

UOPR - Μάθημα 2ο

1. Συμβολικοί και σκληροί σύνδεσμοι (soft links & hard links)
 2. Συμπίεση αρχείων
 3. Ανάγνωση αρχείων και ανακατεύθυνση εισόδου & εξόδου εντολών
 4. Δικαιώματα & ιδιοκτησία αρχείων και καταλόγων
 5. Αλλαγή σε άλλον χρήστη
 6. Αλλαγή δικαιωμάτων
 7. Αλλαγή ιδιοκτησίας
-
-

1. Συμβολικοί και σκληροί σύνδεσμοι (soft links & hard links)

1α] Δημιουργία και διαχείριση σκληρών συνδέσμων (hard links)

Χρησιμοποιούμε την εντολή `stat` για να δούμε **περισσότερες πληροφορίες** για ένα αρχείο, όπως τον **αριθμό inode** και τους **σκληρούς συνδέσμους** του:

```
stat file.txt
```

Το Inode είναι μια δομή δεδομένων σε μορφή μοναδικού αριθμού, στο οποίο περιέχεται πού είναι αποθηκευμένο το αρχείο ή κατάλογος στον δίσκο (σε ποια blocks), και **πληροφορίες**, όπως τα **δικαιώματα**, η **ώρα πρόσβασης**, η **ώρα τροποποίησης**, κ.ά.

Όταν δημιουργούμε ένα αρχείο, το σύστημα αρχείων ομαδοποιεί όλα τις πληροφορίες του αρχείου σε ένα Inode και το συνδέει με **σκληρό σύνδεσμο με το όνομα του αρχείου**.

Inode: 2130106 → σκληρός σύνδεσμος → file.txt

Για να φτιάξω έναν σκληρό σύνδεσμο (`file-hard.txt`) ενός υπάρχοντος αρχείου (`file.txt`) κάνω:

```
ln /home/greeklug/file.txt /home/greeklug/Documents/file-hard.txt
```

Με αυτό δημιουργώ ένα νέο αρχείο που μοιράζεται το ίδιο περιεχόμενο με το αρχικό αρχείο (προσοχή δεν μιλάμε για αντίγραφο αλλά δύο αρχεία που έχουν τα ίδια δεδομένα). Μπορώ να τροποποιήσω οποιοδήποτε από τα δύο αρχεία και αυτόματα το περιεχόμενο, με βάση ότι είναι κοινό, θα υπάρχει και στα δύο αρχεία.

Εκτελώντας την εντολή `stat` , βλέπουμε ότι τα `file.txt` και `file-hard.txt` **μοιράζονται το ίδιο Inode**, αλλά έχουν **διαφορετικά ονόματα** και βρίσκονται σε **διαφορετικές τοποθεσίες**.

Τα δεδομένα αποθηκεύονται μόνο μία φορά, αλλά είναι προσβάσιμα από πολλές θέσεις.

```
stat file.txt
stat file-hard.txt
```

Αν διαγράψουμε το `file.txt`, το `file-hard.txt` δεν θα διαγραφεί, επειδή τα δεδομένα δεν διαγράφονται άμεσα. Το μόνο που αλλάζει είναι ότι το Inode θα έχει πλέον έναν σκληρό σύνδεσμο αντί για δύο.

Αν διαγραφεί και το `file-hard.txt`, τότε το Inode δεν θα έχει κανένα σύνδεσμο και τα δεδομένα θα χαθούν από το σύστημα αρχείων.

Περιορισμοί σκληρών συνδέσμων:

- Δεν μπορούμε να δημιουργήσουμε σκληρούς συνδέσμους για **καταλόγους (directories)** - μόνο για κανονικά αρχεία.
- Δεν μπορούμε να δημιουργήσουμε σκληρούς συνδέσμους **σε διαφορετικά συστήματα αρχείων**.

1β] Δημιουργία και διαχείριση συμβολικών συνδέσμων (soft links or symlinks)

Ένας **συμβολικός σύνδεσμος** είναι μια συντόμευση (shortcut) που δείχνει στο αρχείο, το οποίο με τη σειρά του δείχνει στο Inode:

soft link → αρχείο → Inode

Για να φτιάξω μια συντόμευση (`file-sym.txt`) ενός υπάρχοντος αρχείου (`file.txt`) κάνω:

```
ln -s /home/greeklug/file.txt /home/greeklug/Documents/file-sym.txt
```

Με αυτό δημιουργώ ένα νέο αρχείο `file-sym.txt` που δείχνει σε ένα υπάρχον αρχείο `file.txt`. Το αρχείο του συμβολικού δεσμού δεν περιέχει δεδομένα αλλά μέσω της συντόμευσης φορτώνει τα δεδομένα του αρχείου στόχου της συντόμευσης.

Συνεπώς, αν το αρχείο στόχος διαγραφεί, τότε ο συμβολικός δεσμός σπάει καθώς παραπέμπει σε κάποιο αρχείο που δεν υπάρχει πια!

Μπορούμε επίσης να δημιουργούμε **συμβολικούς συνδέσμους σε καταλόγους**, ή ακόμη και σε αρχεία/καταλόγους που βρίσκονται σε **διαφορετικά συστήματα αρχείων**.

Συνοπτικά:

- Ένας **σκληρός σύνδεσμος** συνδέει **όνομα αρχείου με δεδομένα** σε μια συσκευή αποθήκευσης.
- Ένας **συμβολικός σύνδεσμος** συνδέει **όνομα αρχείου (ή καταλόγου) με άλλο όνομα αρχείου (ή καταλόγου)**, το οποίο μετά δείχνει στα δεδομένα.

2. Συμπύεση αρχείων

Τα δεδομένα των αρχείων συχνά συμπιέζονται για εξοικονόμηση χώρου στον δίσκο και για μείωση του χρόνου μεταφοράς αρχείων πχ. μέσω δικτύου.

Το Linux υποστηρίζει διάφορες μεθόδους συμπίεσης, όπως:

Εντολή	Λειτουργία
<code>gzip</code>	Το πιο συχνά χρησιμοποιούμενο τύπος συμπίεσης σε Linux
<code>bzip2</code>	Παράγει αρχεία μικρότερου μεγέθους από αυτά που παράγει το gzip , αλλά απαιτεί περισσότερο χρόνο
<code>xz</code>	Το πιο αποδοτικό εργαλείο συμπίεσης από πλευράς χώρου σε Linux, με μεγαλύτερο χρόνο από άλλες συμπίεσεις
<code>zip</code>	Η πιο γνωστή μέθοδος συμπίεσης που υποστηρίζεται σε όλα τα λειτουργικά συστήματα. Δεν είναι τόσο αποδοτική όσο άλλες

Αυτές οι τεχνικές διαφέρουν ως προς την αποτελεσματικότητα (πόσο χώρο εξοικονομούν) και τον χρόνο που απαιτούν για τη συμπίεση ή αποσυμπίεση. Γενικά, όσο πιο αποδοτική είναι η μορφή συμπίεσης, τόσο περισσότερος είναι συνήθως ο χρόνος που χρειάζεται.

Επιπλέον, το εργαλείο **tar** χρησιμοποιείται συχνά για να ομαδοποιήσει αρχεία σε ένα αρχείο αρχειοθέτησης και στη συνέχεια να συμπιεστεί ολόκληρο το αρχείο.

Συμπίεση δεδομένων με χρήση του gzip (στηρίζεται στον αλγόριθμο DEFLATE)

Εντολή	Λειτουργία
<code>gzip *</code>	Συμπιέζει όλα τα αρχεία στον τρέχοντα κατάλογο. Κάθε αρχείο μετονομάζεται με κατάληξη .gz .
<code>gzip -r dir1</code>	Συμπιέζει όλα τα αρχεία στον κατάλογο <code>dir1</code> , καθώς και σε όλους τους υποκαταλόγους του.
<code>gunzip -r dir1</code>	Αποσυμπιέζει τα αρχεία που βρίσκονται μέσα στον κατάλογο <code>dir1</code> και σε όλους τους υποκαταλόγους του. Ισοδύναμο με <code>gzip -d</code>
<code>gunzip file.txt.gz</code>	Αποσυμπιέζει το αρχείο <code>file.txt.gz</code>
<code>zcat file.txt.gz</code>	Βλέπω τα περιεχόμενα του συμπιεσμένου αρχείου <code>file.txt.gz</code> , χωρίς να το αποσυμπιέσω

Συμπίεση δεδομένων με χρήση του bzip2 (στηρίζεται στον αλγόριθμο Burrows-Wheeler Transform)

Το **bzip2** παράγει μικρότερα σε μέγεθος αρχεία από το `gzip`, αλλά χρειάζεται περισσότερο χρόνο.

Εντολή	Λειτουργία
<code>bzip2 *</code>	Συμπίεζει όλα τα αρχεία στον τρέχοντα κατάλογο και τα μετονομάζει με κατάληξη .bz2 .
<code>bunzip2 *.bz2</code>	Αποσυμπίεζει όλα τα αρχεία με κατάληξη .bz2 . Ισοδύναμο με bzip2 -d .

Συμπίεση δεδομένων με χρήση του xz (στηρίζεται στον αλγόριθμο LZMA)

Το **xz** είναι το πιο αποδοτικό εργαλείο συμπίεσης ως προς τον χώρο, όμως είναι από τις πιο αργές μεθόδους συμπίεσης.

Εντολή	Λειτουργία
<code>xz *</code>	Συμπίεζει όλα τα αρχεία στον τρέχοντα κατάλογο και τα μετονομάζει με κατάληξη .xz .
<code>xz file.txt</code>	Συμπίεζει το <code>file.txt</code> σε <code>file.txt.xz</code> και διαγράφει το <code>file.txt</code>
<code>xz -d *.xz</code>	Αποσυμπίεζει όλα τα αρχεία με κατάληξη .xz

Χειρισμός αρχείων με χρήση του zip (στηρίζεται στον αλγόριθμο DEFLATE)

Το πρόγραμμα **zip** σπάνια χρησιμοποιείται για συμπίεση αρχείων στο Linux, αλλά μπορεί να χρειαστεί για να ανοίξουμε ή να αποσυμπίεσουμε αρχεία από άλλα λειτουργικά συστήματα, π.χ. Windows.

Εντολή	Λειτουργία
<code>zip /home/greeklug/dir1/greeklug.zip *</code>	Συμπίεζει όλα τα αρχεία μέσα στον κατάλογο <code>/home/greeklug/dir1/</code> , στο αρχείο <code>greeklug.zip</code>
<code>zip -r greeklug.zip ~</code>	Συμπίεζει τον προσωπικό κατάλογο και υποκαταλόγους στο <code>greeklug.zip</code>
<code>unzip greeklug.zip</code>	Εξάγει όλα τα αρχεία από το backup.zip στον τρέχοντα κατάλογο.
<code>unzip -l greeklug.zip</code>	Βλέπω τι περιέχει το αρχείο <code>greeklug.zip</code>

Αρχειοθέτηση και συμπίεση δεδομένων με χρήση του tar

Σε αντίθεση με το zip, συνήθως δεν χρησιμοποιούμε κατευθείαν τις εντολές `gzip` , `bzip2` , `xz` για την συμπίεση, καθώς συμπιέζουν μεμονωμένα τα αρχεία χωρίς να τα "συσκευάζουν" σε έναν φάκελο. Για να το πετύχουμε αυτό, χρησιμοποιούμε το `tar` σε συνδυασμό με τις παραπάνω μεθόδους συμπίεσης. Έτσι σε περιβάλλον Linux, συναντούμε συνήθως συμπιεσμένα αρχεία ως `.tar.gz`, `.tar.bz2`, `.tar.xz` κτλ.

Το **tar** σημαίνει "**t**ape **a**rchive" και επιτρέπει τη δημιουργία ή εξαγωγή αρχείων από ένα **αρχείο αρχειοθέτησης (tarball)**. Παράλληλα, μπορεί να συμπίεσει ή να αποσυμπιέσει περιεχόμενα κατά τη δημιουργία ή εξαγωγή του αρχείου.

Σύνταξη της `tar` στο Linux:

\$ `tar` επιλογές αρχείο-αρχειοθέτησης αρχείο

\$ `tar` επιλογές αρχείο-αρχειοθέτησης /φάκελος

- **tar**: η εντολή.
- **επιλογές**: τροποποιούν τη συμπεριφορά της `tar` .
- **αρχείο-αρχειοθέτησης**: το όνομα του αρχείου που δημιουργούμε ή επεξεργαζόμαστε.
- **αρχείο ή φάκελος προς αρχειοθέτηση**: το περιεχόμενο που θέλουμε να συμπεριλάβουμε.

Εντολή	Λειτουργία
<code>tar -cvf greeklug.tar /home/greeklug/Documents</code>	Δημιουργία ενός tar με όνομα <code>greeklug.tar</code> που περιέχει όλα τα αρχεία μέσα στο <code>/home/greeklug/Documents</code>
<code>tar -cvf /home/greeklug/Desktop/greeklug.tar /home/greeklug/Documents</code>	Δημιουργία ενός tar με όνομα <code>greeklug.tar</code> , στην τοποθεσία <code>/home/greeklug/Desktop/</code> , που περιέχει όλα τα αρχεία μέσα στο <code>/home/greeklug/Documents</code>
<code>tar -tf greeklug.tar</code>	Προβολή των περιεχομένων του <code>greeklug.tar</code>
<code>tar -zcvf dir1.tar.gz dir1</code>	Δημιουργία tarball και συμπίεση χρησιμοποιώντας την <code>gzip</code> , των περιεχομένων του <code>dir1</code> με όνομα <code>dir1.tar.gz</code>
<code>tar -jcvf dir1.tar.bz2 dir1</code>	Δημιουργία tarball και συμπίεση χρησιμοποιώντας την <code>bzip2</code> , των περιεχομένων του <code>dir1</code> με όνομα <code>dir1.tar.bz2</code>
<code>tar -Jcvf dir1.tar.xz dir1</code>	Δημιουργία tarball και συμπίεση χρησιμοποιώντας την <code>xz</code> , των περιεχομένων του <code>dir1</code> με όνομα <code>dir1.tar.bz2</code>

Εντολή	Λειτουργία
<code>tar -xvf dir1.tar.gz</code>	Εξαγωγή του αρχείου <code>dir1.tar.gz</code> , στον τρέχον κατάλογο
<code>tar -xvf dir1.tar.gz -C /home/greeklug/Documents</code>	Εξαγωγή του αρχείου <code>dir1.tar.gz</code> μέσα στον κατάλογο <code>/home/greeklug/Documents</code>
<code>tar -rvf greeklug.tar file1.txt</code>	Προσθήκη του αρχείου <code>file1.txt</code> στο ήδη υπάρχον <code>greeklug.tar</code>
<code>tar -uvf greeklug.tar dir1</code>	Προσθήκη στο <code>greeklug.tar</code> μόνο των νέων αρχείων του <code>dir</code> που δεν είχαν προστεθεί προηγουμένως

Option	Περιγραφή
-c	(create) Δημιουργία νέου αρχείου αρχειοθέτησης
-x	(extract) Εξαγωγή από αρχείο
-f	(file) Ορισμός ονόματος αρχείου
-v	(verbose) Λεπτομερής έξοδος
-A	(concatenate) Συνένωση πολλών tarball
-z	(gzip) Συμπίεση με gzip
-j	(bzip2) Συμπίεση με bzip2
-J	(xz) Συμπίεση με xz
-W	(verify) Έλεγχος ακεραιότητας μετά την εγγραφή
-t	(list) Προβολή περιεχομένων
-u	(update) Προσθήκη μόνο των νέων αρχείων σε ήδη υπάρχων tarball
-C	(directory) Για εκτέλεση ενεργειών σε οριζόμενη τοποθεσία
-r	(append) Προσθήκη άλλων αρχείων σε ήδη υπάρχων tarball

3. Ανάγνωση αρχείων και ανακατεύθυνση εισόδου & εξόδου εντολών

Τα προγράμματα (εντολές), παράγουν ένα αποτέλεσμα (μια **έξοδος**). Η έξοδος μπορεί να είναι:

- Τα αποτελέσματα του προγράμματος, δηλαδή τα δεδομένα που έχει σχεδιαστεί να παράγει το πρόγραμμα (**standard output** ή **stdout**, **file descriptor: 1**)
- Μηνύματα κατάστασης και σφαλμάτων, που μας ενημερώνουν για την κατάσταση εκτέλεσης του προγράμματος (**standard error** ή **stderr**, **file descriptor: 2**)

Για παράδειγμα, το `ls -l /home/greeklug` παράγει:

```
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Desktop
drwxr-xr-x 8 greeklug greeklug 4096 Apr  3 20:00 Documents
drwxr-xr-x 4 greeklug greeklug 4096 Apr  3 19:54 Downloads
drwxrwxr-x 4 greeklug greeklug 4096 Apr  6 00:33 GreekLUG
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Music
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Pictures
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Public
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Templates
drwxr-xr-x 2 greeklug greeklug 4096 Apr  3 00:59 Videos
```

Το παραπάνω αποτέλεσμα (**stdout**), από προεπιλογή δεν αποθηκεύεται σε κάποιο αρχείο, αλλά προβάλλεται στην οθόνη.

Επίσης, οι εντολές λαμβάνουν είσοδο από έναν μηχανισμό που ονομάζεται **standard input** (**stdin- file descriptor: 0**), ο οποίος είναι συνδεδεμένος με το πληκτρολόγιο.

Με την **ανακατεύθυνση εισόδου/εξόδου (I/O redirection)** μπορούμε να αλλάξουμε πού πηγαίνει η έξοδος και από πού προέρχεται η είσοδος.

3α] Ανακατεύθυνση εξόδου (standard output), file descriptor: 1

Για να ανακατευθύνουμε την έξοδο σε κάποιο αρχείο αντί της οθόνης, χρησιμοποιούμε το σύμβολο `>`. Για λόγους απλοποίησης το γράφουμε ως `>`

Παράδειγμα:

i]

```
ls -l /home/greeklug > file-ls.txt
```

Το `file-ls.txt` περιέχει τώρα το αποτέλεσμα της εντολής `ls -l /home/greeklug`

Μπορούμε να το διαβάσουμε χρησιμοποιώντας είτε την `cat` (προβολή αρχείων με μικρό περιεχόμενο), είτε την `less` (χρησιμοποιείται σε αρχεία με μεγάλο περιεχόμενο), ή να προβάλλουμε μέρος των αρχείων (`head`, `tail`)

```
cat file-ls.txt (προβολή όλου του αρχείου)
```

```
head file-ls.txt (προβολή των πρώτων 10 γραμμών του αρχείου)
```

```
head -15 file-ls.txt (προβολή των πρώτων 15 γραμμών του αρχείου)
```

```
tail file-ls.txt (προβολή των τελευταίων 10 γραμμών του αρχείου)
```

```
tail -15 file-ls.txt (προβολή των τελευταίων 15 γραμμών του αρχείου)
```

```
less file-ls.txt (προβολή όλου του αρχείου σε pages)
```

ii]

Αν τώρα θέλουμε να κάνουμε ανακατεύθυνση το αποτέλεσμα της εντολής `ls -l`

`/home/member` πάλι στο ίδιο αρχείο `file-ls.txt` **που ήδη περιέχει** το output της εντολής

`ls -ls /home/greeklug` τότε θα δούμε ότι με την εντολή:

`ls -l /home/member > file-ls.txt` σβήνεται το υπάρχον περιεχόμενο και γράφεται μόνο το νέο.

iii]

Για να προσθέσουμε περιεχόμενο (append) στο `file-ls.txt` τότε θα πρέπει να χρησιμοποιήσουμε το σύμβολο `>>` για την ανακατεύθυνση:

```
ls -l /home/member >> file-ls.txt
```

3β] Ανακατεύθυνση σφαλμάτων (standard error), file descriptor: 2

Για να ανακατευθύνουμε το **standard error**, θα χρησιμοποιήσουμε τον **file descriptor** 2 , ώστε να αναγνωρίζεται από το shell.

Παράδειγμα:

Χρησιμοποιούμε την εντολή `ls -l /home/member` ώστε να προβάλλουμε κάτι στο οποίο δεν έχουμε πρόσβαση, ώστε να παραχθεί σφάλμα:

```
ls: cannot open directory '/home/member/': Permission denied
```

Θα χρησιμοποιήσουμε τον **file descriptor** 2 που συμβολίζεται ως `2>` για να στείλουμε το σφάλμα σε ένα αρχείο:

```
ls -l /home/member 2> file-ls-error.txt
```

3γ] Ταυτόχρονη ανακατεύθυνση εξόδου & σφαλμάτων

Μπορούμε να ανακατευθύνουμε εξίσου το **standard output** και **standard error** ταυτόχρονα, είτε σε διαφορετικά μέρη είτε σε ένα κοινό.

Παράδειγμα διαφορετικών ανακατευθύνσεων:

```
ls /home/member > file-ls.txt 2> file-ls-error.txt
```

Για την κοινή ανακατεύθυνση χρησιμοποιούμε το όρισμα `2>&1` στο τέλος της εντολής ώστε να συνδυάσουμε τις δύο εξόδους σε μία κοινή.

Πρακτικά αντιστοιχεί στην ανακατεύθυνση του **standard error** (2) στο ίδιο μέρος με το **standard output** (1).

Παράδειγμα κοινής ανακατεύθυνσης:

```
ls /home/member > file-output.txt 2>&1
```

3δ] Ανακατεύθυνση εισόδου (standard input), file descriptor: 0

Χρησιμοποιούμε το σύμβολο `<` για να ανακατευθύνουμε το **standard input**.

Παράδειγμα:

```
cat < file1.txt
```

Μπορούμε καλύτερα να καταλάβουμε το **standard input** μέσω ενός χαρακτηριστικού του shell που ονομάζεται **pipelines** (`|`)

Το `|` όταν χρησιμοποιείται ανάμεσα σε 2 εντολές, παίρνει το output της 1ης και το δίνει ως input στην 2η.

Παραδείγματα:

```
du -h /home/savvas/ | less
```

```
du -h /home/savvas/ | sort | less
```

```
ls /home/greeklug/Documents/ /home/greeklug/Downloads/ | sort | less
```

grep - print lines that match patterns

```
ls /home/greeklug/Documents/ /home/greeklug/Downloads/ | sort | grep ".txt"
```

4. Δικαιώματα αρχείων & καταλόγων

Σε ένα σύστημα Linux, μπορούν να συνδεόνται ταυτόχρονα πολλοί χρήστες. Η πρόσβασή τους στηρίζεται σε ένα σύστημα **δικαιωμάτων** με βάση το οποίο παράγονται διάφορα επίπεδα πρόσβασης στα δεδομένα του συστήματος.

Έτσι για να μην μπορεί κάποιος **απλός** χρήστης να δει τα αρχεία κάποιου άλλου χρήστη, κάθε αρχείο και κατάλογος συνοδεύεται από συγκεκριμένα δικαιώματα, που ορίζουν ποιος χρήστης έχει πρόσβαση σε αυτά και αποτρέπουν τους υπόλοιπους από το να τα χειριστούνε.

Παράδειγμα

```
cat /etc/sudoers
```

```
cat: /etc/sudoers: Permission denied
```

Ως απλοί χρήστες, δεν μπορούμε να προβάλλουμε το αρχείο `sudoers`

```
ls -lah /home/greeklug
```

```
-rw----- 1 greeklug greeklug 368 Apr  8 19:04 .bash_history
-rw-r--r-- 1 greeklug greeklug 220 Apr  3 00:43 .bash_logout
-rw-r--r-- 1 greeklug greeklug 3,7K Apr  3 00:43 .bashrc
drwx----- 11 greeklug greeklug 4,0K Apr  3 16:39 .cache
drwxr-xr-x 16 greeklug greeklug 4,0K Apr  3 18:48 .config
drwxr-xr-x  2 greeklug greeklug 4,0K Apr  3 00:59 Desktop
-rw-r--r-- 1 greeklug greeklug  27 Apr  3 00:59 .dmrc
drwxr-xr-x  8 greeklug greeklug 4,0K Apr  3 20:00 Documents
drwxr-xr-x  4 greeklug greeklug 4,0K Apr  3 19:54 Downloads
```

Πεδίο	Έννοια
<code>-rw-r--r--</code>	Δικαιώματα πρόσβασης στο αρχείο
<code>1</code>	Αριθμός hard link του αρχείου

Πεδίο	Έννοια
greeklug	Ιδιοκτήτης (χρήστης στον οποίο ανήκει το αρχείο)
greeklug	Ομάδα (όνομα του group στο οποίο ανήκει το αρχείο)
3,7K	Μέγεθος αρχείου
Apr 3 00:43	Ημερομηνία τροποποίησης
.bashrc	Όνομα αρχείου

- `-rw-r--r--` Όταν τα δικαιώματα ξεκινούν με `-` , σημαίνει ότι αυτό είναι αρχείο
- `drwxr-xr-x` Όταν τα δικαιώματα ξεκινούν με `d` , σημαίνει ότι αυτό είναι κατάλογος (directory)
- `lrwxrwxrwx` Όταν τα δικαιώματα ξεκινούν με `l` , σημαίνει ότι αυτό είναι συμβολικός σύνδεσμος (symlink)

Τα δικαιώματα ενός αρχείου ή καταλόγου αποτελούνται από τρεις τριάδες

`rwXrwxrwx`

1. Η 1η τριάδα αφορά τα δικαιώματα του **ιδιοκτήτη** (user) του αρχείου ή καταλόγου (συνήθως αυτός που δημιούργησε το αρχείο ή κατάλογο).
2. Η 2η τριάδα αφορά τα δικαιώματα της **ομάδας** (group) στο οποίο ανήκουν κάποιοι χρήστες του συστήματος.
3. Η 3η τριάδα αφορά τα δικαιώματα για **όλους τους υπόλοιπους** (others) (άλλους χρήστες που δεν είναι ιδιοκτήτες του αρχείου, ούτε ανήκουν στο group).

Δικαίωμα	Αρχεία	Κατάλογοι
<code>r</code>	Επιτρέπει την ανάγνωση του αρχείου	Επιτρέπει την προβολή των περιεχομένων του καταλόγου, αν είναι ενεργοποιημένο και το <code>x</code>
<code>w</code>	Επιτρέπει να γράψουμε μέσα στο αρχείο	Επιτρέπει την δημιουργία, μετονομασία και διαγραφή αρχείων μέσα στον κατάλογο, αν είναι ενεργοποιημένο και το <code>x</code>
<code>x</code>	Επιτρέπει την εκτέλεση του αρχείου (πχ πρόγραμμα)	Επιτρέπει την πρόσβαση μέσα σε έναν κατάλογο, εφόσον έχουμε πρόσβαση ως ιδιοκτήτης ή μέσω της ομάδας

Παραδείγματα:

A]

`-rw-r--r-- 1 greeklug greeklug 3,7K Apr 3 00:43 .bashrc`

Στο παραπάνω αρχείο `.bashrc` βλέπουμε:

1. `rw-` Ο χρήστης `greeklug` μπορεί να διαβάσει και να γράψει μέσα στο αρχείο
2. `r--` Όσοι ανήκουν στο group `greeklug` , μπορούν να διαβάσουν το αρχείο

3. `r--` Όλοι οι υπόλοιποι, μπορούν να διαβάσουν το αρχείο

B]

```
-rw-rw-r-- 1 greeklug greeklug 76 Apr 3 19:51 file35.txt
```

Στο παραπάνω αρχείο `file35.txt` βλέπουμε:

1. `rw-` Ο χρήστης `greeklug` μπορεί να διαβάσει και να γράψει μέσα στο αρχείο
2. `rw-` Όσοι ανήκουν στο group `greeklug`, μπορούν να διαβάσουν και να γράψουν μέσα στο αρχείο
3. `r--` Όλοι οι υπόλοιποι, μπορούν να διαβάσουν το αρχείο

Γ]

```
drwxr-x--- 17 greeklug greeklug 4,0K Apr 8 18:44 greeklug
```

Στον παραπάνω κατάλογο `greeklug` βλέπουμε:

1. `rwx` Ο χρήστης `greeklug` μπορεί να προβάλλει και να επεξεργαστεί τα αρχεία μέσα στον κατάλογο `Documents`
2. `r-x` Όσοι ανήκουν στο group `greeklug`, μπορούν μόνο να προβάλλουν τα αρχεία μέσα στον κατάλογο `greeklug`
3. `---` Όλοι οι υπόλοιποι, δεν μπορούν ούτε να προβάλλουν τα περιεχόμενα του καταλόγου (δεν μπορούν να κάνουν τίποτα)

5. Αλλαγή δικαιωμάτων αρχείων & καταλόγων (`chmod`)

Αν είμαστε διαχειριστές ή έχουμε δικαιώματα διαχειριστή ή είμαστε οι ιδιοκτήτες των αρχείων ή καταλόγων, μπορούμε να αλλάξουμε τα δικαιώματα τους, ώστε να δώσουμε πρόσβαση σε άλλους χρήστες του συστήματος.

Αυτό μπορεί να γίνει με 2 τρόπους:

1] Αριθμητικό σύστημα (πιο βολικό)

Δικαίωμα	Αριθμός
r	4
w	2
x	1

```
-rw-rw-r-- 1 greeklug greeklug 76 Apr 3 19:51 file35.txt
```

Στο παραπάνω αρχείο `file35.txt` τα δικαιώματα του είναι:

- 6 (4+2) για τον ιδιοκτήτη `greeklug`
- 6 (4+2) για το group `greeklug`

4 (4) για όλους τους άλλους

Συνεπώς, τα συνολικά δικαιώματα του αρχείου `file35.txt` είναι: 664

Αν θέλουμε να κάνουμε το αρχείο εκτελέσιμο για τον ιδιοκτήτη και για το group `greeklug`, δηλαδή να αλλάξουμε τα δικαιώματα ώστε να φαίνεται:

```
-rwxrwxr-- 1 greeklug greeklug 76 Apr 3 19:51 file35.txt
```

τότε θα πρέπει τα δικαιώματα του να είναι:

7 (4+2+1) για τον ιδιοκτήτη `greeklug`

7 (4+2+1) για το group `greeklug`

4 (4) για όλους τους άλλους

Συνεπώς, τα συνολικά νέα δικαιώματα του αρχείου `file35.txt` θα είναι: 774

Αυτό γίνεται χρησιμοποιώντας την εντολή `chmod`:

```
chmod 774 file35.txt
```

2] Συμβολικό σύστημα

Σύμβολο	Έννοια
u	user (ιδιοκτήτης αρχείου)
g	group
o	others
a	all

Παραδείγματα:

- `chmod u+x script.sh` (Προσθέτουμε δικαιώματα εκτέλεσης του αρχείου `script.sh`, στον ιδιοκτήτη του)
- `chmod g+rw file.txt` (Προσθέτουμε δικαιώματα ανάγνωσης και εγγραφής του αρχείου `file.txt`, στο group)
- `chmod o=rw file.txt` (Δίνουμε μόνο δικαιώματα ανάγνωσης και εγγραφής του αρχείου `file.txt`, στους άλλους)
- `chmod u-x file.txt` (Αφαιρούμε δικαιώματα εκτέλεσης του αρχείου `file.txt`, από τον ιδιοκτήτη του)
- `chmod a=rwx file.txt` (Δίνουμε πλήρη δικαιώματα του αρχείου `file.txt`, σε όλους)
- `chmod o=rx dir1` (Δίνουμε μόνο δικαιώματα προβολής του φακέλου `dir1` στους άλλους)

6. Αλλαγή σε άλλον χρήστη

6a] `su`

Μπορούμε να αλλάξουμε σε άλλον χρήστη με την εντολή:

```
su -l otheruser
```

πχ

```
su -l member
```

όπου το option `-l` σημαίνει login

Για την εκτέλεση της εντολής, το σύστημα μας ζητάει να πληκτρολογήσουμε **τον κωδικό του χρήστη που θέλουμε να συνδεθούμε** (member)

Έτσι, **αποσυνδεόμαστε από τον τρέχον χρήστη, ανοίγει νέο shell** και συνδεόμαστε ως νέος χρήστης (member) με το περιβάλλον του νέου χρήστη (**full login**)

Υπάρχει η δυνατότητα να αλλάξουμε με την εντολή `su member` όπου επίσης θα μας ζητήσει τον κωδικό του νέου χρήστη, ωστόσο **διατηρείται το περιβάλλον** του αρχικού μας χρήστη.

6β] root χρήστης

Σε όλες τις διανομές Linux υπάρχει ο... υπέρτατος διαγαλαξιακός... διαχειριστής που ονομάζεται `root`. Έχει πλήρη πρόσβαση σε όλο το σύστημα.

6β1] sudo

Σε αρκετές διανομές ο 1ος χρήστης ενός συστήματος έχει δικαιώματα διαχειριστή μέσω του ειδικού group `sudo` (sudoers).

Για να δούμε αν ανήκουμε στο group `sudo`:

```
id username
```

πχ

```
id greeklug
```

μας δίνει:

```
uid=1000(greeklug) gid=1000(greeklug) groups=1000(greeklug),27(sudo)
```

Βλέπουμε ότι ο χρήστης `greeklug` ανήκει στο group `sudo` και έχει user id: 1000

Σε διανομές που χρησιμοποιούν `sudo`, πχ όπως αυτές που στηρίζονται σε Ubuntu, μπορούμε βάζοντας το `sudo` μπροστά από την εντολή, να την εκτελέσουμε πληκτρολογώντας στη συνέχεια τον **κωδικό του user με τον οποίο είμαστε συνδεδεμένοι**.

Παράδειγμα:

```
sudo chmod u-x file.txt
```

Με την παραπάνω εντολή, **δεν αλλάζουμε χρήστη, ούτε shell**, απλά εκτελούμε την εντολή με τα δικαιώματα διαχειριστή που έχουμε.

Μια διαφορά του Linux με άλλα λειτουργικά συστήματα είναι ότι το σύστημα απαιτεί κάθε φορά από εμάς την εισαγωγή του κωδικού μας, για την εκτέλεση της εντολής (συνήθως δεν θα ζητηθεί για το επόμενο λεπτό, από τη στιγμή που θα εισαχθεί). Αυτό προσθέτει ένα επιπλέον επίπεδο ασφάλειας.

Γενικά χρησιμοποιούμε το `sudo` κάθε φορά που καλούμαστε να κάνουμε εργασίες σχετικά με την διαχείριση του συστήματος, πχ εγκατάσταση ενημερώσεων, εγκατάσταση/αφαίρεση προγραμμάτων, διαχείριση χρηστών, κτλ

Φυσικά, ένας χρήστης με δικαιώματα διαχειριστή έχει πρόσβαση και στα αρχεία όλων των υπόλοιπων χρηστών, διαχειριστών ή μη, αν δεν έχει γίνει κάποια επιπλέον ρύθμιση.

6β2] `su root & su -l root`

Σε διανομές που δεν χρησιμοποιούν `sudo`, πχ όπως στο Debian, μπορούμε να αλλάξουμε σε άλλον χρήστη με την εντολή:

```
su root ή την su -l root
```

Για την εκτέλεση της εντολής, το σύστημα μας ζητάει να πληκτρολογήσουμε **τον κωδικό του root**

Έτσι, **αποσυνδεόμαστε από χρήστη** `greeklug`, **ανοίγει νέο shell** και συνδεόμαστε ως χρήστης `root`

7. Αλλαγή ιδιοκτησίας αρχείων (`chown`)

Αν είμαστε διαχειριστές ή έχουμε δικαιώματα διαχειριστή, μπορούμε να αλλάξουμε την ιδιοκτησία ενός αρχείου ή καταλόγου.

Χρησιμοποιούμε την εντολή `chown` (ownership)

Αλλαγή μόνο ιδιοκτήτη:

Έστω ότι έχουμε τον παρακάτω κατάλογο:

```
drwxrwxr-x 2 member member 4096 Apr 8 20:17 dir1
```

και θέλουμε να αλλάξουμε την ιδιοκτησία του καταλόγου `dir1` από τον τρέχον χρήστη στον χρήστη `greeklug` πληκτρολογούμε:

```
sudo chown greeklug dir1
```

Ο φάκελος θα αλλάξει ως εξής:

```
drwxrwxr-x 2 greeklug member 4096 Apr 8 20:17 dir1
```

```
chown -R greeklug dir1/
```

Αλλάζουμε την ιδιοκτησία του `dir1` και όλων των αρχείων/υποφακέλων του, σε χρήστη `greeklug` (το `-R` ή `--recursive` εφαρμόζει την αλλαγή αναδρομικά)

Αλλαγή ιδιοκτήτη και `group`:

Με την εντολή:

```
sudo chown greeklug:greeklug dir1
```

αλλάζουμε την ιδιοκτησία του καταλόγου σε χρήστη `greeklug` και το `group` σε `greeklug`

poweroff ή shutdown -h (για κλείσιμο του συστήματος)
reboot (για επανεκκίνηση του συστήματος)

Το αρχείο διέπεται από την άδεια:

Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0
Διεθνές (CC BY-NC-SA 4.0)

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>

Ελληνική Ένωση Φίλων ΕΛ/ΛΑΚ (GreekLUG)

<https://www.greeklug.gr/>