

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής  
Εισαγωγή στην Επιστήμη των Υπολογιστών  
2023-24

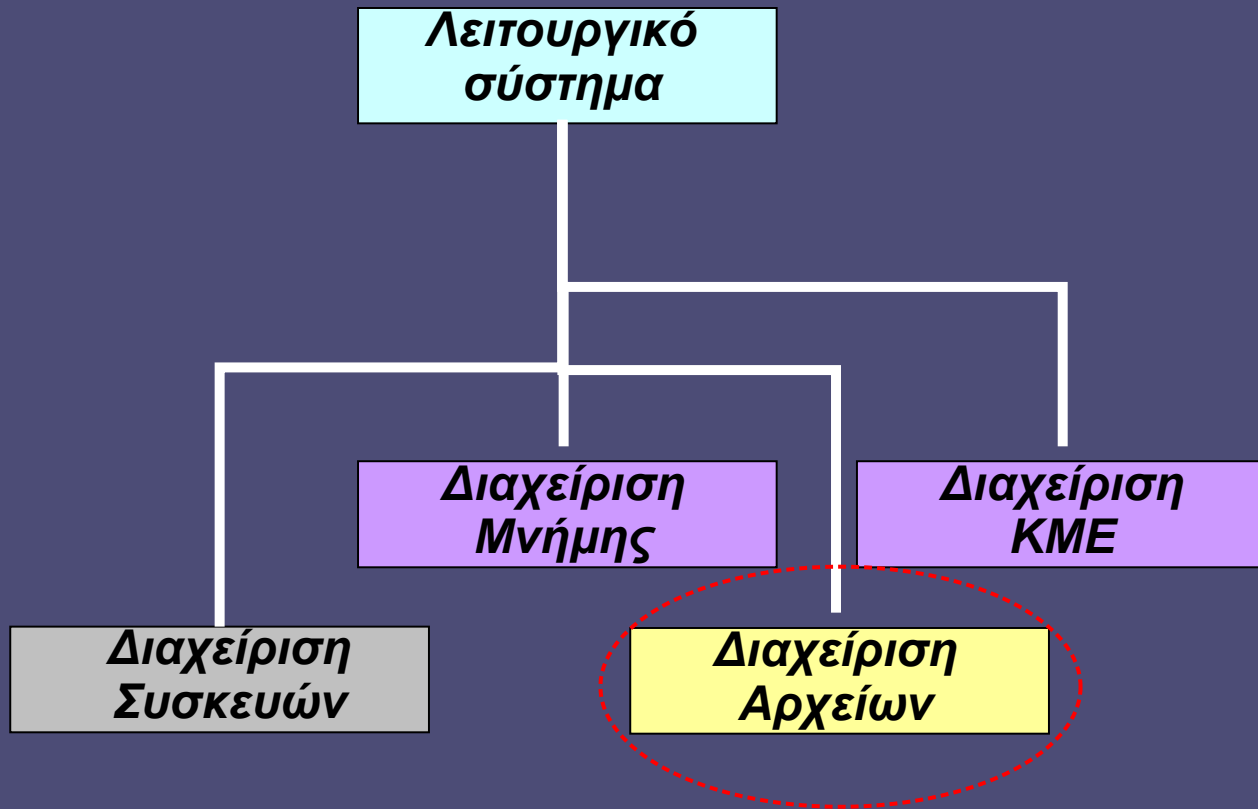
## Λειτουργικά Συστήματα (II) (διαχείριση αρχείων)

<http://mixstef.github.io/courses/csintro/>

Μ.Στεφανιδάκης



# Λειτουργικό Σύστημα: διαχείριση πόρων



# Διαχείριση αρχείων

- Διαχείριση αρχείων από το Λειτουργικό Σύστημα
  - Επίβλεψη αποθήκευσης
    - Αποθήκευση δομών **συστήματος αρχείων** στους δίσκους
  - Λειτουργίες στα αρχεία
    - Δημιουργία
    - Προσπέλαση (ανάγνωση/εγγραφή δεδομένων)
    - Τροποποίηση
    - Διαγραφή
  - Έλεγχος προσπέλασης από εφαρμογές
    - Διαμοιραζόμενη προσπέλαση
    - Αποκλειστική προσπέλαση

# Τι είναι ένα «αρχείο»;

- Μια αφαιρετική δομή (abstraction)
  - Για την αποθήκευση δεδομένων στους δίσκους του συστήματος
    - Προσφέρεται από το λειτουργικό σύστημα
- Αρχεία δεδομένων
  - Ως ακολουθία (stream) από bytes
    - “binary mode” (δυναμικά αρχεία)
  - Ως διαδοχικές γραμμές κειμένου
    - “text mode” (αρχεία [απλού] κειμένου)
- Ειδικά «αρχεία»
  - Συσκευές E/E, κατάλογοι, πληροφορία συστήματος...
    - Ανάλογα με το λειτουργικό σύστημα

# Δυαδικά αρχεία

- Περιέχουν **οποιαδήποτε** ακολουθία bytes
  - Ανάγνωση (read)  $n$  bytes

**read(4) : 2D, 98, 44, 59**

..0F 35 2D 98 44 59 FD 77 47 23 89 12 ..



τρέχουσα θέση (πριν  
την ανάγνωση)



τρέχουσα θέση (μετά  
την ανάγνωση)

- Εγγραφή (write)  $n$  bytes

**write( AA, 91, 11, 15)**

..0F 35 **AA 91 11 15** FD 77 47 23 89 12 ..



τρέχουσα θέση (πριν  
την εγγραφή)



τρέχουσα θέση (μετά  
την εγγραφή)

# Αρχεία Κειμένου

- Διαδοχικές γραμμές από χαρακτήρες
  - Σε κάποια γνωστή κωδικοποίηση
    - ASCII, Unicode (UTF-8) κλπ
  - Τερματισμός γραμμής με '\n' (newline)
    - Windows: 0D 0A – Unix: 0A
- Λειτουργίες
  - Ανάγνωση επόμενης γραμμής
  - Εγγραφή νέας γραμμής
    - στο τέλος του αρχείου (λειτουργία “append”)
- Προσοχή!
  - Μπορούμε να χειριστούμε ένα αρχείο κειμένου ως δυαδικό αρχείο
    - Όχι όμως και το αντίστροφο

# «Ανοίγοντας» ένα αρχείο

- **Άνοιγμα αρχείου**
  - `open(filename,mode)`
    - **filename**: το όνομα του αρχείου (και το μονοπάτι αν χρειάζεται)
    - **mode**: το είδος της ζητούμενης επεξεργασίας
      - read (ανάγνωση μόνο), write (εγγραφή μόνο, τυχόν παλιό αρχείο διαγράφεται), read-write (ανάγνωση-εγγραφή), append (προσθήκη δεδομένων στο τέλος)...
      - binary ή text mode
  - Προετοιμασία δομών ΛΣ για επεξεργασία του αρχείου
  - Υπάρχει και το αντίστοιχο **close()**

# Εκτελέσιμα αρχεία

- **Αρχεία με τον κώδικα μηχανής μιας εφαρμογής**
  - Παράγονται μέσω μιας «αλυσίδας» εργαλείων (toolchain)
    - Ξεκινώντας από το αρχείο πηγαίου κώδικα σε κάποια γλώσσα υψηλού επιπέδου (πρόγραμμα χρήστη)
    - Και καταλήγοντας στο τελικό εκτελέσιμο αρχείο που θα φορτωθεί στη μνήμη από το λειτουργικό σύστημα πριν την εκτέλεση
  - Η μορφή των εκτελέσιμων αρχείων εξαρτάται από την αρχιτεκτονική του επεξεργαστή και το είδος του λειτουργικού συστήματος



# Δημιουργία εκτελέσιμων αρχείων

Αρχείο πηγαίου κώδικα  
(source code file)



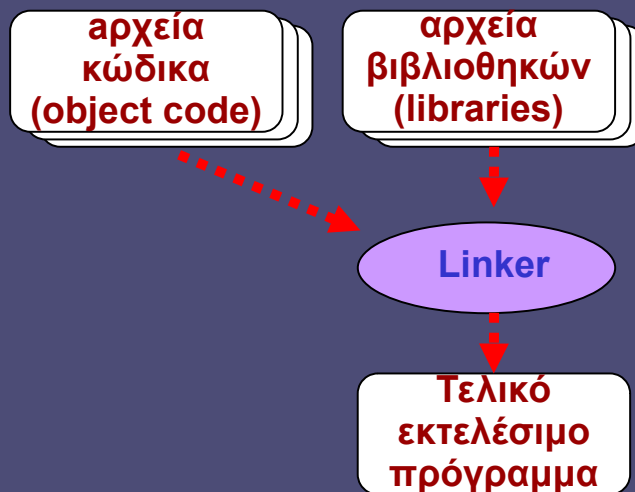
**Μεταγλωττιστής**



Αρχείο εκτελέσιμου κώδικα  
(object code file)

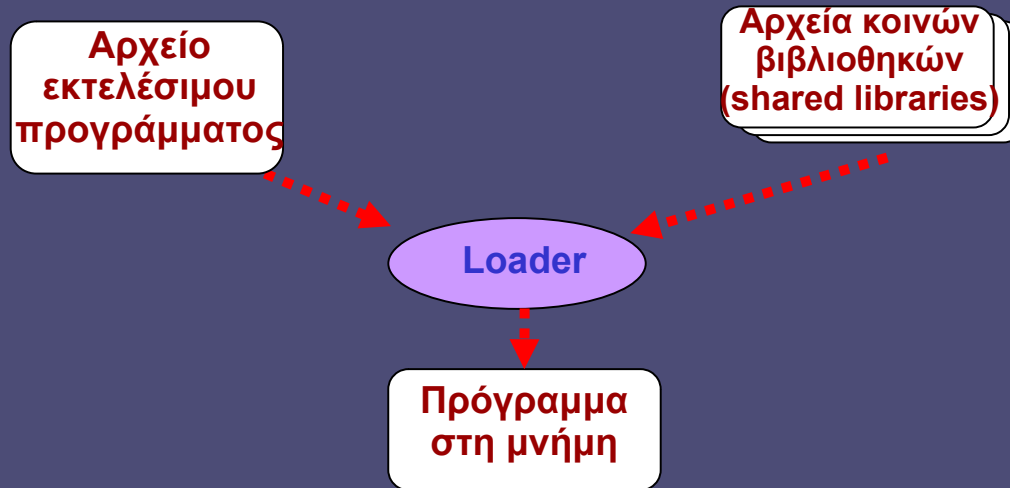
- Ο μεταγλωττιστής (**compiler**) μεταφράζει το αρχείο γλώσσας υψηλού επιπέδου (**αρχείο κειμένου**) σε εκτελέσιμη γλώσσα μηχανής (**δυναμικό αρχείο**)

# Linker (συνδέτης)



- Η έξοδος από τον μεταγλωττιστή (object code) περιέχει κενά
  - Εξωτερικές αναφορές σε μεταβλητές και συναρτήσεις
- **Linker**
  - Συνδέει όλα τα δυαδικά αρχεία κώδικα (object code)
  - Και τα αρχεία κώδικα βιβλιοθηκών που θα περιληφθούν
  - Κατασκευάζει το τελικό εκτελέσιμο πρόγραμμα (δυαδικό αρχείο)

# Loader (φορτωτής)



- **Loader**

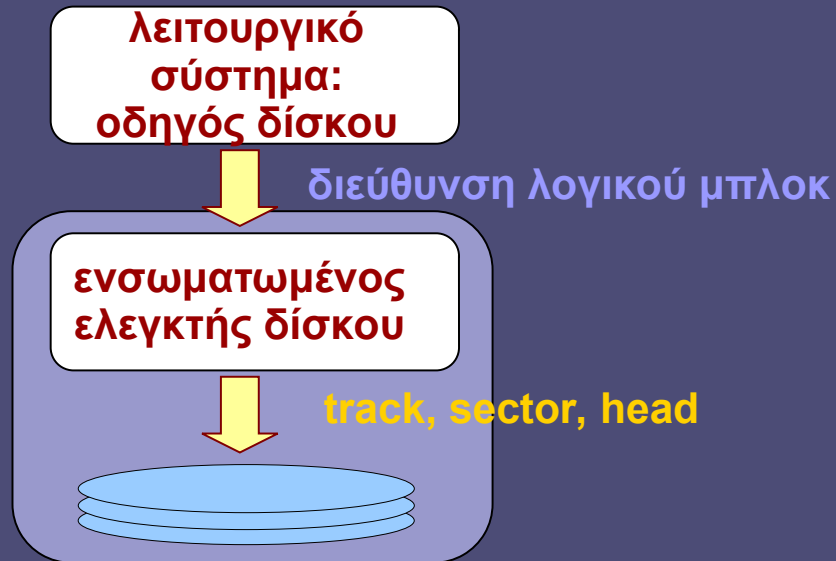
- Προετοιμασία για την εκτέλεση του προγράμματος
  - Φόρτωση κώδικα-δεδομένων στη μνήμη
- Δυναμική σύνδεση κοινών βιβλιοθηκών
  - Κώδικας που δεν ενσωματώνεται στο πρόγραμμά μας αλλά υπάρχει σε μοναδικό αντίγραφο για όλες τις εφαρμογές που τον χρησιμοποιούν

# Συστήματα Αρχείων

- **File systems**
  - Οργάνωση αρχείων
    - Ονόματα αρχείων
    - Σε καταλόγους (directories) / φακέλους (folders)
  - Μέθοδοι για τη δημιουργία, τροποποίηση και διαγραφή αρχείων
  - Διαμοιρασμός κοινών αρχείων
  - Έλεγχος πρόσβασης

# Οργάνωση δίσκων

- Οργάνωση σε ακολουθία λογικών μπλοκ
  - logical block addressing (LBA)
  - $0...n$  λογικά μπλοκ
  - Ανεξάρτητα από τη φυσική κατασκευή του δίσκου



# Διαμέριση δίσκου (partitioning)

- **Partitions**
  - Ο συνολικός δίσκος χωρίζεται σε μικρότερα τμήματα (διαμερίσεις)
- **Γιατί;**
  - Διατήρηση περισσότερων του ενός ΛΣ στο σύστημα
    - dual (multi)-booting
  - Διατήρηση ΛΣ σε ξεχωριστό τμήμα από τα δεδομένα του χρήστη
    - για ευκολότερες επανεγκαταστάσεις

# Το περιεχόμενο ενός partition

- **Boot Block**
  - Εκτελέσιμος κώδικας
    - Ο υπολογιστής μπορεί να ξεκινήσει τη λειτουργία του εκτελώντας κώδικα στο partition αυτό
- **Super Block**
  - Κεντρικές πληροφορίες για το σύστημα αρχείων που περιέχεται στο partition
- **Πληροφορία ελεύθερου χώρου**
- **Πληροφορία θέσης αρχείων**
  - Ποιο αρχείο βρίσκεται πού στον δίσκο
- **Περιεχόμενα Αρχείων και Καταλόγων**

# Πώς αποθηκεύονται τα αρχεία;

- **Αποθήκευση σε μπλοκ (ή “cluster”)**
  - π.χ. 4KB ανά μπλοκ
  - Προφανώς ένα αρχείο μπορεί να καταλαμβάνει περισσότερα από ένα μπλοκ
    - και ένας κατάλογος (φάκελος) επίσης (είναι αρχείο κι αυτός)
- **Μέθοδος δέσμευσης μπλοκ #1: συνεχόμενα μπλοκ**
  - Όλο το αρχείο σε συνεχόμενα μπλοκ
    - Γρήγορη προσπέλαση – αρκεί να ξέρουμε αρχή και τέλος
    - Τι συμβαίνει όταν το αρχείο αλλάζει μέγεθος;
    - Fragmentation

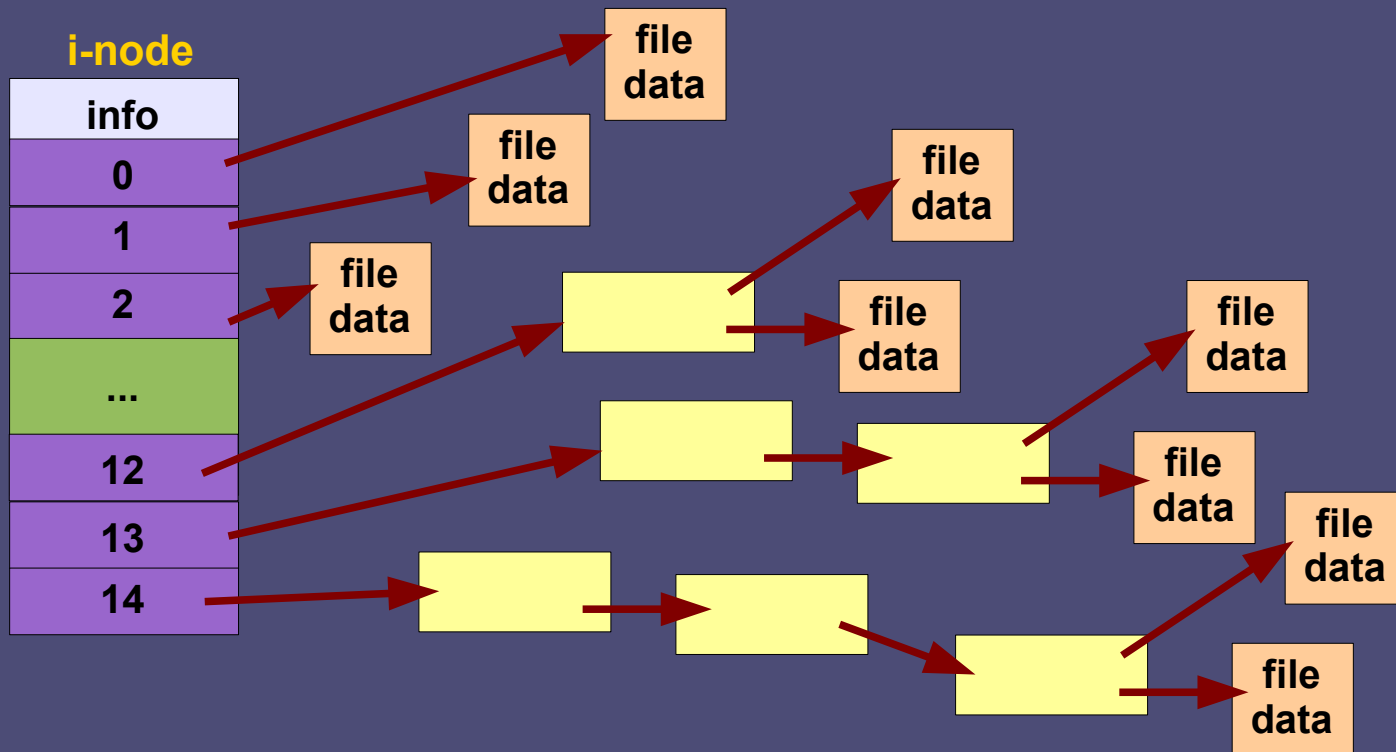


# Πώς αποθηκεύονται τα αρχεία;

- Μέθοδος δέσμευσης μπλοκ #2: διασυνδεδεμένη λίστα
  - Σε κάθε μπλοκ υπάρχει δείκτης για το επόμενο μπλοκ
    - Αρκεί να ξέρουμε το πρώτο μπλοκ
    - Πώς θα προσπελάσω ένα σημείο προς το τέλος του αρχείου;
- Μέθοδος δέσμευσης μπλοκ #2β: διασυνδεδεμένη λίστα (παραλλαγή)
  - Ξεχωριστός πίνακας πληροφορίας για κάθε μπλοκ
  - Αν είναι δεσμευμένο και ποιο είναι το επόμενο
    - Γρήγορη προσπέλαση τυχαίου σημείου αρχείου
    - Ο πίνακας καταλαμβάνει μεγάλο χώρο

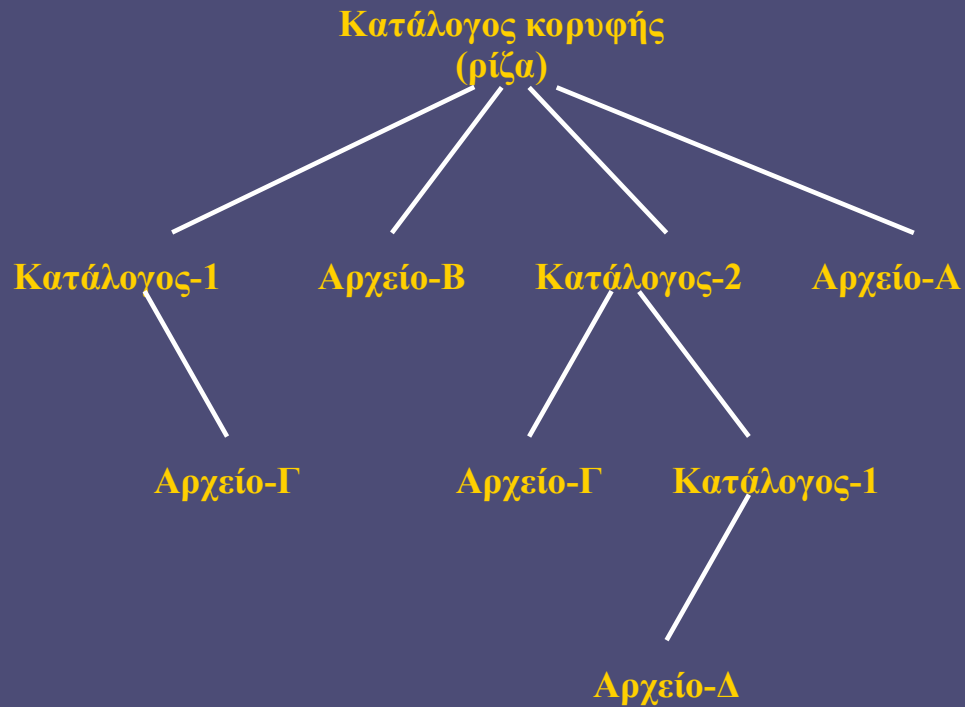
# Πώς αποθηκεύονται τα αρχεία;

- Μέθοδος δέσμευσης μπλοκ #3: έμμεση δεικτοδότηση
  - Ειδικά μπλοκ (i-nodes) περιέχουν δείκτες σε άλλα μπλοκ
    - Π.χ. συστήματα αρχείων στο ΛΣ Unix



# Ιεραρχική δομή καταλόγων

- Η ρίζα (/) είναι στην κορυφή
  - Οι κατάλογοι (φάκελοι) είναι ειδικά αρχεία που περιέχουν ζεύγη (όνομα αρχείου, πληροφορία αρχείου)

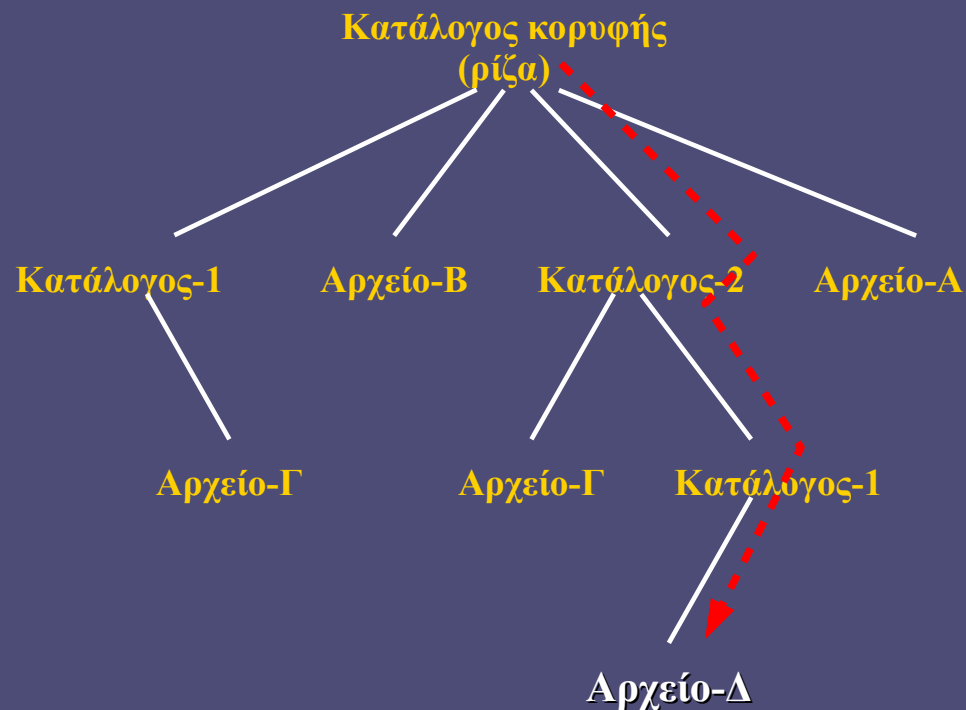


# Πού βρίσκεται η ρίζα;

- **Εξαρτάται από το Λειτουργικό Σύστημα**
  - Windows: κάθε partition που αναγνωρίζεται προστίθεται ως ένα λογικό drive (π.χ. C: D: κλπ), το οποίο αποτελεί τη ρίζα για όλους τους φακέλους που περιέχει
    - άρα έχουμε ένα δάσος από δέντρα
  - τύπου Unix: υπάρχει μια μοναδική ρίζα και κάθε partition μπορεί να τοποθετηθεί (mount) σε οποιοδήποτε σημείο του δέντρου
    - δεν υπάρχουν λογικά drives

# Μονοπάτι Αρχείου (file path)

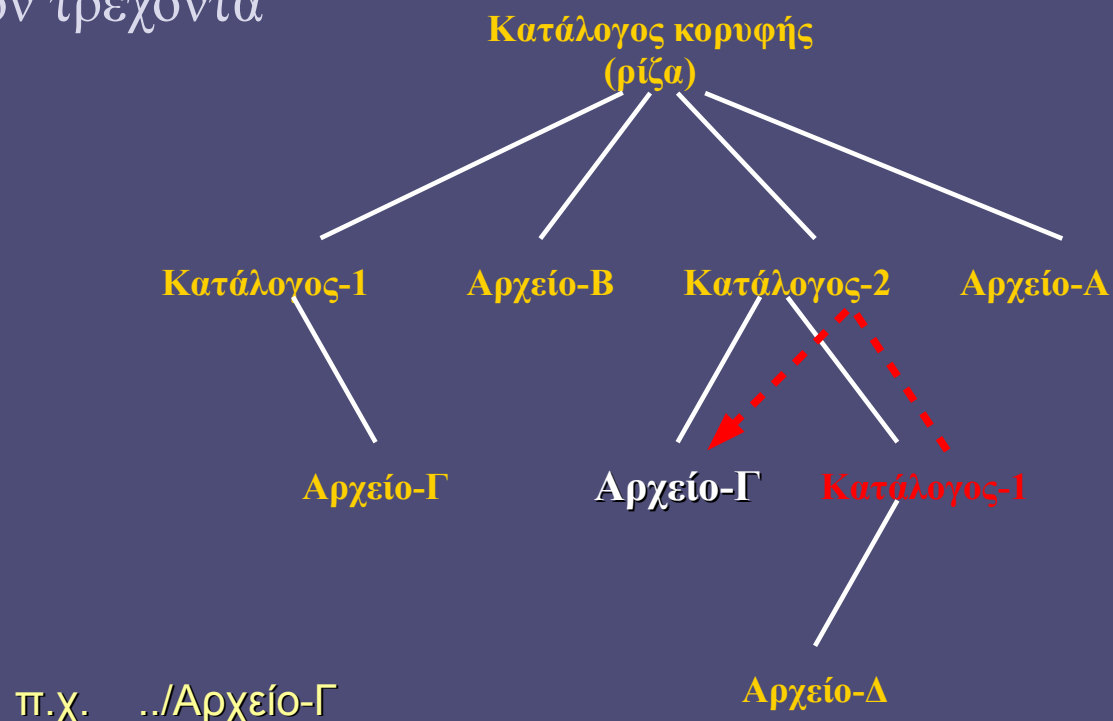
- Από τη ρίζα προς το αρχείο (απόλυτο μονοπάτι)
  - Ως **αναγνωριστικό** του αρχείου που επιλέγουμε



π.χ. /Κατάλογος-2/Κατάλογος-1/Αρχείο-Δ

# Μονοπάτι Αρχείου

- Σχετικό μονοπάτι
  - Έστω ότι ο **τρέχων κατάλογος (φάκελος)** (working directory) είναι ο **Κατάλογος-1**
  - Για ευκολία: **.** είναι ο τρέχων φάκελος και **..** είναι ο φάκελος πάνω από τον τρέχοντα



# Ονόματα Αρχείων

- **Παλαιότερα υπήρχαν περιορισμοί**
  - π.χ η μορφή 8.3
    - 8 χαρακτήρες (αλφαριθμητικοί και ορισμένα σημεία στίξης)
    - τελεία και 3 χαρακτήρες επέκταση
  - Η επέκταση (και σήμερα) δηλώνει τον τύπο του αρχείου
    - χωρίς να ισχύει κατ' ανάγκη
    - .doc .txt .html .pdf .exe κλπ
- **Σήμερα υπάρχει μεγαλύτερη ελευθερία στα ονόματα των αρχείων**
  - αρκεί στον ίδιο κατάλογο το όνομα να είναι μοναδικό
    - υπάρχουν ακόμα μη επιτρεπτοί χαρακτήρες στο όνομα

# Δικαιώματα (Permissions)

- Ποιος χρήστης (ή ομάδα χρηστών) μπορεί να κάνει τι σε ένα αρχείο
  - Παράδειγμα: Unix permissions

permissions		user	group						
drwxr-xr-x	2	mistral	mistral	4096	2007-10-07	21:23	.		
drwxr-xr-x	5	mistral	mistral	4096	2007-10-07	21:05	..		
-rw----r--	1	mistral	mistral	5136	2007-10-07	21:08	banner.gif		
-rw----r--	1	mistral	mistral	4237	2007-10-07	21:23	index.html		

user group other