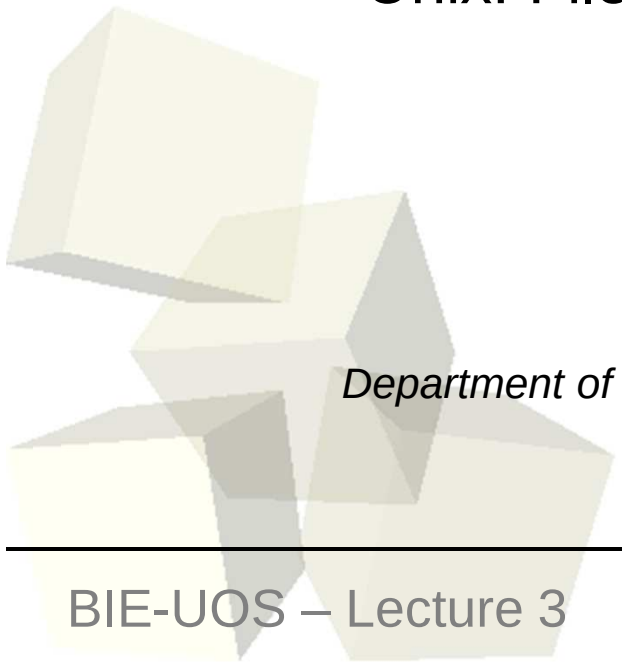
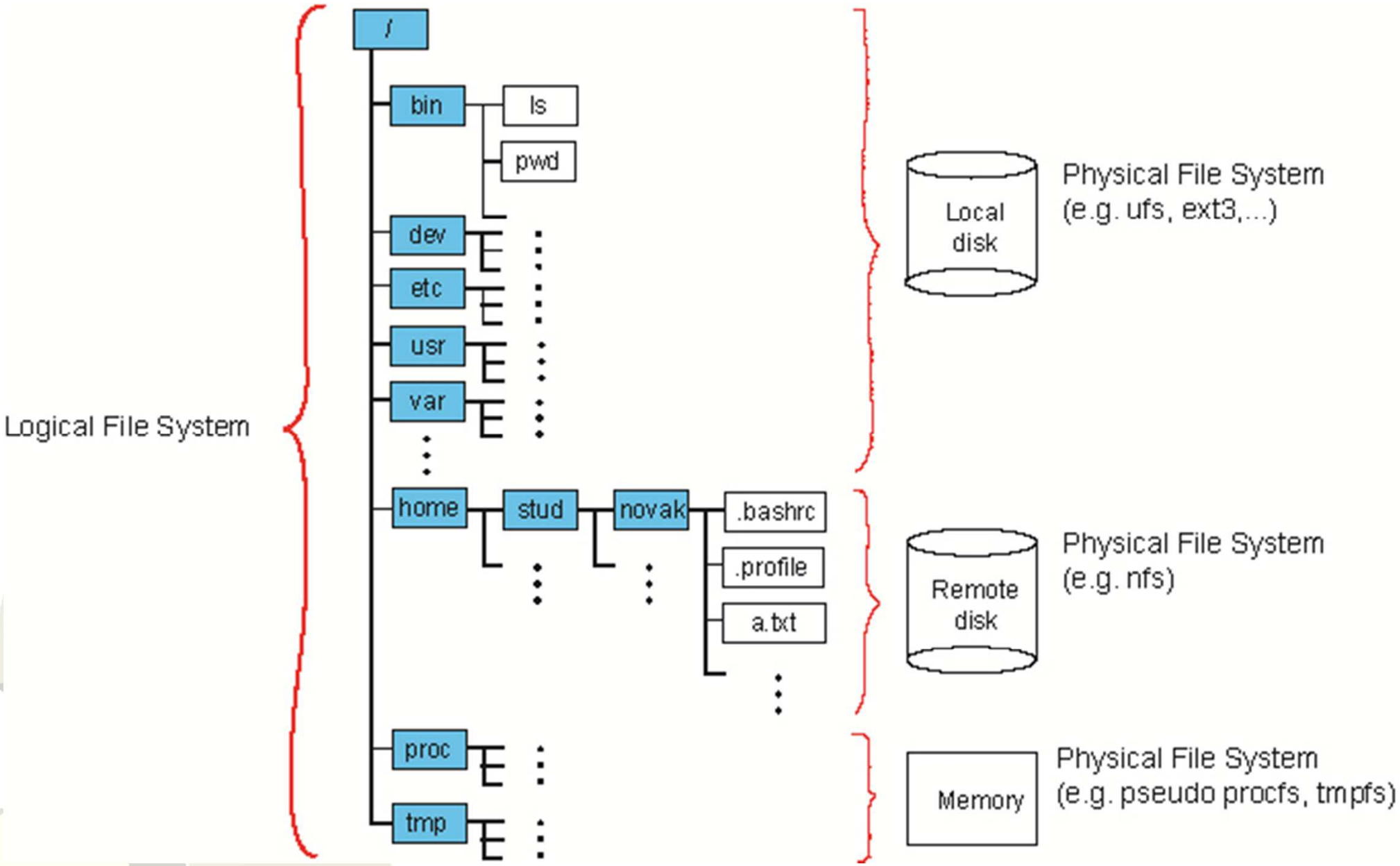
Lecture 3

Unix: File System. Basic file/directory commands.



Department of Computer Systems FIT, Czech Technical University in Prague

File System (FS)



Important Directories

/bin	User commands
/sbin	Administrative utilities
/dev	Special files
/etc	Administrative and configuration files
/home	Home directories
/lib	Shared libraries
/tmp	Temporary files/
/opt	Root of a subtree for add-on application packages
/usr	Platform-dependent and platform-independent sharable files
/var	Root of a subtree for varying files

File = name (names) + attributes + data

- **File name**
  - Maximal size (implementation dependent)
  - Code depends on implementation (ASCII, UTF8,…)
  - Any characters except of character of '**/**'

  - **Name beginning with dot** (hidden file/directory):
    - nenahrazují na příkazové řádce při použití znaků ***** a **?**
    - příkaz **ls** je nevypisuje (lze ale vynutit přepínačem **–a**)
  - **Name dot (.) and double dots (..)** are reserved for
    - **.** – working directory
    - **..** – parent directory

- **File Attributes** (can be display by command `ls -l`)
  - File type:

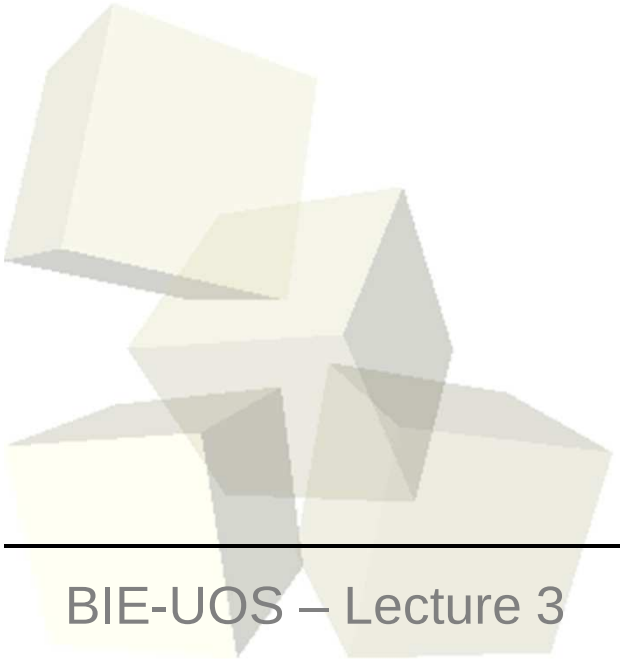
d	directory
-	regular file
c	character device
b	block device
l	symbolic link
p	named pipe

  - File Owner (user and group)
  - Access permissions (r – read, w – write, x – execution, setXid, ACL,…)
  - Time (creation, modification, access)

File

- **Data**
  - File content is saved in data blocks.

- **File access**
  - By system calls: **`open()`**, **`close()`**, **`seek()`**, **`read()`**, **`write()`**, **`stat()`**, …
  - By OS commands: **`more`**, **`less`**, **`cp`**, **`rm`**, **`mv`**, **`ln`**, …



Directory

- **Absolute path**
  - It stats in the root directory **/**
  - It contains the hierarchy of directories between root directory **/** and given file

```
/home/year2010/group12/Smith
```

- **Working directory**
  - It can be display by command **pwd**
  - Its value is saved in shall variable **PWD**
  - It can be change by command **cd new_working_directory**
  - Every process can have different working directory

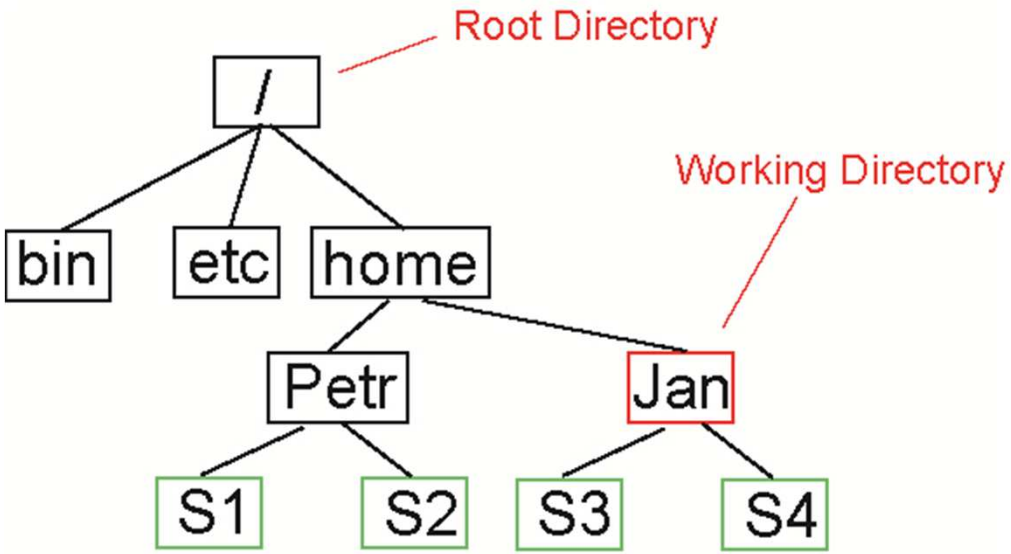
Directory

- **Relative path**
  - It is a path relative to the working directory **$PWD**
  - It contains the hierarchy of directories between **$PWD** and given file

```
PWD=/home/year2010/group15
                    ../group12/Smith
```

- **Home directory**
  - Every user has its home directory.
  - During login process working directory is set to the home directory.
  - Its value is saved in shell variable **HOME.**

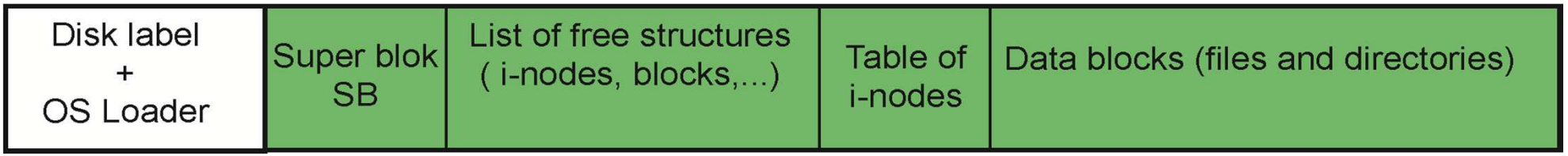
Example



`/home/Petr/S1`	absolute path to S1
`./../Petr/S1`	relative path to S1
`../Petr/S1`	relative path to S1
`/home/Jan/S4`	absolute path to S4
`./S4`	relative path to S4
`S4`	relative path to S4
`../../bin`	relative path to /bin

FS Implementation

Physical Disk Layout:



- **Disk label**
  - Table of disk partitions
- **OS loader**
- **Super blok**
  - file system specification
- **List of free structures**
- **Table of i-nodes**
  - contains file attributes and disk addresses of data blocks where the file content is saved

FS Implementation

Table of i-nodes

	File Attributes	Data block addresses
0	Reserved	
1	Reserved	
2	drwxr-xr-x, root, root, 2, 512, Sep 5, ...	100,
	. . .	. . .
15		
16		
17		
	. . .	. . .

Data block 100

File name	i-node
.	2
..	2

Directory Creation

Table of i-nodes

	File attributes	Data block addresses
0	Reserved	
1	Reserved	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	. . .	. . .
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16		
17		
	. . .	. . .

Commands:

$ mkdir /DIR

Data block 100

File name	i-node
.	2
..	2
DIR	15

Data block 203

File name	i-node
.	15
..	2

Regular File Creation

Table of i-nodes

	File Attributes	Data block addresses
0	Reserved	
1	Reserved	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	. . .	. . .
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 1, 512, Sep 7,...	204,
17		
	. . .	. . .

Commands

$ mkdir /DIR

$ echo "Hello" > /DIR/f.txt

Data block 100

File name	i-node
.	2
..	2
DIR	15

Data block 203

File name	i-node
.	15
..	2
f.txt	16

Data block 204

Hello

Hard Link Creation

Table of i-nodes

	File Attributes	Data block addresses
0	Reserved	
1	Reserved	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	. . .	. . .
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 2, 512, Sep 7,...	204,
17		
	. . .	. . .

Commands:

$ mkdir /DIR

$ echo "Hello" > /DIR/f.txt

$ ln /DIR/f.txt /hl.txt

Data block 100

File name	i-node
.	2
..	2
DIR	15
tl.txt	16

Data block 203

file name	i-node
.	15
..	2
s.txt	16

Data block 204

Hello

Hard Link

- **Creation**

```
ln  original_file_name new_file_name
```

- Attributes and data of one file are **accessible though several file names**.
- It can be created only **inside one physical file system**.
- **It can not point to**
  - directory
  - non existing file

- After creation of hard link, it is not possible to distinguish between original and new file name.

- **Removing**
  - i-node and data are removed when the last name are removed.

Soft Link Creation

Table of i-nodes

	File attributes	Data block addresses
0	Reserved	
1	Reserved	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	. . .	. . .
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 2, 512, Sep 7,...	204,
17	lrwxrwx, root, root, 1, 512, Sep 7,...	205,
	. . .	. . .

Commands:

```
$ mkdir /DIR
$ echo "Hello" > /DIR/f.txt
$ ln  /DIR/f.txt  /hl.txt
$ ln  -s  /DIR/f.txt  /DIR/sl.txt
```

Data block 100

File name	i-node
.	2
..	2
DIR	15
tl.txt	16

Data block 203

File name	i-node
.	15
..	2
s.txt	16
sl.txt	17

Data block 204

Hello

Data block 205

/DIR/f.txt

Soft Link

- **Creation**

```
ln -s original_file_name new_file_name
```

- Link contains original file name in its data block or in its i-node.
- It is possible create soft link
  - between **different physical file systems**
  - to the directory
  - To nonexistent files (error during usage of soft link)

- Some operations are made directly with soft link (**`rm`**), another ones with the file on which the soft link points (**`vi`**).

Basic Commands: Directories

`pwd`	return working directory name
`cd dir`	change working directory
`ls [-ladL] dir`	list contents of directory
`mkdir [-p] dir`	make directories
`rmdir dir`	remove empty directory entries
`rm –r dir`	remove directory entries (even non empty)

Basic Commands: Files

`cp –r dir1 dir2`	dir2 doesn't exist: create copy of dir1 named by name dir2 dir2 exists: create copy of dir1 in directory dir2 (dir2/dir1)
`mv dir1 dir2`	dir2 doesn't exist: rename dir1 to dir2 dir2 exists: move dir 1 to dir2 (dir2/dir1)

- Note: be careful (recursion)

```
cp -r dir1 dir1
```

Basic Commands: Files

`cp f1 f2`	f2 doesn't exist: copy file f1 to file f2 f2 exists: overwrite file f2 by file f1
`cp f1 f2 dir`	files f1 and f2 copy to directory dir
`mv f1 f2`	move/rename file f1 to f2
`rm file`	remove file

Basic Commands: Files

`file file`	determine file type
`cat file`	concatenate and display files
`more file`	browse or page through a text file
`less file`	browse or page through a text file
`od –c file | more`	octal dump (print binary file)
`strings file`	find printable strings in an object or binary file