

Συστήματα Αρχείων: Διεπαφή και Υλοποίηση

Λειτουργικά Συστήματα

6ο εξάμηνο ΣΗΜΜΥ

ακ. έτος 2022-2023

<http://www.cslab.ece.ntua.gr/courses/os>



Εργαστήριο Υπολογιστικών Συστημάτων
ΕΜΠ

Μάιος 2023

Συστήματα Αρχείων (Σ.Α.)

σκελετός

- ▶ Διεπαφή
 - ▶ Αρχεία
 - ▶ Κατάλογοι
 - ▶ Μονοπάτια
 - ▶ Προσάρτηση ΣΑ
 - ▶ Απομακρυσμένα ΣΑ
- ▶ Ζητήματα Υλοποίησης
 - ▶ File Control Block (inode)
 - ▶ Μέθοδοι ανάθεσης
 - ▶ VFS
 - ▶ Άλλα ζητήματα (page cache, journaling, ...)

Διεπαφή ΣΑ

(από τη μεριά του χώρου χρήστη)

Απαιτήσεις χρήστη

- ▶ Διατήρηση δεδομένων μετά την εκτέλεση ενός προγράμματος και μετά το κλείσιμο του υπολογιστή (διεπαφή **αρχείου**)
- ▶ Οργάνωση αρχείων σε ομάδες για ευκολότερη επισκόπηση (διεπαφή **καταλόγου** ή **φακέλου**)
- ▶ Έλεγχος πρόσβασης στα αρχεία και τους καταλόγους (δικαιώματα πρόσβασης)
- ▶ Δυνατότητα χειρισμού αρχείων και καταλόγων μέσα από τα προγράμματα (κλήσεις συστήματος)

Αρχείο: Μόνιμος, συνεχής, λογικός χώρος διευθύνσεων.

- ▶ **Μόνιμος:** Παραμένει προσβάσιμο και μετά τον τερματισμό του προγράμματος (ή το κλείσιμο του υπολογιστή)
- ▶ **Λογικός:** Ξεχωριστό από την φυσική απεικόνισή στην συσκευή αποθήκευσης
- ▶ **Χώρος Διευθύνσεων:** Διευθυνσιοδότηση δεδομένων σε επίπεδο byte
- ▶ **Συνεχής:** Χωρίς κενά

"Everything is a file" – ρητό του Unix

Τύποι Αρχείων

Γενικά το **τι** δεδομένα περιέχει το αρχείο είναι θέμα του χρήστη (ή της εφαρμογής)

- ▶ Πρόγραμμα (executable)
- ▶ Κείμενο (text)
- ▶ Δυαδικά δεδομένα (binary)
- ▶ Απλές Δομές:
 - ▶ Γραμμές (πχ πρόγραμμα .c)
 - ▶ Πεδία σταθερού μεγέθους (πχ ακέραιοι)
 - ▶ Πεδία μεταβλητού μεγέθους (πχ συμβολοσειρές)
- ▶ Σύνθετες δομές (πχ αρχείο pdf)

Ιδιότητες Αρχείων

- ▶ Όνομα
- ▶ Αναγνωριστικό (εσωτερικό)
- ▶ Τύπος
- ▶ Θέση
- ▶ Μέγεθος
- ▶ Δικαιώματα (προστασία)
- ▶ Ώρα / Ημερομηνία πρόσβασης
- ▶ ...

Λειτουργίες Αρχείων

- ▶ Δημιουργία (create)
- ▶ Ανάγνωση (read)
- ▶ Εγγραφή (write)
- ▶ Επανατοποθέτηση (seek)
- ▶ Διαγραφή (delete)
- ▶ Μηδενισμός (truncate)
- ▶ Εκτέλεση (execute)
- ▶ Απεικόνιση στη μνήμη (mmap)

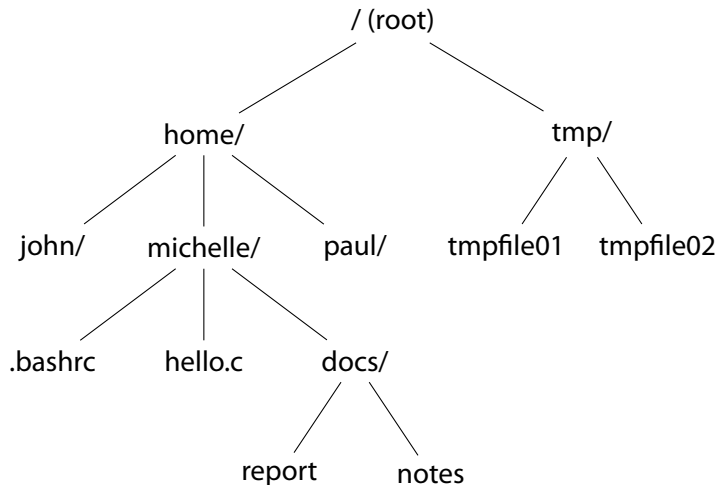
Κατάλογοι (φάκελοι)

- ▶ Κατάλογος ενός επιπέδου
- ▶ Κατάλογος δύο επιπέδων
- ▶ Κατάλογοι δενδρικής δομής

Κόμβοι Ιεραρχίας – Σύνολα κόμβων που μπορούν να είναι:

- ▶ Αρχεία
- ▶ Κατάλογοι

Παράδειγμα Δενδρικής Δομής



Λειτουργίες Καταλόγων

- ▶ Αναζήτηση αρχείου (με βάση το όνομα)
- ▶ Δημιουργία αρχείου
- ▶ Διαγραφή αρχείου
- ▶ Μετονομασία αρχείου
- ▶ Διάσχιση ιεραρχίας

Μονοπάτια

(Σε Σ.Α. τύπου Unix)

Μονοπάτι (path):

Συμβολοσειρα από αναγνωριστικά χωρισμένα από τον χαρακτήρα /

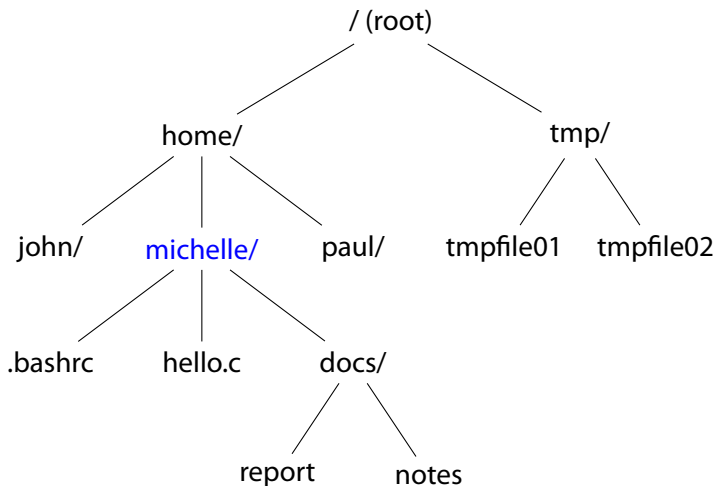
πχ: /this/is/a/path/name

Κανόνες:

- ▶ Το μονοπάτι είναι
 1. **απόλυτο** αν ξεκινάει με / – αφετηρία είναι η αρχή της ιεραρχίας
 2. **σχετικό** (αν όχι) – αφετηρία είναι ο τρέχων κατάλογος (TK)
- ▶ Το αναγνωριστικό:
 - . σηματοδοτεί τον TK
 - .. σηματοδοτεί τον πατέρα του TK

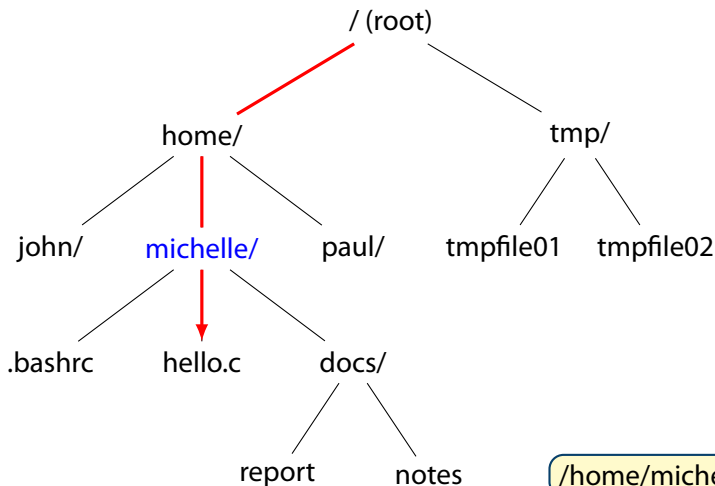
Παραδείγματα μονοπατιών

TK: /home/michelle



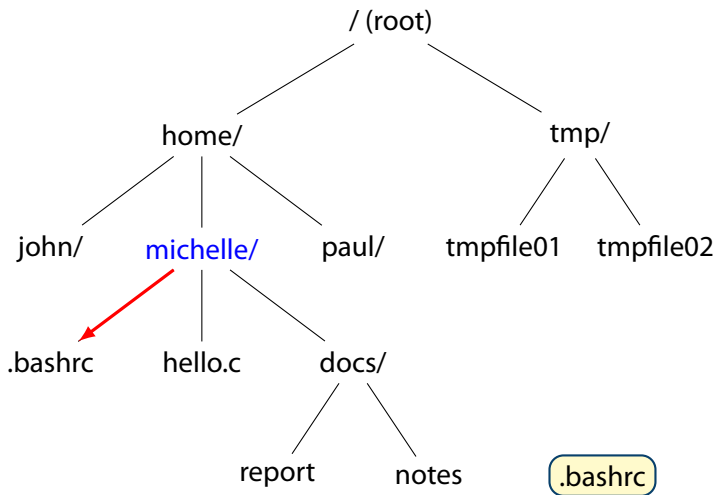
Παραδείγματα μονοπατιών

TK: /home/michelle



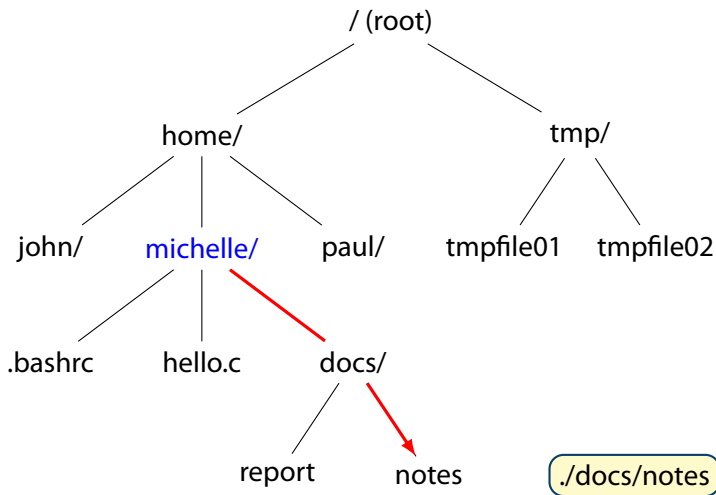
Παραδείγματα μονοπατιών

TK: /home/michelle



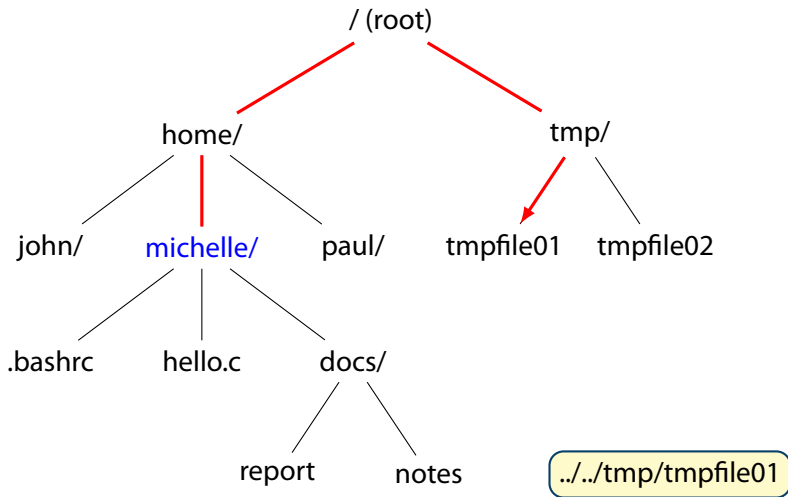
Παραδείγματα μονοπατιών

TK: /home/michelle



Παραδείγματα μονοπατιών

TK: /home/michelle



Προστασία

- ▶ Κατηγορίες:
 - ▶ Χρήστης
 - ▶ Ομάδες
 - ▶ Άλλοι
- ▶ Δικαιώματα:
 - ▶ Ανάγνωση
 - ▶ Εγγραφή
 - ▶ Εκτέλεση
 - ▶ Προσθήκη
 - ▶ Διαγραφή
 - ▶ Λίστα (για καταλόγους)
- ▶ Λίστα Ελέγχου πρόσβασης (ACL)

Σύστημα αρχείων από την οπτική του χρήστη

σύνοψη

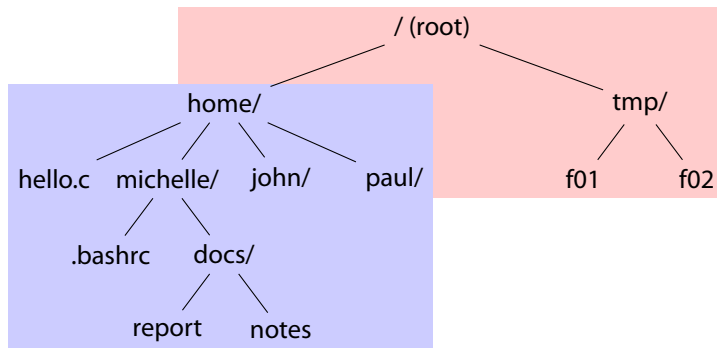
- ▶ Δυνατότητα πρόσβασης σε αρχεία
- ▶ Οργάνωση των αρχείων σε καταλόγους
- ▶ Δικαιώματα πρόσβασης
- ▶ Κλήσεις συστήματος για το χειρισμό αρχείων και καταλόγων

Προσάρτηση ΣΑ

(mount)

Χρειάζονται:

- ▶ Σημείο προσάρτησης (mountpoint)
- ▶ Συσκευή αποθήκευσης (σκληρός δίσκος, flash)



Απομακρυσμένα ΣΑ

remote

- ▶ FTP / WWW
- ▶ Δικτυακά ΣΑ (Network FS)
(πχ NFS, CIFS, AFS)
 - ▶ Οι πόροι βρίσκονται σε απομακρυσμένο υπολογιστή
 - ▶ Πρόσβαση σε αυτούς μέσω δικτύου (πχ TCP/IP)
 - ▶ Πελάτης-Εξυπηρετητής (client-server)
- ▶ Κατανεμημένα ΣΑ (Distributed FS)
(πχ Lustre)
 - ▶ Οι πόροι βρίσκονται σε πολλαπλούς υπολογιστές
 - ▶ Στόχοι:
 - ▶ Προστασία – πλεονασμός πληροφορίας (data redundancy)
 - ▶ Επίδοση – παράλληλη πρόσβαση
 - ▶ Χειρισμός σφαλμάτων (πχ δικτυακό πρόβλημα)

Υλοποίηση ΣΑ

(από τη μεριά του πυρήνα)

Τι χρειάζεται για να υλοποιήσουμε ένα σύστημα αρχείων

- Σχεδιασμός και υλοποίηση αποθήκευσης δεδομένων (αρχείων και καταλόγων) σε μία **συσκευή αποθήκευσης** (π.χ. σκληρός δίσκος)
 - Πραγματικά δεδομένα αρχείων/καταλόγων
 - Μεταδεδομένα αρχείων
 - Ευρετήριο αρχείων και καταλόγων
 - ...
- Δομές δεδομένων και κώδικας (κλήσεις συστήματος) για το χειρισμό των αρχείων και καταλόγων
 - Ανοιχτά αρχεία
 - Αρχεία ανά διεργασία και τρέχουσα θέση ανά αρχείο
 - Τρέχων κατάλογος διεργασίας
 - ...

Συστήματα Αρχείων

Πολλές υλοποιήσεις:

- ▶ UFS
- ▶ FAT16, FAT32
- ▶ NTFS
- ▶ EXT2, EXT3, EXT4
- ▶ ZFS
- ▶ BTRFS
- ▶ XFS, ReiserFS
- ▶ NFS, AFS
- ▶ ...

Συσκευές αποθήκευσης

- Μόνιμα δεδομένα (persistent) (σε αντίθεση με μνήμη)
- Σκληροί Δίσκοι
 - Αργή πρόσβαση
 - Χρόνος αναζήτησης (seek time)
 - Solid State Disks (SSDs)
- Γραμμικός χώρος
 - Προσπέλαση βάσει τμημάτων (blocks)
 - Όχι ιδανικός για χρήστη/εφαρμογές
 - Συστήματα Αρχείων (Ιεραρχική Δομή)
 - Βάσεις Δεδομένων (SQL)

Πληροφορίες ΣΑ στη συσκευή αποθήκευσης

- ▶ Τμήμα ελέγχου εκκίνησης (boot control block)
- ▶ Τμήμα ελέγχου τόμου (volume control block)
- ▶ Δομή ευρετηρίου – ιεραρχική δομή
- ▶ Τμήμα ελέγχου αρχείου (File Control Block – FCB) – ένα ανά αρχείο
- ▶ Πληροφορίες ελεύθερου χώρου

- ▶ **Δεδομένα**
περιεχόμενα των αρχείων
- ▶ **Μέτα-δεδομένα**
δομή αρχείων, περιεχόμενα καταλόγων, ιδιότητες, κλπ ...
(ο,τι δεν είναι Δεδομένα)

File Control Block (FCB)

Πληροφορία ανά αρχείο

- ▶ Δικαιώματα
- ▶ Ημερομηνίες
- ▶ Χρήστης, Ομάδα, Λίστες πρόσβασης
- ▶ Μέγεθος
- ▶ Δεδομένα ή Τοποθεσία δεδομένων

Ανοιχτά αρχεία

Πρόσβαση από εφαρμογή χώρου χρήστη:

- ▶ Άνοιγμα (open)
- ▶ Εγγραφή / Ανάγνωση / Επανατοποθέτηση
(επηρεάζουν την τρέχουσα θέση)
- ▶ Κλείσιμο (close)

Πληροφορίες που χρειάζεται να διατηρεί το ΛΣ:

- ▶ Πίνακας ανοιχτών αρχείων
(δομή περιγραφής αρχείου → File Control Block (FCB))
- ▶ Ανοιχτά αρχεία ανά διεργασία και δικαιώματα πρόσβασης
- ▶ Τρέχουσα θέση ανά αρχείο

Κατάλογοι

Παράδειγμα: UNIX

- ▶ Οργανώνουν την παρουσίαση των αρχείων στο χρήστη (όχι την πραγματική θέση των αρχείων)
- ▶ Είναι κανονικά αρχεία
- ▶ Περιλαμβάνουν το όνομα του αρχείου και το αναγνωριστικό του
 - ▶ Για κάθε αρχείο και κατάλογο που βρίσκεται στον κατάλογο
 - ▶ Για τον τρέχοντα κατάλογο
 - ▶ Για τον γονεϊκό κατάλογο
- ▶ Το ΛΣ τηρεί για κάθε διεργασία τον τρέχοντα κατάλογό της

Μέθοδοι Ανάθεσης – allocation methods

πώς τοποθετούνται τα δεδομένα ενός αρχείου στο δίσκο;

Επιθυμητά χαρακτηριστικά λύσης:

- ▶ Ευέλικτη χρήση του χώρου αποθήκευσης
- ▶ Υποστήριξη σειριακής (serial) και τυχαίας (random) προσπέλασης
- ▶ Αποφυγή σπατάλης του αποθηκευτικού χώρου για μεταδεδομένα

Μέθοδοι:

- ▶ Συνεχόμενη ανάθεση (contiguous allocation)
- ▶ Συνδεδεμένη ανάθεση (linked allocation)
- ▶ Δεικτοδοτημένη ανάθεση (indexed allocation)

Συνεχόμενη ανάθεση

(contiguous allocation)

- ▶ Τα δεδομένα των αρχείων τοποθετούνται συνεχόμενα
- ▶ Αρχικό τμήμα και μέγεθος
- ▶ Σειριακή και τυχαία προσπέλαση
- ▶ Δύσκολη η ανάθεση χώρου
- ▶ Δύσκολη η επέκταση των αρχείων
- ▶ Εξωτερικός κατακερματισμός

- ▶ Γενικά δεν χρησιμοποιείται
- ▶ Επεκτάσεις (extents) – NTFS, XFS, ext4, btrfs

Συνεχόμενη ανάθεση

παράδειγμα

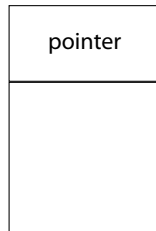
Αρχείο	αρχή	μέγεθος
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Συνδεδεμένη ανάθεση

linked allocation

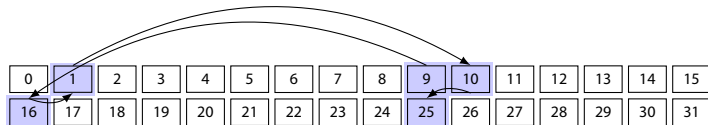
- ▶ Κάθε αρχείο είναι μια λίστα απο τμήματα (blocks).
- ▶ Το κάθε τμήμα περιλαμβάνει δείκτη στο επόμενο.
- ▶ Απλό, όχι σπατάλη χώρου
- ▶ Όχι άμεση πρόσβαση (random access)
- ▶ Αναζήτηση δίσκου (seek)
- ▶ χώρος για κάθε δείκτη
- ▶ Πίνακας ανάθεσης αρχείου (File Allocation Table – FAT)



Συνδεδεμένη ανάθεση

παράδειγμα

Αρχείο	αρχή	τέλος	block	pointer
pizza	9	25	9	16
			16	1
			1	10
			10	25
			25	-1



Δεικτοδοτημένη ανάθεση

Όλοι οι δείκτες του αρχείου σε μια θέση (index block).

Σχήματα:

- ▶ Συνδεδεμένο σχήμα
- ▶ Πολυεπίπεδο ευρετήριο
- ▶ UFS: Συνδυασμένο σχήμα
 - ▶ 12 άμεσα τμήματα
 - ▶ 3 έμμεσα τμήματα:
 - ▶ απλό (single indirect block)
 - ▶ διπλό (double indirect block)
 - ▶ τριπλό (triple indirect block)

Δεικτοδοτημένη ανάθεση

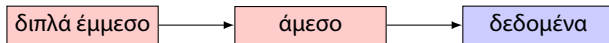
Παράδειγμα

		index block
		9
		16
Αρχείο	index block	1
pizza	19	10
		25
		-1

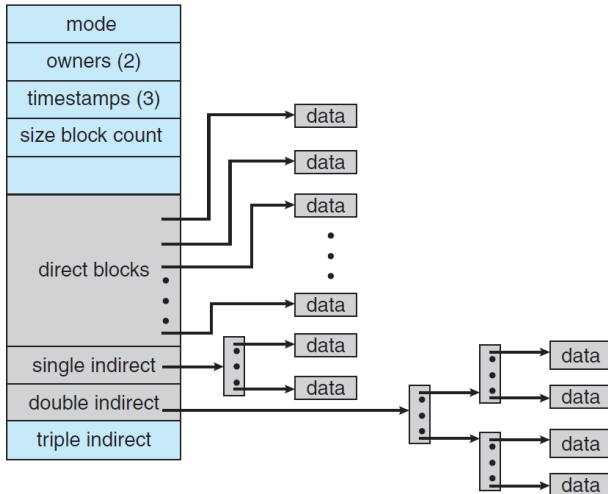
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Έμμεσα τμήματα

indirect blocks



To FCB του UNIX (inode)



Υλοποίηση ευρετηρίου

στη δευτερεύουσα συσκευή αποθήκευσης

- ▶ Το ευρετήριο αντιστοιχίζει αρχείο με FCB.
- ▶ Δομή αναζήτησης με κλειδί (key) το αναγνωστικό του αρχείου και τιμή (value) τη θέση στο δίσκο που βρίσκεται το FCB του αρχείου

Κατάλληλες δομές για ευρετήριο:

- ▶ Γραμμική λίστα
 - ▶ Γραμμική αναζήτηση
- ▶ Πίνακας κατακερματισμού
 - ▶ Χώρος
 - ▶ Συγκρούσεις
- ▶ B-trees ή παραλλαγές (πχ Btrfs, ReiserFS)
- ▶ Οι πληροφορίες αυτές αντιγράφονται στη μνήμη από το ΛΣ για καλύτερη επίδοση

Σύνδεσμοι

πρόσβαση σε αρχείο από διαφορετικά σημεία του ΣΑ

- ▶ στο αρχείο (hard links)
 - ▶ Ίδιο αρχείο (inode) με πολλαπλά ονόματα
 - ▶ Τηρείται reference count για κάθε αρχείο
 - ▶ Διαγραφή hard link: μείωση του reference count
 - ▶ Αν reference count = 0 διαγραφή δεδομένων αρχείου
 - ▶ Αν αλλάξει όνομα ή θέση το αρχείο, ακόμα προσβάσιμο
 - ▶ Δεν μπορεί να συνδέσει αρχείο εκτός του συγκεκριμένου ΣΑ
- ▶ στο όνομα (soft ή symbolic links)
 - ▶ Επίλυση συνδέσμου με βάση το όνομα (το link δείχνει σε όνομα όχι σε inode)
 - ▶ Αν σβήσουμε το soft link το αρχείο δεν επηρεάζεται
 - ▶ Αν σβηστεί ή αλλάξει το αρχείο, το link δείχνει "στο πουθενά"
 - ▶ Μπορεί να συνδέσει αρχείο εκτός του συγκεκριμένου ΣΑ

man ln

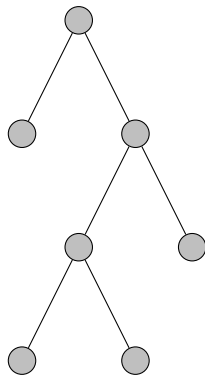
Σύνδεσμοι

Οδηγούν σε δομή γράφου:

- ▶ Ακυκλικός γράφος
- ▶ Γενικός γράφος (με κύκλους):
 - ▶ Κύκλοι οδηγούν σε προβλήματα
 - ▶ Πολλά ΛΣ αποτρέπουν τη δημιουργία συνδέσμων αρχείων (hard links) σε καταλόγους
 - ▶ Γενικά τα ΛΣ περιορίζουν τον αριθμό διάσχισης συμβολικών συνδέσμων (ELOOP).

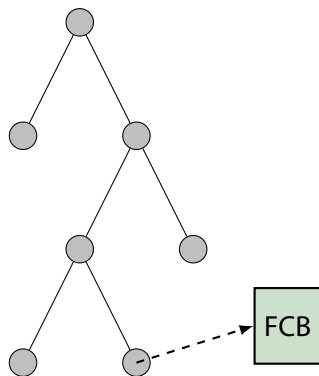
Σύνδεσμοι

► Ιεραρχία



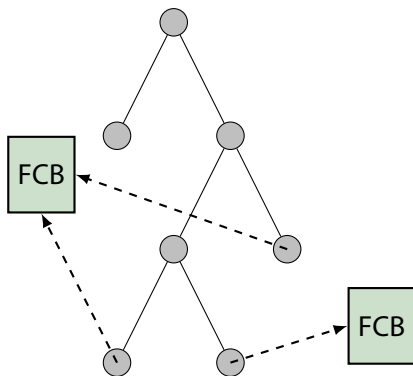
Σύνδεσμοι

- ▶ Ιεραρχία
- ▶ FCBs (inodes)



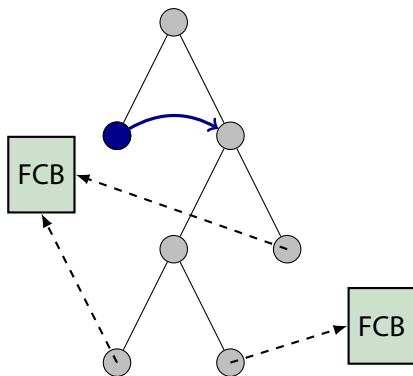
Σύνδεσμοι

- ▶ Ιεραρχία
- ▶ FCBs (inodes)
- ▶ hard links



Σύνδεσμοι

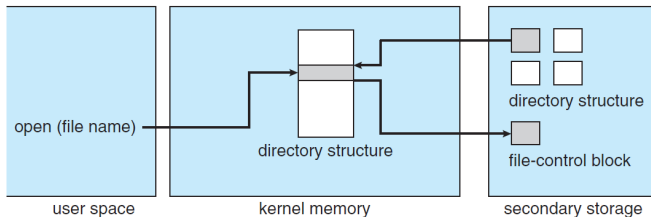
- ▶ Ιεραρχία
- ▶ FCBs (inodes)
- ▶ hard links
- ▶ soft links



Λειτουργίες ΣΑ

open

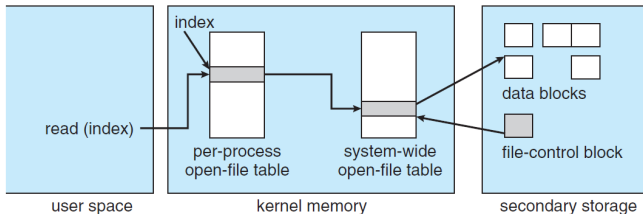
1. Αντιστοίχιση ονόματος με αναγνωριστικό αρχείου από κατάλογο
2. Έυρεση FCB (inode) με τη βοήθεια του ευρετηρίου (directory structure στο παρακάτω σχήμα)
3. Έλεγχος δικαιωμάτων πρόσβασης
4. Ενημέρωση δομών του ΛΣ (ανοιχτά αρχεία, αρχεία ανά διεργασία)
5. Επιστροφή δείκτη στις παραπάνω δομές (file descriptor)



Λειτουργίες ΣΑ

read

1. Ανάγνωση δομών ΛΣ για τα ανοιχτά αρχεία και εύρεση FCB
2. Ανάγνωση δεδομένων
3. Ενημέρωση του δείκτη αρχείου για τη διεργασία



Λειτουργίες ΣΑ

ερωτήσεις

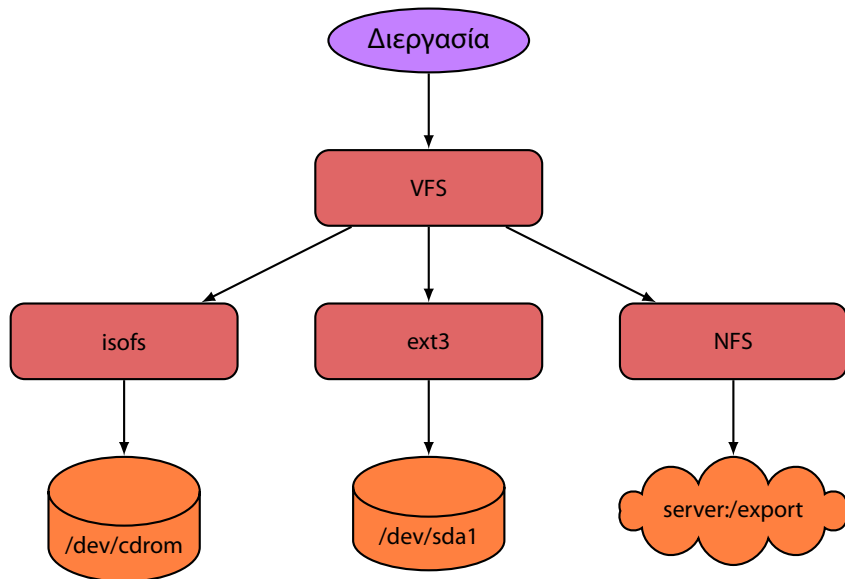
- ▶ Πώς διασχίζεται ένα σχετικό και ένα απόλυτο μονοπάτι;
- ▶ Τί συμβαίνει όταν μεταφέρουμε ένα αρχείο σε άλλο καταλογο;
- ▶ Τί περιλαμβάνει η δημιουργία ενός νέου αρχείου;

Διαχείριση ελεύθερου χώρου

- ▶ Διάνυσμα δυαδικών ψηφίων (bitvector / bitmap)
 - 0 Το τμήμα χρησιμοποιείται
 - 1 Το τμήμα είναι ελεύθερο
- ▶ Συνδεδεμένη λίστα
 - ▶ Ομαδοποίηση (πολλαπλοί δείκτες)
 - ▶ Καταγραφή πλήθους (αρχή, μέγεθος)

Εικονικό Σύστημα Αρχείων

Virtual Filesystem – VFS

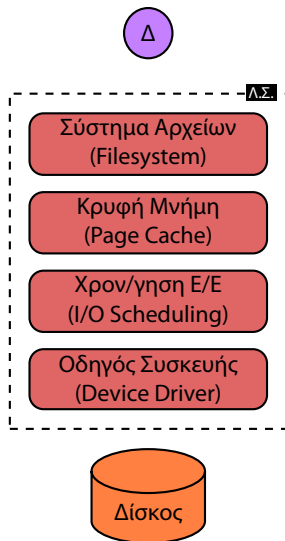


Δομές Linux VFS

- ▶ `struct inode`: Δομή που περιγράφει ένα αρχείο (FCB).
- ▶ `struct file`: Δομή που περιγράφει ένα ανοιχτό αρχείο.
- ▶ `struct super_block`: Δομή που περιγράφει ένα ΣΑ.
- ▶ `struct dentry`: Δομή που περιγράφει μια θέση στην ιεραρχία των ΣΑ.
- ▶ `struct file_operations`: Διαδικασίες για υλοποίηση λειτουργιών σε αρχεία (πχ `read`, `write`). Χρησιμοποιούνται για την υλοποίηση διαφορετικών ΣΑ.

ΛΣ και Συσκευές Αποθήκευσης

- Συστήματα Αρχείων
 - Ιεραρχική δομή πάνω από γραμμικό χώρο (συσκευή)
- Κρυφή Μνήμη
 - Η πρόσβαση στο δίσκο είναι αργή
 - Περιοχές αρχείων στη μνήμη
- Χρονοδρομολόγηση E/E
 - Μεγάλος χρόνος αναζήτησης (seek)
 - Βελτιστοποίηση E/E αιτήσεων
- Οδηγός συσκευής
 - Επικοινωνία με συσκευή
 - Εγγραφή τμημάτων (blocks)



Κρυφή μνήμη

Λειτουργίες:

- ▶ Τυπική E/E (πχ `read()/write()`)
- ▶ E/E που αντιστοιχίζεται στη μνήμη (πχ `mmap()`)

Κρυφές μνήμες

- ▶ `buffer cache` → Τυπική E/E
- ▶ `page cache` → E/E μνήμης
 - ▶ Χρησιμοποιεί `buffer cache`
- ▶ Ενοποιημένη κρυφή μνήμη (`page cache`) (Linux)

ΣΑ με αρχεία καταγραφής

journaled FS

- ▶ Ασύγχρονη λειτουργία, κρυφές μνήμες
- ▶ Προβλήματα από αναπάντεχο τερματισμό λειτουργίας
- ▶ Ανάνηψη, συνέπεια δεδομένων στον δίσκο
- ▶ Ημερολόγιο αλλαγών (journal)
- ▶ Πραγματοποίηση Αλλαγών
 - ▶ Καταγραφή αλλαγών στο journal
 - ▶ Πραγματοποίηση αλλαγών στις δομές του δίσκου
- ▶ ext3, ...