

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΑΝΑΠΤΥΞΗ ΡΟΥΤΙΝΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ ΣΕ  
ΜΟΝΑΔΑ ΔΙΣΚΕΤΑΣ ΓΙΑ ΤΟΝ BGC-8088**

**Πτυχιακή εργασία του**

**Δημήτριου Τζιάτζιου (1615)**

**Επιβλέπων: Δρ. Σπυρίδων Α. Καζαρλής, Καθηγητής**

**ΣΕΡΡΕΣ, ΜΑΡΤΙΟΣ 2014**

Copyright ©

**Τζιάτζιος Δημήτριος, 2013-2014**

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## Υπεύθυνη Δήλωση

**Υπεύθυνη δήλωση** : Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής & Επικοινωνιών του Τ.Ε.Ι Σερρών.

## Ευχαριστίες

Μέσα από την εκπόνηση της πτυχιακής εργασίας επιτεύχθηκαν ορισμένοι στόχοι. Καταρχήν, έγινε η εκμάθηση αναζήτησης πληροφοριών και υπολογιστικών μοντέλων με τη βοήθεια Ελληνικής και ξένης έντυπης και ηλεκτρονικής επιστημονικής βιβλιογραφίας. Επίσης επιλέχθηκε και χρησιμοποιήθηκε ηλεκτρονικό λογισμικό για τη διευκόλυνση της επίλυσης των προβλημάτων που ανέκυψαν κατά την πορεία επεξεργασίας του θέματός μου.

Η συγκεκριμένη πτυχιακή εργασία αποτελεί αφενός το επιστέγασμα ολοκλήρωσης των σπουδών μου και αφετέρου ένα σημαντικό επιστημονικό βήμα για τη στήριξη της μετέπειτα επαγγελματικής μου εξέλιξης.

Η παρούσα πτυχιακή εργασία δε θα μπορούσε να πραγματοποιηθεί χωρίς τη βοήθεια και την επικουρία πολλών παραγόντων. Εξέχουσα όμως θέση στην πορεία πραγμάτωσης και ολοκλήρωσης της εργασίας μου κατέχει η επιστημονική και ψυχολογική βοήθεια και συμπαράσταση του επιβλέποντα καθηγητή μου κ. Σπυρίδων Καζαρλή. Η άοκνη διάθεσή του, για οποιαδήποτε επιστημονική και συγγραφική καθοδήγηση, σε όλα τα στάδια της συνεργασίας μας, υπήρξε η συνισταμένη της επιτυχούς έκβασης. Εκφράζω ταπεινά τις ευχαριστίες μου.

Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου, για την ενθάρρυνση, ηθική συμπαράσταση και οικονομική στήριξη που μου προσέφεραν όλα τα χρόνια των σπουδών μου.

## Περίληψη

Ο BGC 8088 είναι ένας αναπτυξιακός και εκπαιδευτικός ηλεκτρονικός υπολογιστής. Διαθέτει ISA slots για σύνδεση καρτών επέκτασης και μπορεί να δεχθεί κάρτα ελεγκτή μονάδας δισκέτας (FDD Controller).

Το αντικείμενο της πτυχιακής αυτής εργασίας είναι η ανάπτυξη ενός συνόλου ρουτινών, οι οποίες θα επιτρέπουν την διαχείριση αρχείων σε μονάδα δισκέτας 5 και 1/4 ιντσών 360 MB ή 1,2MB, με χρήση ελεγκτή δισκέτας.

Οι υπορουτίνες αποτελούν αυτόνομα τμήματα κώδικα που διεκπεραιώνουν μία συγκεκριμένη εργασία και μπορούμε να τα καλούμε μέσα από το κυρίως πρόγραμμα όσες φορές χρειαστεί. Τα πλεονεκτήματα χρήσης υπορουτινών είναι η μείωση του συνολικού όγκου του κώδικα σε περίπτωση ύπαρξης όμοιων επαναλαμβανόμενων τμημάτων, η διόρθωση καθίσταται ευκολότερη και τέλος ο κώδικας με τη χρήση τους γίνεται πιο κατανοητός και δίνουν τη δυνατότητα επαναχρησιμοποίησης του.

Η δυσκολία αυτής της πτυχιακής εργασίας, έγκειται στην πολυπλοκότητα και την ύπαρξη πολλών και διαφορετικών εντολών προγράμματος όσον αφορά τον BGC 8088 και τη μονάδα δισκέτας καθώς η προσπέλαση συγκεκριμένων τμημάτων της, χρειάζεται γνώση και εξοικείωση με αυτήν.

## Πίνακας περιεχομένων

Υπεύθυνη Δήλωση .....	3
Ευχαριστίες.....	4
Περίληψη.....	5
Ορισμοί.....	10
Κεφάλαιο 1: Αντικείμενο της εργασίας .....	12
1.1 Εισαγωγή. ....	12
1.2 Διάρθρωση της πτυχιακής εργασίας.....	13
Κεφάλαιο 2: Μαγνητικά Μέσα Αποθήκευσης.....	15
2.1 Εισαγωγή. ....	15
2.2 Ιστορική αναδρομή.....	16
2.3 Οργάνωση μαγνητικών δίσκων .....	17
2.4 Σκληροί Δίσκοι.....	19
2.5 Δισκέτες.....	21
2.6 Μονάδες δίσκων RAID. ....	21
2.7 Μονάδες εφεδρικής αποθήκευσης.....	22
2.8 Μαγνητοοπτικοί δίσκοι. ....	23
2.9 Το μέλλον της μαγνητικής αποθήκευσης και το υπερ-παραμαγνητικό φαινόμενο. ....	23
Κεφάλαιο 3: Η Δισκέτα.....	27
3.1 Εισαγωγή. ....	27
3.2 Ιστορία της δισκέτας .....	28
3.3 Η δισκέτα στις αρχές του 21 <sup>ου</sup> αιώνα .....	29
3.4 Χαρακτηριστικά των Οδηγών .....	30
3.4.1 Κεφαλές ανάγνωσης / εγγραφής.....	30
3.4.2 Μηχανισμός κίνησης κεφαλών.....	31
3.4.3 Μηχανισμός περιστροφής .....	33
3.4.4 Αισθητήρας αλλαγής δισκέτας (disk change sensor) .....	34
3.4.5 Συνδετήρες και γέφυρες βραχυκύκλωσης (connectors και jumpers). ....	35
3.4.6 Κυκλώματα του οδηγού δισκέτας .....	35
3.5 Η κατασκευή τους .....	36
3.5.1 Εύκαμπτοι δίσκοι 5.25" .....	36
3.5.2 Εύκαμπτοι δίσκοι 3.5" .....	37
3.5.3 Πυκνότητα επιφανείας.....	38
3.5.4 Κωδικοποίηση - Αποκωδικοποίηση δεδομένων.....	39

3.5.5 Μορφοποίηση χαμηλού και υψηλού επιπέδου ( low-level και high-level formatting ).....	40
3.5.6 Γεωμετρία των εύκαμπτων δίσκων .....	41
3.6 Διασύνδεση δισκετών.....	42
3.6.1 Εισαγωγή.....	42
3.6.2 Διασύνδεση SCSI οδηγών εύκαμπτων δίσκων.....	42
3.6.3 Υλοποίηση ελεγκτή εύκαμπτου δίσκου.....	43
3.6.4 Ταχύτητα του ελεγκτή εύκαμπτου δίσκου.....	43
3.6.5 Πηγές που χρησιμοποιούνται από τους ελεγκτές εύκαμπτων δίσκων.....	44
3.6.6 Παράμετροι του BIOS.....	45
3.6.7 Διαμόρφωση εύκαμπτου δίσκου .....	45
3.7 Μέγεθος και χωρητικότητα .....	46
3.7.1 360 KB 5.25" Floppy.....	46
3.7.2 1.2 MB 5.25" Floppy .....	46
3.7.3 720 KB 3.5" Floppy.....	47
3.7.4 1.44 MB 3.5" Floppy .....	47
3.7.5 2.88 MB 3.5" Floppy.....	48
Κεφάλαιο 4 : Συστήματα Αρχείων .....	49
4.1 Εισαγωγή.....	49
4.2 Λειτουργικά συστήματα και συστήματα αρχείων.....	51
4.2.1 DOS (MS-DOS, PC-DOS κτλ.).....	52
4.2.2 Windows 3.x.....	52
4.2.3 Windows 95.....	52
4.2.4 Windows NT .....	53
4.2.5 UNIX/Linux .....	53
4.3 Δομές διαμερισμάτων δίσκου στο σύστημα FAT .....	54
4.3.1 Εσωτερικές δομές καταλόγου.....	54
4.3.2 Ο κατάλογος της ρίζας και κανονικοί κατάλογοι.....	56
4.3.3 Ονόματα αρχείων και επεκτάσεις.....	57
4.3.4 Μεγάλα ονόματα αρχείων .....	59
4.3.5 Χαρακτηριστικά των αρχείων .....	61
4.3.6 Συστοιχίες (Μονάδες κατανομής) .....	64
4.3.7 Αλυσιδωτή οργάνωση αρχείων και κατανομή των συστοιχιών στο FAT.....	65
4.4 Μειονεκτήματα συστήματος αρχείων FAT.....	66
4.5 Σύγκριση συστημάτων αρχείων NTFS και FAT .....	68
Κεφάλαιο 5: Υπορουτίνες .....	70

5.1 Εισαγωγή.....	70
5.2 INT ( εντολή x86).....	72
5.3 Υπορουτίνες που χρησιμοποιήθηκαν για την εργασία.....	72
5.3.1 Διακοπή Debugging INT3.....	72
5.3.2 Διακοπή INT10h.....	72
5.3.3 Διακοπή INT16h.....	74
5.3.4 Διακοπή INT13h.....	74
Κεφάλαιο 6: Διαχείριση εργασίας και προτάσεις για μελλοντική έρευνα.....	76
6.1 Εισαγωγή.....	76
6.2 Διαχείριση εργασίας.....	76
6.3 Προτάσεις εξέλιξης.....	77
Κεφάλαιο 7: Ο κώδικας της εφαρμογής και αποτελέσματα.....	76



## **Κατάλογος πινάκων**

<b>3.4.3</b> Ενδεικτικές ταχύτητες περιστροφής	<b>34</b>
<b>3.5.6</b> Γεωμετρία δισκετών	<b>42</b>
<b>4.3.2</b> Αντιστοιχία αριθμού καταχωρήσεων προς τύπο διαμερίσματος	<b>58</b>

## Ορισμοί

**FAT (FILE ALLOCATION TABLE):** Ένας πίνακας κατανομής που χρησιμοποιεί το λειτουργικό σύστημα για να εντοπίσει τα αρχεία σε ένα δίσκο.

**BOOT SECTOR:** Ο τομέας εκκίνησης υπάρχει στον τομέα 0 του δίσκου και περιέχει τη βασική γεωμετρία του, που είναι η ομάδα των πληροφοριών που χρειάζονται από το λειτουργικό σύστημα για να χρησιμοποιήσει το δίσκο σωστά.

**ROOT DIRECTORY:** Ο ριζικός κατάλογος είναι ο κύριος κατάλογος των δίσκων. Σε αντίθεση με άλλους που βρίσκονται στην περιοχή δεδομένων του δίσκου, ο ριζικός κατάλογος περιορίζει το συνολικό ποσό των αρχείων ή των καταλόγων που μπορούν να δημιουργηθούν.

**FIRST LOGICAL CLUSTER:** Το πρώτο λογικό πεδίο που καθορίζει σε ποιο σημείο ένα αρχείο ή υποκατάλογος αρχίζει.

**INTERRUPT (INT):** Στον προγραμματισμό των συστημάτων, η διακοπή είναι ένα σήμα στον επεξεργαστή που εκπέμπεται από το υλικό ή το λογισμικό που δείχνει ένα γεγονός που χρειάζεται άμεση προσοχή.

**IDE/ATA:** Η αρχική, βασική προδιαγραφή για την οικογένεια των interfaces των μονάδων αποθήκευσης ονομάζεται IDE/ATA. Το όνομα (Integrated Drive

Electronics) υποδηλώνει περισσότερο την τοποθέτηση της λογικής πλακέτας πάνω στο δίσκο, παρά αναφέρεται σε interface. Το πιο σωστό όνομα είναι AT attachment, το οποίο δηλώνει ένα interface που προσαρτάται σε υπολογιστές τύπου AT. Το IDE/ATA έχει εύρος 16 bits. Αυτό το μέγεθος συνεχίζει να υπάρχει και σήμερα, ακόμη και σε υψηλότερες αποδόσεις με το όνομα ATA-2. Η προδιαγραφή ATA έχει οριστεί σαν ένα πρότυπο ANSI. Ένας κύριος λόγος για τον ορισμό του ATA σαν ένα επίσημο πρότυπο, ήταν ότι εξαφάνιζε μερικά από τα προβλήματα ασυμβατότητας τα οποία βασάνιζαν τους πρώτους IDE/ATA οδηγούς δίσκων. Αυτό γινόταν φανερό όταν οδηγοί δίσκων, που γινόταν από διαφορετικούς κατασκευαστές, τοποθετούνταν σαν κύριος (master) και ακόλουθος (slave) πάνω στο ίδιο κανάλι (όπου και συχνά δε δούλευαν).

**Πρότυπο SCSI:** Το δεύτερο κατά σειρά πιο δημοφιλές interface σκληρών δίσκων. Το SCSI είναι πολύ περισσότερο ανεπτυγμένο από το ανταγωνιστικό του IDE/ATA, έχει αρκετά πλεονεκτήματα σε σχέση με αυτό, τα οποία το κάνουν προτιμητέο για αρκετές περιπτώσεις, συνήθως για υπολογιστές υψηλής ταχύτητας.

**Cylinder-head-sector:** επίσης γνωστό ως CHS, ήταν μία πρώιμη μέθοδος για την παροχή διευθύνσεων σε κάθε φυσικό μπλοκ των δεδομένων σε ένα σκληρό δίσκο.

# Κεφάλαιο 1: Αντικείμενο της εργασίας

## 1.1 Εισαγωγή.

### Γιατί χρειαζόμαστε το FAT;

Η καταχώρηση ενός καταλόγου διαθέτει ένα πεδίο που ονομάζεται το πρώτο λογικό πεδίο το οποίο καθορίζει που το αρχείο ή ο υποκατάλογος αρχίζει. Από τη στιγμή που τα αρχεία και οι κατάλογοι μπορεί να είναι μεγαλύτερα από ένα τομέα, ένας κατάλογος ή ένα αρχείο μπορεί να αποθηκευτεί σε περισσότερους από έναν τομείς. Τα δεδομένα που ανήκουν σε αρχείο ή ένα κατάλογο δεν βρίσκονται πάντα σε συνεχείς θέσεις στην μνήμη. Ο πίνακας κατανομής ως εκ τούτου χρησιμοποιείται για να παρακολουθεί ποιοι τομείς έχουν διατεθεί για ποιο αρχείο.

Για να βρεθεί το σύνολο του περιεχομένου ενός αρχείου, για παράδειγμα, το πρώτο λογικό πεδίο θα έπρεπε να δείξει στον τομέα που κατέχει τα πρώτα 512 bytes δεδομένων. Τα δεδομένα από αυτόν τον τομέα πρέπει να διαβαστούν. Για να προσδιοριστεί αν υπάρχουν περισσότερα δεδομένα, πρέπει κανείς να εξετάσει την καταχώρηση στον πίνακα κατανομής που αντιστοιχεί στο πρώτο λογικό πεδίο του πίνακα κατανομής. Με την εξέταση της τιμής εισόδου του πίνακα κατανομής, μπορεί να προσδιοριστεί εάν υπάρχει κάποιος άλλος τομέας που διατίθεται σε αυτό το αρχείο. Αν υπάρχει, τότε η τιμή λογικού τομέα μεταφράζεται σε τιμή φυσικού τομέα και τα δεδομένα από τον τομέα διαβάζονται. Στη συνέχεια, η καταχώρηση στον πίνακα κατανομής για το δεύτερο τομέα δεδομένων εξετάζεται για να διαπιστωθεί αν πρόκειται για το τέλος του αρχείου. Εάν όχι, η διαδικασία συνεχίζεται. Ως εκ τούτου, ο πίνακας κατανομής επιτρέπει την πρόσβαση των δεδομένων που είναι αποθηκευμένα σε μη συνεχόμενους τομείς της συσκευής αποθήκευσης.

## 1.2 Διάρθρωση της πτυχιακής εργασίας

Με γνώμονα την πληρέστερη επίτευξη του σκοπού της, η παρούσα εργασία αποτελείται από πέντε κεφάλαια.

Το 1<sup>ο</sup> κεφάλαιο αποτελεί την εισαγωγή της εργασίας όπου παρουσιάζεται η σπουδαιότητα του πίνακα κατανομής (FAT). Ειδικότερα μέσα από την ανάλυση των χαρακτηριστικών του πίνακα κατανομής, είμαστε σε θέση να καταλάβουμε με ποιόν ακριβώς τρόπο αποθηκεύονται τα αρχεία σε μια μονάδα αποθήκευσης και πως μπορούμε να τα επεξεργαστούμε. Το εν λόγω κεφάλαιο ολοκληρώνεται με την διάρθρωση της πτυχιακής εργασίας, όπου και αναλύεται η δομή και το περιεχόμενο του κάθε κεφαλαίου.

Στο 2ο κεφάλαιο γίνεται αρχικά μία εισαγωγή για τα μαγνητικά μέσα αποθήκευσης και στη συνέχεια, η ιστορική αναδρομή γύρω από αυτά. Αφού γίνει μία ανάλυση στην δομή και στην οργάνωσή τους, γίνεται μία περαιτέρω αναδρομή για το κάθε μαγνητικό μέσο αποθήκευσης ξεχωριστά. Τέλος, η ενότητα κλείνει με τις πιθανότητες εξέλιξης της μαγνητικής αποθήκευσης μελλοντικά.

Στην 3<sup>η</sup> ενότητα, αναλύεται η δισκέτα, το μέσο στο οποίο η εφαρμογή είναι σχεδιασμένη να χρησιμοποιηθεί. Ξεκινώντας από την αρχή της πορείας της δισκέτας και μέχρι και τον 21<sup>ο</sup> αιώνα όπου και σταμάτησε η μαζική παραγωγή της, καταλήγουμε στα χαρακτηριστικά των δισκετών αλλά και των οδηγών αυτών, και γίνεται μετέπειτα μία αναλυτική παρουσίαση όλων των τύπων δισκετών.

Στο 4<sup>ο</sup> κεφάλαιο, γίνεται ανάλυση των συστημάτων αρχείων. Μετά από ένα σύντομο αλλά και περιεκτικό πέρασμα από λειτουργικά συστήματα και των συστημάτων αρχείων τους, εξηγούνται οι δομές των συστημάτων αρχείων.

Το 5<sup>ο</sup> κεφάλαιο, αποτελείται από τις υπορουτίνες που χρησιμοποιήθηκαν για την ολοκλήρωση της πτυχιακής αυτής εργασίας.

Το 6<sup>ο</sup> και τελευταίο κεφάλαιο, παρουσιάζει στον αναγνώστη την διαχείριση της πτυχιακής εργασίας, καθώς επίσης και προτάσεις για μελλοντική βελτίωση της εφαρμογής.

Στο 7<sup>ο</sup> και τελευταίο κεφάλαιο βρίσκεται ο κώδικας της εφαρμογής καθώς και τα αποτελέσματα της.

## **Κεφάλαιο 2: Μαγνητικά Μέσα Αποθήκευσης**

### **2.1 Εισαγωγή.**

Τα μαγνητικά μέσα αποθήκευσης είναι οι ταινίες (magnetic tapes) και οι δίσκοι (disks). Η εγγραφή των ψηφιακών δεδομένων σε αυτά βασίζεται στη μόνιμη μαγνήτιση μιας μικρής περιοχής του μαγνητικού μέσου με την βοήθεια ενός ηλεκτρομαγνήτη. Η ανάγνωση γίνεται με το πέρασμα του ηλεκτρομαγνήτη πάνω από τη μαγνητισμένη περιοχή και τη δημιουργία επαγωγικού ρεύματος. Η φορά του επαγωγικού ρεύματος εξαρτάται από την πολικότητα της μαγνήτισης και επομένως καθορίζει τη τιμή της ψηφιακής πληροφορίας: 0 ή 1. Στην περίπτωση της εγγραφής έχουμε το αντίστροφο φαινόμενο. Ανάλογα με την τιμή 0 ή 1 που θέλουμε να γραφεί στο δίσκο, διοχετεύεται ρεύμα αντίστοιχης φοράς, δημιουργείται λοιπόν μαγνητικό πεδίο που μαγνητίζει αντίστοιχα την περιοχή του δίσκου.

## 2.2 Ιστορική αναδρομή.

Ο μαγνητικός δίσκος ανακαλύφθηκε το 1956 από την εταιρεία IBM, ως λύση στο πρόβλημα της αργής σειριακής αναζήτησης δεδομένων της μαγνητικής ταινίας. Η IBM κατασκεύασε την πρώτη αποθηκευτική συσκευή που χρησιμοποιούσε μαγνητικούς δίσκους το 1956 και τη νοίκιαζε σε πελάτες της με μηνιαία ή εβδομαδιαία συμβόλαια. Ήταν μια αποθηκευτική συσκευή δεδομένων στο μέγεθος βιβλιοθήκης, με την πρωτοφανή χωρητικότητα για την εποχή των 5.000.000 χαρακτήρων (5 Megabytes), όπως επίσης και τον πρωτοφανή τρόπο τυχαίας προσπέλασης των αποθηκευμένων δεδομένων, χάρις του οποίου η συσκευή ονομάστηκε RAMAC (Random Access Method of Accounting and Control – Τυχαίας Προσπέλασης Μέθοδος Περιγραφής και Ελέγχου). Ο χρήστης της συσκευής αυτής είχε στη διάθεσή του οποιοδήποτε κομμάτι της αποθηκευμένης πληροφορίας σε 0,8 δευτερόλεπτα, σε αντίθεση με τα μερικά λεπτά που έπρεπε να περιμένει από μια συσκευή ανάγνωσης μαγνητικών ταινιών εκείνης της εποχής. Αυτή η αρχέγονη μορφή αποθηκευτικής συσκευής μαγνητικών δίσκων, αποτελείτο από 50 μεταλλικούς δίσκους με 24 ίντσες διάμετρο ο καθένας, οι οποίοι περιστρέφονταν με 1200 στροφές το λεπτό. Δύο μαγνητικές κεφαλές ήταν υπεύθυνες για την εγγραφή και ανάγνωση των δεδομένων και στις δύο πλευρές της επιφάνειας των 50 δίσκων και ήταν ικανές να μεταφέρουν δεδομένα με ρυθμό 12,5 Kilobytes ανά δευτερόλεπτο. Λόγω του υπερβολικού μεγέθους τους, οι πρώτες συσκευές αποθήκευσης σε μαγνητικούς δίσκους ονομάστηκαν καθηλωμένοι δίσκοι.

Πολύ αργότερα, γύρω στο 1967 όταν έκαναν την εμφάνισή τους οι πρώτοι μαλακοί δίσκοι (δισκέτες), επίσης εφεύρεση της IBM, οι ογκώδεις συσκευές με τους μεταλλικούς (καθηλωμένους δίσκους) μετονομάστηκαν σε σκληρούς δίσκους. Η ονομασία αυτή χρησιμοποιείται μέχρι και σήμερα, για τη διαφοροποίηση μεταξύ αναλώσιμων εύκαμπτων πλαστικών δίσκων, που γράφονται και διαβάζονται από τις ανάλογες συσκευές και άκαμπτων δίσκων, κατασκευασμένων από μέταλλο ή γυαλί, που λειτουργούν σαν αυτόνομες αποθηκευτικές συσκευές. Οι σκληροί δίσκοι είναι συσκευές που περιέχουν έναν ή πολλούς άκαμπτους δίσκους, σφραγισμένους μαζί με όλη την ηλεκτρομηχανική υποδομή για την ανάγνωση και εγγραφή τους, μέσα σε αεροστεγή μεταλλικά κουτιά.



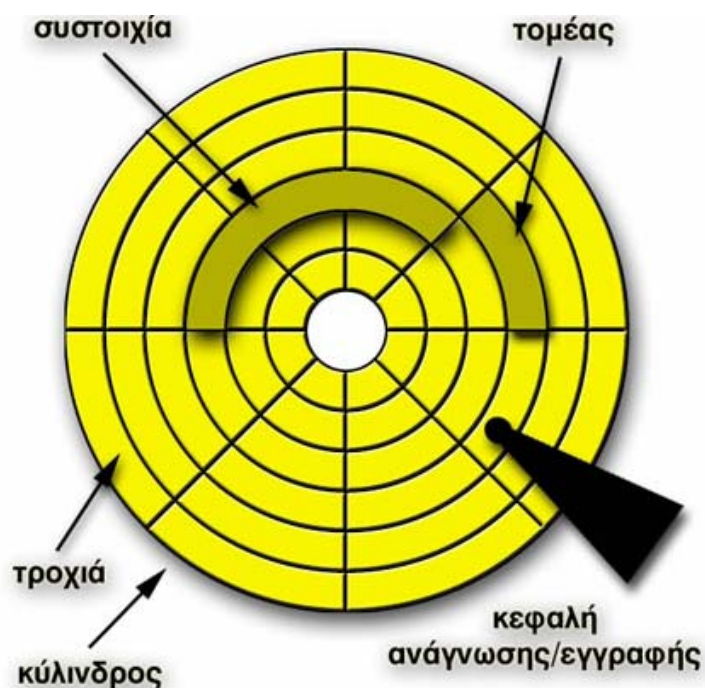
Από τότε μέχρι σήμερα, η κυριότερη εφαρμογή του σκληρού δίσκου συναντάται στη μόνιμη αποθήκευση των δεδομένων της ασταθούς μνήμης του ηλεκτρονικού υπολογιστή (RAM - Random Access Memory), όπως επίσης και για την αποθήκευση λοιπών πληροφοριών και προγραμμάτων, που χρησιμοποιούνται συχνά και δε πρέπει να χαθούν μετά τη διακοπή τροφοδοσίας του υπολογιστή με ηλεκτρικό ρεύμα. Παρομοίως, ο ρόλος των εύκαμπτων πλαστικών δίσκων έχει παραμείνει και αυτός αμετάβλητος, ως μέσο μεταφοράς και εφεδρικής αποθήκευσης δεδομένων.

Οι μαγνητικοί δίσκοι δεδομένων έχουν τις βάσεις τους στην αρχιτεκτονική των γραμμόφωνων (πικάπ), από την οποία εμπνεύστηκαν οι επιστήμονες της IBM τη βασική ιδέα αποθήκευσης δεδομένων σε δίσκο. Το μέσο αποθήκευσης της πληροφορίας έχει σχήμα δίσκου και περιστρέφεται συνεχώς με σταθερή γωνιακή ταχύτητα γύρω από τον άξονά του. Στους μαγνητικούς δίσκους, ένας βραχίονας στο ελεύθερο άκρο του οποίου είναι στερεωμένη η κεφαλή εγγραφής και ανάγνωσης, κινείται από το κέντρο προς την περιφέρεια της επιφάνειας του δίσκου, κατά τέτοιο τρόπο ώστε, σε συνδυασμό με την περιστρεφόμενη κίνηση του δίσκου, η κεφαλή εγγραφής και ανάγνωσης των δεδομένων να έχει πρόσβαση σε όλη την επιφάνεια του μέσου αποθήκευσης.

Σε αντίθεση με τη σπειροειδή εγγραφή του ήχου στους δίσκους βινυλίου, όπου η αποθηκευμένη πληροφορία είναι γραμμικής φύσεως, στους μαγνητικούς δίσκους δεδομένων η πληροφορία είναι οργανωμένη κατά τέτοιο τρόπο, ώστε η εύρεση, ανάγνωση, τροποποίηση και διαγραφή της, να επιτυγχάνεται με όσο το δυνατό ταχύτερο και αξιόπιστο τρόπο. Έτσι, στο μαγνητικό δίσκο αποθήκευσης δεδομένων η πληροφορία είναι οργανωμένη σε ομόκεντρους κύκλους, τα αποκαλούμενα ίχνη (tracks), τα οποία διαιρούνται σε μικρά τόξα του κύκλου, τους λεγόμενους τομείς (sectors). Όλοι οι τομείς έχουν την ίδια αποθηκευτική ικανότητα δεδομένων, η οποία ορίζεται από τον κατασκευαστή του δίσκου. Για παράδειγμα, οι τομείς των σκληρών δίσκων έχουν την ικανότητα αποθήκευσης 512 Bytes, ανεξάρτητα σε ποιο ίχνος βρίσκονται. Στην περίπτωση των σκληρών δίσκων με περισσότερους του ενός μαγνητικούς δίσκους, τα ίχνη για την ίδια ακτίνα του κύκλου που ανήκουν σε όλο το πλήθος των δίσκων αποτελούν τον κύλινδρο (cylinder).

### **2.3 Οργάνωση μαγνητικών δίσκων.**

Η επιφάνεια ενός μαγνητικού δίσκου διαμερίζεται σε ομόκεντρους κύκλους ή τροχιές (tracks). Κάθε τροχιά χωρίζεται σε κυκλικά τόξα ή περιοχές που ονομάζονται τομείς (sectors). Κάθε τομέας έχει μέγεθος σταθερό, με άλλα λόγια περιέχει σταθερό αριθμό από bytes, ο οποίος κυμαίνεται από 512 bytes ως 32 KB ή και περισσότερο. Το σύνολο των δεδομένων που αποθηκεύονται σε ένα τομέα του δίσκου λέγεται ενότητα (block). Δύο ή περισσότεροι τομείς, σχηματίζουν την συστοιχία (cluster), που αποτελεί την μικρότερη μονάδα αποθήκευσης για τα δεδομένα ενός προγράμματος.



Εικόνα 2.1 Οργάνωση μαγνητικού δίσκου

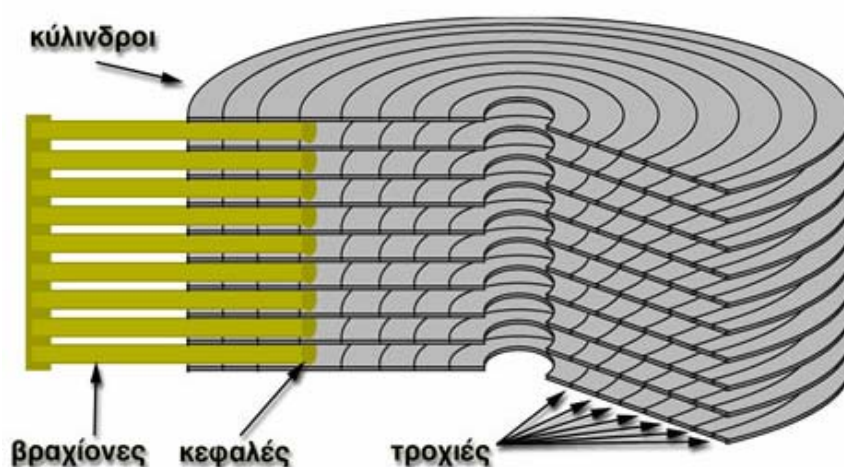
Όπως μπορούμε να δούμε και στην παραπάνω εικόνα, το μέγεθος των τομέων που ανήκουν σε διαφορετικές τροχιές δεν είναι το ίδιο. Οι τομείς που βρίσκονται κοντά στην εξωτερική τροχιά του δίσκου έχουν μεγαλύτερο μήκος απ' ό,τι οι εσωτερικοί

τομείς. Η ποσότητα όμως των δεδομένων που αποθηκεύεται σε κάθε τομέα είναι συνήθως σταθερή· άρα έχουμε μεγαλύτερη πυκνότητα αποθήκευσης όσο πλησιάζουμε προς το κέντρο του δίσκου. Πάνω από την επιφάνεια του δίσκου και σε μικρή απόσταση από αυτόν (εκατομμυριοστά του μέτρου) βρίσκεται η κεφαλή (head) ανάγνωσης/εγγραφής, η οποία φέρει τον ηλεκτρομαγνήτη για την ανάγνωση των 0 και 1 που είναι γραμμένα στην επιφάνεια του δίσκου. Οι κεφαλές μπορεί να είναι σταθερές ή κινούμενες. Στην περίπτωση των σταθερών κεφαλών, πάνω από την επιφάνεια του μαγνητικού δίσκου βρίσκονται πολλές κεφαλές, μια για κάθε τροχιά. Οι κινούμενες κεφαλές είναι οι πλέον διαδεδομένες σε μονάδες μαγνητικών δίσκων. Όταν οι κεφαλές είναι κινούμενες, σε κάθε επιφάνεια του μαγνητικού δίσκου αντιστοιχεί μια μοναδική κεφαλή. Όταν οι κεφαλές είναι κινούμενες, σε κάθε επιφάνεια του μαγνητικού δίσκου αντιστοιχεί μια μοναδική κεφαλή. Η διαδικασία ανάγνωσης/εγγραφής δεδομένων αποτελείται από τα εξής βήματα: Αρχικά μετακινείται η κεφαλή πάνω από την τροχιά στην οποία υπάρχουν ή πρόκειται να γραφούν τα δεδομένα. Ο χρόνος που απαιτείται για τη μετακίνηση αυτή ονομάζεται χρόνος αναζήτησης (seek time). Στη συνέχεια, καθώς ο δίσκος περιστρέφεται συνεχώς, αναμένουμε να βρεθεί η κεφαλή ακριβώς πάνω από τον συγκεκριμένο τομέα και συστοιχία. Ο χρόνος που απαιτείται μέχρι να γίνει αυτό εξαρτάται από το ρυθμό περιστροφής του μαγνητικού δίσκου. Κατόπιν η κεφαλή διαβάζει ή γράφει τα δεδομένα σε συστοιχίες και τα μεταδίδει. Ο χρόνος ανάγνωσης ή εγγραφής εξαρτάται από την ταχύτητα ανάγνωσης/εγγραφής της κεφαλής, από την ταχύτητα των ηλεκτρονικών που υποστηρίζουν το δίσκο, από το πρωτόκολλο διαδρόμου κλπ.

#### **2.4 Σκληροί Δίσκοι.**

Οι σκληροί δίσκοι αποτελούν το κυριότερο μέσο αποθήκευσης στους σύγχρονους υπολογιστές. Αποτελούνται από δύο ή περισσότερους ομοαξονικούς μαγνητικούς δίσκους που περιστρέφονται συνεχώς με την ίδια γωνιακή ταχύτητα σε μικρή απόσταση μεταξύ τους. Οι δίσκοι κατασκευάζονται από μεταλλικό ή κεραμικό κράμα, κατά προτίμηση ελαφρύ αλλά ανθεκτικό σε υψηλές θερμοκρασίες. Η χρήση υλικού ανθεκτικού στη θερμότητα είναι απαραίτητη, καθώς οι σκληροί δίσκοι περιστρέφονται με χιλιάδες στροφές ανά λεπτό, σχεδόν συνεχώς, και επομένως αναπτύσσονται μεγάλες θερμοκρασίες στο εσωτερικό τους. Στις εξωτερικές

επιφάνειες του δίσκου υπάρχει λεία επίστρωση από μαγνητικό υλικό. Οι κεφαλές ανάγνωσης είναι συνήθως για καθένα δίσκο, μια σε κάθε πλευρά. Για να επιτευχθούν μεγάλες ταχύτητες ανάγνωσης/εγγραφής δεδομένων, οι κεφαλές βρίσκονται σε πολύ μικρή απόσταση από την επιφάνεια. Οι σκληροί δίσκοι είναι συσκευασμένοι σε κενό αέρα, καθώς το παραμικρό μόριο σκόνης ή άλλης ακαθαρσίας μπορεί να καταστρέψει τον δίσκο.



Εικόνα 2.2 Διάταξη σκληρού δίσκου

Στην παραπάνω εικόνα βλέπουμε μια τυπική διάταξη σκληρού δίσκου που αποτελείται από ομοαξονικούς μαγνητικούς δίσκους. Οι αντίστοιχες τροχιές από κάθε δίσκο, ορίζουν ένα κύλινδρο (cylinder). Σε κάθε στιγμή, όλες οι κεφαλές είναι τοποθετημένες στις τροχιές που ανήκουν στον ίδιο κύλινδρο. Για την προσπέλαση των δεδομένων στο σκληρό δίσκο, υπάρχει μια μοναδική διεύθυνση για κάθε τομέα, η οποία αποτελείται από τον αριθμό της κεφαλής (δηλαδή τον αριθμό δίσκου), τον αριθμό του κυλίνδρου ή τροχιάς στο δίσκο και τον αριθμό του τομέα μέσα στον κύλινδρο. Η χωρητικότητα (capacity) σε Kilobytes (KB) ενός σκληρού δίσκου δίνεται από το γινόμενο:

**σύνολο κεφαλών × αριθμός κυλίνδρων × αριθμός τομέων/κύλινδρο × KB/τομέα**

Τα χαρακτηριστικά αυτά (κεφαλές, κύλινδροι, τομείς) αποτελούν τη γεωμετρία του σκληρού δίσκου. Η σύγχρονη τεχνολογία επιτρέπει την δημιουργία δίσκων με πυκνότερες τροχιές και πιο μικρούς τομείς ανά μονάδα πληροφορίας, άρα μεγαλύτερες χωρητικότητες.

## 2.5 Δισκέτες.

Οι δισκέτες αποτελούνται από έναν εύκαμπτο πλαστικό δίσκο πάνω στο οποίο βρίσκεται το μαγνητικό υλικό, ενώ το εξωτερικό τους περίβλημα αποτελείται από πλαστικό, προστατεύοντάς τις από υγρασία και σκόνη. Η εγγραφή πάνω στον εύκαμπτο μαγνητικό δίσκο γίνεται και από τις δύο πλευρές, δίνοντας συνολική χωρητικότητα 1,44MB για διάμετρο δίσκου 3,5 ιντσών, και 1,2 MB για διάμετρο 5 ¼ ιντσών. Στις δισκέτες η κεφαλή ανάγνωσης/εγγραφής δε βρίσκεται σε απόσταση από την επιφάνεια του μαγνητικού δίσκου, αλλά έρχεται σε άμεση επαφή με αυτόν. Επιπλέον, οι δισκέτες περιστρέφονται μόνο όταν πρόκειται να γίνει προσπέλαση δεδομένων και όχι συνεχώς. Εξαιτίας των παραπάνω, η ταχύτητα περιστροφής μιας δισκέτας είναι πολύ μικρότερη από αυτή του σκληρού δίσκου, γύρω στις 300 στροφές/λεπτό, και η ταχύτητα προσπέλασης δεδομένων πολύ μικρότερη.

## 2.6 Μονάδες δίσκων RAID.

Οι μονάδες RAID (Redundant Array of Inexpensive/Independent Disks) είναι συστοιχίες δύο ή περισσότερων κοινών σκληρών δίσκων προσφέροντας κυρίως μεγαλύτερη απόδοση και ανοχή σε σφάλματα. Η απόδοση των δίσκων αυξάνεται, καθώς τα δεδομένα δεν αποθηκεύονται σειριακά στους δίσκους αλλά παράλληλα, όπως υπαγορεύει το πρότυπο του RAID. Έτσι διαδοχικά bytes ενός τομέα ή ακόμα

και τα bits του ίδιου byte γράφονται σε ξεχωριστούς δίσκους, ώστε η εγγραφή και η ανάγνωση να γίνονται με παράλληλη λειτουργία όλων των δίσκων. Ένας ελεγκτής RAID αναλαμβάνει το συνδυασμό των bits ή των bytes, ανάλογα με την υλοποίηση, ώστε ο υπολογιστής να βλέπει ένα «ενιαίο» σκληρό δίσκο. Η ανοχή σε σφάλματα επιτυγχάνεται με δύο τρόπους:

- Με καθρεπτισμό των δίσκων (Mirroring).
- Με την ύπαρξη δίσκου με bit ισοτιμίας (parity bit).

Στον καθρεπτισμό υπάρχει ένα πανομοιότυπο αντίγραφο κάθε δίσκου, ώστε σε περίπτωση απώλειας του πρωτότυπου να υπάρχει το αντίγραφο. Στην ισοτιμία, ένας δίσκος αποθηκεύει το bit ισοτιμίας άλλων δύο δίσκων, ώστε σε περίπτωση απώλειας του ενός από αυτούς, με τον δίσκο ισοτιμίας και τον άλλο να μπορούμε να ανακτήσουμε τα δεδομένα αυτού που χάθηκε. Προφανώς και οι δύο λύσεις μας επιβαρύνουν με ύπαρξη δίσκων εφεδρείας που λειτουργούν ως καθρέπτες ή ως δίσκοι ισοτιμίας. Το κόστος απώλειας των δεδομένων όμως είναι πάντα πολύ μεγαλύτερο από αυτό των επιπλέον μονάδων δίσκων.

## **2.7 Μονάδες εφεδρικής αποθήκευσης.**

Οι μονάδες εφεδρείας (backup units) χρησιμοποιούνται για τη δημιουργία αντιγράφων ασφαλείας των δεδομένων που είναι αποθηκευμένα στους υπολογιστές.

Στην κατηγορία αυτή ανήκουν κυρίως οι μαγνητικές ταινίες, οι μαγνητοοπτικοί δίσκοι, και οι οπτικοί δίσκοι CD-RW και DVD.

## **2.8 Μαγνητοοπτικοί δίσκοι.**

Οι μαγνητοοπτικοί δίσκοι έχουν τις ίδιες διαστάσεις με μια εύκαμπτη δισκέτα 5 ¼ ιντσών, αλλά είναι παχύτεροι. Η χωρητικότητα τους ποικίλλει από 128 MB έως 1,2 GB. Συνδυάζουν τη μαγνητική τεχνολογία κατά την εγγραφή τους και την οπτική κατά την ανάγνωση. Έχουν επομένως μεγαλύτερη διάρκεια ζωής από τις μαγνητικές ταινίες, καθώς οι πολλαπλές αναγνώσεις δεν αλλοιώνουν τα αποθηκευμένα δεδομένα. Από την άλλη πλευρά είναι σημαντικά ακριβότερα μέσα αποθήκευσης, ενώ σημαντικά κοστίζουν και οι περιφερειακές συσκευές που τους υποστηρίζουν. Έχουν εμφανιστεί τέτοιοι οδηγοί ανάγνωσης δισκετών, οι οποίοι διαβάζουν και εύκαμπτους οπτικούς δίσκους, δισκέτες δηλαδή που έχουν την ίδια μαγνητοοπτική τεχνολογία και χωρητικότητα που κυμαίνεται από 21-120 MB δεδομένων.

## **2.9 Το μέλλον της μαγνητικής αποθήκευσης και το υπερ-παραμαγνητικό φαινόμενο.**

Είναι γεγονός πως ακόμα και σήμερα η βιομηχανία της μαγνητικής αποθήκευσης βασίζεται στην τεχνολογία που εισήγαγε το 1900 ο εφευρέτης Valdemar Poulsen για τη μαγνητική καταγραφή του ήχου. Δηλαδή, τίποτα περισσότερο από ένα ηλεκτρομαγνήτη ο οποίος ακουμπά ή ίπταται πολύ κοντά σε μια επιφάνεια ικανή να μαγνητίζεται μόνιμα και με την οποία ανταλλάσσουν επαγωγικά τα μαγνητικά τους πεδία. Σίγουρα από τότε μέχρι σήμερα έχουν αλλάξει πολλά, ωστόσο η αρχή λειτουργίας έχει παραμείνει η ίδια, ακόμα και στους σύγχρονους σκληρούς δίσκους, οι οποίοι λόγω της κατασκευής τους επιδεικνύουν τη μεγαλύτερη συγκέντρωση δεδομένων ανά τετραγωνικό εκατοστό από οποιοδήποτε άλλο μέσο μαγνητικής αποθήκευσης.

Οι έρευνες για την εξέλιξη της μαγνητικής αποθήκευσης έχουν όλες επικεντρωθεί στο πως να στριμώξουν όλο και περισσότερα δεδομένα σε όσο το δυνατό μικρότερη επιφάνεια αποθηκευτικού μέσου. Για το λόγω αυτό, με το πέρασμα του χρόνου οι κεφαλές εγγραφής και ανάγνωσης κατασκευάζονται όλο και πιο μικρές, η απόσταση πτήσης τους πλησιάζει όλο και περισσότερο την επιφάνεια αποθήκευσης και το υλικό του μέσου αποθήκευσης εμπλουτίζεται με περισσότερα και μικρότερα σιδηρομαγνητικά στοιχεία, για την αύξηση της ανάλυσής του. Με τις παραπάνω τεχνικές, η βιομηχανία της μαγνητικής αποθήκευσης καταφέρνει ακόμα και σήμερα το διπλασιασμό της χωρητικής ικανότητας των μέσων κάθε 18 μήνες, διατηρώντας ωστόσο το μέγεθός τους σταθερό. Εντούτοις, οι ερευνητές υποστηρίζουν πως το τέλος αυτού του φρενήρη ρυθμού εξέλιξης της μαγνητικής αποθήκευσης φτάνει στο τέλος του, το οποίο ακούει στο όνομα υπερ-παραμαγνητικό φαινόμενο.

Το υπερ-παραμαγνητικό φαινόμενο παρουσιάζεται λόγω της συρρίκνωσης των στοιχειωδών μαγνητικών περιοχών (κόκκων) αποθήκευσης δεδομένων του μέσου, η οποία πραγματοποιείται με σκοπό την αύξηση της χωρητικής του ικανότητας. Το φαινόμενο αυτό εκφράζει την αποσταθεροποίηση της πολικότητας του μαγνητικού πεδίου των μαγνητικών κόκκων του μέσου και έχει ως επακόλουθο την αλλοίωση των αποθηκευμένων δεδομένων. Η αποσταθεροποίηση αυτή συμβαίνει λόγω της μαγνητικής αλληλεπίδρασης μεταξύ των κόκκων με διαφορετική πολικότητα, η οποία γίνεται όλο και πιο έντονη όσο μειώνεται το μέγεθός τους και συνεπώς η μαγνητική τους χωρητικότητα. Το μέγεθος των κόκκων αυτών καθορίζει την ειδική χωρητικότητα του αποθηκευτικού μέσου, δηλαδή τον αριθμό δεδομένων που μπορούν να αποθηκευτούν σε μια τετραγωνική του ίντσα. Σήμερα, οι σύγχρονοι σκληροί δίσκοι έχουν ειδική χωρητική ικανότητα που κυμαίνεται λίγο πάνω από τα 100 Gigabit ανά τετραγωνική ίντσα, η εταιρία Seagate, ήδη από το τρίτο τρίμηνο του 2004, διαθέτει στην αγορά σκληρούς δίσκους με ειδική χωρητική ικανότητα 108 Gigabit ανά τετραγωνική ίντσα. Ωστόσο, εικάζετε από τους επιστήμονες πως με την παρούσα τεχνολογία των σκληρών δίσκων, η ειδική χωρητικότητά τους δε θα ξεπεράσει τα 150 με 200 Gigabit ανά τετραγωνική ίντσα. Από εκεί και πάνω το υπερ-παραμαγνητικό φαινόμενο θα είναι τόσο έντονο, που θα επιβάλει στους κατασκευαστές να αλλάξουν τον τρόπο λειτουργίας και κατασκευής των μαγνητικών σκληρών δίσκων του κοντινού μέλλοντος.



Οι υποψήφιες τεχνολογίες που υπόσχονται να ξεπεράσουν το όριο της ειδικής χωρητικότητας των 200 Gigabit ανά τετραγωνική ίντσα και να διαδεχθούν την παρούσα τεχνολογία μαγνητικής αποθήκευσης των σκληρών δίσκων είναι πολλές.

Μια από αυτές τις τεχνολογίες, είναι η χρήση κεφαλών κάθετης εγγραφής και ανάγνωσης των δεδομένων, η οποία υπόσχεται το διπλασιασμό με τετραπλασιασμό της ειδικής χωρητικότητας, που μπορεί να επιτευχθεί με τη μέθοδο της διαμήκης εγγραφής που χρησιμοποιείται μέχρι σήμερα. Η πληροφορία στα σημερινά αποθηκευτικά μέσα εκφράζεται από μαγνητικά πεδία τα οποία έχουν προσανατολισμό παράλληλο με την επιφάνεια του αποθηκευτικού μέσου. Στην κάθετη εγγραφή της πληροφορίας, τα μαγνητικά πεδία των δεδομένων έχουν προσανατολισμό κάθετο στην επιφάνεια του μέσου, με αποτέλεσμα να γίνεται καλύτερη εκμετάλλευση της επιφανειακής χωρητικότητας του. Ωστόσο, το αποθηκευτικό μέσο πρέπει να είναι ελαφρός παχύτερο από πριν, έτσι ώστε ο όγκος των κόκκων και συνεπώς η μαγνητική τους χωρητικότητα, να παραμείνουν τα ίδια.

Μια άλλη μέθοδος, η οποία υπόσχεται δεκαπλάσια επέκταση των ορίων εμφάνισης του υπερ-παραμαγνητικού φαινομένου, είναι η μαγνητική αποθήκευση σε προσκαλισμένα αποθηκευτικά μέσα. Η τεχνική αυτή υπόσχεται την αποθήκευση ενός bit πληροφορίας σε ένα μαγνητικό κόκκο του αποθηκευτικού μέσου, με την προϋπόθεση βέβαια τον ανάγλυφο σχηματισμό του κόκκου στην επιφάνεια του αποθηκευτικού μέσου. Αυτό μπορεί να επιτευχθεί είτε με λιθογραφικές μεθόδους, οι οποίες είναι ακριβείς αλλά οικονομικά ασύμφορες για τη μαζική παραγωγή μέσων, είτε με τη χρήση κάποιας μήτρας (καλουπιού), όπως ακριβώς γίνεται και στην παραγωγή οπτικών δίσκων, με τη διαφορά πως θα πρέπει αυτή να είναι πολύ πιο λεπτούφης από την αντίστοιχη των οπτικών μέσων, έτσι ώστε να προσφέρει την απαιτούμενη πυκνότητα χαρακτηριστικών.

Άλλη μια πολλά υποσχόμενη τεχνολογία, προτεινόμενη από την εταιρία Seagate, είναι αυτή της μαγνητικής εγγραφής υποβοηθούμενης από θερμότητα. Η μέθοδος αυτή βασίζεται σε μέσα με πολύ πιο σταθερή μαγνητική σύσταση, όπου οι μαγνητικοί κόκκοι μπορούν να συρρικνωθούν ακόμα περισσότερο χωρίς να εμφανίσουν προβλήματα αποσταθεροποίησης. Ωστόσο, λόγω της σταθερότητας του μέσου, απαιτείται πολύ πιο ισχυρό μαγνητικό πεδίο για την εγγραφή και διαγραφή των δεδομένων από την επιφάνειά του, το οποίο είναι αδύνατο να παραχθεί από της

μικροσκοπικές κεφαλές που χρησιμοποιούνται ή πρόκειται να χρησιμοποιηθούν στους μελλοντικούς σκληρούς δίσκους αυτής της τεχνολογίας. Για το λόγο αυτό, η εγγραφή των δεδομένων υποβοηθείται από ακτίνες λέιζερ, οι οποίες θερμαίνουν με ακρίβεια την περιοχή στην οποία θα αποθηκευτούν τα δεδομένα, ώστε να πραγματοποιηθεί η εγγραφή τους ακόμα και με ασθενές μαγνητικό πεδίο, όπως ακριβώς συμβαίνει και στη μαγνητο-οπτική αποθήκευση. Η μέθοδος αυτή υπόσχεται εκατοντάπλασια παράταση του ορίου εμφάνισης του υπερ-παραμαγνητικού φαινομένου, με ύστατο στόχο να επεκτείνει την ειδική χωρητικότητα των μαγνητικών σκληρών δίσκων στα 50 Terabit ανά τετραγωνική ίντσα. Η τεχνολογία αυτή έκανε την παρθενική της εμφάνιση το 2010, χρονιά που η εταιρία είχε σαν στόχο να παρουσιάσει τον πρώτο της σκληρό δίσκο μαγνητικής αποθήκευσης με ειδική χωρητικότητα 1 Terabit ανά τετραγωνική ίντσα.

Οι τεχνολογίες αυτές βρίσκονται ακόμα σε αρκετά πρώιμο στάδιο και ο καιρός που θα καταστούν εμπορικά εκμεταλλεύσιμες και προσιτές σε όλους, είναι ακόμα πολλά χρόνια πιο μπροστά από την εποχή που το υπερ-παραμαγνητικό φαινόμενο θα αρχίσει να προκαλεί προβλήματα στο ρυθμό εξέλιξης των σκληρών δίσκων. Παρομοίως, πριν από μια πενταετία εικαζόταν πως το υπερ-παραμαγνητικό φαινόμενο θα σταματούσε την εξέλιξη της ειδικής χωρητικότητας των μαγνητικών σκληρών δίσκων στα 40 Gigabit ανά τετραγωνική ίντσα. Σήμερα ωστόσο, βλέπουμε πως η ειδική χωρητικότητα των σκληρών δίσκων οδεύει πέρα από τα 100 και προορίζεται να φτάσει τα 150 Gigabit ανά τετραγωνική ίντσα. Το συμπέρασμα είναι πως όσο υπάρχει ζήτηση για όλο και πιο ευρύχωρους, ταχύτερους και συμπαγείς χώρους ψηφιακής αποθήκευσης δεδομένων και όσο η αγορά μπορεί να συντηρήσει την έρευνα ανάπτυξης και εξέλιξης νέων τεχνολογιών, το τέλος του ξέφρενου ρυθμού αύξησης της ειδικής χωρητικότητας των αποθηκευτικών μέσων, το πολύ, να παρουσιάσει μια ελαφριά κάμψη, ωστόσο δε πρόκειται να σταματήσει στα επόμενα 5 με 10 χρόνια, όσο εξωπραγματική και αν δείχνει η υπερπήδηση των εμποδίων που εμφανίζονται προς την πορεία αυτή.

## Κεφάλαιο 3: Η Δισκέτα

### 3.1 Εισαγωγή.

Η δισκέτα, είναι ένα μέσο αποθήκευσης το οποίο αποτελείται από ένα δίσκο με λεπτό και εύκαμπτο μαγνητικό μέσο αποθήκευσης, σφραγισμένο σε ένα ορθογώνιο πλαστικό φορέα επενδυμένο με ύφασμα που απομακρύνει τα σωματίδια της σκόνης. Οι δισκέτες μπορούν να διαβαστούν και να εγγραφούν από οδηγούς μονάδας δισκέτας(FDD). Οι δισκέτες αρχικά ως 8 ιντσών (200 mm) και αργότερα ως 5,25 ιντσών (133 mm) και 3,5 ιντσών (90 mm), ήταν μια πανταχού παρούσα μορφή της αποθήκευσης και ανταλλαγής δεδομένων από τα μέσα της δεκαετίας του 1970 και μέχρι το 2000.

Μέχρι το 2010, οι μητρικές κάρτες των υπολογιστών σπανίως κατασκευάζονταν με υποστήριξη για οδηγούς μονάδας δισκέτας. Οι 3,5 ιντσών δισκέτες μπορούν να χρησιμοποιηθούν με μια εξωτερική μονάδα δισκέτας USB, αλλά οδηγοί USB για δισκέτες των 5,25 ιντσών και 8 ιντσών είναι σπάνιοι ή ανύπαρκτοι, έτσι τέτοιοι τύποι δισκετών πρέπει συνήθως να χειρίζονται από παλαιότερο εξοπλισμό.

Ενώ οι μονάδες δισκέτας έχουν ακόμη κάποιες περιορισμένες χρήσεις, ειδικά με την κληρονομιά του βιομηχανικού εξοπλισμού ηλεκτρονικών υπολογιστών, έχουν

εκτοπιστεί από τις νεότερες μεθόδους αποθήκευσης δεδομένων με πολύ μεγαλύτερη χωρητικότητα, όπως μονάδες flash USB, φορητές εξωτερικές μονάδες σκληρού δίσκου, οπτικοί δίσκοι, κάρτες μνήμης υπολογιστών και δικτύων .

### 3.2 Ιστορία της δισκέτας

Κατά τη διάρκεια της πορείας της δισκέτας, χρησιμοποιήθηκε ένας αριθμός από διαφορετικές μορφές. Τα μεγέθη κυμαίνονται από 8 ίντσες στην πρώτη εμπορευματοποιημένη δισκέτα μέχρι και τις σχετικά σύγχρονες δισκέτες 3,5 ιντσών. Οι δισκέτες έχουν πλέον σε μεγάλο βαθμό αντικατασταθεί από άλλα μέσα αποθήκευσης ( κυρίως μονάδες USB flash και το Compact Disc ) και τη μεταφορά αρχείων δικτύου, είτε μέσω LAN ή του Διαδικτύου.

Η δισκέτα κατασκευάστηκε από την αμερικάνικη εταιρεία IBM το 1971 και ήταν διαστάσεων 8 ιντσών με δικαίωμα ανάγνωσης (όχι εγγραφής) και ο αποθηκευτικός της χώρος ήταν της τάξης των 80KB. Στα επόμενα χρόνια η IBM και εταιρείες όπως οι Memorex, Shugart Associates και Burroughs Corporation προσπάθησαν να φτιάξουν μια πιο καλή έκδοση της δισκέτας. Το 1976 η Shugart Associates κατασκεύασε την πρώτη FDD των 5¼ ιντσών, πρότυπο που ακολούθησαν περισσότερες από 10 εταιρείες. Το 1984 η IBM έφερε στην αγορά την πρώτη δισκέτα διπλής όψης χωρητικότητας 1.2 Megabyte, η οποία ωστόσο δεν είχε μεγάλη απήχηση. Δύο χρόνια αργότερα η IBM κυκλοφόρησε μια δισκέτα 3.5 ιντσών και 720KB χωρητικότητα για τους φορητούς υπολογιστές. Μετά από λίγο καιρό κυκλοφόρησαν από την ίδια εταιρεία η δισκέτα χωρητικότητας 1,44MB η οποία και χρησιμοποιήθηκε ευρέως. Συνέχισε λίγα χρόνια αργότερα, το 1988, με μία καινούργια δισκέτα χωρητικότητας 2,88 MB η οποία τελικά ήταν μεγάλη εμπορική αποτυχία.

Η δισκέτα των 1,44MB καθιερώθηκε γιατί εκτός από το μικρότερο μέγεθος, σε σχέση με τις 5¼ , προσέφεραν μεγαλύτερη αντοχή σε παράγοντες όπως η σκόνη, η υγρασία, οι γρατσουνιές, θερμοκρασία κτλ. Τέλος και άλλες εταιρείες στη συνέχεια προσπάθησαν να βγάλουν μια καλύτερη έκδοση της δισκέτας όμως για διάφορους λόγους δεν πέτυχαν εμπορικά. Μερικά παραδείγματα είναι η δισκέτα της Insite Peripherals, το 1991, με το όνομα Floptical και χωρητικότητα 21 MB, η δισκέτα

SuperDisk από τις εταιρείες Matsushita (Panasonic) και Imation, χωρητικότητας 120 MB και ένα χρόνο αργότερο, το 1997, η Sony παρουσίασε την HiFD χωρητικότητας 150MB.

Την δεκαετία του '00, και ειδικά με την είσοδο άλλων αποθηκευτικών μέσων όπως το CD, η χρήση της δισκέτας άρχισε να μειώνεται δραματικά. Με τα χρόνια πολλές εταιρείες σταμάτησαν την παραγωγή κάτι που έκανε τελικά και η Sony το 2011, η μεγαλύτερη εταιρεία σε παραγωγή δισκέτας, και οδήγησε και πρακτικά στο "θάνατο" της.

Η παρουσία των δισκετών έγινε αισθητή την δεκαετία του 1980 και 1990 με τη χρήση τους για ηλεκτρονικούς υπολογιστές για τη διανομή λογισμικού, τη μεταφορά δεδομένων και τη δημιουργία αντιγράφων ασφαλείας. Πριν από τους σκληρούς δίσκους ήταν οικονομικά προσιτές και συχνά χρησιμοποιούνταν για την αποθήκευση του λειτουργικού συστήματος των υπολογιστών. Οι περισσότεροι υπολογιστές στο σπίτι είχαν ένα κύριο λειτουργικό σύστημα που αποθηκευόταν ως ROM, με την δυνατότητα να φορτώνουν ένα πιο προηγμένο λειτουργικό σύστημα δίσκου από μια δισκέτα. Μέχρι τις αρχές της δεκαετίας του 1990, η αύξηση του μεγέθους του λογισμικού όπως τα Windows ή το Adobe Photoshop που χρειαζόνταν μια ντουζίνα δίσκους ή και ακόμα περισσότερο. Το 1996, υπήρχαν περίπου πέντε δισεκατομμύρια δισκέτες σε χρήση. Στη συνέχεια, η διανομή των μεγαλύτερων προγραμμάτων αντικαταστάθηκε σταδιακά από το CD-ROM και σε απευθείας διαδικτυακή διανομή (για μικρότερα προγράμματα). Μια προσπάθεια για να συνεχιστεί η δισκέτα ήταν ο Superdisk στα τέλη της δεκαετίας του 1990, με χωρητικότητα 120 MB και ήταν συμβατός με το πρότυπο 3,5 ιντσών των δισκετών. Μία "μάχη" σύντομα σημειώθηκε μεταξύ Superdisk και των άλλων υψηλής πυκνότητας αφαιρούμενων δίσκων, αν και τελικά η μνήμη flash, τα CDs και DVDs με δυνατότητα επανεγγραφής, και η αποθήκευση στο διαδίκτυο θα επικρατούσαν. Εξωτερικές μονάδες δισκέτας USB είναι ακόμα διαθέσιμες, και πολλά σύγχρονα συστήματα παρέχουν ειδικό λογισμικό για εκκίνηση από τέτοια μονάδα.

### 3.3 Η δισκέτα στις αρχές του 21<sup>ου</sup> αιώνα

Οι δισκέτες πλέον χρησιμοποιούνται σε επείγουσες φορτώσεις λειτουργικών σε παλαιότερα συστήματα τα οποία δεν έχουν την δυνατότητα για υποστήριξη άλλων νεότερων μέσων, καθώς και για ενημερώσεις BIOS από τη στιγμή που τα περισσότερα BIOS και λειτουργικά μπορούν να συνεχίσουν να εκτελούνται μέσα από τις δισκέτες. Αν οι ενημερώσεις BIOS αποτύχουν, μονάδες δισκέτας μπορούν να χρησιμοποιηθούν για να επιτευχθεί ανάκτηση συστήματος. Οι βιομηχανίες του θεάτρου και της μουσικής εξακολουθούν να χρησιμοποιούν εξοπλισμό που απαιτεί δισκέτα (π.χ. συνθεσάιζερ, δειγματολήπτες, μηχανές τυμπάνων, sequencers, και κονσόλες φωτισμού).

### **3.4 Χαρακτηριστικά των Οδηγών**

#### **3.4.1 Κεφαλές ανάγνωσης / εγγραφής**

Οι κεφαλές ανάγνωσης/εγγραφής ( Read/Write Heads ) χρησιμοποιούνται για τη μετατροπή των ψηφιακών δεδομένων σε ηλεκτρομαγνητικούς παλμούς, κατά το γράψιμο, ενώ το αντίστροφο συμβαίνει κατά την ανάγνωση δεδομένων από τις δισκέτες. Κάτι αντίστοιχο συμβαίνει και με τις κεφαλές ανάγνωσης / εγγραφής των σκληρών δίσκων. Παρόλες τις ομοιότητες μεταξύ των κεφαλών ανάγνωσης/εγγραφής των οδηγών εύκαμπτων δίσκων και των σκληρών δίσκων, υπάρχουν και αρκετές διαφορές. Οι κεφαλές των οδηγών εύκαμπτων δίσκων είναι μεγαλύτερες και λιγότερο ακριβείς από τις κεφαλές των σκληρών δίσκων, αφού η πυκνότητα των ιχνών στις δισκέτες είναι πολύ μικρότερη απ' ό τι στους σκληρούς δίσκους. Οι σκληροί δίσκοι έχουν πυκνότητα αρκετών εκατοντάδων ιχνών ανά ίντσα, ενώ η αντίστοιχη πυκνότητα στις δισκέτες είναι μόνο 135 τομέων ανά ίντσα.

Η κατασκευή των κεφαλών στους οδηγούς εύκαμπτων δίσκων ακολουθεί την παλιά τεχνολογία, που είχε χρησιμοποιηθεί και στους παλιότερους σκληρούς δίσκους. Στην πραγματικότητα, η κεφαλή αποτελείται από ένα μεταλλικό πυρήνα που επικαλύπτεται από αγώγιμο καλώδιο, μαζί φυσικά με τα κατάλληλα κυκλώματα για τη μετατροπή των ηλεκτρομαγνητικών δεδομένων σε ψηφιακά και το αντίστροφο. Οι κεφαλές των

οδηγών εύκαμπτων δίσκων εφάπτονται πάνω στην επιφάνεια της δισκέτας σε αντίθεση με τις κεφαλές στους σκληρούς δίσκους. Έτσι, η μεταφορά δεδομένων από και προς τους εύκαμπτους δίσκους είναι πιο αξιόπιστη και χρησιμοποιεί απλούστερη τεχνολογία σε σχέση με την τεχνολογία στις κεφαλές των σκληρών δίσκων. Εξαιτίας της επαφής των κεφαλών με την επιφάνεια της δισκέτας, η ταχύτητα περιστροφής κυμαίνεται μεταξύ 300 - 360 στροφές ανά λεπτό, σε αντίθεση με τους σκληρούς δίσκους, οι οποίοι περιστρέφονται με ταχύτητα περίπου 5000 στροφών ανά λεπτό (RPM / Rotations Per Minute). Η μικρή ταχύτητα περιστροφής των εύκαμπτων δίσκων οφείλεται στο γεγονός ότι πρέπει να προστατεύονται τόσο η επιφάνεια των δισκετών, όσο και οι κεφαλές από φθορές που έχουν σχέση με την επαφή και τη γρήγορη περιστροφή. Ωστόσο, παρόλες τις προφυλάξεις υπάρχει φθορά πάνω στις κεφαλές καθώς επικολλούνται μαγνητικά οξείδια και ακαθαρσίες, με αποτέλεσμα να απαιτείται ο περιοδικός καθαρισμός τους.

Εκτός από τις κλασικές κεφαλές για ανάγνωση/εγγραφή, οι οδηγοί εύκαμπτων δίσκων διαθέτουν επιπρόσθετα και ειδικά σχεδιασμένες κεφαλές που ονομάζονται tunnel-erase heads. Είναι τοποθετημένες πίσω και σε κάθε πλευρά των κεφαλών ανάγνωσης/εγγραφής. Σκοπός τους είναι να “καθαρίζουν” κάθε “αδέσποτη” μαγνητική πληροφορία πάνω στην επιφάνεια των δισκετών. Συγκεκριμένα, οι κεφαλές ανάγνωσης/εγγραφής μπορεί να εγγράψουν λανθασμένα πληροφορίες εκτός των ορίων των ιχνών. Είναι όμως απαραίτητο το κάθε ίχνος να είναι ανεξάρτητο από τα άλλα, διαφορετικά μπορεί να υπάρξει πρόβλημα αναγνώρισής τους. Έτσι, οι κεφαλές tunnel-erase heads σβήνουν τις άσκοπες μαγνητικές πληροφορίες, κρατώντας “καθαρά” τα ίχνη.

Όλες οι σύγχρονες δισκέτες διαθέτουν δυο επιφάνειες για αποθήκευση δεδομένων, σε αντίθεση με κάποιες παλιές δισκέτες που είχαν μια επιφάνεια μόνο. Εξαιτίας αυτού του γεγονότος, υπάρχουν κεφαλές ανάγνωσης/εγγραφής για κάθε πλευρά της δισκέτας.

### **3.4.2 Μηχανισμός κίνησης κεφαλών**

Ο μηχανισμός κίνησης (head actuator) των κεφαλών τοποθετεί τις κεφαλές ανάγνωσης/εγγραφής πάνω στα σωστά ίχνη, στην επιφάνεια της δισκέτας. Ο μηχανικός βραχίονας, στην άκρη του οποίου είναι τοποθετημένη η κεφαλή, μεταβάλλει τη θέση της κεφαλής με τη βοήθεια ενός μηχανικού εξαρτήματος που ονομάζεται βηματικός κινητήρας ( stepper motor ). Ο βηματικός κινητήρας μετακινεί το βραχίονα, άρα και την κεφαλή προς το κέντρο ή την περιφέρεια της επιφάνειας της δισκέτας, έτσι ώστε σε κάθε απλή μετακίνηση, η κεφαλή να συναντά το επόμενο ίχνος. Η τεχνολογία των βηματικών κινητήρων αρχικά εφαρμόστηκε στους σκληρούς δίσκους, όμως γρήγορα αντικαταστάθηκε εξαιτίας του φαινομένου της θερμικής διαστολής της επιφανείας των σκληρών δίσκων. Επειδή η πυκνότητα των ιχνών στους σκληρούς δίσκους είναι μεγάλη και ο βηματικός κινητήρας μετακινούσε την κεφαλή σε διαδοχικά ίχνη, μια θερμική διαστολή, είχε ως αποτέλεσμα να τοποθετείται η κεφαλή σε διαφορετικό ίχνος από το πραγματικό. Στους εύκαμπτους δίσκους, όπου η πυκνότητα των ιχνών είναι πολύ μικρότερη, μια θερμική διαστολή δεν μπορεί να μπερδέψει δυο διαδοχικά ίχνη. Εξαιτίας του γεγονότος αυτού, η τεχνολογία των βηματικών κινητήρων εξακολουθεί να χρησιμοποιείται στους οδηγούς εύκαμπτων δίσκων, ενώ έχει πάψει να χρησιμοποιείται στους σκληρούς δίσκους.

Υπάρχει περίπτωση ο βραχίονας και κατά συνέπεια η κεφαλή, να μην μετακινείται ακριβώς πάνω στα προκαθορισμένα ίχνη. Αυτό είναι γνωστό ως πρόβλημα ευθυγράμμισης των κεφαλών ( head alignment problem ). Όταν οι κεφαλές δεν είναι ευθυγραμμισμένες με τα ίχνη, παρουσιάζεται το εξής πρόβλημα: αν κάνουμε format τη δισκέτα, μπορούμε να γράψουμε και να διαβάσουμε δεδομένα χωρίς πρόβλημα, αφού το φορμάρισμα δημιουργεί ίχνη που είναι ακριβώς προσπελάσιμα από τις κεφαλές του οδηγού στον οποίο έγινε το φορμάρισμα. Αν όμως προσπαθήσουμε να γράψουμε ή να διαβάσουμε από τη δισκέτα με τη βοήθεια ενός άλλου οδηγού, θα υπάρξει πρόβλημα, αφού οι κεφαλές του δε θα προσπελαύνουν ακριβώς τα διαφορετικής, από τα προκαθορισμένα, ακτίνας ίχνη. Παλιότερα, το πρόβλημα ευθυγράμμισης των κεφαλών απαιτούσε την απομάκρυνση του οδηγού και την επέμβαση κάποιου ειδικού μηχανικού. Σήμερα το κόστος αγοράς ενός καινούργιου οδηγού είναι τόσο μικρό, που συμφέρει η αγορά νέου οδηγού. Ο μηχανισμός μετακίνησης των κεφαλών στους οδηγούς εύκαμπτων δίσκων είναι πολύ πιο αργός, σε σχέση με τον αντίστοιχο μηχανισμό στους σκληρούς δίσκους. Έτσι, ο χρόνος που απαιτείται για να μετακινηθεί η κεφαλή στους σκληρούς δίσκους, από το πλέον



εσωτερικό ίχνος στο άκρως εξωτερικό, είναι περίπου 20 milliseconds, ενώ στους οδηγούς εύκαμπτων δίσκων είναι σχεδόν ο δεκαπλάσιος χρόνος. Αυτός είναι και ένας από τους λόγους, εξαιτίας των οποίων οι σκληροί δίσκοι είναι πολύ πιο γρήγοροι από τους οδηγούς δισκετών.

### 3.4.3 Μηχανισμός περιστροφής

Όταν η δισκέτα εισέρχεται μέσα στον οδηγό εύκαμπτου δίσκου ένας σφιχτήρας (clamp) "γραπώνει" τη δισκέτα από τις ειδικές οπές που διαθέτει στο κέντρο της. Ο σφιχτήρας καταλήγει σε έναν κινητήρα περιστροφής (spindle motor), ο οποίος καθώς περιστρέφει το σφιχτήρα, αναγκάζει και την επιφάνεια της δισκέτας να περιστρέφεται γύρω από αυτόν. Η ταχύτητα περιστροφής εξαρτάται από τον τύπο του οδηγού και μετριέται σε στροφές ανά λεπτό (**RPM**). Μερικές ενδεικτικές ταχύτητες περιστροφής παρατίθενται στον παρακάτω πίνακα:

**Πίνακας 1.** Ενδεικτικές ταχύτητες περιστροφής

Τύπος δισκέτας	Ταχύτητα περιστροφής
360 KB 5.25"	300 RPM
1.2 MB 5.25"	360 RPM
720 KB 3.5"	300 RPM
1.44 MB 3.5"	300 RPM
2.88 MB 3.5"	300 RPM

Η μικρή ταχύτητα περιστροφής είναι ένας ακόμα λόγος, εξαιτίας του οποίου η απόδοση, σε χρόνο, των δισκετών είναι πολύ μικρή σε σχέση με άλλα αποθηκευτικά μέσα. Ο χρόνος απόκρισης και μεταφοράς δεδομένων από και προς τη δισκέτα, επηρεάζεται πάρα πολύ από την ταχύτητα περιστροφής. Πρέπει να σημειωθεί, ότι σε αντίθεση με τους σκληρούς δίσκους, όπου οι κεφαλές αιωρούνται λίγο πάνω από την επιφάνεια, οι κεφαλές των οδηγών δισκετών εφάπτονται στις δισκέτες. Το αποτέλεσμα είναι ότι υπάρχει φθορά, τόσο στις κεφαλές, όσο και στην επιφάνεια της δισκέτας. Για να μειωθεί η φθορά, ο μηχανισμός κίνησης των κεφαλών αποσύρει τις κεφαλές και σταματά η περιστροφή της δισκέτας όταν ο οδηγός δε διαβάζει ή γράφει. Κατά συνέπεια, όταν δοθεί η επόμενη εντολή για διάβασμα ή γράψιμο, υπάρχει μια καθυστέρηση μισού περίπου δευτερολέπτου μέχρι ο κινητήρας περιστροφής να αποκτήσει την απαιτούμενη ταχύτητα.

#### **3.4.4 Αισθητήρας αλλαγής δισκέτας (disk change sensor)**

Οι σύγχρονοι οδηγοί δισκετών περιέχουν έναν ειδικό αισθητήρα (disk-change sensor), ο οποίος στέλνει ένα σήμα στον ελεγκτή του εύκαμπτου δίσκου (floppy controller) όταν βγάζουμε μια δισκέτα από τον οδηγό και εισάγουμε μια νέα. Με τον τρόπο αυτό, αυξάνεται η απόδοση του οδηγού, αφού ο ελεγκτής είναι ενήμερος όταν τοποθετείται νέα δισκέτα μέσα στον οδηγό. Ο ελεγκτής του εύκαμπτου δίσκου, γνωρίζοντας πότε δεν έχει γίνει αλλαγή δισκέτας, αποφεύγει την επανεξέταση της δομής της κάθε φορά που γίνεται αναφορά στον οδηγό. Βλάβες στα κυκλώματα του αισθητήρα αλλαγής δισκέτας μπορούν να προκαλέσουν παράξενα συμπτώματα στη λειτουργία του οδηγού. Το πιο συνηθισμένο πρόβλημα παρουσιάζεται όταν αλλάξουμε δισκέτα και το σύστημα δεν αναγνωρίζει την αλλαγή. Έτσι, όταν προσπαθούμε να αναφερθούμε σε ένα αρχείο της νέας δισκέτας, το σύστημα αντιδρά πληροφορώντας ότι δεν υπάρχει τέτοιο αρχείο. Στην πραγματικότητα το αρχείο υπάρχει, αλλά ο ελεγκτής του εύκαμπτου δίσκου γνωρίζει μόνο τα περιεχόμενα της παλιάς δισκέτας. Επίσης, αν προσπαθήσουμε να γράψουμε στη νέα δισκέτα, το μόνο που θα καταφέρουμε είναι να μπερδέψουμε τα περιεχόμενα της, αφού ο ελεγκτής του εύκαμπτου δίσκου νομίζει ότι θέλουμε να γράψουμε στην παλιά δισκέτα.

Όλες οι σύγχρονες δισκέτες διαθέτουν δύο επιφάνειες για αποθήκευση δεδομένων, σε αντίθεση με κάποιες παλιές δισκέτες που είχαν μια επιφάνεια μόνο. Εξαιτίας αυτού του γεγονότος υπάρχουν κεφαλές ανάγνωσης / εγγραφής για κάθε πλευρά της δισκέτας.

#### **3.4.5 Συνδετήρες και γέφυρες βραχυκύκλωσης (connectors και jumpers).**

Για τη σύνδεση του οδηγού εύκαμπτου δίσκου με τον υπολογιστή, χρησιμοποιούνται δύο τύποι συνδετήρων. Επίσης, υπάρχουν jumpers (γέφυρες βραχυκύκλωσης), με τη βοήθεια των οποίων μπορούμε να κάνουμε τις σωστές ρυθμίσεις (configuration) του οδηγού, αλλά στην πράξη σπάνια χρησιμοποιούνται. Ο drive select (DS) jumper, χρησιμοποιείται για να επιλέξουμε αν ο οδηγός θα είναι A: ή B:, αν και το πρότυπο είναι να χρησιμοποιήσουμε ένα καλώδιο (floppy cable) για να ελέγχουμε ποιος οδηγός είναι ο A: και ποιος ο B:. Ο συνδετήρας ρεύματος (power connector) είναι ο ένας από τους δύο συνδετήρες και τροφοδοτεί με ρεύμα τον οδηγό. Οι παλιοί οδηγοί των 5.25", χρησιμοποιούν ως συνδετήρα ρεύματος το πρότυπο συνδετήρα 4 συρμάτων (4 wire connector), όπου βρίσκεται εφαρμογή σε σκληρούς δίσκους, CD-ROMS, καθώς και σε άλλους οδηγούς. Οι περισσότεροι οδηγοί δισκετών των 3.5" χρησιμοποιούν τον μικρής ισχύος συνδετήρα (mini power connector). Μπορούμε με τη βοήθεια προσαρμοστών (adaptors) να μετατρέψουμε τους οδηγούς, έτσι ώστε να δέχονται συνδετήρα μικρής ισχύος. Ο δεύτερος συνδετήρας, χρησιμοποιείται για τη μεταφορά των δεδομένων από και προς τον οδηγό. Το πρότυπο είναι ένας ειδικός 34-pin συνδετήρας.

#### **3.4.6 Κυκλώματα του οδηγού δισκέτας**

Κάθε οδηγός δισκετών περιέχει ολοκληρωμένα κυκλώματα που ελέγχουν τις λειτουργίες του. Έτσι οι κεφαλές ανάγνωσης / εγγραφής, το spindle motor, ο head actuator καθώς και άλλα μέρη του οδηγού ελέγχονται από αυτά τα κυκλώματα. Υπάρχει επίσης και ένα SCSI interface chip προκειμένου να μπορεί ο οδηγός να αναγνωρίζει το πρότυπο διασύνδεσης SCSI.

## 3.5 Η κατασκευή τους

### 3.5.1 Εύκαμπτοι δίσκοι 5.25"

Οι πρώτες δισκέτες που παρουσιάστηκαν, είχαν διάμετρο 8" και στη συνέχεια εμφανίστηκαν οι δισκέτες των 5.25". Κάθε δισκέτα 5.25" αποτελείται από τον πραγματικό δίσκο και το προστατευτικό περίβλημα. Ο δίσκος είναι φτιαγμένος από λεπτό εύκαμπτο πλαστικό, το οποίο έχει επιστρωθεί με μαγνητικό υλικό. Στο κέντρο υπάρχει μια τρύπα που χρησιμεύει στο να “γραπώνει” ο οδηγός το δίσκο, προκειμένου να τον περιστρέψει (φυσικά το πλαστικό περίβλημα δεν περιστρέφεται). Το πλαστικό περίβλημα έχει μια οπή, έτσι ώστε να μένει ο δίσκος εκτεθειμένος στις κεφαλές ανάγνωσης / εγγραφής. Η οπή είναι τόσο μεγάλη, ώστε να μπορεί ο μηχανισμός κίνησης κεφαλής (head actuator) να μετακινεί τις κεφαλές σε όλα τα ίχνη και να μπορούν αυτές να διαβάσουν ή να γράψουν σε ένα ίχνος. Μια εγκοπή μαζί με το κάλυμμα της παίζουν τον ρόλο του διακόπτη του ρυθμίζει αν τα περιεχόμενα της δισκέτας προστατεύονται (write-protect) από κάθε απόπειρα μεταβολής τους.

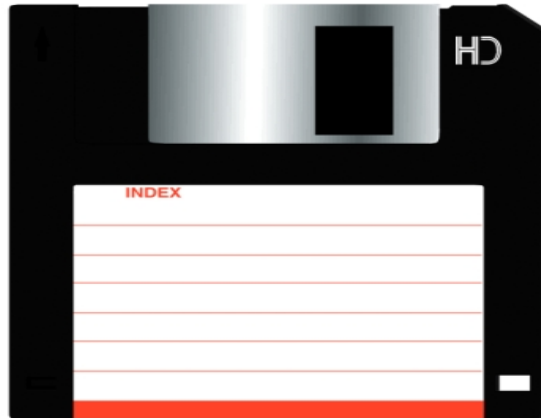


### **Εικόνα 3.1 Δισκέτα 5.25"**

Το πλαστικό περίβλημα των δισκετών των 5.25" δεν προσφέρει ικανοποιητική προστασία στο δίσκο. Το γεγονός αυτό, σε συνδυασμό με το μεγάλο μέγεθος της δισκέτας, έχει ως αποτέλεσμα να μπορούν να καμφθούν εύκολα και άρα να είναι επιρρεπής στις καταστροφές. Οι δισκέτες πρέπει να φυλάσσονται μέσα στο χάρτινο προστατευτικό περίβλημα και σε ειδικούς χώρους. Επίσης, το γεγονός ότι η επιφάνεια του δίσκου είναι εν μέρει εκτεθειμένη στο εξωτερικό περιβάλλον έχει ως αποτέλεσμα να υπάρχει μεγάλος κίνδυνος καταστροφών από ποικίλους παράγοντες. Πρέπει να σημειωθεί, ότι αν γράψουμε με ένα στυλό, που διαθέτει σφαιρική άκρη, πάνω στο πλαστικό περίβλημα, τότε υπάρχει κίνδυνος καταστροφής του δίσκου. Η έλλειψη αντοχής και το μεγάλο σχετικά μέγεθος ήταν μερικές από τις βασικές αιτίες που δε χρησιμοποιούνται πλέον οι δισκέτες των 5.25".

#### **3.5.2 Εύκαμπτοι δίσκοι 3.5"**

Οι δισκέτες των 3.5" προέκυψαν από τις δισκέτες των 5.25" με κάποιες βελτιώσεις, που έγιναν στις τελευταίες προκειμένου να αυξηθεί η αντοχή τους. Το πλαστικό περίβλημα είναι πολύ πιο σκληρό με αποτέλεσμα η δισκέτα να είναι πιο ανθεκτική. Η οπή που χρησιμοποιείται από τις κεφαλές ανάγνωσης / εγγραφής έχει πλέον καλυφθεί με μια συρταρωτή "πόρτα" που ανοίγει όταν η δισκέτα βρίσκεται μέσα στον οδηγό. Με τον τρόπο αυτό, προστατεύουμε το δίσκο και από κάθε λογής εξωτερικό παράγοντα, αφού πλέον δεν είναι άμεσα εκτεθειμένος στο εξωτερικό περιβάλλον. Επίσης, το μέγεθος της δισκέτας είναι σαφώς μικρότερο (3.5"). Η εγκοπή προστασίας κατά του γραψίματος έχει αντικατασταθεί από μια συρταρωτή πόρτα. Όταν η πόρτα είναι ανοιχτή ( οπή ανοιχτή ), η δισκέτα είναι write-protected και το αντίστροφο. Η αλλαγή κατάστασης ( write-protected $\Leftrightarrow$ write enabled ) γίνεται εύκολα, με το άνοιγμα ή το κλείσιμο της πόρτας. Η μεγάλη τρύπα στο κέντρο των δισκετών των 5.25", έχει αντικατασταθεί από ένα μικρό μεταλλικό δίσκο με εσοχές, που αυξάνει την ανθεκτικότητα των δισκετών των 3.5".



**Εικόνα 3.2** Δισκέτα 3.5"

### 3.5.3 Πυκνότητα επιφανείας

Η πυκνότητα της επιφανείας των δισκετών, έχει να κάνει με την ποσότητα των δεδομένων που μπορούν να αποθηκεύσουν σε μια ορισμένη επιφάνεια. Η πυκνότητα εξαρτάται από δυο παράγοντες: πόσα ίχνη χωράνε στη δισκέτα ( track density ) και πόσα bits μπορεί να περιέχει το κάθε ίχνος ( bit density ). Ο κατάλληλος συνδυασμός των δυο παραγόντων ονομάζεται πυκνότητα επιφανείας ή πυκνότητα περιοχής ( area density ). Ο όρος πυκνότητα περιοχής χρησιμοποιείται κυρίως για να εκφράσει την πυκνότητα των σκληρών δίσκων. Για να εκφράσουμε την πυκνότητα των δισκετών χρησιμοποιούμε είτε την πυκνότητα των ιχνών ( track density : εκφρασμένη σε tracks per inch / TPI ), είτε την πυκνότητα των bits ( bit density : εκφρασμένη σε bits per inch / BPI ).

Η χαμηλότερης πυκνότητας δισκέτες (lower – density) χαρακτηρίζονται ως “double density” (διπλής πυκνότητας). Εκ πρώτης όψεως αυτό φαίνεται αντιφατικό, αλλά στην πραγματικότητα οι δισκέτες αυτές είναι οι διάδοχοι των πρώτων δισκετών που είχαν ακόμα μικρότερη πυκνότητα. Ο όρος double density συγκρίνει τις δισκέτες

αυτές με τις σαφώς μικρότερης πυκνότητας προγενέστερες δισκέτες. Δεύτερον, αν και για τα δυο είδη δισκετών (3.5" και 5.25"), χρησιμοποιούνται οι όροι "double density" και "high density" η ερμηνεία τους για κάθε είδος δισκέτας (3.5" και 5.25"), είναι τελείως διαφορετική. Τρίτον, διαφορές στην πυκνότητα μπορούν να οφείλονται είτε σε διαφορές στην πυκνότητα των ιχνών, είτε στην πυκνότητα των bits, είτε και στα δύο. Παρεμπιπτόντως, όλες οι δισκέτες των 3.5" έχουν την ίδια πυκνότητα ιχνους (135 TPI). Οι δισκέτες "double density", έχουν συνήθως τυπωμένο πάνω τους το διακριτικό "DD", ενώ οι δισκέτες "high density" φέρουν το διακριτικό "HD". Μερικοί άνθρωποι πιστεύουν ότι οι δισκέτες 3.5" HD, όσο και οι δισκέτες 3.5" DD, έχουν τον ίδιο αριθμό ιχνών ανά ίντσα και η μόνη διαφορά μεταξύ τους έγκειται στην ποιότητά τους. Στην πραγματικότητα, όσο υψηλότερη πυκνότητα χρησιμοποιούμε, τόσο τα δεδομένα είναι πιο κοντά μεταξύ τους και τόσο μεγαλύτερος είναι ο κίνδυνος να συμπέσουν μεταξύ τους. Επομένως, η υψηλότερη πυκνότητα (δισκέτες HD) απαιτεί πιο αδύνατο σήμα εγγραφής και διαφορετικό υλικό επικάλυψης, από ότι απαιτεί η χαμηλότερη πυκνότητα (δισκέτες DD). Τέλος, θα πρέπει να επισημάνουμε ότι τις δισκέτες HD μπορούν να τις χειριστούν μόνο οι high-density οδηγοί, ενώ τις δισκέτες DD τις χειρίζονται τόσο οι double-density οδηγοί, όσο και οι lower-density οδηγοί.

#### **3.5.4 Κωδικοποίηση - Αποκωδικοποίηση δεδομένων**

Η κωδικοποίηση των δεδομένων στις δισκέτες είναι η διαδικασία μετατροπής των ψηφιακών πληροφοριών σε μαγνητικές πληροφορίες οι οποίες αποθηκεύονται στην επιφάνεια του δίσκου. Οι δισκέτες, για την κωδικοποίηση των δεδομένων χρησιμοποιούν την τεχνολογία της τροποποιημένης διαμόρφωσης συχνότητας (modified frequency modulation / MFM).

### 3.5.5 Μορφοποίηση χαμηλού και υψηλού επιπέδου ( low-level και high-level formatting )

Η διαδικασία της μορφοποίησης ( formatting ), της μαγνητικής επιφανείας των δισκετών και σκληρών δίσκων, περιλαμβάνει δύο βήματα. Το πρώτο βήμα έχει να κάνει με τη δημιουργία της πραγματικής δομής στην επιφάνεια του δίσκου, που κρατά τα δεδομένα. Αυτό σημαίνει τη χάραξη των ίχνων και το μαρκάρισμα της αρχής του κάθε τομέα και ίχνους. Το βήμα αυτό είναι γνωστό και ως μορφοποίηση χαμηλού επιπέδου ( low-level formatting ), ενώ πολλές φορές αποκαλείται και πραγματική μορφοποίηση “true formatting”.

Το δεύτερο βήμα κατά τη διαδικασία της μορφοποίησης, ονομάζεται μορφοποίηση υψηλού επιπέδου ( high-level formatting ). Κατά τη διάρκεια του δευτέρου βήματος, δημιουργούνται οι λογικές δομές του δίσκου, όπως το FAT ([http://www.it.uom.gr/project/mycomputer/storage/file\\_sys/dom\\_diam.html](http://www.it.uom.gr/project/mycomputer/storage/file_sys/dom_diam.html) (File Allocation Table) καθώς και ο ριζικός κατάλογος. Η υψηλού επιπέδου μορφοποίηση, χρησιμοποιώντας τη δομή που δημιούργησε η χαμηλού επιπέδου μορφοποίηση, προετοιμάζει το δίσκο, στο να κρατά και να χειρίζεται τα αρχεία με βάση το επιλεγμένο σύστημα αρχείων.

Για τη μορφοποίηση των σκληρών δίσκων μεσολαβεί ένα ενδιάμεσο βήμα που είναι γνωστό ως διαμέριση. Το γεγονός αυτό, συνδυαζόμενο με την απίστευτη πολυπλοκότητα των νέων σκληρών δίσκων, οδήγησε στην επιλογή του να κάνει τη μορφοποίηση χαμηλού επιπέδου ο κατασκευαστής και η μορφοποίηση υψηλού επιπέδου να γίνεται από το χρήστη, με τη βοήθεια μιας εντολής του λειτουργικού συστήματος ( για το DOS είναι η εντολή FORMAT ). Οι δισκέτες δεν απαιτούν την διαδικασία της διαμέρισης, εξαιτίας της απλούστερης κατασκευής τους και τα δυο βήματα της μορφοποίησης ( χαμηλού επιπέδου και υψηλού επιπέδου), εκτελούνται όταν χρησιμοποιούμε την κατάλληλη εντολή FORMAT του λειτουργικού συστήματος.

Από τη στιγμή που έχει τελειώσει η χαμηλού επιπέδου μορφοποίηση των δισκετών, έχουν “τυπωθεί” όλα τα ίχνη στην επιφάνειά της. Το stepper motor τοποθετεί το μηχανισμό κίνησης κεφαλών πάνω στα ίχνη, έτσι ώστε οι κεφαλές ανάγνωσης / εγγραφής να μπορούν να επιτελέσουν το έργο τους. Αν οι κεφαλές παρεκκλίνουν από



την ακριβή θέση τους, να μην θα γίνει το φορμάρισμα της δισκέτας, αλλά θα μπορεί να διαβάσει ή να γράψει δεδομένα πάνω της μόνο ο οδηγός που έκανε το φορμάρισμα. Αυτό οφείλεται στο γεγονός ότι τα τυπωμένα ίχνη δε βρίσκονται στη προδιαγεγραμμένη θέση, αλλά έχουν μια παρέκκλιση. Η παρέκκλιση αναγνωρίζεται μόνο από τον οδηγό που την έκανε και μόνο αυτός θα μπορεί να διαβάσει ή να γράψει από τη δισκέτα. Μια προτεινόμενη λύση είναι η επαναμορφοποίηση της δισκέτας από έναν οδηγό που δεν έχει πρόβλημα

### 3.5.6 Γεωμετρία των εύκαμπτων δίσκων

Ο όρος γεωμετρία αναφέρεται στην οργανωτική δομή των δεδομένων. Έτσι, για τις δισκέτες, η γεωμετρία έχει να κάνει με τον αριθμό των επιφανειών αποθήκευσης, τον αριθμό των ιχνών ανά επιφάνεια και τον αριθμό των τομέων ανά ίχνος. Σε όλες τις δισκέτες υπάρχει ο ίδιος αριθμός τομέων ανά ίχνος, παρόλο που τα πλέον εσωτερικά ίχνη είναι πολύ πιο μικρά από τα άκρως εξωτερικά. Από τη στιγμή που όλες οι νεότερες δισκέτες έχουν δυο επιφάνειες, ο μόνος σημαντικός παράγοντας είναι ο αριθμός των τομέων και των ιχνών. Παρακάτω παρουσιάζεται ένας πίνακας με τις γεωμετρίες για διάφορα είδη μορφοποίησης

**Πίνακας 2.** Γεωμετρία δισκετών

Γεωμετρικές Προδιαγραφές	Tracks	Τομείς ανά Track	
360 KB 5.25"	40		9
1.2 MB 5.25"	80		15
720 KB 3.5"	80		9
1.44 MB 3.5"	80		18
2.88 MB 3.5"	80		36

---

Όλες οι δισκέτες, καθώς και οι νεότεροι σκληροί δίσκοι περιέχουν 512 bytes ανά τομέα. Τέλος, πρέπει να σημειωθεί ότι δεν είναι διαθέσιμοι όλοι οι τομείς για αποθήκευση πραγματικών δεδομένων, αφού κάποια από αυτά χρησιμοποιούνται για την αποθήκευση της δομής των αρχείων.

### **3.6 Διασύνδεση δισκετών**

#### **3.6.1 Εισαγωγή**

Ο ελεγκτής του εύκαμπτου δίσκου είναι ένα είδος hardware, που συνδέει τον οδηγό με το υπολογιστικό σύστημα. Διαχειρίζεται τη ροή των δεδομένων βάσει της χρησιμοποιούμενης διασύνδεσης ( interface ), από τον οδηγό προς τον επεξεργαστή ή τη μνήμη και αντίστροφα. Παρακάτω αναλύονται διάφοροι παράμετροι σχετικά με τη διασύνδεση των εύκαμπτων δίσκων.

#### **3.6.2 Διασύνδεση SCSI οδηγών εύκαμπτων δίσκων**

Υπάρχουν ειδικού τύπου οδηγοί εύκαμπτου δίσκου που συνδέονται απευθείας στο δίαυλο SCSI. Ωστόσο, σπάνια χρησιμοποιείται αυτή η αρχιτεκτονική για προσωπικούς υπολογιστές και τη συναντάμε μόνο σε εξειδικευμένες βιομηχανικές εφαρμογές. Φυσικά, για να είναι δυνατή αυτή η σύνδεση θα πρέπει το σύστημα να διαθέτει ελεγκτή SCSI, ο οποίος να υποστηρίζει οδηγούς αυτού τέτοιου είδους. Η σύνδεση επιτυγχάνεται με τη χρησιμοποίηση ενός ειδικού καλωδίου. Οι σύγχρονοι υπολογιστές αν και χρησιμοποιούν σκληρούς SCSI και διάφορα SCSI περιφερειακά, εντούτοις η τεχνολογία των SCSI δε χρησιμοποιείται για τους οδηγούς δισκέτας. Από τη στιγμή που οι δισκέτες δεν είναι το βασικό μέσο αποθήκευσης ή το βασικότερο μέσο μεταφοράς δεδομένων, είναι πιο αποδοτικό και οικονομικό να χρησιμοποιούμε οδηγούς εύκαμπτων δίσκων που δεν έχουν δυνατότητα σύνδεσης με το δίαυλο SCSI.

Συνήθως οδηγοί εύκαμπτων δίσκων διαθέτουν δικό τους σύστημα διασύνδεσης που ονομάζεται διασύνδεση εύκαμπτου δίσκου ( floppy disk interface). Τέλος, θα πρέπει να πούμε ότι τα κυκλώματα του ελεγκτή εύκαμπτου δίσκου είναι τοποθετημένα πάνω στη μητρική πλακέτα.

### 3.6.3 Υλοποίηση ελεγκτή εύκαμπτου δίσκου

Κάποτε ο ελεγκτής εύκαμπτου δίσκου ήταν μια ξεχωριστή κάρτα και τοποθετούνταν μέσα σε μια υποδοχή ( slot ) επέκτασης της μητρικής πλακέτας. Στη συνέχεια ενσωματώθηκε μέσα σε μια κάρτα που περιελάμβανε και άλλους ελεγκτές ( IDE/ATA σκληρών δίσκων, σειριακών και παράλληλων θυρών επικοινωνίας ). Αυτή η πολυσύνθετη κάρτα χρησιμοποιήθηκε ευρύτατα σε υπολογιστές με αρχιτεκτονική ISA και VLB από το 1990 ως το 1994. Αφού οι ελεγκτές εύκαμπτων δίσκων δεν άλλαξαν ουσιαστικά δομή, το μοντέλο με την πολλαπλής χρήσης κάρτα έγινε το στάνταρ. Στους Pentium και μεταγενέστερους υπολογιστές, που χρησιμοποιούν τεχνολογία διαύλου PCI, σχεδόν πάντα ο ελεγκτής εύκαμπτου δίσκου είναι τοποθετημένος μέσα στο Super I/O chip της μητρικής πλακέτας. Το Super I/O chip περιέχει εκτός του ελεγκτή εύκαμπτου δίσκου και τους ελεγκτές σειριακών/παράλληλων θυρών επικοινωνίας, καθώς και των σκληρών δίσκων IDE/ATA.

### 3.6.4 Ταχύτητα του ελεγκτή εύκαμπτου δίσκου

Οι ελεγκτές εύκαμπτων δίσκων των νέων υπολογιστών υποστηρίζουν όλους τους τύπους των οδηγών εύκαμπτων δίσκων. Ωστόσο, μερικοί παλιότεροι ελεγκτές εύκαμπτων δίσκων δεν μπορούν να αναγνωρίσουν τους καινούργιους τύπους οδηγών. Η απαιτούμενη ταχύτητα που πρέπει να έχουν οι ελεγκτές εύκαμπτου δίσκου, έχει σχέση με την πυκνότητα των bits ( bits ανά ίντσο ). Από τη στιγμή που η υψηλότερη πυκνότητα σημαίνει περισσότερες πληροφορίες στον ίδιο χώρο, απαιτείται μεγαλύτερος ρυθμός μετάδοσης δεδομένων, για να εξασφαλιστεί ότι τα σωστά

δεδομένα θα φτάσουν στο σωστό χρόνο στον παραλήπτη τους. Υπάρχουν βασικά τρεις διαφορετικές ταχύτητες για τους ελεγκτές εύκαμπτου δίσκου:

- **250 Kilobits ανά δευτερόλεπτο:** Είναι η χαμηλότερη ταχύτητα μεταφοράς δεδομένων ενός ελεγκτή εύκαμπτου δίσκου. Πρόκειται για απαρχαιωμένους ελεγκτές που χρησιμοποιούνταν για double density 360 KB 5.25" και 720 KB 3.5" οδηγούς.
- **500 Kilobits ανά δευτερόλεπτο:** Ο ελεγκτής εύκαμπτου δίσκου με ταχύτητα 500 Kilobits, συναντάται σε ένα μεγάλο αριθμό υπολογιστών και υποστηρίζει όλους τους τύπους των οδηγών εύκαμπτων δίσκων εκτός από τον οδηγό για δισκέτες 2.88 MB 3.5". Γνώρισε μεγάλη εμπορική επιτυχία, αλλά τα τελευταία χρόνια άρχισε να αντικαθίσταται από τον ελεγκτή με ρυθμό 1 Megabit/s.
- **1 Megabit ανά δευτερόλεπτο:** Πρόκειται για τη μεγαλύτερη ταχύτητα που μπορεί να έχει ένας ελεγκτής εύκαμπτου δίσκου. Το μοντέλο αυτό χρησιμοποιείται ευρέως στις μέρες μας.

### 3.6.5 Πηγές που χρησιμοποιούνται από τους ελεγκτές εύκαμπτων δίσκων

Σε γενικές γραμμές, οι πηγές του υπολογιστή που χρησιμοποιούνται από τον ελεγκτή εύκαμπτου δίσκου είναι οι ακόλουθοι:

- **Γραμμή διακοπής:** Το στάνταρ IRQ ( Interrupt Request ) για τους ελεγκτές εύκαμπτων δίσκων είναι το 6. Όπως είναι φυσικό οι ελεγκτές που ελέγχουν τη λειτουργία των άλλων περιφερειακών δε θα χρησιμοποιούν το IRQ6.
- **Κανάλι DMA:** Το 2 είναι η στάνταρ τιμή για το κανάλι DMA ( Direct Access Memory Channels ) που χρησιμοποιείται από τους ελεγκτές εύκαμπτων δίσκων.
- **Διευθύνσεις E/E:** Η στάνταρ διεύθυνση E/E για τους ελεγκτές εύκαμπτων δίσκων κυμαίνεται στο διάστημα 3F0-3F7.

### 3.6.6 Παράμετροι του BIOS

Φυσιολογικά, το BIOS κάθε υπολογιστή υποστηρίζει δυο διαφορετικούς τύπους οδηγού δισκετών. Μέσα από το BIOS setup program μπορούμε να ρυθμίσουμε δυο παραμέτρους που έχουν σχέση με τον τύπο του οδηγού A: και B:. Στην πραγματικότητα, οι δυνατές επιλογές για τον τύπο του κάθε οδηγού καθορίζονται από τον αριθμό των διαφορετικών τύπων που μπορεί να υποστηρίξει ο ελεγκτής εύκαμπτου δίσκου. Υπάρχουν και κάποιες άλλες παράμετροι του BIOS που έχουν άμεση σχέση με τους οδηγούς εύκαμπτων δίσκων:

- **Boot Sequence:** Η παράμετρος καθορίζει τη σειρά των οδηγών την οποία θα ακολουθεί το BIOS κατά τη διαδικασία της εκκίνησης προκειμένου να βρει και να φορτώσει το λειτουργικό σύστημα.
- **Floppy Drive Seek:** Η παράμετρος μας παρέχει την δυνατότητα να απενεργοποιούμε τον έλεγχο που ελέγχει την παρουσία ή όχι ενός οδηγού κατά την διαδικασία εκκίνησης του υπολογιστή.
- **Swap Floppy Drives:** Η παράμετρος δίνει τη δυνατότητα για αλλαγή της ταυτότητας των οδηγών εύκαμπτων δίσκων έτσι ώστε ο οδηγός A: να φαίνεται ως B: και το αντίστροφο.

### 3.6.7 Διαμόρφωση εύκαμπτου δίσκου

Από τη στιγμή που υπάρχει μόνο ένα interface και δυο μόνο διαφορετικοί οδηγοί δισκετών, η διαμόρφωση των οδηγών εύκαμπτων δίσκων ( configuration ), είναι σχετικά εύκολη εργασία. Αρκεί να συνδέσουμε τους οδηγούς με τους σωστούς συνδετήρες και φυσικά να γίνει σωστή σύνδεση του κάθε οδηγού με τον αντίστοιχο πάροχο του ρεύματος. Αν δεν τηρήσουμε τους σωστούς κανόνες σύνδεσης θα

δημιουργηθούν προβλήματα, όπως για παράδειγμα να είναι συνεχώς αναμμένο το ενδεικτικό λαμπάκι λειτουργίας (LED), ακόμα και όταν ο οδηγός δεν εργάζεται.

### **3.7 Μέγεθος και χωρητικότητα**

#### **3.7.1 360 KB 5.25" Floppy**

Ο παλαιότερος τύπος μορφοποίησης είναι ο 360 KB 5.25" floppy disk. Αυτός ο τύπος μορφοποίησης είναι ο μοναδικός που χρησιμοποιεί 40 tracks ανά επιφάνεια για την αποθήκευση των δεδομένων των χρηστών, ενώ όλοι οι υπόλοιποι διαθέτουν 80 tracks ανά επιφάνεια. Εξαιτίας του μικρού, συγκριτικά, αριθμού tracks, η μορφοποίηση 360 KB 5.25" floppy disk δίνει τη μικρότερη δυνατή χωρητικότητα. Ο τύπος 360 KB 5.25" floppy disk είναι απαρχαιωμένος και είχε αντικατασταθεί από τον τύπο 1.2 MB 5.25" floppy disk.

#### **3.7.2 1.2 MB 5.25" Floppy**

Η υψηλής πυκνότητας (high density) δισκέτες των 1.2 MB στις 5.25", πρωτοπαρουσιάστηκαν από την IBM, το 1984. Η αύξηση της χωρητικότητας σε σχέση με τις δισκέτες των 360 KB στις 5.25", ανήλθε στο 200%. Οι οδηγοί των 360KB μπορούσαν να διαβάσουν και τις δισκέτες των 1.2 MB αν και περιστασιακά εμφανίζονταν μερικά προβλήματα. Από τη στιγμή που οι δισκέτες 1.2 MB των 5.25" είχαν μεγαλύτερη πυκνότητα, απαιτούσαν έναν ελεγκτή εύκαμπτου δίσκου με προδιαγραφές μεταφοράς δεδομένων γύρω στα 500 Kbits/sec. Σχεδόν όλοι οι οδηγοί δισκετών υποστηρίζουν αυτό τον αριθμό μεταφοράς δεδομένων. Όταν πρωτοπαρουσιάστηκαν οι δισκέτες των 1.2 MB στις 5.25", η χωρητικότητά τους θεωρούνταν ικανοποιητική, σε σχέση με τη χωρητικότητα των σκληρών δίσκων εκείνης της εποχής, που έφτανε περίπου στα 20 MB. Πάντως οι δισκέτες των 5.25"

αντικαταστάθηκαν από τις δισκέτες των 3.5", εξαιτίας των φυσικών πλεονεκτημάτων των δεύτερων.

### **3.7.3 720 KB 3.5" Floppy**

Η αρχική έκδοση των δισκετών με διάμετρο 3.5", είχε χωρητικότητα 720 KB και πρωτοπαρουσιάστηκε το 1986. Αυτή η έκδοση των 3.5" δεν έγινε ποτέ δημοφιλής εξαιτίας, τόσο της μειωμένης χωρητικότητας της κατά 40% σε σχέση με την δισκέτα των 1.2 MB στις 5.25", όσο και εξαιτίας της γρήγορης εισόδου των δισκετών 3,5" με υψηλή πυκνότητα, που τις αντικατέστησε γρήγορα. Οι υπολογιστές που κατασκευάστηκαν πριν το 1985, δεν υποστηρίζουν τις δισκέτες των 3.5", εξαιτίας διάφορων περιορισμών που έθετε το BIOS.

### **3.7.4 1.44 MB 3.5" Floppy**

Ο μοναδικός τύπος δισκετών που εξακολουθούν να χρησιμοποιούνται μέχρι σήμερα, είναι οι δισκέτες 1.44 MB των 3.5" οι οποίες πρωτοπαρουσιάστηκαν το 1987 από την IBM. Έγιναν μέσα σε μικρό χρονικό διάστημα ιδιαίτερα δημοφιλείς και αποτελούν σήμερα πρότυπο. Σχεδόν όλοι οι υπολογιστές που κατασκευάστηκαν μετά το 1987, διαθέτουν τουλάχιστον έναν οδηγό των 1.44 MB των 3.5". Η τιμή τους έπεσε πάρα πολύ εξαιτίας της απλής τεχνολογίας τους και της μεγάλης παραγωγής. Επειδή η πυκνότητα των δισκετών των 1.44 MB στις 3.5" είναι μεγαλύτερη από την αντίστοιχη των δισκετών των 720 KB στις 3.5", απαιτούσαν έναν ελεγκτή εύκαμπτου δίσκου με προδιαγραφές μεταφοράς δεδομένων κοντά στα 500 Kbits/sec. Φυσικά οι δισκέτες των 720 KB στις 3.5", εξακολουθούν να διαβάζονται από τους οδηγούς των 1.44 MB 3.5".

### 3.7.5. 2.88 MB 3.5" Floppy

Η δισκέτα με τη μεγαλύτερη χωρητικότητα είναι η 2.88 MB στις 3.5" και παρουσιάστηκε από την εταιρία Toshiba στα τέλη της δεκαετίας του '80. Η δισκέτα αυτή προσφέρει τη διπλάσια χωρητικότητα από τη δισκέτα των 1.44 MB στις 3.5", χρησιμοποιώντας ειδικό υλικό επικάλυψης της επιφανείας της δισκέτας, καθώς και ειδική μέθοδο εγγραφής των δεδομένων. Η δισκέτα των 2.88 MB στις 3.5" ποτέ δε γνώρισε εμπορική επιτυχία, παρόλο που έχει μεγαλύτερη χωρητικότητα από 1.44 MB. Τρεις κυρίως λόγοι έχουν επιδράσει σ' αυτό το γεγονός:

- **Αδράνεια:** Την εποχή που παρουσιάστηκαν οι δισκέτες των 2.88 MB, υπήρξε μια εκπληκτική ανάπτυξη της αγοράς υπολογιστών, που διέθεταν οδηγούς δισκετών των 1.44MB. Έτσι, κανείς δεν ήταν πρόθυμος να αλλάξει τον οδηγό των 1.44 MB, ο οποίος δεν μπορούσε να διαβάσει δισκέτες των 2.88 MB.
- **Συμβατότητα:** Η πολύ υψηλή πυκνότητα (36 τομείς ανά ίχνος) των δισκετών των 2.88 MB, απαιτεί έναν ελεγκτή εύκαμπτου δίσκου με προδιαγραφές μεταφοράς δεδομένων γύρω στα 1 Mbits/sec. Οι περισσότεροι παλιοί υπολογιστές διέθεταν ελεγκτή εύκαμπτου δίσκου με προδιαγραφές μεταφοράς δεδομένων, γύρω στα 500 Kbits/sec. Για να χειριστούν το νέο τύπο δισκέτας, έπρεπε να προμηθευτούν με μια νέα κάρτα, η οποία φυσικά αυξάνει το κόστος. Πάντως όλα τα νέα μοντέλα υποστηρίζουν οδηγούς για δισκέτες των 2.88 MB.
- **Κόστος κατασκευής:** Το υλικό επικάλυψης των δισκετών των 2.88 MB είναι πολύ ακριβότερο από εκείνο που χρησιμοποιείται για τις δισκέτες των 1.44 MB. Επειδή η παραγωγή δισκετών των 2.88 MB δεν υπήρξε ποτέ μεγάλη, το κόστος αγορά τους έμεινε μεγάλο σε σχέση με το κόστος αγοράς των δισκετών των 1.44 MB.



## Κεφάλαιο 4 : Συστήματα Αρχείων

### 4.1 Εισαγωγή.

Ένα σύστημα αρχείων ( ΣΑ, file system ή FS ) είναι μια μέθοδος για την αποθήκευση και την οργάνωση των δεδομένων των αρχείων που χρησιμοποιούνται σε ηλεκτρονικό υπολογιστή. Ουσιαστικά, οργανώνει αυτά τα αρχεία σε μια βάση δεδομένων με σκοπό την αποθήκευση, την οργάνωση, τη διαχείριση, και την ανάκτηση τους από το λειτουργικό σύστημα του υπολογιστή.

Τα συστήματα αρχείων χρησιμοποιούνται σε συσκευές αποθήκευσης δεδομένων, όπως οι σκληροί δίσκοι ή τα CD-ROM για να παρακολουθείται η φυσική θέση των αρχείων πάνω στο μέσο. Πέρα από αυτό, τα συστήματα αρχείων μπορούν να παρέχουν πρόσβαση σε δεδομένα σε ένα διακομιστή αρχείων, δρώντας ως πελάτες για ένα πρωτόκολλο δικτύου, ή μπορεί να είναι εικονικά και να υπάρχουν μόνο ως τρόπος πρόσβασης για τα εικονικά δεδομένα. Διαφέρουν από μια υπηρεσία καταλόγου ή μητρώου.

Τα περισσότερα συστήματα αρχείων κάνουν χρήση μιας συσκευής αποθήκευσης δεδομένων η οποία προσφέρει πρόσβαση σε έναν πίνακα φυσικών τομέων ορισμένου μεγέθους, κατά κανόνα μια δύναμη του 2 σε μέγεθος (512 bytes ή 1, 2, ή 4 KB είναι οι πιο συχνές). Το σύστημα αρχείων είναι υπεύθυνο για να οργανώσει τους τομείς αυτούς σε αρχεία και καταλόγους, καθώς και για την παρακολούθηση του ποιοι τομείς ανήκουν σε ποιο αρχείο και ποιοι δεν χρησιμοποιούνται. Τα περισσότερα συστήματα αρχείων αντιστοιχούν τα δεδομένα σε μονάδες σταθερού μεγέθους (“δέσμες” ή “μπλοκ”) που περιέχουν ένα συγκεκριμένο αριθμό τομέων δίσκου (συνήθως 1-64) που είναι και το μικρότερο ποσό χώρου στο δίσκο που μπορεί να διατεθεί για να διατηρήσει ένα αρχείο. Ωστόσο, τα συστήματα αρχείων δεν χρειάζεται να κάνουν χρήση μιας συσκευής αποθήκευσης. Ένα σύστημα αρχείων μπορεί να χρησιμοποιηθεί για να οργανώνει και να εκπροσωπεί την πρόσβαση σε οποιαδήποτε δεδομένα, είτε πρόκειται για αποθήκευση ή δημιουργία.

Το βασικό σύστημα αρχείων που χρησιμοποιείται από την πλειοψηφία των υπολογιστών είναι το FAT. Αυτό το όνομα προκύπτει από τη συντομογραφία του file allocation table (πίνακας κατανομής αρχείων), το οποίο είναι μια από τις βασικές λογικές δομές, που το σύστημα αρχείων χρησιμοποιεί. Συνεπώς δεν είναι το πιο κατάλληλο όνομα για να αναφέρεται σε όλο το σύστημα αρχείων. Το FAT χρησιμοποιείται κυρίως από το DOS και τα Windows.

Το FAT32 είναι μια βελτίωση του τυπικού FAT συστήματος αρχείων. Ονομάστηκε FAT32 γιατί επιτρέπει τη χρήση 32-bit αριθμών για να αναπαραστήσουν τον αριθμό της συστοιχίας, αντί για 16-bit αριθμούς που χρησιμοποιούσε το τυπικό FAT (το οποίο αποκαλούνταν και FAT16 για αυτό το λόγο). Το FAT32 παρουσιάστηκε στα Windows 95 OEM Service Release 2 και μερικές φορές αποκαλείται Windows 95b και υποστηρίζεται από αυτή την έκδοση των Windows 95, καθώς και από την έκδοση του DOS που είναι "ενσωματωμένη". Παλιότερα λειτουργικά συστήματα δεν μπορούν να διαβάσουν ένα δίσκο, ο οποίος είναι διαμορφωμένος με FAT32. Το FAT32 δημιουργήθηκε βασικά για ένα λόγο: οι κατασκευαστές σκληρών δίσκων άρχισαν να κατασκευάζουν σκληρούς δίσκους μεγαλύτερους από 2GB και το FAT16 υποστηρίζει το μέγιστο μέχρι 2GB για κάθε λογικό δίσκο. Έτσι το FAT32 επέκτεινε αυτή τη δυνατότητα στα 8GB και μπορεί να χειριστεί ακόμα μεγαλύτερους δίσκους χρησιμοποιώντας τις ίδιες βασικές δομές. Επιπρόσθετα, για να επιτρέψει μεγαλύτερα ανεξάρτητα λογικά διαμερίσματα ή λογικούς δίσκους, το FAT32 εξοικονομεί σπαταλημένο χώρο εξαιτίας του άχρηστου χώρου, γιατί χρησιμοποιεί συστοιχίες πολύ μικρότερες σε μέγεθος από αυτές του FAT16. Η συνέπεια αυτού είναι ότι ο αριθμός των συστοιχιών που χρησιμοποιούνται είναι πολύ μεγαλύτερος, κάτι που σημαίνει μικρότερη απόδοση (περισσότερη μνήμη για να κρατάει το μεγαλύτερο FAT). Εκτός από τη διαφορά στον τρόπο με τον οποίο οι συστοιχίες είναι προσδιορισμένες και αριθμημένες, το FAT32 είναι στην ουσία το ίδιο με το κανονικό FAT και οι περιγραφές των δομών αρχείων του FAT ταιριάζουν και για το FAT32.

Η Microsoft ενσωμάτωσε διάφορες βελτιώσεις στις δυνατότητες διαχείρισης του δίσκου στα Windows 95. Η πρόσβαση στο σύστημα αρχείων μπορεί να γίνει χρησιμοποιώντας υψηλής ταχύτητας, προστατευμένης χρήσης (protected mode), 32-bit οδηγούς, ή για συμβατότητα τις παλιότερες 16-bit ρουτίνες του DOS. Προστέθηκε επίσης υποστήριξη για μεγαλύτερα ονόματα αρχείων και για καλύτερο έλεγχο σε

θέματα όπως το κλείδωμα του δίσκου, έτσι ώστε βοηθητικά προγράμματα να μπορούν να έχουν πρόσβαση στο δίσκο, σε κατάσταση αποκλειστικής χρήσης ( exclusive mode ), χωρίς το φόβο ότι στο μεταξύ τον χρησιμοποιούν άλλα προγράμματα. Παρά το νέο όνομα και τις νέες ικανότητες το VFAT ως σύστημα αρχείων, είναι βασικά το ίδιο με το FAT. Οι περισσότερες από τις νέες ικανότητες σχετίζονται με το πως το σύστημα αρχείων χρησιμοποιείται και όχι με τις πραγματικές δομές στο δίσκο. Το VFAT διαχειρίζεται τυπικά FAT16 διαμερίσματα, καθώς και διαμερίσματα FAT32, κάτω από Windows 95 OSR2 ή μεταγενέστερο. Η μόνη σημαντική διαφορά, σε σχέση με τις πραγματικές δομές, είναι η πρόσθεση μεγάλων ονομάτων αρχείων. Με την επέκταση των μεγάλων ονομάτων των αρχείων, τα Windows 95 χρησιμοποιούν τις ίδιες λογικές δομές με το DOS ή τα Windows 3.x που χρησιμοποιούσαν FAT. Το NT σύστημα αρχείων που χρησιμοποιείται από τα Windows NT είναι τελείως διαφορετικό και ασύμβατο με το σύστημα αρχείων FAT, που χρησιμοποιείται από το DOS και τις άλλες εκδόσεις των Windows. Το NTFS μπορεί να χρησιμοποιηθεί μόνο από τα Windows NT, καθώς τα άλλα λειτουργικά συστήματα δεν έχουν τη δυνατότητα να χρησιμοποιήσουν τη μορφοποίηση του δίσκου με NTFS. Το NTFS είναι ουσιαστικά πολύ καλύτερο από το FAT. Πρόκειται για ένα ισχυρό σύστημα, το οποίο περιλαμβάνει συμπίεση αρχείο προς αρχείο, πλήρη έλεγχο πρόσβασης, απόδοση χαρακτηριστικών, καθώς και άλλα χαρακτηριστικά. Επίσης, δεν έχει πρόβλημα με το μέγεθος των συστοιχιών και περιορισμούς στο μέγεθος του σκληρού δίσκου, όπως έχει το FAT, επίσης διαθέτει και άλλα χαρακτηριστικά που βελτιώνουν την απόδοση, όπως είναι η υποστήριξη RAID (Redundant Arrays of Inexpensive Disks). Το μόνο σημείο στο οποίο το FAT υπερέχει του NTFS είναι στη συμβατότητα με παλαιότερο λογισμικό.

## 4.2 Λειτουργικά συστήματα και συστήματα αρχείων

Διαφορετικά λειτουργικά συστήματα χρησιμοποιούν διαφορετικά συστήματα αρχείων. Κάποια είναι σχεδιασμένα να λειτουργούν με περισσότερα από ένα, για λόγους συμβατότητας, ενώ άλλα λειτουργούν μόνο με το δικό τους σύστημα αρχείων. Παρακάτω θα ρίξουμε μια σύντομη ματιά στα πιο συνηθισμένα λειτουργικά συστήματα και τα συστήματα αρχείων που αυτά χρησιμοποιούν.

#### 4.2.1 DOS (MS-DOS, PC-DOS κτλ.)

Το κλασικό DOS, έκδοση 6.22 και προηγούμενες, χρησιμοποιεί το σύστημα αρχείων FAT. Αυτό περιλαμβάνει τις δύο παραλλαγές FAT12 και FAT16. Τα Windows 95 είχαν μια “ενσωματωμένη” έκδοση του DOS την 7.x. Η OEM Service Release 2 έκδοση για Windows 95, υποστηρίζει FAT32 και έτσι το DOS 7.1, που έρχεται μαζί με τα Windows 95 OSR2, επίσης υποστηρίζει FAT32. Τα FAT32 διαμερίσματα (partitions) του δίσκου δεν είναι συμβατά με τις προηγούμενες από τις 7.x εκδόσεις του DOS.

#### 4.2.2 Windows 3.x

Τα Windows 3.x δεν είναι ένα πραγματικά ανεξάρτητο λειτουργικό σύστημα πολυεργασίας. Έτρεχε “πάνω” από το DOS και τις περισσότερες φορές χρησιμοποιούσε τις υπηρεσίες και τις ρουτίνες του DOS για την πρόσβαση στο δίσκο. Συνεπώς, χρησιμοποιεί το ίδιο FAT σύστημα αρχείων όπως και το DOS. Η τελευταία έκδοση των Windows 3.x, Windows for Workgroups 3.11, περιλάμβανε μια βελτίωση, που ονομαζόταν “32-Bit File Access”. Αυτό ήταν ένα χαρακτηριστικό που αναφερόταν στη χρήση 32-bit προστατευμένης κατάστασης λειτουργίας ρουτινών για πρόσβαση στο δίσκο, αντί για τη χρήση των τυπικών 16-bit ρουτινών του DOS. Στην πραγματικότητα, αυτό ήταν η πρώτη υλοποίηση του VFAT συστήματος αρχείων που χρησιμοποιήθηκε από τα Windows 95, μολονότι δεν περιλαμβάνονταν όλα τα χαρακτηριστικά του VFAT, παρά μόνο η χρήση 32-bit ρουτινών πρόσβασης. Το μόνο διαφορετικό εδώ ήταν το πως γινόταν η πρόσβαση στο δίσκο. Η δομή του συστήματος αρχείων ήταν καθαρά FAT.

#### 4.2.3 Windows 95

Τα Windows 95 ήταν το μεγάλο "συμβιβαστικό" λειτουργικό σύστημα. Σύμφωνα με κάποιες απόψεις, είχε το δικό του τρόπο χειρισμού της πρόσβασης στο σκληρό δίσκο, αλλά σύμφωνα με άλλες απόψεις μοιάζει και συχνά χρησιμοποιεί το τυπικό DOS. Πρόκειται για την αντιπαράθεση που έχουν τα Windows 95, μεταξύ απόδοσης και συμβατότητας με παλιότερο λογισμικό. Τα Windows 95 περιλαμβάνουν μια έκδοση του DOS, DOS 7.x, το οποίο είναι σχεδιασμένο να συνεργάζεται μ' αυτά και τις δομές των αρχείων τους. Το "επίσημο" σύστημα αρχείων που χρησιμοποιούν τα Windows 95 είναι το VFAT, το οποίο υποστηρίζεται και από το DOS 7.x. Αρχίζοντας με Windows 95 OEM Service Release (OSR2), υποστηρίζεται επίσης FAT32, το οποίο επιτρέπει τη χρήση μεγαλύτερων διαμερισμάτων σκληρού δίσκου.

#### 4.2.4 Windows NT

Τα Windows NT, είναι μια νέα υλοποίηση στα Windows. Σε αντίθεση με τις άλλες εκδόσεις των Windows τα Windows NT δε βασίζονταν στο DOS. Τα Windows NT υποστηρίζουν δύο διαφορετικά συστήματα αρχείων. Το πρώτο είναι το NTFS ( NT FILE SYSTEM ). Πρόκειται για ένα σύστημα αρχείων σαν αυτό του UNIX. Το δεύτερο ήταν το κανονικό FAT, όπως αυτό που χρησιμοποιούσε το DOS. Η υποστήριξη του FAT γινόταν για λόγους συμβατότητας ή για την περίπτωση που κάποιοι υπολογιστές χρησιμοποιούν και τα Windows NT και ένα άλλο λειτουργικό σύστημα. Αν και τα Windows NT μπορούν να διαβάσουν και FAT και NTFS διαμερίσματα, αυτά δεν είναι συμβατά μεταξύ τους.

#### 4.2.5 UNIX/Linux

Υπάρχουν πολλές παραλλαγές του UNIX διαθέσιμες για προσωπικούς υπολογιστές. Η πιο συνηθισμένη και δημοφιλής είναι αυτή του Linux. Χρησιμοποιεί συστήματα αρχείων του UNIX, τα οποία είναι τελείως διαφορετικά από τις παραλλαγές του FAT, που χρησιμοποιούνται από το DOS και τα Windows.

### 4.3 Δομές διαμερισμάτων δίσκου στο σύστημα FAT

Για να γίνει κατανοητό τι ακριβώς κάνει αυτός ο πίνακας, πρέπει πρώτα να γίνει κατανοητό πως είναι κατανεμημένος ο χώρος του σκληρού δίσκου στο DOS (και στα παράγωγα του, που χρησιμοποιούν επίσης FAT). Καθώς τα δεδομένα είναι αποθηκευμένα σε τομείς των 512 bytes στο σκληρό δίσκο, για λόγους απόδοσης, ένα αρχείο δεν αποτελείται από ξεχωριστούς τομείς. Ο λόγος είναι ότι θα προκαλούνταν πολλές σπατάλες (χρόνου και χώρου) προκειμένου να αποθηκεύεται πληροφορία για τα κομμάτια των αρχείων, όταν αυτά είναι τόσο μικρά (όπως είναι γνωστό, ένα αρχείο δεν αποθηκεύεται σε συνεχόμενες θέσεις στο δίσκο, αλλά είναι διεσπαρμένο σ' αυτόν). Έτσι, ο σκληρός δίσκος είναι χωρισμένος σε μεγαλύτερα κομμάτια που ονομάζονται συστοιχίες ( clusters ) ή εναλλακτικά μονάδες κατανομής ( allocation units ). Κάθε συστοιχία περιέχει έναν αριθμό από τομείς. Τυπικά, οι συστοιχίες ποικίλλουν σε μέγεθος από 2048 bytes μέχρι 32768 bytes, το οποίο αντιστοιχεί σε 4 μέχρι 64 τομείς η κάθε συστοιχία. Ο πίνακας κατανομής αρχείων είναι το μέρος, όπου είναι αποθηκευμένες πληροφορίες σχετικές με τις συστοιχίες. Κάθε συστοιχία έχει μια καταχώρηση μέσα στο FAT, που περιγράφει πως χρησιμοποιείται. Αυτό λέει στο λειτουργικό σύστημα ποια κομμάτια του δίσκου είναι ελεύθερα και ποιά χρησιμοποιούνται από αρχεία. Οι πίνακες κατανομής αρχείων είναι τοποθετημένοι στην περιοχή του δίσκου αμέσως μετά τον τομέα εκκίνησης του διαμερίσματος. Κάθε διαμέρισμα πραγματικά περιέχει δύο ίδια αντίγραφα του FAT. Προφανώς, το δεύτερο χρησιμοποιείται ως εφεδρικό, σε περίπτωση που συμβεί κάποια ζημιά στο πρώτο. Μια ζημιά στο FAT μπορεί να προκαλέσει απώλεια δεδομένων, καθώς εκεί είναι αποθηκευμένη η εγγραφή για το ποια κομμάτια του δίσκου περιέχουν ποια αρχεία. Το πρόβλημα με αυτό το ενσωματωμένο αντίγραφο είναι ότι κρατείται ακριβώς δίπλα στο πρώτο, ώστε αν προκληθεί κάποια ζημιά, από κατεστραμμένους τομείς για παράδειγμα στο πρώτο FAT, είναι πολύ πιθανό το πρόβλημα να έχει επίσης επεκταθεί και στο εφεδρικό.

#### 4.3.1 Εσωτερικές δομές καταλόγου

Κάθε αρχείο του υπολογιστή είναι αποθηκευμένο σε ένα κατάλογο. Ο κατάλογος δεν είναι τίποτα παραπάνω από ένα αρχείο, το οποίο όμως έχει ειδική δομή και είναι ορισμένο στο δίσκο, έτσι ώστε να έχει ειδική σημασία. Ο κατάλογος είναι ένας πίνακας που περιέχει πληροφορίες σχετικές με τα αρχεία και τους υποκαταλόγους που υπάρχουν σ' αυτόν, καθώς και πληροφορίες για το που ξεκινούν τα αρχεία ή οι υποκατάλογοι στο δίσκο. Κάθε καταχώρηση σε ένα κατάλογο καταλαμβάνει 32 bytes σε μήκος και αποθηκεύει τις παρακάτω πληροφορίες:

- **Όνομα αρχείου και επέκταση:** Πρόκειται για ένα όνομα 11 χαρακτήρων του αρχείου χρησιμοποιώντας τη συμβατική τυποποίηση των ονομάτων αρχείων του DOS (8 χαρακτήρες για όνομα, 3 για την επέκταση του αρχείου και το διαχωρισμό αυτών των δύο με την τελεία "."), για παράδειγμα MYTEXT.TXT. Να σημειωθεί εδώ ότι η τελεία μεταξύ του ονόματος του αρχείου και της επέκτασης είναι για δική μας χρήση και δεν αποθηκεύεται πραγματικά στο δίσκο.
- **File Attribute Byte:** Υπάρχουν διάφορα χαρακτηριστικά, τα οποία το λειτουργικό σύστημα χρησιμοποιεί για να δώσει ειδική χρήση σε συγκεκριμένα αρχεία. Αυτά είναι αποθηκευμένα σε ένα μόνο byte σε κάθε καταχώρηση του καταλόγου. Να σημειωθεί εδώ ότι ένα τέτοιο χαρακτηριστικό φανερώνει αν μια καταχώρηση αναπαριστά ένα "πραγματικό" αρχείο ή έναν υποκατάλογο.
- **Τελευταία Μέρα/Ωρα Αλλαγής:** Υπάρχει ένας χώρος για κάθε αρχείο που δείχνει τη μέρα και την ώρα που το αρχείο δημιουργήθηκε ή τροποποιήθηκε. Ωστόσο, αυτά τα πεδία μπορούν να τροποποιηθούν αυθαίρετα από κάποιο πρόγραμμα και να μετατραπούν σε οτιδήποτε αυτό θέλει. Συνεπώς, δεν πρέπει να λαμβάνονται και πολύ σοβαρά υπόψη.
- **Το μέγεθος του αρχείου:** Το μέγεθος του αρχείου σε bytes.
- **Σύνδεσμος στην αρχική συστοιχία:** Ο αριθμός της συστοιχίας που αρχίζει το αρχείο ή ο υποκατάλογος είναι αποθηκευμένα στον κατάλογο. Αυτό επιτρέπει στο λειτουργικό σύστημα να βρει ένα αρχείο, όταν το χρειάζεται και πως όλα τα διαφορετικά αρχεία και κατάλογοι είναι συνδεδεμένα στο δίσκο.

### 4.3.2 Ο κατάλογος της ρίζας και κανονικοί κατάλογοι

Ο κατάλογος στη “βάση” της δομής των καταλόγων, η οποία ορίζει το λογικό δέντρο που οργανώνει τα αρχεία στο σκληρό δίσκο είναι ο κατάλογος της ρίζας (root). Πρόκειται για ένα ειδικό κατάλογο καθώς ακολουθεί ειδικούς κανόνες που δεν ισχύουν στους άλλους, τους “κανονικούς” καταλόγους στο σκληρό δίσκο. Σε κάθε διαμερίσμα του δίσκου, μπορεί να υπάρχει μόνο ένας κατάλογος ρίζας. Το να είχε περισσότερους θα οδηγούσε σε χάος και άλλωστε δεν υπάρχει καμιά ανάγκη για ύπαρξη περισσότερων του ενός. Είναι σταθερά τοποθετημένος στην αρχή του διαμερίσματος, κατευθείαν μετά τα δύο αντίγραφα του FAT. Σε αυτό το σημείο είναι και η διαφορά του από τους άλλους (υπό)καταλόγους που βρίσκονται οπουδήποτε μέσα στο δίσκο. Πέραν του ότι βρίσκεται σε συγκεκριμένη περιοχή μέσα στο δίσκο, ο κατάλογος της ρίζας έχει επίσης και περιορισμό ως προς το μέγεθος του. Οι κανονικοί κατάλογοι έχουν αυθαίρετο μέγεθος και χρησιμοποιούν το χώρο στο δίσκο όπως και τα αρχεία. Όταν χρειάζονται περισσότερο χώρο για να κρατήσουν περισσότερες καταχωρήσεις, οι κατάλογοι μπορούν να επεκταθούν με τον ίδιο τρόπο που το κάνουν και τα αρχεία. Ο κατάλογος της ρίζας είναι περιορισμένος σε ένα συγκεκριμένο αριθμό καταχωρήσεων εξαιτίας της ειδικής του θέσης. Ο αριθμός των καταχωρήσεων που ένας κατάλογος ρίζας μπορεί να κρατήσει εξαρτάται από τον τύπο του διαμερίσματος.

**Πίνακας 3.** Αντιστοιχία αριθμού καταχωρήσεων προς τύπο διαμερίσματος.

Τύπος διαμερίσματος	Μέγιστος αριθμός καταχωρήσεων
360 KB 5.25"	112



1.2 MB 5.25"	224
720 KB 3.5"	112
1.44 MB 3.5"	244
2.88 MB 3.5"	512

Να σημειωθεί ότι, η νεότερη FAT32 έκδοση του συστήματος αρχείων FAT δεν έχει περιορισμούς στην τοποθέτηση και το μέγεθος του καταλόγου της ρίζας. Με αυτή τη βελτίωση μπορούμε να μεταχειριστούμε τον κατάλογο της ρίζας όπως τους κανονικούς καταλόγους και έτσι αυτός μπορεί να μεταφερθεί και να επεκταθεί σε μέγεθος όπως κάθε κατάλογος. Υπάρχουν ακόμα μερικά ειδικά πράγματα σχετικά με τον κατάλογο της ρίζας. Ένα είναι ότι δεν μπορεί να διαγραφεί και ο λόγος γι' αυτό είναι φανερός. Επίσης ο κατάλογος αυτός δεν έχει γονέα, καθώς βρίσκεται στην κορυφή της δομής του δέντρου.

### 4.3.3 Ονόματα αρχείων και επεκτάσεις

Το όνομα ενός αρχείου είναι χωρισμένο σε δύο μέρη:

- **Όνομα αρχείου:** Αυτό το μέρος του ονόματος του αρχείου πρέπει να είναι μεταξύ ενός και οκτώ χαρακτήρων σε μήκος (για το DOS και τα Windows 3.x). Ένας ειδικός κώδικας χρησιμοποιείται σαν πρώτος χαρακτήρας του ονόματος για να προσδιορίσει διαγραμμένα αρχεία.
- **Επέκταση αρχείου:** Η επέκταση του αρχείου, η οποία είναι προαιρετική και έτσι μπορεί να είναι από κανένα μέχρι τρεις χαρακτήρες.

Η επέκταση μπορεί να θεωρηθεί ως καθορισμός του τύπου του αρχείου. Προσδιορίζει, με μια πρώτη ματιά, στο χρήστη και στο σύστημα το είδος του κάθε αρχείου. Για παράδειγμα, ένα αρχείο με κατάληξη "EXE" είναι κανονικά ένα εκτελέσιμο αρχείο, το "TXT" σημαίνει αρχείο κειμένου και το "DAT" αρχείο

δεδομένων. Στην πραγματικότητα, δεν υπάρχει κάτι το ιδιαίτερο με τα ονόματα των επεκτάσεων. Το να υποδηλώνει η επέκταση “EXE” κάποιο εκτελέσιμο αρχείο είναι μια συνήθεια, δεν είναι κάτι που επιβάλλεται από το σύστημα. Μπορούμε για παράδειγμα, να ανοίξουμε τον κειμενογράφο, να δημιουργήσουμε ένα αρχείο κειμένου και να το σώσουμε με επέκταση “EXE”. Δε θα υπάρξει κανένα πρόβλημα. Τότε λοιπόν γιατί χρησιμοποιούνται οι επεκτάσεις; Ουσιαστικά είναι ένας σύντομος τρόπος οργάνωσης των αρχείων από τον τύπο τους και χρησιμοποιούνται από διάφορα κομμάτια του λογισμικού, συμπεριλαμβανομένου και του DOS, για να προσδιορίσουν ποιά προγράμματα πρέπει να χρησιμοποιηθούν για να ανοίξουν κάποια συγκεκριμένα αρχεία, χωρίς να κοιτάζουν στη δομή των αρχείων. Ο λόγος που κάποια αρχεία έχουν την επέκταση “EXE” είναι ότι, το DOS είναι προγραμματισμένο κατά τέτοιο τρόπο ώστε όταν πληκτρολογούμε το όνομα ενός αρχείου, αν το αρχείο έχει κάποια συγκεκριμένη επέκταση, το DOS προσπαθεί να τα εκτελέσει. Μια από τις επεκτάσεις αυτές είναι η “EXE”. Έτσι αν δημιουργήσουμε το αρχείο κειμένου με επέκταση “EXE” και πληκτρολογήσουμε το όνομα του, το DOS θα προσπαθήσει να τρέξει αυτό το πρόγραμμα. Το σίγουρο είναι ότι δε θα τα καταφέρει και καλό είναι αν προσπαθούμε να κάνουμε κάτι τέτοιο, να μην έχουμε αφήσει κάποιες εργασίες χωρίς να τις έχουμε σώσει.

Παρόμοια, άλλα προγράμματα συνήθως προσπαθούν να δουν μόνο αρχεία που έχουν την επέκταση που αυτά χρησιμοποιούν. Αν τρέξουμε το Word και επιλέξουμε να ανοίξουμε ένα αρχείο, το Word θα ψάξει για “DOC” αρχεία, από το “document” (έγγραφο). Είναι λοιπόν σημαντικό να χρησιμοποιούνται συνεπείς επεκτάσεις αρχείων. Αυτή η χρήση των επεκτάσεων από προγράμματα είναι αρκετά διαδεδομένη και υπάρχουν εκατοντάδες διαφορετικές επεκτάσεις που χρησιμοποιούνται. Αν για παράδειγμα, πατήσουμε δύο φορές το πλήκτρο του ποντικιού πάνω σε ένα αρχείο στο Windows Explorer των Windows, αυτός αυτόματα θα εκκινήσει το πρόγραμμα που ξέρει ότι χρησιμοποιεί το αρχείο που επιλέξαμε και θα πει στο πρόγραμμα να ανοίξει το αρχείο αυτό. Ο Explorer αποφασίζει ποιό πρόγραμμα πρέπει να ανοιχτεί, απλά κοιτάζοντας την επέκταση του αρχείου που επιλέχτηκε και όχι αναλύοντας οτιδήποτε μέσα στο αρχείο.

#### 4.3.4 Μεγάλα ονόματα αρχείων

Μέχρι την έκδοση των Windows 95, όλα τα ονόματα αρχείων που χρησιμοποιούσε το DOS ή τα Windows 3.x, ήταν περιορισμένα σε ένα όνομα μέχρι οκτώ χαρακτήρες και επιπλέον τρεις χαρακτήρες για την επέκταση. Αυτός ο περιορισμός οδήγησε τους χρήστες στο να δημιουργούν πάρα πολύ μυστικιστικά ονόματα και αυτή η κατάσταση που επικρατούσε 15 χρόνια μετά τη δημιουργία των πρώτων προσωπικών υπολογιστών έμοιαζε περίεργη, ειδικά όταν η Microsoft ήθελε να συγκρίνει τις ευκολίες που παρείχε σε σχέση με αυτές ενός Macintosh. Οι χρήστες ήθελαν να ονομάσουν τα αρχεία τους χρησιμοποιώντας μεγάλα ονόματα, γιατί θα τους βοηθούσαν να θυμούνται τι ακριβώς περιέχει το αρχείο μερικούς μήνες μετά από τη δημιουργία του.

Έτσι, η Microsoft αποφάσισε να επιτρέψει τη δημιουργία μεγάλων ονομάτων αρχείων, long file name (LFN), στα Windows 95, όπως είχε κάνει και στα Windows NT. Τα τελευταία, ωστόσο, είχαν ένα σύστημα αρχείων σχεδιασμένο τελείως διαφορετικά για να επιτρέπουν μεγάλα ονόματα αρχείων. Η Microsoft είχε μεγάλο πρόβλημα στο χειρισμό των Windows 95, ήθελε να διατηρήσει τη συμβατότητα με τις υπάρχουσες δομές των δίσκων, το παλαιότερο DOS και τα Windows 3.x, και τις παλιότερες εφαρμογές. Δεν μπορούσε απλά να απορρίψει οτιδήποτε παλαιότερο και να δημιουργήσει κάτι τελείως καινούριο, καθώς αν γινόταν αυτό, τα παλαιότερα προγράμματα δε θα μπορούσαν να διαβάσουν τα αρχεία που είχαν μεγάλα ονόματα.

Αυτό που ήθελε η Microsoft ήταν ένας τρόπος να υλοποιήσει μεγάλα ονόματα αρχείων έτσι ώστε οι παρακάτω στόχοι να επιτευχθούν:

- Τα Windows 95 καθώς και οι εφαρμογές που είναι γραμμένες σε αυτά, να μπορούν να χρησιμοποιήσουν ονόματα αρχείων πολύ μεγαλύτερα από συνολικά 11 χαρακτήρες.
- Τα νέα μεγάλα ονόματα να μπορούν να αποθηκευτούν στα υπάρχοντα DOS διαμερίσματα, χρησιμοποιώντας τις τυπικές δομές του καταλόγου, για συμβατότητα.

- Το παλαιότερο, πριν τα Windows 95, λογισμικό να μπορεί επίσης να έχει πρόσβαση, με κάποιο τρόπο, στα αρχεία που χρησιμοποιούσαν τα καινούρια ονόματα.

Το σύστημα αρχείων VFAT εκπληρώνει αυτούς τους στόχους, στα περισσότερα σημεία τους. Μεγάλα ονόματα αρχείων μέχρι 255 χαρακτήρες ανά αρχείο, μπορεί να αντιστοιχίσουν σε κάποιο αρχείο των Windows 95 ή κάποιου προγράμματος γραμμένο για αυτά. Υποστήριξη για αυτά τα μεγάλα ονόματα αρχείων παρέχεται και από την έκδοση του DOS (7.x), που έρχεται μαζί με τα Windows 95. Οι επεκτάσεις των αρχείων εξακολουθούν να υφίστανται, για να καθορίζουν τον τρόπο που τα αρχεία χρησιμοποιούνται από το λογισμικό. Τα μεγάλα ονόματα αρχείων είναι αποθηκευμένα σε κανονικούς καταλόγους χρησιμοποιώντας τις τυπικές καταχωρήσεις, αλλά χρησιμοποιούν ένα αριθμό κόλπων. Το σύστημα αρχείων των Windows 95 δημιουργεί μια τυπική καταχώρηση για το αρχείο, στην οποία τοποθετεί το “ψευδώνυμο” του αρχείου. Στη συνέχεια χρησιμοποιεί επιπλέον καταχωρήσεις για να κρατήσει το υπόλοιπο του ονόματος. Ένα απλό μεγάλο όνομα ενός αρχείου μπορεί να χρησιμοποιήσει αρκετές καταχωρήσεις (καθώς κάθε καταχώρηση έχει μόνο 32 bytes μήκος) και γι' αυτό δεν είναι ενδεδειγμένο τα μεγάλα αυτά ονόματα, να αποθηκεύονται στον κατάλογο της ρίζας, που ο αριθμός των καταχωρήσεων είναι περιορισμένος.

Επιπλέον, για να βεβαιώσουν ότι οι παλιότερες εκδόσεις του DOS δε θα μπερδευτούν από αυτή τη μη τυποποιημένη χρήση, κάθε μια επιπλέον καταχώρηση του καταλόγου που χρησιμοποιείται για να κρατάει πληροφορίες για το μεγάλο όνομα του αρχείου, έχει σαν ετικέτα τον ακόλουθο ασυνήθιστο συνδυασμό χαρακτηριστικών αρχείου: read-only (αρχείο ανάγνωσης μόνο), hidden (κρυφό αρχείο), system και volume label (ετικέτα συστήματος και διαμερίσματος). Ο σκοπός αυτού είναι να βεβαιωθούμε ότι καμία παλαιότερη έκδοση του DOS δε θα επιχειρήσει να κάνει οτιδήποτε με αυτές τις καταχωρήσεις, καθώς και να γράψει κάτι άλλο σε αυτές τις καταχωρήσεις νομίζοντας ότι δε χρησιμοποιούνται. Αυτός ο συνδυασμός χαρακτηριστικών είναι η αιτία που το παλαιότερο λογισμικό αγνοεί τις επιπλέον καταχωρήσεις καταλόγου, που χρησιμοποιούνται από το VFAT. Καθώς τα μεγάλα ονόματα αρχείων ήταν μια πολύ σπουδαία ιδέα και βελτίωνε τη λειτουργικότητα των Windows 95, η Microsoft έκανε μεγάλη προσπάθεια για να κρατήσει τα νέα ονόματα συμβατά με το παλαιότερο

λογισμικό. Ωστόσο, υπάρχουν αρκετά προβλήματα στη χρησιμοποίηση μεγάλων ονομάτων αρχείων:

- **Προβλήματα συμβατότητας με παλαιότερες εφαρμογές:** Καθώς σημειώνει τις επιπλέον καταχωρήσεις ως read-only, hidden, system και volume label, για να “ξεγελάσει” τις τυπικές εφαρμογές ώστε να μην τις πειράξουν, ένα βοηθητικό πρόγραμμα του δίσκου δε θα ξεγελαστεί. Αν χρησιμοποιήσουμε μια έκδοση του DOS που δεν είναι ενήμερη για τα μεγάλα ονόματα αρχείων, θα ανιχνεύσει αυτές τις καταχωρήσεις σαν λάθη στο δίσκο και ευχαρίστως θα τις “διόρθωνε”. Οι εφαρμογές που τρέχουν κάτω από Windows 95, πρέπει να είναι ενήμερες για τα μεγάλα ονόματα αρχείων για να δουλεύουν κανονικά.
- **“Απώλεια” των μεγάλων ονομάτων αρχείων με το παλαιότερο λογισμικό:** Καθώς παλαιότερες εφαρμογές θα δουλέψουν με ένα αρχείο με μεγάλο όνομα χρησιμοποιώντας το “ψευδώνυμο” του, καθώς δεν έχουν την ικανότητα να έχουν πρόσβαση σε ολόκληρο το όνομα. Είναι εύκολο κάποιο από αυτά τα προγράμματα να “κόψει” το όνομα του αρχείου από ατύχημα. Ένα συνηθισμένο τέτοιο πρόβλημα μπορεί να προκύψει αν χρησιμοποιούμε παλαιότερο λογισμικό του DOS για εφεδρική αποθήκευση αρχείων (backup), που δεν είναι ενήμερο για τα μεγάλα ονόματα αρχείων, θα σώσει μόνο τα “ψευδώνυμα” και θα προκληθεί κάποιο πρόβλημα (crash), οπότε χρειάζεται να το αποκαταστήσουμε ενώ προηγουμένως τα μεγάλα ονόματα αρχείων θα έχουν χαθεί.
- **Προβλήματα με επαναπροσδιορισμό ψευδωνύμου μικρών ονομάτων αρχείων:** Αντιγράφοντας ένα αρχείο με μεγάλο όνομα από ένα διαμέρισμα σε ένα άλλο ή επαναφέροντας το από ένα εφεδρικό, μπορεί να προκαλέσει την αλλαγή του “ψευδωνύμου” που σχετίζεται με το αρχείο αυτό.

Γενικά, τα μεγάλα ονόματα αρχείων είναι μια χρήσιμη βελτίωση στη λειτουργικότητα του FAT συστήματος αρχείων, αλλά χρειάζεται να ξέρουμε τα προβλήματα που μπορεί να προκύψουν όταν τα χρησιμοποιούμε, ειδικά με παλαιότερο λογισμικό.

#### 4.3.5 Χαρακτηριστικά των αρχείων

Κάθε αρχείο είναι αποθηκευμένο σε ένα κατάλογο και χρησιμοποιεί μια καταχώρηση καταλόγου η οποία περιγράφει τα χαρακτηριστικά του, όπως είναι το όνομα του, το μέγεθος του καθώς και ένα δείκτη για το σημείο που είναι αποθηκευμένο το αρχείο στο δίσκο. Ένα από τα στοιχεία που αποθηκεύονται είναι ένα σύνολο από χαρακτηριστικά αρχείου (attributes files) που δίνει στο DOS και σε άλλες πληροφορίες σχετικά με το αρχείο, καθώς και πως προτείνεται να αντιμετωπιστεί. Η χρήση των χαρακτηριστικών είναι εθελοντική. Αυτό σημαίνει ότι κάθε λογισμικό μπορεί να κοιτάξει μέσα στην καταχώρηση του καταλόγου για να διακρίνει τα χαρακτηριστικά του αρχείου και με βάση αυτά, να πάρει “έξυπνες” αποφάσεις σχετικά με το πως θα χειριστεί το αρχείο. Για παράδειγμα, ένα πρόγραμμα διαχείρισης αρχείων, βλέποντας ότι ένα αρχείο είναι σημειωμένο ως read-only, θα ήταν συνετό να ενημερώσει το χρήστη πριν το διαγράψει. Ωστόσο, δεν είναι υποχρεωμένο να κάνει κάτι τέτοιο. Κάποιο πρόγραμμα που ξέρει τι κάνει μπορεί να παραβλέψει τα χαρακτηριστικά του αρχείου. Όπως ειπώθηκε, το DOS και τα περισσότερα άλλα λειτουργικά συστήματα, αντιστοιχούν καθορισμένες έννοιες στα χαρακτηριστικά των αρχείων και διαμορφώνουν τη συμπεριφορά τους ανάλογα με αυτά. Αν πληκτρολογήσουμε “DIR” στο DOS, τότε δε θα δούμε τα αρχεία που έχουν το χαρακτηριστικό του κρυφού (hidden). Για να εμφανιστούν και αυτά τα αρχεία πρέπει να πληκτρολογήσουμε “DIR /AH”. Ένα αρχείο μπορεί να έχει περισσότερα από ένα χαρακτηριστικά προσαρτημένα σε αυτό, παρόλα αυτά μόνο ορισμένοι συνδυασμοί έχουν λογική. Τα χαρακτηριστικά είναι αποθηκευμένα σε ένα απλό byte, με κάθε bit να αναπαριστά ένα συγκεκριμένο χαρακτηριστικό (πραγματικά, μόνο έξι από τα οκτώ bits του byte χρησιμοποιούνται). Κάθε bit που είναι μονάδα σημαίνει ότι το αρχείο έχει αυτό το χαρακτηριστικό ενεργοποιημένο. Αυτή η μέθοδος είναι ένας συνηθισμένος τρόπος που οι “ναι / όχι” παράμετροι αποθηκεύονται στον υπολογιστή για την εξοικονόμηση χώρου. Τα παρακάτω είναι τα χαρακτηριστικά, καθώς και το bit που χρησιμοποιούν στο byte των χαρακτηριστικών: Συνεπώς το byte των χαρακτηριστικών για ένα hidden , read-only directory θα είναι 00010011, το οποίο είναι απλά οι κώδικες για τα τρία αυτά χαρακτηριστικά, τοποθετημένα μαζί. Ακολουθεί μια πιο λεπτομερής περιγραφή του τι κάθε χαρακτηριστικό σημαίνει και επιπλέον πως συνήθως χρησιμοποιούνται. Τα όσα αναγράφονται παρακάτω αφορούν και τα αρχεία και τους καταλόγους, εκτός φυσικά από το χαρακτηριστικό του καταλόγου:

- **Read-only (Ανάγνωση μόνο):** Τα περισσότερα προγράμματα, όταν βλέπουν ένα αρχείο σημειωμένο ως read-only, αρνούνται να το σβήσουν ή να το τροποποιήσουν. Για παράδειγμα, το DOS θα πει “Access denied” αν προσπαθήσουμε να διαγράψουμε ένα read-only αρχείο. Από την άλλη μεριά ο Windows Explorer θα διέγραφε πολύ ευχαρίστως. Κάποια άλλα προγράμματα θα ακολουθούσαν μια μέση λύση: θα τροποποιήσαν ή θα διέγραφαν το αρχείο, αφού όμως πρώτα ζητούσαν επιβεβαίωση.
- **Hidden (Κρυφό αρχείο):** Αυτό το χαρακτηριστικό δεν είναι δύσκολο να κατανοηθεί τι ακριβώς είναι. Αν το αρχείο είναι σημειωμένο ως κρυφό, τότε υπό κανονικές συνθήκες δεν μπορούμε να το δούμε. Το DOS για παράδειγμα, δε θα το εμφανίσει αν πληκτρολογήσουμε την εντολή “DIR”.
- **System (αρχεία συστήματος):** Αυτό το χαρακτηριστικό προσδιορίζει κάποια συγκεκριμένα αρχεία που χρησιμοποιούνται από το σύστημα και δεν πρέπει να μεταβληθούν ή να μετακινηθούν από το δίσκο. Στην ουσία αυτό το χαρακτηριστικό είναι πιο σημαντικό από το read-only.
- **Volume label (Ετικέτα λογικού δίσκου):** Κάθε λογικός δίσκος μπορεί να αντιστοιχηθεί με ένα αναγνωριστικό όνομα, είτε την ώρα που μορφοποιείται, είτε αργότερα από κάποιο πρόγραμμα, όπως είναι η εντολή “LABEL” του “DOS”. Η ετικέτα διαμερίσματος είναι αποθηκευμένη στον κατάλογο της ρίζας σαν μια καταχώρηση αρχείου.
- **Directory:** Αυτό είναι το bit, το οποίο ξεχωρίζει τις καταχωρήσεις που περιέχουν αρχεία από αυτές που περιέχουν υποκαταλόγους μέσα στον παρόντα κατάλογο. Θεωρητικά, ένα αρχείο μπορεί να μετατραπεί σε κατάλογο αλλάζοντας αυτό το bit, φυσικά όμως στην πράξη η προσπάθεια μεταβολής αυτού του bit θα προκαλέσει μπέρδεμα, καθώς η καταχώρηση για ένα κατάλογο έχει ειδική διαμόρφωση.
- **Archive:** Αυτό είναι ένα ειδικό bit, που χρησιμοποιείται ως ένας “σύνδεσμος επικοινωνίας” μεταξύ των εφαρμογών λογισμικού που μεταβάλλουν αρχεία και αυτά χρησιμοποιούνται για εφεδρική αποθήκευση. Τα περισσότερα προγράμματα που χρησιμοποιούνται για εφεδρική αποθήκευση επιτρέπουν στο χρήστη να δημιουργήσει ένα "προσαναζητικό" αντίγραφο, το οποίο θα επιλέγει για εφεδρική αποθήκευση μόνο τα αρχεία που άλλαξαν από τη τελευταία εφεδρική αποθήκευση. Αυτό το bit χρησιμοποιείται για αυτό το σκοπό. Όταν το πρόγραμμα κάνει εφεδρική αποθήκευση, τότε μετατρέπει αυτό το bit σε μηδέν. Αν κάποιο πρόγραμμα μεταβάλλει το αρχείο, τότε θέτει και το bit αυτό ένα. Έτσι την επόμενη φορά που το πρόγραμμα της

εφεδρικής αποθήκευσης θα τρέξει, θα ξέρει κοιτώντας αυτό το bit αν το αρχείο έχει αλλάξει και αν συνεπώς χρειάζεται να το δημιουργήσει κάποιο αντίγραφο ασφαλείας. Η μεταβολή αυτού του bit είναι προαιρετική. Το πρόγραμμα της εφεδρικής αποθήκευσης βασίζεται στο ότι τα άλλα προγράμματα θα χρησιμοποιήσουν αυτό το bit σωστά. Ορισμένα προγράμματα ωστόσο, μπορεί να μεταβάλλουν το αρχείο χωρίς όμως να μεταβάλλουν και αυτό το bit, ευτυχώς όμως τα περισσότερα μεταβάλλουν και τα δύο.

#### 4.3.6 Συστοιχίες (Μονάδες κατανομής)

Η μικρότερη μονάδα χώρου ενός σκληρού δίσκου, που το λογισμικό μπορεί να έχει πρόσβαση είναι το τομέας, ο οποίος περιλαμβάνει 512 bytes. Είναι πιθανό να υπάρχει ένα σύστημα κατανομής του δίσκου, που σε κάθε αρχείο να εκχωρούνται τόσοι ξεχωριστοί τομείς, όσους χρειάζεται. Για παράδειγμα, ένα αρχείο μεγέθους 1 MB θα απαιτούσε περίπου 2048 ξεχωριστούς τομείς, για να αποθηκεύσει τα δεδομένα του. Κάτω από το FAT σύστημα αρχείων (και τα περισσότερα άλλα) δε χρησιμοποιούνται ξεχωριστοί τομείς. Υπάρχουν αρκετοί λόγοι απόδοσης για αυτό. Θα ήταν πολύ δύσκολο να διαχειριστεί το λειτουργικό σύστημα το δίσκο αν τα αρχεία ήταν σπασμένα σε κομμάτια των 512 bytes. Ένα διαμέρισμα 2 GB, χρησιμοποιώντας ανεξάρτητους τομείς των 512 bytes, θα χρειαζόταν 4 εκατομμύρια ξεχωριστούς τομείς και το να κρατήσει πληροφορίες για τον καθένα θα απαιτούσε πολύ χρόνο και χώρο. Κάποια λειτουργικά συστήματα κατανέμουν το χώρο στα αρχεία με τους τομείς, αλλά απαιτούν κάποια ανεπτυγμένη νοημοσύνη για να το κάνουν αυτό σωστά. Το FAT σχεδιάστηκε πολλά χρόνια πριν, είναι ένα απλό σύστημα αρχείων και δεν μπορεί να διαχειριστεί ξεχωριστούς τομείς. Αυτό που κάνει το FAT αντίθετα, είναι να ομαδοποιεί τους τομείς σε μεγαλύτερες ομάδες, οι οποίες ονομάζονται συστοιχίες ή μονάδες κατανομής. Το μέγεθος της συστοιχίας καθορίζεται κυρίως από το μέγεθος του διαμερίσματος: μιλώντας γενικά, μπορούμε να πούμε ότι μεγαλύτερα διαμερίσματα χρησιμοποιούν μεγαλύτερου μεγέθους συστοιχίες. Για κάθε διαμέρισμα, το μέγεθος της συστοιχίας κυμαίνεται από 4 τομείς (2048 bytes) μέχρι 64 τομείς (32768 bytes). Οι εύκαμπτοι δίσκοι χρησιμοποιούν πολύ μικρότερες συστοιχίες και σε κάποιες περιπτώσεις χρησιμοποιούν ακόμα και συστοιχίες του ενός



τομέα. Οι τομείς μιας συστοιχίας είναι συνεχόμενοι, έτσι οι συστοιχίες καταλαμβάνουν ένα συνεχόμενο χώρο στο δίσκο. Το μέγεθος της συστοιχίας (και συνεπώς και το μέγεθος του διαμερίσματος) έχει μεγάλη επιρροή στην απόδοση και στη χρησιμοποίηση του δίσκου. Το μέγεθος της συστοιχίας αποφασίζεται όταν δημιουργούνται τα διαμερίσματα. Κάποια προγράμματα (όπως το Partition Magic ) μπορούν να τροποποιήσουν το μέγεθος της συστοιχίας ενός ήδη υπάρχοντος διαμερίσματος (μέσα σε όρια), αλλά τις περισσότερες φορές όταν καθορίζεται το μέγεθος του διαμερίσματος καθορίζεται και το μέγεθος της συστοιχίας. Κάθε αρχείο πρέπει να κατανέμεται σε ένα ακέραιο αριθμό από συστοιχίες. Μια συστοιχία είναι η μικρότερη μονάδα χώρου του δίσκου που μπορεί να κατανεμηθεί σε ένα αρχείο και για αυτό το λόγο οι συστοιχίες πολλές φορές ονομάζονται μονάδες κατανομής. Αυτό σημαίνει ότι αν ένα διαμέρισμα χρησιμοποιεί συστοιχίες μεγέθους 8192 bytes, ένα αρχείο μεγέθους 8000 bytes χρησιμοποιεί μια συστοιχία (8192 bytes στο δίσκο), ενώ ένα αρχείο μεγέθους 8300 bytes χρησιμοποιεί δύο συστοιχίες (16384 bytes στο δίσκο). Γι' αυτό το λόγο το μέγεθος της συστοιχίας είναι τόσο σημαντικό στην προσπάθεια αποδοτικής χρήσης του δίσκου, μεγαλύτερο μέγεθος συστοιχίας συνεπάγεται περισσότερο χαμένο χώρο στο δίσκο.

#### 4.3.7 Αλυσιδωτή οργάνωση αρχείων και κατανομή των συστοιχιών στο FAT

Ο πίνακας κατανομής αρχείων (FAT) χρησιμοποιείται για να κρατάει πληροφορίες για το ποιές συστοιχίες ανήκουν σε ποιά αρχείο. Το λειτουργικό σύστημα και κατά επέκταση κάθε εφαρμογή λογισμικού, μπορεί να αποφασίσει που είναι τοποθετημένα τα δεδομένα ενός αρχείου, χρησιμοποιώντας την καταχώρηση καταλόγου για το αρχείο και τις καταχωρήσεις στον πίνακα κατανομής αρχείων. Παρομοίως, ο πίνακας κατανομής αρχείων κρατάει στοιχεία για το ποιές συστοιχίες είναι ελεύθερες για χρήση. Όταν μια εφαρμογή χρειάζεται να δημιουργήσει ή να επεκτείνει ένα αρχείο, απαιτεί περισσότερες συστοιχίες από το λειτουργικό σύστημα, το οποίο τις βρίσκει στο πίνακα κατανομής αρχείων. Υπάρχει μια καταχώρηση στον πίνακα κατανομής

αρχείων για κάθε συστοιχία που χρησιμοποιείται στο δίσκο. Κάθε καταχώρηση περιέχει μια τιμή που φανερώνει πως χρησιμοποιείται η κάθε συστοιχία. Υπάρχουν διαφορετικοί κώδικες που χρησιμοποιούνται για τις διάφορες πιθανές καταστάσεις, στις οποίες μια συστοιχία μπορεί να βρίσκεται. Κάθε συστοιχία που είναι σε χρήση από ένα αρχείο, έχει στην καταχώρησή της στο FAT, ένα αριθμό συστοιχίας που συνδέει την τρέχουσα συστοιχία με την επόμενη συστοιχία που χρησιμοποιεί το αρχείο. Αυτή η συστοιχία έχει στην καταχώρησή της, τον αριθμό της συστοιχίας μετά από αυτή. Η τελευταία συστοιχία που χρησιμοποιείται από ένα αρχείο είναι σημειωμένη με ένα ειδικό κώδικα, που λέει στο σύστημα ότι είναι η τελευταία συστοιχία του αρχείου. Συχνά είναι ένα νούμερο όπως το 65535 (11 μονάδες στο δυαδικό σύστημα). Καθώς οι συστοιχίες είναι συνδεδεμένες με αυτό τον τρόπο, συνηθίζεται να λέγεται ότι είναι αλυσιδωτές. Κάθε αρχείο, που χρησιμοποιεί περισσότερες από μια συστοιχίες, είναι αλυσιδωτό με αυτό τον τρόπο. Εκτός από τον αριθμό της επόμενης συστοιχίας ή τον κωδικό που χρησιμοποιείται για να φανερώνει τη τελευταία συστοιχία ενός αρχείου, η καταχώρηση μιας συστοιχίας μπορεί να περιέχει κάποιους άλλους ειδικούς κώδικες για να δείχνουν την κατάστασή της. Ένας ειδικός κώδικας, συνήθως το μηδέν, τίθεται στην καταχώρηση του FAT, κάθε μη χρησιμοποιούμενης συστοιχίας. Αυτό φανερώνει στο λειτουργικό σύστημα ποιές συστοιχίες είναι ελεύθερες για να χρησιμοποιηθούν από αρχεία που χρειάζονται και άλλο αποθηκευτικό χώρο. Ένας άλλος κώδικας χρησιμοποιείται για να δείξει τις “χαλασμένες” συστοιχίες. Υπάρχουν συστοιχίες, στις οποίες ένα βοηθητικό πρόγραμμα δίσκου (disk utility), έχει προηγουμένως εντοπίσει έναν ή περισσότερους ελαττωματικούς τομείς, εξαιτίας ελαττωμάτων του δίσκου. Αυτές οι συστοιχίες σημειώνονται ως ελαττωματικές έτσι ώστε να μην προσπαθήσει κανένα πρόγραμμα να τις χρησιμοποιήσει.

#### **4.4 Μειονεκτήματα συστήματος αρχείων FAT**

Ο τρόπος που το FAT σύστημα αρχείων κατανέμει το χώρο και συνδέει τα αρχεία, προκαλεί μερικά συνηθισμένα λάθη τα οποία αναδύονται με τον καιρό. Πρόκειται για λογικά λάθη στο δίσκο και όχι για φυσικά προβλήματα, όπως είναι οι

καταστραμμένοι τομείς. Τα περισσότερα από αυτά τα προβλήματα μπορούν να εντοπιστούν από ειδικά προγράμματα που αναλύουν ολόκληρο το σύστημα. Τα λάθη του συστήματος αρχείων είναι μερικές φορές το αποτέλεσμα μιας βλάβης στο δίσκο, που η ρίζα της μπορεί να είναι κάποιο πρόβλημα στο hardware. Αυτά τα λάθη συνεπώς προκύπτουν από κάποιο πρόβλημα του συστήματος, το οποίο μπορεί να προκαλέσει φθορά στο δίσκο, όπως είναι ο ανταγωνισμός για πόρους, ελαττωματικοί οδηγοί και άλλα. Τις περισσότερες φορές, ωστόσο, τα προβλήματα του συστήματος αρχείων προκύπτουν από προβλήματα του λογισμικού. Το “κρέμασμα” κάποιων προγραμμάτων για παράδειγμα, συχνά αφήνει διάσπαρτες συστοιχίες, στις οποίες έχει κατανεμηθεί χώρος στο δίσκο, αλλά δεν αντιστοιχούν σε κανένα αρχείο. Ένα πρόβλημα τροφοδοσίας σε ένα υπολογιστή την ώρα που αυτό τρέχει Windows τις περισσότερες φορές προκαλεί ένα ή περισσότερα προβλήματα, εξαιτίας κάποιων αρχείων που δεν έχουν κλείσει σωστά. Για αυτό το λόγο υποτίθεται ότι πρέπει πάντα να κλείνουν τα Windows πριν κλείσει ο υπολογιστής. Τα παρακάτω είναι τα πιο συνηθισμένα λάθη που συναντώνται σε ένα FAT δίσκο:

- **Χαμένες συστοιχίες:** Ουσιαστικά κάθε χρήστης DOS έχει αντιμετωπίσει αυτό το πρόβλημα. Οι χαμένες συστοιχίες είναι στην ουσία συστοιχίες, οι οποίες είναι σημειωμένες στον πίνακα κατανομής αρχείων ως χρησιμοποιούμενες, αλλά το σύστημα δεν μπορεί να τις ενώσει με κανένα αρχείο. Προγράμματα ελέγχου του δίσκου μπορούν να ψάξουν για χαμένες συστοιχίες σε ολόκληρο το δίσκο, ακολουθώντας την παρακάτω διαδικασία ή κάποια παρόμοια. Στην αρχή δημιουργούν ένα αντίγραφο του πίνακα κατανομής αρχείων στην κύρια μνήμη, σημειώνοντας όλα τα αρχεία που είναι σημειωμένα ως χρησιμοποιούμενα. Αρχίζοντας από τον κατάλογο της ρίζας, ακολουθούν τις συστοιχίες που χρησιμοποιούνται από κάθε αρχείο σημειώνοντας τα ως “εξηγήσιμα”, καθώς αυτά είναι συνδεδεμένα με ένα αρχείο. Μετά κάνει την ίδια διαδικασία για όλους τους υποκαταλόγους της ρίζας και μετά για τους δικούς τους υποκαταλόγους και πάει λέγοντας. Όταν τελειώσει τα αρχεία που είναι σημειωμένα στον πίνακα κατανομής ως χρησιμοποιούμενα πρέπει να είναι και εξηγήσιμα. Όσα δεν είναι δεν ανήκουν σε κανένα αρχείο. Οι χαμένες συστοιχίες είναι συνήθως το αποτέλεσμα της διακοπής της δραστηριότητας

του αρχείου με κάποιο τρόπο. Το πρόγραμμα που τα εντοπίζει, παρέχει συνήθως και τη δυνατότητα καθαρισμού τους, κάνοντας τα πάλι διαθέσιμα ή σώζοντας τα σαν ένα αρχείο. Στην τελευταία περίπτωση, το πρόγραμμα δημιουργεί ένα όνομα αρχείου και ενώνει τις χαμένες συστοιχίες με αυτό. Συνήθως αυτό το αρχείο μετά θα καταστραφεί, αλλά τουλάχιστον μπορούμε να δούμε τα χαμένα δεδομένα και σε μερικές περιπτώσεις να τα επαναφέρουμε.

- **Cross-Linked Αρχεία:** Σε σπάνιες περιπτώσεις δύο αρχεία μπορεί να δείχνουν στα ίδια δεδομένα στο δίσκο. Και τα δύο αρχεία θα έχουν τον ίδιο αριθμό αρχικής συστοιχίας στην καταχώρηση του καταλόγου, δείχνοντας στην ίδια συστοιχία. Αυτό είναι πρόβλημα, καθώς κάθε φορά που χρησιμοποιείται το ένα αρχείο θα γράφεται πάνω στο άλλο. Η μόνη λύση σε αυτό το πρόβλημα είναι η δημιουργία αντιγράφων για κάθε ένα από τα σχετιζόμενα αρχεία. Γενικά, θα χαθούν τα περιεχόμενα του ενός από τα δύο αρχεία (στην πραγματικότητα, από τη στιγμή που θα ανακαλυφθεί το πρόβλημα αυτό θα έχουν χαθεί τα περιεχόμενα του ενός αρχείου, καθώς κανένα τομέας δεν μπορεί να περιέχει πληροφορίες για δύο αρχεία την ίδια στιγμή).
- **Άκυρα αρχεία ή κατάλογοι:** Πολύ σπάνια, οι εσωτερικές δομές ενός αρχείου ή καταλόγου μπορεί να υποστούν βλάβη, οπότε κάποιες καταχωρήσεις να μην ακολουθούν τους κανόνες, σχετικά με το πως ένα αρχείο ή κατάλογος υποτίθεται ότι είναι σχεδιασμένος. Ένα παράδειγμα είναι ένας κατάλογος, ο οποίος δεν έχει ένα δείκτη προς το γονικό του κατάλογο ή ένα αρχείο που δεν έχει ένα έγκυρο αριθμό αρχικής συστοιχίας. Τα προβλήματα αυτά μπορούν να εντοπιστούν και να διορθωθούν από προγράμματα ελέγχου του δίσκου.
- **Λάθη στον Πίνακα Καταχώρησης Αρχείων:** Μερικές φορές οι καταχωρήσεις ενός πίνακα καταχωρήσεων μπορούν να αλλάξουν ή να πάρουν άκυρες τιμές. Τα προβλήματα αυτά μπορούν, πάλι, να εντοπιστούν και να διορθωθούν από προγράμματα ελέγχου του δίσκου.

#### 4.5 Σύγκριση συστημάτων αρχείων NTFS και FAT

Το NTFS είναι το προτιμώμενο σύστημα αρχείων των Windows. Πλεονεκτεί σε πολλά σημεία ως προς το παλαιότερο σύστημα αρχείων FAT32, όπως:

- Τη δυνατότητα αυτόματης αποκατάστασης της σωστής λειτουργίας μετά από ορισμένα σφάλματα που παρουσιάζονται στους δίσκους, δυνατότητα που δεν υπάρχει στο FAT32.
- Τη βελτιωμένη υποστήριξη για σκληρούς δίσκους μεγαλύτερης χωρητικότητας.
- Τη βελτιωμένη ασφάλεια, χάρη στη δυνατότητα χρήσης δικαιωμάτων και κρυπτογράφησης, ώστε η πρόσβαση σε κάποια αρχεία να επιτρέπεται μόνο σε συγκεκριμένους χρήστες που έχετε εγκρίνει.

Το FAT32, καθώς και το σπανιότερα χρησιμοποιούμενο FAT, ήταν τα συστήματα αρχείων που χρησιμοποιήθηκαν σε παλαιότερες εκδόσεις λειτουργικών συστημάτων των Windows, όπως τα Windows 95, Windows 98 και Windows Millennium Edition . Το FAT32 δεν διαθέτει τα χαρακτηριστικά ασφαλείας που παρέχει το NTFS. Αυτό σημαίνει ότι αν σε ένα διαμέρισμα ή τόμο του υπολογιστή σας χρησιμοποιείτε το FAT32, τότε κάθε χρήστης που έχει πρόσβαση στον υπολογιστή σας θα μπορεί να διαβάσει οποιοδήποτε αρχείο περιέχει. Επίσης, το σύστημα αρχείων FAT32 έχει περιορισμούς ως προς το μέγεθος. Δεν μπορείτε να δημιουργήσετε ένα διαμέρισμα FAT32 που να υπερβαίνει τα 32GB. Επίσης, δεν μπορείτε να αποθηκεύσετε ένα αρχείο με μέγεθος μεγαλύτερο από 4GB σε ένα διαμέρισμα FAT32. Ο κύριος λόγος για τον οποίο μπορεί να χρησιμοποιηθεί το σύστημα αρχείων FAT32 είναι εάν ο υπολογιστής σας πρόκειται να λειτουργεί κάποιες φορές με Windows 95, Windows 98 ή Windows Millennium Edition , και αν πρόκειται για υπολογιστή με ρυθμίσεις παραμέτρων πολλαπλής εκκίνησης. Σε μια τέτοια περίπτωση, πρέπει να εγκαταστήσετε το παλαιότερο λειτουργικό σύστημα σε ένα διαμέρισμα FAT32 ή FAT και να βεβαιωθείτε ότι πρόκειται για πρωτεύον διαμέρισμα (δηλαδή διαμέρισμα που μπορεί να φιλοξενήσει ένα λειτουργικό σύστημα). Επίσης, πρέπει να διαμορφώσετε με το σύστημα αρχείων FAT32 όλα τα διαμερίσματα στα οποία θέλετε να έχετε πρόσβαση όταν χρησιμοποιείτε αυτές τις παλαιότερες εκδόσεις των Windows. Αυτές οι παλαιότερες εκδόσεις των Windows υποστηρίζουν την πρόσβαση

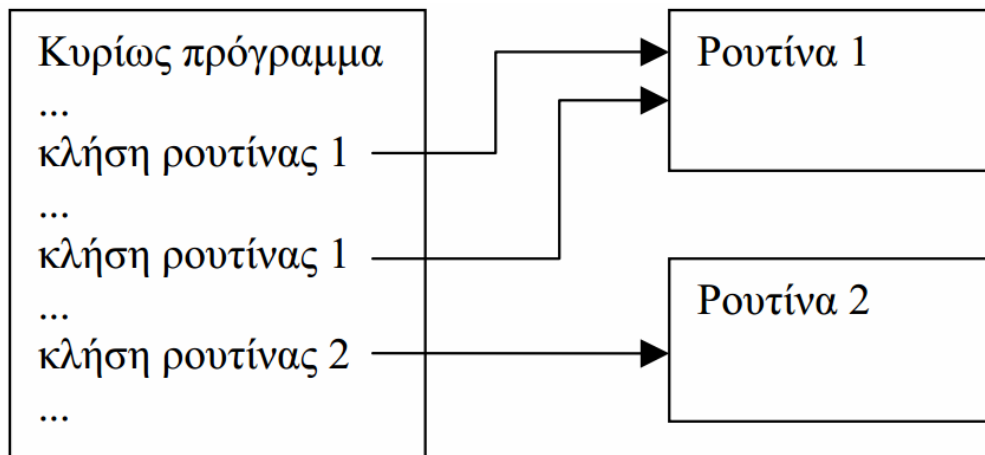
σε διαμερίσματα ή τόμους NTFS μέσω δικτύου, αλλά όχι στον ίδιο τον υπολογιστή σας.

## **Κεφάλαιο 5: Υπορουτίνες**

### **5.1 Εισαγωγή**

Οι υπορουτίνες αποτελούν αυτόνομα τμήματα κώδικα που διεκπεραιώνουν μία συγκεκριμένη εργασία και μπορούμε να τις καλούμε μέσα από το κυρίως πρόγραμμα όσες φορές χρειαστεί ( π.χ. κώδικας που εναλλάσσει δύο θέσεις μνήμης, κώδικας που κάνει σύνθετες πράξεις, κώδικας πρόσθεσης αριθμών πολλών Bytes ). Τα πλεονεκτήματα της χρήσης υπορουτινών είναι τα εξής:

- Συμβάλλουν στη μείωση του συνολικού όγκου του κώδικα σε περίπτωση ύπαρξης πολλών επαναλαμβανόμενων τμημάτων.
- Καθιστούν την διόρθωση του κώδικα ευκολότερη γιατί διορθώνουμε τον κώδικα της ρουτίνας μια φορά και αυτό ισχύει για όλο το πρόγραμμα.
- Κάνουν τον κώδικα πιο κατανοητό και δίνουν τη δυνατότητα επαναχρησιμοποίησης του.



**Σχήμα 1.**Κλήση υπορουτινών μέσα από το κυρίως πρόγραμμα.

## 5.2 INT ( εντολή x86)

Η INT είναι μία εντολή σε γλώσσα ASSEMBLY για επεξεργαστές x86, που προκαλεί μία διακοπή λογισμικού. Παίρνει τον αριθμό διακοπής σαν τιμή byte. Όταν χρησιμοποιηθεί γράφεται ως INT X, όπου το X είναι η διακοπή λογισμικού που πρέπει να παραχθεί(0-255). Ανάλογα με το πρόγραμμα που χρησιμοποιούμε, ο αριθμός διακοπής λογισμικού συχνά δίνεται σαν δεκαεξαδική τιμή, μερικές φορές με το πρόθεμα 0x ή το επίθεμα h. Για παράδειγμα, η διακοπή λογισμικού int 21h, (33 στο δεκαδικό σύστημα), προκαλεί τη λειτουργία που υποδεικνύεται από το 34<sup>ο</sup> διάλυσμα του πίνακα διακοπών να εκτελεστεί.

## 5.3 Υπορουτίνες που χρησιμοποιήθηκαν για την εργασία

Σε αυτή την ενότητα θα παρουσιαστούν αναλυτικά οι υπορουτίνες υλικού και λογισμικού που χρησιμοποιήθηκαν για την ολοκλήρωση της πτυχιακής εργασίας αναλυτικά, έτσι ώστε να παρουσιαστούν πληροφορίες για αυτές, πιθανά προβλήματα (bugs), με στόχο να δοθεί μία ολοκληρωμένη εικόνα για αυτές και το τι προσφέρει η κάθε μία στον προγραμματιστή.

### 5.3.1 Διακοπή Debugging INT3

Η διακοπή αυτή χρησιμοποιείται έξυπνα από το πρόγραμμα MONITOR, κατά την εκτέλεση ενός προγράμματος, με καθορισμό διευθύνσεων παύσης (breakpoints). Μπορεί να χρησιμοποιηθεί ως η τελευταία εντολή ενός προγράμματος, καθώς μόλις εκτελείται, εμφανίζει τις τιμές των καταχωρητών και επιστρέφει στο πρόγραμμα MONITOR.

### 5.3.2 Διακοπή INT10h



Διακοπή που παρέχει υπηρεσίες βίντεο. Οι υπηρεσίες αυτές περιλαμβάνουν τη ρύθμιση της λειτουργίας βίντεο, την εμφάνιση χαρακτήρων, και αρχέτυπα γραφικών (γραφή και ανάγνωση pixels σε λειτουργία γραφικών).

- INT 10h / AH = 2 - Ορίζει τη θέση του κέρσορα στην οθόνη. Είσοδος DH= γραμμή, DL= στήλη, BH= αριθμός σελίδας(0..7).

Παράδειγμα :

```
mov dh, 10
```

```
mov dl, 20
```

```
mov bh, 0
```

```
mov ah, 2
```

```
int10h
```

- INT 10h/ AH = 0Eh – Εκτύπωση χαρακτήρα στην οθόνη.

Είσοδος: AL = χαρακτήρας προς εκτύπωση. Εκτυπώνει έναν χαρακτήρα στην οθόνη, προχωρά τον κέρσορα και ολισθαίνει την οθόνη όσο χρειάζεται, η εκτύπωση γίνεται πάντα στην ενεργή σελίδα video.

Παράδειγμα:

```
mov al, 'a'
```

mov ah, 0eh

int10h

### 5.3.3 Διακοπή INT16h

Διακοπή που προσφέρει υπηρεσίες πληκτρολογίου.

- INT 16h/ AH= 00h – Διαβάζει ένα πλήκτρο από το πληκτρολόγιο (χωρίς ηχώ στην οθόνη) Επιστρέφει: AH= ο scan code του BIOS. AL = ο ASCII χαρακτήρας. (αν έχει πατηθεί πλήκτρο, αυτό αφαιρείται από τον buffer του πληκτρολογίου).

### 5.3.4 Διακοπή INT13h

Διακοπή η οποία παρέχει υπηρεσίες ανάγνωσης και εγγραφής σε σκληρό δίσκο και σε δισκέτα χρησιμοποιώντας την CHS μέθοδο.

- INT 13h/ AH= 00h – Εκτελεί επαναφορά (reset) του συστήματος δίσκων. Σε πολλά συστήματα υπάρχει η πιθανότητα να γίνει επαναφορά και στον κύριο (master) αλλά και στον ακόλουθο (slave) σκληρό δίσκο ταυτόχρονα.

- INT 13h/ AH= 02h – Διαβάζει sectors από το δίσκο και τα αντιγράφει στη μνήμη.

Πιθανά σφάλματα οφείλονται στον κινητήρα ο οποίος αποτυγχάνει να γυρίσει αρκετά γρήγορα. Η διαδικασία ανάγνωσης πρέπει να γίνει τουλάχιστον τρεις φορές, με την επαναφορά των δίσκων μεταξύ των προσπαθειών.

- INT 13h/ AH= 03h – Γράφει sectors από τη μνήμη στο δίσκο.

Πιθανά σφάλματα οφείλονται στον κινητήρα ο οποίος αποτυγχάνει να γυρίσει αρκετά γρήγορα. Η διαδικασία εγγραφής πρέπει να γίνει τουλάχιστον τρεις φορές, με την επαναφορά των δίσκων μεταξύ των προσπαθειών.

Είσοδος:

AL= αριθμός των sectors που θα διαβαστούν/εγγραφούν(πρέπει να είναι ≠0)

CH= αριθμός κυλίνδρου(0..79).

CL= αριθμός sector (1..18).

DH= αριθμός κεφαλής(0..1).

DL= αριθμός δίσκου(0..3).

ES:BX δείκτης στα δεδομένα.

Επιστρέφει:

CF γίνεται 1 αν υπάρχει σφάλμα.

CF γίνεται 0 αν η εγγραφή/ανάγνωση είναι επιτυχής.

AH= κατάσταση(0 – αν είναι επιτυχής).

AL= αριθμός sectors που μεταφέρθηκαν.

## **Κεφάλαιο 6: Διαχείριση εργασίας και προτάσεις για μελλοντική έρευνα.**

### **6.1 Εισαγωγή**

Στο παρόν κεφάλαιο παρουσιάζεται συνοπτικά η διαχείριση της πτυχιακής εργασίας. Στη συνέχεια γίνεται αναφορά σε επισημάνσεις, βελτιώσεις και προτάσεις για περαιτέρω έρευνα σχετικά με τη βελτίωση του κώδικα.

### **6.2 Διαχείριση εργασίας**

Το συνολικό διάστημα που δαπανήθηκε για την ολοκλήρωση της πτυχιακής εργασίας είναι τρία εξάμηνα, από τον Μάιο 2012 μέχρι τον Νοέμβριο του 2013. Το διάστημα αυτό είναι αρκετά μεγάλο, αλλά η εκπόνηση της πρακτικής εργασίας συνέβαλε σε αυτό. Ο πραγματικός χρόνος που χρειάστηκε για να ολοκληρωθεί αυτή η εργασία είναι περίπου 9 μήνες, συμπεριλαμβανομένου και του παρόντος συγγράμματος.

Αρχικά πραγματοποιήθηκε ανάλυση απαιτήσεων της εργασίας μέσα από έντυπες και ηλεκτρονικές πηγές, έτσι ώστε να επιτευχθεί μία καλύτερη πορεία γύρω από την επίτευξη των στόχων. Η αναζήτηση πληροφοριών αρχικά εστιάστηκε στις υπορουτινές καθαυτές, τη σύνταξη τους και τη σωστή δομή τους. Στη συνέχεια έγινε μία ανάλυση γύρω από τις απαιτήσεις του κυρίως μενού και τις δυνατότητες που θα προσέφερε αυτό στον χρήστη. Προτεραιότητα του κυρίως μενού αποτελούσε να είναι λιτό ενώ ταυτόχρονα να παρουσιάζει όλες τις δυνατότητες του στον χρήστη. Για την διαχείριση του κώδικα χρησιμοποιήθηκε emulator (emu 8086) ο οποίος ήταν ικανός να προσφέρει τις υπηρεσίες του bgc-8088 σε ηλεκτρονικό υπολογιστή.

Τα διάφορα προβλήματα που προέκυψαν κατά τη διάρκεια συγγραφής του κώδικα, ήταν κυρίως τεχνικής φύσεως και συγκεκριμένα του οδηγού δισκέτας. Η επίλυση των προβλημάτων αυτών επιτεύχθηκε σε συνεννόηση με τον επιβλέποντα Καθηγητή.

### 6.3 Προτάσεις εξέλιξης

Ο συγκεκριμένος κώδικας προσφέρει πολλές δυνατότητες εξέλιξης παρότι ο οδηγός δισκέτας και η δισκέτα, δεν χρησιμοποιούνται ευρέως σήμερα, παρά μόνο σε καταστάσεις ανάγκης. Πάνω στην συγκεκριμένη εργασία θα μπορούσε να επιτευχθεί η δημιουργία ενός πλήρους μηχανισμού αναζήτησης αρχείων σε μονάδα δίσκου και δισκέτας. Στο κυρίως μενού θα μπορούσε να προστεθεί η επιλογή αναζήτησης αρχείων. Με αυτήν τη δυνατότητα θα μπορούσαμε να αναζητήσουμε μεγάλο όγκο αρχείων με τα ονόματά τους χωρίς να πρέπει να ψάχνουμε κάθε εγγραφή του ριζικού

καταλόγου, κάτι το οποίο θα έκανε τη χρησιμοποίηση του συγκεκριμένου κώδικα ευκολότερη και γρηγορότερη. Επίσης η δυνατότητα μετακίνησης και η αντιγραφή και επικόλληση των αρχείων σε άλλες καταχωρήσεις του ριζικού καταλόγου θα έδιναν περισσότερη ευκολία στη διαχείριση τους. Με τις συγκεκριμένες δυνατότητες ο κώδικας θα μετατρεπόταν σε μία πλήρη μορφή διαχείρισης αρχείων σε δισκέτα.

## Κεφάλαιο 7: Ο κώδικας της εφαρμογής και αποτελέσματα

Αρχικά, ξεκινώντας την εφαρμογή εμφανίζεται το κυρίως μενού μέσα από το οποίο μπορούμε να επιλέξουμε τη λειτουργία την οποία επιθυμούμε.

```
WELCOME
SUBROUTINE PROGRAM VERSION EMU.0.2
CHOOSE OPERATION
1. FILE OPERATIONS
2. INFO
3. CHECK FAT
4. CHECK ROOT
5. EXIT PROGRAM
_
```

## Εικόνα 8.1 Κυρίως Μενού

```
#make_bin#  
  
include 'emu8086.inc'  
  
DEFINE_CLEAR_SCREEN  
  
welcome:  
  
CALL CLEAR_SCREEN  
  
print " WELCOME"  
  
gotoxy 0,1  
  
print " SUBROUTINE PROGRAM VERSION EMU.0.2"  
  
gotoxy 0,2  
  
print " CHOOSE OPERATION"  
  
gotoxy 0,3  
  
print "1. FILE OPERATIONS"  
  
gotoxy 0,4  
  
print "2. INFO"  
  
gotoxy 0,5  
  
print "3. CHECK FAT"  
  
gotoxy 0,6  
  
print "4. CHECK ROOT"  
  
gotoxy 0,7  
  
print "5. EXIT PROGRAM"  
  
gotoxy 0,8  
  
mov ah,00  
  
int 16h  
  
cmp al,30h  
  
je welcome  
  
cmp al,31h  
  
je fileoperations  
  
cmp al,32h
```

je INFO

CMP AL,33H

je fat

cmp al,34h

je rot

cmp al,35h

je end

jnle welcome

Επιλέγοντας το info, εμφανίζεται ένα παράθυρο με τις πληροφορίες της πτυχιακής εργασίας.



**Εικόνα 8.2** Πληροφορίες

INFO:

CALL CLEAR\_SCREEN

PRINT " TEI SERRWN PLIROFORIKIS K EPIKOINWNIWN"

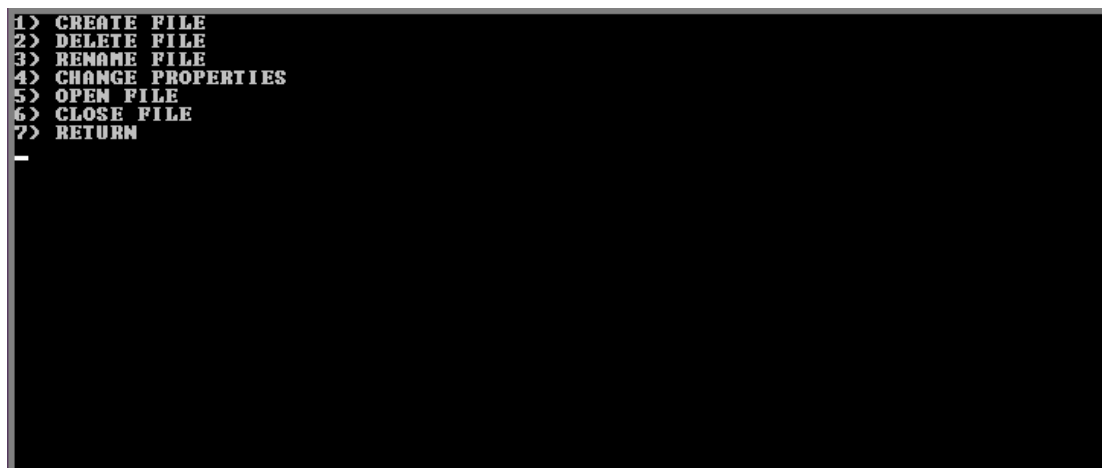
GOTOXY 0,1

PRINT " SPIROS KAZARLIS"



```
GOTOXY 0,2  
  
PRINT " DIMITRIS TZIATZIOS 1615"  
  
GOTOXY 0,3  
  
PRINT " ALL RIGHTS RESERVED"  
  
GOTOXY 0,4  
  
PRINT " PRESS ANY KEY TO CONTINUE"  
  
GOTOXY 0,5  
  
MOV AH,00  
  
INT 16H  
  
JMP WELCOME
```

Επιλέγοντας file operations, εμφανίζεται το κυρίως παράθυρο από όπου μπορούμε να διαχειριστούμε τα αρχεία ή να τα δημιουργήσουμε.



**Εικόνα 8.3** Μενού διαχείρισης αρχείων

```

FILEOPERATIONS:

CALL CLEAR_SCREEN

start:

CALL CLEAR_SCREEN

MOV SI,00

STRA:

mov [si+0d000h],00

inc si

cmp si,200h

jne STRA

PRINT "1) CREATE FILE"

GOTOXY 0,1

PRINT "2) DELETE FILE"

GOTOXY 0,2

PRINT "3) RENAME FILE"

GOTOXY 0,3

PRINT "4) CHANGE PROPERTIES"

GOTOXY 0,4

PRINT "5) OPEN FILE"

GOTOXY 0,5

PRINT "6) CLOSE FILE"

GOTOXY 0,6

PRINT "7) RETURN"

GOTOXY 0,7

mov ah,00

int 16h

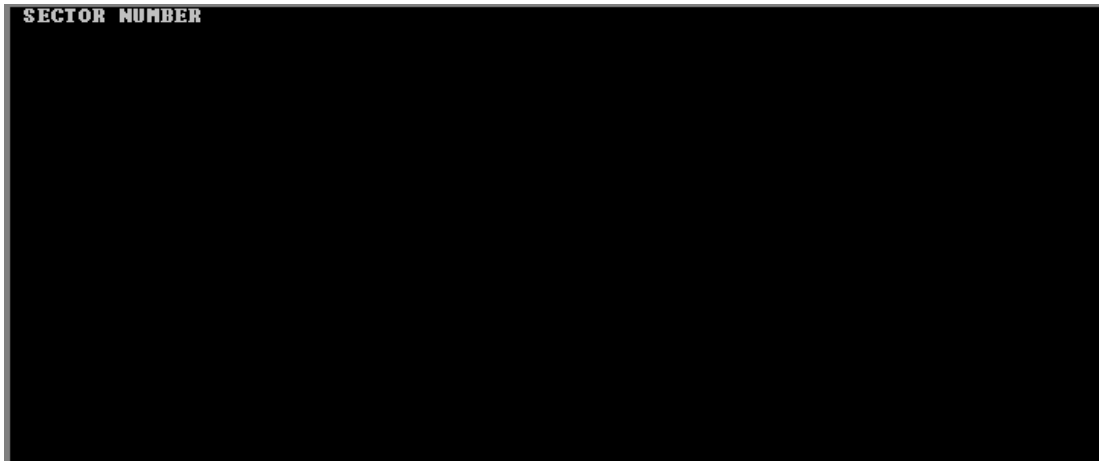
cmp al,30h

je start

```

```
cmp al,31h
je create
cmp al,32h
je delete
cmp al,33h
je rename
cmp al,34h
je change
cmp al,35h
je open
CMP AL,36H
JE CLOSE
CMP AL,37H
JE WELCOME
jnle start
CLOSE:
CALL CLEAR_SCREEN
MOV SI,00
RETRI:
mov [SI+0af00h],00
INC SI
CMP SI,200H
JNE RETRI
JMP FILEOPERATIONS
```

Επιλέγοντας open file, μπορούμε να ανοίξουμε το αρχείο που δημιουργήσαμε δίνοντας τον αριθμό του sector.



**Εικόνα 8.4** Αριθμός sector

open:

```
CALL CLEAR_SCREEN
```

```
mov [0b270h],00
```

```
mov [0b272h],00
```

```
mov [0b273h],00
```

```
mov [0b274h],00
```

```
srat32:
```

```
loop3333:
```

```
xor ax,ax
```

```
xor dx,dx
```

```
xor bx,bx
```

```
xor cx,cx
```

```
mov [0b400h],00
```

```
mov [0b401h],00
```

```
mov [0b500h],00
```

```
mov [0b501h],00
```

```
mov [0b600h],00
```

```
mov [0f900h],00
```

```
mov [0b400h],00
mov [0b601h],00
call clear_Screen
print "SECTOR NUMBER"
mov ah,00
int 16h
mov [0b400h],al
sub [0b400h],30h
mov bl,[0b400h]
mov ax,100
mul bl
mov [0b401h],ax
mov dx,[0b401h]
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,10
mul bl
mov [0b501h],ax
mov cx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bx,[0b600h]
add dx,cx
add dx,bx
```

```
mov [0b400h],dx
mov ax,dx
cmp dx,21h
jl srat32
cmp dx,2cfh
xor bx,bx
mov bl,31
sub ax,bx
mov si,ax
mov dx,[0b400h]
xor ax,ax
mov ax,dx
mov bl,12h
div bl
mov [0b270h],al
mov bh,ah
xor ax,ax
mov al,bh
xor bx,bx
mov bl,09
div bl
mov [0b272h],al
mov [0b273h],ah
mov di,[0b273h]
xor ax,ax
mov al,[0b273h]
xor bx,bx
mov bl,09
div bl
```

```
mov [0b274h],ah
xor ax,ax
xor dx,dx
xor bx,bx
xor cx,cx
mov ax,0100h
mov es,ax
mov bx,0af00h
mov ah,02h
mov al,01
mov ch,[0b270h]
mov cl,[0b274h]
mov dh,[0b272h]
mov dl,00
int 13h
mov [0b270h],00
mov [0b272h],00
mov [0b273h],00
mov [0b274h],00
jmp start
```

Αν επιλέξουμε δημιουργία αρχείου, διαλέγουμε τον αριθμό του sector στον οποίο θέλουμε να αποθηκεύσουμε το αρχείο, και στη συνέχεια επιλέγουμε από ποιο σημείο του sector θέλουμε να ξεκινάει και το μέγεθος των δεδομένων που θα γράψουμε στο αρχείο. Στη συνέχεια επιλέγουμε αν θέλουμε να γράψουμε χαρακτήρες ή bytes.

FOR BYTES PRESS 0 , FOR CHARS PRESS 1

**Εικόνα 8.5** Επιλογή bytes ή χαρακτήρων για εγγραφή

Αφού επιλέξουμε τι θα γράψουμε στο αρχείο, πρέπει να δώσουμε το όνομα του αρχείου και την κατάληξή του, 8 bytes για το όνομα και 3 για την κατάληξη. Το όνομα πρέπει να είναι πάντα με κεφαλαία και αν είναι μικρότερο των 8 bytes, συμπληρώνουμε τα υπόλοιπα με το space button.

Τέλος επιλέγουμε σε ποιο sector του Root directory θέλουμε να αποθηκεύσουμε το αρχείο και σε ποιά 32άδα μέσα σε αυτό.

create:

```
CALL CLEAR_SCREEN
```

```
mov ax,0100h
```

```
mov es,ax
```

```
mov bx,0a000h
```

```
mov [0a000h+si],240
```

```
mov [0a001h+si],255
```

```
mov [0a002h+si],255
```

```
mov ah,03h
```

```
mov al,09
```



```
mov ch,00
mov cl,02
mov dh,00
mov dl,00
int 13h
srat:
loop333:
mov [0d000h],00
inc si
cmp si,200h
jne loop333
xor ax,ax
xor dx,dx
xor bx,bx
xor cx,cx
mov [0b400h],00
mov [0b401h],00
mov [0b500h],00
mov [0b501h],00
mov [0b600h],00
mov [0f900h],00
mov [0b400h],00
mov [0f800h],00
mov [0f802h],00
CALL CLEAR_SCREEN
PRINT " SECTOR NUMBER"
mov ah,00
int 16h
mov [0b400h],al
```

```
sub [0b400h],30h
mov bl,[0b400h]
mov ax,100
mul bl
mov [0b401h],ax
mov dx,[0b401h]
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,10
mul bl
mov [0b501h],ax
mov cx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bx,[0b600h]

add dx,cx
add dx,bx
mov [0b400h],dx
mov ax,dx
cmp dx,21h
cmp dx,2cfh
xor bx,bx
mov bl,31
```

```
sub ax,bx

mov si,ax

mov [0f900h],si

;mov [0b400h],si

mov ax,0100h

mov es,ax

mov bx,0a000h

mov [0a001h+si],255

mov [0a002h+si],15

mov ah,03h

mov al,09

mov ch,00

mov cl,02

mov dh,00

mov dl,00

int 13h

mov ax,0100h

mov es,ax

mov bx,0e000h

mov ah,02h

mov al,09

mov ch,00

mov cl,02

mov dh,00

mov dl,00

int 13h

mov dx,[0b400h]

xor ax,ax

mov ax,dx
```

```
mov bl,12h
div bl
mov [0b270h],al
mov bh,ah
xor ax,ax
mov al,bh
xor bx,bx
mov bl,09
div bl
mov [0b272h],al
mov [0b273h],ah
mov di,[0b273h]
xor ax,ax
mov al,[0b273h]
xor bx,bx
mov bl,09
div bl
mov [0b274h],ah
mov ah,00
int 10h
mov ah,2
mov dh,1
mov dl,0
mov bh,0
int 10h
mov ah,0eh
mov al,20h
int 10h
mov al,53h
```

int 10h  
mov al,54h  
int 10h  
mov al,41h  
int 10h  
mov al,52h  
int 10h  
mov al,54h  
int 10h  
mov al,49h  
int 10h  
mov al,4eh  
int 10h  
mov al,47h  
int 10h  
mov al,20h  
int 10h  
mov al,42h  
int 10h  
mov al,59h  
int 10h  
mov al,54h  
int 10h  
mov al,45h  
int 10h  
mov al,20h  
int 10h  
mov al,26h  
int 10h

```
mov al,20h
int 10h
mov al,53h
int 10h
mov al,49h
int 10h
mov al,5ah
int 10h
mov al,45h
int 10h
mov ax,00
mov dx,00
mov bx,00
loop:
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,100
mul bl
mov [0b501h],ax
mov dx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bl,[0b600h]
mov ax,10
```

```
mul bl

mov [0b601h],ax

mov cx,[0b601h]

mov ah,00

int 16h

mov [0b700h],al

sub [0b700h],30h

mov bx,[0b700h]

add dx,cx

add dx,bx

mov [0b500h],dx

cmp dx,1ffh

jnl loop

mov si,dx

xor ax,ax

loop2:

mov ah,00

int 16h

mov [0b500h],al

sub [0b500h],30h

mov bl,[0b500h]

mov ax,100

mul bl

mov [0b501h],ax

mov dx,[0b501h]

mov ah,00

int 16h

mov [0b600h],al

sub [0b600h],30h
```

```

mov bl,[0b600h]

mov ax,10

mul bl

mov [0b601h],ax

mov cx,[0b601h]

mov ah,00

int 16h

mov [0b700h],al

sub [0b700h],30h

mov bx,[0b700h]

add dx,cx

add dx,bx

cmp dx,00

je loop2

mov [0b500h],dx

mov [0f990h],dx

cmp dx,200h

jnle loop2

mov di,dx

xor dx,dx

mov dx,00h

mov [0b000h],dx

cmp [0b000h],00h

jnle loop

mov [0b650h],di

call clear_screen

boc:

print " FOR BYTES PRESS 0 , FOR CHARS PRESS 1"

mov ah,00

```



```
int 16h
cmp al,30h
je bytes
cmp al,31h
je chars
jnle boc
BYTES:
mov ah,00
int 16h
cmp al,41h
je adc
cmp al,42h
je adc1
cmp al,43h
je adc2
cmp al,44h
je adc3
cmp al,45h
je adc4
cmp al,46h
je adc5
jne addp
adc:
mov [0b900h],160
jmp add0
adc1:
mov [0b900h],176
jmp add0
adc2:
```

```
mov [0b900h],192
jmp add0
adc3:
mov [0b900h],208
jmp add0
adc4:
mov [0b900h],224
jmp add0
adc5:
mov [0b900h],240
jmp add0
addp:
sub al,30h
mov bl,10
mul bl
mov [0b900h],al
add0:
mov ah,00
int 16h
cmp al,41h
je adc0
cmp al,42h
je adc01
cmp al,43h
je adc02
cmp al,44h
je adc03
cmp al,45h
je adc04
```

```
cmp al,46h
je adc05
jne addp1
adc0:
mov [0b902h],10
jmp add00
adc01:
mov [0b902h],11
jmp add00
adc02:
mov [0b902h],12
jmp add00
adc03:
mov [0b902h],13
jmp add00
adc04:
mov [0b902h],14
jmp add00
adc05:
mov [0b902h],15
jmp add00
addp1:
sub al,30h
mov [0b902h],al
add00:
mov bh,[0b900h]
mov bl,[0b902h]
add bh,bl
mov [si+0d000h],bh
```

```
inc si

dec di

cmp di,00

je fin

jnle bytes

cmp [0b00h],00h

jnle bytes

fin:

mov ax,0100h

mov es,ax

mov bx,0d000h

mov ah,03h

mov al,01

mov ch,[0b270h]

mov cl,[0b274h]

mov dh,[0b272h]

mov dl,00

int 13h

mov ax,0100h

mov es,ax

mov bx,0af00h

mov ah,02h

mov al,01

mov ch,[0b270h]

mov cl,[0b274h]

mov dh,[0b272h]

mov dl,00

int 13h

JMP ROOT
```

```
chars:
mov ah,00
int 16h
mov [si+0d000h],al
inc si
dec di
cmp di,00
jne chars
mov ax,0100h
mov es,ax
mov bx,0d000h
mov ah,03h
mov al,01
mov ch,[0b270h]
mov cl,[0b274h]
mov dh,[0b272h]
mov dl,00
int 13h
mov ax,0100h
mov es,ax
mov bx,0af00h
mov ah,02h
mov al,01
mov ch,[0b270h]
mov cl,[0b274h]
mov dh,[0b272h]
mov dl,00
int 13h
mov [0b270h],00
```

```
mov [0b272h],00
mov [0b273h],00
mov [0b274h],00
jmp ROOT
ROOT:
mov ah,2
mov dh,4
mov dl,0
mov bh,0
int 10h
call clear_screen
print " NAME 8 bytes AND EXTENSION 3 bytes"
mov ah,00
int 16h
mov [0a300h],al
mov ah,00
int 16h
mov [0a301h],al
mov ah,00
int 16h
mov [0a302h],al
mov ah,00
int 16h
mov [0a303h],al
mov ah,00
int 16h
mov [0a304h],al
mov ah,00
int 16h
```

```
mov [0a305h],al
mov ah,00
int 16h
mov [0a306h],al
mov ah,00
int 16h
mov [0a307h],al
mov ah,00
int 16h
mov [0a308h],al
mov ah,00
int 16h
mov [0a309h],al
mov ah,00
int 16h
mov [0a30ah],al
mov [0a30bh],21h
mov [0a30ch],64h
mov [0a30fh],7dh
mov [0a310h],14h
mov [0a311h],2bh
mov [0a312h],46h
mov [0a313h],00h
mov [0a316h],24h
mov [0a317h],28h
mov [0a318h],46h
mov [0a319h],00h
mov [0a312h],46h
mov [0a316h],24h
```

```
mov [0a317h],28h

mov ah,2

mov dh,6

mov dl,0

mov bh,0

int 10h

ROOTENTRYcr:

call clear_screen

print " ROOT ENTRY 00 - 13"

mov ah,00

int 16h

sub al,30h

mov bl,10

mul bl

mov [0a203h],al

mov ah,00

int 16h

sub al,30h

mov [0a205h],al

mov bh,[0a203h]

mov bl,[0a205h]

add bh,bl

cmp bh,00

je root1

cmp bh,01

je rooot

cmp bh,02

je rooot1

cmp bh,03
```



```
je rooot2
cmp bh,04
je rooot3
cmp bh,05
je rooot4
cmp bh,06
je rooot5
cmp bh,07
je rooot6
cmp bh,08
je rooot7
cmp bh,09
je rooot8
cmp bh,10
je rooot9
cmp bh,11
je rooot10
cmp bh,12
je rooot11
cmp bh,13
je rooot12
jnle ROOTENTRYcr
root1:
mov [0f800h],02
mov [0f802h],00
jmp rootpos
root:
mov [0f800h],03
mov [0f802h],00
```

```
jmp rootpos  
  
root1:  
  
mov [0f800h],04  
  
mov [0f802h],00  
  
jmp rootpos  
  
root2:  
  
mov [0f800h],05  
  
mov [0f802h],00  
  
jmp rootpos  
  
root3:  
  
mov [0f800h],06  
  
mov [0f802h],00  
  
jmp rootpos  
  
root4:  
  
mov [0f800h],07  
  
mov [0f802h],00  
  
jmp rootpos  
  
root5:  
  
mov [0f800h],08  
  
mov [0f802h],00  
  
jmp rootpos  
  
root6:  
  
mov [0f800h],09  
  
mov [0f802h],00  
  
jmp rootpos  
  
root7:  
  
mov [0f800h],01  
  
mov [0f802h],01  
  
jmp rootpos
```

```
root8:
mov [0f800h],02
mov [0f802h],01
jmp rootpos

root9:
mov [0f800h],03
mov [0f802h],01
jmp rootpos

root10:
mov [0f800h],04
mov [0f802h],01
jmp rootpos

root11:
mov [0f800h],05
mov [0f802h],01
jmp rootpos

root12:
mov [0f800h],06
mov [0f802h],01
jmp rootpos

rootpos:
mov ah,2
mov dh,8
mov dl,0
mov bh,0
int 10h
call clear_screen
print " ROOT POSITION 00 - 15"

loop1111:
```

```
mov ah,00
int 16h
mov [0a500h],al
sub [0a500h],30h
mov bl,[0a500h]
cmp bl,02h
jnl loop1111
mov ah,00
int 16h
mov [0a501h],al
sub [0a501h],30h
mov bl,[0a500h]
mov ax,10
mul bl
mov [0a502h],ax
mov bl,[0a502h]
mov cl,[0a501h]
add cl,bl
mov [0a503h],cl
cmp cl,16h
jnl loop1111
mov ax,20h
mul cl
mov si,ax
mov al,[0a300h]
mov [si+0a600h],al
mov al,[0a301h]
mov [si+0a601h],al
mov al,[0a302h]
```

```
mov [si+0a602h],al
mov al,[0a303h]
mov [si+0a603h],al
mov al,[0a304h]
mov [si+0a604h],al
mov al,[0a305h]
mov [si+0a605h],al
mov al,[0a306h]
mov [si+0a606h],al
mov al,[0a307h]
mov [si+0a607h],al
mov al,[0a308h]
mov [si+0a608h],al
mov al,[0a309h]
mov [si+0a609h],al
mov al,[0a30ah]
mov [si+0a60ah],al
mov al,[0a30bh]
mov [si+0a60bh],al
mov al,[0a30eh]
mov [si+0a60eh],al
mov al,[0a30fh]
mov [si+0a60fh],al
mov al,[0a310h]
mov [si+0a610h],al
mov al,[0a311h]
mov [si+0a611h],al
mov al,[0a312h]
mov [si+0a612h],al
```

```
mov al,[0a313h]
mov [si+0a613h],al
mov al,[0a316h]
mov [si+0a616h],al
mov al,[0a317h]
mov [si+0a617h],al
mov al,[0a318h]
mov [si+0a618h],al
mov al,[0a319h]
mov [si+0a619h],al
mov dx,[0f900h]
mov [si+0a61ah],dx
mov dx,[0f990h]
mov [si+0a61ch],dx
mov al,[si+0a6bh]
mov [0a400h],al
mov ax,0100h
mov es,ax
mov bx,0a600h
mov ah,03h
mov al,01
mov ch,01
mov cl,[0f800h]
mov dh,[0f802h]
mov dl,00
int 13h
jmp start
```

Η επιλογή delete, σβήνει το περιεχόμενο του αρχείου δίνοντας τον αριθμό του sector καθώς και τον αριθμό του Root sector και της 32άδας μέσα σε αυτό που επιλέξαμε στη δημιουργία.

```
delete:
```

```
CALL CLEAR_SCREEN
```

```
srat2:
```

```
xor ax,ax
```

```
xor dx,dx
```

```
xor bx,bx
```

```
xor cx,cx
```

```
mov [0b400h],00
```

```
mov [0b401h],00
```

```
mov [0b500h],00
```

```
mov [0b501h],00
```

```
mov [0b600h],00
```

```
mov [0f900h],00
```

```
mov [0b400h],00
```

```
mov [0f800h],00
```

```
mov [0f802h],00
```

```
mov [0b270h],00
```

```
mov [0b272h],00
```

```
mov [0b273h],00
```

```
mov [0b274h],00
```

```
mov [0b601h],00
```

```
CALL CLEAR_SCREEN
```

```
PRINT " SECTOR NUMBER"
```

```
mov ah,00
```

```
int 16h
```

```
mov [0b400h],al
sub [0b400h],30h
mov bl,[0b400h]
mov ax,100
mul bl
mov [0b401h],ax
mov dx,[0b401h]
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,10
mul bl
mov [0b501h],ax
mov cx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bx,[0b600h]
add dx,cx
add dx,bx
mov ax,dx
cmp dx,21h
jl srat2

xor bx,bx
mov bl,31
```



```
sub ax,bx

mov si,ax

xor ax,ax

mov ax,dx

mov bl,12h

div bl

mov [0b270h],al

mov bh,ah

xor ax,ax

mov al,bh

xor bx,bx

mov bl,09

div bl

mov [0b272h],al

mov [0b273h],ah

mov di,[0b273h]

xor ax,ax

mov al,[0b273h]

xor bx,bx

mov bl,09

div bl

mov [0b274h],ah

cmp [0a002h+si],00

je error

jne con

error:

print " FILE DOESNT EXIST"

jmp srat2

mov [0b500h],00
```

```
mov [0f990h],00
con:
mov [0f900h],si
mov [0b400h],si
mov ax,0100h
mov es,ax
mov bx,0a000h
mov [0a001h+si],15
mov [0a002h+si],00
mov ah,03h
mov al,09
mov ch,00
mov cl,02
mov dh,00
mov dl,00
int 13h
mov ax,0100h
mov es,ax
mov bx,0e000h
mov ah,02h
mov al,09
mov ch,00
mov cl,02
mov dh,00
mov dl,00
int 13h
mov si,00
clear:
mov [si+0d000h],00
```

```
inc si
cmp si,200h
jne clear
mov ax,0100h
mov es,ax
mov bx,0d000h
mov si,00
mov ah,03h
mov al,01
mov ch,[0b270h]
mov cl,[0b274h]
mov dh,[0b272h]
mov dl,00
int 13h
mov ax,0100h
mov es,ax
mov bx,0af00h
mov ah,02h
mov al,01
mov ch,[0b270h]
mov cl,[0b274h]
mov dh,[0b272h]
mov dl,00
int 13h
mov [0a300h],00
mov [0a301h],00
mov [0a302h],00
mov [0a303h],00
mov [0a304h],00
```

```
mov [0a305h],00
mov [0a306h],00
mov [0a307h],00
mov [0a308h],00
mov [0a309h],00
mov [0a30ah],00
mov [0a30bh],00
mov [0a30eh],00
mov [0a30fh],00
mov [0a310h],00
mov [0a311h],00
mov [0a312h],00
mov [0a313h],00
mov [0a316h],00
mov [0a317h],00
mov [0a318h],00
mov [0a319h],00
mov [0a312h],00
mov [0a316h],00
mov [0a317h],00
mov ah,2
mov dh,6
mov dl,0
mov bh,0
int 10h
ROOTENTRY2:
call clear_screen
print " ROOT ENTRY 0 - 13"
mov ah,00
```

```
int 16h
sub al,30h
mov bl,10
mul bl
mov [0a203h],al
mov ah,00
int 16h
sub al,30h
mov [0a205h],al
mov bh,[0a203h]
mov bl,[0a205h]
add bh,bl
cmp bh,00
je roott1
cmp bh,01
je rooott
cmp bh,02
je rooott1
cmp bh,03
je rooott2
cmp bh,04
je rooott3
cmp bh,05
je rooott4
cmp bh,06
je rooott5
cmp bh,07
je rooott6
cmp bh,08
```

```
je rooott7
cmp bh,09
je rooott8
cmp bh,10
je rooott9
cmp bh,11
je rooott10
cmp bh,12
je rooott11
cmp bh,13
je rooott12
jnle ROOTENTRY2
roott1:
mov [0f800h],02
mov [0f802h],00
jmp root001
rooott:
mov [0f800h],03
mov [0f802h],00
jmp root001
rooott1:
mov [0f800h],04
mov [0f802h],00
jmp root001
rooott2:
mov [0f800h],05
mov [0f802h],00
jmp root001
rooott3:
```

```
mov [0f800h],06
mov [0f802h],00
jmp root001
roott4:
mov [0f800h],07
mov [0f802h],00
jmp root001
roott5:
mov [0f800h],08
mov [0f802h],00
jmp root001
roott6:
mov [0f800h],09
mov [0f802h],00
jmp root001
roott7:
mov [0f800h],01
mov [0f802h],01
jmp root001
roott8:
mov [0f800h],02
mov [0f802h],01
jmp root001
roott9:
mov [0f800h],03
mov [0f802h],01
jmp root001
roott10:
mov [0f800h],04
```

```

mov [0f802h],01

jmp root001

roott11:

mov [0f800h],05

mov [0f802h],01

jmp root001

roott12:

mov [0f800h],06

mov [0f802h],01

jmp root001

root001:

mov ah,2

mov dh,8

mov dl,0

mov bh,0

int 10h

call clear_screen

print " ROOT POSITION 0 - 15"

ROOT0001:

mov ah,00

int 16h

mov [0a500h],al

sub [0a500h],30h

mov bl,[0a500h]

cmp bl,02h

jnl ROOT0001

mov ah,00

int 16h

mov [0a501h],al

```



```
sub [0a501h],30h
mov bl,[0a500h]
mov ax,10
mul bl
mov [0a502h],ax
mov bl,[0a502h]
mov cl,[0a501h]
add cl,bl
mov [0a503h],cl
cmp cl,16h
jnl root001
mov ax,20h
mul cl
mov si,ax
mov al,[0a300h]
mov [si+0a600h],00
mov al,[0a301h]
mov [si+0a601h],00
mov al,[0a302h]
mov [si+0a602h],00
mov al,[0a303h]
mov [si+0a603h],00
mov al,[0a304h]
mov [si+0a604h],00
mov al,[0a305h]
mov [si+0a605h],00
mov al,[0a306h]
mov [si+0a606h],00
mov al,[0a307h]
```

```
mov [si+0a607h],00
mov al,[0a308h]
mov [si+0a608h],00
mov al,[0a309h]
mov [si+0a609h],00
mov al,[0a30ah]
mov [si+0a60ah],00
mov al,[0a30bh]
mov [si+0a60bh],00
mov al,[0a30eh]
mov [si+0a60eh],00
mov al,[0a30fh]
mov [si+0a60fh],00
mov al,[0a310h]
mov [si+0a610h],00
mov al,[0a311h]
mov [si+0a611h],00
mov al,[0a312h]
mov [si+0a612h],00
mov al,[0a313h]
mov [si+0a613h],00
mov al,[0a316h]
mov [si+0a616h],00
mov al,[0a317h]
mov [si+0a617h],00
mov al,[0a318h]
mov [si+0a618h],00
mov al,[0a319h]
mov [si+0a619h],00
```

```
mov [si+0a60ah],00
mov [si+0a609h],00
mov [si+0a61ah],00
mov [si+0a61bh],00
mov [si+0a608h],00
mov [si+0a60bh],00
mov [si+0a60eh],00
mov [si+0a60fh],00
mov [si+0a616h],00
mov [si+0a612h],00
mov [si+0a61ch],00
mov [si+0a61dh],00
mov ax,0100h
mov es,ax
mov bx,0a600h
mov ah,03h
mov al,01
mov ch,01
mov cl,[0f800h]
mov dh,[0f802h]
mov dl,00
int 13h
jmp start
```

Αν επιλέξουμε την μετονομασία αρχείου, δίνοντας τον αριθμό του sector που αποθηκεύσαμε το αρχείο, μπορούμε να δώσουμε στο αρχείο μας νέο όνομα και κατάληξη.

```

rename:

CALL CLEAR_SCREEN

srat3:

xor ax,ax

xor dx,dx

xor bx,bx

xor cx,cx

xor si,si

mov [0b400h],00

mov [0b401h],00

mov [0b500h],00

mov [0b501h],00

mov [0b600h],00

mov [0f900h],00

mov [0b400h],00

mov [0f800h],00

mov [0f802h],00

mov [0b270h],00

mov [0b272h],00

mov [0b273h],00

mov [0b274h],00

mov [0b601h],00

CALL CLEAR_SCREEN

PRINT " SECTOR NUMBER"

mov ah,00

int 16h

mov [0b400h],al

sub [0b400h],30h

mov bl,[0b400h]

```

```
mov ax,100
mul bl
mov [0b401h],ax
mov dx,[0b401h]
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,10
mul bl
mov [0b501h],ax
mov cx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bx,[0b600h]
add dx,cx
add dx,bx
mov ax,dx
cmp dx,21h
jl sr3
xor bx,bx
mov bl,31
sub ax,bx
mov si,ax
mov [0f900h],si
cmp [0a002h+si],00
```

```
je error0

jne conn

error0:

print " FILE DOESNT EXIST"

jmp srat3

conn:

mov ah,2 ; "ROOT"

mov dh,4

mov dl,0

mov bh,0

int 10h

call clear_screen

print " NAME 8 bytes AND EXTENSION 3 bytes"

mov ah,00

int 16h

mov [0a300h],al

mov ah,00

int 16h

mov [0a301h],al

mov ah,00

int 16h

mov [0a302h],al

mov ah,00

int 16h

mov [0a303h],al

mov ah,00

int 16h

mov [0a304h],al

mov ah,00
```

```
int 16h
mov [0a305h],al
mov ah,00
int 16h
mov [0a306h],al
mov ah,00
int 16h
mov [0a307h],al
mov ah,00
int 16h
mov [0a308h],al
mov ah,00
int 16h
mov [0a309h],al
mov ah,00
int 16h
mov [0a30ah],al
ROOTENTRY0:
call clear_screen
print " ROOT ENTRY 0 - 13"
mov ah,00
int 16h
sub al,30h
mov bl,10
mul bl
mov [0a203h],al
mov ah,00
int 16h
sub al,30h
```

```
mov [0a205h],al
mov bh,[0a203h]
mov bl,[0a205h]
add bh,bl
cmp bh,00
je erooot1
cmp bh,01
je rooot
cmp bh,02
je rooot1
cmp bh,03
je rooot2
cmp bh,04
je rooot3
cmp bh,05
je rooot4
cmp bh,06
je rooot5
cmp bh,07
je rooot6
cmp bh,08
je rooot7
cmp bh,09
je rooot8
cmp bh,10
je rooot9
cmp bh,11
je rooot10
cmp bh,12
```



```
je rooot11
cmp bh,13
je rooot12
jnle ROOTENTRY0
erooot1:
mov [0f800h],02
mov [0f802h],00
jmp root111
rooot:
mov [0f800h],03
mov [0f802h],00
jmp root111
rooot1:
mov [0f800h],04
mov [0f802h],00
jmp root111
rooot2:
mov [0f800h],05
mov [0f802h],00
jmp root111
rooot3:
mov [0f800h],06
mov [0f802h],00
jmp root111
rooot4:
mov [0f800h],07
mov [0f802h],00
jmp root111
rooot5:
```

mov [0f800h],08

mov [0f802h],00

jmp root111

rooooot6:

mov [0f800h],09

mov [0f802h],00

jmp root111

rooooot7:

mov [0f800h],01

mov [0f802h],01

jmp root111

rooooot8:

mov [0f800h],02

mov [0f802h],01

jmp root111

rooooot9:

mov [0f800h],03

mov [0f802h],01

jmp root111

rooooot10:

mov [0f800h],04

mov [0f802h],01

jmp root111

rooooot11:

mov [0f800h],05

mov [0f802h],01

jmp root111

rooooot12:

```
mov [0f800h],06
mov [0f802h],01
jmp root111
root111:
mov ah,2
mov dh,8
mov dl,0
mov bh,0
int 10h
call clear_screen
print " ROOT POSITION 0 - 15"
loop101:
mov ah,00
int 16h
mov [0a500h],al
sub [0a500h],30h
mov bl,[0a500h]
cmp bl,02h
jnl loop101
mov ah,00
int 16h
mov [0a501h],al
sub [0a501h],30h
mov bl,[0a500h]
mov ax,10
mul bl
mov [0a502h],ax
mov bl,[0a502h]
mov cl,[0a501h]
```

```
add cl,bl
mov [0a503h],cl
cmp cl,16h
jnl loop101
mov ax,20h
mul cl
mov si,ax
mov al,[0a300h]
mov [si+0a600h],al
mov al,[0a301h]
mov [si+0a601h],al
mov al,[0a302h]
mov [si+0a602h],al
mov al,[0a303h]
mov [si+0a603h],al
mov al,[0a304h]
mov [si+0a604h],al
mov al,[0a305h]
mov [si+0a605h],al
mov al,[0a306h]
mov [si+0a606h],al
mov al,[0a307h]
mov [si+0a607h],al
mov al,[0a308h]
mov [si+0a608h],al
mov al,[0a309h]
mov [si+0a609h],al
mov al,[0a30ah]
mov [0a60ah],al
```

```
mov dx,[0f900h]

mov [si+0a61ah],dx

mov ax,0100h

mov es,ax

mov bx,0a600h

mov ah,03h

mov al,01

mov ch,01

mov cl,[0f800h]

mov dh,[0f802h]

mov dl,00

int 13h

jmp start
```

Η επιλογή change properties, μας δίνει τη δυνατότητα να αλλάξουμε την ιδιότητα του αρχείου.



**Εικόνα 8.6** Αλλαγή ιδιότητας του αρχείου

change:

```
CALL CLEAR_SCREEN
```

```
srat4:

xor ax,ax

xor dx,dx

xor bx,bx

xor cx,cx

xor si,si

mov [0b400h],00

mov [0b401h],00

mov [0b500h],00

mov [0b501h],00

mov [0b600h],00

mov [0f900h],00

mov [0b400h],00

mov [0f800h],00

mov [0f802h],00

mov [0b270h],00

mov [0b272h],00

mov [0b273h],00

mov [0b274h],00

mov [0b601h],00

CALL CLEAR_SCREEN

PRINT " SECTOR NUMBER"

mov ah,00

int 16h

mov [0b400h],al

sub [0b400h],30h

mov bl,[0b400h]

mov ax,100

mul bl
```

```
mov [0b401h],ax
mov dx,[0b401h]
mov ah,00
int 16h
mov [0b500h],al
sub [0b500h],30h
mov bl,[0b500h]
mov ax,10
mul bl
mov [0b501h],ax
mov cx,[0b501h]
mov ah,00
int 16h
mov [0b600h],al
sub [0b600h],30h
mov bx,[0b600h]
add dx,cx
add dx,bx
mov [0b400h],dx
mov ax,dx
cmp dx,21h
cmp dx,2cfh
xor bx,bx
mov bl,31
sub ax,bx
mov si,ax
mov [0f900h],si
cmp [0a002h+si],00
je error001
```

```
jne conn0

error001:

print " FILE DOESNT EXIST"

jmp srat4

error01:

print " FILE DOESNT EXIST"

jmp srat4

conn0:

call clear_screen

PRINT " CHOOSE PROPERTIES"

mov ah,2

mov dh,1

mov dl,0

mov bh,0

int 10h

print " 1) HIDDEN 0"

mov ah,2

mov dh,2

mov dl,0

mov bh,0

int 10h

PRINT " 2) SYSTEM 1"

mov ah,2

mov dh,3

mov dl,0

mov bh,0

int 10h

PRINT " 3) VOLUME LABEL 2"

mov ah,2
```



```
mov dh,4

mov dl,0

mov bh,0

int 10h

PRINT " 4) SUBDIRECTORY 3"

mov ah,2

mov dh,5

mov dl,0

mov bh,0

int 10h

PRINT " 5) ARCHIVE 4"

mov ah,2

mov dh,6

mov dl,0

mov bh,0

int 10h

PRINT " 6) HIDDEN SUBDIRECTORY 5"

mov ah,2

mov dh,7

mov dl,0

mov bh,0

int 10h

PRINT " 7) HIDDEN ARCHIVE 6"

MOV AH,00

INT 16H

cmp al,30h

je hidden

cmp al,31h

je system
```

```
cmp al,32h
je volume
cmp al,33h
je subd
cmp al,34h
je archive
cmp al,35h
je hsub
cmp al,36h
je harc
jnle conn
hidden:
mov [0a30bh],02h
jmp Rten
system:
mov [0a30bh],04h
jmp Rten
volume:
mov [0a30bh],08h
jmp Rten
subd:
mov [0a30bh],10h
jmp Rten
archive:
mov [0a30bh],20
jmp Rten
hsub:
mov [0a30bh],12h
jmp Rten
```

```
harc:
mov [0a30bh],22h
jmp Rten
Rten:
call clear_screen
print " ROOT ENTRY 0 - 13"
mov ah,00
int 16h
sub al,30h
mov bl,10
mul bl
mov [0a203h],al
mov ah,00
int 16h
sub al,30h
mov [0a205h],al
mov bh,[0a203h]
mov bl,[0a205h]
add bh,bl
cmp bh,00
je root1
cmp bh,01
je rooot
cmp bh,02
je rooot1
cmp bh,03
je rooot2
cmp bh,04
je rooot3
```

```
cmp bh,05
je rooot4
cmp bh,06
je rooot5
cmp bh,07
je rooot6
cmp bh,08
je rooot7
cmp bh,09
je rooot8
cmp bh,10
je rooot9
cmp bh,11
je rooot10
cmp bh,12
je rooot11
cmp bh,13
je rooot12
jnle Rten
root01:
mov [0f800h],02
mov [0f802h],00
jmp root711
rooot00:
mov [0f800h],03
mov [0f802h],00
jmp root711
rooot01:
mov [0f800h],04
```

```
mov [0f802h],00
jmp root711
root02:
mov [0f800h],05
mov [0f802h],00
jmp root711
root03:
mov [0f800h],06
mov [0f802h],00
jmp root711
root04:
mov [0f800h],07
mov [0f802h],00
jmp root711
root05:
mov [0f800h],08
mov [0f802h],00
jmp root711
root06:
mov [0f800h],09
mov [0f802h],00
jmp root711
root07:
mov [0f800h],01
mov [0f802h],01
jmp root711
root08:
mov [0f800h],02
mov [0f802h],01
```

```
jmp root711

root09:

mov [0f800h],03

mov [0f802h],01

jmp root711

root010:

mov [0f800h],04

mov [0f802h],01

jmp root711

root011:

mov [0f800h],05

mov [0f802h],01

jmp root711

root012:

mov [0f800h],06

mov [0f802h],01

jmp root711

root711:

mov ah,2

mov dh,8

mov dl,0

mov bh,0

int 10h

call clear_screen

print " ROOT POSITION 0 - 15"

loop51:

mov ah,00

int 16h

mov [0a500h],al
```

```
sub [0a500h],30h
mov bl,[0a500h]
cmp bl,02h
jnl loop51
mov ah,00
int 16h
mov [0a501h],al
sub [0a501h],30h
mov bl,[0a500h]
mov ax,10
mul bl
mov [0a502h],ax
mov bl,[0a502h]
mov cl,[0a501h]
add cl,bl
mov [0a503h],cl
cmp cl,16h
jnl root711
mov ax,20h
mul cl
mov si,ax
mov al,[0a30bh]
mov [si+0a60bh],al
mov dx,[0f900h]
mov [si+0a61ah],dx
mov ax,0100h
mov es,ax
mov bx,0a600h
mov ah,03h
```

```
mov al,01

mov ch,01

mov cl,[0f800h]

mov dh,[0f802h]

mov dl,00

int 13h

jmp start

fat:

mov ax,0100h

mov es,ax

mov bx,0a000h

mov [0a000h+si],240

mov [0a001h+si],255

mov [0a002h+si],255

mov ah,03h

mov al,09

mov ch,00

mov cl,02

mov dh,00

mov dl,00

int 13h

mov ax,0100h

mov es,ax

mov bx,0e000h

mov ah,02h

mov al,09

mov ch,00

mov cl,02

mov dh,00
```



```
mov dl,00

int 13h

jmp welcome

rot:

MOV [0A205H],00

MOV [0A203H],00

CALL CLEAR_SCREEN

PRINT " ROOT ENTRY 00 - 13"

mov ah,00

int 16h

sub al,30h

mov bl,10

mul bl

mov [0a203h],al

mov ah,00

int 16h

sub al,30h

mov [0a205h],al

mov bh,[0a203h]

mov bl,[0a205h]

add bh,bl

cmp bh,00

JE ROT1

CMP bh,01

JE ROT2

CMP bh,02

JE ROT3

CMP bh,03

JE ROT4
```

CMP bh,04  
JE ROT5  
CMP bh,05  
JE ROT6  
CMP bh,06  
JE ROT7  
CMP bh,07  
JE ROT8  
CMP bh,08  
JE ROT9  
CMP bh,09  
JE ROT10  
CMP bh,10  
JE ROT11  
CMP bh,11  
JE ROT12  
CMP bh,12  
JE ROT13  
CMP bh,13  
JE ROT14  
ROT1:  
mov ax,0100h  
mov es,ax  
mov bx,0f000h  
mov ah,02h  
mov al,01  
mov ch,01  
mov cl,02  
mov dh,00

```
mov dl,00
int 13h
jmp welcome
ROT2:
mov ax,0100h
mov es,ax
mov bx,0f000h
mov ah,02h
mov al,01
mov ch,01
mov cl,03
mov dh,00
mov dl,00
int 13h
jmp welcome
ROT3:
mov ax,0100h
mov es,ax
mov bx,0f000h
mov ah,02h
mov al,01
mov ch,01
mov cl,04
mov dh,00
mov dl,00
int 13h
jmp welcome
ROT4:
mov ax,0100h
```

```
mov es,ax
mov bx,0f000h
mov ah,02h
mov al,01
mov ch,01
mov cl,05
mov dh,00
mov dl,00
int 13h
jmp welcome
ROT5:
mov ax,0100h
mov es,ax
mov bx,0f000h
mov ah,02h
mov al,01
mov ch,01
mov cl,06
mov dh,00
mov dl,00
int 13h
jmp welcome
ROT6:
mov ax,0100h
mov es,ax
mov bx,0f000h
mov ah,02h
mov al,01
mov ch,01
```

```
mov cl,07  
  
mov dh,00  
  
mov dl,00  
  
int 13h  
  
jmp welcome  
  
ROT7:  
  
mov ax,0100h  
  
mov es,ax  
  
mov bx,0f000h  
  
mov ah,02h  
  
mov al,01  
  
mov ch,01  
  
mov cl,08  
  
mov dh,00  
  
mov dl,00  
  
int 13h  
  
jmp welcome
```

```
ROT8:  
  
mov ax,0100h  
  
mov es,ax  
  
mov bx,0f000h  
  
mov ah,02h  
  
mov al,01  
  
mov ch,01  
  
mov cl,09  
  
mov dh,00  
  
mov dl,00  
  
int 13h
```

```
jmp welcome

ROT9:

mov ax,0100h

mov es,ax

mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,01

mov dh,01

mov dl,00

int 13h

jmp welcome

ROT10:

mov ax,0100h

mov es,ax

mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,02

mov dh,01

mov dl,00

int 13h

jmp welcome

ROT11:

mov ax,0100h

mov es,ax
```

```
mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,03

mov dh,01

mov dl,00

int 13h

jmp welcome

ROT12:

mov ax,0100h

mov es,ax

mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,04

mov dh,01

mov dl,00

int 13h

jmp welcome

ROT13:

mov ax,0100h

mov es,ax

mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,05
```

```
mov dh,01

mov dl,00

int 13h

jmp welcome

ROT14:

mov ax,0100h

mov es,ax

mov bx,0f000h

mov ah,02h

mov al,01

mov ch,01

mov cl,06

mov dh,01

mov dl,00

int 13h

jmp welcome

end:

call clear_screen

print " THANK YOU FOR YOUR TIME"

HLT
```

Παρακάτω φαίνεται η δημιουργία ενός αρχείου με το όνομα MYFILE01.TXT το οποίο αλλάξαμε την ιδιότητα του σε κρυφό (hidden). Το συγκεκριμένο αρχείο καταλαμβάνει τον 33ό sector του FAT. Το μέγεθός του είναι 10 Bytes.



registers		1000:0000		1002:000C	
	H	L			
AX	0E	52	10000:	4D 077	M
BX	F0	00	10001:	59 089	Y
CX	01	02	10002:	46 070	F
DX	00	00	10003:	49 073	I
CS	F400		10004:	4C 076	L
IP	0190		10005:	45 069	E
SS	0100		10006:	30 048	0
SP	FFF4		10007:	31 049	1
BP	0000		10008:	54 084	T
SI	0070		10009:	58 088	X
DI	0000		1000A:	54 084	T
DS	0100		1000B:	02 002	0
ES	0100		1000C:	00 000	NULL
			1000D:	00 000	NULL
			1000E:	64 100	d
			1000F:	7D 125	>
			10010:	14 020	η
			10011:	2B 043	+
			10012:	46 070	F
			10013:	00 000	NULL
			10014:	00 000	NULL
			10015:	00 000	NULL
			10016:	24 036	\$
			10017:	28 040	<
			10018:	46 070	F
			10019:	00 000	NULL
			1001A:	02 002	0
			1001B:	00 000	NULL
			1001C:	0A 010	NEWL
			1001D:	00 000	NULL
			1001E:	00 000	NULL
			1001F:	00 000	NULL
			10020:	00 000	NULL
			10021:	00 000	NULL
			10022:	00 000	NULL
			10023:	00 000	NULL
			10024:	00 000	NULL
			10025:	00 000	NULL
			10026:	00 000	NULL
			10027:	00 000	NULL
			10028:	00 000	NULL
			10029:	00 000	NULL
			1002A:	00 000	NULL
			1002B:	00 000	NULL
			1002C:	00 000	NULL
			1002D:	00 000	NULL
			1002E:	00 000	NULL
			1002F:	00 000	NULL
			10030:	00 000	NULL
			10031:	00 000	NULL
			10032:	00 000	NULL
			10033:	00 000	NULL
			10034:	00 000	NULL
			10035:	00 000	NULL
			10036:	00 000	NULL
			10037:	00 000	NULL
			10038:	00 000	NULL
			10039:	00 000	NULL
			1003A:	00 000	NULL
					...

Εικόνα 8.7 Αποτέλεσμα Δημιουργίας αρχείου

Στην επόμενη εικόνα υπάρχει το FAT, το οποίο μας πληροφορεί πως η πρώτη εγγραφή του δηλαδή ο sector νούμερο 33 δεν είναι ελεύθερος και εκεί βρίσκεται το αρχείο που δημιουργήσαμε.

registers		e 000	0100:0053
	H	L	
AX	0E	4D	JMP 0147h
BX	F0	00	POP SI
CX	01	02	POP AX
DX	00	00	PUSH AX
CS	0100		PUSH BX
IP	0053		PUSH DX
SS	0100		MOU AH, 02h
SP	FFFA		MOU DH, 01h
BP	0000		MOU DL, 00h
SI	0042		MOU BH, 00h
DI	0000		INT 010h
DS	0100		POP DX
ES	0100		POP BX
		0F000: F0 240 Ω	POP AX
		0F001: FF 255 RES	PUSH AX
		0F002: FF 255 RES	PUSH SI
		0F003: FF 255 RES	JMP 08Fh
		0F004: 0F 015 *	AND [BP + DI] + 055h, DI
		0F005: 00 000 NULL	INC DX
		0F006: 00 000 NULL	PUSH DX
		0F007: 00 000 NULL	DEC DI
		0F008: 00 000 NULL	PUSH BP
		0F009: 00 000 NULL	PUSH SP
		0F00A: 00 000 NULL	DEC CX
		0F00B: 00 000 NULL	DEC SI
		0F00C: 00 000 NULL	INC BP
		0F00D: 00 000 NULL	AND [BX + SI] + 052h, DI
		0F00E: 00 000 NULL	DEC DI
		0F00F: 00 000 NULL	INC DI
		0F010: 00 000 NULL	PUSH DX
		0F011: 00 000 NULL	INC CX
		0F012: 00 000 NULL	DEC BP
		0F013: 00 000 NULL	AND [BP] + 045h, DL
		0F014: 00 000 NULL	PUSH DX
		0F015: 00 000 NULL	PUSH BX
		0F016: 00 000 NULL	DEC CX
		0F017: 00 000 NULL	DEC DI
		0F018: 00 000 NULL	DEC SI
		0F019: 00 000 NULL	AND [DI] + 04Dh, AL
		0F01A: 00 000 NULL	PUSH BP
		0F01B: 00 000 NULL	CS:
		0F01C: 00 000 NULL	XOR [00032h], CH
		0F01D: 00 000 NULL	MOU SI, 0006Bh
		0F01E: 00 000 NULL	CS:
		0F01F: 00 000 NULL	MOU AL, [SI]
		0F020: 00 000 NULL	CMP AL, 00h
		0F021: 00 000 NULL	JZ 0A0h
		0F022: 00 000 NULL	INC SI
		0F023: 00 000 NULL	MOU AH, 0Eh
		0F024: 00 000 NULL	INT 010h
		0F025: 00 000 NULL	JMP 092h
		0F026: 00 000 NULL	POP SI
		0F027: 00 000 NULL	POP AX
		0F028: 00 000 NULL	PUSH AX
		0F029: 00 000 NULL	PUSH BX
		0F02A: 00 000 NULL	PUSH DX
		0F02B: 00 000 NULL	MOU AH, 02h
		0F02C: 00 000 NULL	MOU DH, 02h
		0F02D: 00 000 NULL	...
		0F02E: 00 000 NULL	
		0F02F: 00 000 NULL	
		0F030: 00 000 NULL	
		0F031: 00 000 NULL	
		0F032: 00 000 NULL	
		0F033: 00 000 NULL	
		0F034: 00 000 NULL	
		0F035: 00 000 NULL	
		0F036: 00 000 NULL	
		0F037: 00 000 NULL	
		0F038: 00 000 NULL	
		0F039: 00 000 NULL	
		0F03A: 00 000 NULL	

screen source reset aux vars debug stack flags

Εικόνα 8.8 File Allocation Table

## **Βιβλιογραφία**

### **Ελληνική**

1. Δρ. Καζαρλής Σπ., Διαφάνειες Θεωρίας Αρχιτεκτονικής Υπολογιστών.
2. Δρ. Καζαρλής Σπ., Σημειώσεις Εργαστηρίου Αρχιτεκτονικής Υπολογιστών.
3. Μαδεμλής Ι., 8086 assembler tutorial I-O.
4. Δρ. Νικολαΐδης Α.(2012) Διαφάνειες θεωρίας Λειτουργικών Συστημάτων.
5. Επισκόπηση των συστημάτων αρχείων FAT, HPFS και NTFS. Διαθέσιμο στη διεύθυνση: <https://support.microsoft.com/kb/100108/el> [πρόσβαση 2013].
6. Μαγνητικά Μέσα Αποθήκευσης. Διαθέσιμο στη διεύθυνση [http://ebooks.edu.gr/modules/ebook/show.php/DSB103/173/1209\\_4422/](http://ebooks.edu.gr/modules/ebook/show.php/DSB103/173/1209_4422/) [πρόσβαση 2013].
7. Σύγκριση συστημάτων αρχείων NTFS και FAT . Διαθέσιμο στη διεύθυνση: <http://windows.microsoft.com/el-gr/windows-vista/comparing-ntfs-and-fat-file-systems> [πρόσβαση 2013].
8. Συστήματα αρχείων. Διαθέσιμο στη διεύθυνση: [http://www.it.uom.gr/project/mycomputer/storage/file\\_sys/](http://www.it.uom.gr/project/mycomputer/storage/file_sys/) [πρόσβαση 2013].

## **Βιβλιογραφία**

### **Ξενόγλωσση**

1. Brown Ralph, Ralf Brown's Interrupt List – Release 61, (2000).
2. Chidanandan Archana, An overview on FAT12, (2004-2005).
3. Eager Bob, A tutorial on the FAT file system, (2012).
4. Stoffregen Paul, Understanding FAT32 File Systems, (2005). Διαθέσιμο στη διεύθυνση: <http://www.pjrc.com/tech/8051/ide/fat32.html> [πρόσβαση 2013].
5. Verstak Alex, Introduction to FAT, (1998).
6. 8.3 FILENAME. Διαθέσιμο στη διεύθυνση: [http://en.wikipedia.org/wiki/8.3\\_filename](http://en.wikipedia.org/wiki/8.3_filename) [πρόσβαση 2013].
7. Cylinder-head-sector. Διαθέσιμο στη διεύθυνση: <http://en.wikipedia.org/wiki/Cylinder-head-sector> [πρόσβαση 2013].