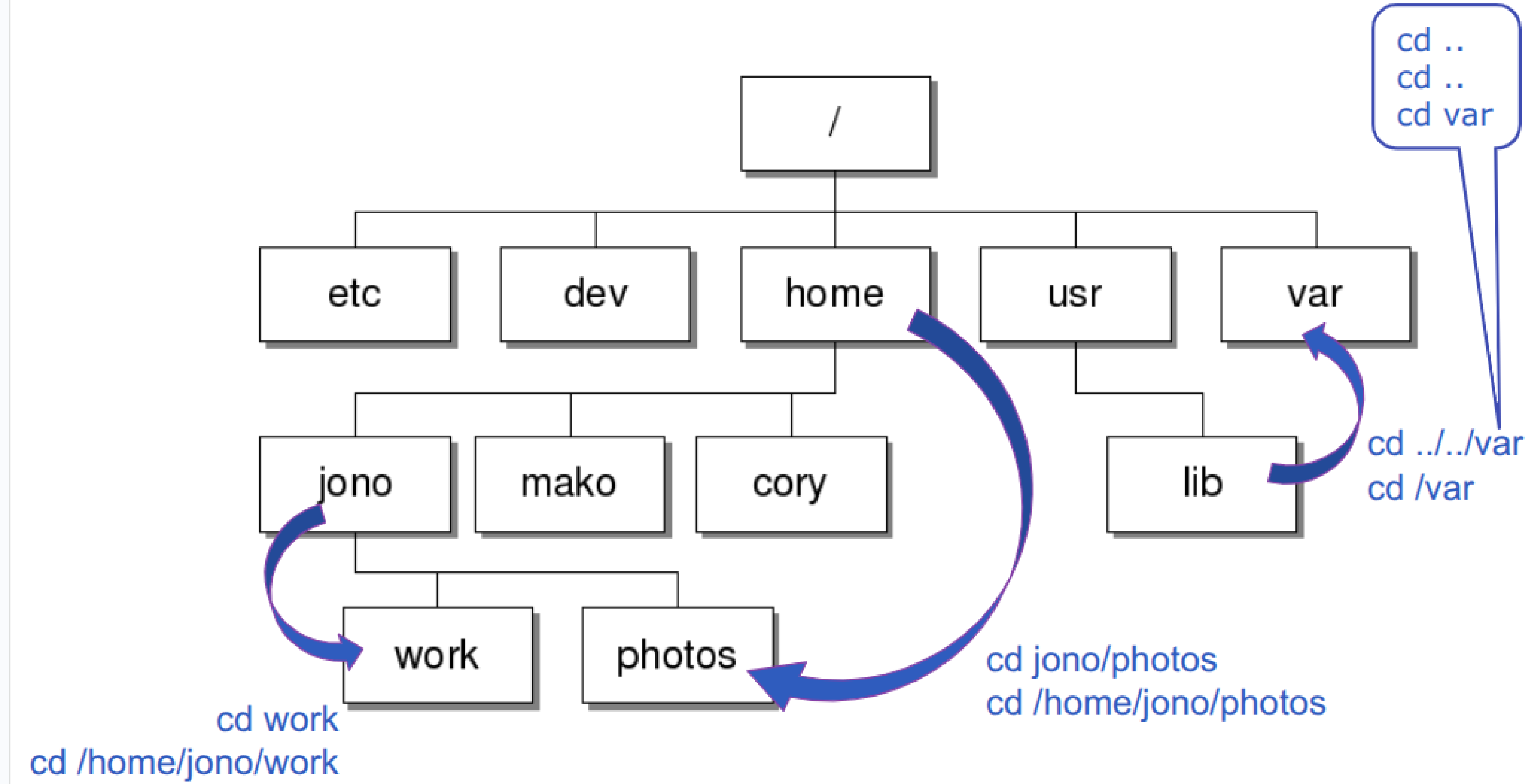
- dot (.) always refers to the current working directory
- two dots (..) always refers to the directory above the current working directory

These are used in <u>relative paths</u>

Examples:

- **cd ./Documents** -> go to the folder Documents in the current directory (this is equivalent to cd Documents)
- **cd ../../tmp** -> go two directories up from the current directory and then into the tmp directory

Karel de Grote Hogeschool



```
cd ..
cd ..
cd var
```

```
cd ../../var
cd /var
```

3

```
cd jono/photos
cd /home/jono/photos
```

```
cd work
cd /home/jono/work
```

Trailing slash in paths

You can use a trailing slash / at the end of a path to make it clear that it is a directory (and not a file)

- `/var/log/` → log is a directory (trailing slash)

  `/var/log` → log is a directory (but it could also be a file)

- `ls -lh /var/log/messages` → gives information about file messages

- `ls -lh /var/log/messages/` → returns an error (the folder messages doesn't exist)

Karel de Grote Hogeschool

Home directories

- every user has a home directory
- the home folder can always be referred to as “~”
- e.g.

  cd ~ -> takes you to your home directory

  cd ~/Documents -> takes you to your Documents directory

  cd ~adam -> takes you to the home directory of the user adam
- ~ is a shortcut for /home/<username> and is therefore an absolute path (even though there is no / is the beginning)

Karel de Grote Hogeschool

4

Exercises

Karel de Grote Hogeschool

Exercises

- KdG
  - Ex0301
  - ex0302
  - ex0303
- RedHat
  - ch03s02
  - Ch03s04
  - ch03s06

Karel de Grote Hogeschool

Creating and deleting directories

5

Creating directories

You can create a directory with the “mkdir” command:

- **mkdir** test -> relative path
- mkdir /home/kris/test -> absolute path
- mkdir test1 test2 -> creates 2 directories

But the underlying directory must already exist:

- mkdir /home/kris/blah/test -> gives an error if /home/kris/blah does not exist

Solution:

- **mkdir -p** /home/kris/blah/test -> creates all necessary directories if they don’t exist

Deleting directories

You can delete a directory with the “rmdir” command:

- **rmdir** blah/test/ -> only deletes “test” (we use a relative path here)
- rmdir /home/kris/blah -> only deletes “blah” (we use an absolute path here)

The directory has to be <u>empty</u>!

You can also remove a non-empty directory with the “rm” command:

- **rm -r** blah -> -r stands for “recursive”

Karel de Grote Hogeschool

Creating and deleting files

6

Karel de Grote Hogeschool

Creating files

You can create an empty file with the “touch” command:

- **touch** my_script.sh

The touch command changes the “modify date” of a file when it exists.

If it does not exist it is created.

You can add a line of text to a file using the “echo” command (see also later):

- **echo** "Some text" **>>** my_script.sh

Deleting files

You can delete a file using the “rm” commando:

- **rm** my_script.sh

Linux does <u>not</u> ask for confirmation!

The file does <u>not</u> go to a trash bin!

What do the options **-f** en **-i** do?

What would the command “rm -rf /” do? Do <u>not</u> try!

Karel de Grote Hogeschool

7

Copying and moving files

Karel de Grote Hogeschool

Copying files

Before changing a system file it is a good idea to first make a backup copy of it.

This can be done with the “cp” command:

cp webserver.conf webserver.conf.old

You can copy multiple files in one command:

cp bestand1 bestand2 /tmp/

You can also copy a complete directory with “cp”:

cp -r Documents/ /tmp/Documents.backup

Karel de Grote Hogeschool

Moving files

You can move files with the “mv” command:

mv my_script.sh ../bin/

This command is also used to rename a file:

mv my_script.sh backup_script.sh

Or the combination(move and rename):

mv my_script.sh /tmp/backup_script.sh

Karel de Grote Hogeschool

Exercises

- KdG
  - ex0304
- RedHat

8

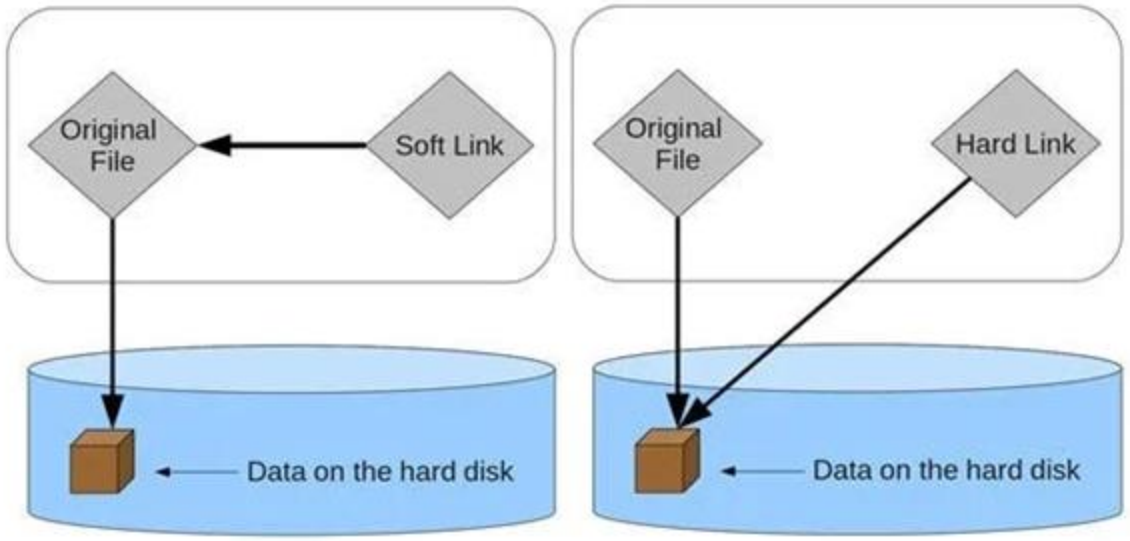
Symbolic and hard links

Karel de Grote Hogeschool

Links

There are 2 ways to create a link to a file in Linux.

A link is a reference to another file (in Windows this is called a “shortcut”)



Source: Soft Link and Hard Link in Linux (Medium)

Symbolic Links

“Symbolic” of “Soft” links are made with the “ln -s” command. Try out the following code:

```
echo "This is a test" >>test
ln -s test test_slink
ls -li
cat test
cat test_slink
echo "This is a test" >>test
ls -li
cat test_slink
stat *
rm test
ls -li
cat test_slink
rm test_slink
```

Hard Links

“Hard” links are made with the “ln” command. Try out the following code:

```
echo "This is a test" >>test
ln test test_hlink
ls -li
cat test
cat test_hlink
echo "This is a test" >>test
ls -li
cat test_hlink
stat *
rm test
ls -li
cat test_hlink
rm test_hlink
```

Links

symbolic link	hard link
can refer to another medium(other disk, partition, network drive)	always on the same medium (that supports these links)
can become invalid (when the original disappears)	no difference between original and link
can refer to a file or directory	can only refer to a file
is a bit slower (the link has to be followed)	same speed as with original file

Exercise

1. Create a symbolic link and a hard link on your Desktop that refers to your Documents directory
2. Go to these directories on the command prompt
3. Check in which directory you are using “pwd”

Karel de Grote Hogeschool

Exercises

- KdG
  - ex0304
- RedHat

9

Shell expansions

Karel de Grote Hogeschool

Pattern matching (aka Globbing)

You can specify multiple files or directories using “wildcards”

- * means 0 or more random characters
- ? means 1 random character
- [abc] means only one of the characters a, b, or c
- [[:upper:]] means an uppercase character

See globbing, Try:

```
$ echo /var/log/m*
```

Karel de Grote Hogeschool

Pattern matching (aka Globbing)

Examples:

`ls /etc/pas*` -> shows all filenames that start with “pas” in the directory /etc

`ls /etc/*cd*` -> shows all filenames that contain “cd” in the directory /etc

`ls /etc/*.conf` -> shows all filenames that end in “.conf” in the directory /etc

`ls /etc/???.conf` -> shows all filenames that have exactly 3 characters and end in “.conf” in the directory /etc

`ls /etc/[afc]*.conf` -> shows all filenames that start with an ‘a’, ‘f’, or ‘c’ and end in “.conf” the directory /etc

`ls -ld /etc/[[:upper:]]*`-> shows all filenames that start with an uppercase character in the directory /etc

Karel de Grote Hogeschool

Variable expansion

```
firstname=john
```

no spaces!!!

```
lastname=doe
echo "my name is $firstname $lastname"
echo "my name is $firstname_$lastname"
```

Will not work (_$)

```
echo "my name is ${firstname}_${lastname}"
```

good practice to use curly braces.

*You can protect from expansion using "escaping" with *

```
echo the variable \$firstname contains $firstname
```

variables don't have to be declared
they are always of type String

Variable expansion

```
fullname=john doe
echo "my name is ${fullname}"
```

What is wrong with this code?

What is the solution?

Karel de Grote Hogeschool

Brace expansion examples

```
$ touch {index,contact,home}.html
   !!No white spaces!!!

$ touch redhat{6..9}

$ cp my_config{,.old}
```

Brace expansion

More advanced usage: create a directory structure

```
vhosts/
          www.ansible.com/
                    cgi-bin/
                    html/
          www.open-shift.com/
                    cgi-bin/
                    html/
          www.redhat.com/
                    cgi-bin/
                    html/
```

```
$ mkdir -pv vhosts/www.{ansible,open-shift,redhat}.com/{cgi-bin,html}
$ tree -f vhosts
```

Karel de Grote Hogeschool

Command substitution

```
echo Today is $(date +%A).
```

```
touch  my_config_$(date +%F)
```

10

Exercises

Karel de Grote Hogeschool

Exercises

- KdG
  - ex0309
  - ex3099
- RedHat
  - ch03s06
  - ch03s08
  - ch03s11