OPERATING SYSTEMS I.

PERMISSIONS

koczka.ferenc@uni-eszterhazy.hu

1. Why do we need a permission system?
2. How store operating systems their users and groups?
3. How does the permission system work in Unix-like systems?
4. How can you set rights?
5. What special rights are exist?
6. What are the attributes?
7. Why is better the permission system better in Windows?
8. What should you learn if you want to be a system administrator?

```
-rw-------  1 root root    1448 szept 24 14:07 sha
-rw-r--r--  1 root root     103 dec    4  2014 she
drwxr-xr-x  2 root root    4096 dec    4  2014 ske
-rw-r--r--  1 root root    7096 febr  28  2014 sma
-rwxr-xr-x  1 root root    5753 febr  28  2014 sma
drwxr-xr-x  3 root root    4096 dec    5  2014 sma
drwxr-xr-x  2 root root    4096 szept 14 14:38 snm
drwxr-xr-x  2 root root    4096 febr   8 22:02 ssh
drwxr-xr-x  4 root root    4096 dec   29 00:30 ssl
-rw-r--r--  1 root root     139 jan   12 07:49 sub
-rw-------  1 root root     117 szept 24 14:07 sub
-rw-r--r--  1 root root     139 jan   12 07:49 sub
-rw-------  1 root root     117 szept 24 14:07 sub
-r--r-----  1 root root     745 febr  10  2014 sud
drwxr-xr-x  2 root root    4096 febr   8 17:42 sud
drwxr-xr-x 37 root root    4096 dec   29 15:33 syn
-rw-r--r--  1 root root    2084 ápr    1  2013 sy
drwxr-xr-x  2 root root    4096 febr  16  2015 sys
drwxr-xr-x  2 root root    4096 dec    5  2014 sys
drwxr-xr-x  3 root root    4096 febr  10 23:10 sys
drwxr-xr-x  2 root root    4096 dec    4  2014 ter
-rw-r--r--  1 root root      16 nov   15 18:13 tim
-rw-r--r--  1 root root    1260 júl    1  2013 uc
drwxr-xr-x  4 root root    4096 febr  10 23:09 ude
drwxr-xr-x  3 root root    4096 dec    4  2014 ufw
-rw-r--r--  1 root root     321 jún   20  2013 up
drwxr-xr-x  3 root root    4096 nov   15 18:18 upd
drwxr-xr-x  2 root root    4096 nov   15 18:18 upd
drwxr-xr-x  2 root root    4096 dec    4  2014 upd
-rw-r--r--  1 root root     222 ápr   11  2014 up
drwxr-xr-x  2 root root    4096 dec    4  2014 vim
lrwxrwxrwx  1 root root      23 dec    4  2014 vtr
drwxr-xr-x  2 root root    4096 dec    4  2014 w3m
-rw-r--r--  1 root root    4812 febr   7  2014 wge
-rw-r--r--  1 root root    1343 jan    9  2007 wod
drwxr-xr-x  2 root root    4096 nov   15 18:18 wpa
drwxr-xr-x  4 root root    4096 dec    4  2014 X11
drwxr-xr-x 37 root root    4096 dec   29 15:33 syn
-rw-r--r--  1 root root    2084 ápr    1  2013 sy
drwxr-xr-x  2 root root    4096 febr  16  2015 sys
drwxr-xr-x  2 root root    4096 dec    5  2014 sys
drwxr-xr-x  3 root root    4096 febr  10 23:10 sys
drwxr-xr-x  2 root root    4096 dec    4  2014 ter
-rw-r--r--  1 root root      16 nov   15 18:13 tim
-rw-r--r--  1 root root    1260 júl    1  2013 uc
```

TRADITIONAL PERMISSIONS IN

UNIX

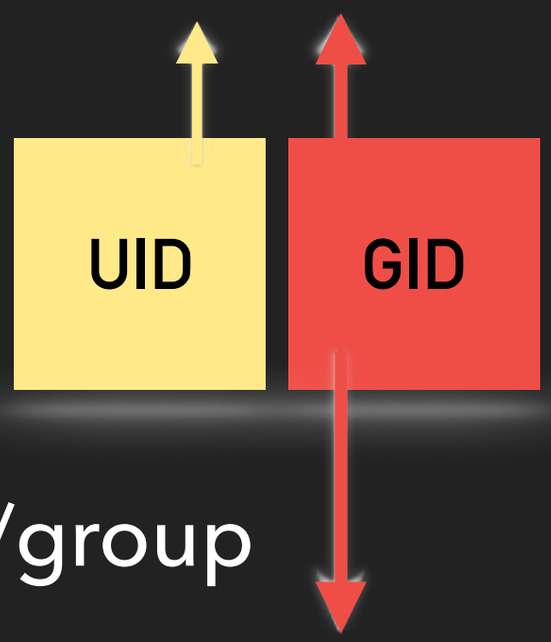
USERS AND RIGHTS IN UNIX

/etc/shadow

```
B0QZ4Y:$1$Qh8FBkSS$NPFHHeP4naTAwy64YU0Bb/:16845:0:99999:7:::
O3O6AB:$1$1e8L6tNV$NoF4hOShLuuWJFfdsDFax0:16845:0:99999:7:::
```

/etc/passwd

```
RHPY5Y:x:1918:112:Tóth Dzsenifer:/home/2016/RHPY5Y:/bin/bash
NU70VQ:x:1919:200:Tóth Nóra:/home/2016/NU70VQ:/bin/bash
CDAV40:x:1920:1921:Tóth Tamás:/home/2016/CDAV40:/bin/bash
```

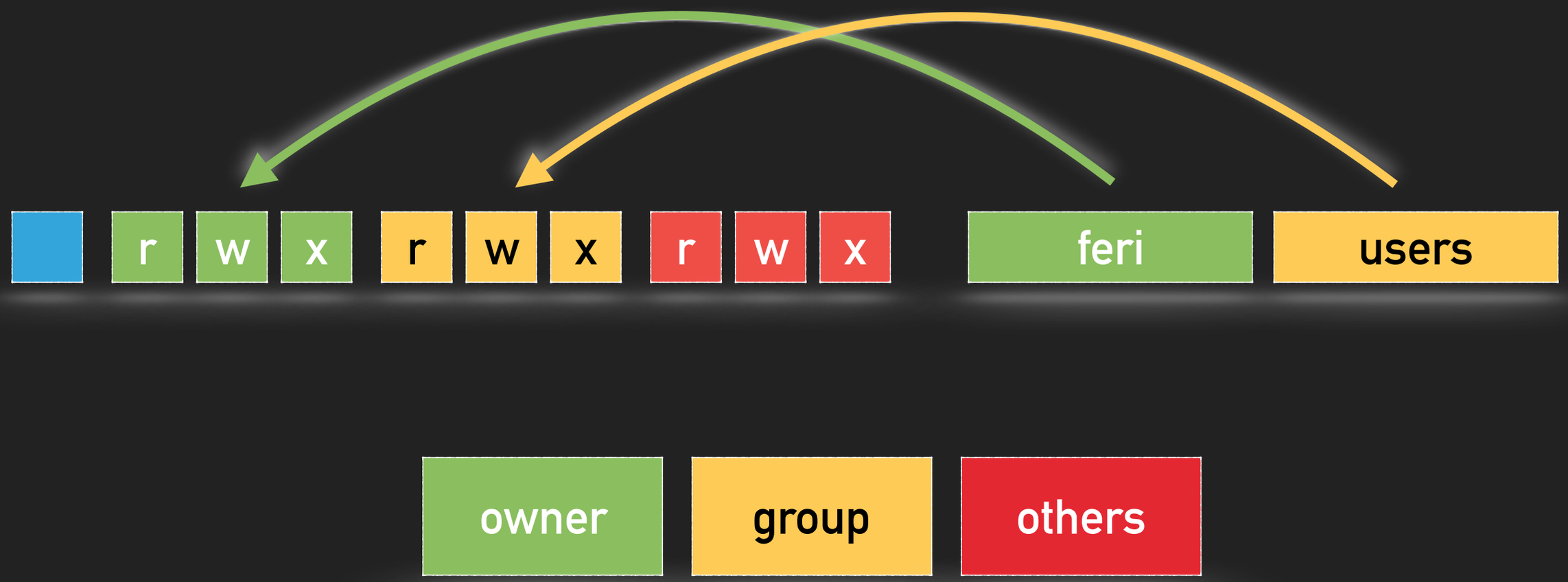


/etc/group

```
riders:x:112:
programmers:x:200:RHPY5Y,B6WSLG
CDAV40:x:1921:
```

ACCESS RIGHTS IN THE FILESYSTEM

```
-rw-r--r-- 1 feri users 1653 jan   12 07:50 test.txt
```



File rights

- **r** — The **CONTENT** of the file is readable for the owner.
- **w** — The **CONTENT** of the file is writable for the owner.
- **x** — The file is a program which is e**X**ecutable.
  The program rights are **SAME** with its owner.

Folder rights

- **r** — The **CONTENT** of the folder is readable for the owner.
  The content is the list of files and folders.
- **w** — The **CONTENT** of the folder is writable for the owner. The content is the list of files and folders.
- **x** — The user can enter into that folder.

WHEN CAN YOU DELETE A FILE OR A FOLDER?

If the folder which contains them is writable!

```
drwxr-xr-x 2 root root 4096 febr  11 08:56 /var/tmp/feri
```

```
-rwxrwxrwx 1 feri users 1021112 okt    7  2014 test.txt
```

SET RIGHTS

chmod UGO object [-R]

4	2	1	4	2	1	4	2	1
r	w	x	r	w	x	r	w	x

```
rwx:  4+2+1  =  7
r-x:  4+0+1  =  5
rw-:  4+2+0  =  6
```

```
chmod 750 apple.txt
```

chmod who operand right object [-R]

```
u+r:   user + read
g-w:   group - write
a=rw:  világ: rw-
```

```
chmod u+r apple.txt
```

CHANGE OWNER OR GROUP

```
-rw-r--r-- 1 feri users 1653 jan   12 07:50 test.txt
```

```
chown owner object [-R]
chgrp owner object [-R]
```

Only for the root!

```
chown feri /var/tmp/apple.txt
```

```
chgrp users /var/tmp/apple.txt
```

THE PROBLEM OF QUOTA

```
drwxr-xr-x 2 root root 4096 febr  11 08:56 /var/tmp/feri
```

```
-rwxrwxrwx 1 feri users 1021112 okt    7  2014 test.txt
```

SETUID

```
-r--r----- 1 root shadow 24962 febr  18 13:24 /etc/shadow
```

```
B0QZ4Y:$1$Qh8FBkSS$NPFHHeP4naTAwy64YU0Bb/:16845:0:99999:7:::
O3O6AB:$1$1e8L6tNV$NoF4hOShLuuWJFfdsDFax0:16845:0:99999:7:::
```

```
-rwsr-xr-x 1 root root 47032 jan   27 01:50 /usr/bin/passwd
```

r w s

The program runs with ITS OWNER rights.

SETGID

```
-rwxr-sr-x 1 root root     125222 febr  18 21:26 runMe
```

r w s

The program runs with its GROUPS rights.

A SETUID EXAMPLE

1

```
root@columbo:/var/tmp# ls -l
-rwxr-xr-x 1 root root 14622 febr  28 13:40 sDemo

root@columbo:/var/tmp# ./sDemo
root:$6$nn20QLBRlcw8c6fsYJrl9HSENkiT4S/
Y7omU.kc6nUt/:16623:0:99999:7:::
daemon:*:16623:0:99999:7:::
```

2

```
feri@columbo:/var/tmp$ ./sDemo
Unable to open file: /etc/shadow
```

3

```
root@columbo:/var/tmp# chmod 4755 sDemo
root@columbo:/var/tmp# ls -l
összesen 16
-rwsr-xr-x 1 root root 14622 febr  28 13:40 setUidDemo
```

4

```
feri@columbo:/var/tmp$ ./sDemo
root:$6$nn20QLBRlcw8c6fsYJrl9HSENkiT4S/
Y7omU.kc6nUt/:16623:0:99999:7:::
daemon:*:16623:0:99999:7:::
```

```
#include <iostream>
#include <fstream>
#include <string>

#define FILENAME "/etc/shadow"

using namespace std;

int main () {
  string line;
  ifstream sf (FILENAME);
  if (sf.is_open()) {
    while ( getline (sf, line) ) {
      cout << line << '\n';
    }
    sf.close();
  }
  else cout << "Unable to open file: " << FILENAME << endl;

  return 0;
}
```

STICKY BIT

C:\users*username*\AppData\Local\Temp

Writable only for the owner

/tmp

Writable for all users

```
drwxrwxrwt 55 root root 12288 febr  18 21:50 /tmp
```

r w t

If the sticky bit is set on a folder,
files in it are writable only for their owners

SETTING SPECIAL RIGHTS

chmod SUGO object [-R]

4	2	1	4	2	1	4	2	1	4	2	1
s	s	t	r	w	x	r	w	x	r	w	x

```
rwx:  4+2+1  =  7
r-x:  4+0+1  =  5
rw-:  4+2+0  =  6
```

```
chmod 4755 /tmp/bash
```

chmod who operand right object [-R]

```
g+s:   group setuid
u+s:   user setuid
o+t:   other sticky bit
```

```
chmod g+s /tmp/bash
```

INHERITING GROUP RIGHTS

Inheriting rights is possible with setting setGID and deleting Execute bit.

4	2	1	4	2	1	4	2	1	4	2	1
-	s	-	r	w	-	r	w	-	r	w	-

```
chmod 2660 mail
 drw-rwS--- 2 mail users 4096 febr  18 22:31 mail

 touch mail/tesztFile

 ls -l mail
 -rw-r----- 1 root users    0 febr  18 23:32 testFile
```

UMASK

What rights have a brand new file? It determines the umax command.

Warning!
In the parameter of umask you must specify rights which you DON'T want to enable!

umask 027

4	2	1	4	2	1	4	2	1
r	w	x	r	-	x	-	-	-

```
umask 027
drwxr-x--- 2 root root  4096 febr  18 23:10 Data
-rw-r----- 1 root root     0 febr  18 23:11 test.txt
```

ATTRIBUTES IN EXT4 FILE SYSTEM

Attributes are extra values which COULD implement special services or functions.

Example: prevent modifying a file or folder:

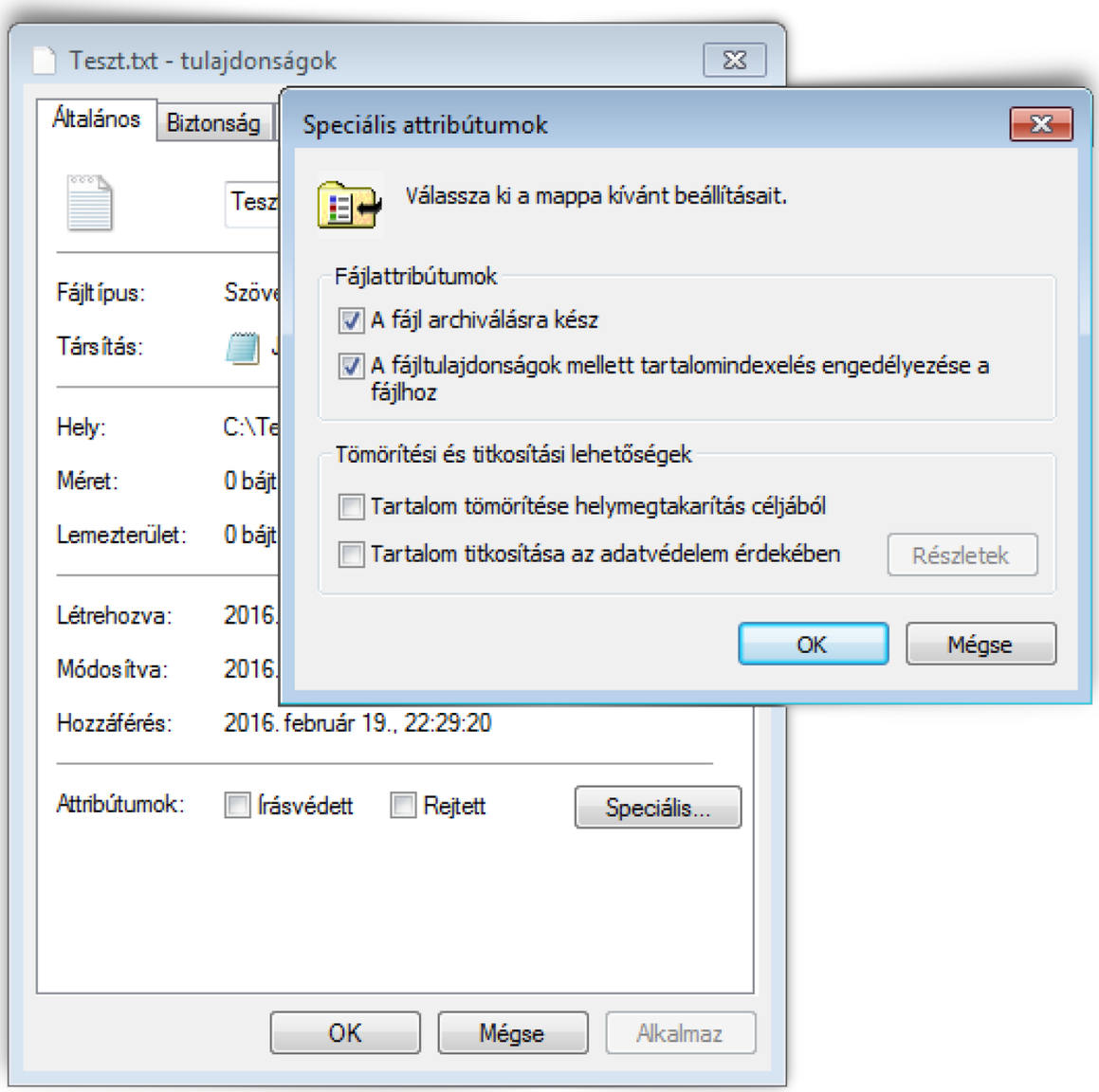
```
lsattr zh.txt
--------------e-- ./test.txt
chattr +i zh.txt

lsattr
----i---------e-- ./test.txt

rm test.txt
-bash: zh.txt: Permission denied
```

- append only (a)
- compressed (c)
- no dump (d)
- extent format (e),
- immutable (i),
- data journalling (j),
- secure deletion (s)
- no tail-merging (t),
- undeletable (u),
- no atime updates (A),
- no copy on write (C),
- synchronous directory updates (D),
- synchronous updates (S),
- top of directory hierarchy (T).

ATTRIBUTES IN WINDOWS

Content of write protected files are unchangeable.

Meaning of the **Archive** attribute.

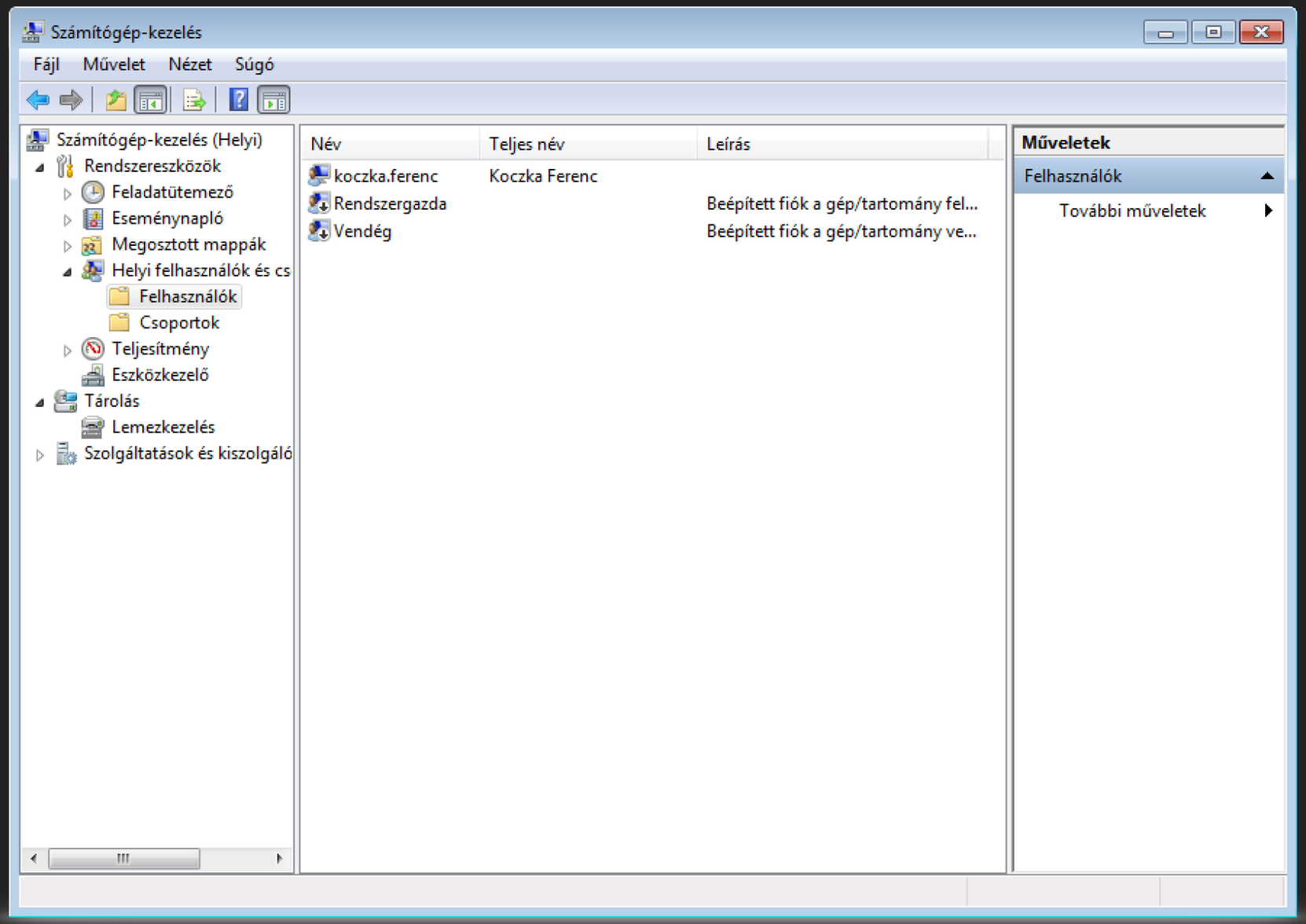


PERMISSION SYSTEM IN

WINDOWS

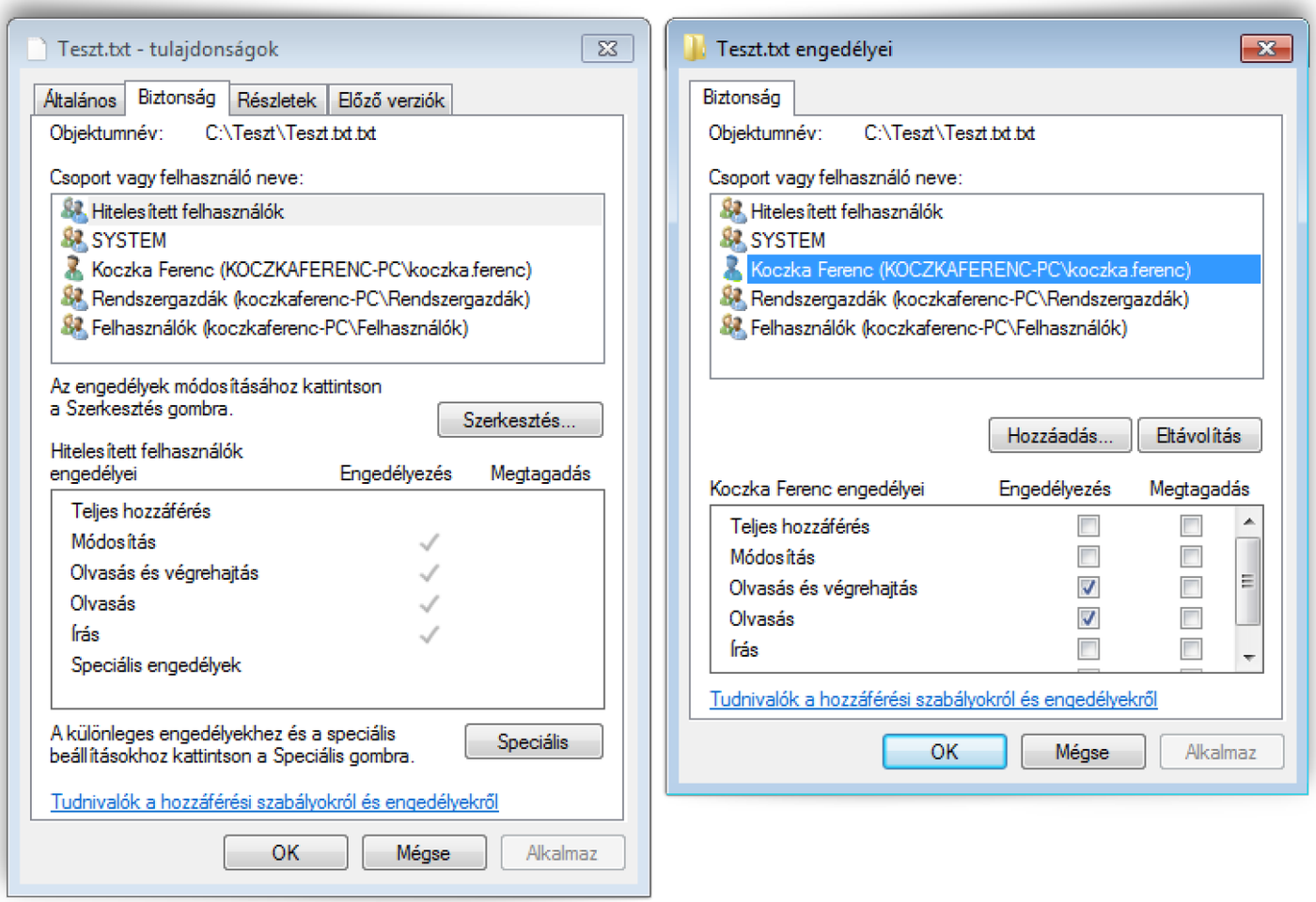
WINDOWS

User, groups and rights in Windows.



JOGOK BEÁLLÍTÁSA

Rights in NTFS are more sophisticated.

Every object could have attached more users and groups, the administrator can assign different rights for each of them.

There are options for prohibition, they are “stronger” then permissions.