

ΕΡΓΑΣΤΗΡΙΟ 1

Εξοικείωση με το Linux

1. Συνδεθείτε στο σύστημα.

Ανάλογα με το σύστημα είτε η σύνδεση γίνεται αυτόματα (autologin) είτε με το χέρι καταχωρώντας το login και το password. Στο Ubuntu linux η σύνδεση γίνεται αυτόματα και ο χρήστης μεταφέρεται απευθείας στο γραφικό περιβάλλον.

2. Εκτελέστε την εντολή whoami για να εκτυπώσετε το login name.

```
amarg@amarg-vbox:~$ whoami
amarg
```

3. Εκτελέστε την εντολή pwd για να εκτυπώσετε τον τρέχοντα κατάλογο.

```
amarg@amarg-vbox:~$ pwd
/home/amarg
```

4. Εκτελέστε τις εντολές ls, ls -l και ls -la. Τι παρατηρείτε?

Η εντολή ls εκτυπώνει μόνο τα ονόματα των αρχείων και των υποκαταλόγων του καταλόγου που δέχεται ως όρισμα.

```
amarg@amarg-vbox:~$ ls
acc1          flength1      myCopy.c      prog.c         sigExample5.c
acc1.o        flength1.c    myCopy.o      prog.o         sigExample5.o
a.out         flength1.o    myCopy.o      Public         sigExample6
awkscript     flength.c     mydir         results        sigExample6a
check         flength.o     mydir2        rfile          sigExample6a.o
```

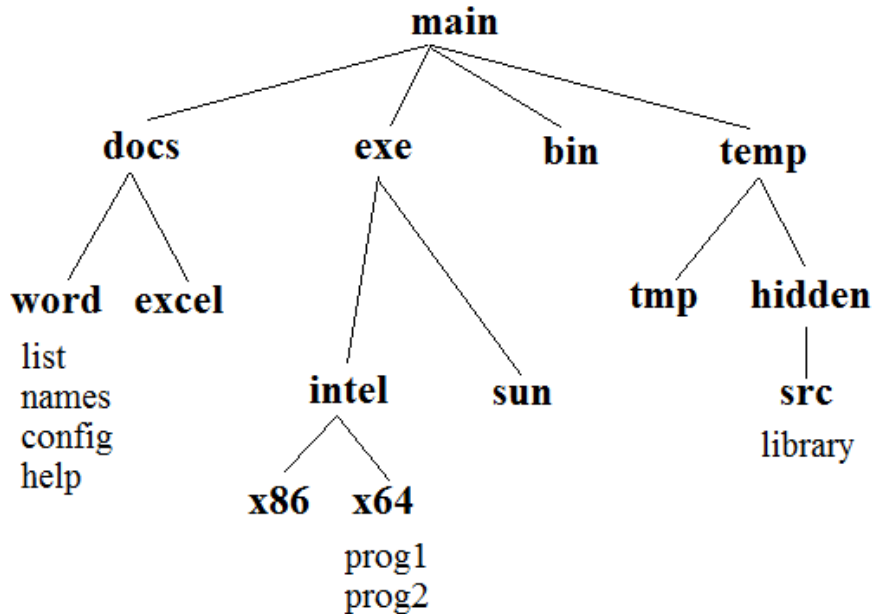
Η εντολή ls -l εκτυπώνει μία γραμμή για κάθε αρχείο ή υποκατάλογο του τρέχοντος καταλόγου η οποία περιλαμβάνει: τα δικαιώματα πρόσβασης, το πλήθος των συνδέσμων προς το αρχείο ή τον κατάλογο, τον κάτοχο και την ομάδα του κατόχου του αρχείου ή του υποκαταλόγου, το μέγεθός του, την ημερομηνία και την ώρα της τελευταίας προσπέλασης και το όνομά του. Όσες γραμμές ξεκινούν με το γράμμα d αναφέρονται σε καταλόγους, ενώ όσες γραμμές ξεκινούν με μία παύλα (-) αναφέρονται σε κανονικά αρχεία.

```
amarg@amarg-vbox:~$ ls -l
total 49544
-rwxrwxr-x 1 amarg amarg 16872 Σεν 29 12:13 acc1
-rw-rw-r-- 1 amarg amarg 2264 Σεν 29 12:13 acc1.o
-rwxrwxr-x 1 amarg amarg 16952 Σεν 29 20:44 a.out
-rw-rw-r-- 1 amarg amarg 167 Σεν 22 10:27 awkscript
-rwxrwxr-x 1 amarg amarg 214 Σεν 22 09:11 check
-rw-rw-r-- 1 amarg amarg 189 Σεν 29 11:54 chownEx.c
-rw-rw-r-- 1 amarg amarg 1680 Σεν 29 11:55 chownEx.o
```

Η εντολή ls -la εκτυπώνει εκτός από τα συνήθη αρχεία που εκτυπώνει η ls -l και τα κρυφά αρχεία και καταλόγους, το όνομα των οποίων ξεκινά με τελεία (.).

```
amarg@amarg-vbox:~$ ls -l -a
total 49608
drwxr-xr-x 28 amarg amarg 4096 Οκτ 11 18:06 .
drwxr-xr-x 3 root root 4096 Σεν 19 19:57 ..
-rwxrwxr-x 1 amarg amarg 16872 Σεν 29 12:13 acc1
-rw-rw-r-- 1 amarg amarg 2264 Σεν 29 12:13 acc1.o
-rwxrwxr-x 1 amarg amarg 16952 Σεν 29 20:44 a.out
-rw-rw-r-- 1 amarg amarg 167 Σεν 22 10:27 awkscript
-rw----- 1 amarg amarg 7998 Οκτ 9 12:15 .bash_history
-rw-r--r-- 1 amarg amarg 220 Σεν 19 19:57 .bash_logout
-rw-r--r-- 1 amarg amarg 3771 Σεν 19 19:57 .bashrc
drwx----- 11 amarg amarg 4096 Σεν 19 20:18 .cache
-rwxrwxr-x 1 amarg amarg 214 Σεν 22 09:11 check
```

5. Να κατασκευάσετε το επόμενο δέντρο καταλόγων (Υποδείξεις: τα ονόματα με έντονη γραφή [**main**, **docs**, **exe**, **bin**, **temp**, **word**, **excel**, **intel**, **x86**, **x64**, **sun**, **tmp**, **hidden**, **src**] είναι ονόματα καταλόγων, ενώ τα υπόλοιπα [**list**, **names**, **config**, **help**, **prog1**, **prog2**, **library**] είναι ονόματα κενών αρχείων κειμένου. Το υποδέντρο με ρίζα το **temp** θα κατασκευαστεί από τον κατάλογο **excel** ενώ όλα τα υπόλοιπα τμήματα του δέντρου από τον κατάλογο **main**.



Μεταφερόμαστε στον προσωπικό μας κατάλογο εκτελώντας την εντολή **cd** και μετά εκτελούμε την εντολή **rm -r main** για να διαγράψουμε το δέντρο με ρίζα το **main** εάν υπάρχει γιατί στην περίπτωση αυτή είναι αδύνατο να δημιουργήσουμε κατάλογο με το ίδιο όνομα **main** στην ίδια θέση. Στη συνέχεια δημιουργούμε τον κατάλογο **main** εκτελώντας την εντολή **mkdir main** και μεταβαίνουμε σε αυτόν τον κατάλογο με την εντολή **cd main** αφού σύμφωνα με την εκφώνηση το μεγαλύτερο τμήμα του δέντρου θα πρέπει να κατασκευαστεί ευρισκόμενοι στον κατάλογο **main**.

```

amarg@amarg-vbox:~$ cd
amarg@amarg-vbox:~$ rm -r main
amarg@amarg-vbox:~$ mkdir main
amarg@amarg-vbox:~$ cd main
amarg@amarg-vbox:~/main$
    
```

Το υπόλοιπο τμήμα του δέντρου μπορεί να κατασκευαστεί με πολλούς τρόπους. Για παράδειγμα, μπορούμε να εργασθούμε οριζόντια, κατασκευάζοντας αρχικά τους καταλόγους **docs**, **exe**, **bin** και **temp**, στη συνέχεια να μπούμε διαδοχικά (α) στον κατάλογο **docs** και να κατασκευάσουμε τους καταλόγους **word** και **excel**, (β) στον κατάλογο **exe** και να κατασκευάσουμε τους καταλόγους **intel** και **sun**, κ.ο.κ. Η άλλη λύση είναι να κινηθούμε κατακόρυφα δηλαδή να κατασκευάσουμε τον κατάλογο **docs** και οτιδήποτε υπάρχει κάτω από αυτόν, ύστερα τον κατάλογο **exe** και οτιδήποτε υπάρχει κάτω από αυτόν, κ.ο.κ. Επειδή η δεύτερη προσέγγιση μας επιτρέπει να εξοικειωθούμε με την έννοια του υποδέντρου, είναι και αυτή που τελικά θα επιλέξουμε. Οι καταλόγοι του δέντρου θα κατασκευαστούν με την **mkdir**, ενώ τα αρχεία που υπάρχουν σε αυτούς θα κατασκευαστούν με την **touch**. Το τελευταίο υποδέντρο με ρίζα το **temp**, θα κατασκευαστεί σε δεύτερο χρόνο, αφού σύμφωνα με την εκφώνηση αυτό θα πρέπει να κατασκευαστεί από τον κατάλογο **excel**. Σε πρώτη φάση λοιπόν έχουμε:

```

amarg@amarg-vbox:~/main$ mkdir docs
amarg@amarg-vbox:~/main$ mkdir docs/word
amarg@amarg-vbox:~/main$ touch docs/word/list
amarg@amarg-vbox:~/main$ touch docs/word/names
amarg@amarg-vbox:~/main$ touch docs/word/config
amarg@amarg-vbox:~/main$ touch docs/word/help
amarg@amarg-vbox:~/main$ mkdir docs/excel
amarg@amarg-vbox:~/main$ mkdir exe
amarg@amarg-vbox:~/main$ mkdir exe/intel
amarg@amarg-vbox:~/main$ mkdir exe/intel/x86
amarg@amarg-vbox:~/main$ mkdir exe/intel/x64
amarg@amarg-vbox:~/main$ touch exe/intel/x64/prog1
amarg@amarg-vbox:~/main$ touch exe/intel/x64/prog2
amarg@amarg-vbox:~/main$ mkdir bin
amarg@amarg-vbox:~/main$

```

Ο τελευταίος κλάδος του δέντρου που περιλαμβάνει το `temp` και οτιδήποτε βρίσκεται από κάτω, θα κατασκευαστεί από τον κατάλογο `excel`. Αρχικά πρέπει να πάμε σε αυτόν τον κατάλογο και επειδή βρισκόμαστε στη `main` αυτό γίνεται εκτελώντας την εντολή **`cd docs/excel`**. Παρατηρούμε τώρα πως ο κατάλογος **`temp`** θα πρέπει να κατασκευαστεί κάτω από τον κατάλογο **`main`** ο οποίος βρίσκεται κάτω από τον προσωπικό μας κατάλογο `~`. Κατά συνέπεια, αυτό το τμήμα του δέντρου θα κατασκευαστεί με την επόμενη αλληλουχία εντολών:

```

amarg@amarg-vbox:~/main/docs/excel$ mkdir ~/main/temp
amarg@amarg-vbox:~/main/docs/excel$ mkdir ~/main/temp/tmp
amarg@amarg-vbox:~/main/docs/excel$ mkdir ~/main/temp/tmp/hidden
amarg@amarg-vbox:~/main/docs/excel$ mkdir ~/main/temp/tmp/hidden/src
amarg@amarg-vbox:~/main/docs/excel$ touch ~/main/temp/tmp/hidden/src/library
amarg@amarg-vbox:~/main/docs/excel$

```

ΣΧΟΛΙΑ. Η παραπάνω λύση είναι ενδεικτική αλλά αποτελεί έναν από τους ενδεδειγμένους τρόπους κατασκευής του δέντρου. Ο άλλος τρόπος είναι να μην χρησιμοποιούμε ονόματα διαδρομών αλλά μόνο ονόματα καταλόγων – ωστόσο, επειδή εάν χρησιμοποιήσουμε στην `mkdir` απλά ονόματα καταλόγων, τότε, επειδή ο νέος κατάλογος δημιουργείται στον κατάλογο στον οποίο βρισκόμαστε, θα πρέπει συνέχεια να χρησιμοποιούμε την εντολή `cd` για να μεταφερόμαστε στον κατάλογο μέσα στον οποίο θέλουμε να κατασκευάσουμε το νέο κατάλογο, κάτι που είναι αρκετά χρονοβόρο. Για παράδειγμα, εάν είμαστε στον κατάλογο `main` και θέλουμε να κατασκευάσουμε τον κατάλογο `docs` και μέσα σε αυτόν τον κατάλογο `word`, αντί να γράψουμε `mkdir docs` και μετά `mkdir docs/word` μπορούμε αρχικά να κατασκευάσουμε τον κατάλογο `docs` γράφοντας `mkdir docs`, μετά να μεταβούμε σε αυτόν τον κατάλογο γράφοντας `cd docs` και αφού μπούμε στον κατάλογο `docs` να δημιουργήσουμε τον κατάλογο `word` με την εντολή `mkdir word`. Μετά θα πρέπει να επιστρέψουμε στον κατάλογο `docs` γράφοντας `cd ..` (οι δύο τελείες `..` πάντα συμβολίζουν τον γονικό κατάλογο του καταλόγου στον οποίο βρισκόμαστε) και να δημιουργήσουμε τον κατάλογο `excel` ως `mkdir excel`. Από τον κατάλογο `docs` ανεβαίνουμε στον `main` γράφοντας `cd ..` και κατασκευάζουμε τον κατάλογο `exe` ως `mkdir exe`, κ.ο.κ. Πρόκειται ασφαλώς για μία χρονοβόρα διαδικασία.

Τα παραπάνω είναι ουσιαστικά και θα πρέπει να κατανοηθούν πλήρως προκειμένου να επιλυθούν οι υπόλοιπες ασκήσεις. Από οποιοδήποτε σημείο του δέντρου *A* και αν βρισκόμαστε, η διαδρομή προς ένα οποιοδήποτε άλλο σημείο *B* είναι **μία και μοναδική**. Πώς θα πάμε για παράδειγμα, από τον κατάλογο `x64` στον κατάλογο `word`? Θα πρέπει να ανεβούμε στον κατάλογο `intel` (γονικός κατάλογος του `x64`, άρα `..`), από τον `intel` να ανέβουμε πιο πάνω στον `exe` (γονικός κατάλογος του `intel`, άρα `..`) από τον `exe` να ανέβουμε ακόμη πιο ψηλά, στον `main` (γονικός κατάλογος του `exe`, άρα `..`) και μετά να κατεβούμε αρχικά στον `docs` και μετά στον `word`. Για να πάμε λοιπόν από τον `x64` στον `word` θα πρέπει να εκτελέσουμε την ακόλουθη αλληλουχία εντολών:

```
amarg@amarg-vbox:~/main/exe/intel/x64$ pwd
/home/amarg/main/exe/intel/x64 ----- εκκίνηση
amarg@amarg-vbox:~/main/exe/intel/x64$ cd ..
amarg@amarg-vbox:~/main/exe/intel$ cd ..
amarg@amarg-vbox:~/main/exe$ cd ..
amarg@amarg-vbox:~/main$ cd docs
amarg@amarg-vbox:~/main/docs$ cd word
amarg@amarg-vbox:~/main/docs/word$ pwd
/home/amarg/main/docs/word ----- τερματισμός
amarg@amarg-vbox:~/main/docs/word$
```

Αντί να χρησιμοποιήσουμε πέντε φορές την εντολή `cd` με ορίσματα (1) `..` (2) `..` (3) `..` (4) `docs` και (5) `word` μπορούμε να κατασκευάσουμε τη διαδρομή ανάμεσα στους καταλόγους `x64` και `docs` συνδυάζοντας τα παραπάνω ορίσματα με τον χαρακτήρα `/`. Αυτή η διαδρομή είναι η

`../../docs/word`

Έχοντας κατασκευάσει αυτή τη διαδρομή, μπορούμε να μεταβούμε από τον κατάλογο `x64` στον κατάλογο `word` με μία απλή κλήση της εντολής `cd` που θα δέχεται ως όρισμα την παραπάνω διαδρομή με το αποτέλεσμα να είναι το ίδιο:

```
amarg@amarg-vbox:~/main/exe/intel/x64$ pwd
/home/amarg/main/exe/intel/x64
amarg@amarg-vbox:~/main/exe/intel/x64$ cd ../../docs/word
amarg@amarg-vbox:~/main/docs/word$ pwd
/home/amarg/main/docs/word
amarg@amarg-vbox:~/main/docs/word$
```

Με άλλα λόγια, η απλή κλήση της `cd` με όρισμα μία διαδρομή που αποτελείται από πέντε τμήματα, είναι ισοδύναμη με πέντε κλήσεις της `cd` με την κάθε `cd` να παίρνει διαδοχικά τα ορίσματα (1), (2), (3), (4) και (5).

6. Από τον κατάλογο `x64` να εμφανίσετε τα περιεχόμενα του καταλόγου `word`

Η κατάσταση έχει περιγραφεί στο προηγούμενο σχόλιο. Βέβαια, εδώ δεν χρειάζεται να μεταβούμε στον κατάλογο `word` και κατά συνέπεια, δεν χρειάζεται να καλέσουμε την `cd`. Θα καλέσουμε απλά από τον κατάλογο `x64` την εντολή `ls -l` περνώντας ως όρισμα στην εντολή τη διαδρομή προς τον κατάλογο `word` με σημείο εκκίνησης τον κατάλογο `x64`, έτσι ώστε το λειτουργικό σύστημα να μπορεί να εντοπίσει τον κατάλογο `word` στο δέντρο καταλόγων:

```
amarg@amarg-vbox:~/main/exe/intel/x64$ ls -l ../../docs/word
total 0
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 config
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 help
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 list
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 names
amarg@amarg-vbox:~/main/exe/intel/x64$ ----- ΔΕΝ έχουμε φύγει από τον κατάλογο x64 !!!
```

7. Ποια είναι η απόλυτη διαδρομή του αρχείου `prog2`? Ποιες είναι οι σχετικές διαδρομές του αρχείου `prog1` όταν ο τρέχων κατάλογος είναι ο (α) `x86`, (β) `excel` και (γ) `hidden`? Πότε η σχετική διαδρομή προς ένα κατάλογο ταυτίζεται με την απόλυτη διαδρομή;

Η απόλυτη διαδρομή προς ένα αρχείο ή κατάλογο είναι η διαδρομή προς το αρχείο ή τον κατάλογο ξεκινώντας από τη ρίζα του δέντρου ενώ η σχετική διαδρομή είναι διαφορετική ανάλογα με το σημείο εκκίνησης το οποίο μπορεί να είναι οποιοδήποτε. Η σχετική με την απόλυτη διαδρομή ταυτίζονται όταν το σημείο εκκίνησης είναι η ρίζα. Επομένως:

- Η απόλυτη διαδρομή προς το αρχείο `prog2` είναι η `/home/amarg/main/exe/intel/x64/prog2`.
- Η σχετική διαδρομή προς το αρχείο `prog1` όταν βρισκόμαστε στον κατάλογο `x86` είναι η `../x64/prog1`.
- Η σχετική διαδρομή προς το αρχείο `prog1` όταν βρισκόμαστε στον κατάλογο `excel` είναι η `../../exe/intel/x64/prog1`.

- Η σχετική διαδρομή προς το αρχείο *prog1* όταν βρισκόμαστε στον κατάλογο *hidden* είναι η *../exe/intel/x64/prog1*.

8. Από τον κατάλογο *x86* να διαγράψετε με την εντολή *rmdir* τον κατάλογο *exe*. Τι παρατηρείτε?

```
amarg@amarg-vbox:~/main/exe/intel/x86$ rmdir ../..
rmdir: failed to remove '../..': Directory not empty
amarg@amarg-vbox:~/main/exe/intel/x86$
```

Ο κατάλογος δεν διαγράφεται γιατί δεν είναι άδειος και κατά συνέπεια θα πρέπει να χρησιμοποιήσουμε την εντολή *rm -r*. Για να δούμε όμως, διαγράφεται με αυτή την εντολή *rm -r*?

```
amarg@amarg-vbox:~/main/exe/intel/x86$ rm -r ../..
rm: refusing to remove '.' or '..' directory: skipping '../..'
amarg@amarg-vbox:~/main/exe/intel/x86$
```

Ο κατάλογος ΔΕΝ διαγράφεται διότι βρίσκομαι σε υποκατάλογό του !! (ο *x86* είναι υποκατάλογος του *exe*) – είναι σαν να κόβω το κλαδί που κάθομαι: εάν διαγράψω τον κατάλογο *exe* εγώ τι θα γίνω? Είναι ακριβώς το ίδιο παράδοξο με το να γυρίσει κάποιος πίσω στο χρόνο και να σκοτώσει τον παππού του οπότε εξαφανίζεται και ο ίδιος ☺

9. Από τον κατάλογο *main* να διαγράψετε τους καταλόγους *x86* και *hidden*.

```
amarg@amarg-vbox:~/main$ rm -r exe/intel/x86
amarg@amarg-vbox:~/main$ rm -r temp/hidden
amarg@amarg-vbox:~/main$
```

10. Από τον κατάλογο *bin*: (α) να αντιγράψετε το αρχείο *names* στον κατάλογο *x64* με όνομα *prog3*. (β) Να μετακινήσετε το αρχείο *prog1* στον κατάλογο *sun*. (γ) Να μετακινήσετε το αρχείο *list* στον κατάλογο *tmp*.

```
amarg@amarg-vbox:~/main/bin$ cp ../docs/word/names ../exe/intel/x86/prog3
amarg@amarg-vbox:~/main/bin$ mv ../exe/intel/x64 ../exe/sun
amarg@amarg-vbox:~/main/bin$ mv ../docs/word/list ../temp/tmp
amarg@amarg-vbox:~/main/bin$
```

Ας επαληθεύσουμε τις παραπάνω πράξεις: βρίσκονται τα αρχεία στις νέες τους θέσεις? **Φυσικά και βρίσκονται!**

```
amarg@amarg-vbox:~/main/bin$ ls -l ../exe/intel/x86
total 0
-rw-rw-r-- 1 amarg amarg 0 Okt 11 20:50 prog3
amarg@amarg-vbox:~/main/bin$ ls -l ../exe/sun
total 0
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:40 prog1
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:40 prog2
amarg@amarg-vbox:~/main/bin$ ls -l ../temp/tmp
total 4
drwxrwxr-x 3 amarg amarg 4096 Okt 11 19:47 hidd
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 list
amarg@amarg-vbox:~/main/bin$
```

11. Να μεταβείτε στο root directory του συστήματος αρχείων του και να δημιουργήσετε το αρχείο *file1*. Τι παρατηρείτε?

```
amarg@amarg-vbox:~/main/bin$ cd /
amarg@amarg-vbox:/$ touch file1
touch: cannot touch 'file1': Permission denied
```

Η διαδικασία δεν είναι επιτρεπτή διότι οι απλοί χρήστες μπορούν να δημιουργήσουν αρχεία μόνο στον προσωπικό τους κατάλογο. Μόνο ο διαχειριστής του συστήματος (root) έχει δικαίωμα να δημιουργήσει οπουδήποτε αρχεία.

12. Να καταγράψετε τη μάσκα δικαιωμάτων των αρχείων list, config και prog1 στις τρέχουσες θέσεις τους.

Εάν δεν θυμόμαστε πού βρίσκονται τα παραπάνω αρχεία μπορούμε να τα βρούμε με τη find (ξέρουμε πως βρίσκονται στο δέντρο καταλόγων με ρίζα τη main). Στη συνέχεια εμφανίζουμε τη μάσκα δικαιωμάτων με την ls -l.

```
amarg@amarg-vbox:~$ find main -name list
main/temp/tmp/list
amarg@amarg-vbox:~$ ls -l main/temp/tmp/list
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 main/temp/tmp/list
amarg@amarg-vbox:~$ find main -name config
main/docs/word/config
amarg@amarg-vbox:~$ ls -l main/docs/word/config
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 main/docs/word/config
amarg@amarg-vbox:~$ find main -name prog1
main/exe/sun/prog1
amarg@amarg-vbox:~$ ls -l main/exe/sun/prog1
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:40 main/exe/sun/prog1
amarg@amarg-vbox:~$
```

Παρατηρήστε πως η απόλυτη διαδρομή που επέστρεψε η find διαβιβάστηκε ως όρισμα στην εντολή ls -l έτσι ώστε να εμφανίσει μόνο τη γραμμή που αφορά στο συγκεκριμένο αρχείο. Η μάσκα δικαιωμάτων rw-rw-r-- (664) είναι κοινή για όλα τα αρχεία κάτι που είναι αναμενόμενο. Από την τιμή αυτή της μάσκας οδηγούμαστε στο συμπέρασμα πως η τιμή της umask είναι ίση με 002.

13. Ποιος είναι ο κάτοχος και η ομάδα του αρχείου list? Χρησιμοποιώντας τις εντολές chown και chgrp να αλλάξετε αυτές τις παραμέτρους σε τιμές της αρεσκείας σας.

Από την παραπάνω έξοδο διαπιστώνουμε πως ο κάτοχος και η ομάδα του κατόχου για το αρχείο list είναι ο χρήστης amarg (σε άλλους υπολογιστές φυσικά θα βγάλει διαφορετικά αποτελέσματα). Ας ορίσουμε ως κάτοχο και ως ομάδα κατόχου για το αρχείο list το χρήστη και την ομάδα sys. Η διαδικασία ακολουθεί στη συνέχεια.

```
amarg@amarg-vbox:~$ chown sys main/temp/tmp/list
chown: changing ownership of 'main/temp/tmp/list': Operation not permitted
amarg@amarg-vbox:~$ chgrp sys main/temp/tmp/list
chgrp: changing group of 'main/temp/tmp/list': Operation not permitted
amarg@amarg-vbox:~$ sudo chown sys main/temp/tmp/list
amarg@amarg-vbox:~$ sudo chgrp sys main/temp/tmp/list
amarg@amarg-vbox:~$ ls -l main/temp/tmp/list
-rw-rw-r-- 1 sys sys 0 Okt 11 19:39 main/temp/tmp/list
amarg@amarg-vbox:~$
```

Παρατηρήστε πως η απλή κλήση των εντολών δεν είναι επιτρεπτή γιατί οι εντολές chown και chgrp εκτελούνται μόνο από το διαχειριστή του συστήματος (root). Για το λόγο αυτό τις εκτελούμε μέσα από την εντολή sudo δίδοντας τον κωδικό μας αν και όταν ζητηθεί. Στην περίπτωση αυτή οι εντολές εκτελούνται κανονικά όπως φαίνεται από την έξοδο της τελευταίας εντολής ls -l.

14. Η μάσκα δικαιωμάτων του αρχείου list να γίνει 555

```
amarg@amarg-vbox:~$ chmod 555 main/temp/tmp/list
chmod: changing permissions of 'main/temp/tmp/list': Operation not permitted
amarg@amarg-vbox:~$ sudo chown amarg main/temp/tmp/list
amarg@amarg-vbox:~$ sudo chgrp amarg main/temp/tmp/list
amarg@amarg-vbox:~$ chmod 555 main/temp/tmp/list
amarg@amarg-vbox:~$ ls -l main/temp/tmp/list
-r-xr-xr-x 1 amarg amarg 0 Okt 11 19:39 main/temp/tmp/list
amarg@amarg-vbox:~$
```

Παρατηρούμε πως αρχικά η εντολή `chmod` δεν μπορεί να εκτελεστεί και ο λόγος είναι πως δεν είμαστε ο κάτοχος του αρχείου αφού προηγουμένως ως κάτοχο ορίσαμε το χρήστη `sys`. Εάν όμως χρησιμοποιώντας εκ νέου τις εντολές `chown` και `chgrp` επαναφέρουμε τον κάτοχο και την ομάδα του κατόχου στην τιμή `amarg`, η εντολή εκτελείται χωρίς πρόβλημα. Η εντολή `ls -l` μας λέει πως η μάσκα του αρχείου `list` είναι η `r-xr-xr-x` ή ισοδύναμα `555`, όπως άλλωστε ήταν αναμενόμενο.

15. Η μάσκα δικαιωμάτων του καταλόγου word να γίνει 555.

```
amarg@amarg-vbox:~/main$ chmod 555 docs/word
amarg@amarg-vbox:~/main$ ls -l docs/word
total 0
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 config
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 help
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 names
amarg@amarg-vbox:~/main$
```

16. Να διαγράψετε το αρχείο names και να σχολιάσετε το αποτέλεσμα.

```
amarg@amarg-vbox:~/main$ cd docs/word
amarg@amarg-vbox:~/main/docs/word$ ls -l
total 0
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 config
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 help
-rw-rw-r-- 1 amarg amarg 0 Okt 11 19:39 names
amarg@amarg-vbox:~/main/docs/word$ rm names
rm: cannot remove 'names': Permission denied
amarg@amarg-vbox:~/main/docs/word$
```

Παρατηρούμε πως η διαγραφή του αρχείου `names` δεν είναι επιτρεπτή !! Για ποιο λόγο συμβαίνει αυτό? Ο κατάλογος `word` έχει μάσκα δικαιωμάτων `555` δηλαδή `r-xr-xr-x` και κατά συνέπεια καμία ομάδα χρηστών δεν έχει δικαίωμα εγγραφής. Τι σημαίνει εγγραφή για ένα κατάλογο? Το ίδιο ακριβώς που σημαίνει και για ένα αρχείο: δυνατότητα τροποποίησης του περιεχομένου του. Με άλλα λόγια δεν έχουμε δικαίωμα να τροποποιήσουμε τα περιεχόμενα του καταλόγου και ως εκ τούτου η διαγραφή αρχείων από αυτόν (που συνιστά τροποποίηση του περιεχομένου του) δεν είναι επιτρεπτή. Το ίδιο πράγμα θα συνέβαινε και εάν προσπαθούσαμε να δημιουργήσουμε ένα νέο αρχείο σε αυτόν τον κατάλογο όπως φαίνεται στη συνέχεια (αφού και η δημιουργία αρχείου συνιστά τροποποίηση των περιεχομένων του καταλόγου).

```
amarg@amarg-vbox:~/main/docs/word$ rm names
rm: cannot remove 'names': Permission denied
amarg@amarg-vbox:~/main/docs/word$ touch newFile
touch: cannot touch 'newFile': Permission denied
amarg@amarg-vbox:~/main/docs/word$
```

Είναι ενδιαφέρον πως αυτός ο περιορισμός αφορά μόνο στα περιεχόμενα του καταλόγου και όχι στα περιεχόμενα των αρχείων του, τα οποία τροποποιούνται κανονικά !! στο επόμενο παράδειγμα, η εντολή `ls -l` αποθηκεύει χωρίς

κανένα πρόβλημα την έξοδο της στο αρχείο `names` τα περιεχόμενα του οποίου επομένως τροποποιούνται, χωρίς όμως το ίδιο να μπορεί να διαγραφεί !!!

```
amarg@amarg-vbox:~/main/docs/word$ ls -l > names
amarg@amarg-vbox:~/main/docs/word$ cat names
total 0
-rw-rw-r-- 1 amarg amarg 0 0κτ  11 19:39 config
-rw-rw-r-- 1 amarg amarg 0 0κτ  11 19:39 help
-rw-rw-r-- 1 amarg amarg 0 0κτ  12 00:05 names
amarg@amarg-vbox:~/main/docs/word$
```

17. Η μάσκα του καταλόγου `temp` να γίνει 666

```
amarg@amarg-vbox:~$ ls -l main
total 16
drwxrwxr-x 2 amarg amarg 4096 0κτ  11 19:40 bin
drwxrwxr-x 4 amarg amarg 4096 0κτ  11 19:39 docs
drwxrwxr-x 4 amarg amarg 4096 0κτ  11 20:51 exe
drw-rw-rw- 3 amarg amarg 4096 0κτ  11 20:42 temp
amarg@amarg-vbox:~$
```

18. Να διαγράψετε τον κατάλογο `tmp` και να σχολιάσετε το αποτέλεσμα

```
amarg@amarg-vbox:~$ rm -r main/temp/tmp
rm: cannot remove 'main/temp/tmp': Permission denied
amarg@amarg-vbox:~$ cd main/temp
bash: cd: main/temp: Permission denied
amarg@amarg-vbox:~$
```

Παρατηρούμε πως η διαγραφή του καταλόγου `tmp` δεν είναι επιτρεπτή !! Για ποιο λόγο συμβαίνει αυτό? Ο κατάλογος `temp` έχει μάσκα δικαιωμάτων 666 δηλαδή `r w - r w - r w` - και κατά συνέπεια καμία ομάδα χρηστών δεν έχει δικαίωμα εκτέλεσης. Τι σημαίνει εκτέλεση για ένα κατάλογο? Σημαίνει το να μπορεί κανείς να μεταβεί σε αυτόν τον κατάλογο εκτελώντας την εντολή `cd` (*change directory*). Είναι προφανές πως για να μπορέσει το σύστημα να διαγράψει τον κατάλογο `tmp` ο οποίος είναι υποκατάλογος του `temp`, θα πρέπει πρώτα να μπει σε αυτόν τον κατάλογο με την `cd`, διαδικασία η οποία δεν είναι επιτρεπτή αφού απαγορεύεται από τα δικαιώματα πρόσβασης. Για το λόγο αυτό η διαγραφή του καταλόγου `tmp` δεν επιτρέπεται. Αυτό είναι κάτι που πιστοποιείται και από την εκτέλεση της εντολής `cd tmp` η οποία δεν είναι επιτρεπτή: η μάσκα δικαιωμάτων 666 δεν επιτρέπει τη μετάβαση στον κατάλογο `temp`!!

19. Η μάσκα του καταλόγου `x86` να γίνει 333

```
amarg@amarg-vbox:~$ chmod 333 main/exe/intel/x86
amarg@amarg-vbox:~$ ls -l main/exe/intel
total 4
d-wx-wx-wx 2 amarg amarg 4096 0κτ  11 20:49 x86
amarg@amarg-vbox:~$
```

20. Να εμφανίσετε τα περιεχόμενα του καταλόγου `x64` και να σχολιάσετε το αποτέλεσμα.

```
amarg@amarg-vbox:~$ ls main/exe/intel/x86
ls: cannot open directory 'main/exe/intel/x86': Permission denied
amarg@amarg-vbox:~$ ls -l main/exe/intel/x86
ls: cannot open directory 'main/exe/intel/x86': Permission denied
amarg@amarg-vbox:~$
```

Παρατηρούμε πως η εμφάνιση των περιεχομένων του καταλόγου `x64` δεν είναι επιτρεπτή !! Για ποιο λόγο συμβαίνει αυτό? Ο κατάλογος `x64` έχει μάσκα δικαιωμάτων 333 δηλαδή `- w x - w x - w x` και κατά συνέπεια καμία ομάδα χρηστών δεν έχει δικαίωμα ανάγνωσης. Τι σημαίνει ανάγνωση για ένα κατάλογο? Σημαίνει να μπορεί κάποιος να

εκτελέσει την `ls` για να εμφανίσει τα περιεχόμενα του καταλόγου. Επειδή όμως η τρέχουσα μάσκα δικαιωμάτων δεν επιτρέπει κάτι τέτοιο, η εντολή `ls` δεν είναι δυνατόν να εκτελεστεί. Αυτό είναι κάτι που συνηθίζεται στους καταλόγους αρχείων των `ftp servers` έτσι ώστε τα περιεχόμενα των καταλόγων του συστήματος να μην είναι ορατά σε όλους.

21. Να οριστεί ως default mode για τους νέους καταλόγους το 744. Ποια είναι η τιμή της μάσκας για τα αρχεία?

Για να ορίσουμε ως default mode για καταλόγους το 744 θα πρέπει να χρησιμοποιήσουμε τιμή `umask` ίση με 033 αφού είναι $777 - 033 = 744$. Με άλλα λόγια, όλοι οι κατάλογοι που θα δημιουργηθούν μετά τον ορισμό της μάσκας στο 033 θα έχουν δικαιώματα πρόσβασης `r w x r - - r - -` (744) ενώ όλα τα αρχεία που θα δημιουργηθούν μετά τον ορισμό της μάσκας στο 033 θα έχουν δικαιώματα πρόσβασης `r w - r - - r - -` (644) (δηλαδή αφαιρούμε το `x` από όλες τις ομάδες χρηστών) όπως πιστοποιείται από το επόμενο παράδειγμα.

```
amarg@amarg-vbox:~$ umask 033
amarg@amarg-vbox:~$ touch f0003
amarg@amarg-vbox:~$ ls -l f0003
-rw-r--r-- 1 amarg amarg 0 Okt 12 01:04 f0003
amarg@amarg-vbox:~$ mkdir d0003
amarg@amarg-vbox:~$ ls -l | grep d0003
drwxr--r-- 2 amarg amarg 4096 Okt 12 01:04 d0003
amarg@amarg-vbox:~$
```

22. Να εμφανίσετε τα περιεχόμενα του αρχείου `/etc/passwd`.

```
amarg@amarg-vbox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

(στο screenshot δεν φαίνονται όλες οι γραμμές αλλά μόνο οι πρώτες εξ αυτών)

23. Να αναζητήσετε τα αρχεία του καταλόγου `/usr` με μέγεθος από 100 έως 500 kbytes και για το καθένα από αυτά να εκτυπώσετε το όνομά τους, το μέγεθός τους και τη μάσκα δικαιωμάτων τους.

```
amarg@amarg-vbox:~$ find /usr -size +100 -size -500 -printf "%f\t%s\t%m\n"
fcnal-test.sh      81322    755
ktest.pl           100428   755
btrfs.h 57736      644
ext4.h 68404      644
emu10k1.h          90803    644
ocelot_hsio.h      56321    644
bmp-abt.h          71087    644
videodev2.h        91601    644
v4l2-controls.h    52087    644
ethtool.h          74798    644
cec-funcs.h        53025    644
bpf.h 138346      644
```

(στο screenshot δεν φαίνονται όλες οι γραμμές αλλά μόνο οι πρώτες εξ αυτών)

24. Να εκτυπωθεί η απόλυτη διαδρομή της εντολής `find`

```
amarg@amarg-vbox:~$ which find
/usr/bin/find
```

25. Να εκτυπωθούν οι απόλυτες διαδρομές των αρχείων που περιέχουν στο όνομά τους τη συμβολοσειρά find.

```
amarg@amarg-vbox:~$ whereis find
find: /usr/bin/find /usr/share/man/man1/find.1.gz /usr/share/info/find.info-1.gz
      /usr/share/info/find.info.gz /usr/share/info/find.info-2.gz
```

26. Να εκτυπωθεί το πλήρες περιεχόμενο του root directory – δηλαδή το δικό του και όλων των υποκαταλόγων του.

Στην εικόνα φαίνονται οι λίγες πρώτες από τις χιλιάδες γραμμές της εξόδου της εντολής

```
amarg@amarg-vbox:~$ ls -lR /
/:
total 970048
lrwxrwxrwx    1 root root          7 Σεπ  19 19:50 bin -> usr/bin
drwxr-xr-x    3 root root      4096 Σεπ  26 09:47 boot
drwxrwxr-x    2 root root      4096 Σεπ  19 19:55 cdrom
drwxr-xr-x   18 root root      4060 Οκτ  12 09:42 dev
drwxr-xr-x  130 root root     12288 Οκτ  11 18:19 etc
drwxr-xr-x    3 root root      4096 Σεπ  19 19:57 home
```

27. Να εκτυπώσετε τις 5 πρώτες και τις 5 τελευταίες γραμμές του αρχείου /etc/passwd.

Εμφάνιση πέντε πρώτων γραμμών

```
amarg@amarg-vbox:~$ head -n 5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

Εμφάνιση πέντε τελευταίων γραμμών

```
amarg@amarg-vbox:~$ tail -n 5 /etc/passwd
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
amarg:x:1000:1000:Athanasios Margaris,,,:/home/amarg:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper::/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

28. Να εμφανίσετε τη δεντρική δομή του ριζικού καταλόγου του συστήματος.

Η έξοδος της εντολής είναι τεράστια και εκτυπώνεται ταχύτατα. Για να εμφανίσουμε ένα τμήμα της αποθηκεύουμε την έξοδο στο αρχείο treeOut με τον τελεστή ανακατεύθυνσης εξόδου > (το αρχείο που προκύπτει έχει μέγεθος 48 Mbytes και περιέχει περίπου 628 χιλιάδες γραμμές) και στη συνέχεια εμφανίζουμε τις 20 πρώτες γραμμές του με την εντολή head (στον υπολογιστή μου η εντολή tree δεν είχε εγκατασταθεί κατά την εγκατάσταση του λειτουργικού και το σύστημα ζήτησε και την εγκατέστησε επί τόπου).

```

amarg@amarg-vbox:~$ tree / > treeOut
amarg@amarg-vbox:~$ wc -l treeOut
628428 treeOut
amarg@amarg-vbox:~$ ls -l treeOut
-rw-rw-r-- 1 amarg amarg 48360825 Okt 12 10:06 treeOut
amarg@amarg-vbox:~$ head -n 20 treeOut
/
├── bin -> usr/bin
├── boot
│   ├── config-5.4.0-47-generic
│   ├── config-5.4.0-48-generic
│   ├── grub
│   │   ├── fonts
│   │   │   └── unicode.pf2
│   │   ├── gfxblacklist.txt
│   │   ├── grub.cfg
│   │   ├── grubenv
│   │   ├── i386-pc
│   │   │   ├── 915resolution.mod
│   │   │   ├── acpi.mod
│   │   │   ├── adler32.mod
│   │   │   ├── affs.mod
│   │   │   ├── afs.mod
│   │   │   ├── ahci.mod
│   │   │   ├── all_video.mod
│   │   │   └── aout.mod

```

29. Αναφερόμενοι στην άσκηση με το δέντρο καταλόγων, να μετακινήσετε τον κατάλογο `exe` και όλο το δέντρο που βρίσκεται κάτω από αυτόν στον κατάλογο `bin` ευρισκόμενοι στον προσωπικό σας κατάλογο.

Αρχικά θα πρέπει να ορίσουμε τα κατάλληλα δικαιώματα πρόσβασης διότι αυτά που ορίσαμε στις προηγούμενες ασκήσεις δεν επιτρέπουν την εκτέλεση της άσκησης. Από τον προσωπικό μας κατάλογο και χρησιμοποιώντας την εντολή `chmod` με τη μορφή

`chmod -R 755 main`

ορίζουμε για τον κατάλογο `main` καθώς και για οτιδήποτε βρίσκεται μέσα σε αυτόν (αυτό γίνεται με το διακόπτη `-R`) ως μάσκα δικαιωμάτων, την

`755 → 111 101 101 → rwxr-xr-x`

και πλέον μπορούμε να εκτελέσουμε την άσκηση. Αν και η εντολή `cp` με το διακόπτη `-r` επιτρέπει την αντιγραφή δεντρικής δομής, ενώ η εντολή `rm` με το διακόπτη `-r` επιτρέπει τη διαγραφή δεντρικής δομής, ωστόσο, η εντολή `mv` δεν διαθέτει διακόπτη `-r` που θα επέτρεπε τη μετακίνηση δεντρικής δομής. Για να το κάνουμε αυτό, θα χρησιμοποιήσουμε πρώτα την `cp -r` για την αντιγραφή στη νέα θέση και στη συνέχεια την εντολή `rm -r` για τη διαγραφή από την παλιά θέση με τελικό αποτέλεσμα τη μετακίνηση της δεντρικής δομής από την παλιά στη νέα θέση. Οι παραπάνω εντολές μαζί με την εικόνα της αρχικής δεντρικής δομής παρουσιάζονται στην επόμενη οθόνη

```
amarg@amarg-vbox:~$ chmod -R 755 main
amarg@amarg-vbox:~$ tree main
main
├── bin
├── docs
│   ├── excel
│   └── word
│       ├── config
│       ├── help
│       └── names
├── exe
│   ├── intel
│   │   └── x86
│   │       └── prog3
│   └── sun
│       ├── prog1
│       └── prog2
└── temp
    └── tmp
        ├── hidden
        │   └── src
        │       └── library
        └── list

12 directories, 8 files
amarg@amarg-vbox:~$ cp -r main/exe main/bin
amarg@amarg-vbox:~$ rm -r main/exe
amarg@amarg-vbox:~$
```

ενώ η νέα δεντρική δομή που προέκυψε από την παραπάνω διαδικασία ακολουθεί στη συνέχεια. Από τη μελέτη της επόμενης οθόνης και από τη σύγκρισή της με την προηγούμενη, διαπιστώνουμε πως το δέντρο καταλόγων με ρίζα τον κατάλογο `exe` μετακινήθηκε όντως κάτω από τον κατάλογο `bin`.

```
amarg@amarg-vbox:~$ tree main
main
├── bin
│   └── exe
│       ├── intel
│       │   └── x86
│       │       └── prog3
│       └── sun
│           ├── prog1
│           └── prog2
├── docs
│   ├── excel
│   └── word
│       ├── config
│       ├── help
│       └── names
└── temp
    └── tmp
        ├── hidden
        │   └── src
        │       └── library
        └── list

12 directories, 8 files
amarg@amarg-vbox:~$
```


30. Να διαβάσετε το αρχείο βοήθειας της εντολής `sort` και να ταξινομήσετε τα περιεχόμενα του τρέχοντος καταλόγου κατά φθίνουσα διάταξη ως προς το μέγεθος των αρχείων.

Το αποτέλεσμα αυτής της διαδικασίας παρουσιάζεται στη συνέχεια (εκτυπώνονται οι λίγες πρώτες γραμμές).

```
amarg@amarg-vbox:~$ ls -l | sort -k 5 -n -r
-rw-rw-r-- 1 amarg amarg 48360825 Οκτ 12 10:06 treeOut
-rw-r--r-- 1 amarg amarg 41492888 Οκτ 9 12:47 full
--wxr--r-- 1 amarg amarg 7959821 Σεπ 20 19:58 file2.doc
-rwxrwxr-x 1 amarg amarg 19624 Σεπ 23 15:10 prog
-rw----- 1 amarg amarg 19624 Σεπ 30 12:13 prog2
-rw----- 1 amarg amarg 19624 Σεπ 29 20:47 prog1
-rwxrwxr-x 1 amarg amarg 17480 Οκτ 7 19:28 sigExample8
-rwxrwxr-x 1 amarg amarg 17328 Σεπ 30 15:59 dirFun
-rwxrwxr-x 1 amarg amarg 17272 Οκτ 2 15:23 pipeEx3
-rwxrwxr-x 1 amarg amarg 17208 Οκτ 2 10:44 pipeEx2
-rwxrwxr-x 1 amarg amarg 17184 Οκτ 6 21:14 sigExample6
```

Ο μοναδικός τρόπος για να ανακτήσουμε το μέγεθος των αρχείων είναι η κλήση της εντολής `ls -l`, η έξοδος της οποίας θα χρησιμοποιηθεί στη συνέχεια από τη `sort`, ως είσοδος, για να πραγματοποιήσει την ταξινόμηση. Αυτό μπορεί να γίνει με δύο τρόπους: είτε να αποθηκεύσει η `ls -l` την έξοδό της σε ένα προσωρινό αρχείο, έστω `tempFile` ως

```
ls -l > tempFile
```

το οποίο στη συνέχεια θα χρησιμοποιηθεί ως *input file* της `sort`, η οποία επομένως θα κληθεί ως

```
sort -k 5 -n -r
```

είτε να στείλει απευθείας η `ls` την έξοδό της στη `sort` μέσω μιας τεχνικής που είναι γνωστή ως διασωλήνωση και η οποία θα μελετηθεί με πολύ μεγάλη λεπτομέρεια σε επόμενο μάθημα

```
ls -l | sort -k 5 -n -r
```

(το αποτέλεσμα σε αμφότερες τις περιπτώσεις είναι το ίδιο). Όσον αφορά στη χρήση της `sort`, ο τρόπος με τον οποίο καλείται τεκμηριώνεται ως εξής: (α) η ταξινόμηση θα πρέπει να γίνει ως προς το μέγεθος του αρχείου, που είναι η πέμπτη στήλη στην έξοδο της `ls -l`. Αυτό είναι κάτι που γνωστοποιείται στη `sort` με το διακόπτη `-k 5` (αντιλαμβάνεστε επομένως πως ένα θέλαμε η ταξινόμηση να γίνει ως προς το όνομα του αρχείου, που είναι η πρώτη στήλη θα γράφαμε `-k 1`, k.o.k.). (β) η προεπιλεγμένη συμπεριφορά της `sort` είναι να ταξινομεί την είσοδό της θεωρώντας πως αυτή αποτελείται από λέξεις και όχι από αριθμούς και ως εκ τούτου δεν πραγματοποιεί αριθμητική ταξινόμηση, όπως θέλουμε, αλλά αλφαριθμητική ταξινόμηση, δηλαδή ταξινομεί τις συμβολοσειρές όπως αυτές εμφανίζονται στον τηλεφωνικό κατάλογο. Με άλλα λόγια, μπορεί, ας πούμε, το 1100 ως αριθμός να είναι μεγαλύτερος από το 543, αλλά επειδή λεξικογραφικά το 1 είναι πριν από το 5, πρώτα θα εμφανίσει το 1100 και μετά το 543 που είναι λάθος. Προκειμένου να ενημερώσουμε τη `sort` πως θα πρέπει να πραγματοποιήσει αριθμητική ταξινόμηση, δηλαδή να χειριστεί τις συμβολοσειρές ως αριθμούς, προσθέτουμε το διακόπτη `-n` (από τη λέξη *numeric*), οπότε η είσοδος θα ταξινομηθεί σωστά (για να το δείτε στην πράξη εκτελέστε την παραπάνω εντολή χωρίς το `-n` και εξετάστε το αποτέλεσμα). (γ) Η προεπιλεγμένη συμπεριφορά της `sort` είναι αύξουσα ταξινόμηση, δηλαδή από το μικρότερο προς το μεγαλύτερο. Ωστόσο, εμείς θέλουμε την αντίστροφη, φθίνουσα ταξινόμηση από το μεγαλύτερο προς το μικρότερο και αυτό είναι κάτι που το επιτυγχάνουμε προσθέτοντας το διακόπτη `-r` (από τη λέξη *reverse* → αντίστροφος) (για να το δείτε στην πράξη εκτελέστε την παραπάνω εντολή χωρίς το `-r` και εξετάστε το αποτέλεσμα). Εάν λοιπόν η `sort` κληθεί με τον παραπάνω τρόπο, θα εμφανίσει το αποτέλεσμα της προηγούμενης οθόνης που είναι αυτό που θέλουμε.