



**Πανεπιστήμιο Πελοποννήσου**

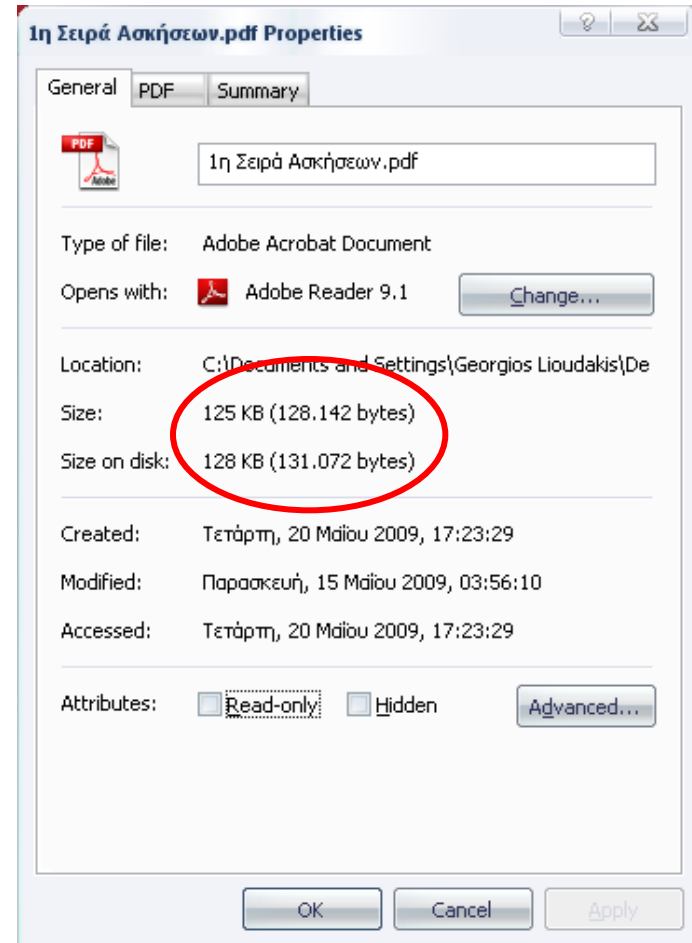
**Τμήμα Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών**

**Λειτουργικά Συστήματα – Προγραμματισμός Συστήματος**

**Συστήματα Αρχείων**

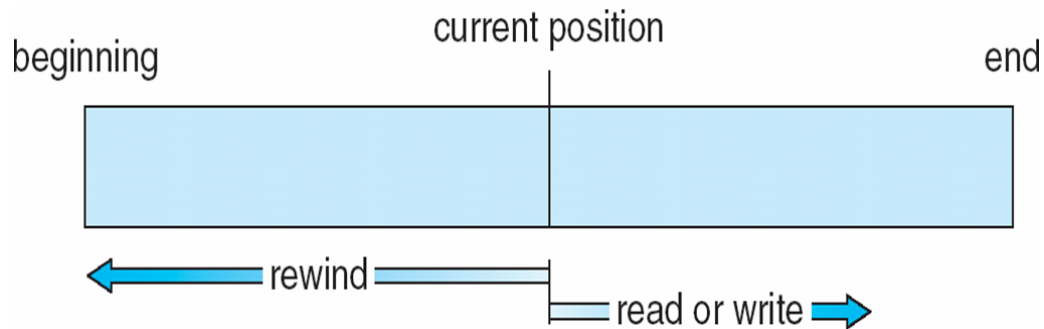
# Μπλοκ

- Τα αρχεία δεν αποθηκεύονται στο δίσκο “χύμα”, αλλά σε μπλοκ (block) προκαθορισμένου μεγέθους
- Κάθε αρχείο καταλαμβάνει ένα ή περισσότερα μπλοκ
- Δηλαδή, ο δίσκος μπορεί να θεωρηθεί μία γραμμική αλληλουχία από μπλοκ σταθερού μεγέθους



# Πρόσβαση σε Αρχεία

- Σειριακή πρόσβαση (sequential access)



- Αρχεία τυχαίας προσπέλασης (random access files)
  - π.χ., στη γλώσσα Java υπάρχει η κλάση `java.io.RandomAccessFile`



# Μεταδεδομένα Αρχείων

- Μεταδεδομένα (metadata) ή χαρακτηριστικά (attributes)

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to



# Λειτουργίες Αρχείων

- Δημιουργία (Create)
- Διαγραφή (Delete)
- Άνοιγμα (Open)
- Κλείσιμο (Close)
- Ανάγνωση (Read)
- Εγγραφή (Write)
- Προσάρτηση (Append)
- Αναζήτηση (Seek)
- Λήψη χαρακτηριστικών (Get attributes)
- Καθορισμός χαρακτηριστικών (Set attributes)
- Μετονομασία (Rename)



# Παράδειγμα [1]

```
#include <sys/types.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

#define BUF_SIZE 4096
#define OUTPUT_MODE 0700

int main(int argc, char *argv[])
{
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if(argc != 3) exit(1);

    in_fd = open(argv[1], O_RDONLY);
    if(in_fd < 0) exit(2);
    out_fd = creat(argv[2], OUTPUT_MODE);
    if (out_fd < 0) exit(3);
```



# Παράδειγμα [2]

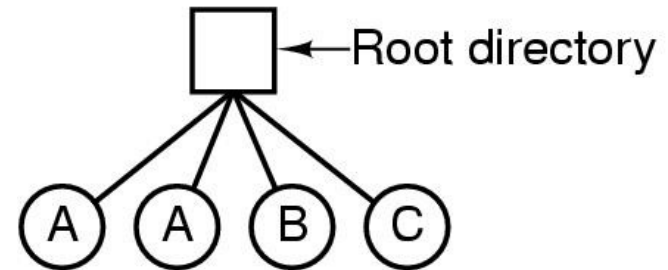
```
while(TRUE)
{
    rd_count = read(in_fd, buffer, BUF_SIZE);
    if(rd_count <= 0) break;
    wt_count = write(out_fd, buffer, rd_count);
    if (wt_count <= 0) exit(4);
}

close(in_fd);
close(out_fd);
if (rd_count == 0)
    exit(0);
else
    exit(5);
}
```

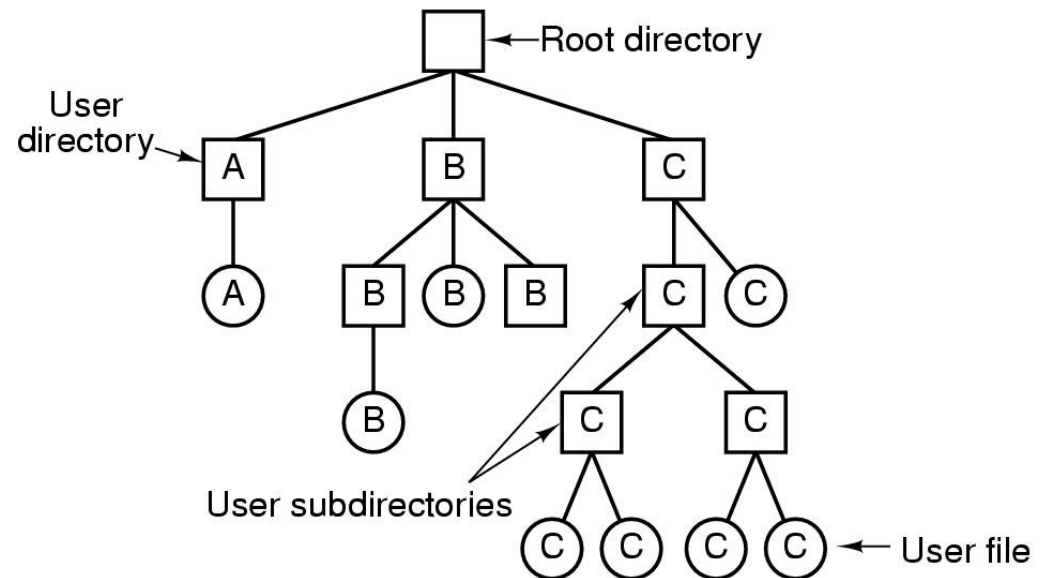


# Συστήματα Καταλόγων

- Συστήματα καταλόγων ενός επιπέδου

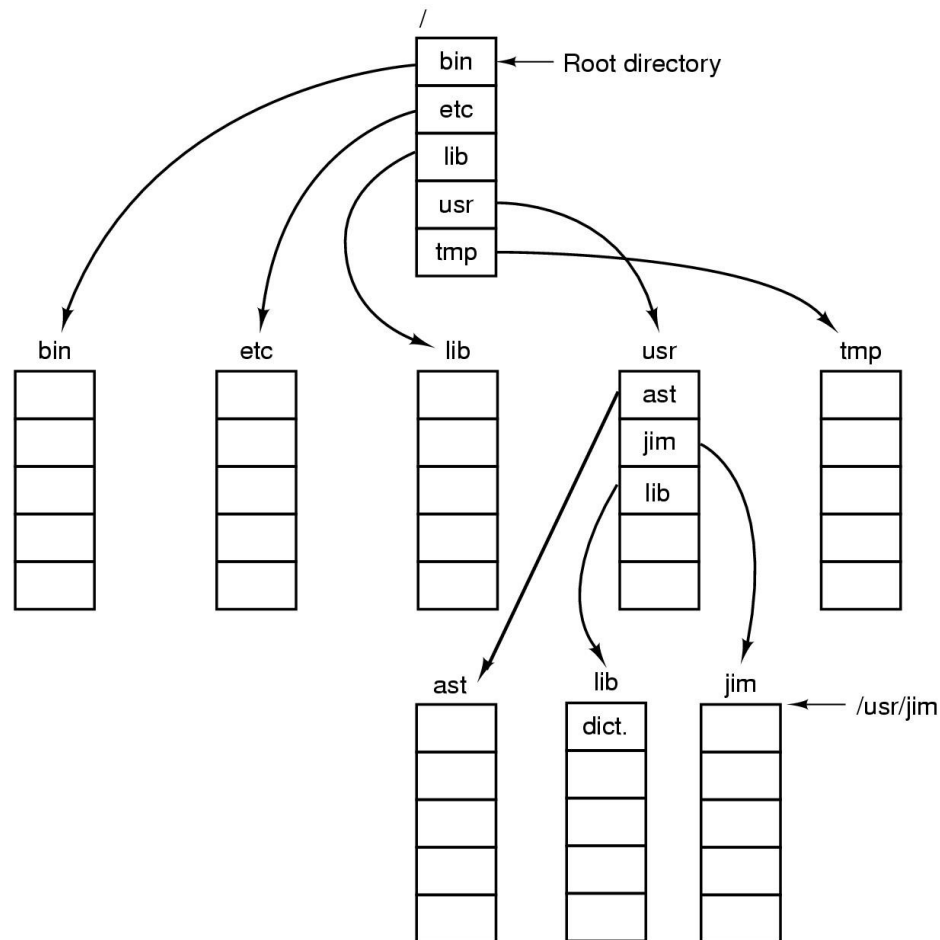


- Ιεραρχικά συστήματα καταλόγων





# Δέντρο Καταλόγων στο UNIX

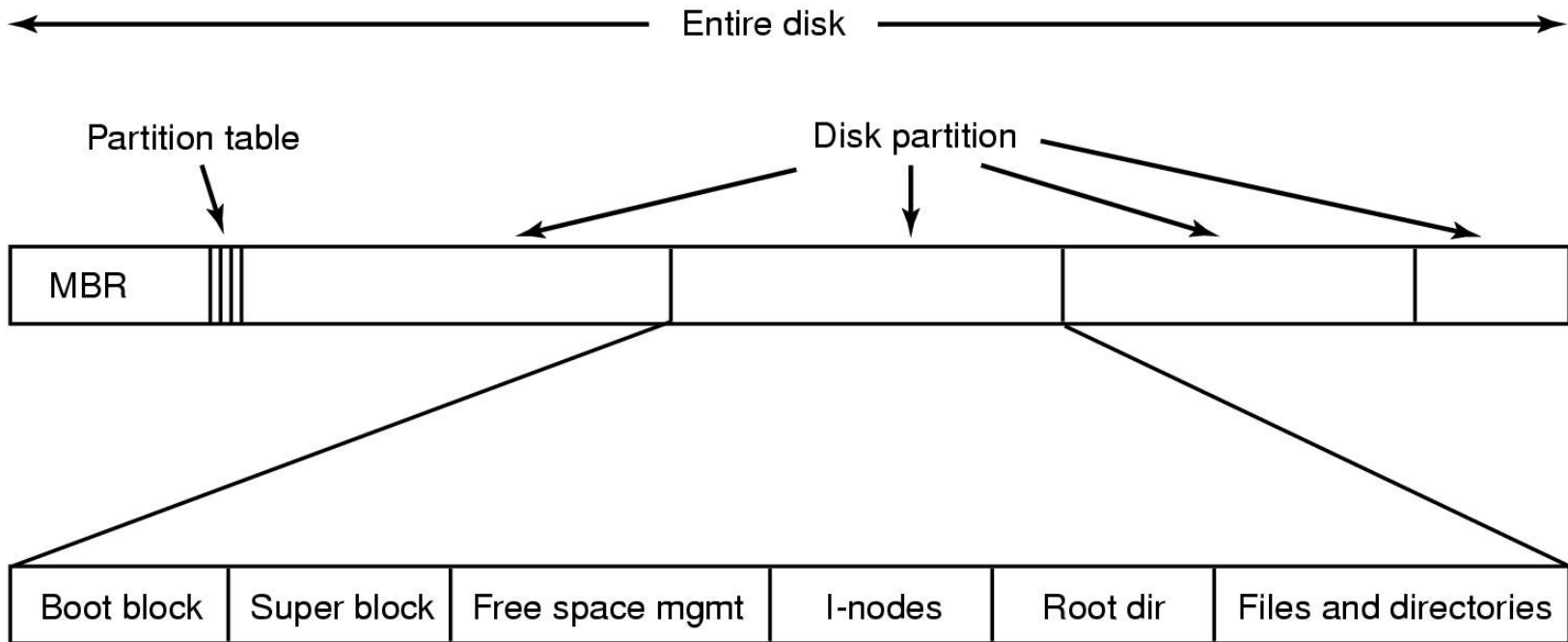


# Λειτουργίες Καταλόγων

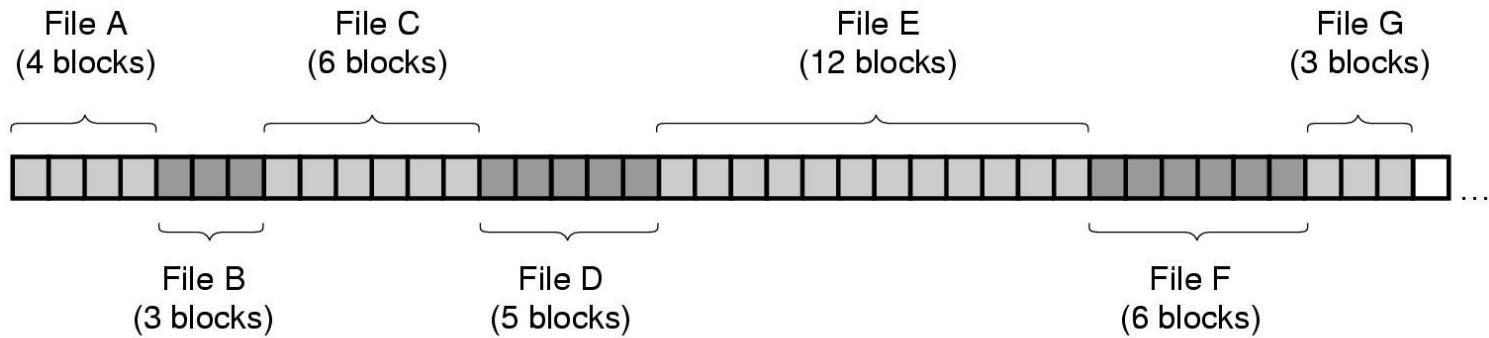
- Δημιουργία (Create)
- Διαγραφή (Delete)
- Άνοιγμα καταλόγου (Opendir)
- Κλείσιμο καταλόγου (Closedir)
- Ανάγνωση καταλόγου (Readdir)
- Μετονομασία (Rename)
- Σύνδεση (Link)
  - Συμβολική σύνδεση (symbolic link)
  - Σκληρή σύνδεση (hard link)
- Αποσύνδεση (Unlink)



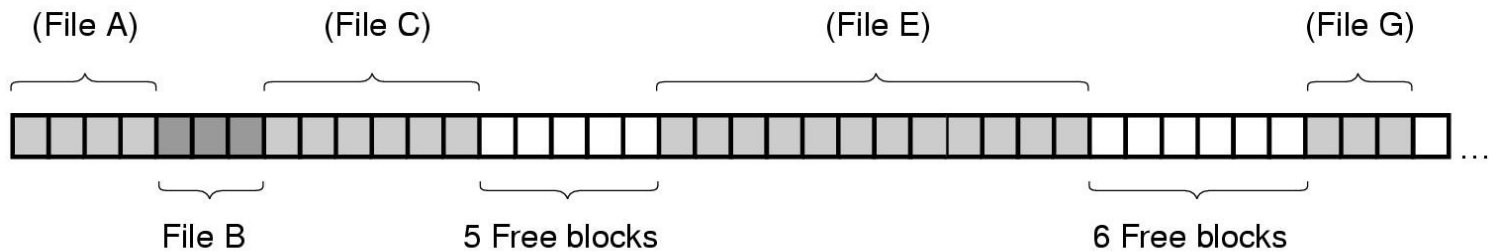
# Διάταξη Συστήματος Αρχείων



# Συνεχής Κατανομή Αρχείων



(a)

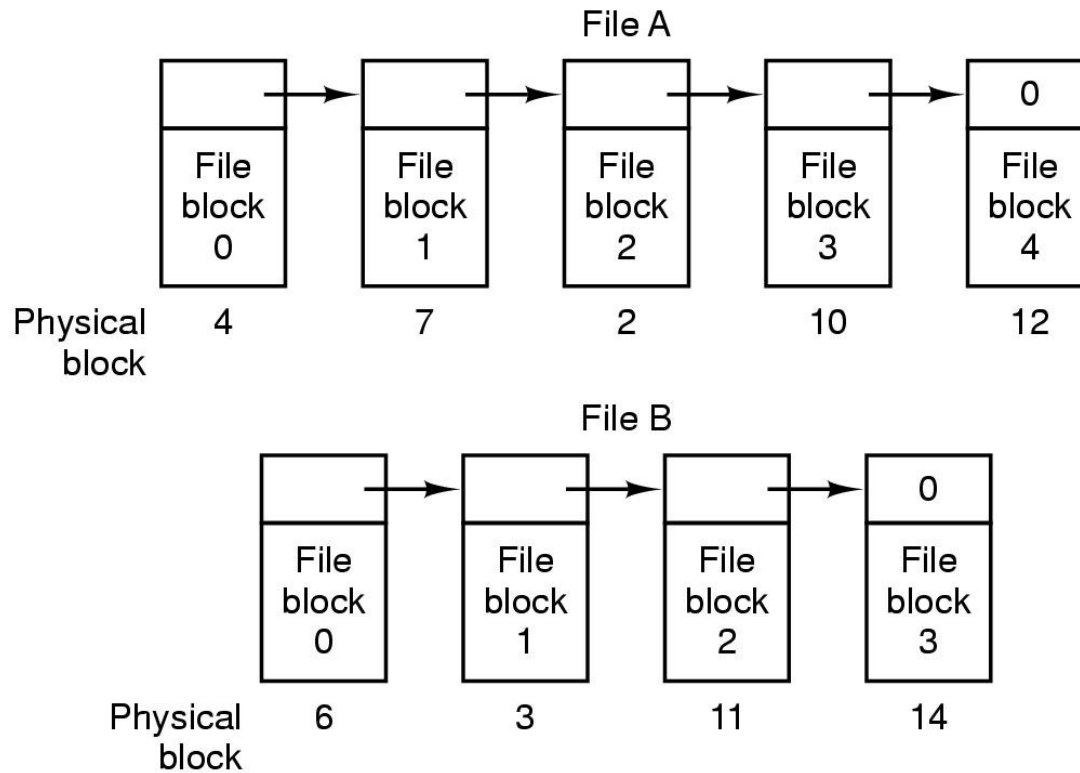


(b)

- Μεγάλο πρόβλημα: fragmentation



# Κατανομή Συνδεδεμένης Λίστας

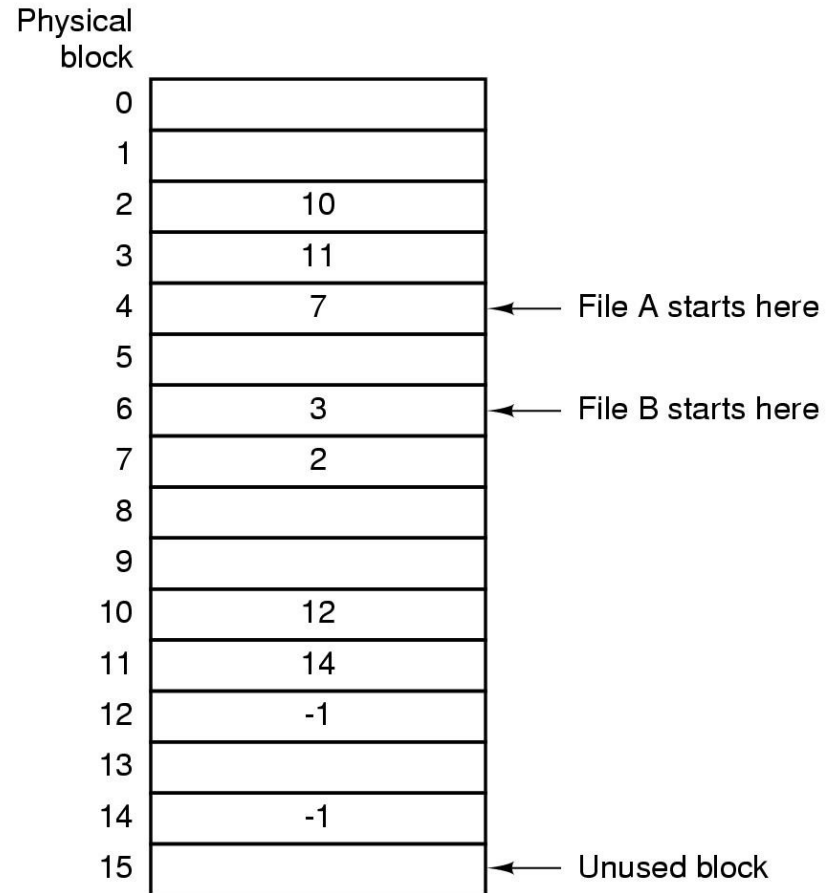


- Αργή τυχαία προσπέλαση
- Απώλεια ωφέλιμου χώρου για τους δείκτες



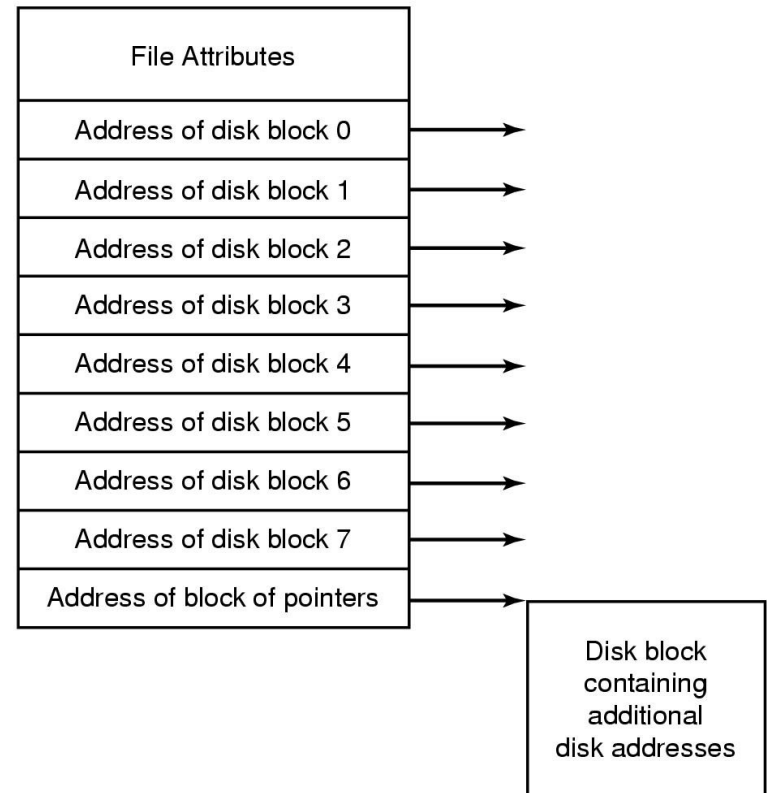
# Πίνακας Κατανομής Αρχείων

- File Allocation Table (FAT)
- Χρησιμοποιήθηκε από MS-DOS, Windows 95 & 98
- Μεγάλο πρόβλημα: η σπατάλη χώρου στη μνήμη



# Κόμβοι $i$ (i-nodes)

- Σε κάθε αρχείο αντιστοιχίζεται μία δομή που καλείται κόμβος  $i$  (index-node – i-node)
- Πλεονέκτημα: η εξοικονόμηση χώρου στη μνήμη
- Απαιτήσεις μνήμης:  $n \times k$ , όπου  $n$  το μέγεθος της δομής και  $k$  ο αριθμός των ανοικτών αρχείων



# Υλοποίηση Καταλόγων

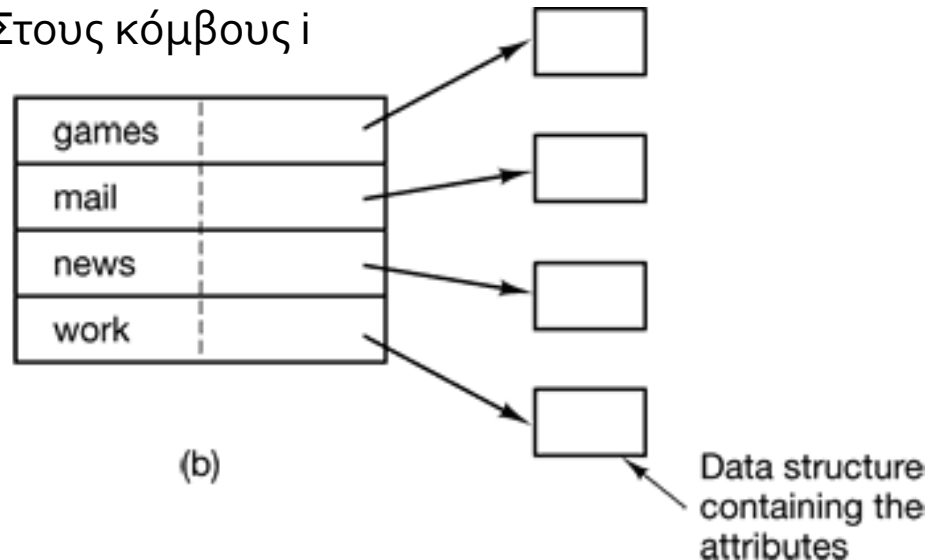
- Εναλλακτικές προσεγγίσεις αποθήκευσης μεταδεδομένων:

Στην καταχώρηση  
του καταλόγου

games	attributes
mail	attributes
news	attributes
work	attributes

(a)

Στους κόμβους  $i$

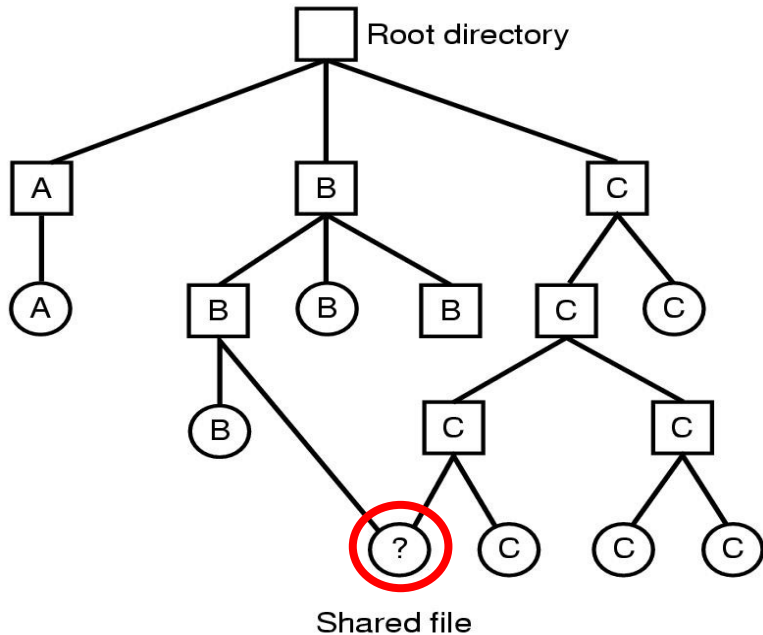


(b)





# Κοινόχρηστα Αρχεία



- Πραγματικές συνδέσεις (hard links)
  - Κάθε κατάλογος περιέχει ένα δείκτη στον κόμβο  $i$
  - Ο κόμβος  $i$  περιέχει ένα μετρητή αναφοράς
  - Το αρχείο ουσιαστικά περιλαμβάνεται σε όλους τους καταλόγους
- Συμβολικές συνδέσεις (symbolic links)
  - Μόνο ένας κατάλογος περιλαμβάνει το αρχείο
  - Οι άλλοι κατάλογοι περιλαμβάνουν δείκτες στο αρχείο και όχι πληροφορίες για τον κόμβο  $i$  ή το μπλοκ στο δίσκο
  - Η διαγραφή του αρχείου αφήνει δείκτες “σκουπίδια”



# Συστήματα Αρχείων με Καταγραφική Δομή

- Αυξανόμενες απαιτήσεις σε ταχύτητα ενώ ο χρόνος αναζήτησης αρχείων στους δίσκους παραμένει μεγάλος → συμφόρηση!
- Συστήματα Αρχείων με Καταγραφική Δομή (Log-structured File Systems – LFS)
  - Στόχος: εκμετάλλευση εύρους ζώνης του δίσκου
  - Προσέγγιση: δόμηση δίσκου σαν αρχείο καταγραφής (log file)
    - Οι εγγραφές τοποθετούνται αρχικά σε προσωρινή μνήμη
    - Περιοδική εγγραφή τους στο δίσκο σε ένα μόνο τμήμα, στο τέλος του αρχείου καταγραφής – ανάμιξη κόμβων i, μπλοκ αρχείων, κλπ.
    - Χρήση χάρτη για εντοπισμό κόμβων i και εντοπισμός αρχείων μέσω κόμβων I
    - Στο παρασκήνιο: νήμα καθαρισμού (cleaner thread) που ξεφορτώνεται τις παρωχημένες εγγραφές
- Στην πράξη: Ημερολογιακά Συστήματα Αρχείων (Journaling File Systems)
  - Τήρηση του αρχείου καταγραφής πριν την υποκείμενη ενέργεια (π.χ., εγγραφή)
  - NTFS, ext3

