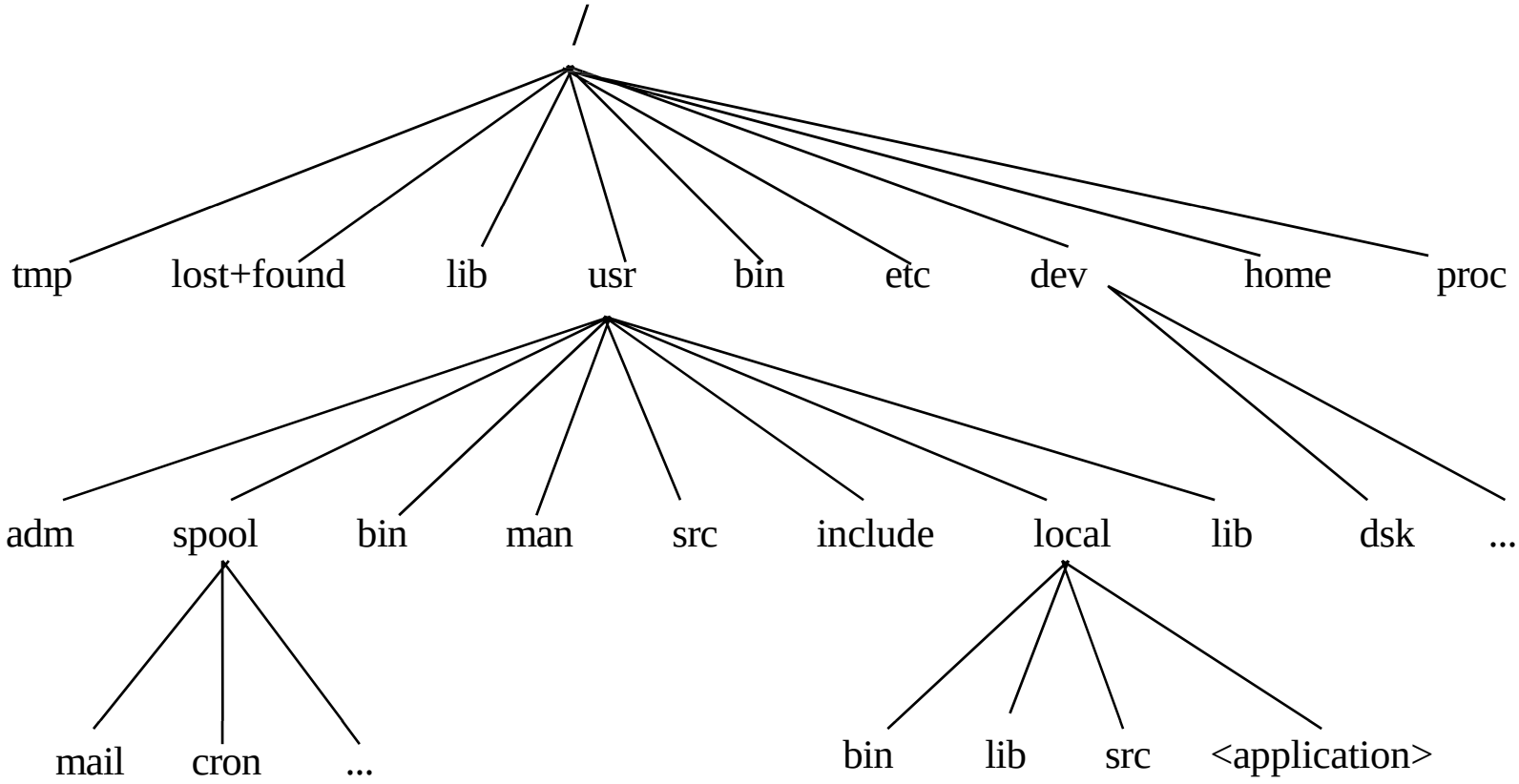
The file system

The file tree



The file tree

- Hierarchical structure of files and directories
- The root is denoted by `/`.
- Paths are separated by `/`.
- Object names are case-sensitive sequences of up to 255 characters.
- It is preferable not to use the characters: ?, *, &. `, ',",<,>.
- advisable to limit to:A to Z , a to z, 0 to 9, the underscore character ` _`, the dash `-` and the dot `.`.

The file tree

•The dot (.) as the first character of a name signifies a hidden file.

•Avoid names containing accented characters or blanks.

•The dot (.) does not necessarily separate the name from its extension.

•**Examples**:

report.ps.gz
postgresql-2.3.6-src.tar.Z
README

The main directories

- /boot: contains files required for system startup.
- /bin , /sbin, /usr/bin, /usr/sbin,...: contains binary files
- /etc: contains system configuration files
- /home: contains home directories for single users
- /root: contains the administrator's home directory .
- /usr: contains the rest of the system's data and programs

The paths

- Each object (file or directory) can be **accessed in two ways**

possible **ways** :

- **Absolute path**: specify the name of the object by giving its complete access path starting from the root of the tree (/)

Example: /usr/local/seminars/learning/text1

- **Relative path**: specify object name with path relative to current directory.

Example: seminars/learning/text1

Represents the relative name of text1 with respect to the

current directory /usr/local

Handling directories

- **mkdir** Creates a directory
- **rmdir** deletes an **empty** directory
- **cd** change directory
- **pwd** displays working directory.

Create a directory

• mkdir: Creates an empty directory

•You need to take into account where you are when creating a directory

if the directory_name is relative (i.e. doesn't start with /).

•You can create a whole branch of directories

•**Syntax** :

$ mkdir [option] rep1 rep2...repn

•**Examples**:

$mkdir folder

$mkdir -p grandpere/pere/son

Delete a directory

- rmdir: Remove Directory, deletes an empty directory
- **Syntax**
- $rmdir directory_name
- Example:
- **rmdir p6**

Changing a directory

- cd : Change Directory, go to another directory
- **Syntax**

$cd [options] [directory_name]

- **Examples:**

$cd go to connection directory.

$cd .. go to parent directory of current directory.

$cd /usr/share/go to /usr/share/ directory

Where am I?

- **pwd : Print Working Directory: Displays the absolute name of the current directory**
- **For example**:
- **cd /usr/share/doc**

$ cd ..

$ pwd

/usr/share

File handling

- **ls** displays a file's attributes**.**
- **cp** copy files**.**
- **touch** creates afile
- **file** gives the type a file
- **rm** delete files**.**
- **mv** rename or move a file

The ls command

•**ls**: display the **list** of objects in the specified location.

Syntax :

$ ls [options] [parameters]

Examples:

$ls

$ls -a

$ls -al

$ls -alh

The ls command: Examples

$ ls

Rep text1 text2 text3

$ ls –a displays hidden files

./ ../ .cache Rep text1 text2 text3

$ ls -l

drwxr-xr-x	2	user1 group1	89	Sep4 16:34	Rep
-rwxr--r--	3	user 2 group1	71	Sep 4 16:34	text1

ls command output

The -l option is present for each object displayed:

1) object type

2) The 9 characteristics of object protection.

3) The number of physical links pointing to the object (not part of our course)

4) User name (owner).

5) The group to which the owner user belongs.

6) Object size in bytes.

7) Date and time of last modification.

The cp command

- **cp**: copy files from a source to a destination

.**Syntax** :

$cp [options] source destination

- If **destination** is a directory, then copy source to this directory, keeping the same name.
- If **destination does** not exist, then create a destination file with the same content as the **source** file.

•If the **destination** file exists, confirmation will be requested before replacing the existing file

The cp command

- **Some options**:
- -i: to request confirmation when an action has been performed
- -R allows recursive copying of directories (all their contents)

The mv command

•**mv**: move or rename a file.

•**Syntax** :

$mv [options] source destination

Options :

•i: request confirmation

Example1:To move a file to another directory and give it a new name, type the following:

```
mv intro manual/chap1
```

This moves the `intro` **file** to the **`manual/chap1`** directory. The name **intro** is removed from the current directory, and the same file appears as `chap1` in the `manual` directory.

The mv command

Example 2:To move a file to another directory, keeping the same name, type the following:

`mv chap3 manual`

This moves `chap3` to `manual/chap3`.

Example 3:

To rename a file, type the following:

`mv appendix apndx.a`

This renames the `appendix` file to `apndx.a`.

If a file named `apndx.a` already exists, its old contents are replaced with those of the `appendix` file.

The rm command

- rm : ReMove : delete a file

- Caution! The file cannot be recovered

- **Syntax** :

$rm [option] file

- **Option** :

-i confirmation request (desirable)

-f to force deletion

-R or –r to delete the contents of a directory recursively

The rm command

Example 1:To delete the file named `myfile`, type the following:

```
rm myfile
```

Example2: To delete all the files in the `mydir` directory, one by one, type the following:

```
rm -i mydir/*
```

After each file name displays, type y and press Enter to delete the file. Or to keep the file, just press Enter.

The touch command

- **touch**: Creates an empty file
- if the name doesn't exist, it will be created
- If the name exists then the modification date will be changed.

- **Syntax** :

$ touch file_name

- **Example**:

$touch linux.txt

$ls

linux.txt