

# OPERATING SYSTEMS

## – Laboratory 1 –

### 1. 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:

- server: `linux.scs.ubbcluj.ro`
- port: `8937`
- protocol: `SSH`
- username: `[ex: abir1234]`

#### 1.b. For Linux or MacOS users:

```
ssh username@linux.scs.ubbcluj.ro
```

```
ssh username@172.30.0.9
```

### 2. PRACTICE

Cerințele de mai jos vor fi rezolvate folosind următoarele comenzi Linux:

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

Creați un fișier text în care veți scrie comenzile folosite pentru a rezolva cerințele de mai jos, precum și răspunsurile la întrebări.

1. Listați conținutul următoarelor directoare: `/` `/bin` `/usr` `/etc` `/usr/include`  
Acolo unde este cazul, paginați listarea (`ls` | `less`).

2. Căutați textul `printf` în fișierul `/usr/include/stdio.h` (folosind `less`).

3. Creați următoarea structură de directoare și fișiere în directorul vostru personal:

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

4. Copiați întreg conținutul (recursiv) directorului `abc` în directorul `zz` (`abc` va deveni subdirector în directorul `zz`).

5. Copiați conținutul lui `abc` în directorul `zz` fără a suprascrie fișierele cu același nume (`x`, în cazul nostru).

6. Copiați fișierele `t1` și `t2` din directorul `abc` în directorul `tt` (folosind specificator generic).

7. Creați un nou director cu dreptul de acces `x`, dar nu și `r`.

Creați un fișier în acest director. Ce observați ?

Dați apoi acestui director dreptul de acces `r` și eliminați `x`. Ce observați ?

8. Dați drepturile de acces potrivite astfel încât:

- oricine să poată vizualiza conținutul directoarelor `abc` și `abc/t`
- oricine să poată adăuga fișiere în directorul `abc/t`
- oricine să poată citi fișierele `x`, `y`, `t1`, `t2`, `t3` din `abc`, dar să nu poată citi fișierele `a` și `b` din directorul `abc/t`.

Listați în format lung fișierele `t`, `t1`, `t2`, `t3` din directorul `abc` (astfel încât să fie vizibile drepturile de acces ale lui `t`, și nu cele ale fișierelor pe care acesta le conține).

9. Comanda `cp /dev/zero /dev/null` este un fel de "ciclu infinit" (nu se termină).

Lansați comanda de mai sus, mutați-o în background, listați procesele active, terminați comanda (în ambele variante: comanda trebuie mutată în foreground și oprită cu `^C` sau cu `kill`).

### **ATENȚIE**

*Comanda de mai sus consumă mult din resursele sistemului.*

*Nu o lăsați să ruleze mai mult decât aveți nevoie.*

*Terminați această comandă înainte de închiderea sesiunii de lucru !*

10. Creați în directorul `tt` o legătură simbolică cu numele `c` către directorul `abc`.

Explorați funcționarea acestei legături simbolice vis-a-vis de comenzile `cd` și `pwd`.