OPERATING SYSTEMS
– Laboratory 1 –

THE ATTENDANCE REQUIREMENTS

- the attendance at laboratory hours is MANDATORY
- minimum 12 attendances (minimum 10 attendances for repeating students)

1. UNIX COMMANDS

- the structure of UNIX commands:

```
command [options] [values]
```

  - `command`: the first word in the command line (lowercase letters and/or digits), it is a small program that completes a task
  - `options`:
    - short option - a single letter preceded by *a single hyphen* (-)
    - long option - a word preceded by *a double hyphen* (--)
  - `values`: may be mandatory, optional or it may not exist
  - the command line values are separated by *spaces*
  - the command line interpreter (the shell) is *case-sensitive*
- examples:
  - the command only: `pwd, ls`
  - command + option: `ls -l, ls -a (ls --all)`
  - command + value: `mkdir new_dir, touch new_file`
  - command + option + value: `ls -l /etc, cat -n hello.c`

2. USING THE UNIX COMMAND LINE MANUAL

- open manual pages to find more information about a command:
  `man nume_comandă`
- localise the manual page/section for a command: `apropos, whatis`
- the manual contains several sections (type man man - manual for command man):

```
The table below shows the section numbers of the manual followed by the types of pages they contain.

1   Executable programs or shell commands
2   System calls (functions provided by the kernel)
3   Library calls (functions within program libraries)
4   Special files (usually found in /dev)
5   File formats and conventions eg /etc/passwd
6   Games
7   Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8   System administration commands (usually only for root)
9   Kernel routines [Non standard]
```

- consult a certain section of the manual:
  `man [section_number] command  (ex. man 3 printf )`

- navigate in manual with arrows (not mouse) and these commands:
  - –previous page: `b, PgUp` (în cazuri foarte rare se suspendă execuția)
  - –next page: `SPACE, PgDn` (în cazuri foarte rare se suspendă execuția)
  - –search: `/` (slash) followed by a word we want to search for, then enter
  - –exit and close the manual: `q (lowercase – it is case sensitive!)`
- online manual: `https://linux.die.net/man/`

3. UNIX COMMANDS AND SHORTCUTS

- the format of a UNIX command:

```
command [options] [arguments values]
```

- cmds for navigating through the file system:

Command	Description	Effect
`pwd`	**p**rint **w**orking **d**irectory	Print the path cu current directory
`ls`	**l**i**s**t	List the content of the current directory
`cd` *dir*	**c**hange **d**irectories	Change the current directory to *dir*

- commands for directories:

Command	Effect
`mkdir` *nume_dir*	Create directory *nume_dir*
`cp` *dir_src dir_dest*	Copy directory *dir_src* into directory *dir_dest*
`mv` *dir_src dir_dest*	Move/rename directory *dir_src* into *dir_dest*
`rmdir` *nume_dir*	Delete directory *nume_dir*

- commands for working with files:

Command	Effect
`touch` *nume_fisier*	Create an empty file called *nume_fisier*
`cp` *fis_src fis_dest*	Create a copy of the file *fis_src* named *fis_dest*
`mv` *fis_src fis_dest*	Move/rename file *fis_src* into *fis_dest*
`rm` *nume_fisier*	Delete file *nume_fisier*
`cat` *nume_fisier*	List the content of file *nume_fisier*

- `in all cases, the argument values` *nume_dir or fis_dest or nume_fisier* ***can be an absolute path in relation to the file system (ex. /home/alina/SO/lab5/program.c) or a relative path in relation to the current directory (ex. if I am in dir SO : lab5/program.c)***

- for creating files, we can use directly a cmd line text editor like joe, for example:

```
joe filename   #and then edit/type and save and exit file with Ctrl+K+X
```

- other commands:

`help, history, clear, cut, file, grep, head, less, more, sort, tail, wc, who, whoami, users, uname`

- special keys:
  - `TAB` - autocompletion of the command line (ex. *long file name, type start of it then tab*)
  - `Arrows up (↑) and down(↓)` - navigate through commands history typed by you

- Useful key combinations:

Combination	Effect
Ctrl-C	Stop the execution of the current program
Ctrl-Z	Suspend the execution of the current program (remains on in background)
Ctrl-D	Close the working session (sometimes equivalent with EOF)
Ctrl-S	Lock the console (never saves a file 😊) you can't type anything anymore
Ctrl-Q	Unlock the console after using Ctrl+S by reflex to try to save a file
Ctrl-K	Cut/copy text from current position to end of line
Ctrl-Y	Paste copied text with *Ctrl-K (you can also use mouse select and then **right mouse click to paste** – it will paste the selected)*
Ctrl-R	Search commands history
Ctrl-A	Move cursor at the beginning of cmd line
Ctrl-B	Move cursor back one char
Ctrl-F	Move cursor forward one char
Ctrl-E	Move cursor at the end of the cmd line

- Example: run command sort without any input - stop it with Ctrl+C.

4. FILES AND DIRECTORIES PERMISSIONS

- Each file or directory has specific access rights (*permissions*)
- To list these rights: `ls -l` (the first character: d if directory, then group rights)
  - `rwx rwx rwx`
    u g o
- meaning:
  - **u** (owner) - rights for the owner of the file
  - **g** (group) - rights for the group user of the owner
  - **o** (others) - rights for other users (not in group) - anyone else
- rights symbol meaning:
  - **r** (read) - right for reading the file/dir contents
  - **w** (write) - to write/modify in the file or dir content (create/delete files in directory)
  - **x** (execute) - right to execute the file or have dir access

- number representation (use with chmod):

  r (read) = 4 **w** (write) = 2 **x** (execute) = 1

- change rights for file/dir (+ add right, - remove right, ugo – for specific group/users):

```
chmod +x file
chmod 755 file
chmod g+r file
chmod u+rw, g+r-w, g+r file
chmod u=rwx, g=rw, o=r file
```

5. INSTALL A LINUX DISTRIBUTION - HOMEWORK

MacOS Users – OK, just use https://brew.sh/

Linux users – OK (we use Ubuntu)

Windows users – several options:

5.a. Create a Virtual machine on your computer and install Linux on it:

http://www.cs.ubbcluj.ro/~rares/course/os/res/setup/virtualbox/index.html

5.b. Create and use a virtual machine in cloud (AWS Cloud):

https://aws.amazon.com/free/

5.c. Windows 10 + users can install *Ubuntu Linux App Module*, from Windows Store and enable WLS: https://docs.microsoft.com/en-us/windows/wsl/install

5.d. DANGEROUS: Install Ubuntu on a separate partition along Windows (may cause data loss – do it on your own risk)

6. CONNECTING TO A REMOTE LINUX SERVER

1.a. For Windows users:

- Download and install PuTTY (a SSH client):

  https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

- Run the application and connect to the Linux server from home:
  - server: www.scs.ubbcluj.ro (from home), linux.scs.ubbcluj.ro (from campus)
  - port: 8937 (from home), 22 (from campus)
  - protocol: SSH
  - username: [ex: abir1234]
  - password (the cursor does not move when you type, press enter at the end)

1.b. For Linux, MacOS users (including Ubuntu App in Windows) use command ssh:

```
ssh username@www.scs.ubbcluj.ro  -p 8937     #from home
ssh username@linux.scs.ubbcluj.ro        #from campus (at the lab)
```

7. PRACTICE

Solve the following requirements using these commands:

`man`	`mkdir`	`ps`	`file`	`df`
`ls -l -d -a -p`	`cd`	`jobs`	`ln`	`du`
`cat`	`rm -r -i -f`	`bg`	`find`	`diff`
`less`	`rmdir`	`fg`	`cp -r -i -f`	`pwd`
`more`	`chmod`	`kill`	`mv -i -f`	`passwd`

Create a text file in which you can write the requirements and the commands used. Learn/practice to use text editors (chose one of them): joe, vim, nano, pico, mcedit, … - **HOMEWORK**
`Useful tip: open two consoles – one for editing, one for running commands to solve the tasks below`

1. List the content of directories: `/ /bin /usr /etc /usr/include`
Where is the case, use paging display `(ls | less)`.

2. Search for text `printf` in file `/usr/include/stdio.h` (using `less`).

3. Create the following directory structure in your personal home directory:

```
(dir. personal)
|
+-- abc
|    +-- x (file)
|    +-- y (file)
|    +-- t1 (file)
|    +-- t2 (file)
|    +-- t3 (file)
|    +-- t (directory)
|        +-- a (file)
|        +-- b (file)
|
+-- zz (directory)
|    +-- x (file)
|
+-- tt (directory)
```

4. Copy (recursively) all the content of directory `abc` in directory `zz` (`abc` will become subdirectory of `zz`).

5. Copy the content of `abc` in dir `zz` without overwriting files with the same name (`x`, in our case).

6. Copy files `t1` and `t2` from directory `abc` in directory `tt` (using generic specifier).

7. Create a new directory with `x rights`, but not `r`.
Create a file in this directory. What happens ?
Give the directory again right for `r` and remove `x`. What happens?

8. Give the necessary access right so that:
- anyone can see the contents of dirs `abc` and `abc/t`
- anyone can add files in dir `abc/t`
- anyone can read files `x`, `y`, `t1`, `t2`, `t3` from `abc`, but can not read file `a` and `b` from directory `abc/t`.

9. Use long listing for files `t`, `t1`, `t2`, `t3` in dir `abc` (to see acess right of `t`, not of the files contained).

10. Command `cp /dev/zero /dev/null` is an infinite cycle.
We need to move it in background (`Crtl+Z`), and we can see with `ps` the processes in executiong, then stop it with ^C or `kill -9 PID`).

ATTENTION This command uses system resources and might cause slow down – don't let it run too long, make sure you stop it before closing the terminal session, otherwise your account risks suspension for using server resources!!!

11. Create in directory `tt` a symbolic link named `c` to directory `abc`.
Explore this functionality using `cd` and `pwd`.

RESOURCES:

- Filesystem navigation: `http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html`
- Directory/file handling: `http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html`