ELTE INFORMATIKA

Computer Systems

Operating system tasks

Review

- Contact, grading
- The goals and the content of the subject
- Computers of yesterday, today and tomorrow
- Signals, information
- Storage of information
  - Fix-point, floating-point number representation
  - ASCII, UTF-8 etc. Code tables
- Architecture, main elements
- Operating systems

What follows next …

- Software
  - From machine code to operating system
- Operating systems and their programming possibilities
  - UNIX (Linux) possibilities
    - Usage of language tools (machine code, C++, Java,etc.)
    - Shell script
  - Windows
    - Batch, language tools (machine code, C++, Java,etc.)
    - PowerShell

What comes today?

- The tasks of operating systems
- Services offered by an operating system
- User interface
  - Character and graphical type
- File systems and their role
- File system permissions
- Base operations
  - Important operating system commands

The tasks of an operating system

- To ensure an adequate user interface
- To handle and store files
- To handle peripherial devices
- To support network services
- To implement base tasks e.g.
  - Elementary text editing
  - Handling the network
  - Etc.

Operating system services

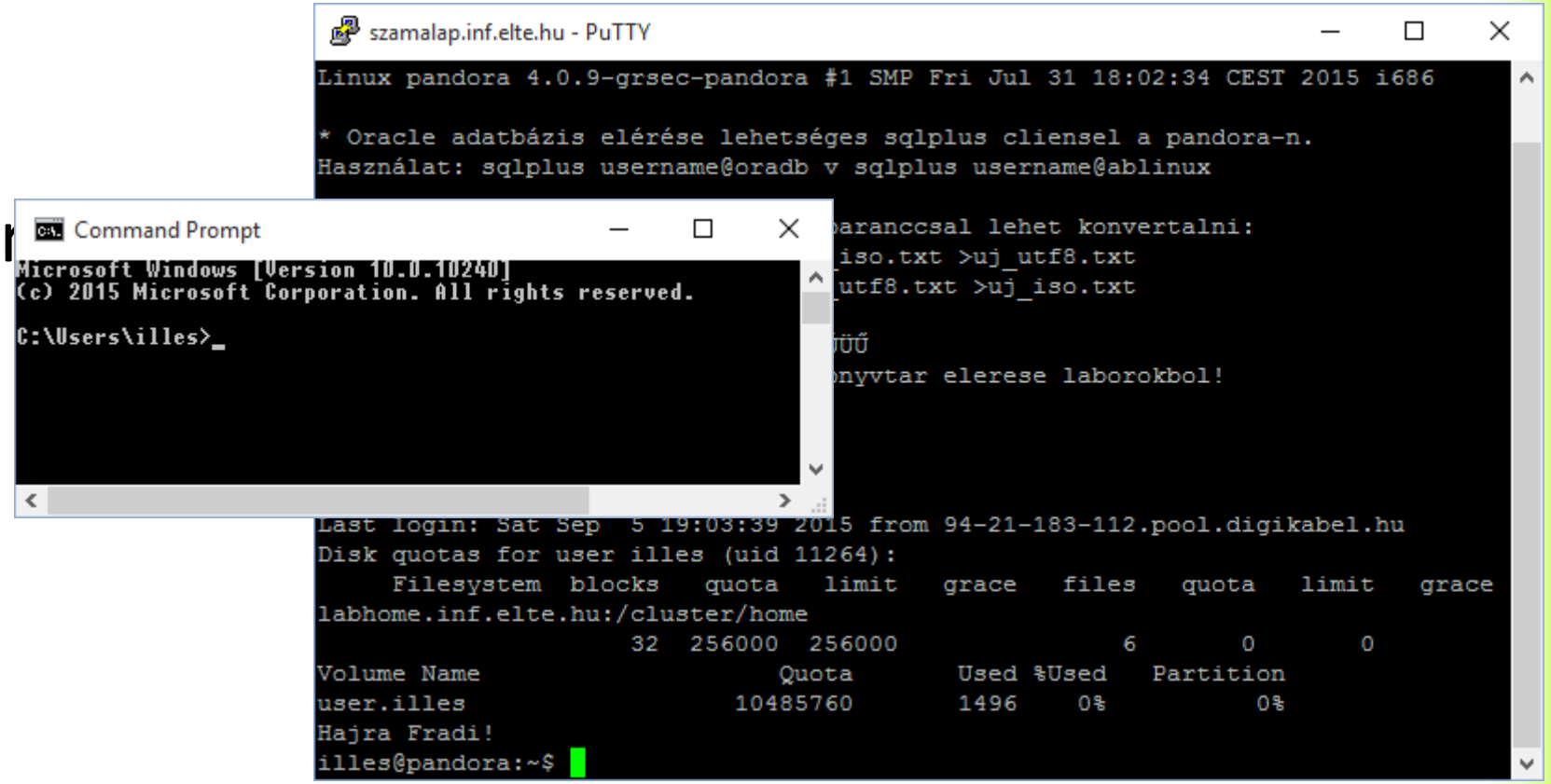
- Cliens – server differences
- The usage of common, distributed data storage devices
- The usage of the common printing service
- Handling of services
  - Mailing, web, terminal connections etc.
  - Network services (DNS, DHCP, etc.)
- Handling of users
  - Informational database

The history of Unix (Linux)

- In the 60th, AT&T Bell Lab, Dennis Ritchie, Ken Thomson
- Free for universities
- In the 80th the manufacturer (HP, IBM, Sun, SGI, DEC, etc.) re-shaped it for their own product
  - Several versions (HPUX, Solaris, Irix, etc.)
  - 2 main trends(AT&T System V, BSD)
- Freeness is ceased (except BSD)
- Standardization: POSIX
- In the 90th, LINUX (Linus Torvalds)

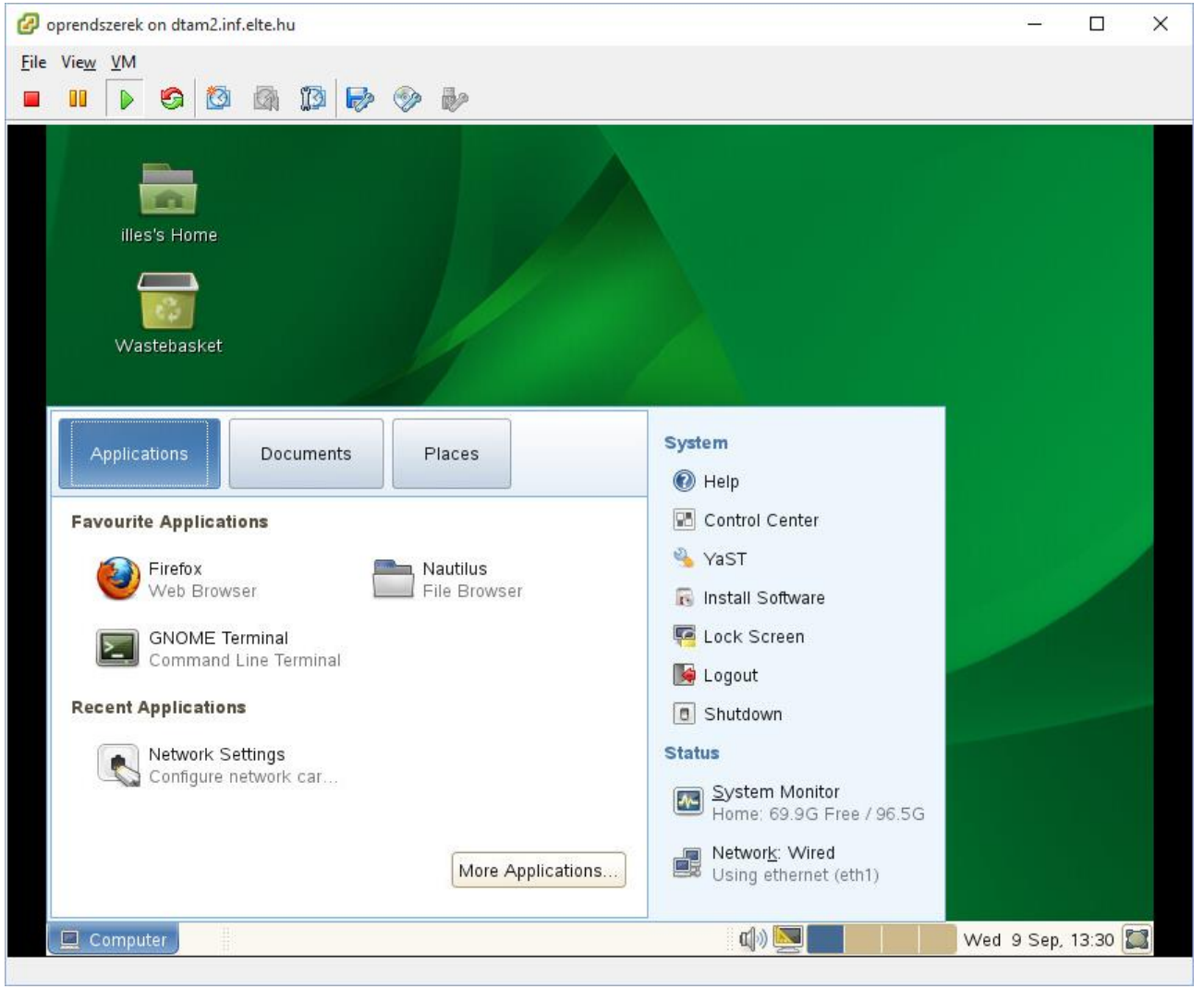
Character user interfaces

- Linux (putty)
- Windows (CMD)
- Terminal emulator
  - Character settings
  - Character sets
  - Terminal type



Graphical user interfaces

- Graphical
  - Windows 7,8,10
  - Linux
  - Mac. OS.



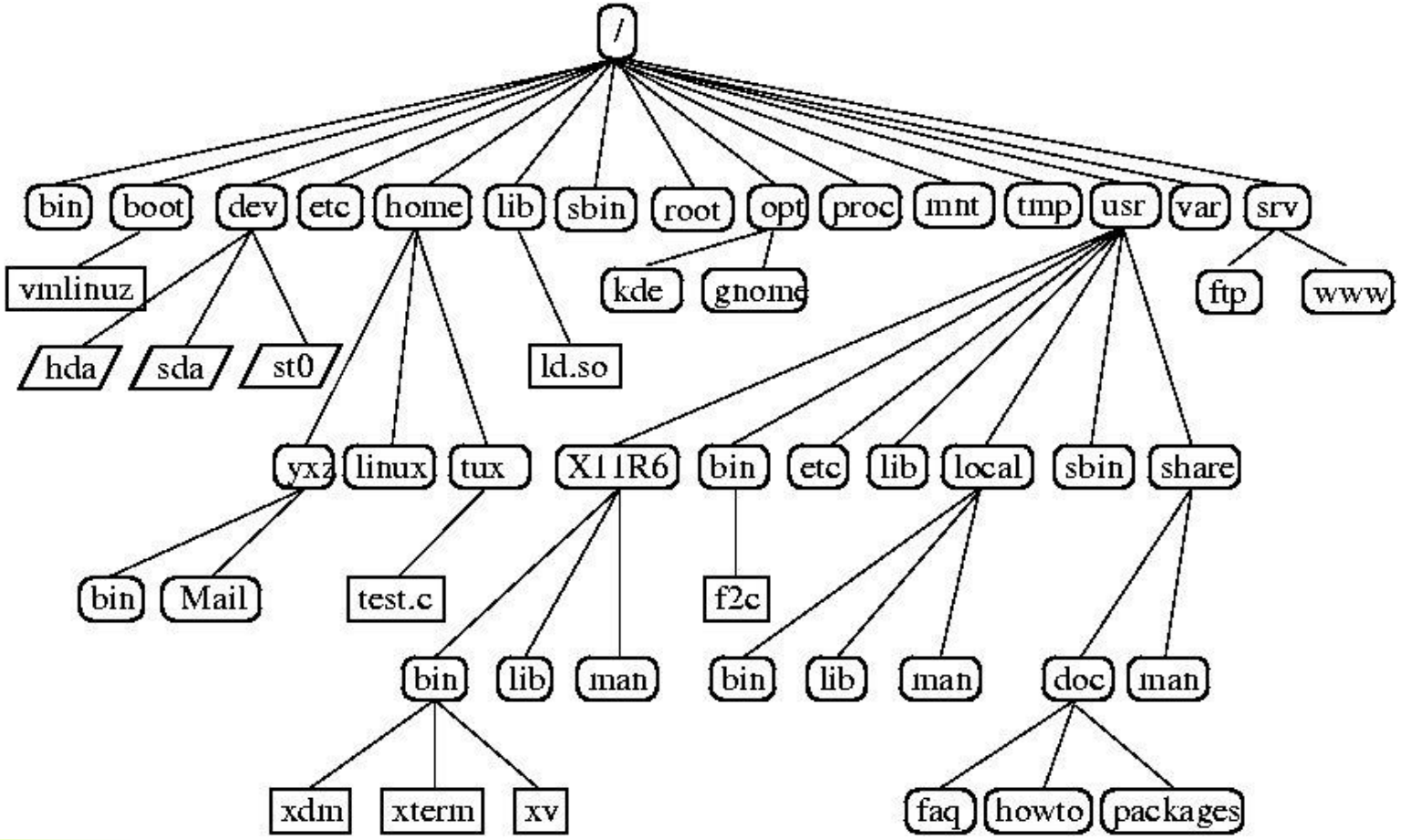
Popular operating systems

- Most general one is: Windows
  - We will not learn Windows commands, general UI usage.
  - But we will learn how to write scripts in Windows as well!
- Linux-UNIX
  - More and more popular, mostly in education, research. The most of servers are Linux based!
  - We will learn about basic commands and how to write shell script programs!
- macOS (earlier OSX)
- Mobile operating systems (iOS,Android,WP…)

File systems and their role

- Tree structure, several entry points in Windows
- Typical Unix directory elements
  - / ,there is only one root, this is /
  - /dev/… the common directory of devices
    - E.g.: /dev/fd0h1440, 1.4 floppy, /dev/null: wastebin
  - /etc/… the directory of configuration files
    - E.g.: /etc/passwd, the list of users
  - /home, /h, the users' directory
    - E.g.:/h/i/illes
  - /usr/…/usr/local/…, system (local) directories
    - E.g.: /usr/bin/cc, /bin/sh <-> /usr/bin/sh
  - /var/…, assistants for working, e.g. log files

Open Suse file system



Important shell types

- Shell: classical user interface program, derived from Unix system
- In Windowsban it is: CMD
- In Unix there are several:
  - Sh (Bourne shell)
  - Ksh (Korn shell)
  - Csh (C shell)
  - Sh (Posix shell, earlier Bourne)
  - Bash (Bourne again shell)
    - The default shell of each users!
    - Command history, line editing, filename auto-completion, alias handling

Important features of BASH I.

- Main connecting point (everything is done here)
- Command line editing, completion(tab)
  - If it is not definite, then writes out the choice.
- Usage of previously given commands (up-, down-arrow)
  - history n (writes the previous n commands)
- Usage of aliases
  - Alias name=text
  - E.g.: alias dir=”ls –l”
  - dir a*

Important features of BASH II.

- Structure of commands
  - Primary, secondary prompt: PS1,PS2
  - Syntax of a command: PS1 name parameter(s) (enter)
  - If it feels that there is no end of the command, we get the secondary prompt!
  - Two or more commands are in the same line: ;
  - Remark: #
- Login process: /etc/profile, after it execution of ~/.profile
  - The collection of local commands: .profile, etc.: PATH
    - Instead of .profile may be .bash_profile or .bash_login too!
  - Exit: .bash_logout

Features of UNIX file system

- It’s structure is hierarchical
- Mainly there are 2 different types of entry
  - Directory (symbol: d)
  - File (symbol: -, pipe: p)
- Devices also get „filenames” (/dev directory)
- Link, special file-entry
  - Hard link, only for files, inside the file system, reference number for entry changes (in Windows mklink)
  - Soft link(symbol: l), like „shortcut” in Windows!
- Today’ versions are logged, greater safety, consistency
  - Ext2, Ext3, Ext4FS

File-, directory names, conventions

- Length of name is not limited!
- Any type of character can be used!
  - But it is not adviced!
- Suggestion: do not use in names spaces, accentuated characters, special (*,%,$,etc.) characters!
- There is no file extension in the meaning of Windows!
  - .exe, .txt may be used, but it does not mean anything!
- If the starting character is . (dot), then it is a sealed entry!

Special filename references

- So, recommended characters in filenames are: letter, numbers,.,_,-
- How can we refer to several names in the same time?
- Special characters: *,?,[],!
  - ? replaces one character
  - * replaces any amount of optional characters (0 too)
    - * it does not replace the starting dot in filenames!
  - [abc] one of the listed symbols
  - [!abc] one character which is not among the listed ones
    - [A-Z] capital letters
    - [1-9] numbers between 1,9

File features

- Name
- Size
- Date of creation
- Owner
- The group of owner
- Hard link number
- Permissions

```
összesen 31
drwxr-xr-x  2 illes  10715 2048 dec   13  2013 Asztal
drwxr-xr-x  2 illes  10715 2048 dec   13  2013 Dokumentumok
-rw-r--r--  1 illes  10715   19 szept 15  2014 elso
-rw-r--r--  1 illes  10715   53 okt    2  2014 joetext1
```

Command execution, parameters

- Command: sequence of characters till enter!
- Command interpretation
  - The interpreter split it by delimiters (space)
  - First word is: the name of the command
  - The other words are: parameters
  - In the graphical interface the adequate (click) event starts a command!

Base commands I.

- ls, ls –l, ls –al #directory list
- pwd, cd, mkdir, rmdir #directory operations
- chmod, chown, chgrp, umask # permission
- passwd # password setting
- cp, mv, rm, ln # file operations
  - ln –s #soft link
- mail, telnet(ssh), ftp, nfs(mount) #arpanet
  - ssh név@host
- echo Hello world! #Writes to the display (standard output)
- man ls # ls command manual, help!

Base commands II.

- who, whoami #who is logged in
- talk username [terminal] # to start a talk
  - write user [tty]
- mesg no # to forbid talking
- clear #to clear the character display
- date #to write date and time
- finger user #to write information about a given user

Base commands III.

- find – searching
  - find . –name apple.tree
- tar (tape archive) –key [f file] files
  - Important keys:
    - c, to create archive
    - x, to eXtract from archive
    - t, to list of the content
    - v, to write out filenames on the display
  - E.g: tar –cvf apple.tar *.txt
    - tar –xvf apple.tar *

Base commands IV.

- touch filename # creates an empty files, timestamp modification
- cat, head, tail # to see the content of the file
- more, paging forward, less paging forward and backward
- read a # to read from the keyboard into the variable a till enter
  - read a b # to read into variables a and b. It reads into a till first space, the followings characters are read into b
  - line – one line of the input will be written to the output
- diff file1 file2 # to compare files
- zip, unzip, gzip, #compressing
  - zip apple.zip *.txt # compresses all txt files into apple.zip-be.
- …and a lot more….
  - Helps the MAN!

Permissions I.

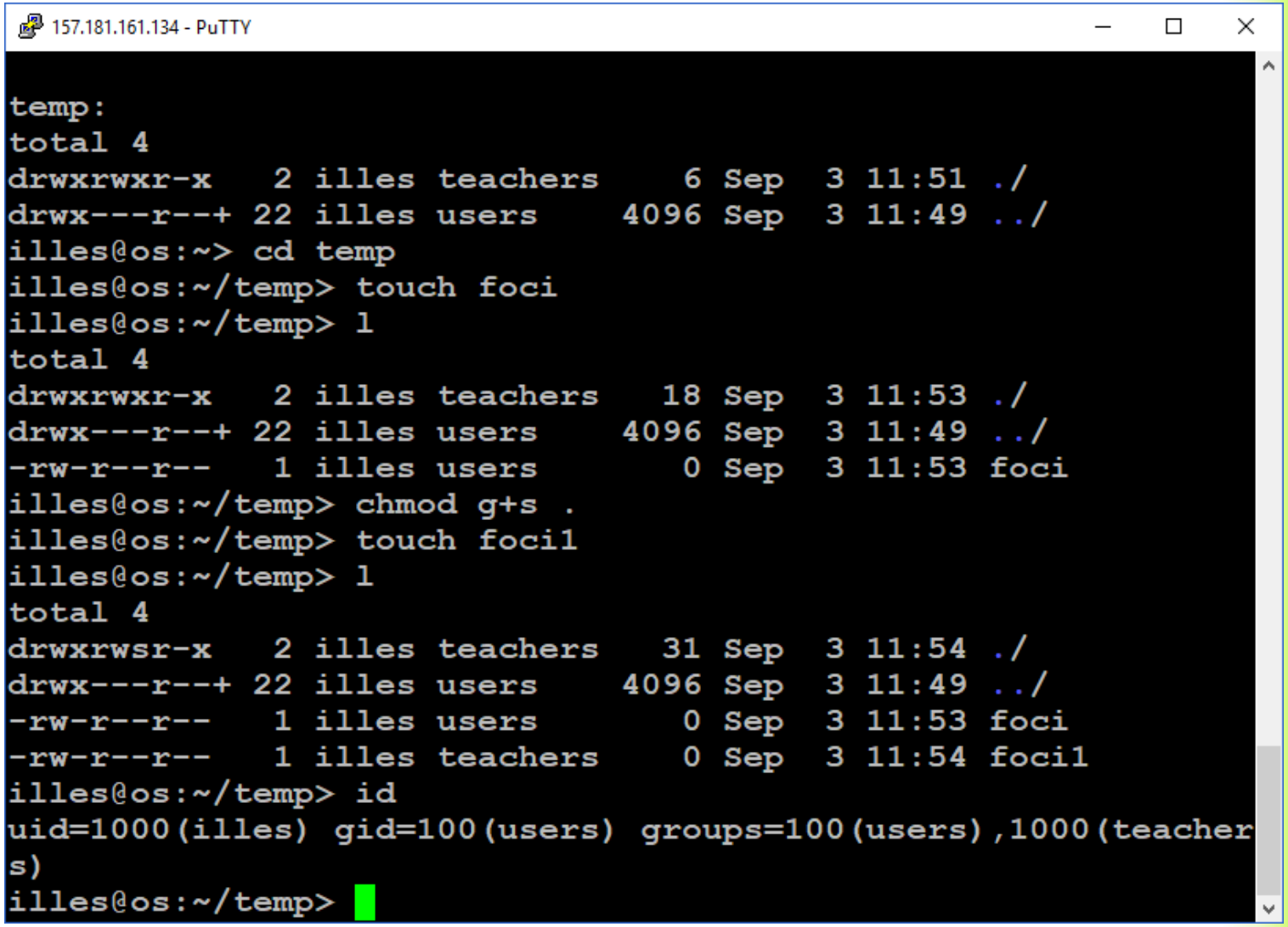
- Basically a 3x3 system exists (octal system)
  - Each of the entries has:
    - User permission (u)
    - Group permission (g)
    - Anybody else (others) permission (o)
  - Each permission consists of three parts
    - R – reading permission
    - W – writing permission
    - X – execution permission
- To set permission: chmod g+w apple.tree

Permissions II.

- Handle of r,w,x permissions as 3 bit long numbers!
  - Octal system!
- Default permission: 644
- umask, giving bits, to which we **do not** give permissions
  - E.g: umask 111 # a new file will have a permission: rw-rw-rw
  - default: umask 022
- Additional permissions: e.g.: chmod 6644 apple
  - setuid, the command is executed by the permissions of the file and not of the executor (x helyett S)
  - setgid, command is executed with the file-group permissions
  - sticky bit, file, directory only an own file can be deleted

Additional permissions– Group inheritance

- Directory GUID bit: after creating a new file in the directory, the owner group of the new file will be the directory group and not the users primary group!
- Attention: umask is live! (If write perm. is necessary, the umask must change as well)



Permissions III.

- These permission settings are quite good, but not the perfect one!
  - E.g in Windows we can add to each file unique users with different permissions!
- How can we do it in unix/linux?
- Solution: ACL (Access Control List)
  - setfacl – setting
    - setfacl –R –m u:Steve:rxw apple.tree
    - setfacl –d –m u:Steve:rwx newdir

      # after creating a file in the newdir, the new file
      # inherits the d(efault) rights
    - See more details in man!
  - getfacl - reading

Permissions - File systems

- To user access rights with chmod, setfacl works in classic Linux/Unix filesystems. (Ext2FS, Ext3FS,BTRFS, etc.)
  - df – display filesystems, to see mounted filesystems
- In distributed system this feature could modify!
  - In case of NFS (Network File System) we can use them.
  - In case of AFS (Andrew File System) not!

```
AFS                              2147483647        0 2147483647   0% /afs
illes@valerie:~$ cd ..
illes@valerie:/afs/inf.elte.hu/user/i/il$
```

Access rights in AFS

- fs command is the main adminnistrative tool in AFS (OpenAFS) filesystem.
- To list rights: fs listacl dirname (fs la)(la – listacl abbr.)
- To set rights: fs setacl dirname (fs sa)
- The full fs manual is: man fs
  - man fs_listacl or man fs_setacl, to see listacl or setacl manual
  - Internet, eg: http://docs.openafs.org/Reference/1/fs_setacl.html

ELTE INFORMATIKA