



File System.

Basic file/directory commands.

Department of Computer Systems FIT, Czech Technical University in Prague

©Jan Trdlička, 2014





- **Logical file system**
 - List of directories.
 - User can use commands like `cd`, `pwd`, `ls`, `cp`,... to change current working directory, get info about path to directory,...
 - Logical file system can consist of several physical file systems (use commands `mount`, `df`).
 -
- **Physical file system**
 - Subtree of directories that is saved on one physical devices (e.g. local disk, remote disk, or memory).



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 length is file system dependent.
- Code depends on implementation (ASCII, UTF8,...).
- Any characters is allowed except of character of `/`.
- **Name beginning with dot** (hidden file/directory):
 - It is not substituted by symbols `*` a `?`
 - Command `ls` doesn't list them by default (use option `-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, ACL,...).
- Time (creation, modification, access).



- **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`,...



- **Absolute path**

- It starts in the root directory `/`.
- It contains the hierarchy of directories between root directory `/` and given file

`/home/stud/smith`

- **Working directory**

- It can be display by command `pwd`.
- Its value is saved in shell variable `PWD`.
- It can be change by command `cd new_working_directory`.
- Every process can have different working 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/stud/smith`

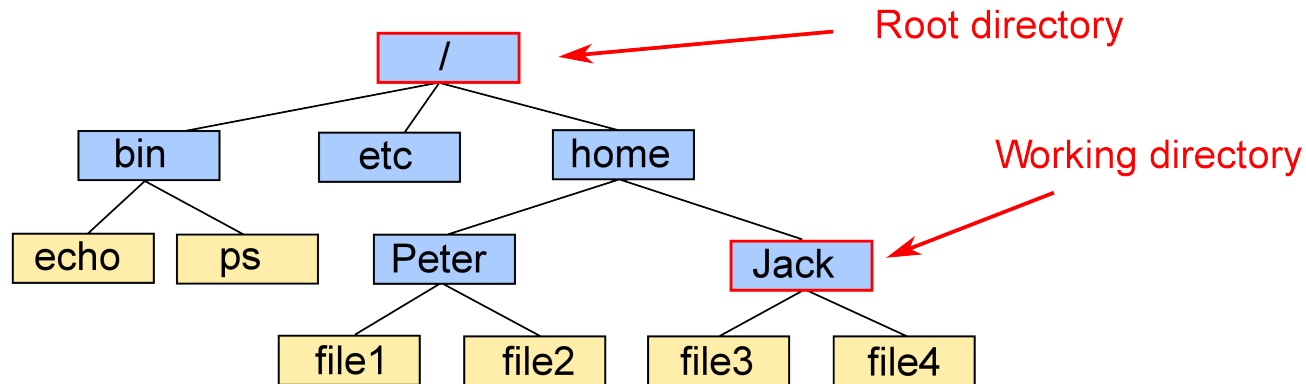
`../../../etc`

- **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`.



Directories - example



`/home/Peter/file1`

absolute path to file `file1`

`../../Peter/file1`

relative path to file `file1`

`../Peter/file1`

relative path to file `file1`

`/home/Jack/file4`

absolute path to file `file4`

`./file4`

relative path to file `file4`

`file4`

relative path to file `file4`

`/bin/ps`

absolute path to command `ps`

`../../bin/ps`

relative path to command `ps`





FS Implementation

- **Physical disk layout**

Disk label + OS loader	Super block	List of free structures (e.g. i-nodes, data blocks,...)	Table of i-nodes	Data blocks (content of files/directories)
------------------------------	-------------	--	---------------------	---

- **Disk label**
 - Table of disk partitions (every partition can contain different file system).
- **OS loader**
- **Super blok**
 - File system specific information.
- **List of free structures**
- **Table of i-nodes**
 - It contains file attributes and disk addresses of data blocks where the file content is saved.



FS Implementation

Commands

```
$ cd ; ls -lid .
```

```
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 14:58, ...	100,

Data blocks

100

File name	i-node number
.	236
..	235

/home/honza





Directory creation

Commands

```
$ cd ; ls -lid .
```

```
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza
```

```
$ mkdir dir
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 3, honza, users, 4096, Oct 8, 14:58, ...	100,
237	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 15:17, ...	152,

Data blocks

100		152	
File name	i-node number	File name	i-node number
.	236	.	237
..	235	..	236
dir	237		
/home/honza		/home/honza/dir	



Regular file creation

Commands

```
$ cd ; ls -lid .
```

```
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza
```

```
$ mkdir dir
```

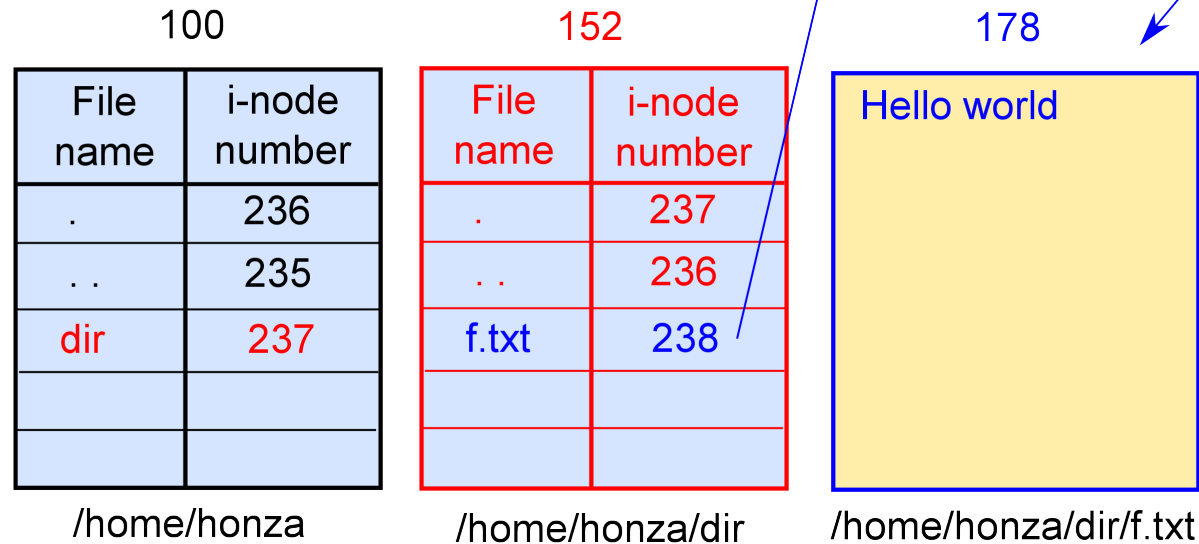
```
$ echo "Hello world" > dir/f.txt
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 3, honza, users, 4096, Oct 8, 14:58, ...	100,
237	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 15:17, ...	152,
238	-rw-r--r--, 1, honza, users, 12, Oct 8, 15:20, ...	178,

Data blocks





Hard link creation

Commands

```
$ cd ; ls -lid .
```

```
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza
```

```
$ mkdir dir
```

```
$ echo "Hello world" > dir/f.txt
```

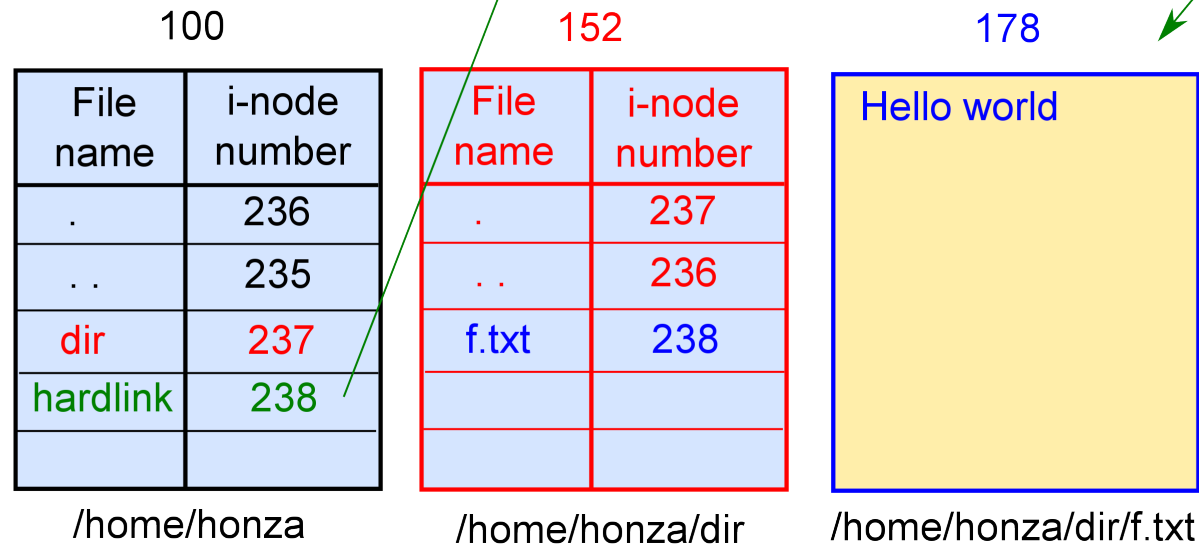
```
$ ln dir/f.txt hardlink.txt
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 3, honza, users, 4096, Oct 8, 14:58, ...	100,
237	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 15:17, ...	152,
238	-rw-r--r--, 2, honza, users, 12, Oct 8, 15:20, ...	178,

Data blocks





- **Creation**

`ln original_file_name new_file_name`

- Attributes and data of one file are **accessible through 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

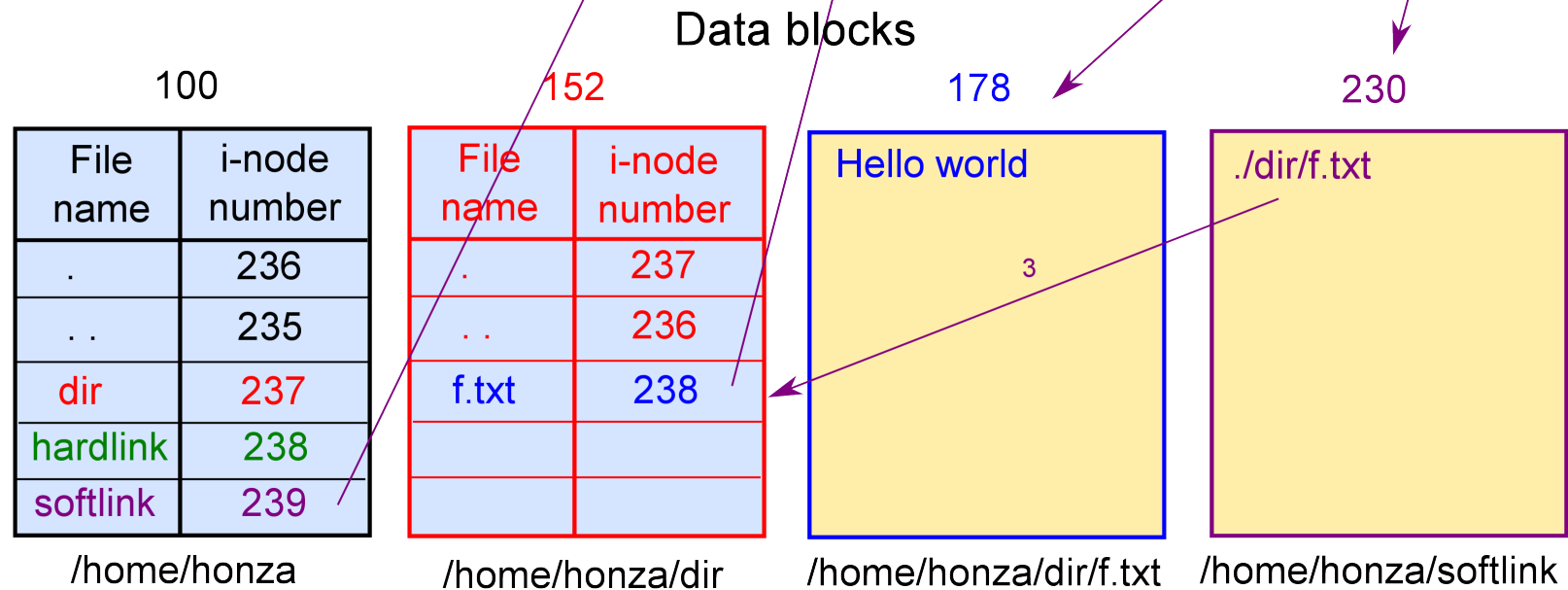
Commands

```
$ cd ; ls -lid .  
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza  
  
$ mkdir dir  
  
$ echo "Hello world" > dir/f.txt  
  
$ ln dir/f.txt hardlink.txt  
  
$ ln -s ./dir/f.txt softlink.txt
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 3, honza, users, 4096, Oct 8, 14:58, ...	100,
237	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 15:17, ...	152,
238	-rw-r--r--, 2, honza, users, 12, Oct 8, 15:20, ...	178,
239	lrwxrwxrwx, 1, honza, users, 13, Oct 8, 15:31, ...	230,





- **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 non existing files (error during usage of the soft link).
- Some operations are made directly with soft link (**rm**), another ones with the file on which the soft link points (**vi**).



Original file removing

Commands

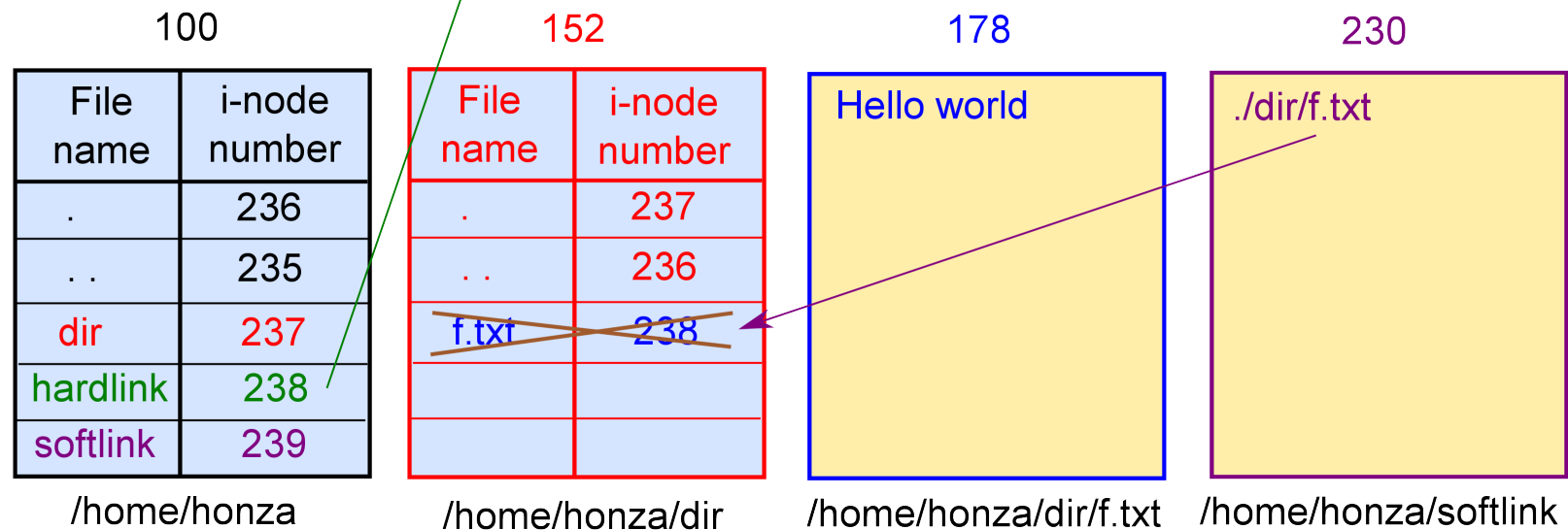
```
$ cd ; ls -lid .  
236 drwxr-xr-x 2 honza users 4096 Oct 8 15:12 /home/honza  
  
$ mkdir dir  
  
$ echo "Hello world" > dir/f.txt  
  
$ ln dir/f.txt hardlink.txt  
  
$ ln -s ./dir/f.txt softlink.txt  
  
$ rm dir/f.txt
```

Table of i-nodes

	File attributes	Data block addresses

236	drwxr-xr-x, 3, honza, users, 4096, Oct 8, 14:58, ...	100,
237	drwxr-xr-x, 2, honza, users, 4096, Oct 8, 15:17, ...	152,
238	-rw-r--r--, 1, honza, users, 12, Oct 8, 15:20, ...	178,
239	lrwxrwxrwx, 1, honza, users, 13, Oct 8, 15:31, ...	230,

Data blocks





Basic Commands: Directories

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



Basic Commands: Files

<code>cp -r dir1 dir2</code>	dir2 doesn't exist: create copy of dir1 named by name dir2 dir2 exists: create copy of dir1 in directory dir2 (dir2/dir1)
<code>mv dir1 dir2</code>	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

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



Basic Commands: Files

<code>file file</code>	determine file type
<code>cat file</code>	concatenate and display files
<code>more file</code>	browse or page through a text file
<code>less file</code>	browse or page through a text file
<code>od -c file more</code>	octal dump (print binary file)
<code>strings file</code>	find printable strings in an object or binary file