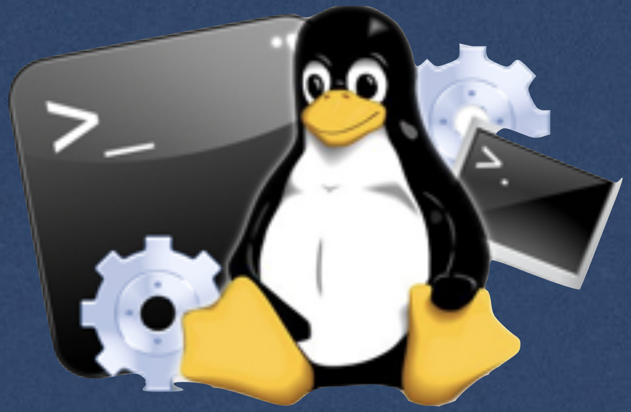
Lesson #3



First Exercises

Roberto De Virgilio

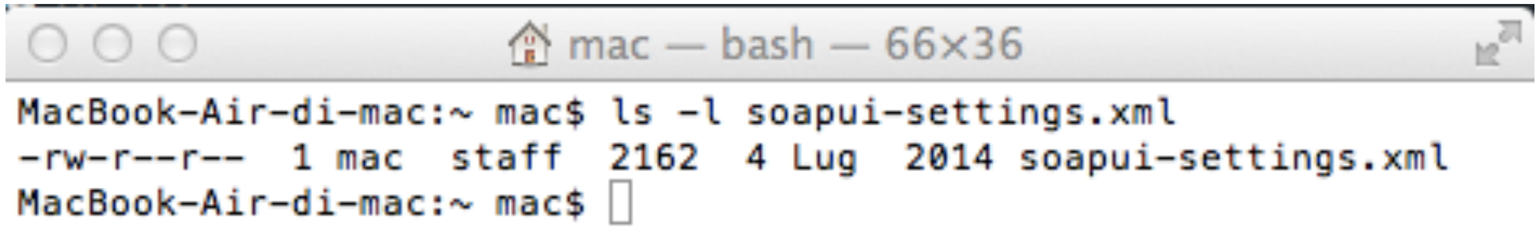
Sistemi operativi - 26 October 2015

Linux Permissions

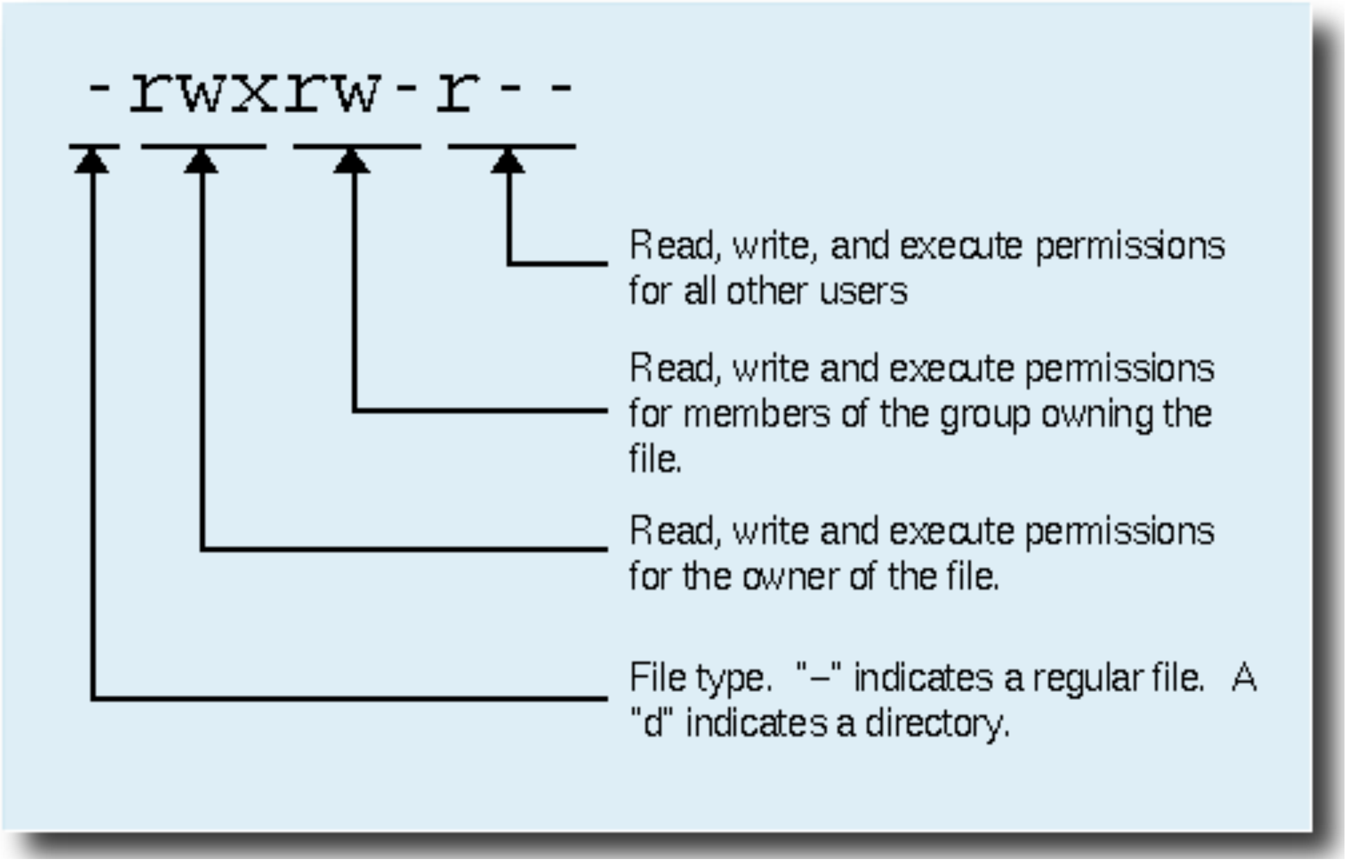
- *Linux permissions dictate 3 things you may do with a file,* ***read****,* ***write*** *and* ***execute****. They are referred to in Linux by a single letter each.*
  - ***r*** *read - you may view the contents of the file.*
  - ***w*** *write - you may change the contents of the file.*
  - ***x*** *execute - you may execute or run the file if it is a program or script.*
- *For every file we define 3 sets of people for whom we may specify permissions.*
  - ***owner*** *- a single person who owns the file. (typically the person who created the file but ownership may be granted to some one else by certain users)*
  - ***group*** *- every file belongs to a single group.*
  - ***others*** *- everyone else who is not in the group or the owner.*
- *Three permissions and three groups of people. That's about all there is to permissions really. Now let's see how we can view and change them.*

View Linux Permissions

- *To view Linux Permissions, digit* ***ls -l [file]***



```
MacBook-Air-di-mac:~ mac$ ls -l soapui-settings.xml
-rw-r--r--  1 mac  staff  2162  4 Lug  2014 soapui-settings.xml
MacBook-Air-di-mac:~ mac$ 
```



-rwxrw-r--

Read, write, and execute permissions for all other users

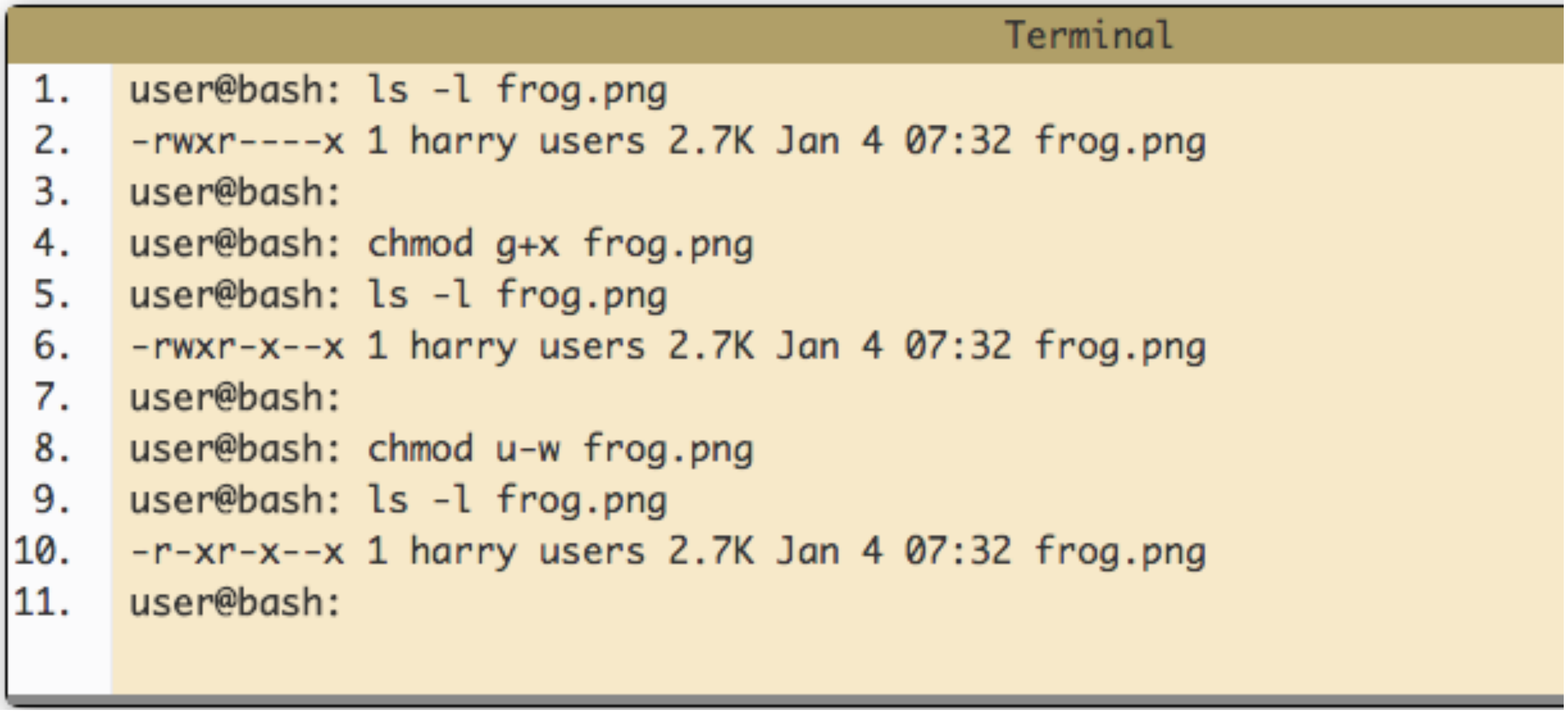
Read, write and execute permissions for members of the group owning the file.

Read, write and execute permissions for the owner of the file.

File type. "–" indicates a regular file. A "d" indicates a directory.

Change Linux Permissions

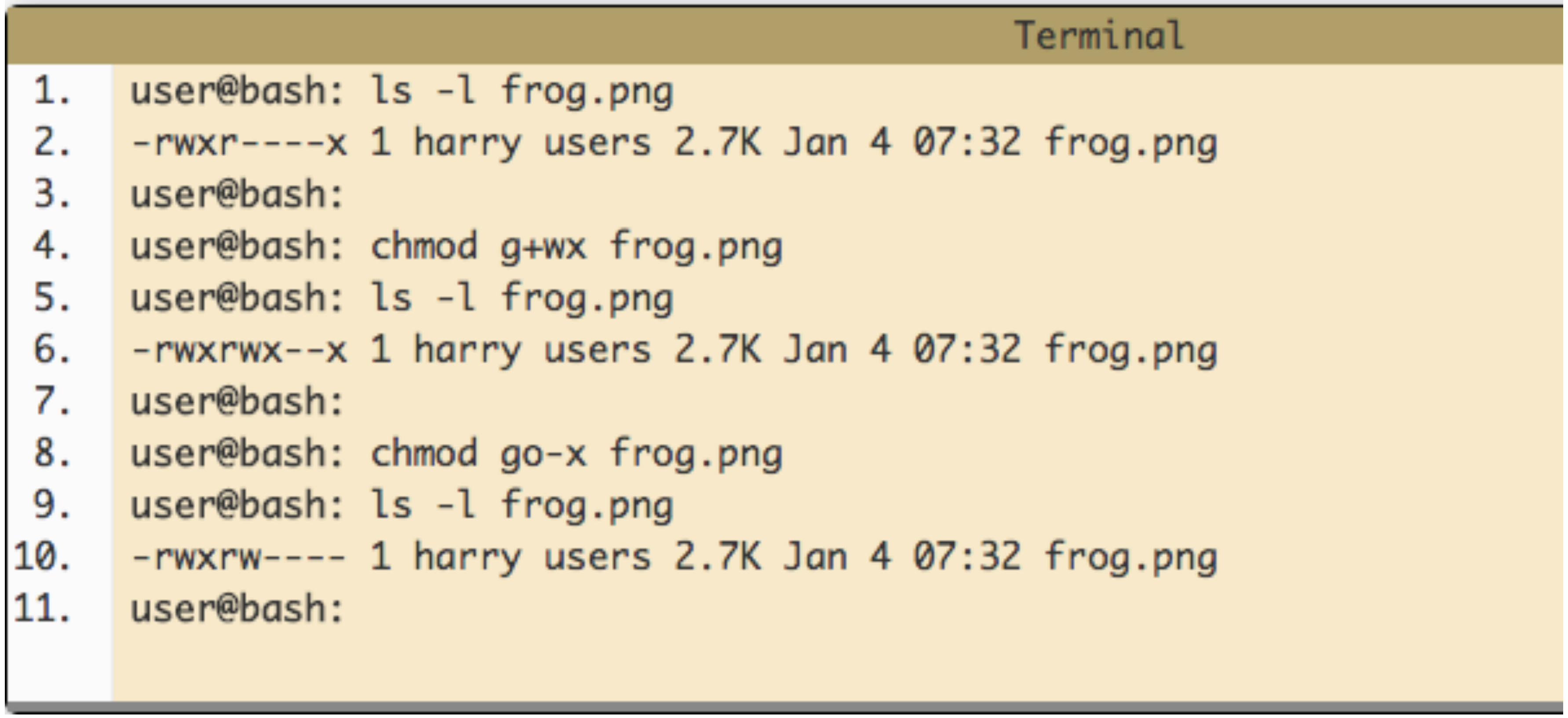
- *To change Linux Permissions, digit* ***chmod [permissions] [path]***
- *Grant the execute permission to the group. Then remove the write permission for the owner.*



```
user@bash: ls -l frog.png
-rwxr----x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod g+x frog.png
user@bash: ls -l frog.png
-rwxr-x--x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod u-w frog.png
user@bash: ls -l frog.png
-r-xr-x--x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
```

Change Linux Permissions

- *To change Linux Permissions, digit* ***chmod [permissions] [path]***
- *Don't want to assign permissions individually? We can assign multiple permissions at once.*



```
user@bash: ls -l frog.png
-rwxr----x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod g+wx frog.png
user@bash: ls -l frog.png
-rwxrwx--x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod go-x frog.png
user@bash: ls -l frog.png
-rwxrw---- 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
```

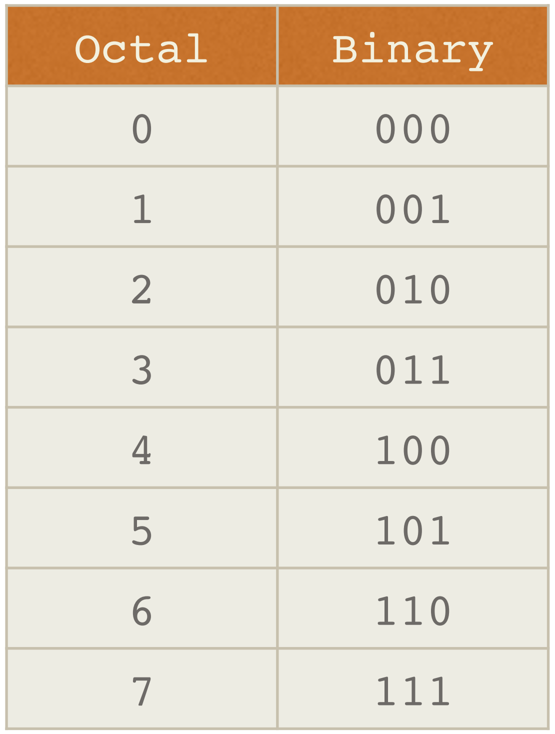
Change Linux Permissions

- *The method outlined above is not too hard for setting permissions but it can be a little tedious if we have a specific set of permissions we should like to apply regularly to certain files. Luckily, there is a shorthand way to specify permissions that makes this easy.*

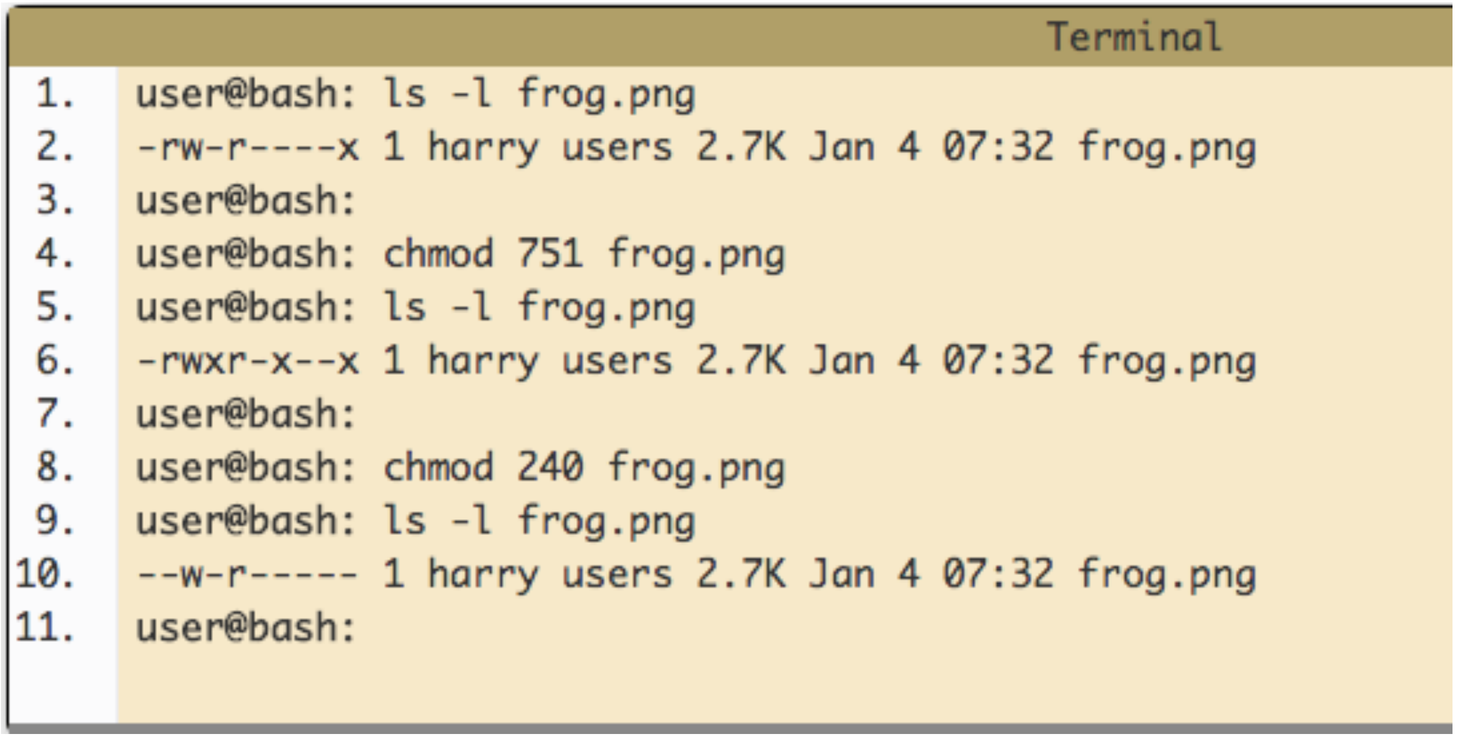
Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Change Linux Permissions

- *The method outlined above is not too hard for setting permissions but it can be a little tedious if we have a specific set of permissions we should like to apply regularly to certain files. Luckily, there is a shorthand way to specify permissions that makes this easy.*



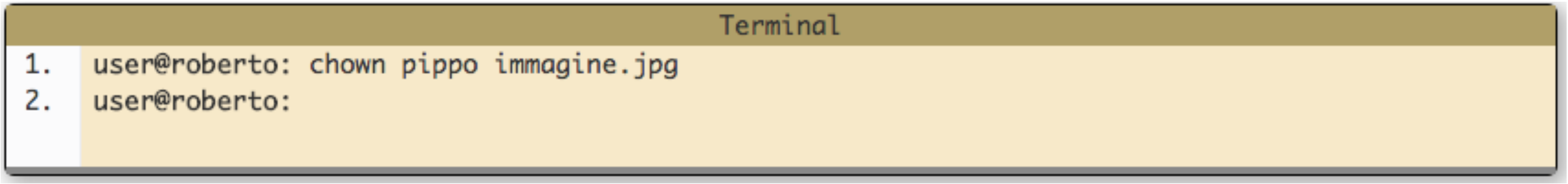
Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



```
user@bash: ls -l frog.png
-rw-r----x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod 751 frog.png
user@bash: ls -l frog.png
-rwxr-x--x 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
user@bash: chmod 240 frog.png
user@bash: ls -l frog.png
--w-r----- 1 harry users 2.7K Jan 4 07:32 frog.png
user@bash:
```

Change Linux Permissions

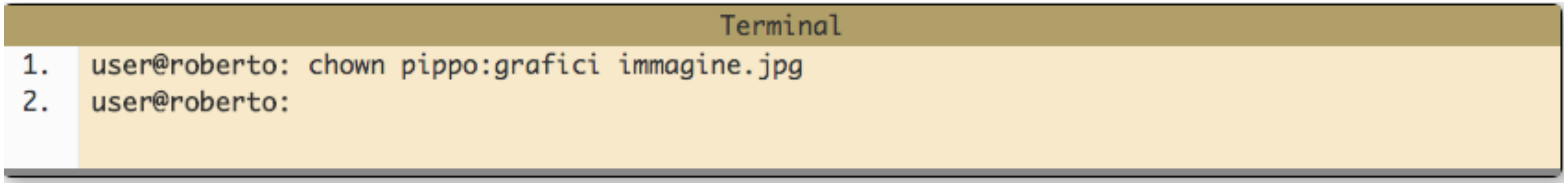
- *To change Linux Owner of a file, digit* ***chown [new owner] [file]***
- *However, to use the chown command you have to be a superuser.*



```
user@roberto: chown pippo immagine.jpg
user@roberto:
```

Change Linux Permissions

- *It is possible to change both Linux Owner and Group of a file;*
- *digit* ***chown [newowner:newgroup] [file]***
- *However, to use the chown command you have to be a superuser..*



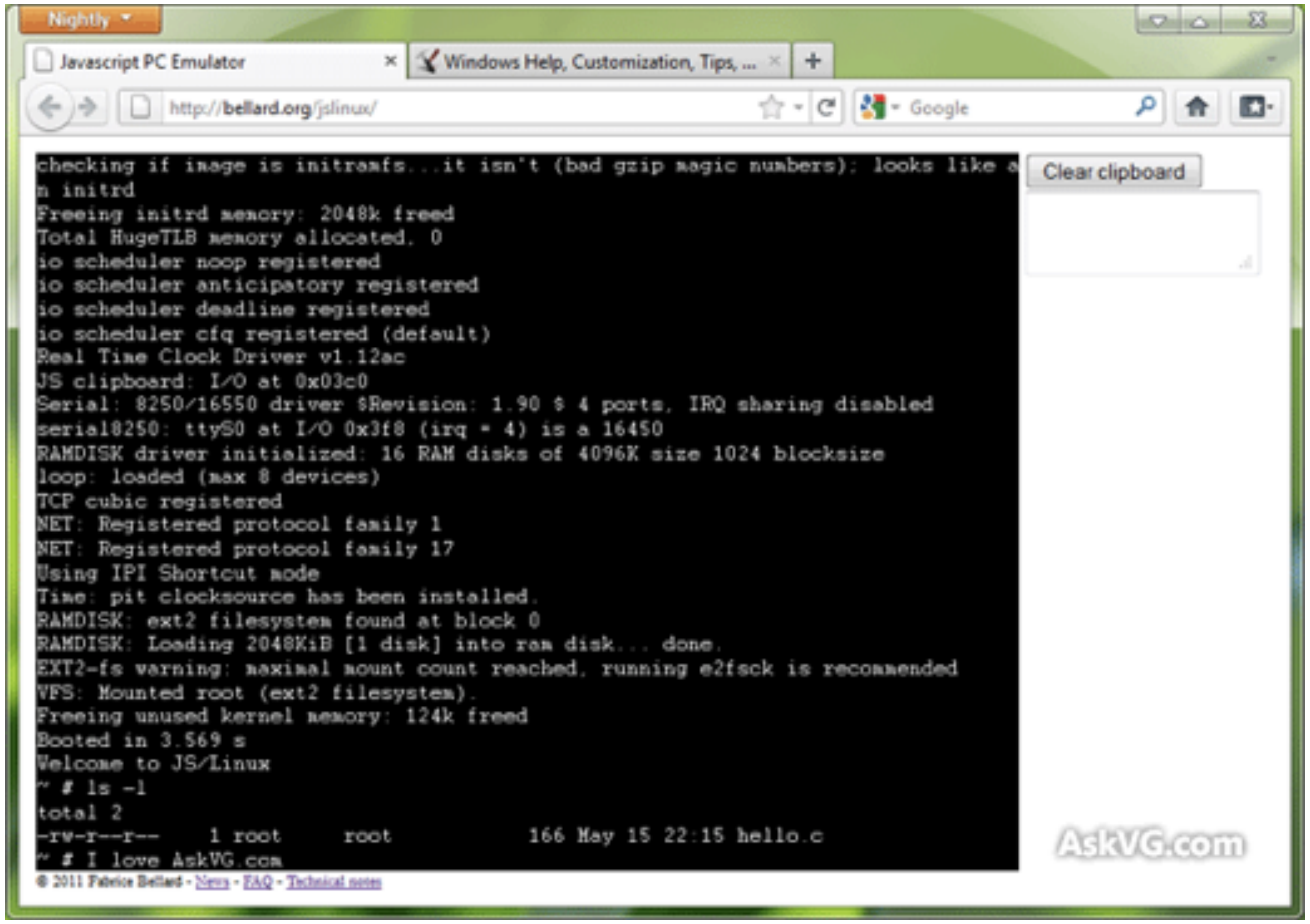
```
user@roberto: chown pippo:grafici immagine.jpg
user@roberto:
```

Change Linux Permissions

- *It is possible to change both Linux Owner and Group of a file;*
- *digit **chgrp [newgroup] [file]***
- *However, to use the chown command you have to be a superuser..*

Exercise

- *You should use your personal terminal to execute linux commands*
- *However if you would use an emulator (only for the first exercises) to avoid any damages please visit http://bellard.org/jslinux/*



Nightly
Javascript PC Emulator
Windows Help, Customization, Tips, ...
http://bellard.org/jslinux/
Google
Clear clipboard
AskVG.com

```
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like a
n initrd
Freeing initrd memory: 2048k freed
Total HugeTLB memory allocated, 0
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Real Time Clock Driver v1.12ac
JS clipboard: I/O at 0x03c0
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16450
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI Shortcut mode
Time: pit clocksource has been installed.
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 2048KiB [1 disk] into ram disk... done.
EXT2-fs warning: maximal mount count reached, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 124k freed
Booted in 3.569 s
Welcome to JS/Linux
~ # ls -l
total 2
-rw-r--r--    1 root     root           166 May 15 22:15 hello.c
~ # I love AskVG.com
```

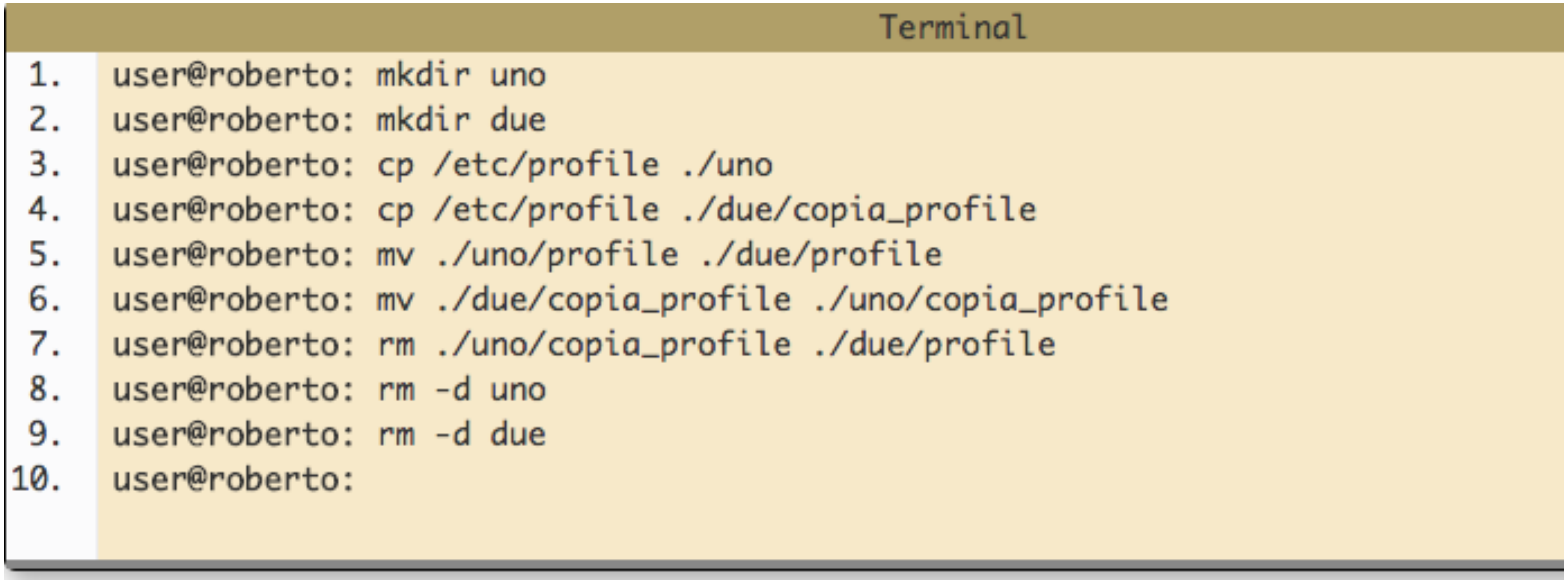
© 2011 Fabrice Bellard - News - FAQ - Technical notes

Warm-Up Exercise

- *Execute a set of linux commands as follows*

1. *create two directories with names “uno” and “due”.*
2. *copy the file /etc/profile into uno.*
3. *copy the file /etc/profile into due but renaming it as copia-profile.*
4. *move profile from uno to due and move copia-profile from due to uno*
5. *delete profile and copia-profile with a single command*
6. *delete the directories uno and due.*
7. *test the command touch to modify access date of a file or to create new file.*

Warm-Up Exercise



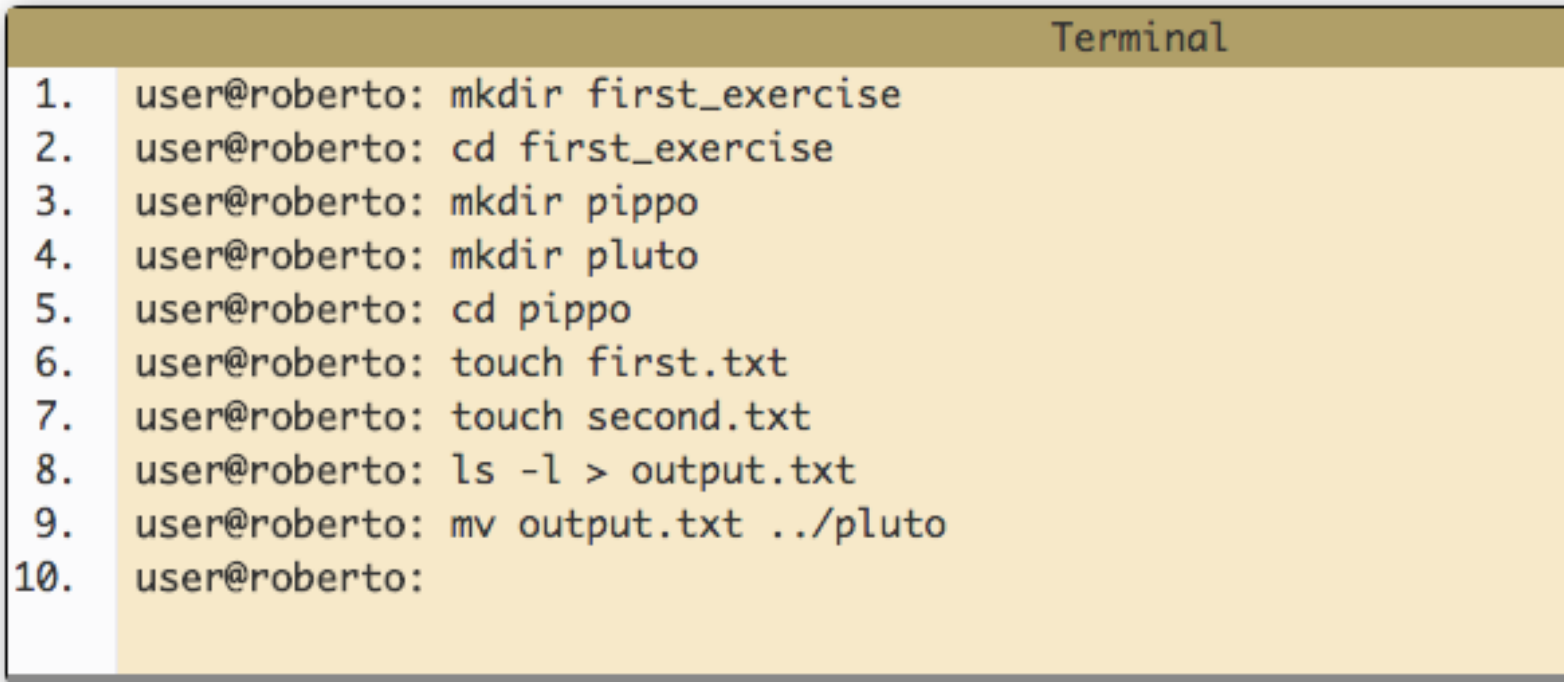
```
user@roberto: mkdir uno
user@roberto: mkdir due
user@roberto: cp /etc/profile ./uno
user@roberto: cp /etc/profile ./due/copia_profile
user@roberto: mv ./uno/profile ./due/profile
user@roberto: mv ./due/copia_profile ./uno/copia_profile
user@roberto: rm ./uno/copia_profile ./due/profile
user@roberto: rm -d uno
user@roberto: rm -d due
user@roberto:
```

Exercise 1

- *Execute a set of linux commands as follows*

1. *create the directory* ***first_exercise***
2. *create the directories* ***pippo*** *and* ***pluto*** *into first_exercise*
3. *enter into the directory pippo*
4. *create two files* ***first.txt*** *and* ***second.txt***
5. *list all files (and permissions) of the directory pippo and redirect the output into the file* ***output.txt***
6. *move the file output.txt in the directory pluto*

Exercise 1



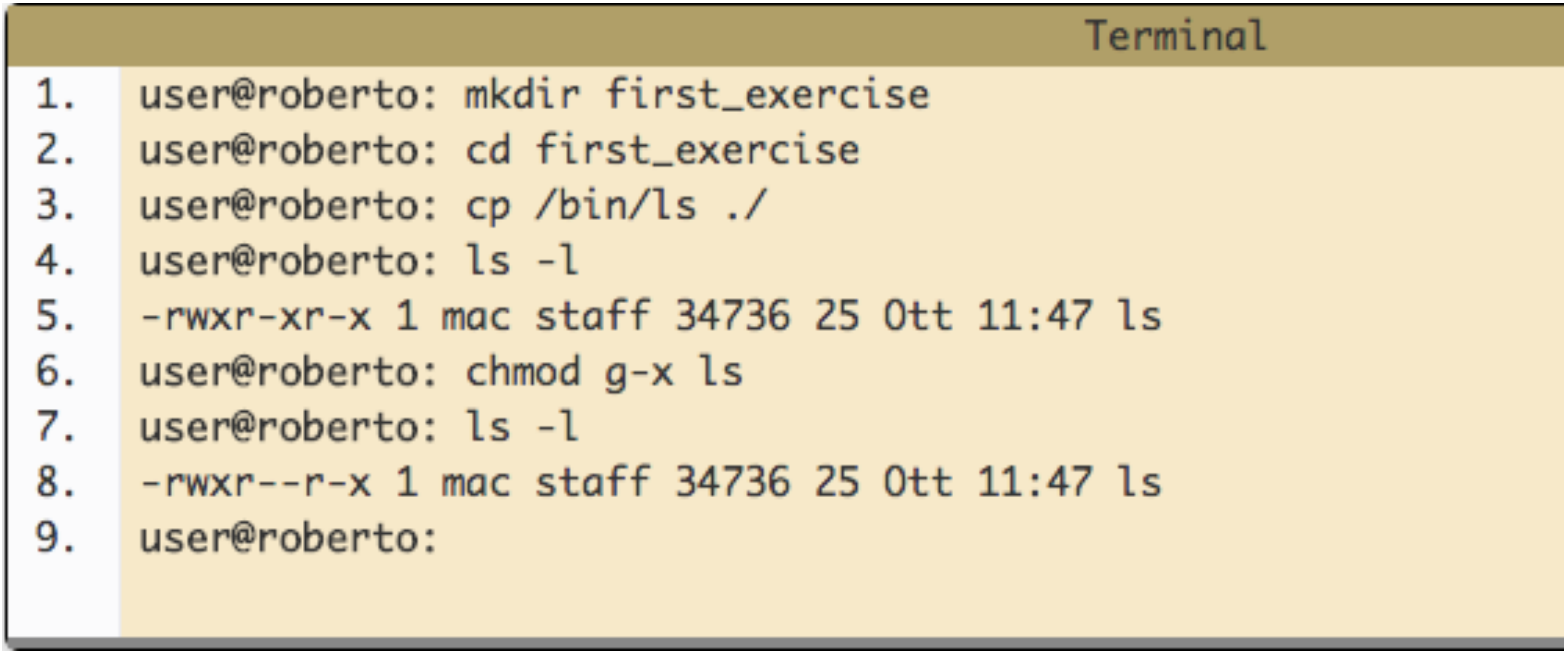
```
user@roberto: mkdir first_exercise
user@roberto: cd first_exercise
user@roberto: mkdir pippo
user@roberto: mkdir pluto
user@roberto: cd pippo
user@roberto: touch first.txt
user@roberto: touch second.txt
user@roberto: ls -l > output.txt
user@roberto: mv output.txt ../pluto
user@roberto:
```

Exercise 2

- *Execute a set of linux commands as follows*

1. *create the directory **first_exercise***
2. *copy the file **/bin/ls** in the directory first_exercise*
3. *view the permission of the copied file*
4. *delete the execution permission on the copied file for the group*
5. *view the new permission of the copied file*
6. *delete the file*

Exercise 2



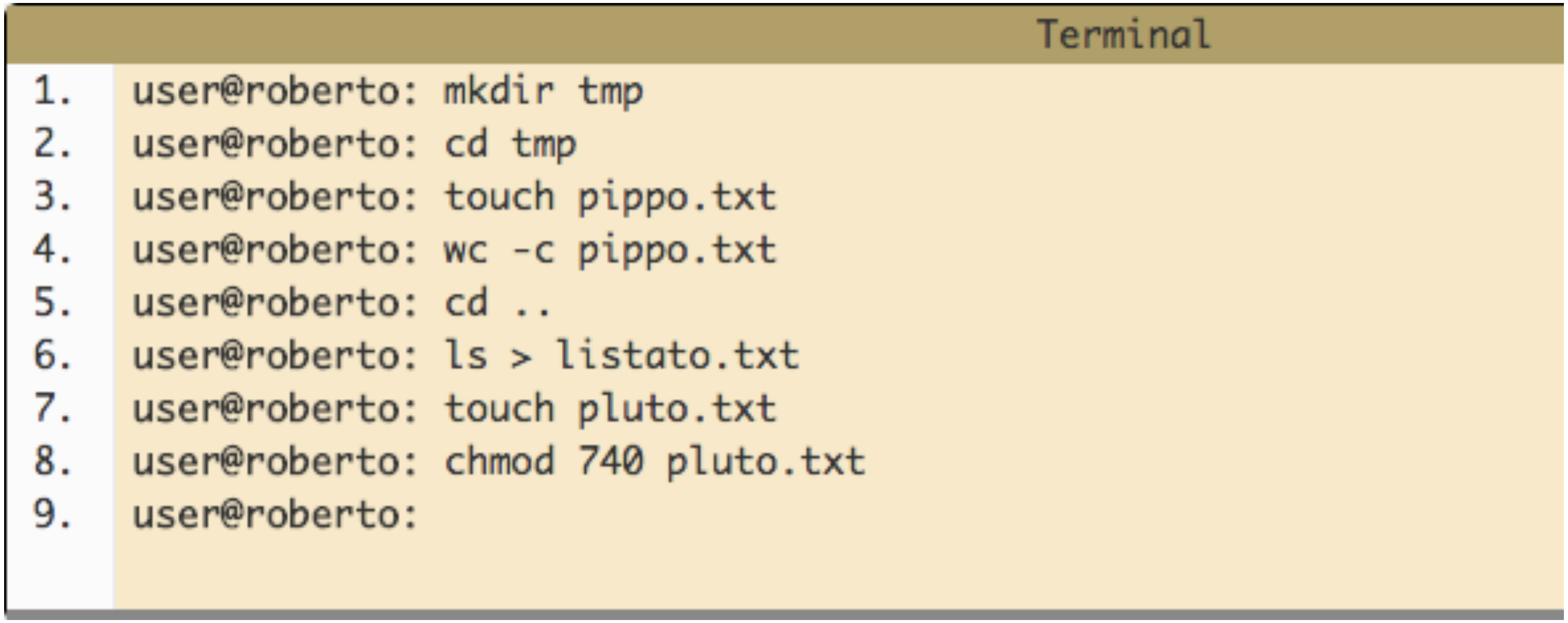
```
user@roberto: mkdir first_exercise
user@roberto: cd first_exercise
user@roberto: cp /bin/ls ./
user@roberto: ls -l
-rwxr-xr-x 1 mac staff 34736 25 Ott 11:47 ls
user@roberto: chmod g-x ls
user@roberto: ls -l
-rwxr--r-x 1 mac staff 34736 25 Ott 11:47 ls
user@roberto:
```

Exercise 3

- *Execute a set of linux commands as follows*

1. *create the directory* ***tmp***
2. *enter into tmp*
3. *create a text file* ***pippo.txt***
4. *view the dimension (byte) of pippo.txt*
5. *go to the "father" directory*
6. *list all files and redirect into* ***listato.txt***
7. *create a text file* ***pluto.txt***
8. *regarding pluto.txt, assign read and write permissions to the owner, read permission to the group and no permissions to the others*

Exercise 3



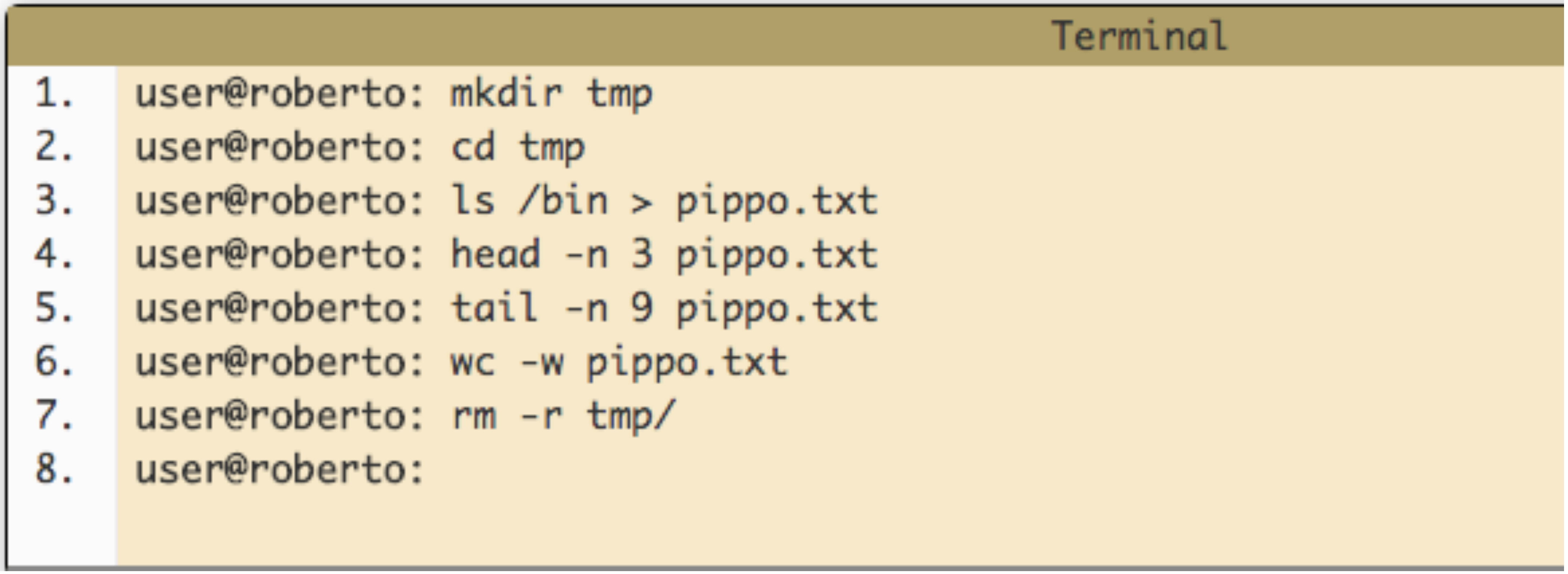
```
user@roberto: mkdir tmp
user@roberto: cd tmp
user@roberto: touch pippo.txt
user@roberto: wc -c pippo.txt
user@roberto: cd ..
user@roberto: ls > listato.txt
user@roberto: touch pluto.txt
user@roberto: chmod 740 pluto.txt
user@roberto:
```

Exercise 4

- *Execute a set of linux commands as follows*

1. *create the directory* ***tmp***
2. *enter into tmp*
3. *list the files of the directory* ***/bin*** *and redirect to* ***pippo.txt***
4. *view the first three rows of pippo.txt*
5. *view the last nine rows of pippo.txt*
6. *view the number of words in pippo.txt*
7. *remove the directory tmp with all contents*

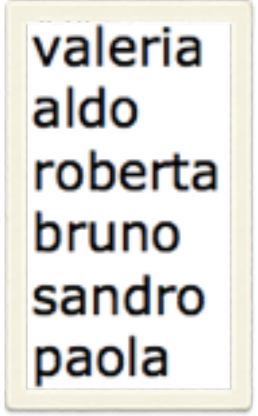
Exercise 4



```
user@roberto: mkdir tmp
user@roberto: cd tmp
user@roberto: ls /bin > pippo.txt
user@roberto: head -n 3 pippo.txt
user@roberto: tail -n 9 pippo.txt
user@roberto: wc -w pippo.txt
user@roberto: rm -r tmp/
user@roberto:
```

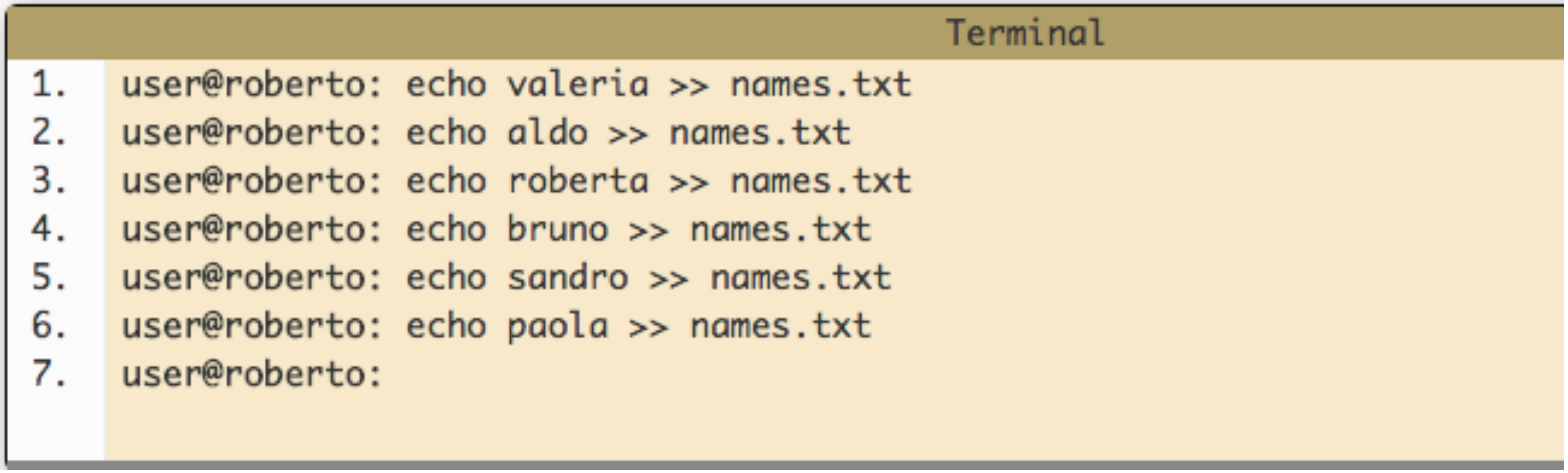
Exercise 5

- *Using the linux shell, build the text file* ***names.txt***



```
valeria
aldo
roberta
bruno
sandro
paola
```

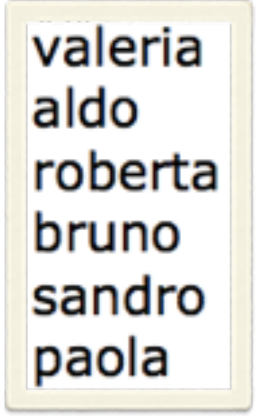
Exercise 5



```
user@roberto: echo valeria >> names.txt
user@roberto: echo aldo >> names.txt
user@roberto: echo roberta >> names.txt
user@roberto: echo bruno >> names.txt
user@roberto: echo sandro >> names.txt
user@roberto: echo paola >> names.txt
user@roberto:
```

Exercise 6

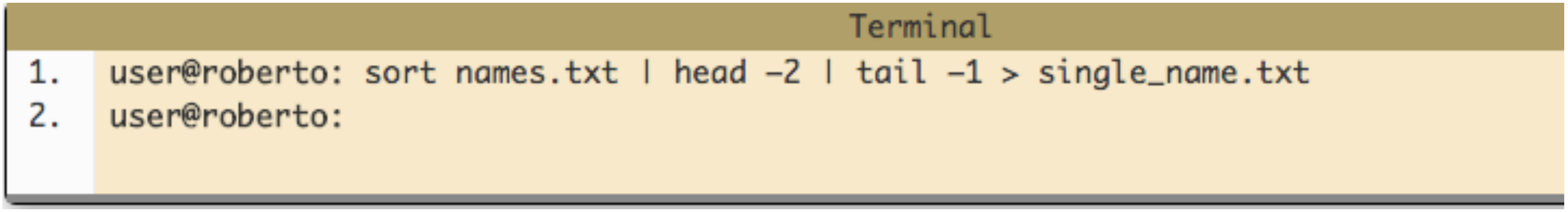
- *Using the text file* ***names.txt***



```
valeria
aldo
roberta
bruno
sandro
paola
```

- *Execute a combination of linux commands for creating a new text file* ***single_name.txt*** *containing the row of names.txt coming as second row in the Lexicographical order (in our case the row is* ***bruno****).*

Exercise 6

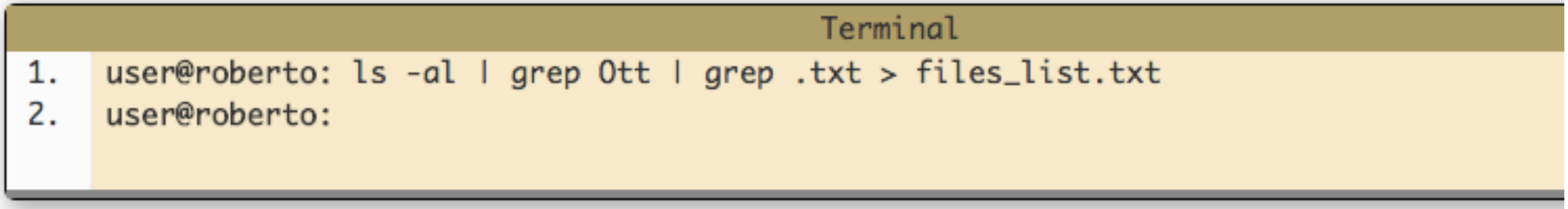


```
user@roberto: sort names.txt | head -2 | tail -1 > single_name.txt
user@roberto:
```

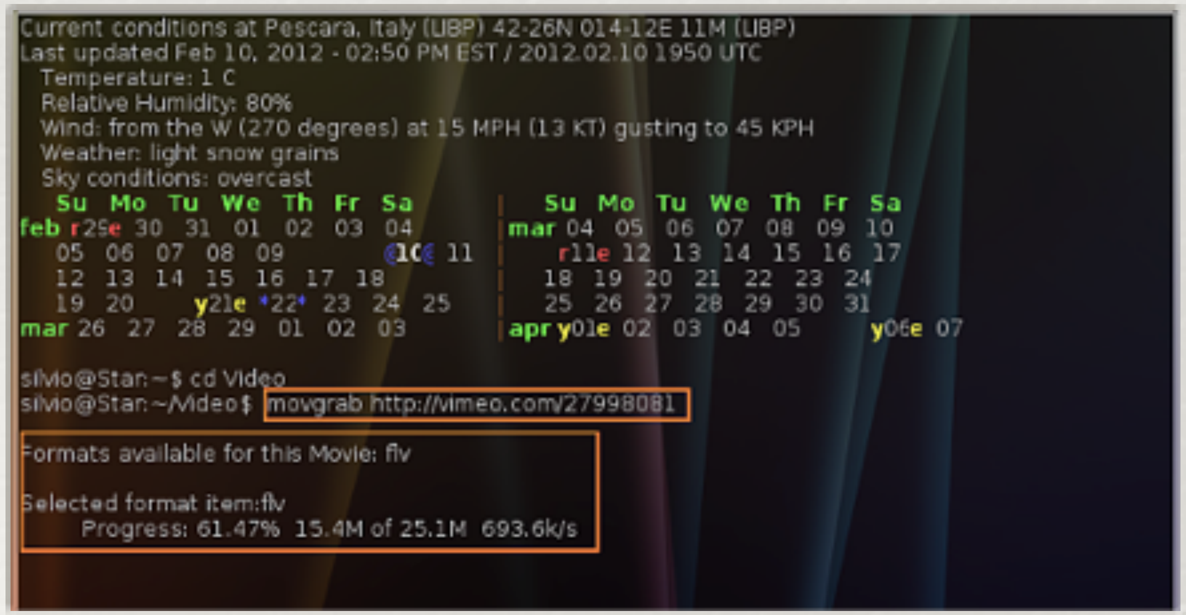
Exercise 7

- *Execute a combination of linux commands for creating a new text file* ***files_list.txt*** *containing the list of files (both visible and invisible) ending with* ***.txt*** *and built in* ***October****.*

Exercise 7



```
user@roberto: ls -al | grep Ott | grep .txt > files_list.txt
user@roberto:
```



```
Current conditions at Pescara, Italy (LIBP) 42-26N 014-12E 11M (LIBP)
Last updated Feb 10, 2012 - 02:50 PM EST / 2012.02.10 1950 UTC
  Temperature: 1 C
  Relative Humidity: 80%
  Wind: from the W (270 degrees) at 15 MPH (13 KT) gusting to 45 KPH
  Weather: light snow grains
  Sky conditions: overcast
   Su  Mo  Tu  We  Th  Fr  Sa    |   Su  Mo  Tu  We  Th  Fr  Sa
feb r29e 30  31  01  02  03  04  | mar 04  05  06  07  08  09  10
   05  06  07  08  09      10  11 |    r11e 12  13  14  15  16  17
   12  13  14  15  16  17  18    |   18  19  20  21  22  23  24
   19  20     y21e *22* 23  24  25 |   25  26  27  28  29  30  31
mar 26  27  28  29  01  02  03    | apr y01e 02  03  04  05      y06e 07

silvio@Star:~$ cd Video
silvio@Star:~/Video$ movgrab http://vimeo.com/27998081

Formats available for this Movie: flv

Selected format item:flv
     Progress: 61.47%  15.4M of 25.1M  693.6k/s
```