

# Λειτουργικά Συστήματα Ι

## Ασκήσεις Πράξης

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών  
Σχολή Μηχανικών



---

Περισσότερα για Αρχεία

---

# Σύνδεσμοι

Ένας σύνδεσμος αποτελεί μια αναφορά σε ένα άλλο αρχείο του συστήματος. Η δημιουργία επιπλέον συνδέσμων γίνεται με την εντολή **ln** (link). Με τη δημιουργία συνδέσμων μπορούμε να αναφερόμαστε με διαφορετικά ονόματα σε κάποιο αρχείο.

Υπάρχουν δύο είδη συνδέσμων:

- Κανονικοί ή πραγματικοί σύνδεσμοι (hard links)

Καθώς το όνομα του αρχείου δεν είναι αποθηκευμένο στα μεταδεδομένα του κόμβου πληροφοριών αλλά βρίσκεται στο αρχείο καταλόγου στον οποίο ανήκει, μπορούμε να συνδέσουμε ένα inode με περισσότερα ονόματα.

# Σύνδεσμοι

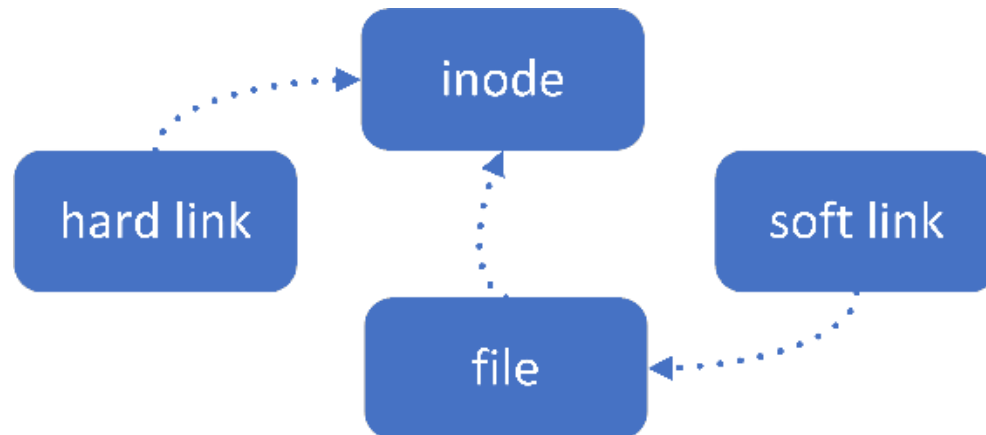
- Συμβολικοί σύνδεσμοι (soft links ή symlinks)

Πρόκειται για μικρά αρχεία κειμένου που είναι δείκτες (συντομεύσεις) σε άλλα αρχεία ή καταλόγους. Ένας συμβολικός σύνδεσμος ουσιαστικά περιέχει τη διαδρομή για το αρχείο (ή τον κατάλογο) στόχο. Καθώς δεν αναφέρονται σε κάποιο κόμβο πληροφοριών αλλά σε μονοπάτι μπορούν να βρίσκονται σε διαφορετικά συστήματα αρχείων. Ένας συμβολικός σύνδεσμος έχει δικό του αριθμό inode, αποτελεί δηλαδή ένα απολύτως ξεχωριστό αρχείο. Το δε μέγεθός του ισούται με το μήκος του μονοπατιού στο οποίο “δείχνει”.

# Σύνδεσμοι

Οι συμβολικοί σύνδεσμοι είναι το αντίστοιχο των εικονιδίων συντόμευσης των Windows. Αν διαγραφεί το αρχείο στο οποίο δείχνουν καθίστανται άχρηστοι.

Το σχήμα που ακολουθεί αναδεικνύει τη διαφορά μεταξύ πραγματικών και συμβολικών συνδέσμων:



# Σύνδεσμοι

Η βασική μορφή σύνταξης της εντολής **ln** είναι:

**ln** [OPTION]... TARGET LINKNAME

Η σημαντικότερη επιλογή της εντολής είναι η **-s** (εναλλακτικά **--symbolic**) με τη χρήση της οποίας δημιουργούμε έναν συμβολικό σύνδεσμο αντί του προκαθορισμένου πραγματικού συνδέσμου.

# Σύνδεσμοι

Παράδειγμα δημιουργίας νέου αρχείου στη συνέχεια ενός πραγματικού σύνδεσμου σ' αυτό με την εντολή ln:

```
$ touch testfile                # create new file
$ ls -il testfile              # show inode number
1048926 -rw-r--r-- 1 nemo staff 0 Nov 13 23:11
testfile
$ ln testfile myhardlink      # create hard link
$ ls -il testfile myhardlink
1048926 -rw-r--r-- 2 nemo staff 0 Nov 13 23:11
myhardlink
1048926 -rw-r--r-- 2 nemo staff 0 Nov 13 23:11
testfile
```

# Σύνδεσμοι

Η επόμενη εντολή δημιουργεί έναν συμβολικό σύνδεσμο σε έναν κατάλογο:

```
$ ln -s Music/mp3/RHCP RHCP # symbolic link
```

```
$ ls -l RHCP
```

```
lrwxrwxrwx 1 nemo staff 14 Nov 13 23:40 RHCP ->  
Music/mp3/RHCP
```

Ο πρώτος χαρακτήρας “l” της γραμμής εξόδου της εντολής ls συμβολίζει έναν σύνδεσμο (link) και στην τελευταία στήλη το βέλος υποδεικνύει πως ο σύνδεσμος RHCP δείχνει στον κατάλογο Music/mp3/RHCP. Το δε μέγεθος του συνδέσμου είναι 14, όσο και το μήκος του μονοπατιού.



# Προβολή Περιεχόμενου Αρχείων

Η εντολή **cat** (concatenate) προβάλλει τα περιεχόμενα αρχείων κειμένου στην οθόνη το ένα μετά το άλλο. Η σύνταξή της εντολής είναι:

**cat** [OPTION]... [FILE]...

Οι πιο σημαντικές επιλογές της είναι:

- n προσθέτει αρίθμηση γραμμών στην έξοδο
- s συμπύσσει πολλαπλές κενές γραμμές σε μία μόνο
- v εμφανίζει μη εκτυπώσιμους χαρακτήρες (σε σημειογραφία ^ και M)

# Προβολή Περιεχόμενου Αρχείων

```
$ cat /etc/timezone /etc/networks
```

```
Europe/Athens                # /etc/timezone
default      0.0.0.0         # /etc/networks
loopback     127.0.0.0
link-local   169.254.0.0
localnet     195.130.100.100
```

```
$ cat -n file.txt
```

```
1 Peter Frampton 188 1962 252241313
2 Anna Frampton 170 1963 252241313
3 Alice Bauhaus 167 1965 2106399547
4 Steffen Reichert 188 1963 2310620789
5 Wolfgang Hartmann 189 1958 2521091466
```

# Προβολή Περιεχόμενου Αρχείων

Στην περίπτωση που το αρχείο που θέλουμε να προβάλουμε έχει περισσότερο κείμενο από μία οθόνη, θα πρέπει να κάνουμε χρήση άλλων πιο εξιδεικευμένων εντολών.

Η εντολή **more** προβάλλει αρχεία κειμένου στην οθόνη, διακόπτοντας την προβολή σε κάθε μία οθόνη. Η μετακίνηση στην επόμενη οθόνη με τη χρήση του πλήκτρου [**Space**], στην προηγούμενη με το πλήκτρο [**b**] και η πρόωρη εγκατάλειψη της προβολής γίνεται με το πλήκτρο [**q**] όπως ακριβώς και στην εντολή **man**. Η δε εμφάνιση της βοήθειας γίνεται με τα πλήκτρα [**h**] ή [**?**].

Η σύνταξη της εντολής είναι:

**more** [OPTION]... FILE...

# Προβολή Περιεχόμενου Αρχείων

Η εντολή **head** τυπώνει στην οθόνη τις 10 πρώτες γραμμές ενός αρχείου κειμένου. Αν τα αρχεία είναι περισσότερα του ενός, τυπώνει πριν από τις γραμμές το αντίστοιχο όνομα αρχείου.

Η σύνταξή της έχει τη μορφή:

**head** [OPTION]... FILE...

Οι πιο σημαντικές επιλογές της είναι:

- n NUM** τυπώνει τις NUM πρώτες γραμμές αντί για τις προκαθορισμένες 10
- c NUM** τυπώνει τα NUM πρώτα bytes κάθε αρχείου

# Προβολή Περιεχομένου Αρχείων

Ακολουθούν μερικά παραδείγματα της εντολής `more`:

```
$ head -n 3 /etc/group          # first 3 lines
```

```
root:x:0:
```

```
daemon:x:1:
```

```
bin:x:2:
```

```
$ head -c 22 /etc/group        # first 22 bytes
```

```
root:x:0:
```

```
daemon:x:1:
```

Η εντολή **tail** έχει την αντίθετη λειτουργία από την `head`, τυπώνει δηλαδή στην οθόνη τις 10 τελευταίες γραμμές ενός αρχείου κειμένου. Με την επιλογή `-n` μπορούμε να τυπώσουμε διαφορετικό αριθμό γραμμών από το τέλος.

# Ανακατεύθυνση Εισόδου και Εξόδου

Κάθε πρόγραμμα που εκτελείται στη γραμμή εντολών συνδέεται αυτόματα με τρεις ροές δεδομένων για την επικοινωνία με το περιβάλλον του. Υπάρχουν τρεις προκαθορισμένες ροές: η προκαθορισμένη ροή εισόδου (standard input) που αντιστοιχεί στο πληκτρολόγιο, η προκαθορισμένη ροή εξόδου (standard output) που αντιστοιχεί στην οθόνη και η προκαθορισμένη ροή μηνυμάτων λάθους (standard error) που αντιστοιχεί επίσης στην οθόνη.

Οι ροές διαθέτουν επίσης αρίθμηση και συγκεκριμένα ο αριθμός 0 αντιστοιχεί στην προκαθορισμένη είσοδο, ο αριθμός 1 στην προκαθορισμένη έξοδο και ο αριθμός 2 στην προκαθορισμένη έξοδο λαθών.

# Ανακατεύθυνση Εισόδου και Εξόδου

Οι αριθμοί αυτοί ονομάζονται περιγραφείς αρχείων (file descriptors).



Κατά την εκτέλεση ενός προγράμματος, η είσοδος ή/και η έξοδος ενός προγράμματος μπορούν να ανακατευθυνθούν από/σε κάποιο αρχείο της επιλογής μας.

# Ανακατεύθυνση Εισόδου και Εξόδου

Η εντολή `cat` προβάλλει τα περιεχόμενα αρχείων κειμένου στην οθόνη. Αν δε δοθεί ως όρισμα κάποιο αρχείο διαβάζει από την προκαθορισμένη είσοδο (το πληκτρολόγιο δηλαδή) και γράφει στην προκαθορισμένη έξοδο (την οθόνη), αναπαράγει δηλαδή την είσοδό της στην έξοδο.

Μπορούμε να αλλάξουμε την προκαθορισμένη της αυτή λειτουργία, ανακατευθύνοντας την έξοδό της σ' ένα αρχείο. Έτσι αντί να εμφανίζεται το κείμενο στην οθόνη, γράφεται στο αρχείο. Η ανακατεύθυνση εξόδου γίνεται με τον τελεστή `>` ακολουθούμενου από ένα όνομα αρχείου.



# Ανακατεύθυνση Εισόδου και Εξόδου

Η εντολή που ακολουθεί δημιουργεί ένα αρχείο κειμένου:

```
$ cat > cicero.txt           # redirect to file
Omnium rerum principia parva sunt
Philosophia vero omnium mater artium
Non enim omnis error stultitia est dicenda
                                # [Ctrl]+[D] to exit
```

Σε περίπτωση που το αρχείο cicero.txt υπάρχει ήδη, η ανακατεύθυνση εξόδου θα διαγράψει πρώτα τα παλιά του δεδομένα. Αν θέλουμε να προσαρτήσουμε δεδομένα στο τέλος ενός υπάρχοντος αρχείου θα πρέπει να χρησιμοποιήσουμε τον τελεστή >>.

# Ανακατεύθυνση Εισόδου και Εξόδου

```
$ cat >> cicero.txt                # append to file
Quae est domestica sede iucundior? # [Ctrl]+[D]
$ cat cicero.txt
Omnium rerum principia parva sunt
Philosophia vero omnium mater artium
Non enim omnis error stultitia est dicenda
Quae est domestica sede iucundior?
```

Ο τελεστής >> ανακατευθύνει επίσης την έξοδο σε αρχείο, με τη διαφορά πως αν το αρχείο υπάρχει κάνει προσθήκη, αλλιώς το δημιουργεί.

# Ανακατεύθυνση Εισόδου και Εξόδου

Στις περιπτώσεις που δε θέλουμε τα διαγνωστικά μηνύματα να εμφανίζονται στην οθόνη, αλλά να οδηγούνται σε κάποιο αρχείο, ανακατευθύνουμε τα μηνύματα λάθους προτάσσοντας τον αριθμό 2 του τελεστή ανακατεύθυνσης (2>).

Αν θέλουμε να κάνουμε προσάρτηση των μηνυμάτων λάθους σε αρχείο θα χρησιμοποιήσουμε τη σημειογραφία 2>>.

```
$ cat no.file 2> errors # 2>> to append
$ cat errors
cat: no.file: No such file or directory
```

# Ανακατεύθυνση Εισόδου και Εξόδου

Για να ανακατευθύνουμε την κανονική έξοδο και την έξοδο λαθών στο ίδιο αρχείο, ανακατευθύνουμε την έξοδο λαθών στην έξοδο και την κανονική έξοδο σε αρχείο:

```
$ ls -l file.txt no.file > output 2>&1
```

```
$ cat output
```

```
ls: cannot access 'no.file': No such file or  
directory
```

```
-rw-r--r-- 1 nemo staff 176 Feb 10 23:04 file.txt
```

Το σύμβολο **&** αριστερά από τον αριθμό 1 προσδιορίζει την προκαθορισμένη έξοδο. Χωρίς αυτό η έξοδος λαθών θα ανακατευθυνόταν σε ένα αρχείο με το όνομα “1”.

# Ανακατεύθυνση Εισόδου και Εξόδου

Η εντολή **wc** (word count) τυπώνει το πλήθος των γραμμών, των λέξεων και των bytes ενός αρχείου κειμένου που τις δίνεται ως όρισμα. Ως γραμμές θεωρούνται τα αλφαριθμητικά που τερματίζουν με τον χαρακτήρα νέας γραμμής ενώ ως λέξεις τα αλφαριθμητικά που χωρίζονται με τον κενό χαρακτήρα. Η σύνταξη της εντολής είναι:

**wc** [OPTION]... FILE...

Οι πιο σημαντικές επιλογές της είναι:

- c, --bytes** τυπώνει το πλήθος των χαρακτήρων (bytes).
- l, --lines** τυπώνει το πλήθος των γραμμών
- w, --words** τυπώνει το πλήθος των λέξεων

# Ανακατεύθυνση Εισόδου και Εξόδου

Η προκαθορισμένη είσοδος της εντολής είναι το πληκτρολόγιο και η προκαθορισμένη της έξοδος η οθόνη.

Χωρίς αρχεία ως ορίσματα, η `wc` διαβάζει δεδομένα από το πληκτρολόγιο και με τον τερματισμό της εισόδου, εμφανίζει τα αποτελέσματα στην οθόνη:

`$ wc`

`No dark sarcasm in the classroom`

`Teachers leave them kids alone`

`# [Ctrl]+[D]`

`2            11            64`

# Ανακατεύθυνση Εισόδου και Εξόδου

Η ανακατεύθυνση εισόδου γίνεται με τον τελεστή `<` ακολουθούμενου από το επιθυμητό αρχείο:

```
$ wc < file.txt
```

```
5  25 176
```

Παρόμοιο αποτέλεσμα θα πάρουμε αν χρησιμοποιήσουμε το αρχείο ως όρισμα στην εντολή:

```
$ wc file.txt
```

```
5  25 176 file.txt
```

Με ανακατεύθυνση εισόδου δεν εμφανίζεται το όνομα του αρχείου καθώς αυτό δεν είναι γνωστό στην εντολή, η οποία διαβάζει δεδομένα από την προκαθορισμένη είσοδό της.

# Αναζήτηση Αρχείων

Η εντολή **find** χρησιμοποιείται για την ανεύρεση αρχείων με βάση ένα ή περισσότερα κριτήρια. Χαρακτηριστικά κριτήρια αναζήτησης της είναι: το όνομα του αρχείου, ο τύπος του, ο χρόνος προσπέλασης ή τροποποίησής του, το όνομα του ιδιοκτήτη του αρχείου, το μέγεθος του αρχείου κ.τ.λ. Όπως θα δούμε και στη συνέχεια, η εντολή πέρα από το να αναζητά αρχεία μπορεί και να εκτελέσει κάποια ενέργεια πάνω σε αυτά.

Η σύνταξή της εντολής **find** είναι:

**find** [OPTIONS] [starting-point...] [expression]



# Αναζήτηση Αρχείων

Οι επιλογές της εντολής είναι -P, -L και -H και έχουν σχέση με τη συμπεριφορά της όσον αφορά τους συμβολικούς συνδέσμους. Εξ ορισμού η find δεν ακολουθεί ποτέ συμβολικούς συνδέσμους. Ακολουθεί μία λίστα καταλόγων στους οποίους θα γίνει η αναζήτηση. Αν παραληφθεί το σημείο εκκίνησης της αναζήτησης, η αναζήτηση γίνεται στον τρέχοντα κατάλογο εργασίας και στους υποκαταλόγους του. Οι εκφράσεις αποτελούνται από τα κριτήρια αναζήτησης αλλά και από τυχόν ενέργειες (action) που θέλουμε να εκτελεστούν στα αρχεία που βρέθηκαν.

# Αναζήτηση Αρχείων

Ακολουθεί ο πίνακας με τις σημαντικότερες εκφράσεις:

Έκφραση	Περιγραφή
<b>-name pattern</b>	Η αναζήτηση γίνεται σύμφωνα με το όνομα.
<b>-type t</b>	Αναζήτηση σύμφωνα με τον τύπο του αρχείου. Η τιμή t μπορεί να είναι f για απλό αρχείο, d για κατάλογο, b για ειδικό αρχείο τύπου block κ.ο.κ.
<b>-user uname</b>	Αναζήτηση για αρχεία ιδιοκτησίας του χρήστη uname
<b>-links n</b>	Αναζήτηση με βάση τον αριθμό των συνδέσμων (hard links).
<b>-size n[cwbkMG]</b>	Η αναζήτηση γίνεται σύμφωνα με το μέγεθος του αρχείου. Ο αριθμός n μπορεί να ακολουθείται από τη μονάδα μεγέθους. Η προκαθορισμένη μονάδα είναι b (block των 512K). Το c αντιστοιχεί σε bytes, το k σε Kibibytes (KiB, 1024 bytes), το M για Mebibytes (MiB, 1024x2014 = 1048576 bytes) κ.ο.κ.

# Αναζήτηση Αρχείων

<b>-newer file</b>	Αναζήτηση για αρχεία που έχουν τροποποιηθεί πιο πρόσφατα από το αρχείο file.
<b>-atime n</b> <b>-amin n</b>	Η αναζήτηση γίνεται σύμφωνα με τον χρόνο της τελευταίας προσπέλασης του αρχείου. Η τιμή του n αντιστοιχεί σε ημέρες για την atime και λεπτά amin.
<b>-mtime n</b> <b>-mmin n</b>	Η αναζήτηση γίνεται σύμφωνα με τον χρόνο της τελευταίας τροποποίησης του αρχείου. Η τιμή του n ορίζεται σε ημέρες και λεπτά αντίστοιχα.
<b>-empty</b>	Το αρχείο (απλό ή κατάλογος) είναι άδειο.
<b>-perm mode</b>	Αναζήτηση σύμφωνα με τις άδειες πρόσβασης του αρχείου.

# Αναζήτηση Αρχείων

Σε όλες τις εκφράσεις που δέχονται ένα αριθμητικό όρισμα  $n$ , αυτό μπορεί να προσδιοριστεί ως  $+n$  για τιμές μεγαλύτερες του  $n$ ,  $-n$  για τιμές μικρότερες του  $n$  και  $n$  για τιμές ίσες με  $n$ .

Επίσης σε όλες τις εκφράσεις μπορούν να εφαρμοστούν οι λογικοί τελεστές AND, OR και NOT. Οι παράθεση δύο ή περισσότερων εκφράσεων χωρισμένες με κενό αντιπροσωπεύει μια συζευκτική πρόταση. Εναλλακτικά για το λογικό AND μπορούμε να χρησιμοποιήσουμε τον τελεστή **-a**. Ο τελεστής **-o** αντιπροσωπεύει το λογικό OR και το σύμβολο **!** αριστερά μιας έκφρασης αντιπροσωπεύει το NOT.

# Αναζήτηση Αρχείων

Οι σημαντικότερες επιλογές δράσης (action) είναι:

- delete** Διαγράφει τα αρχεία που βρέθηκαν.
- exec cmd;** Εκτελεί την εντολή `cmd` για κάθε αρχείο που εντοπίζει. Οι χαρακτήρες “{}” μετά την εντολή `cmd` αντικαθίστανται από το όνομα του αρχείου που βρέθηκε.
- ls** Τυπώνει στην οθόνη πληροφορίες για κάθε αρχείο που εντοπίζει στη μορφή της `ls -dils`.
- ok cmd;** Όπως η `-exec`, με τη διαφορά πως ζητείται επιβεβαίωση πριν από την εκτέλεση της `cmd`.

# Αναζήτηση Αρχείων

**-print** Τυπώνει στην οθόνη το όνομα του αρχείου που εντοπίζει. Αποτελεί την προκαθορισμένη επιλογή για την έκδοση GNU.

**-printf format** Τυπώνει στην οθόνη πληροφορίες για τα αρχεία που εντοπίζει ανάλογα με την ακολουθία διαφυγής και τον κωδικό μορφοποίησης, Τα σημαντικότερα από αυτά είναι:

**\n** Αλλαγή γραμμής

**\t** Οριζόντιος στηλοθέτης

**%a** Χρονοσφραγίδα τελευταίας προσπάθειας στη μορφή `ctime()` της C

# Αναζήτηση Αρχείων

<b>%c</b>	Χρονοσφραγίδα τελευταίας αλλαγής
<b>%f</b>	Το όνομα του αρχείου χωρίς το μονοπάτι
<b>%F</b>	Το σύστημα αρχείων στο οποίο βρίσκεται το αρχείο
<b>%g</b>	Το όνομα της ομάδας στην οποία ανήκει το αρχείο
<b>%i</b>	Ο αριθμός κόμβου πληροφοριών
<b>%n</b>	Το πλήθος των συνδέσμων του αρχείου
<b>%s</b>	Το μέγεθος του αρχείου σε byte
<b>%u</b>	Το όνομα του ιδιοκτήτη του αρχείου
<b>%U</b>	Ο αριθμός ταυτότητας του ιδιοκτήτη του αρχείου
<b>%y</b>	Ο τύπος του αρχείου

# Αναζήτηση Αρχείων

```
$ find . -name '*.txt' -print      # -name \*.txt
./data.txt
./file.txt
./OS.Book/awk/file.txt
$ find /dev -type c -print      # character files
/dev/vcsa6
...
/dev/rfkill
$ find /dev -type c -exec ls -l {} \; # escape ;
crw-rw---- 1 root tty 7, 134 Nov 2 19:55
/dev/vcsa6
...
```



# Αναζήτηση Αρχείων

Η επόμενη εντολή `find` εντοπίζει τα αρχεία του καταλόγου `/usr/src` που έχουν επέκταση `.c` και έχουν μέγεθος μεγαλύτερο από 100K:

```
$ find /usr/src -name '*.c' -size +100k -print  
/usr/src/linux-headers-6.0/scripts/lexer.lex.c  
/usr/src/linux-headers-5.1/scripts/lexer.lex.c
```

Η παράθεση δύο κριτηρίων χωρισμένα με κενό αντιπροσωπεύει μια συζευκτική πρόταση. Αν χρησιμοποιήσουμε τον τελεστή `-o` που αντιπροσωπεύει το λογικό OR, το πλήθος των αρχείων που πληρούν το ένα ή το άλλο κριτήριο θα είναι προφανώς μεγαλύτερο:

# Αναζήτηση Αρχείων

```
$ find /usr/src -name '*.c' -o -size +300k -print  
/usr/src/linux-headers-5.19/include/mac80211.h
```

...

```
/usr/src/Python-3.11.4/Objects/unicodeobject.c
```

Η επιλογή **-maxdepth levels** καθορίζει το μέγιστο βάθος καταλόγων στο οποίο θα γίνει αναζήτηση. Η εντολή `find` που ακολουθεί αναζητεί ειδικά αρχεία τύπου `block` μόνο στον κατάλογο εκκίνησης της αναζήτησης:

```
$ find /dev -maxdepth 1 -type b -print  
/dev/loop13
```

...

```
/dev/loop0
```

# Αναζήτηση Αρχείων

Μία εναλλακτική εντολή αναζήτησης είναι η **locate**, η οποία είναι γρήγορη και απλή στη σύνταξη. Η locate είναι πιο γρήγορη από την find, καθώς δεν ψάχνει στο σύστημα αρχείων αλλά σε μια βάση δεδομένων που ενημερώνεται σε τακτικά χρονικά διαστήματα.

Η σύνταξη της locate έχει ως εξής:

**locate** [OPTION]... PATTERN...

Οι πιο σημαντικές επιλογές της είναι:

**-c, --count**      εύρεση του πλήθους των αρχείων που  
                         ταιριάζουν

**-i, --ignore-case**      αγνόηση διάκρισης πεζών κεφαλαίων

# Αναζήτηση Αρχείων

```
$ locate hello      # search for filename *hello*  
/boot/grub/i386-pc/hello.mod
```

```
...
```

```
/usr/src/Python-3.7.0/Lib/__phello__.foo.py
```

```
$ locate -c hello      # count
```

```
54
```

```
$ locate -ic hello      # ignore case
```

```
56
```

Έκτακτη ενημέρωση της βάσης δεδομένων γίνεται με την εντολή updatedb:

```
$ sudo updatedb
```

# Αναζήτηση Αρχείων

Τέλος, η εντολή **whereis** εντοπίζει το εκτελέσιμο αρχείο, τον πηγαίο κώδικα και τη σελίδα εγχειριδίου για μια εντολή.

Η σύνταξη της **whereis** έχει ως εξής:

**whereis** [OPTION] [-BMS directory... -f] name...

```
$ whereis find
```

```
find: usr/bin/find /usr/share/man/man1/find.1.gz
```

```
$ whereis -b find      # where is binary of find
```

```
find: /usr/bin/find
```

```
$ whereis -m locate    # where is manual of locate
```

```
locate: /usr/share/man/man1/locate.1.gz
```

# Σύγκριση Αρχείων

Η πιο απλή εντολή σύγκρισης είναι η **cmp** (compare), η οποία συγκρίνει δύο αρχεία byte προς byte και εμφανίζει το πρώτο byte και τη γραμμή στα οποία διαφέρουν. Αν είναι ίδια δεν εμφανίζει κανένα μήνυμα.

Η σύνταξη της εντολής είναι:

**cmp** [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]

Η σημαντικότερη επιλογή της εντολής cmp είναι η **-i SKIP1[:SKIP2]** (εναλλακτικά **--ignore-initial=SKIP1[:SKIP2]**) που επιτρέπει τη σύγκριση δύο αρχείων από κάποιο byte διαφορετικό του πρώτου.

# Σύγκριση Αρχείων

Η εντολή **diff** συγκρίνει αρχεία και καταλόγους και μπορεί επιπλέον να δείξει και τις διαφορές τους. Η έξοδός της αποτελείται από οδηγίες για τον τρόπο αλλαγής του πρώτου αρχείου ώστε να γίνει ίδιο με το δεύτερο.

Η σύνταξη της εντολής είναι:

**diff** [OPTION]... FILES

Οι πιο σημαντικές επιλογές της είναι:

- i, --ignore-case** αγνόηση διάκρισης πεζών κεφαλαίων
- y, --side-by-side** έξοδος σε δύο στήλες
- B, --ignore-blank-lines** αγνόηση κενών γραμμών

# Σύγκριση Αρχείων

Η εντολή **comm** συγκρίνει δυο ταξινομημένα αρχεία γραμμή-γραμμή. Η σύνταξή της είναι:

**comm** [OPTION]... FILE1 FILE2

Χωρίς επιλογές, παράγει ως έξοδο τρεις στήλες: η πρώτη στήλη περιέχει τις γραμμές που είναι μοναδικές στο πρώτο αρχείο, η δεύτερη στήλη τις μοναδικές γραμμές του δεύτερου αρχείου και η τρίτη στήλη περιέχει τις κοινές γραμμές των δύο αρχείων. Οι σημαντικότερες επιλογές της εντολής είναι οι **-1**, **-2** και **-3** που καταστέλλουν αντίστοιχα τις γραμμές της 1ης, 2ης και 3ης στήλες της εξόδου.



# Ιδιοκτησία και Άδειες Πρόσβασης

Η αλλαγή των αδειών πρόσβασης (ή αλλιώς των δικαιωμάτων) ενός αρχείου γίνεται με την εντολή **chmod** (change mode), που έχει σύνταξη:

**chmod** [OPTION]... MODE FILE...

όπου MODE είναι οι άδειες πρόσβασης, οι οποίες μπορούν να τεθούν με δύο διαφορετικούς τρόπους. Η πρώτη μορφή αποτελείται από σύμβολα όπως **r**, **w** και **x** και η δεύτερη μορφή αποτελείται από έναν τετραψήφιο οκταδικό αριθμό. Οι σημαντικότερη επιλογή είναι το **-R** (recursively) για να αλλάξουν οι άδειες πρόσβασης σε αρχεία και υποκαταλόγους αναδρομικά σε ολόκληρη την ιεραρχία του ορίσματος.

# Ιδιοκτησία και Άδειες Πρόσβασης

Η συμβολική μορφή έχει σύνταξη:

**[u<sub>g</sub>oa...][[+|=][perms...]...]**

όπου το σύμβολο **u** αντιστοιχεί στον ιδιοκτήτη του αρχείου (user), το **g** αντιστοιχεί στην ομάδα του αρχείου (group), το **o** αντιστοιχεί στους υπόλοιπους χρήστες (others) και το **a** αντιστοιχεί σ' όλους τους χρήστες (all). Το σύμβολο **+** προσθέτει τις άδειες πρόσβασης που ακολουθούν (perms), το σύμβολο **-** αφαιρεί τις άδειες αυτές, ενώ το **=** θέτει επακριβώς τις συγκεκριμένες άδειες. Όσον αφορά τις άδειες πρόσβασης είναι (κανένα ή) κάποια από τα σύμβολα: **r**, **w**, **x** και τα **t**, **s**.

# Ιδιοκτησία και Άδειες Πρόσβασης

Ακολουθούν μερικά παραδείγματα της εντολής `chmod`:

```
$ ls -l file.txt          # show current permissions
-rw-r--r-- 1 nemo staff 176 Feb 28 20:49 file.txt
$ chmod u+x file.txt      # add execute to user
$ ls -l file.txt
-rwxr--r-- 1 nemo staff 176 Feb 28 20:49 file.txt
$ chmod a=rw,u-x file.txt  # set r,w to all,
                           # remove x from user
$ ls -l file.txt
-rw-rw-rw- 1 nemo staff 176 Feb 28 20:49 file.txt
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Το σύμβολο **t** είναι γνωστό ως **sticky bit** και εφαρμόζεται σε κοινόχρηστους καταλόγους. Στους καταλόγους στους οποίους έχει εφαρμοστεί το bit αυτό, οι χρήστες μπορούν να διαγράψουν μόνο τα δικά τους αρχεία. Οι επόμενες εντολές δημιουργούν έναν κατάλογο και ενεργοποιούν το sticky bit σ' αυτόν:

```
$ mkdir mytmp # make new directory
$ chmod +t mytmp # set sticky bit on
$ ls -l mytmp # show permissions
drwxr-xr-t 2 nemo staff 4096 Mar  3 20:47 mytmp
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Το σύμβολο **s** αναφέρεται στις άδειες πρόσβασης **suid** (set user ID) και **sgid** (set group ID). Όταν είναι ενεργοποιημένο το suid σε ένα εκτελέσιμο αρχείο, μπορεί να το εκτελέσει οποιοσδήποτε με τα δικαιώματα του ιδιοκτήτη του αρχείου. Για παράδειγμα, η εντολή `passwd` που αλλάζει το συνθηματικό ενός χρήστη είναι ιδιοκτησίας του χρήστη `root` και έχει ενεργό το suid:

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59680 May 17 2017
/usr/bin/passwd
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Την εντολή αυτή πρέπει να έχουν δικαίωμα να την εκτελούν όλοι οι χρήστες, ώστε να μπορούν να αλλάζουν το συνθηματικό τους. Όμως η εκτέλεση της εντολής επιφέρει αλλαγές στα αρχεία **/etc/passwd** και **/etc/shadow** τα οποία επιτρέπεται να τροποποιήσει μόνο ο υπερχρήστης. Η λύση δίνεται με την ενεργοποίηση του `suid` για το αρχείο `/usr/bin/passwd` που καθορίζει πως η διεργασία θα λειτουργήσει με το `uid` του ιδιοκτήτη του αρχείου.

Η ενεργοποίησή του `sgid` σε έναν κατάλογο έχει ως αποτέλεσμα όλα τα αρχεία που θα δημιουργηθούν μέσα στον κατάλογο να ανήκουν στην ομάδα στην οποία αυτός ανήκει.

# Ιδιοκτησία και Άδειες Πρόσβασης

Η αριθμητική μορφή της `chmod` αποτελείται από τέσσερα οκταδικά ψηφία με τιμές από 0 έως 7. Το πρώτο από τα τέσσερα ψηφία είναι προαιρετικό και αναφέρεται στα `suid`, `sgid` και `sticky bit`.

Τα τρία δεξιά ψηφία προσδιορίζουν τις άδειες πρόσβασης του ιδιοκτήτη του αρχείου, της ομάδας και των υπόλοιπων χρηστών αντίστοιχα. Η τιμή καθενός από τα τρία αυτά ψηφία υπολογίζεται ως άθροισμα των επιμέρους αδειών, όπου η τιμή 4 αντιστοιχεί στην ανάγνωση, η τιμή 2 στην εγγραφή και η τιμή 1 στην εκτέλεση. Η τιμή 0 προφανώς αντιστοιχεί στην απουσία κάθε άδειας.

# Ιδιοκτησία και Άδειες Πρόσβασης

Οκταδική	Άδεια πρόσβασης	Δυαδικό	Συμβολική
7	Πλήρης (4+2+1)	111	rwX
6	Ανάγνωση και Εγγραφή (4+2)	110	rw-
5	Ανάγνωση και Εκτέλεση (4+1)	101	r-X
4	Ανάγνωση	100	r--
3	Εγγραφή και Εκτέλεση (2+1)	011	-wX
2	Εγγραφή	010	-w-
1	Εκτέλεση	001	--X
0	Καμία άδεια	000	---



# Ιδιοκτησία και Άδειες Πρόσβασης

```
$ ls -l file.txt                # current permissions
-rw-rw---- 1 nemo staff 176 Feb 28 20:49 file.txt
$ chmod 0666 file.txt           # a=rw
$ ls -l file.txt
-rw-rw-rw- 1 nemo staff 176 Feb 28 20:49 file.txt
$ chmod 0740 file.txt           # u=rwx, g=r
$ ls -l file.txt
-rwxr----- 1 nemo staff 176 Feb 28 20:49 file.txt
$ chmod 4740 file.txt           # u+s
$ ls -l file.txt
-rwsr----- 1 nemo staff 176 Feb 28 20:49 file.txt
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Η `umask` προσδιορίζει τα δικαιώματα που θέλουμε να αφαιρεθούν από τα προκαθορισμένα δικαιώματα (0666 για τα αρχεία και 0777 για τους καταλόγους). Χωρίς ορίσματα η εντολή εμφανίζει την τρέχουσα τιμή της μάσκας.

Η σύνταξή της είναι:

**`umask [-S] [mode]`**

Η επιλογή **`-S`** εμφανίζει τη μάσκα στη συμβολική μορφή:

```
$ umask                # current user mask
0022
```

```
$ umask -S            # user mask in symbolic mode
```

```
u=rwx,g=rx,o=rx
```

# Ιδιοκτησία και Άδειες Πρόσβασης

```
$ umask 0007                # set user mask to 0007
$ umask -S                  # in symbolic mode
u=rwx,g=rwx,o=
$ touch myfile              # create new file
$ ls -l myfile              # show permissions
-rw-rw---- 1 nemo staff    0 Mar  5 15:14 myfile
$ umask 0133
$ mkdir mydir                # create new directory
$ ls -ld mydir
drw-r--r-- 2 nemo staff 4096 Mar  5 15:22 mydir
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Ο προκαθορισμένος ιδιοκτήτης ενός αρχείου είναι ο χρήστης που το δημιούργησε. Η εντολή **chown** (change owner) αλλάζει τον ιδιοκτήτη ενός αρχείου και μπορεί να εκτελεστεί μόνο από τον χρήστη root. Η σύνταξή της είναι:

**chown** [OPTION]... [OWNER][:[GROUP]] FILE...

Στη θέση του νέου ιδιοκτήτη μπορεί να τεθεί ο αριθμός χρήστη (uid). Η εντολή chown μπορεί επίσης να χρησιμοποιηθεί και για την αλλαγή της ομάδας του αρχείου. Η σημαντικότερη επιλογή της εντολής είναι η **-R** με την οποία μπορούμε να αλλάξουμε αναδρομικά ιδιοκτήτη σε καταλόγους και τα περιεχόμενά τους.

# Ιδιοκτησία και Άδειες Πρόσβασης

```
$ chown alice myfile          # sudo ?  
chown: changing ownership of 'myfile': Operation  
not permitted  
$ sudo chown alice myfile     # set alice as owner  
$ ls -l myfile  
-rw-rw---- 1 alice staff    0 Mar  6 14:01 myfile  
$ chown -R bob:students mydir  
$ ls -l  
drwxrwx--- 2 bob students 4096 Mar  6 14:02 mydir  
Νέος ιδιοκτήτης του abc γίνεται ο ιδιοκτήτης του αρχείου xyz:  
$ sudo chown --reference=/home/alice/xyz abc
```

# Ιδιοκτησία και Άδειες Πρόσβασης

Η εντολή **chgrp** (change group) αλλάζει την ομάδα ενός αρχείου. Ένας απλός χρήστης μπορεί να καταχωρίσει κάποιο αρχείο μόνο στις ομάδες στις οποίες είναι μέλος, κάτι που προφανώς δεν ισχύει για τον υπερχρήστη.

Η σύνταξη της εντολής είναι:

**chgrp** [OPTION]... GROUP FILE...

Στη θέση της νέας ομάδας μπορεί να τεθεί ο αριθμός ομάδας (gid).

Η σημαντικότερη επιλογή της εντολής είναι η **-R** με την οποία μπορούμε να αλλάξουμε αναδρομικά ομάδα σε καταλόγους και τα περιεχόμενά τους.

# Ιδιοκτησία και Άδειες Πρόσβασης

Ακολουθούν δύο παραδείγματα της `chgrp` που επιδεικνύουν τη χρήση της:

```
$ sudo chgrp students file
```

```
$ ls -l file
```

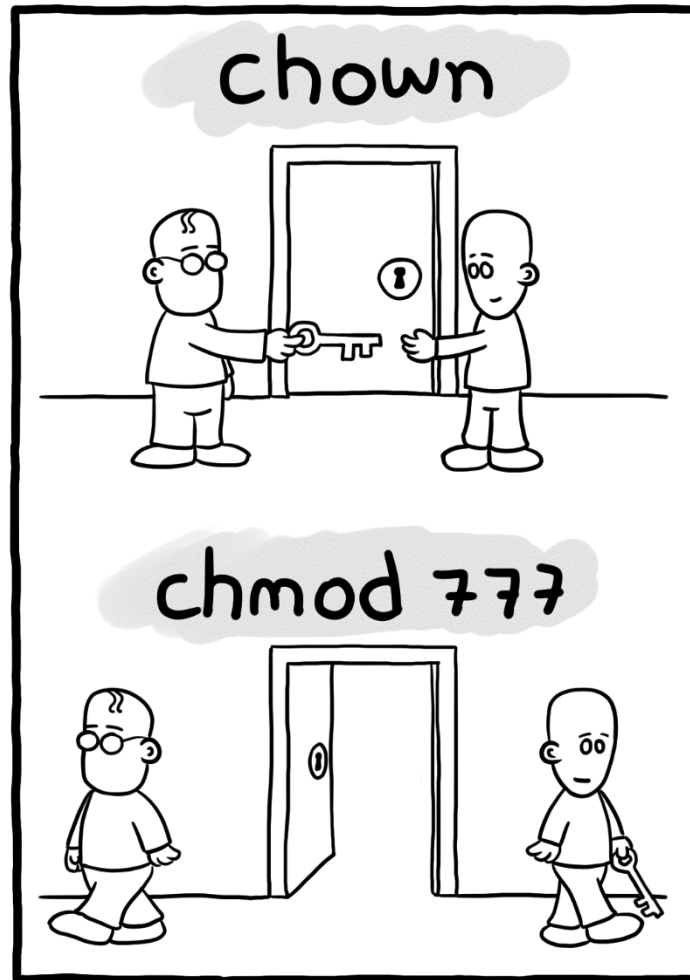
```
-rw-rw---- 1 alice students 0 Mar  6 14:01 file
```

```
$ sudo chgrp -R staff mydir
```

```
$ ls -ld mydir
```

```
drwxrwx--- 2 bob staff 4096 Mar  6 14:02 mydir
```

# Ερωτήσεις



Daniel Stori {turnoff.us}