

Λειτουργικά Συστήματα (HY321)

Διάλεξη 14: Οργάνωση Συστημάτων Αρχείων



Αρχεία: Ονόματα Θέσεων στο Δίσκο



- Οπτική του χρήστη:
 - Μια ακολουθία από bytes με συγκεκριμένο όνομα



- Οπτική του ΛΣ:
 - Σύνολο από ορμαθούς (blocks/sectors)
- Σύστημα αρχείων
 - Μεταφραστής!
 - Όνομα: Μετατόπιση -> Ορμαθός δίσκου
 - Βασικές λειτουργίες
 - Δημιουργία / Διαγραφή αρχείου
 - Ανάγνωση από / Εγγραφή σε αρχείο
 - Μπορώ να έχω λειτουργίες με...
 - Όσο το δυνατό λιγότερες προσπελάσεις στο δίσκο
 - Και όσο το δυνατό μικρότερη επιβάρυνση σε χώρο;



Ε, και Πού το Δύσκολο;

- Παρόμοια με την Ιδεατή Μνήμη
 - **Μεταδεδομένα** του συστήματος αρχείων (ΣΑ):
 - Δομές δεδομένων για την αναπαράσταση απεικονίσεων (μεταφράσεων)
 - Όπως και οι πίνακες σελίδων
 - Πίνακας σελίδων: # ιδεατής σελίδας σε # φυσικού πλαισίου



- Μεταδεδομένα ΣΑ: Όνομα + μετατόπιση σε ορμαθό δίσκου



- Κατάλογος: Όνομα αρχείου / καταλόγου σε ορμαθό δίσκου



- Άλλαξε ο Μανωλιός...

Ιδεατή Μνήμη και Συστήματα Αρχείων: Βίοι Παράλληλοι αλλά και Διαφορετικοί...



- Παρόμοια ζητούμενα:
 - Διάφανη «επανατοποθέτηση», ελευθερία όσον αφορά το μέγεθος, προστασία
- Ενίστε ευκολότερο πρόβλημα
 - Ο χρόνος υπολογισμού για τις «μεταφράσεις» στο ΣΑ είναι «αμελητέος» (δε χρειάζεται κάτι σαν τον TLB).
 - Οι πίνακες σελίδων αντιμετωπίζουν αραιούς χώρους διευθύνσεων και τυχαίες προσπελάσεις. Τα αρχεία είναι «πυκνά» ($0 \dots \text{filesize}-1$) & συχνά σχεδόν ακολουθιακά.
- Ενίστε όμως και δυσκολότερο
 - Κάθε επίπεδο μετάφρασης => πιθανή προσπέλαση στο δίσκο.
 - Μεγάλη επιβάρυνση σε χώρο!
 - Μα τι χώρο; Ο δίσκος είναι μεγάλος. Τι συμβαίνει;
 - Για τη διαχείριση;
 - Να μη βάλουμε και μια cache;
 - Ο χώρος της cache δεν είναι ποτέ αρκετός
 - Τα δεδομένα που θα μεταφέρεις με μια προσπέλαση δεν είναι ποτέ αρκετά
 - Πόσα; Όσο μια διαδρομή;
 - Μεγάλη ποικιλία μεγεθών: Πολλά <10k, μερικά όμως αρκετά GB.

Λίγη «Διαίσθηση» του Προβλήματος



- Η **επίδοση** του ΣΑ εξαρτάται άμεσα από τον **# προσπελάσεων δίσκου**
 - Κάθε προσπέλαση: 10δες msec.
 - 50-100 παραπάνω προσπελάσεις => 1 δευτερόλεπτο.
 - Δισεκατομμύρια εντολών CPU στον ίδιο χρόνο!
- Κόστος προσπέλασης: εξαρτάται κυρίως από τη μετακίνηση
 - Μπορούμε να φέρουμε 1024 φορές περισσότερα δεδομένα με ~46% επιβάρυνση.
 - 1 τομέας = $\sim 10\text{ms} + \sim 3\text{ms} + 0.006\text{msec} = \sim 13\text{ms}$. (για δίσκο 10000 rpm)
 - 1 διαδρομή (1024 τομείς) = $\sim 10\text{ms} + 6\text{ms} = \sim 16\text{ms}$.
- Παρατηρήσεις:
 - Όλοι οι ορμαθοί σε ένα αρχείο τείνουν να χρησιμοποιούνται μαζί, ακολουθιακά
 - Όλα τα αρχεία / ονόματα σε έναν κατάλογο τείνουν να χρησιμοποιούνται μαζί.
 - Πώς θα το εκμεταλλευτούμε;

Συνήθεις Τρόποιι Προσπέλασης



- Ακολουθιακή

- Δεδομένα αρχείων προσπελούνονται με τη σειρά
- Πολύ πολύ πολύ συνηθισμένος τρόπος προσπέλασης
- Π.χ. ένας μεταφραστής διαβάζει ένα αρχείο, παίζετε ένα mp3 ή ένα DVD...

- «Τυχαία»

- Προσπέλαση κάθε τμήματος του αρχείου απευθείας, χωρίς να περάσουμε από τα προηγούμενα
- Π.χ. χώρος στο δίσκο που χρησιμοποιεί το σύστημα ιδεατής μνήμης, βάσεις δεδομένων ...

- Με «κλειδί»

- Ψάξε για έναν ορμαθό που περιέχει συγκεκριμένες τιμές.
- Π.χ. συσχετιστική βάση δεδομένων, κατάλογος...
- Συνήθως δεν υποστηρίζεται άμεσα από το ΛΣ

Και Πώς... Θα Βρω τα Δεδομένα ενός Αρχείου στο Δίσκο;



- Διαχείριση δίσκου
 - Πού είναι τα περιεχόμενα κάθε αρχείου στο δίσκο;
 - Σε ποιο ορμαθό αντιστοιχεί κάθε μετατόπιση από την αρχή του αρχείου;
 - Περιγραφέας αρχείου (file descriptor): Δομή δεδομένων για κάθε αρχείο
 - Αποθηκεύονται στο δίσκο μαζί με τα αρχεία
- Δεν ξεχνώ...
 - Τα περισσότερα αρχεία είναι μικρά...
 - Το «ελάχιστο» (βασικό) κόστος (επιβάρυνση) πρέπει να είναι μικρό
 - ... αλλά το μεγαλύτερο μέρος του δίσκου ανήκει σε μεγάλα αρχεία ...
 - ... στα οποία αναφέρεται και το μεγαλύτερο ποσοστό των ενεργειών I/O
 - Πρέπει να μπορώ να αναπαραστήσω «φθηνά» αυτά τα αρχεία ...
 - ...και να τα προσπελάσω γρήγορα
 - Θέλω γρήγορη ακολουθιακή και τυχαία προσπέλαση

Απλός Μηχανισμός: Τοποθέτηση σε Συνεχόμενες Θέσεις



- Μοιάζει με την τμηματοποίηση μνήμης
 - Όταν φτιάχνεται ένα αρχείο ο χρήστης **προκαθορίζει το μέγεθος**
 - Δεσμεύεται απευθείας όλος ο χώρος σε **συνεχόμενες θέσεις**
 - Περιεχόμενα περιγραφέα: **Αρχή, μέγεθος**
 - (Προ)ιστορικό παράδειγμα: IBM OS/360

- Τι γίνεται αν το αρχείο γ απαιτεί 2 τομείς;
- Αν το α χρειαστεί να μεγαλώσει κατά 2 τομείς;

Δίσκος



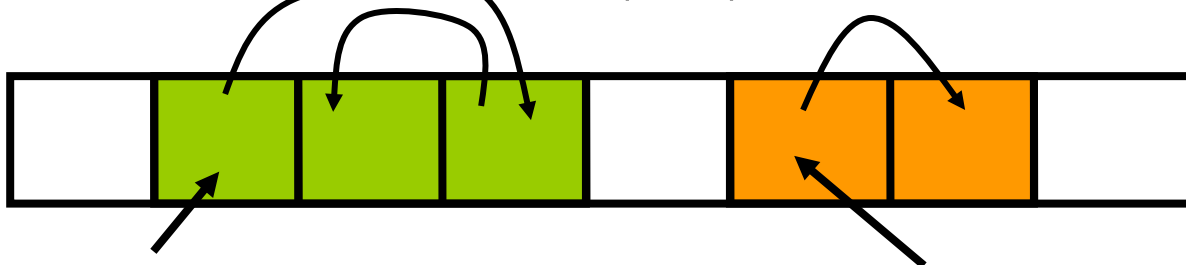
Αρχείο α (αρχή=1, μήκος=3) **Αρχείο β (αρχή=5, μήκος=2)**

- Πλεονεκτήματα:
 - Απλό
 - Γρήγορη προσπέλαση (ακολουθιακή και τυχαία)
- Μειονεκτήματα (τι συνέβη στην τμηματοποίηση);
 - Δύσκολο να προβλέψεις το μέγεθος εξαρχής
 - Κατακερματισμός



Διασυνδεδεμένοι Ορμαθοί

- Κατά βάση **διασυνδεδεμένες λίστες** στο δίσκο
 - Διασυνδεδεμένη λίστα όλων των ελεύθερων ορμαθών
 - Περιγραφέας αρχείου: **Δείκτης στον 1^ο ορμαθό**
 - **Καθένας** ορμαθός περιέχει **δείκτη στον επόμενο**
- Παραδείγματα: DOS (FAT), Alto ...



Αρχείο α (αρχή=1, μήκος=3) Αρχείο β (αρχή=5, μήκος=2)

- Πλεονεκτήματα:
 - Δυναμική αύξηση / μείωση του μεγέθους αρχείων
 - Εξάλειψη κατακερματισμού
 - Εύκολη ακολουθιακή προσπέλαση
- Μειονεκτήματα:
 - «Αδύνατη» η τυχαία προσπέλαση
 - Ακόμα και η ακολουθιακή προβληματική

• Αν θέλουμε να μεγαλώσουμε το α;

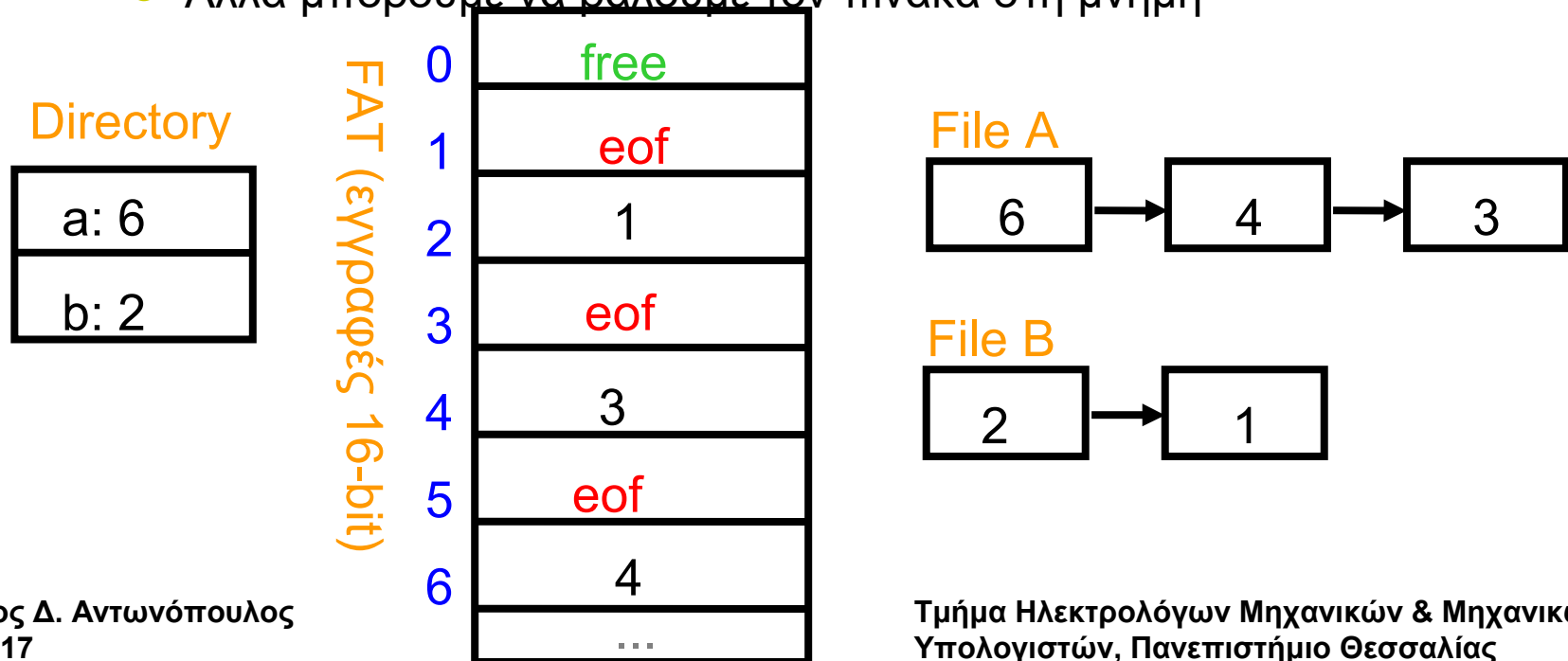
• Αν θέλουμε να βάλουμε το γ με μήκος 3;

• Αν θέλουμε να βρούμε τον τελευταίο ορμαθό του α;



Σύστημα Αρχείων DOS (FAT)

- Διασυνδεδεμένες λίστες ορμαθών
- Κόλπο: Ο χώρος για τις διασυνδεδεμένες λίστες σε **συγκεκριμένο σημείο** στο δίσκο (**FAT** – File Allocation Table)
 - Αντί σε κάθε ορμαθό δείκτης στον επόμενο
 - Και πάλι διατρέχουμε ακολουθίες δεικτών
 - Αλλά μπορούμε να βάλουμε τον πίνακα στη μνήμη





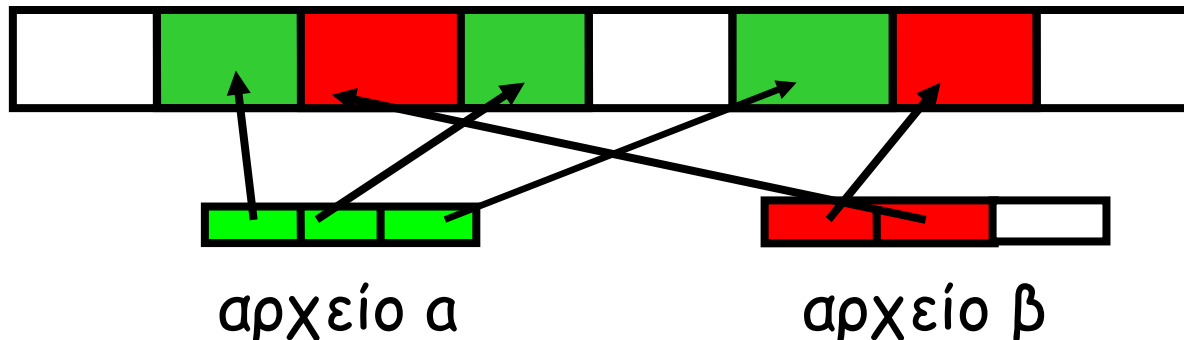
Περί του FAT (16) ...

- Εγγραφή 16 bits
 - Μέγιστο μέγεθος FAT;
 - 2^{16} εγγραφές * 2 bytes/εγγραφή = 128 KBytes
 - Μέγιστο μέγεθος δίσκου (έστω τομέας 0.5 Kbytes);
 - 2^{16} τομείς * 0.5 Kbytes/τομέα = 32 Mbytes
 - Μικρούλι...
 - Ας μεγαλώσουμε τους τομείς!
 - + γρηγορότερη προσπέλαση
 - - παραπάνω κατακερματισμός (εσωτερικός)
- **Επιβάρυνση** σε χώρο
 - Στοιχειώδης (128KB / 32 MB => ~ 4%)
- **Αξιοπιστία** (αν χάσουμε το FAT χάσαμε)
 - Αντίγραφο
- Που είναι ο αρχικός κατάλογος;
 - Σε συγκεκριμένο σημείο στο δίσκο
 - Αμέσως μετά το FAT (και ενδεχομένως και το αντίγραφό του)
- Φαντάζεστε τι είναι το FAT32; ☺



Αρχεία με Δεικτοδότηση

- Σε κάθε αρχείο αντιστοιχεί ένα **διάνυσμα** με **όλους τους δείκτες** στους ορμαθούς του αρχείου
 - Δεν ξεχνώ: Πίνακες σελίδων
 - Από τι περιορίζεται το μέγεθος του αρχείου;
 - Μέγεθος του διανύσματος
 - Δημιουργία αρχείου
 - Δέσμευσε το διάνυσμα δεικτών
 - Δέσμευσε ορμαθούς στο δίσκο **όταν** χρειάζεται

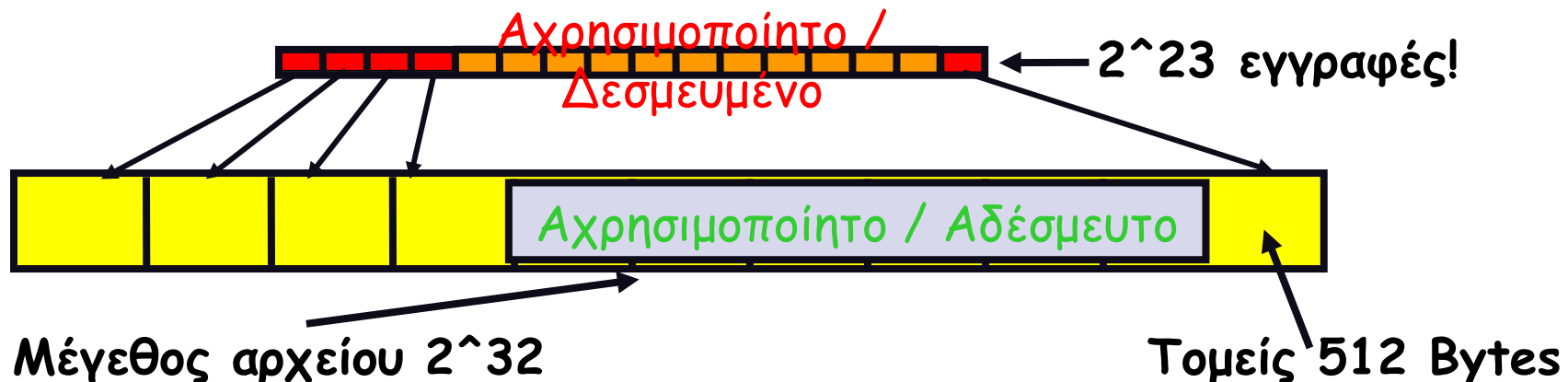


- Πλεονεκτήματα;
 - Εύκολη ακολουθιακή και τυχαία προσπέλαση
- Μειονεκτήματα;
 - Μεγάλη **συνεχής** περιοχή δίσκου (για το διάνυσμα)
 - Άντε ξανά τα ίδια

Αρχεία με Δεικτοδότηση – Μέρος 2^ο



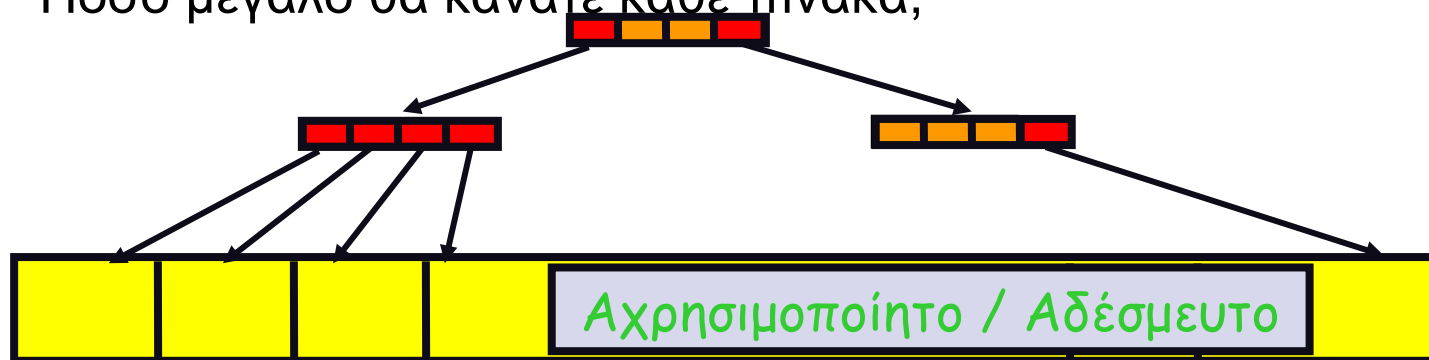
- Μα τι ζητάω;
 - Αύξηση κατά βούληση
 - Όχι αχρησιμοποίητα τμήματα δίσκου
 - Όχι αναγκαστικά συνεχείς περιοχές δίσκου
- Είναι τόσοοοοο πρόβλημα;
 - Αρχείο μέγιστου μεγέθους 4GB, τομείς 512 Bytes
 - Πόσοι τομείς (μέγιστο); 8M τομείς
 - Πόσο μεγάλο διάνυσμα; 8M θέσεις
 - Πόσος χώρος (αν θέλω π.χ. 32 bits για να δεικτοδοτήσω κάθε τομέα); 32M
 - Αν χρησιμοποιώ 64M από το αρχείο, τι επιβάρυνση; 33%



Αρχεία με Δεικτοδότηση – Μέρος 3^ο



- Και τι κάνω τώρα;
 - Τι έκανα με τους πίνακες σελίδων;
 - Μικροί πίνακες με δείκτες
 - Δείχνουν σε άλλους πίνακες ...
 - ... και οι δείκτες αυτών σε άλλους πίνακες ...
 - ... κοκ ... και οι δείκτες αυτών σε τομείς
 - Πόσο μεγάλο θα κάνατε κάθε πίνακα;



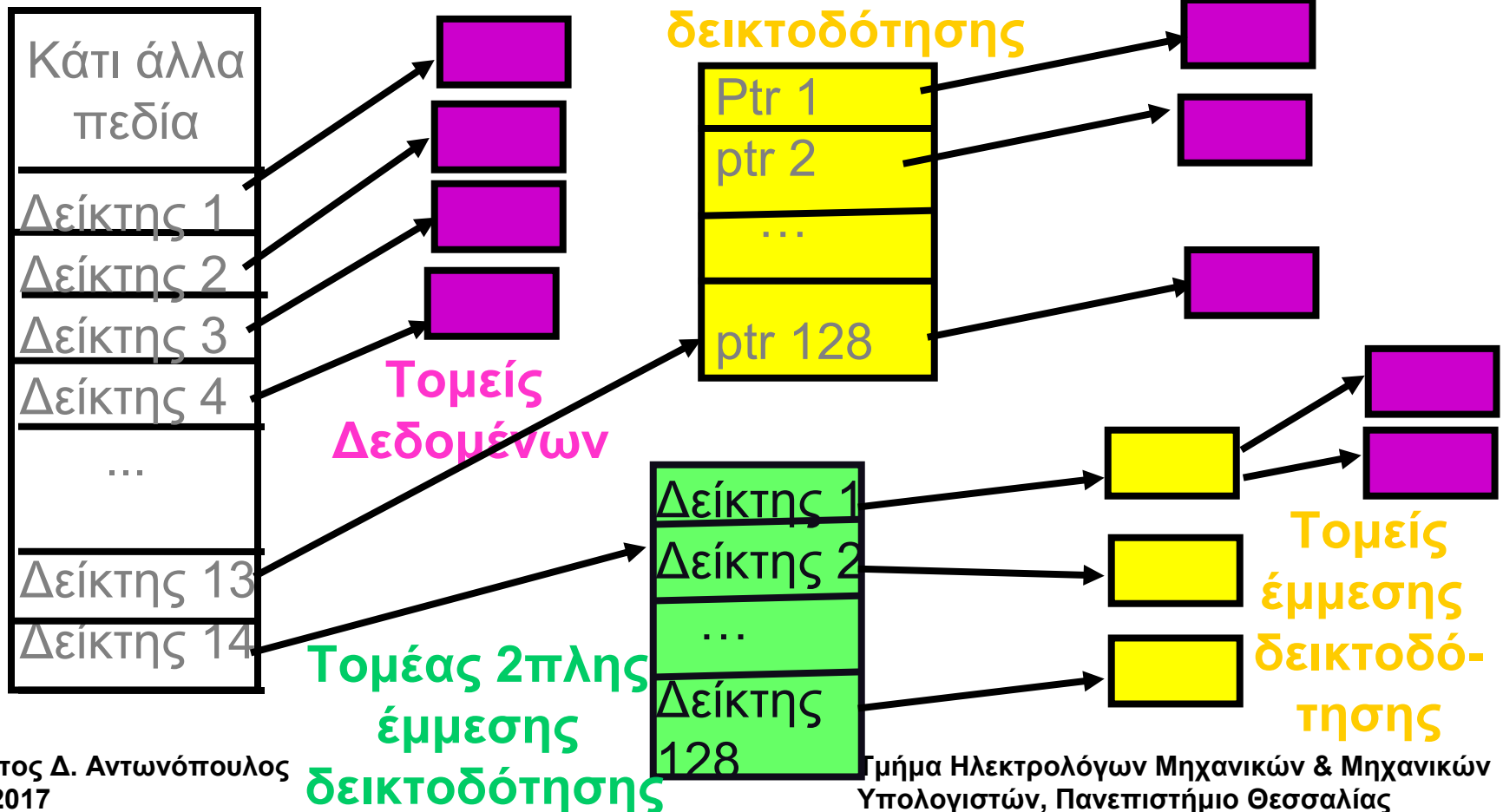
- Μειονέκτημα;
 - Πολλαπλές προσπελάσεις (διαχείρισης) στο δίσκο για μια πραγματική προσπέλαση
 - Η κακή περίπτωση όσον αφορά το χώρο;

Αρχεία με Δεικτοδότηση Πολλαπλών Επιπέδων (π.χ. BSD)



- Περιγραφέας αρχείου (**inode**):

- 14 δείκτες
- Και κάτι άλλα πεδία...





Λίγα Παραπάνω για τα inodes

- Τα inodes αποθηκεύονται σε **διάνυσμα σταθερού μήκους**
 - Το μέγεθος του διανύσματος καθορίζεται κατά τη διαμόρφωση του συστήματος αρχείων και δε μπορεί να αλλάξει. Παλιά βρισκόταν μαζεμένο σε μια πλευρά του δίσκου
 - Τώρα το απλώνουμε σε όλο το δίσκο
 - Γιατί;
- Ο «δείκτης» ενός inode στο διάνυσμα λέγεται **i-number**. Το ΛΣ αναφέρεται στα αρχεία με το i-number τους.
- Όταν ανοίγουμε ένα αρχείο το inode μεταφέρεται στη μνήμη.
 - Όταν το κλείνουμε γράφεται πίσω στο δίσκο (αν έχει αλλάξει)



Αρχεία και Κατάλογοι

- Ο χρήστης πρέπει να μπορεί να ξαναβρεί το αρχείο του...
- Προσέγγιση α'
 - Ας θυμάται που ακριβώς υπάρχει το αρχείο στο δίσκο
 - Πώς δηλαδή μπορεί και θυμάται το Α.Φ.Μ. του;
 - ☹️ Οι άνθρωποι θέλουν ευνόητα, εύληπτα ονόματα
 - Κατάλογοι: Απεικόνιση ονομάτων σε ορθογώνιους δίσκους



Σας Αρέσει η Ιστορία;

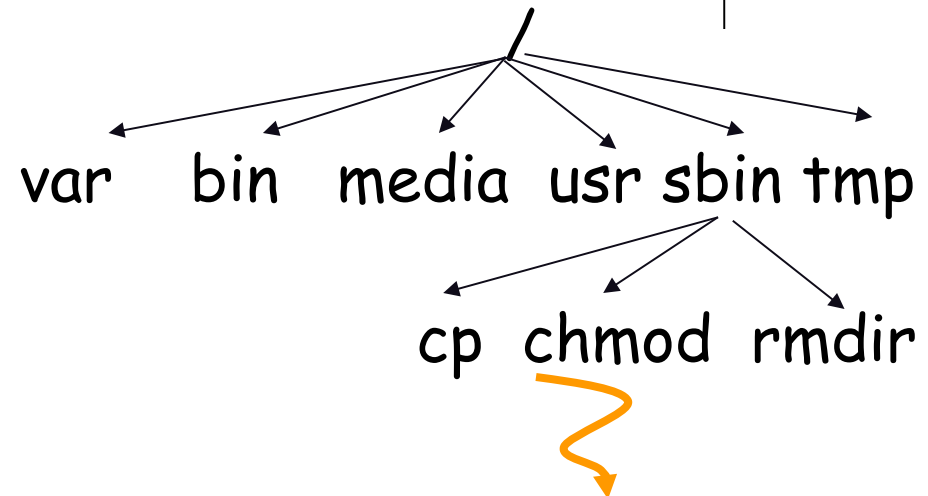
- Προσέγγιση β': Ένας κατάλογος για όλο το σύστημα
 - Σε γνωστή θέση στο δίσκο
 - Περιέχει ζεύγη <όνομα, ορμαθός δίσκου>
 - Αν ένας χρήστης χρησιμοποιήσει ένα όνομα, δε μπορεί να το χρησιμοποιήσει κανείς άλλος
 - Παράδειγμα: Παλιοί προσωπικοί υπολογιστές
- Προσέγγιση γ': Ας έχουμε ένα κατάλογο για κάθε χρήστη
 - Καλύτερο, αλλά όχι αρκετά.
 - Αν κάνεις $1s$ σε 10,000 αρχεία την πάτησες.
 - Πολλοί από τους καθηγητές σας δουλεύουν/με κάπως έτσι
- Προσέγγιση δ': Ιεραρχική ονοματολογία
 - Ο κατάλογος περιέχει αρχεία ή και άλλους καταλόγους
 - Το σύστημα αρχείων έχει τη μορφή δέντρου (ενίοτε και γράφου).
 - Ούτως ή άλλως οι μεγάλοι χώροι ονομάτων τείνουν να οργανώνονται ιεραρχικά... (Διευθύνσεις ip, Ονόματα στο internet, κλπ)





Ιεραρχικοί Κατάλογοι στο Unix

- Χρησιμοποιούνται από τη δεκαετία του '60
 - Το Unix τους υιοθέτησε
- Οι **κατάλογοι** αποθηκεύονται στο δίσκο **σαν κανονικά αρχεία**
 - Ο inode απλά **μαρκάρεται** σαν inode καταλόγου με ένα ειδικό bit
 - Οι χρήστες μπορούν να τους διαβάσουν σαν και οποιοδήποτε άλλο αρχείο
 - Μόνο ειδικά προγράμματα μπορούν να τους γράψουν
 - Γιατί;
 - Το αρχείο στο οποίο οδηγεί ένας δείκτης μπορεί να είναι ένας άλλος κατάλογος
 - Το ΣΑ οργανώνεται σαν δέντρο
 - Πότε θα γίνει κατευθυνόμενος άκυκλος γράφος;
- Απλό
- Επιτάχυνση προσπέλασης σε αρχεία ⇔ επιτάχυνση προσπέλασης σε καταλόγους



<name, inode#>

<var, 1021>

<bin, 1020>

<media, 1022>

<usr, 4123>

<sbin, 1001>

<tmp, 1011>

...



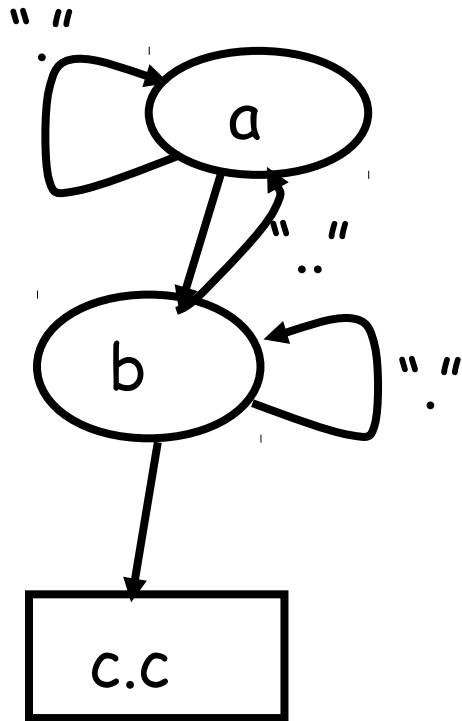
Από Πού να Ξεκινήσω;

- Εκκίνηση (Bootstrapping): Πού αρχίζει το ΣΑ;
 - Αρχικός κατάλογος (root).
 - inode #2 στο Unix.
 - Οι 0 και 1 σε διατεταγμένη υπηρεσία...
- Ειδικά ονόματα:
 - Αρχικός κατάλογος του ΣΑ (root): “/”
 - Τρέχων κατάλογος: “.”
 - «Γονικός» κατάλογος: “..” (αλλιώς δε θα μπορούσαμε να μετακινηθούμε προς τα πάνω ...)
 - Ο αρχικός κατάλογος του χρήστη: “~”
- Χρειάζονται μόνο 2 εντολές για να κινηθούμε στον κατάλογο:
 - **cd** ‘όνομα’: Πήγαινε στον κατάλογο “όνομα”
 - **ls** : Δείξε όλα τα ονόματα (αρχεία, υποκαταλόγους) στον τρέχοντα κατάλογο

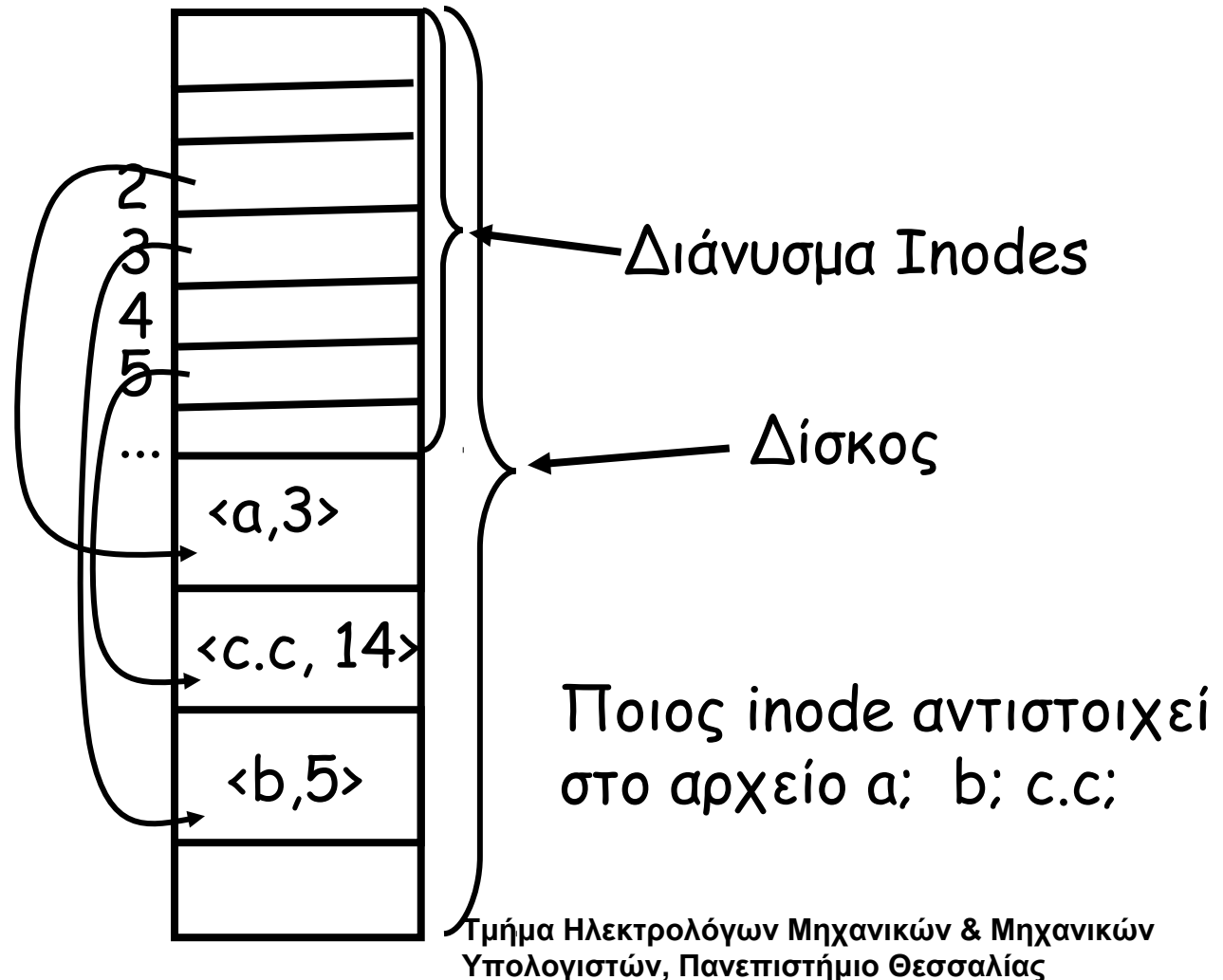


Παράδειγμα: /a/b/c.c

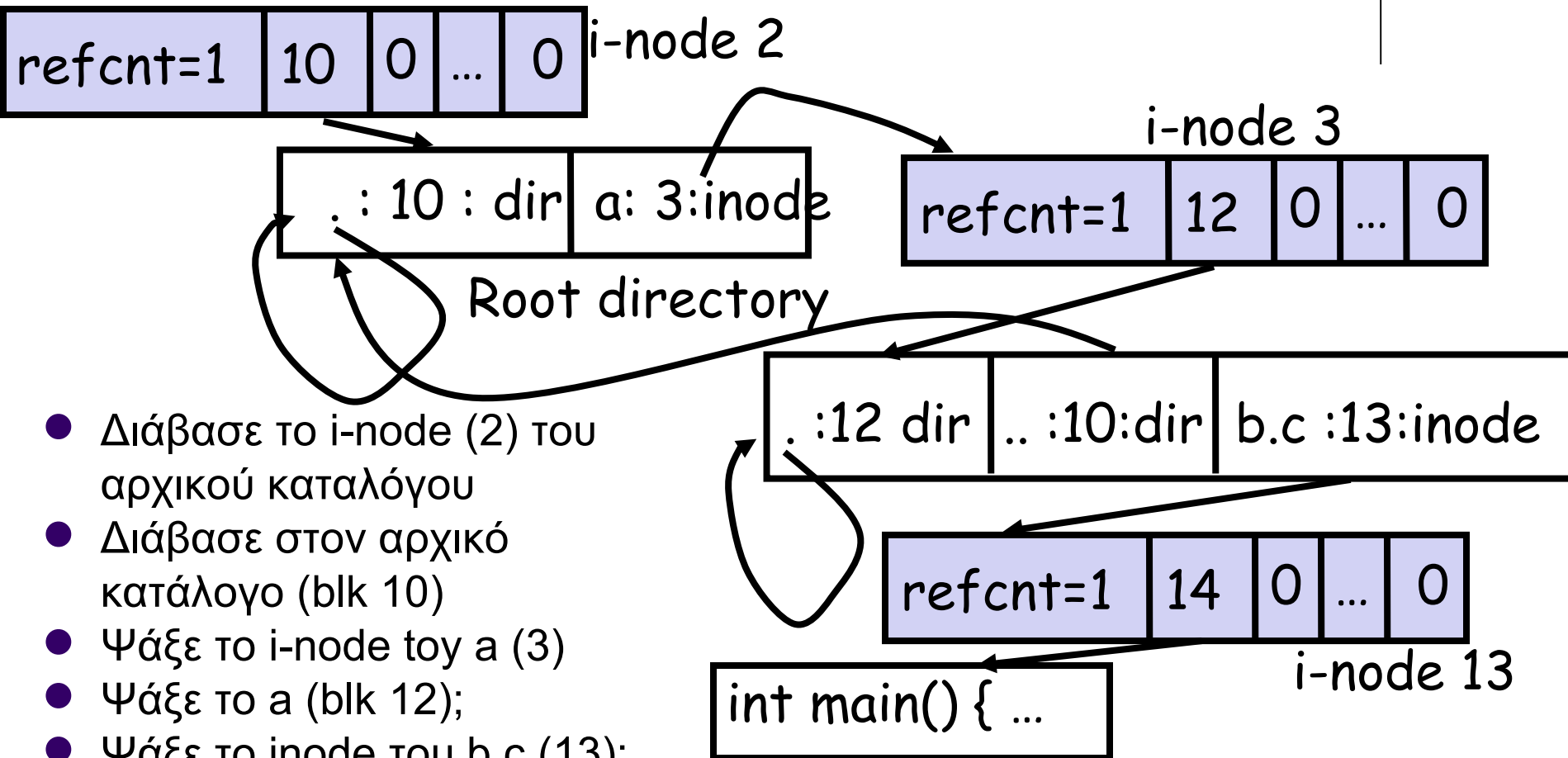
Λογικά Ονόματα



Φυσική Οργάνωση



Παράδειγμα: Αλλαγή byte 4 στο /a/b.c



- Διάβασε το i-node (2) του αρχικού καταλόγου
- Διάβασε στον αρχικό κατάλογο (blk 10)
- Ψάξε το i-node του a (3)
- Ψάξε το a (blk 12);
- Ψάξε το inode του b.c (13);
- Χρησιμοποίησε το inode για να βρεις το blk που αντιστοιχεί στο byte 4 (μέγεθος blk = 512, οπότε μετατόπιση = 0, άρα blk 14); διάβασε και άλλαξε



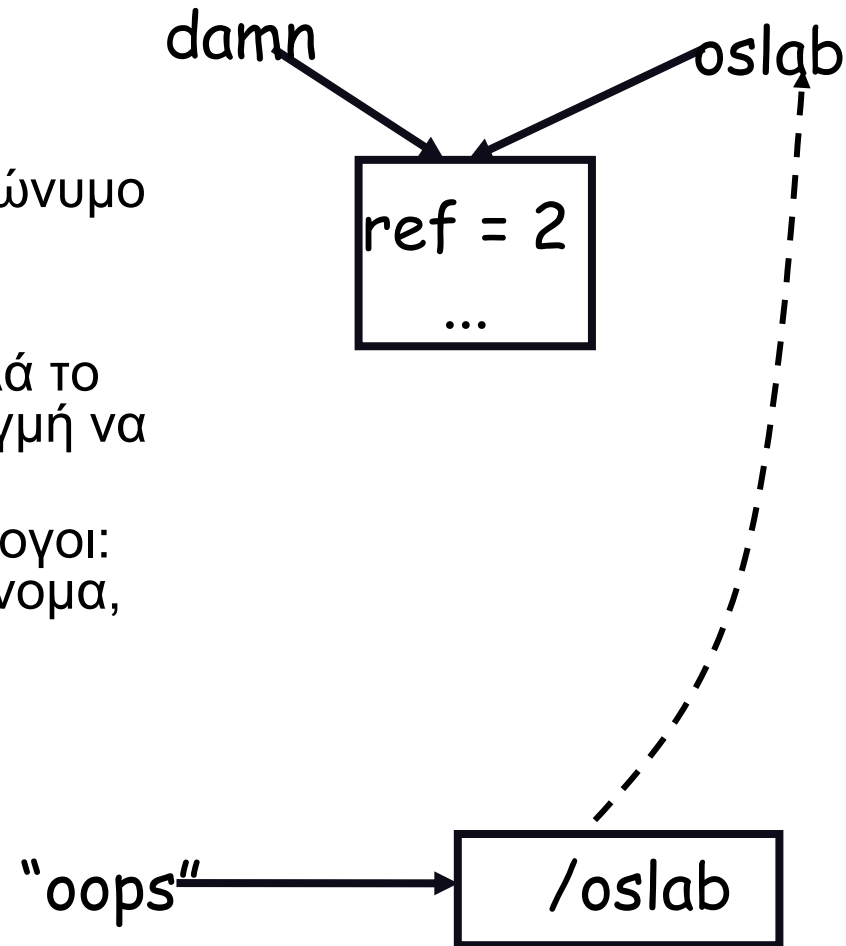
Ο Τρέχων Κατάλογος

- Δύσκολο να δίνουμε πάντα πλήρεις καταλόγους (**απόλυτα ονόματα**)
 - Δηλαδή τη διαδρομή από την αρχή του δέντρου καταλόγων
 - Κάθε διεργασία σχετίζεται με έναν “**τρέχοντα κατάλογο**”.
 - Ονόματα αρχείων που δεν αρχίζουν με “/” θεωρούνται **σχετικά** ως προς τον τρέχοντα κατάλογο,
- Τα κελύφη (το περιβάλλον με το οποίο αλληλεπιδρά ο χρήστης) διατηρούν μια προκαθορισμένη λίστα καταλόγων στους οποίους ψάχνουν για αρχεία
 - “**Μονοπάτι αναζήτησης**”
 - Για μονοπάτι αναζήτησης { A, B, C } το κέλυφος θα ψάξει πρώτα στον κατάλογο A, μετά στον B, μετά στον C
 - Μπορούμε να παρακάμψουμε τη σειρά αναζήτησης δίνοντας συγκεκριμένο κατάλογο έναρξης
 - “./my_program”
- Μας θυμίζει τίποτα;
 - **Τοπικότητα!!!**

Συνώνυμα: “Hard” & “Soft” Σύνδεσμοι



- Μπορούν να δείχνουν πολλοί κατάλογοι στο ίδιο αρχείο (**Hard links**)
 - Το Unix αποθηκεύει τον αριθμό των δεικτών (“hard links”) στον inode
 - Το “**ln** damn oslab” φτιάχνει ένα συνώνυμο (‘oslab’) για το ‘damn’
- **Soft links**:
 - Δείκτης σε ένα αρχείο/κατάλογο, αλλά το αντικείμενο μπορεί οποιαδήποτε στιγμή να σβηστεί (η να μην υπήρχε εξ αρχής)
 - Φτιάχνονται στο Unix όπως οι κατάλογοι: κανονικός αρχείο δείχνεται από το όνομα, με απλώς ένα ειδικό “symlink” bit
 - Όταν το ΣΑ συναντήσει συμβολικό σύνδεσμο (soft link) αυτόματα τον μεταφράζει (αν είναι δυνατό)
 - Εντολή “**ln -s** oslab oops”





Τα Περιεχόμενα του inode

- Από το προηγούμενο: «Ένας inode περιέχει δείκτες στα blocks του αρχείου και ορισμένα ‘άλλα πράγματα’»
 - Άλλα πράγματα:
 - Μέγεθος
 - Χρόνος αλλαγής / προσπέλασης
 - ID κατόχου / ομάδας,
 - Bits προστασίας ή ACLs
 - Π.χ. :
 - “ls -l test.txt”
-rw-r--r-- 1 cda faculty 203 May 5 2017 temp.txt
 - Παράδειγμα:
 - “ln test.txt my_test”
 - “ls -l my test”
-rw-r--r-- 1 cda faculty 203 May 5 2017 my_test