C2110 *UNIX and programming*

Lesson 2 / Module 3

PS / 2020 Distance form of teaching: Rev2

Petr Kulhanek

kulhanek@chemi.muni.cz

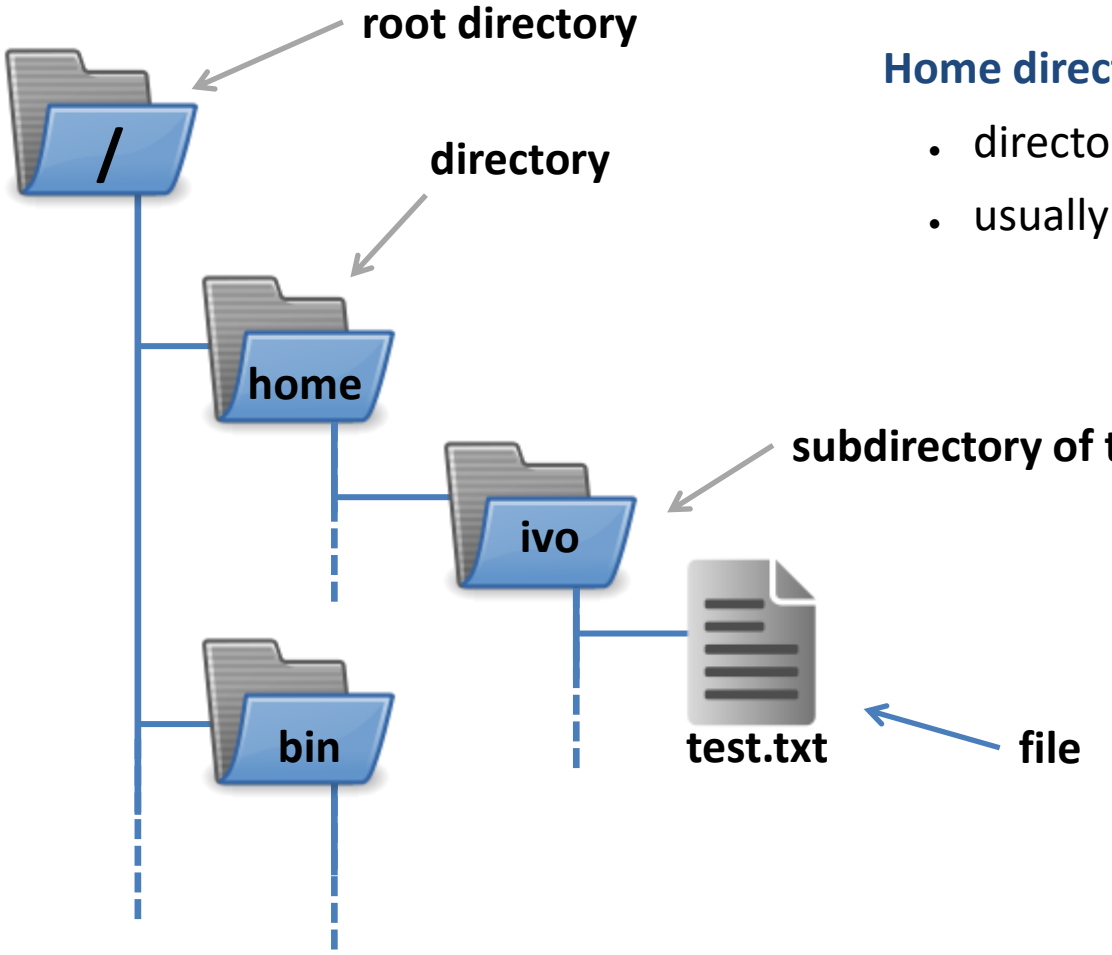
National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

File System

- Structure
- Differences from MS Windows
- Relative and Absolute paths
- Wildcards
- Basic Commands

File System Structure

UNIX uses **hierarchical** directory **file system** composed of directories (folders) and files. All directories and files are located in a **single root directory (/)**.



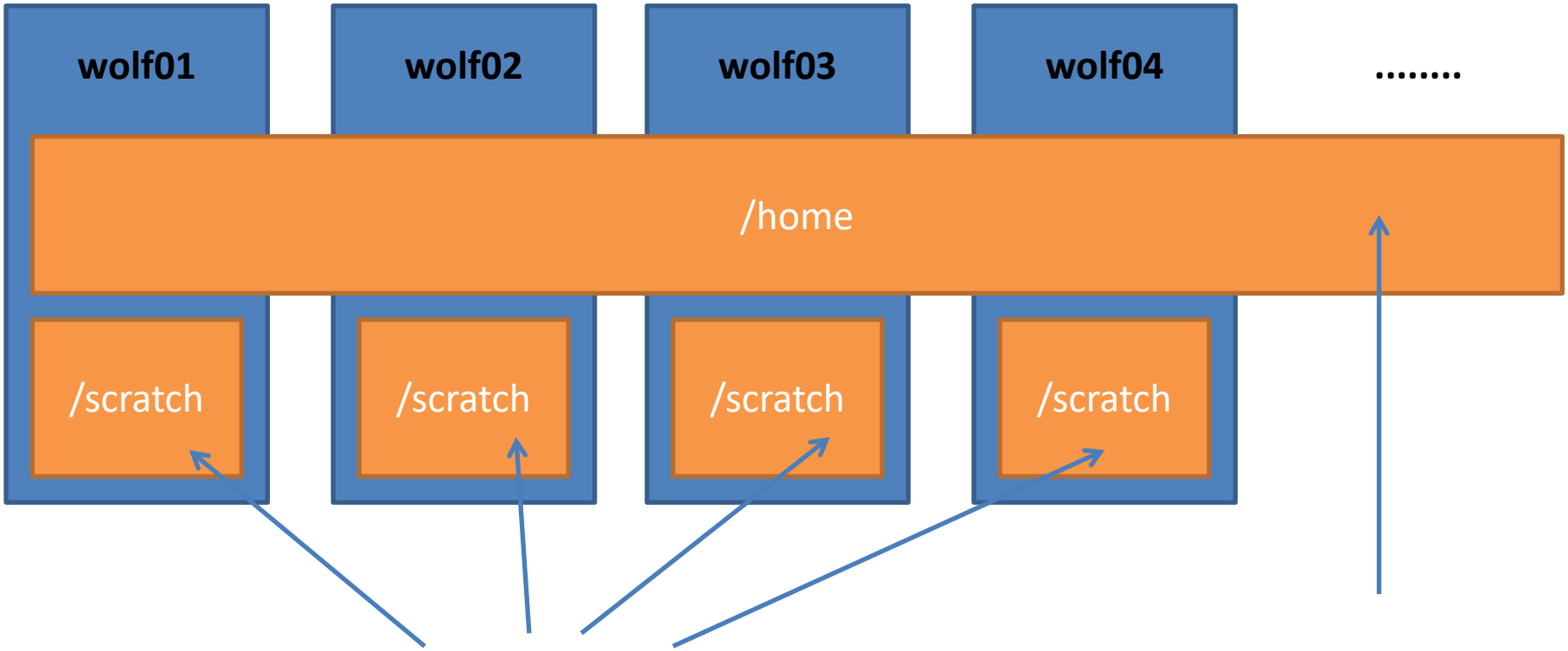
Home directory:

- directory used for **data and user settings**
- usually in the path **/home/username**

Comparison with MS Windows

Property	Linux (ext2/ext3/ext4)	MS Windows (FAT32, NTFS)
Disk Partitions	No Disk partitions are mounted as directories.	C:, D:, etc. However, it is also possible to mount them as directories (ntfs).
Names	Case sensitive	Not case sensitive
Separation of names	Slash	Back slash
Access rights	Yes POSIX	Yes (only NTFS) ACL
Equipment (hardware)	As special files.	No

File System on the WOLF Cluster

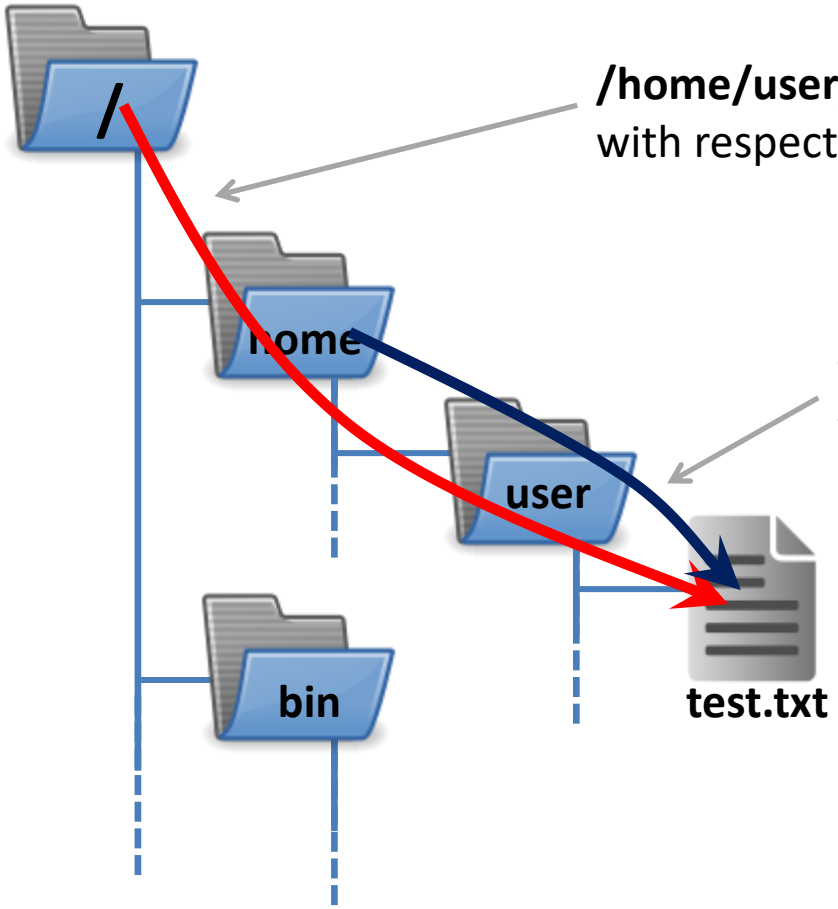


Different content on each node.
Data on volume /scratch up **are not backed up** and can be **deleted at any time** without prior notice.
Capacity **is not limited** by quota per user.

Shared content on all nodes of the WOLF cluster.
Data are **backed up**. Backups are available in form of snapshots in the directory
/backup/<date>/WOLF/wolf.ncbr.muni.cz/home
Capacity per user is limited by quota **1.5 GB**.

Identification of directories and files

Path to a directory or file may be written as **absolute** or **relative**. Directory and file names are separated by **slash /**.



/home/user/test.txt - absolute path identifying file test.txt with respect to the root directory

user/test.txt - relative path identifying the file test.txt with respect to the current directory (in this case it is the directory /home)

Types of Paths

Paths in the file system (regardless of OS) can be **relative** or **absolute**. The meaning of relative paths depends on the current working directory, while the meaning of absolute paths is independent of the current working directory.

The absolute path it is always listed relative to the root or home directory. So it has to start with either a slash **/** or tilde **~**.

/home /kulhanek/Documents/domaci_ukol.txt

Using the tilde:
~ home directory of the logged in user
~username the home directory of user **username**

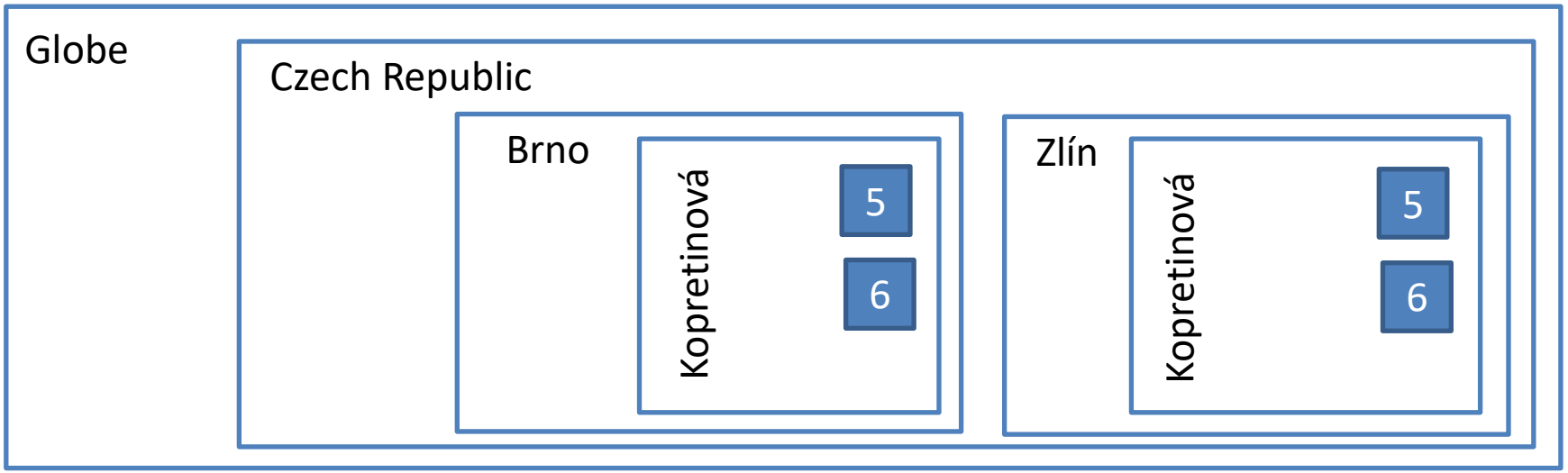
Relative path is the path listed to the current/working directory. (The absolute path of the working directory can be obtained with the command **pwd**.)

../alois/Documents

Special directory names:
. (dot) current directory
.. (two dots) parent directory

Paths vs postal addresses

A very close parallel to paths are, for example, postal addresses:



Relative address:

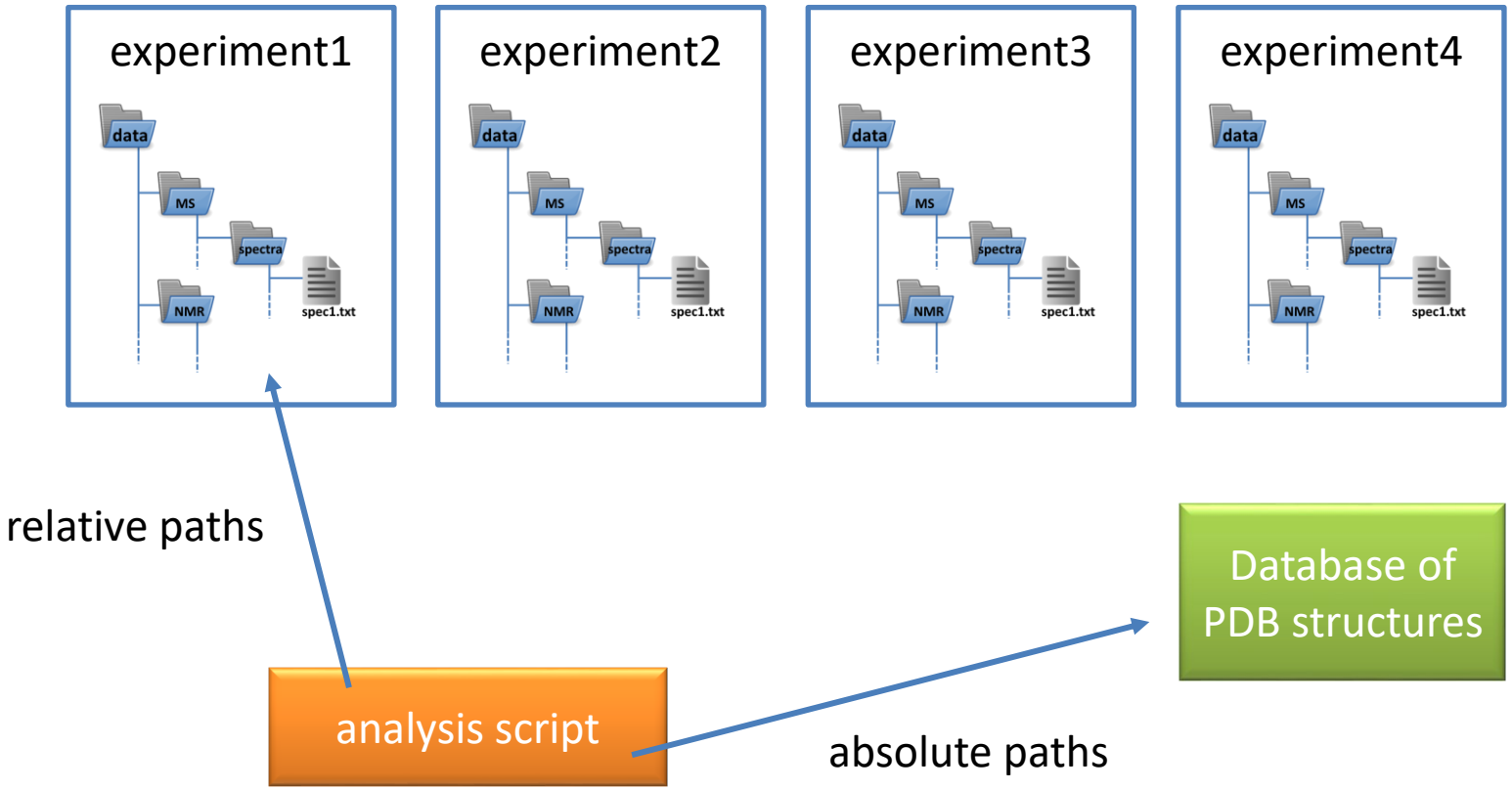
Kopretinová 5

Absolute address:

/Globe/Czech Republic/Brno/Kopretinová 5

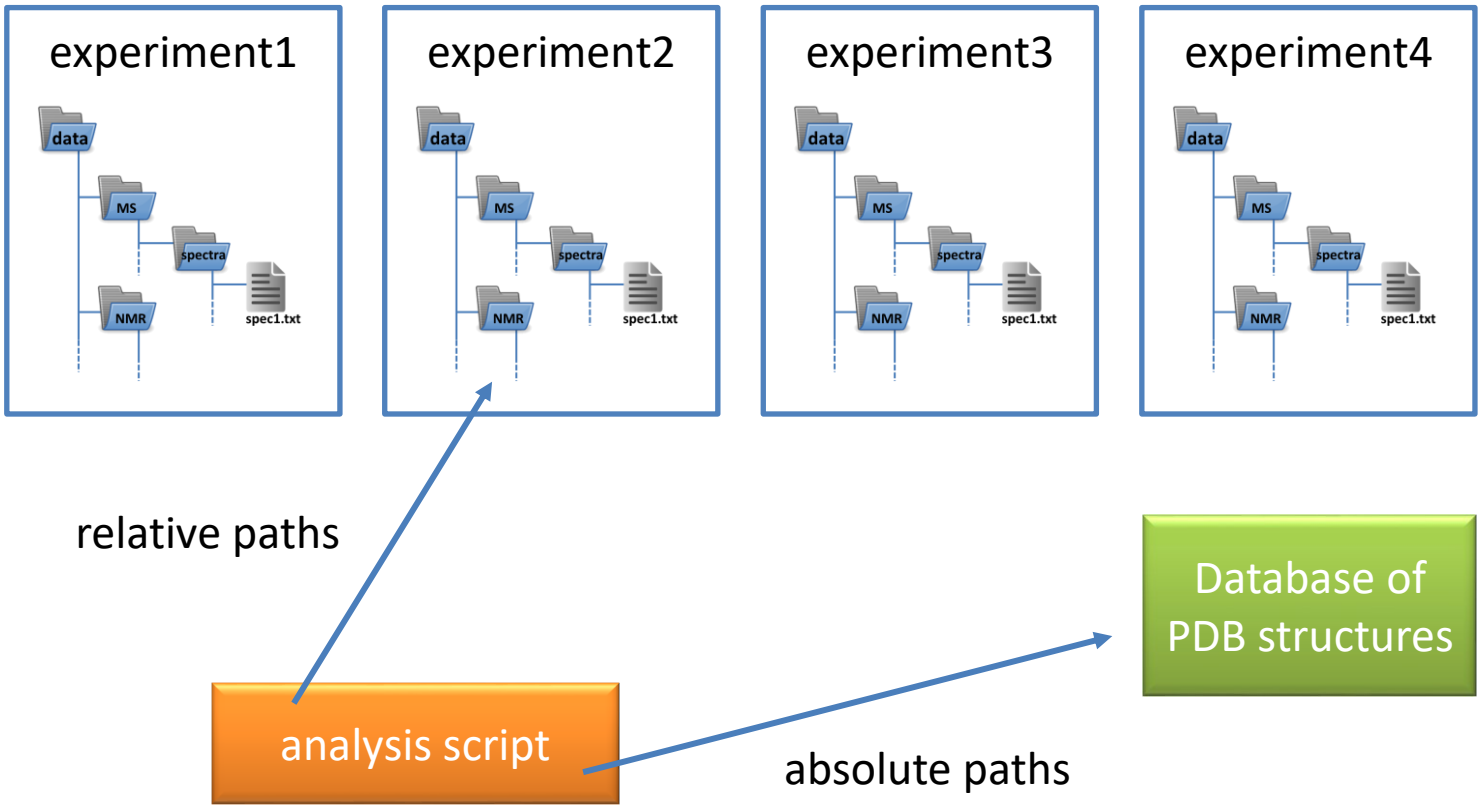
Use of relative and absolute paths

In everyday work, both types are often used, even in parallel.



Use of relative and absolute paths

In everyday work, both types are often used, even in parallel.



Wildcards in File Names

Wildcards (special characters) in file or directory names:

- ***** - anything in the name (without hidden files)
- **?** - one character in the name
- **[]** - range (one character) in the name, e.g., [ajk], [a,j,k], [a-j]

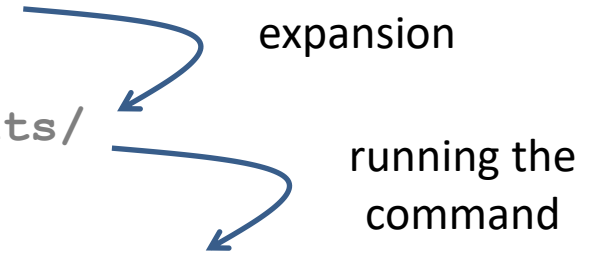
Expansion of wildcards is performed by **shell** (command line access environment) **before starting** the command itself. Expansion can be prevented by enclosing the name in quotation marks or by using a backslash before the special character. In this case, the expansion can be performed by a running command (e.g., find).

Example:

The current directory contains files 01.pdf and 02.pdf

Entered command: **`$ cp *.pdf Documents/`**

Added command: `$ cp 01.pdf 02.pdf Documents/`



Special Characters in File Names

Examples:

```
$ cp *.pdf Documents/
```

copies all pdf documents from the current directory to the subdirectory Documents

```
$ rm *
```

deletes all files in the current directory (except directories and hidden files and directories)

```
$ mv A? Tmp/
```

moves files with a name beginning with a letter "A" and containing two characters into the directory "Tmp"

Special characters can be **used concurrently** in both directory and file names:

Examples:

```
$ cp ~/task[1,4,5]/*.pdf Documents/
```

copies all pdf documents from subdirectories ukol1, ukol4 and ukol5 in the home directory to a subdirectory Documents

Basic Commands

File system (basic commands):

cd	**changes the current working directory**
pwd	**lists the path to the current working directory**
ls	**lists the contents of the directory**
mkdir	creates a directory
rmdir	deletes directory (must be empty)
cp	copies a file or directory
mv	moves a file or directory
rm	deletes a file or directory
find	searches for files or directories

"**Golden Trinity**"- I need to know how to get there (**cd**). where I am, if I do not know it (**pwd**) and what is there (**ls**)

Useful tips:

$ cd the command without the specified argument sets the home directory as the working directory

It is recommended to have **at least two terminals opened in the source and destination directories**, to use **automatic completion** (TAB), and to check the contents of directories before and after the operation.

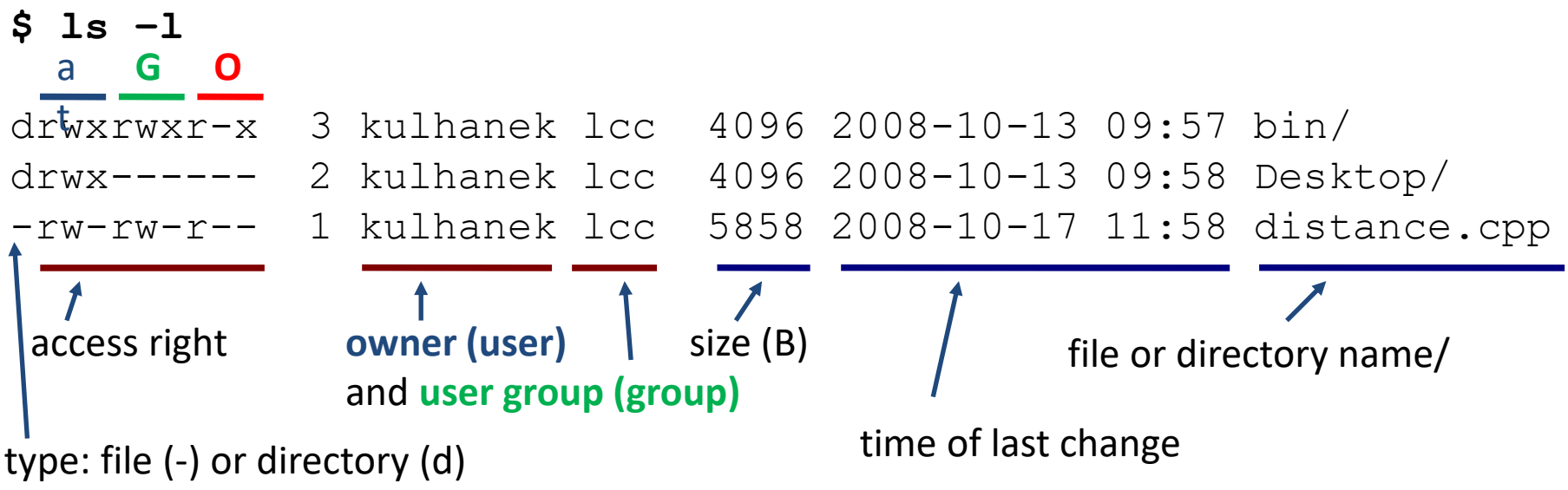
Access Rights - Prelude

Access rights determine what operations a user can perform on files or directories in the file system.

Access rights:

r	ability to read the file	to list the contents of the directory
w	ability to change file	change directory content (create or delete a file or directory)
x	ability to run the file	ability to enter the directory

Each file or directory has a designated owner and user group. Access rights are listed separately for **file owner (u)**, **user group (g)** and **other users (o)**.



Creating Directories

- **Creating a directory**

```
$ mkdir direktory_name
```

- **Creating a nested directories**

```
$ mkdir -p name_dir1/name_dir2/name_dir3
```

option -p (parents) causes parent directories to be created if they do not exist

Copying

- **Command cp is used to copy**

```
$ cp file1 file2
```

creates a copy of file "file1" named "file2"

```
$ cp file1 file2 file3 directory1/
```

copies files "file1", "file2", "file3" into the directory "directory1"

```
$ cp -r directory1 directory2
```

creates a copy of directory "directory1" named "directory2"; if directory "directory2" already exists, creates a copy of directory "directory1" as a subdirectory of directory "directory2"

```
$ cp -r file1 directory2 file3 directory1/
```

copies files "file1", "file3" and directory "directory2" to directory "directory1"

Option -r (recursive) must be used to copy the contents of directories.

Moving

- **Command mv is used to move or rename**

```
$ mv file1 file2
```

renames file "file1" to "file2"

```
$ mv file1 file2 file3 directory1/
```

moves files "file1", "file2", "file3" to directory "directory1"

```
$ mv directory1 directory2
```

renames directory "directory1" to "directory2"; if directory "directory2" already exists, moves directory "directory1" to directory "directory2"

```
$ mv file1 directory2 file3 directory1 /
```

moves the files "file1", "file3" and the directory "directory2" to the directory "directory1"

Removing

- **Command rm is uded to remove**

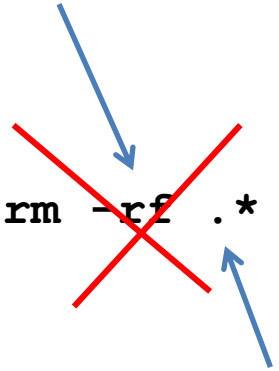
```
$ rm file1
```

deletes file "file1"

```
$ rm -r directory1
```

deletes directory "directory1"

recursion (r) and without query (f - force)



`.*` -> `..` (removing even upwards)

Searching for Files

You can use the command **find** to search for files.

if not specified, it is searched in the current directory

```
$ find [where] what
```

search is recursive (default)

Search query (**what**) is composed of subqueries that can be joined by logical operators.

Frequently used queries:

-name *pattern*	finds all files that have a name *pattern* *pattern* may contain special characters: *,?, [] **(when using special characters we state *pattern* in quotes)**
-type *C*	finds all files of type ***C*** (file, directory, etc., for list of types see man find)

Logical operators:

-and	left and right queries are met simultaneously
-or	left or right query is met

File Search - Examples

```
$ find /home/ -name '*.txt'
```
in directory /home/, finds all files that end with .txt

```
$ find ~kulhanek -Name '*.txt' -or -name '*.hpp'
```
in directory /home/kulhanek, finds all files that have a end with .txt or .hpp

```
$ find -name 'D*' -and -type d
```
in the current directory, finds all subdirectories whose names begin with the letter D

Hidden Files and Directories

Names of **hidden** files or directories **start with a dot**. They are normally not displayed, but they can be listed using the command **ls -a**. Special characters *, ? and [] do not include hidden files in the normal context.

Hidden files contain system configuration, and if you don't know what you're doing, don't delete or change them.

Command Line Query Format

You can change the appearance of the command line query. The system variable used for this is **PS1** (man bash). If the current format does not suit you (it displays the name of the last directory from the current path), you can change the look as follows:

Default settings:

```
$ PS1="[\u@\h \W]$ "
```

Adjusted settings:

```
$ PS1="[\u@\h \w]$ "
```

uppercase and lowercase letter w

The adjusted setting will display **whole path** to the working directory.

The change will only take effect in the terminal where it was made. The settings can be made permanent by inserting the command at the end of **hidden file ~/.bashrc** on a separate line. Use a text editor (**gedit** or **kwrite**) to change the contents of the file. The change in settings will be reflected in the newly opened terminals.

Exercise 1

1. Log in to the wolf02.ncbr.muni.cz workstation.
2. Create directories in the home directory: **Documents/C2110**
3. Copy the lectures into the C2110 directory in pdf format from directory: **/home/kulhanek/Documents/C2110/Presentations**
4. Create a directory **studmat** in your home directory
5. Copy the presentation **C2110-CZ-L02.pdf** from your directory **Documents/C2110 t**o the directory **studmat** and name it **presentation.pdf**
6. Create a directory **/scratch/username/test**
7. Copy file **prezentace.pdf** to directory **/scratch/username/test**
8. In the directory **/scratch/username/test**, rename the presentation to **lesson2.pdf**
9. Log in to the wolf03.ncbr.muni.cz workstation
10. Is the lekce2.pdf file present in the directory **/scratch/username/test**? Discuss the observation.
11. Delete directories **/scratch/username/test** and **~/studmat**

Try to learn to use:

- multiple terminals
- auto complete (TAB key)
- simplified copying: mark with the left button / insert with the middle mouse button
- command line history

username - your username

Homework

- **Practicing Commands**

Homework

1. Log in to the WOLF cluster workstation.
2. Create a directory in your home directory **Data**
3. Copy the contents of the directory **/home/kulhanek/Documents/C2110/Lesson02** including subdirectories to the directory **Data**
4. Find all files with the extension **.cpp** which are located in the directory **Data** (write the file names on the screen)
5. Create a directory **Headers** in the directory **/scratch/username**
6. Copy all files from the directory **/home/kulhanek/Documents/C2110/Lesson02/dev/src** that have an ending **.h** to the directory Headers
7. Move all files from the directory **/home/kulhanek/Documents/C2110/Lesson02/dev/src** that have an ending **.cpp** to the the directory Headers. What really happened and why?
8. What is the size in B and kB of file **/home/kulhanek/Documents/C2110/Lesson02/dev/src/GraphicsSetup.cpp**
9. In the directory **Headers** delete all files with the extension **.h** and containing the word **Graphics** at the beginning of the file name
10. Delete the directory **Headers**