

Συστήματα Αρχείων: Διεπαφή και Υλοποίηση

Λειτουργικά Συστήματα

6ο εξάμηνο ΣΗΜΜΥ

ακ. έτος 2019-2020

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



Εργαστήριο Υπολογιστικών Συστημάτων
ΕΜΠ

Ιούνιος 2020

Συστήματα Αρχείων (Σ.Α.)

σκελετός

- ▶ Διεπαφή
 - ▶ Αρχεία
 - ▶ Κατάλογοι
 - ▶ Μονοπάτια
 - ▶ Προσάρτηση ΣΑ
 - ▶ Απομακρυσμένα ΣΑ
- ▶ Ζητήματα Υλοποίησης
 - ▶ VFS
 - ▶ Ανάθεση
 - ▶ Ζητήματα (page cache, journaling, ...)

Διεπαφή ΣΑ

(από τη μεριά του χώρου χρήστη)

Συσκευές αποθήκευσης

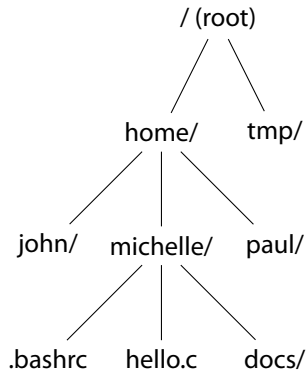
- Μόνιμα δεδομένα (persistent) (σε αντίθεση με μνήμη)
- Σκληροί Δίσκοι
 - Αργή πρόσβαση
 - Χρόνος αναζήτησης (seek time)
 - Solid State Disks (SSDs)
- Γραμμικός χώρος
 - Προσπέλαση βάση τμημάτων (blocks)
 - Όχι ιδανικός για χρήστη/εφαρμογές
 - Συστήματα Αρχείων (Ιεραρχική Δομή)
 - Βάσεις Δεδομένων (SQL)

Συστήματα αρχείων

Παράδειγμα

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)

Παράδειγμα:

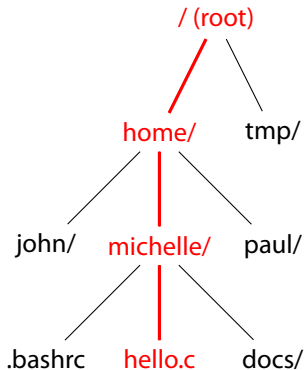


Συστήματα αρχείων

Παράδειγμα

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)

Παράδειγμα:



```
/home/michelle/hello.c
```

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello World!\n");
    return 0;
}
```

Αρχείο: Μόνιμος, συνεχής, λογικός χώρος διευθύνσεων.

- ▶ *Μόνιμος:* Παραμένει προβάσιμο 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)
- ▶ Απεικόνιση στη μνήμη (mmap)

Ανοιχτά Αρχεία

Πρόβαση από εφαρμογή χώρου χρήστη:

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

Πληροφορίες που χρειάζεται να διατηρεί το ΛΣ:

Ανοιχτά Αρχεία

Πρόβαση 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))
- ▶ FCB:
 - ▶ Αρχείο στο Σ.Α. (i-node στο UNIX)
 - ▶ τύπος αρχείου, μέγεθος αρχείου
 - ▶ δείκτες προς δεδομένα στο Σ.Α.
 - ▶ πληροφορίες ιδιοκτησίας, δικαιωμάτων
 - ▶ ...

Μέθοδοι Πρόσβασης

πώς αποθηκεύονται δεδομένα στα αρχεία;

- ▶ Ακολουθιακή (sequential)
- ▶ Άμεση
(εγγραφές σταθερού μήκους)
- ▶ Έμμεση
(π.χ. ευρετήρια / δείκτες)

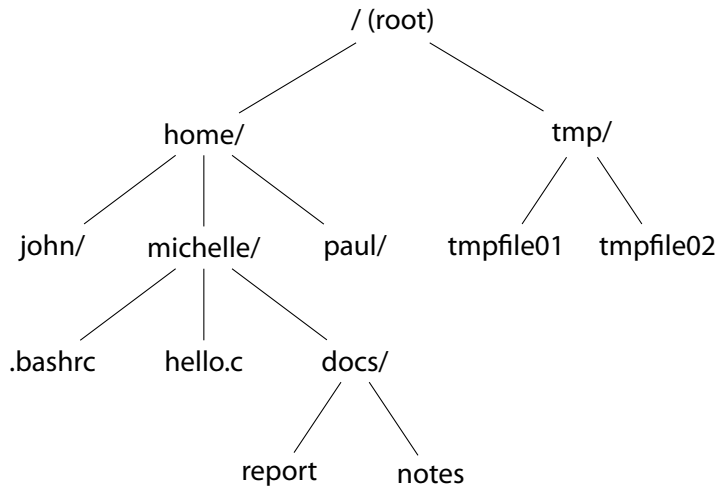
Κατάλογοι

- ▶ 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 στο Σ.Α.

(τύπου Unix)

Μονοπάτι (path):

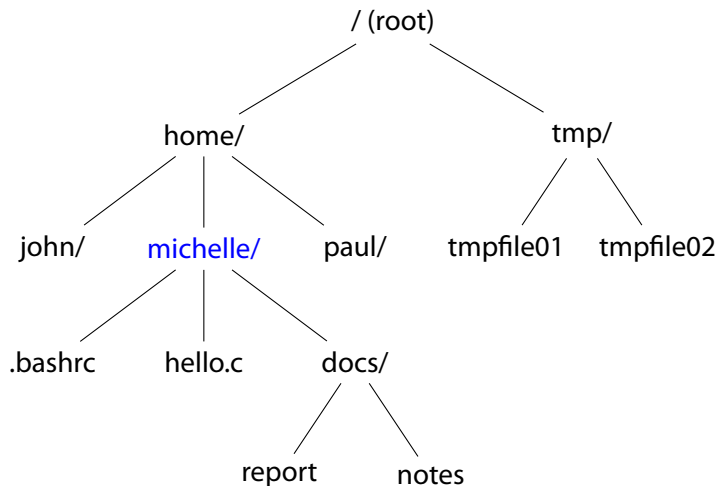
Συμβολοσειρα από αναγνωριστικά χωρισμένα από τον χαρακτήρα /
πχ: /this/is/a/path/name

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

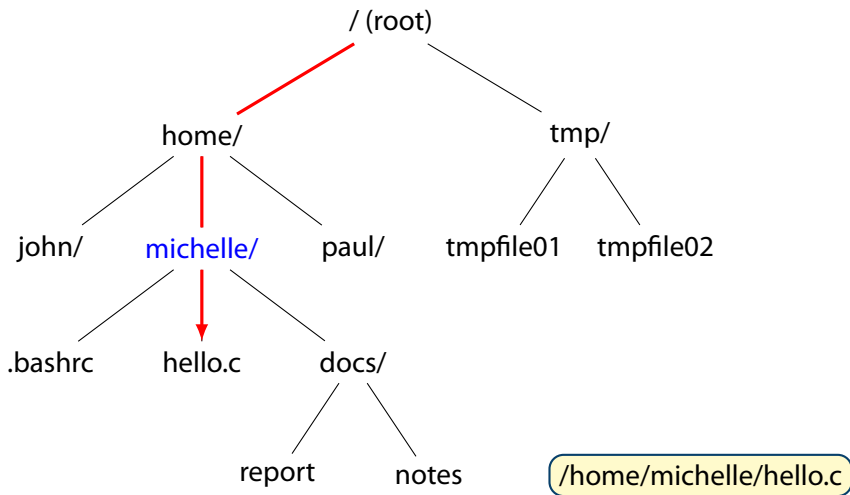
Παραδείγματα μονοπατιών

TK: /home/michelle



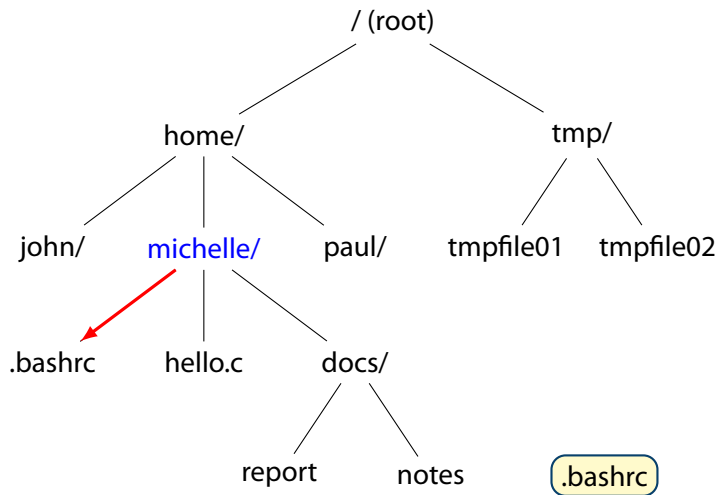
Παραδείγματα μονοπατιών

TK: /home/michelle



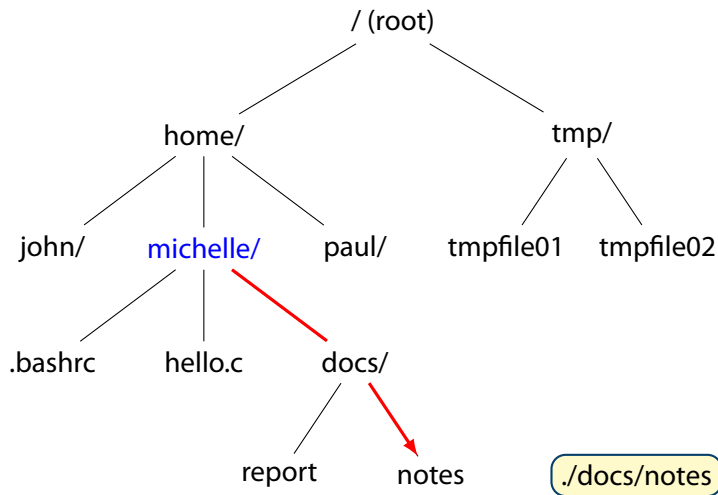
Παραδείγματα μονοπατιών

TK: /home/michelle



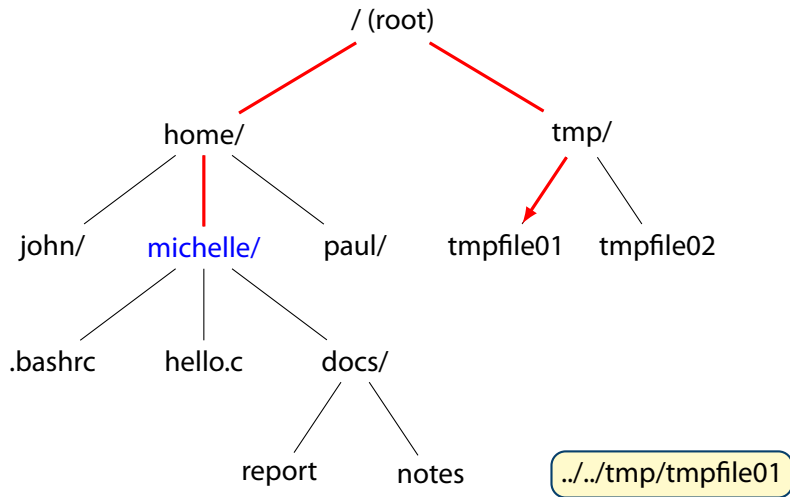
Παραδείγματα μονοπατιών

TK: /home/michelle



Παραδείγματα μονοπατιών

TK: /home/michelle



Σύνδεσμοι, δομή Γράφου

Σύνδεσμοι:

- ▶ στο αρχείο (hard links)
 - ▶ Ίδιο αρχείο (inode) με πολλαπλά ονόματα
 - ▶ reference count
- ▶ στο όνομα (soft links)
 - ▶ Επίλυση συνδέσμου με βάση το όνομα

Οδηγούν σε δομή γράφου:

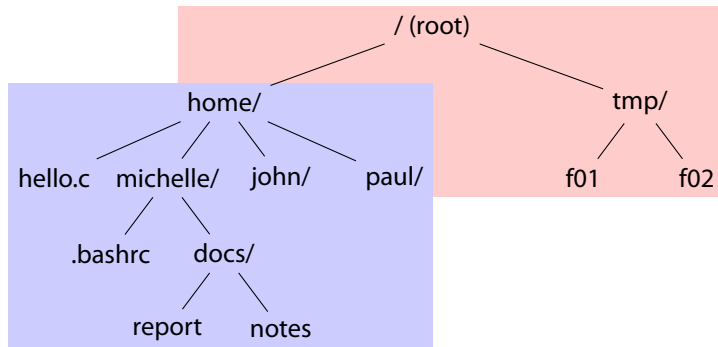
- ▶ Ακυκλικός γράφος
- ▶ Γενικός γράφος (με κύκλους):
 - ▶ Κύκλοι οδηγούν σε προβλήματα
 - ▶ Πολλά ΛΣ αποτρέπουν τη δημιουργία συνδέσμων αρχείων (hard links) σε καταλόγους
 - ▶ Γενικά τα ΛΣ περιορίζουν τον αριθμό διάσχισης συμβολικών συνδέσμων (ELOOP).

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)

Εφαρμογή Δικτυακών ΣΑ

Μηχανήματα εργαστηρίου:

- ▶ Χωρίς δίσκο
- ▶ Προσαρτήσεις:
 - ▶ κατάλογος ρίζας (/)
 - ▶ κατάλογος χρηστών (/home)
 - ▶ ...
- ▶ Κοινά αρχεία (de-duplication)

Προστασία

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

Υλοποίηση ΣΑ

(από τη μεριά του πυρήνα)

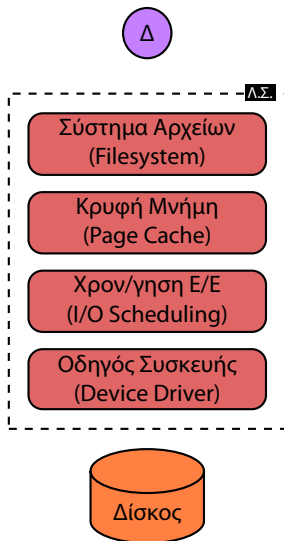
Συστήματα Αρχείων

Πολλές υλοποιήσεις:

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

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)



- ▶ **Δεδομένα**
περιεχόμενα των αρχείων

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

Πληροφορίες που διατηρεί ένα ΣΑ

- ▶ Τμήμα ελέγχου εκκίνησης (boot control block)
- ▶ Τμήμα ελέγχου τόμου (volume control block)
- ▶ Δομή καταλόγων – ιεραρχική δομή
- ▶ Τμήμα ελέγχου αρχείου (FCB) – ένα ανά αρχείο

Οι πληροφορίες:

- ▶ Υπάρχουν στη δευτερεύουσα συσκευή αποθήκευσης (δίσκος)
- ▶ Αποθηκεύονται και στη μνήμη για βελτίωση της ταχύτητας (caching)

Στη μνήμη αποθηκεύονται:

- ▶ Πληροφορίες για την προσάρτηση
- ▶ Οικουμενικός πίνακας ανοιχτών αρχείων
- ▶ Πίνακας ανοιχτών αρχείων διεργασίας

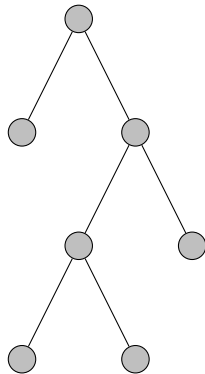
FCB

Πληροφορία ανά αρχείο

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

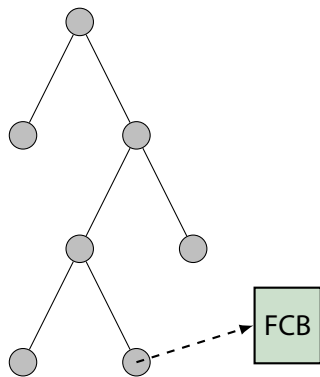
Αναπαράσταση ΣΑ

► Ιεραρχία



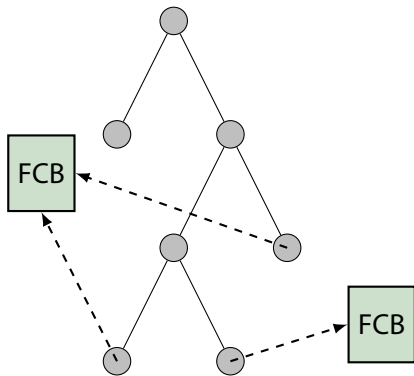
Αναπαράσταση ΣΑ

- ▶ Ιεραρχία
- ▶ FCBs (inodes)



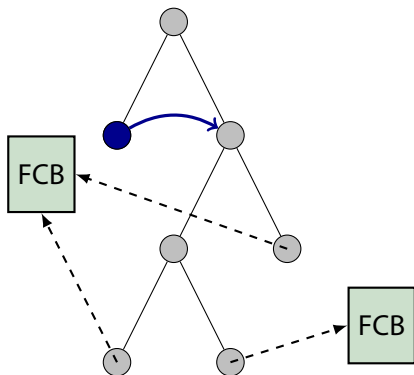
Αναπαράσταση ΣΑ

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

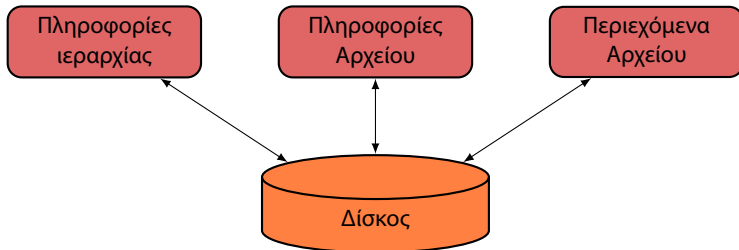
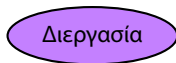


Αναπαράσταση ΣΑ

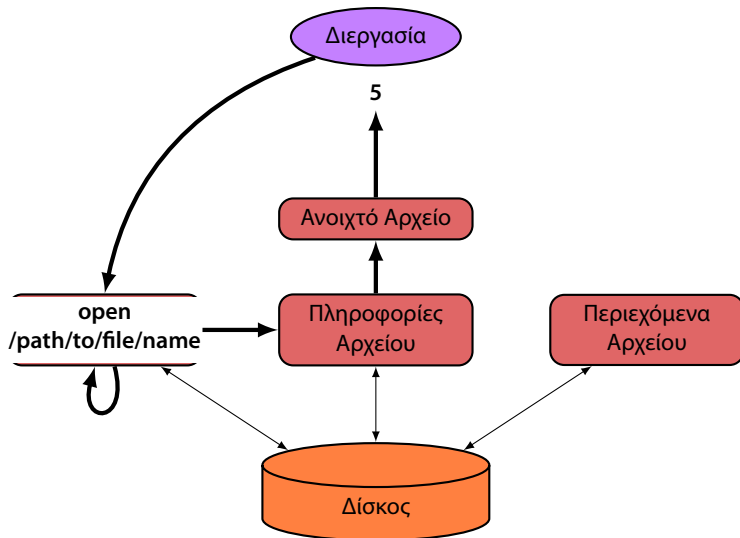
- ▶ Ιεραρχία
- ▶ FCBs (inodes)
- ▶ hard links
- ▶ soft links



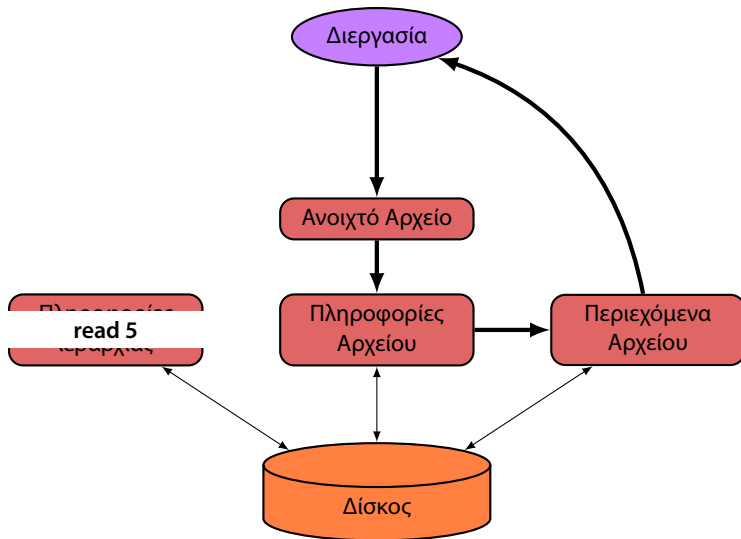
Λειτουργίες ΣΑ



Λειτουργίες ΣΑ

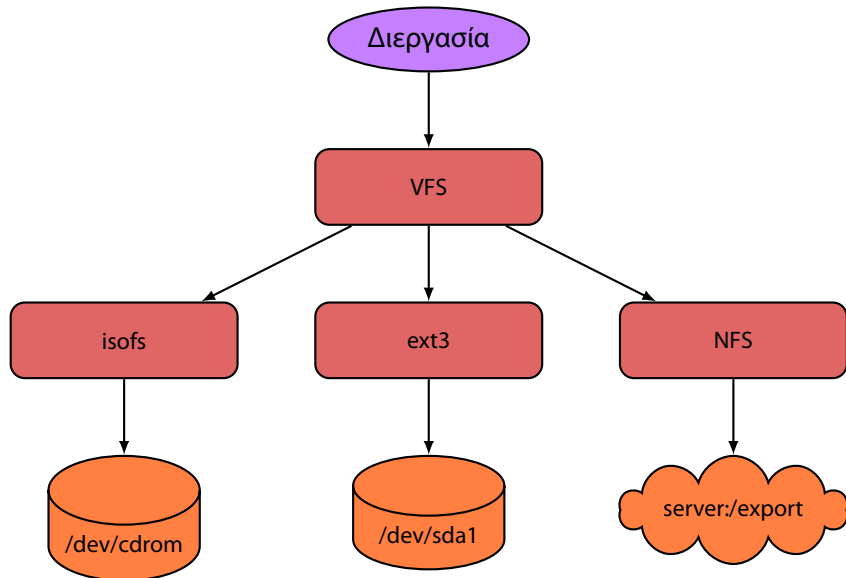


Λειτουργίες ΣΑ



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.

Υλοποίηση καταλόγου

στη δευτερεύουσα συσκευή αποθήκευσης

- ▶ 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

Μέθοδοι Ανάθεσης

(allocation methods)

(πχ ο χρήστης ζήτησε να γραφούν 4096 bytes σε ένα αρχείο)

Μέθοδοι:

- ▶ Συνεχόμενη ανάθεση (contiguous allocation)
- ▶ Συνδεδεμένη ανάθεση (linked allocation)
- ▶ Ανάθεση με ευρετήριο (indexed allocation)

Συνεχόμενη ανάθεση

(contiguous allocation)

- ▶ Τα δεδομένα των αρχείων τοποθετούνται συνεχόμενα
- ▶ Αρχικό τμήμα και μέγεθος
- ▶ Ακολουθιακή (sequential) και Άμεση (random) πρόσβαση (access)
- ▶ Δύσκολη η ανάθεση χώρου
- ▶ Δύσκολη η επέκταση των αρχείων
- ▶ Εξωτερικός κατακερματισμός

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

Συνεχόμενη ανάθεση

παράδειγμα

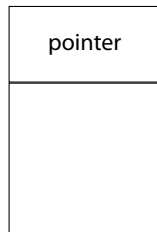
Αρχείο	αρχή	μέγεθος
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

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

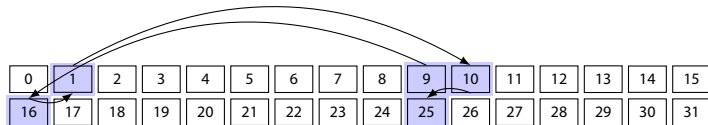
- ▶ Κάθε αρχείο είναι μια λίστα απο τμήματα (blocks).
- ▶ Το κάθε τμήμα περιλαμβάνει δείκτη στο επόμενο.
- ▶ Απλό, όχι σπατάλη χώρου
- ▶ Όχι άμεση πρόσβαση (random access)
- ▶ Αναζήτηση δίσκου (seek)
- ▶ χώρος για κάθε δείκτη
- ▶ Πίνακας ανάθεσης αρχείου (File Allocation Table – FAT)



Συνδεδεμένη ανάθεση

παράδειγμα

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



Ανάθεση με ευρετήριο

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)

Ανάθεση με ευρετήριο

Παράδειγμα

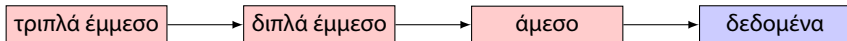
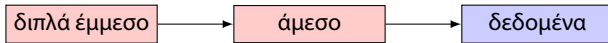
Αρχείο	index block
pizza	19

index block
9
16
1
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



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)

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 (ΔΕΗ)
- ▶ Ανάνηψη, συνέπεια δεδομένων στον δίσκο
- ▶ Ημερολόγιο αλλαγών (journal)
- ▶ Πραγματοποίηση Αλλαγών
 - ▶ Καταγραφή αλλαγών στο journal
 - ▶ Πραγματοποίηση αλλαγών στις δομές του δίσκου
- ▶ ext3, ...