





Θεωρητική Μελέτη των “Journaling Filesystems”

Της σπουδάστριας
Ποτσίκα Πολυξένης

10 Ιανουαρίου 2003

Πτυχιακή Εργασία μέρος των απαιτήσεων του Τμήματος
Τηλεπληροφορικής και Διοίκησης

ΕΥΧΑΡΙΣΤΙΕΣ

Στην εκπόνηση της παρούσας Πτυχιακής Εργασίας συνέβαλε τα μέγιστα ο επιβλέπων καθηγητής μου κ. Ι. Τσούλος. Νιώθω βαθύτατα την ανάγκη να τον ευχαριστήσω τόσο για την υπόδειξη του θέματος όσο και για την καθοδήγηση σε όλα τα στάδια εκπόνησης της εργασίας μου. Η βοήθειά του ήταν σημαντικότερη για την ολοκλήρωση του συγγραφικού μέρους της εργασίας. Τον ευχαριστώ επίσης για τις σημαντικότερες παρατηρήσεις, τα εύστοχα σχόλια και την αμέριστη συμπαράσταση κατά την διάρκεια της συγγραφής της εργασίας μου.

Τέλος, θα ήθελα να ευχαριστήσω τα μέλη της τριμελούς μου επιτροπής κ. Ε. Στεργίου, Καθηγητή του Τμήματος Τηλεπληροφορικής και Διοίκησης του ΤΕΙ ΗΠΕΙΡΟΥ, και τον κ. Α. Γέροντα, Επιστημονικό Συνεργάτη του Τμήματος Τηλεπληροφορικής και Διοίκησης του ΤΕΙ ΗΠΕΙΡΟΥ για την πολύτιμη συμβολή τους στη διόρθωση του τελικού κειμένου.

Abstract

Η Πτυχιακή Εργασία που παρουσιάζεται έχει σαν αντικείμενο την μελέτη των Journaling Filesystems – Συστήματα αρχείων. Το σύστημα αρχείων είναι το μέρος του λειτουργικού συστήματος που ασχολείται με τα αρχεία. Ένα σύστημα αρχείων αποτελεί τη μέθοδο ονομασίας, αποθήκευσης και οργάνωσης των αρχείων στον υπολογιστή μας. Το σύστημα αρχείων βελτιώνει την ασφάλεια, επιτρέποντας μας να κρυπτογραφούμε αρχεία και φακέλους και να περιορίζουμε την πρόσβαση σε αρχεία.

Αυτή η εργασία παρουσιάζει κάποιες βασικές έννοιες γύρω από τα αποθηκευτικά μέσα, τους τρόπους προσπελάσεως και υλοποίησης συστημάτων αρχείων. Επίσης, αναφέρεται σε διάφορων ειδών αρχεία όπως σε αρχεία χαρακτήρα, αρχεία μπλοκ και στην προσθήκη διαφορετικών συστημάτων αρχείων σε ένα σύστημα.

Τέλος, αναφέρεται στα συστήματα αρχείων FAT, EXT2, EXT3, REISERFS, XFS και πιο συγκεκριμένα αναφέρεται στα εργαλεία διαχείρισεως που χρησιμοποιούν, στις επιδόσεις τους, στη σύγκριση μεταξύ τους και σε άλλα θέματα που αποσκοπούν στην βαθύτερη ενημέρωση μας γύρω από τα λειτουργικά συστήματα όσον αφορά τα Journaling Filesystems αυτή τη περίοδο.

Περιεχόμενα

	Σελίδα
1. Βασικές Έννοιες	
1.1 Αποθηκευτικά Μέσα	2
1.2 Ορισμός συστήματος αρχείων	4
1.3 Τρόποι προσπελάσεως	4
1.4 Τρόποι υλοποίησης συστημάτων αρχείων	5
1.5 Μεταδεδομένα συστημάτων αρχείων	9
2. VFS	
2.1 Κόμβοι -δ (inodes)	11
2.2 Αρχεία χαρακτήρα	12
2.3 Αρχεία μπλοκ	12
2.4 Αρχεία απεικονιζόμενα στην μνήμη	12
2.5 Προσθήκη διαφορετικών συστημάτων αρχείων	13
2.6 Προσάρτηση δίσκων	13
2.7 Τα συστήματα αρχείων στο Linux	14
3. Τα συστήματα αρχείων FAT	
3.1 Το σύστημα αρχείων FAT16	18
3.2 Το σύστημα αρχείων FAT32	20
3.3 Το σύστημα αρχείων VFAT	23
3.4 Το σύστημα αρχείων NTFS	24
3.5 Κρυπτογραφία στο NTFS	25
3.6 Μεταδεδομένα στο NTFS	27
4. Το σύστημα αρχείων EXT2	
4.1 Ιστορική παρουσίαση	31
4.2 Βασικές αρχές υλοποίησης	33
4.3 Σύγκριση με FAT	33
4.4 Εργαλεία διαχείρισης του EXT2	34
4.5 Κρυπτογραφία στο EXT2	35
4.6 Ασφάλεια στο EXT2	36
4.7 Συμπίεση αρχείων	37
4.8 Μειονεκτήματα του EXT2	38
4.9 Επιδόσεις του EXT2	38
5. Το σύστημα αρχείων EXT3	
5.1 Ιστορική Παρουσίαση	43
5.2 Βασικές αρχές υλοποίησης	43
5.3 Επιδόσεις EXT3	44
5.4 Εργαλεία διαχείρισης EXT3	44
5.5 Σύγκριση με EXT2	45
5.6 Από το EXT2 στο EXT3 και αντίστροφα	47
6. Το σύστημα αρχείων REISERFS	
6.1 Ιστορική Αναδρομή	51

6.2	Εργαλεία διαχείρισεως	52
6.3	Επιδόσεις	52
7.	Το σύστημα αρχείων XFS	
7.1	Ιστορική Αναδρομή	57
7.2	Εργαλεία διαχείρισεως	57
7.3	Επιδόσεις	58
8.	Σύγκριση των journaling filesystems	
8.1	Ταχύτητα	60
8.2	Ασφάλεια	61
8.3	Συμβατότητα	62
8.4	Συμπίεση	62
8.5	Ανάκτηση δεδομένων	63
8.6	Μέγεθος κατατμήσεων	64
9.	Συμπεράσματα – Προτάσεις	67
10.	Βιβλιογραφία	68

ΔΗΛΩΣΗ ΠΕΡΙ ΛΟΓΟΚΛΟΠΗΣ

Όλες οι προτάσεις οι οποίες παρουσιάζονται σε αυτό το κείμενο και οι οποίες ανήκουν σε άλλον αναγνωρίζονται από τα εισαγωγικά και υπάρχει η σαφής δήλωση του συγγραφέα. Τα υπόλοιπα αναγραφόμενα είναι επινόηση του γράφοντος ο οποίος φέρει και την καθολική ευθύνη για αυτό το κείμενο και δηλώνω υπεύθυνα ότι δεν υπάρχει λογοκλοπή σε αυτό το κείμενο.

Όνοματεπώνυμο:

Υπογραφή:

Ημερομηνία:

ΚΕΦΑΛΑΙΟ 1 Βασικές Έννοιες

Αποθηκευτικά Μέσα

Ορισμός συστήματος αρχείων

Τρόποι προσπελάσεως

Τρόποι υλοποίησης συστημάτων αρχείων

Μεταδεδομένα συστημάτων αρχείων

Περίληψη

Στο κεφάλαιο αυτό περιγράφονται τα είδη των αποθηκευτικών μέσων και πιο συγκεκριμένα τα μαγνητικά και αποθηκευτικά μέσα, τα είδη προσπελάσεως των αρχείων, τους τρόπους υλοποίησης συστημάτων αρχείων και τα διάφορα μεταδεδομένα που καταχωρούν τα συστήματα αρχείων.

1.1 Αποθηκευτικά Μέσα

Τα μαγνητικά μέσα αποθήκευσης υπάρχουν εδώ και αρκετά χρόνια, και έχουν χρησιμοποιηθεί ευρέως σε κάθε είδους εφαρμογή. Παρόλα αυτά, το κόστος ανά MB καθώς και η μειωμένη διάρκεια ζωής τους, έκανε φανερή από πολύ νωρίς την ανάγκη για εύρεση καλύτερων λύσεων. Αυτές τις ανάγκες ικανοποιούν τα οπτικά μέσα αποθήκευσης, τα οποία φαίνονται να κυριαρχούν στον χώρο των πολυμέσων σήμερα. Αυτό δεν σημαίνει ότι η χρήση τους είναι αποκλειστική. Το κύριο πρόβλημα τους είναι η ταχύτητα. Απαιτητικές εφαρμογές, όπως η διανομή video, δεν μπορούν να υλοποιηθούν με οπτικά μέσα. Γενικά, όχι μόνο σε εφαρμογές πολυμέσων, το πρόβλημα της επιλογής αποθηκευτικού μέσου είναι πολυσύνθετο. Κάποιος που πρόκειται να κάνει μια τέτοια επιλογή πρέπει να λάβει υπόψη του τους ακόλουθους παράγοντες:

- την ποσότητα που θα αποθηκευτεί και τον απαιτούμενο χρόνο προσπέλασης των δεδομένων
- το είδος της πληροφορίας που πρόκειται να αποθηκευτεί: κείμενο, εικόνες, ήχος, video
- τη μεταβλητότητα της πληροφορίας, τους ρυθμούς με τους οποίους λαμβάνεται και αλλάζει
- ασφάλεια της πληροφορίας

Τα αποθηκευτικά μέσα χωρίζονται σε:

✓ Μαγνητικά αποθηκευτικά μέσα

Τα μαγνητικά αποθηκευτικά μέσα είναι κατάλληλα για δεδομένα που απαιτούν αλλαγές και προσπελάσεις. Χρησιμοποιούνται συνήθως κατά την επεξεργασία των δεδομένων, ενώ για την αρχειοθέτηση προτιμούνται τα οπτικά. Σε εφαρμογές, όπως το video-on-demand, κατά τις οποίες μεγάλες ποσότητες πληροφορίας που μεταβάλλονται με το χρόνο πρέπει να μεταδοθούν γρήγορα, τα μαγνητικά μέσα είναι η μόνη επιλογή.

Μαγνητικά μέσα είναι ο **Οδηγός Δισκέτας**, ο **Σκληρός Δίσκος** και οι **Μαγνητικές Ταινίες (Tapes)**. Ο Οδηγός Δισκέτας χρησιμεύει στην καταχώρηση στοιχείων σε δισκέτες, ενώ ο Σκληρός Δίσκος έχει σαν κύριο χαρακτηριστικό την αποθήκευση μεγάλου όγκου δεδομένων, με υψηλές ταχύτητες ανάκτησή τους.[1]

Όσον αφορά τώρα τις Μαγνητικές Ταινίες χρησιμοποιούνται για την αποθήκευση μεγάλης ποσότητας πληροφορίας και συνήθως για την δημιουργία αντιγράφων ασφαλείας (backup).[2]

✓ Οπτικά αποθηκευτικά μέσα

Στα οπτικά αποθηκευτικά μέσα ανήκουν το CD (Compact Disk), CD-ROM (Compact Disk-Read Only Memory), CD-ROM XA(ψηφιακό οπτικό μέσο αποθήκευσης), CD-I(ψηφιακό οπτικό μέσο αποθήκευσης), WORM(Write Once Read Many, CD Recordable), CD-DA(Compact Disk-Digital Audio), CD-R(Compact Disk Recordable), Photo CD, DVD(Digital Video Disk). Το CD είναι κατάλληλο για μουσική και για αποθήκευση ψηφιακής πληροφορίας. Το CD-ROM είναι κατάλληλο για χρήση σε υπολογιστές. Το CD-ROM XA και το CD-I υποστηρίζουν μίξη κειμένου, εικόνας, ήχου και video. Το WORM είναι κατάλληλο για ανάγνωση και για εγγραφή. Το CD-DA κυριαρχεί στο χώρο της αναπαραγωγής μουσικής. Το CD-R είναι κατάλληλο για εγγραφή CDs. Το Photo CD χρησιμεύει στην αποθήκευση εικόνων που προέρχονται από scanning φιλμ. Το DVD χρησιμεύει στην αναπαραγωγή Video. Στον πίνακα 1.1 βλέπουμε διάφορα είδη δίσκων CD καθώς επίσης και τις δυνατότητές τους.

Δυνατότητα	CD-DA	CD-ROM	CD-i	CD-ROM XA	CD-R	Photo CD	Video CD
Ήχος (PCM)	✓	✓	✓	✓	(✓)	-	-
Δεδομένα εφαρμογής	-	✓	✓	✓	(✓)	-	-
Ήχος (ADPCM)	-	-	✓	✓	(✓)	-	-
Εικόνα (RGB)	-	-	✓	✓	(✓)	-	-
Εικόνα (φωτογραφίες YCC)	-	-	-	-	(✓)	✓	-
Video(MPEG-1)	-	-	(✓)	-	(✓)	-	✓
ISO 9660	-	(✓)	-	(✓)	(✓)	✓	-
Multisession	-	(✓)	-	(✓)	(✓)	✓	-
Οδηγοί που το υποστηρίζουν	CD-DA CD-ROM CD-I CD-ROM XA Photo CD	CD-ROM CD-ROM XA	CD-I	CD-ROM CD-I CD-ROM XA	Εξαρτάται από τον τρόπο εγγραφής	CD-I CD-ROM XA Photo CD	CD-DA CD-I CD-ROM XA Video CD

	Video CD						
--	----------	--	--	--	--	--	--

Πίνακας 1.1: Είδη δίσκων CD

Οι παρενθέσεις υποδεικνύουν πιθανή δυνατότητα. Για παράδειγμα, οι δίσκοι CD-ROM ακολουθούν το ISO 9660 file format(τρόπος αποθήκευσης των αρχείων πάνω στο δίσκο).[3]

1.2 Ορισμός συστήματος αρχείων

Το μέρος του λειτουργικού συστήματος που ασχολείται με τα αρχεία είναι γνωστό ως σύστημα αρχείων. Ένα σύστημα αρχείων αποτελεί τη μέθοδο ονομασίας, αποθήκευσης και οργάνωσης των αρχείων στον υπολογιστή μας. Το σύστημα αρχείων μπορεί να βελτιώσει την ασφάλεια επιτρέποντας μας να κρυπτογραφούμε αρχεία και φακέλους και να περιορίζουμε την πρόσβαση σε αυτά.[4]

1.3 Τρόποι προσπέλασης

Τα πρώτα λειτουργικά συστήματα είχαν μόνο έναν τρόπο προσπέλασης, τη **σειριακή προσπέλαση**(sequential access). Στα συστήματα αυτά μια διεργασία μπορούσε να διαβάσει όλα τα bytes ή τις εγγραφές ενός αρχείου στη σειρά, ξεκινώντας από την αρχή, αλλά δε μπορούσε να παραλείπει μερικά ή να τα διαβάζει εκτός σειράς. Όταν εμφανίστηκαν οι δίσκοι για αποθήκευση αρχείων, τα Bytes και οι εγγραφές ενός αρχείου μπορούσαν να διαβάζονται χωρίς συγκεκριμένη σειρά, ή να προσπελούνται εγγραφές βάσει του κλειδιού τους και όχι βάσει τη θέση τους. Τα αρχεία των οποίων τα bytes και οι εγγραφές μπορούν να διαβαστούν με οποιαδήποτε σειρά καλούνται **αρχεία τυχαίας προσπέλασης**.

Τα αρχεία τυχαίας προσπέλασης είναι απαραίτητα για πολλές εφαρμογές, όπως π.χ για τις βάσεις δεδομένων. Αν ένας πελάτης μιας αεροπορικής εταιρίας επικοινωνήσει για να κλείσει μια θέση σε συγκεκριμένη πτήση, το πρόγραμμα κρατήσεων πρέπει να μπορεί να προσπελάσει την εγγραφή της πτήσης χωρίς να χρειάζεται να διαβάσει πρώτα όλες τις εγγραφές χιλιάδων άλλων πτήσεων.

Σε παλιότερα λειτουργικά συστήματα, σε μεγάλους υπολογιστές τα αρχεία διαχωρίζονται κατά τη δημιουργία τους σε σειριακά και σε αρχεία τυχαίας προσπέλασης. Τα σύγχρονα λειτουργικά συστήματα δε χρειάζεται να κάνουν τέτοιου είδους διαχωρισμούς, καθώς όλα τα αρχεία θεωρούνται αυτόματα τυχαίας προσπέλασης.[5]

1.4 Τρόποι υλοποίησης συστημάτων αρχείων

Το βασικό ζήτημα κατά την υλοποίηση της αποθήκευσης των αρχείων είναι να μπορούμε να γνωρίζουμε κάθε στιγμή ποια μπλοκ¹ του δίσκου ανήκουν σε ποιο αρχείο. Σε διαφορετικά συστήματα χρησιμοποιούνται ποικίλες μέθοδοι. Κάποιες από αυτές τις μεθόδους είναι οι παρακάτω:

- οΣυνεχής κατανομή
- οΚατανομή συνδεδεμένης λίστας
- οΚατανομή συνδεδεμένης λίστας με χρήση δείκτη
- οΚόμβοι-δ

Σύμφωνα με τη **συνεχή κατανομή** το πιο απλό σχήμα κατανομής είναι να αποθηκεύσουμε κάθε αρχείο σαν ένα συνεχόμενο μπλοκ δεδομένων πάνω στο δίσκο. Έτσι σε ένα δίσκο με μπλοκ του 1k ένα αρχείο των 50k θα κατανέμεται σε 50 συνεχή μπλοκ. Αυτό το σχήμα έχει δύο σημαντικά πλεονεκτήματα. Πρώτον είναι απλό στην υλοποίηση αφού το να γνωρίζουμε που βρίσκονται τα μπλοκ του αρχείου περιορίζεται στο να θυμόμαστε έναν και μόνο αριθμό: τη διεύθυνση στο δίσκο του πρώτου μπλοκ. Δεύτερον, η απόδοση είναι πολύ καλή γιατί όλο το αρχείο μπορεί να διαβαστεί από το δίσκο με μια και μόνη ενέργεια. Καμιά άλλη μέθοδος κατανομής δε μπορεί να συναγωνιστεί τη μέθοδο συνεχούς κατανομής.

Η συνεχής κατανομή έχει όμως και δύο σημαντικά μειονεκτήματα: Πρώτον, δεν είναι ευέλικτη, εκτός αν είναι γνωστό το μέγιστο μέγεθος του αρχείου την ώρα που αυτό δημιουργείται. Το δεύτερο μειονέκτημα είναι ο κατακερματισμός του δίσκου².

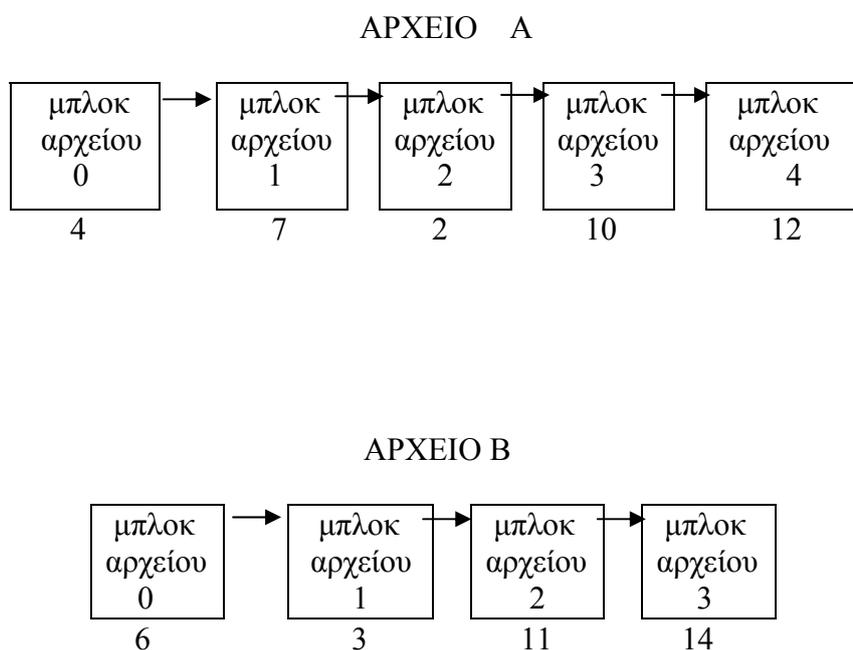
¹ Οι πληροφορίες που αποθηκεύονται σε έναν τομέα αποτελούν το block. Το μέγεθος του Block μπορεί να είναι 128, 256, 512, 1024 bytes. Τα μπλοκ είναι δυνάμεις του 2 και χρησιμοποιούνται για ταχύτερη προσπέλαση.

² Όπως αποθηκεύουμε αρχεία στον δίσκο, τα αρχεία αποθηκεύονται σαν μια συνδεδεμένη λίστα από συστοιχίες. Αν διαγράψουμε ένα αρχείο, δημιουργείται ένα κενό ανάμεσα στις συστοιχίες. Την επόμενη φορά που θα αποθηκεύσουμε ένα αρχείο, το πρόγραμμα θα προσπαθήσει να χωρέσει το αρχείο στο κενό. Αν το αρχείο είναι πολύ μεγάλο και δεν χωρά στο κενό, τότε το πρόγραμμα βάζει όσο περισσότερο από το αρχείο μπορεί εκεί μέσα. Το υπόλοιπο αρχείο αποθηκεύεται στην επόμενη συνδεδεμένη λίστα από συστοιχίες. Στην περίπτωση αυτή, το αρχείο λέγεται **κατακερματισμένο**.

Σύμφωνα με τη **κατανομή συνδεδεμένης λίστας** τα αρχεία πρέπει να τα διατηρούμε ως μια συνδεδεμένη λίστα από μπλοκ δίσκου όπως φαίνεται στο σχήμα 1.1. Το πρώτο τμήμα του κάθε μπλοκ χρησιμοποιείται ως δείκτης στο επόμενο. Το υπόλοιπο του μπλοκ είναι δεδομένα.

Αντίθετα με το τι συμβαίνει στη συνεχή κατανομή με τη μέθοδο αυτή μπορεί να χρησιμοποιηθεί κάθε μπλοκ στο δίσκο. Δεν υπάρχει απώλεια χώρου εξαιτίας του κατακερματισμού του δίσκου. Επίσης, είναι αρκετό να αποθηκεύει κανείς στην καταχώρηση καταλόγου απλώς τις διευθύνσεις των πρώτων μπλοκ. Το υπόλοιπο αρχείο μπορεί να βρεθεί ψάχνοντας από εκεί.

Από την άλλη πλευρά αν και είναι αρκετά απλό να διαβαστεί ένα αρχείο σειριακά, η τυχαία προσπέλαση είναι πάρα πολύ αργή. Επίσης, η ποσότητα των δεδομένων που αποθηκεύονται στο μπλοκ δεν αντιστοιχεί πλέον σε μία δύναμη του δύο αφού και ο δείκτης καταλαμβάνει ορισμένα Bytes. Αυτό όμως δεν είναι ικανοποιητικό καθώς υπάρχουν προγράμματα που γράφουν και διαβάζουν σε μπλοκ με μέγεθος δυνάμεων του δυο.



Σχήμα 1.1: Συνδεδεμένη λίστα από μπλοκ δίσκου

Σύμφωνα με τη **κατανομή συνδεδεμένης λίστας με χρήση δείκτη** και τα δύο μειονεκτήματα της κατανομής με συνδεδεμένη λίστα μπορούν να απαλειφθούν παίρνοντας

τη λέξη με το δείκτη από κάθε μπλοκ και τοποθετώντας τη σε έναν πίνακα ή ευρετήριο στη μνήμη. Το σχήμα 1.2 δείχνει πως θα έμοιαζε ο πίνακας αυτός σύμφωνα με το παράδειγμα του παραπάνω σχήματος. Και στα δύο σχήματα έχουμε δύο αρχεία. Το αρχείο A χρησιμοποιεί τα μπλοκ του δίσκου 4,7,2,10 και 12 με αυτή τη σειρά ενώ το αρχείο B χρησιμοποιεί τα μπλοκ του δίσκου 6,3,11 και 14 με αυτή τη σειρά. Χρησιμοποιώντας τον παρακάτω πίνακα μπορούμε να ξεκινήσουμε από το μπλοκ 4 και να ακολουθήσουμε την αλυσίδα μέχρι το τέλος. Το ίδιο μπορεί να συμβεί αν ξεκινήσουμε από το μπλοκ 6.

0	
1	
2	10
3	11
4	7
5	
6	3
7	2
8	
9	
10	12
11	14
12	0
13	
14	0

← Το αρχείο A ξεκινά εδώ

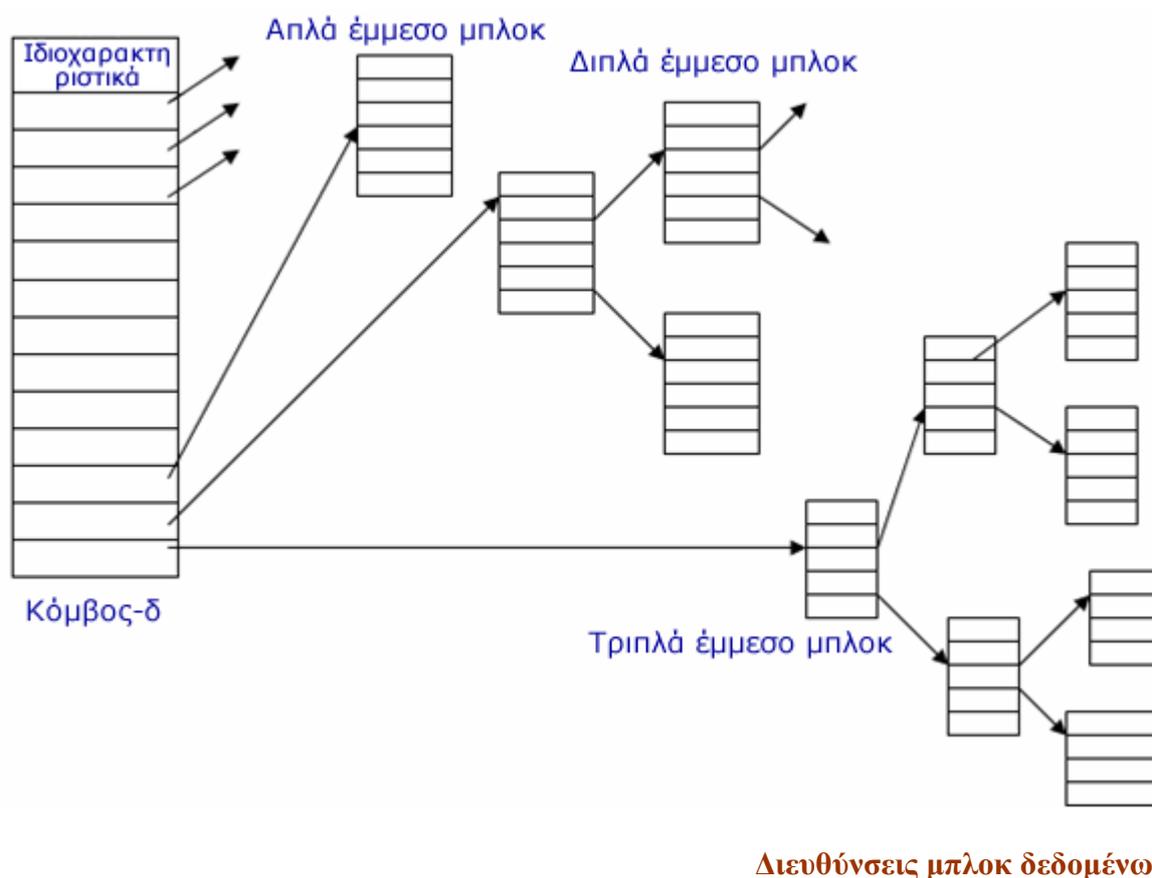
← Το αρχείο B ξεκινά εδώ

Σχήμα 1.2: Κατανομή συνδεδεμένης λίστας με χρήση δείκτη

Χρησιμοποιώντας αυτή την οργάνωση όλο το μπλοκ είναι ελεύθερο για δεδομένα. Επιπλέον, η τυχαία προσπέλαση είναι πολύ πιο εύκολη. Αν και πρέπει πάντα να ακολουθήσουμε την αλυσίδα για να βρούμε μια οποιαδήποτε μετατόπιση μέσα στο αρχείο, η αλυσίδα βρίσκεται πάντα ολόκληρη στη μνήμη και έτσι μπορούμε να την ακολουθήσουμε χωρίς να χρειάζονται αναφορές στο δίσκο. Όπως και στην προηγούμενη μέθοδο είναι αρκετό για την καταχώρηση στον κατάλογο να αποθηκεύεται ένας μόνο ακέραιος(ο αριθμός του πρώτου μπλοκ),όπως επίσης είναι δυνατόν να βρεθούν όλα τα μπλοκ, χωρίς να έχει σημασία το μέγεθος του αρχείου. Το κύριο μειονέκτημα αυτής της μεθόδου είναι ότι για να λειτουργήσει πρέπει να διατηρείται στη μνήμη συνεχώς όλος ο πίνακας. Με ένα μεγάλο δίσκο, έστω 500.000 μπλοκ του 1K, ο πίνακας θα έχει 500.000 καταχωρήσεις, κάθε μια από τις οποίες θα πρέπει να καταλαμβάνει τουλάχιστον 3 Bytes. Για ταχύτητα στην αναζήτηση χρειάζονται 4 Bytes. Έτσι ο πίνακας θα καταλαμβάνει 1.5

με 2 megabytes συνεχώς ανάλογα με το αν το σύστημα έχει βελτιστοποίηση χώρου ή χρόνου.

Η τέταρτη και τελευταία μέθοδος που χρησιμοποιείται για να γνωρίζουμε ποια μπλοκ ανήκουν σε ποιο αρχείο συνίσταται στο να συσχετίζουμε κάθε αρχείο με ένα μικρό πίνακα ο οποίος ονομάζεται **κόμβος-δ** και ο οποίος περιέχει τα χαρακτηριστικά και τις διευθύνσεις στο δίσκο των μπλοκ του αρχείου όπως φαίνεται στο σχήμα 1.3.



Σχήμα 1.3: Κόμβος-δ

Οι πρώτες διευθύνσεις στο δίσκο αποθηκεύονται στον ίδιο τον κόμβο-δ έτσι ώστε για μικρά αρχεία όλες οι απαραίτητες πληροφορίες να βρίσκονται ακριβώς στον ίδιο κόμβο-δ τον οποίο φορτώνουμε από το δίσκο στη μνήμη όταν ανοίγουμε το αρχείο. Για λίγο μεγαλύτερα αρχεία, μία από τις διευθύνσεις στον κόμβο-δ είναι η διεύθυνση ενός μπλοκ στο δίσκο το οποίο καλείται απλά έμμεσο μπλοκ. Το μπλοκ αυτό περιέχει συμπληρωματικές διευθύνσεις στο δίσκο. Αν και αυτό δεν είναι αρκετό, μία ακόμα διεύθυνση στον κόμβο-δ περιέχει τη διεύθυνση ενός μπλοκ το οποίο καλείται διπλά έμμεσο μπλοκ και το οποίο περιέχει μία λίστα από απλά έμμεσα μπλοκ. Καθένα από αυτά

τα απλά έμμεσο μπλοκ παρέχουν πρόσβαση σε μερικές εκατοντάδες μπλοκ δεδομένων. Αν και αυτό δεν είναι αρκετό τότε μπορεί να χρησιμοποιηθεί ένα τριπλά έμμεσο μπλοκ.[6]

1.5 Μεταδεδομένα συστημάτων αρχείων

Πέρα από τα πραγματικά δεδομένα σε ένα σύστημα αρχείων υπάρχουν και τα μεταδεδομένα ή metadata τα οποία δίνουν πληροφορίες για τα ίδια τα δεδομένα και χρησιμοποιούνται για την περιγραφή των πληροφοριών και της κατάστασης στην οποία βρίσκονται. [7]

Κάθε σύστημα αρχείων καταχωρεί διαφορετικά μεταδεδομένα όπως για παράδειγμα το όνομα και τις ιδιότητες κάθε αρχείου [8]

Τα περισσότερα συστήματα αρχείων καταχωρούν τα παρακάτω:

- 1.Ιδιοκτήτης -προσδιορίζει τον ιδιοκτήτη και το group του filesystem
- 2.permission bits-Bits δικαιωμάτων
- 3.timestamps- χρόνοι πρόσβασης και τροποποίησης καταχωρούνται για κάθε αντικείμενο Filesystem[9]

ΚΕΦΑΛΑΙΟ 2 VFS

Κόμβοι –δ (inodes)

Αρχεία χαρακτήρα

Αρχεία μπλοκ

Αρχεία απεικονιζόμενα στην μνήμη

Προσθήκη διαφορετικών συστημάτων αρχείων

Προσάρτηση δίσκων

Τα συστήματα αρχείων στο Linux

Περίληψη

Στο κεφάλαιο αυτό περιέχονται πληροφορίες για το VFS, τις ιδιότητες των κόμβων- δ, τα αρχεία χαρακτήρα και τα αρχεία μπλοκ, τον τρόπο πρόσβασης στα αρχεία, τον τρόπο με τον οποίο προστίθενται νέα συστήματα αρχείων και δίσκοι στο σύστημα και τέλος τα διάφορα συστήματα αρχείων που υπάρχουν.

Ο πυρήνας του Linux περιέχει ένα εικονικό σύστημα αρχείων, το λεγόμενο VFS(Virtual File System). Το VFS γνωρίζει τα συστήματα αρχείων που υπάρχουν στο σύστημα και υποστηρίζονται από τον πυρήνα χρησιμοποιώντας έναν πίνακα που δημιουργείται κατά τη διάρκεια της διαμόρφωσης πυρήνων. Κάθε είσοδος σε αυτόν τον πίνακα περιγράφει ένα σύστημα αρχείων: περιέχει το όνομα του συστήματος αρχείων και μια λίστα από λειτουργίες στο σύστημα αρχείων έτσι ώστε να μπορούμε να χρησιμοποιήσουμε τις ίδιες λειτουργίες εισόδου / εξόδου για κάθε σύστημα αρχείων. Αυτές οι λειτουργίες αρχικοποιούν τις μεταβλητές και εισάγουν το σύστημα αρχείων στο VFS. Ένα προσαρτημένο σύστημα αρχείων περιέχει:

- I. Πληροφορίες που είναι κοινές για κάθε σύστημα αρχείων
- II. Πληροφορίες που αφορούν το συγκεκριμένο σύστημα αρχείων [10]

2.1 Κόμβοι-δ (Inodes)

Οι κόμβοι-δ είναι δομές δεδομένων που αποθηκεύουν πληροφορίες σχετικές με τα δεδομένα, τα δικαιώματα και τους δείκτες προς τα υπόλοιπα δεδομένα. Κάθε αρχείο, κατάλογος έχει ένα VFS inode και καθορίζεται με αυτό το Inode στο σύστημα το οποίο βρίσκεται. Τα VFS inode υπάρχουν στη μνήμη του πυρήνα και κρατούνται στη κρυφή μνήμη³ του VFS καθώς είναι χρήσιμα στο σύστημα. Τα VFS inode παρέχουν τις ακόλουθες πληροφορίες:

Αριθμός inode (Inode Number): Αυτό είναι ο αριθμός των inode και είναι μοναδικός για το συγκεκριμένο σύστημα αρχείων.

User ids: Περιγραφείς ιδιοκτησιών

Χρόνος(Times): Χρόνος δημιουργίας, τροποποίησης, εγγραφής

Μέγεθος block(Block size): το μέγεθος του block. Για παράδειγμα 1024 Bytes.

Κλείδωμα(Lock): Αυτό το πεδίο χρησιμοποιείται για το κλείδωμα του VFS inode, όταν για παράδειγμα διαβάζεται από το σύστημα αρχείων.

Dirty: Δείχνει εάν το VFS inode έχει γραφεί και αν ναι σημαίνει ότι δεν μπορούμε να το χρησιμοποιήσουμε. [11]

³ cache=προσωρινή αποθήκευση δεδομένων. Βοηθητική μνήμη όταν δε φτάνει η RAM

2.2-2.3 Αρχεία χαρακτήρα-Αρχεία μπλοκ

Τα αρχεία χαρακτήρων σχετίζονται με την είσοδο και την έξοδο και χρησιμοποιούνται για να μοντελοποιήσουν συσκευές σειριακής εισόδου/ εξόδου όπως τερματικά, εκτυπωτές, δίκτυα. Τα αρχεία μπλοκ χρησιμοποιούνται για να μοντελοποιήσουν τους δίσκους. Τα αρχεία υπάρχουν για να εξομοιώνουμε τις συσκευές εισόδου/ εξόδου έτσι ώστε να συμπεριφέρονται σαν αρχεία. Με αυτό τον τρόπο αυτές μπορούν να διαβαστούν και να γραφούν χρησιμοποιώντας τις ίδιες κλήσεις συστήματος που χρησιμοποιούμε για να διαβάζουμε και να γράφουμε σε αρχεία.

Πιο συγκεκριμένα τα **αρχεία μπλοκ** χρησιμοποιούνται για την μοντελοποίηση συσκευών, που αποτελούνται από ένα σύνολο τυχαία διευθυνσιοδοτημένων μπλοκ όπως οι δίσκοι. Η χαρακτηριστική ιδιότητα ενός αρχείου μπλοκ είναι ότι μπορεί να διευθυνσιοδοτηθεί και να προσπελαστεί ξεχωριστά. Με άλλα λόγια ένα πρόγραμμα μπορεί να ανοίξει ένα αρχείο μπλοκ και να διαβάσει για παράδειγμα το μπλοκ 124 χωρίς να είναι απαραίτητο να διαβάσει πρώτα το μπλοκ 0 έως 123.

Τα **αρχεία χαρακτήρων** χρησιμοποιούνται για την μοντελοποίηση συσκευών που βασική τους λειτουργία είναι η ροή χαρακτήρων παρά η τυχαία προσπέλαση μπλοκ. Τα τερματικά, οι εκτυπωτές, τα ποντίκια και οι διασυνδέσεις δικτύου είναι τυπικά παραδείγματα που χρησιμοποιούν αρχεία χαρακτήρων. Τα αρχεία χαρακτήρων μπορούν να προσπελαστούν μόνο σειριακά.[12]

2.4 Αρχεία απεικονιζόμενα στη μνήμη

Η πρόσβαση σε αρχεία μπορεί να γίνει χωρίς τη χρήση των εντολών read/ write/ seek αλλά απλά με την απευθείας απεικόνισή τους στη μνήμη. Η πρόσβαση στα αρχεία με αυτόν τον τρόπο γίνεται μέσω κλήσεων στο λειτουργικό σύστημα και στη συνέχεια χρησιμοποίησης της αντίστοιχης μνήμης (μέσω δεικτών ή πινάκων). Οι σχετικές κλήσεις είναι:

map⁴: Σύνδεση μιας περιοχής της μνήμης με ένα αρχείο

Unmap⁵: Τερματισμός της αντίστοιχης σύνδεσης

⁴ Δίνει ένα όνομα αρχείου και μια ιδεατή διεύθυνση οπότε το λειτουργικό σύστημα αναγκάζεται να απεικονίσει το αρχείο στο χώρο διεύθυνσεων ξεκινώντας από αυτή την ιδεατή διεύθυνση-

⁵ Απομάκρυνση των αρχείων από το χώρο της μνήμης και σταμάτημα εκτέλεσης του προγράμματος της διεργασίας

Για παράδειγμα έστω ένα αρχείο `f` με μήκος `64K` το οποίο απεικονίζεται στην διεύθυνση `512K`. Τότε οποιαδήποτε εντολή μηχανής που διαβάζει τα περιεχόμενα ενός `Byte` που βρίσκεται στη διεύθυνση `512K` παίρνει το `Byte 0` του αρχείου, κτλ. Όταν η διεργασία σταματήσει, το τροποποιημένο αρχείο παραμένει στο δίσκο, όπως ακριβώς θα γινόταν αν οι αλλαγές είχαν γίνει με μια σειρά από κλήσεις συστήματος `SEEK` και `WRITE`. [13]

2.5 Προσθήκη διαφορετικών συστημάτων αρχείων

Το `VFS` (`Virtual Filesystem`) επιτρέπει την προσάρτηση πολλών διαφορετικών συστημάτων αρχείων ταυτόχρονα. Όσον αφορά τις πρώτες εκδόσεις, όλες οι προσβάσεις των συστημάτων αρχείων βασιζόνταν στο `minix` σύστημα αρχείων. Έτσι και τώρα οι προσβάσεις των συστημάτων αρχείων πρέπει να στηρίζονται σε κάποιο άλλο σύστημα αρχείων. [14]

Για τη προσάρτηση ενός συστήματος αρχείων χρησιμοποιούμε την εντολή `mount`. Σκοπός της εντολής `Mount` είναι να καθιστά κάθε σύστημα αρχείων διαθέσιμο.

2.6 Προσάρτηση δίσκων

Τα διαφορετικά `partitions` των δίσκων μπορούν να περιέχουν τους διαφορετικούς τύπους συστημάτων αρχείων αλλά μόλις προστεθούν, πρέπει να παρουσιάσουν ένα κοινό `filesystem interface`.

Προσάρτηση νέου δίσκου

Δημιουργούμε έναν κατάλογο όπου θα προσαρτηθεί ο νέος δίσκος, για παράδειγμα τον κατάλογο `/new-disk` και προσάρτηση του δίσκου σε αυτόν:

```
mkdir /new-disk
```

```
mount -t ext2 /dev/hdb1 /new-disk [15]
```

2.7 Τα συστήματα αρχείων στο Linux

Ένα από τα χαρακτηριστικά γνωρίσματα του Linux είναι ότι υποστηρίζει πολλά διαφορετικά συστήματα αρχείων. Το Linux υποστηρίζει συστήματα αρχείων όπως το: ext, ext2, ext3, reiserfs, xfs, xia, minix, umsdos, MS-dos, vfat, proc, smb, ncp, iso9660, sysv, hpfs, affs και ufs. Πιο συγκεκριμένα ας τα χωρίσουμε στις ακόλουθες κατηγορίες:

Συστήματα αρχείων βασισμένα σε δίσκους: ext2 (linux native), ufs(BSD), fat(DOS-FS), vfat(WIN95), hpfs(OS/2), minix, Isofs(CDROM), sysv(Sysv UNIX), hfs(Macintosh), affs(Amiga Fast FS), NTFS, adfs

Δικτυακά συστήματα αρχείων: nfs, Coda, AFS, smbfs, ncpfs

Ειδικά συστήματα αρχείων: procfs, umsdos(UNIX IN DOS), userfs(ανακατεύθυνση στο χρήστη)

Forthcoming συστήματα αρχείων: devs(device filesystem), DFS-DCE distribyted FS

Διάφορα συστήματα αρχείων: cfs(encrypt filesystem), cfs(cache filesystem), ftpfs(ftp filesystem), mailfs(mail filesystem), pgfs(postgres versioning filesystem)

Στο Linux, τα συστήματα αρχείων που μπορεί να χρησιμοποιήσει το σύστημα, συνδυάζονται σε μια ενιαία ιεραρχική δομή δέντρου. Το σύστημα αρχείων περιέχει: α)τις πληροφορίες που περιέχονται στα αρχεία του συστήματος αρχείων, όπως για παράδειγμα πληροφορίες προστασίας αρχείων, β) τη δομή του συστήματος αρχείων, γ)πληροφορίες που περιέχονται στους καταλόγους του συστήματος αρχείων, όπως για παράδειγμα τα αρχεία που περιέχουν, δ)ασφάλεια αυτών των πληροφοριών .

Το εικονικό σύστημα αρχείων(VFS) του Linux εφαρμόζεται έτσι ώστε:

-η πρόσβαση στα αρχεία να είναι όσο το δυνατόν πιο γρήγορη

-να είμαστε σίγουροι ότι τα αρχεία και τα στοιχεία τους διατηρούνται σωστά. [16]

ΚΕΦΑΛΑΙΟ 3 Τα συστήματα αρχείων FAT

Το σύστημα αρχείων FAT16

Το σύστημα αρχείων FAT32

Το σύστημα αρχείων VFAT

Το σύστημα αρχείων NTFS

Κρυπτογραφία στο NTFS

Μεταδεδομένα στο NTFS

Περίληψη

Στο κεφάλαιο αυτό δίνονται πληροφορίες για το FAT, για τα χαρακτηριστικά του FAT16 και του FAT32. Επίσης, αναφέρεται στο VFAT και στα χαρακτηριστικά γνωρίσματα, στις ιδιότητες, στις αρχές, στις άδειες και στα μεταδεδομένα του NTFS.

Το FAT προέρχεται από τα αρχικά File Allocation Table. Είναι πίνακες που αποθηκεύονται στο σκληρό δίσκο, στον οποίο υπάρχει καταχωρημένη η ακριβής θέση κάθε αρχείου. Το σύστημα FAT αρχικά χρησιμοποιήθηκε για την αποθήκευση δεδομένων σε δισκέτες και ήταν γνωστό ως FAT 12. Αργότερα προέκυψε το FAT 16 που χρησιμοποιείται ακόμα και σήμερα σε σκληρούς δίσκους μικρής χωρητικότητας, κάτω από 2GB. Στο FAT 16 κάθε cluster⁶ του σκληρού δίσκου έχει μέγεθος 8 Kb. Αυτό σημαίνει ότι ακόμα και αν το μέγεθος ενός αρχείου είναι ας πούμε 2 Kb, αυτό θα καταλάβει ένα ολόκληρο cluster δηλαδή 8 Kb. Ένα αρχείο με μέγεθος 16,4 Kb θα καταλάβει 3 ολόκληρα clusters, συνολικού μεγέθους 25 Kb περίπου. Αυτό οδηγεί σε απώλεια σημαντικής χωρητικότητας του σκληρού δίσκου, αφού συμβαίνει με όλα τα αρχεία που είναι αποθηκευμένα σε αυτόν.

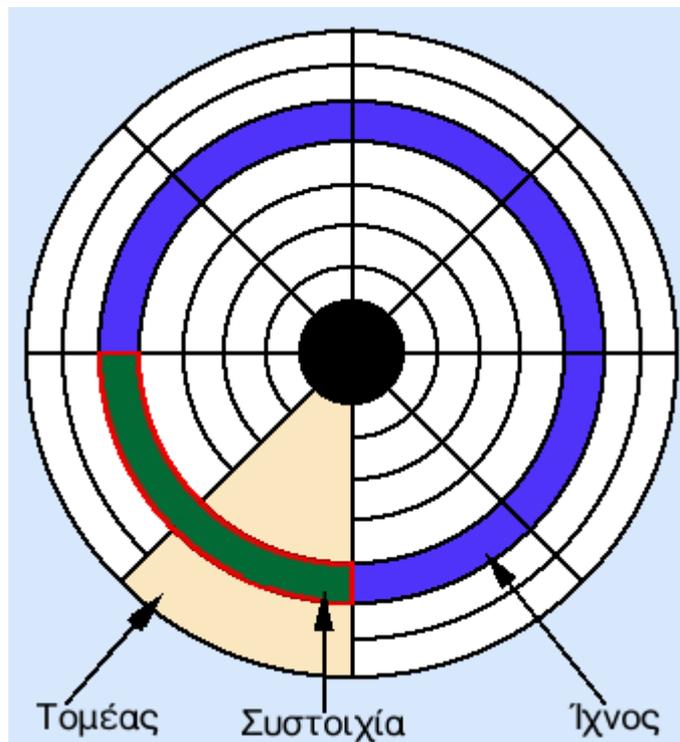
Το σύστημα FAT 32 είναι κατάλληλο για σκληρούς δίσκους άνω των 2GB. Στο σύστημα αυτό κάθε cluster έχει μέγεθος 4 Kb, το μισό μέγεθος δηλαδή του FAT 16. Έτσι το αρχείο του προηγούμενου παραδείγματος που είχε μέγεθος 2Kb πιάνει αυτή τη φορά 4 Kb και χάνονται μόνο 2 Kb αντί για 6 Kb που θα χάνονταν με το FAT 16. Δηλ. αξιοποιείται καλύτερα ο σκληρός δίσκος. Το σύστημα αρχείων NTFS, έχει ακόμα μικρότερο μέγεθος cluster, που είναι 2 Kb. Όμως το NTFS είναι κατάλληλο μόνο για Windows NT και 2000 και δεν αναγνωρίζεται από τα Windows 95/98 και τα Millenium. Για να αλλάξουμε το σύστημα αρχείων ενός σκληρού δίσκου και να δημιουργήσουμε ένα καινούριο σύστημα αρχείων πρέπει να κάνουμε format οπότε θα χαθούν όλα τα περιεχόμενά του. [17]

Για να γίνει κατανοητό τι ακριβώς κάνει ο πίνακας κατανομής αρχείων, πρέπει πρώτα να γίνει κατανοητό πως είναι κατανεμημένος ο χώρος του σκληρού δίσκου. Για λόγους απόδοσης ένα αρχείο δεν αποτελείται από ξεχωριστούς **τομείς**⁷. Ο λόγος είναι ότι θα προκαλούνταν μεγάλη σπατάλη(χρόνου και χώρου) αν ένα αρχείο αποθηκεύονταν σε διαφορετικούς τομείς, όταν αυτά είναι τόσο μικρά (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. Κάθε συστοιχία

⁶ Η τομή του ίχνους με δύο ή περισσότερους τομείς ονομάζεται **συστοιχία (cluster)** ή εναλλακτικά μονάδες κατανομής (allocation units) και είναι η μικρότερη ομάδα δεδομένων που μπορούμε να προσπελάσουμε στο δίσκο. 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

⁷ Κάθε ίχνος έχει χωριστεί σε μικρότερες μονάδες που ονομάζονται **τομείς (sectors)**. Κάθε τομέας μπορεί να χωρέσει 512 bytes δεδομένων, συν μερικά επιπρόσθετα bytes τα οποία χρησιμοποιούνται για εσωτερικό έλεγχο του δίσκου και για αναγνώριση και διόρθωση λαθών

έχει μια καταχώρηση μέσα στο FAT η οποία λέει στο λειτουργικό σύστημα ποια κομμάτια του δίσκου είναι ελεύθερα και ποια χρησιμοποιούνται από αρχεία.



Σχήμα 3.1: Οργάνωση επιφάνειας δίσκου: Ίχνη / τομείς / συστοιχίες

Οι πίνακες κατανομής αρχείων(FAT) είναι τοποθετημένοι στην περιοχή του δίσκου αμέσως μετά τον **τομέα εκκίνησης**⁸ του διαμερίσματος.

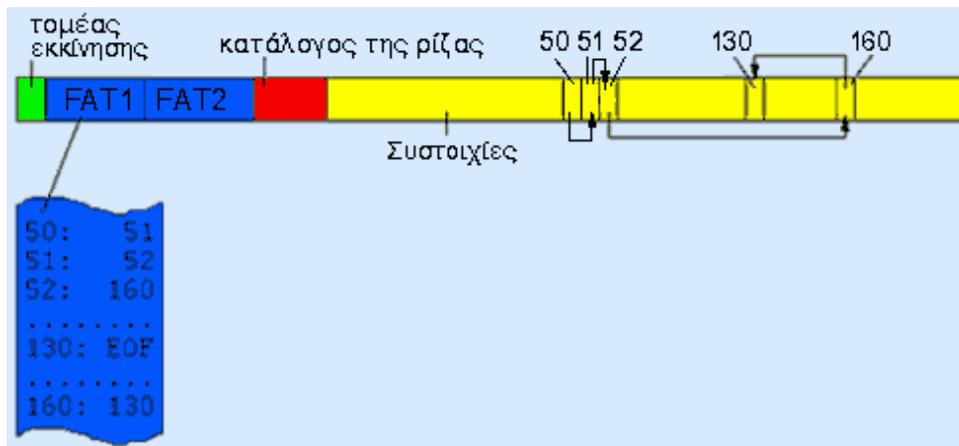
Κάθε διαμέρισμα περιέχει δύο ίδια αντίγραφα του FAT. Το δεύτερο αντίγραφο χρησιμοποιείται ως εφεδρικό, σε περίπτωση που συμβεί κάποια ζημιά στο πρώτο. Μια ζημιά στο FAT μπορεί να προκαλέσει απώλεια δεδομένων. Το πρόβλημα με το εφεδρικό αντίγραφο είναι ότι βρίσκεται ακριβώς δίπλα στο πρώτο, έτσι ώστε αν προκληθεί κάποια

⁸ Κάθε DOS διαμέρισμα (επίσης αποκαλείται και λογικός δίσκος) έχει το δικό του τομέα εκκίνησης διαμερίσματος. Είναι διαφορετικός από το βασικό τομέα εκκίνησης, ο οποίος ελέγχει ολόκληρο το δίσκο, αλλά είναι όμοιος με τις αρχές του. Κάθε τομέας εκκίνησης διαμερίσματος περιλαμβάνει τα παρακάτω:

Παράμετροι Block Δίσκου (Disk Parameter Block): Ορισμένες φορές αποκαλείται και media parameter block. Πρόκειται για έναν πίνακα δεδομένων που περιλαμβάνει ειδικές πληροφορίες, σχετικές με το διαμέρισμα, όπως οι προδιαγραφές του (μέγεθος, αριθμός των τομέων που περιλαμβάνει κλπ), την ετικέτα δίσκου (label name) και άλλες παρόμοιες πληροφορίες.

Κώδικας Εκκίνησης Διαμερίσματος (Volume Boot Code): Αυτός ο κώδικας χρησιμοποιείται για να ξεκινήσει να φορτώνει το λειτουργικό σύστημα.

ζημιά, από κατεστραμμένους τομείς για παράδειγμα στο πρώτο FAT, είναι πολύ πιθανό το πρόβλημα να επεκταθεί και στο εφεδρικό.



Σχήμα 3.2: Πίνακας Κατανομής Αρχείων(FAT) [18]

3.1 Το σύστημα αρχείων FAT16

Το FAT χρησιμοποιείται από το DOS και τα Windows 95/98/NT/2000/XP. Όταν κάποιος αναφέρεται σε «FAT» έκδοση γενικά, αναφέρεται συνήθως σε FAT-16. Στο FAT-16 το μέγιστο μέγεθος των clusters που χρησιμοποιείται είναι 65,526. Κάθε μια από τις εισόδους στο FAT αντιστοιχεί σε μία συστοιχία στο δίσκο.

Στον πίνακα 3.1 βλέπουμε ιδιότητες του FAT-16:

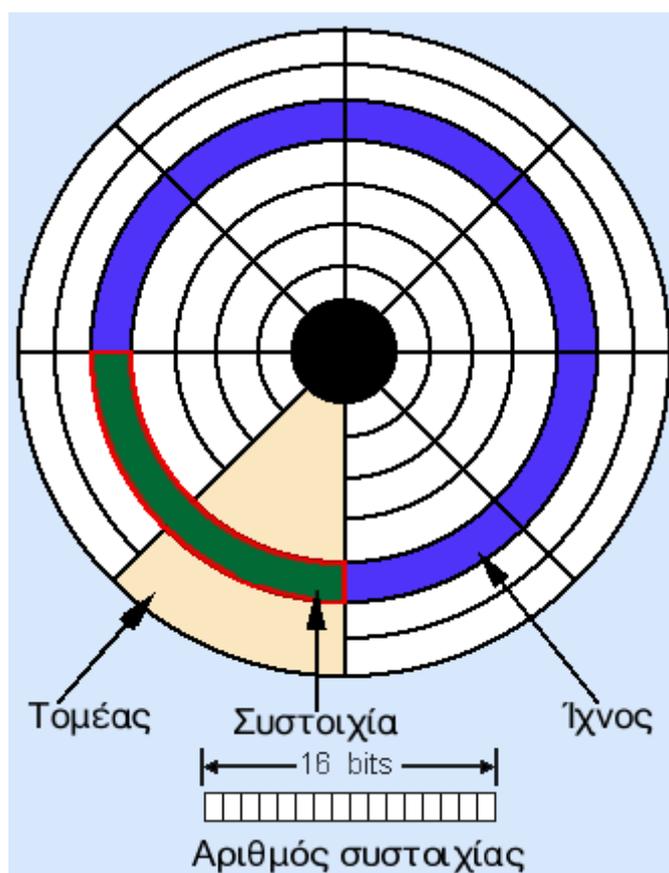
ΙΔΙΟΤΗΤΑ	FAT-16
Χρησιμοποιείται από:	Μικρούς σκληρούς δίσκου των 2 GB
Μέγεθος κάθε εισόδου FAT:	16 bits
Μέγιστος αριθμός καταχωρήσεων:	65526
Μέγεθος Cluster που χρησιμοποιείται:	2 KB με 32 KB

Πίνακας 3.1: Ιδιότητες FAT-16

Στον πίνακα 3.2 βλέπουμε το FAT-16 όσον αφορά τη διαμέριση και τον ανεκμετάλλετο χώρο :

Μέγεθος διαμέρισης	Μέγεθος Cluster	Αριθμός αρχείων	Άχρηστος χώρος
Λιγότερος από 128 MB	2K	2,000	2MB
128 με 256 MB	4K	4,000	8MB
256 με 512 MB	8K	8,000	32MB
512MB με 1GB	16K	16,000	128MB
1GB με 2 GB	32K	32,000	512MB

Πίνακας 3.2: Διαμέριση και ανεκμετάλλετος χώρος FAT-16[19]



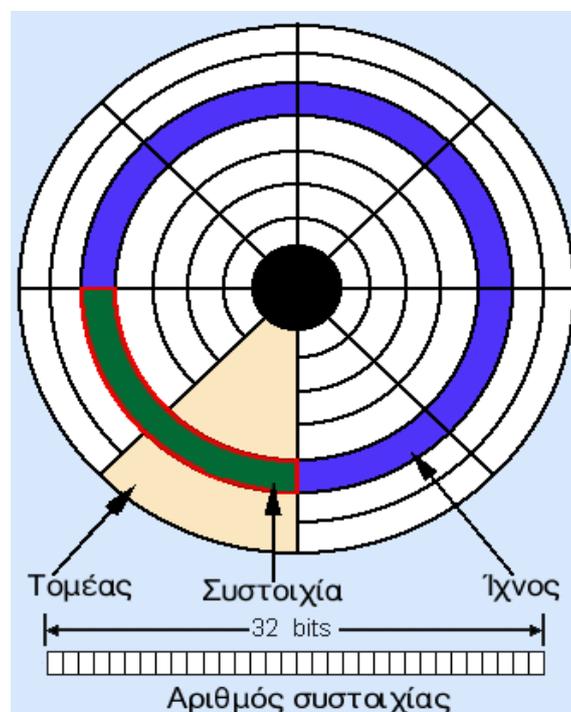
Σχήμα 3.3: FAT (ή FAT16) σύστημα αρχείων [20]

3.2 Το σύστημα αρχείων FAT32

Το FAT32 είναι μια βελτίωση του 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 FAT16. Το FAT32 παρουσιάστηκε στα Windows 95. 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 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. Επιπρόσθετα, το FAT32 εξοικονομεί χώρο, γιατί χρησιμοποιεί συστοιχίες πολύ μικρότερες σε μέγεθος από αυτές του FAT16. Η συνέπεια αυτού όμως 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 FAT και οι περιγραφές των δομών αρχείων του FAT ταιριάζουν και για το FAT32. [21]



Σχήμα 3.4: FAT32 σύστημα αρχείων αρχείων

Μέγεθος συστοιχίας και μέγεθος πίνακα καταχώρησης αρχείων

Ο κανόνας στην επιλογή του μεγέθους της συστοιχίας λέει ότι "το μικρότερο είναι καλύτερο". Όσο τα διαμερίσματα που χρησιμοποιούν FAT16 γίνονται μεγαλύτερα και ο χώρος που δαπανάται γίνεται όλο και μεγαλύτερος, η ανάγκη για τη χρησιμοποίηση του FAT32, για να μειωθεί το μέγεθος της συστοιχίας, γίνεται επιτακτική. Καθώς το FAT32 επιτρέπει τη χρήση μεγαλύτερων σκληρών δίσκων και σπουδαία μείωση στο μέγεθος της συστοιχίας, υπάρχει μια ουσιαστική αιτία χρήσης του FAT32.

Ας θεωρήσουμε ένα διαμέρισμα λίγο μικρότερο από 2GB, το μεγαλύτερο μέγεθος που μπορεί να υποστηρίξει το FAT16. Αν αυτό το διαμέρισμα χρησιμοποιεί FAT16, θα χρειαστεί ένα πίνακα καταχώρησης αρχείων με $2^{16}-1$ (65525) συστοιχίες μέσα σε αυτόν, με την κάθε συστοιχία να καταλαμβάνει 32 KB στο δίσκο. Το μεγάλο μέγεθος των συστοιχιών θα προκαλέσει ένα μεγάλο ποσό χαμένου χώρου στο δίσκο, στις περισσότερες περιπτώσεις. Συνεπώς, συστήνεται σε αυτό το διαμέρισμα να χρησιμοποιηθεί FAT32, το οποίο θα κατεβάσει το μέγεθος της συστοιχίας από τα 32 στα 4 KB. Αυτό θα μειώσει τον άχρηστο χώρο στο δίσκο, κοντά στο 90% και ενδεχομένως να ελευθερώσει εκατοντάδες MB, προηγούμενου χαμένου χώρου στο δίσκο.

Ωστόσο, υπάρχει και μια άλλη πλευρά αυτού του γεγονότος: η μείωση του μεγέθους της συστοιχίας δε θα υπάρξει χωρίς κάποιο αντίτιμο. Καθώς κάθε συστοιχία είναι μικρότερη, πρέπει να υπάρχουν περισσότερες για να καλύψουν την ίδια ποσότητα του δίσκου. Έτσι αντί για 65525 συστοιχίες, τώρα θα έχουμε 524208. Επιπλέον, οι καταχωρήσεις στον πίνακα καταχώρησης έχουν μήκος 32 bits (4 bytes), πολύ μεγαλύτερο σε σχέση με τις καταχωρήσεις του FAT16 που καταλάμβαναν μόνο δύο bytes. Το τελικό αποτέλεσμα είναι ότι το μέγεθος του πίνακα καταχώρησης αρχείων, είναι 16 φορές μεγαλύτερο για το FAT32 σε σχέση με το FAT16. Ο πίνακας 3.3 συνοψίζει τα παραπάνω:

Τύπος Πίνακα Καταχώρησης	FAT16	FAT32
Μέγεθος συστοιχίας	32KB	4KB
Μέγιστος Αριθμός Καταχωρήσεων του Πίνακα	65525	524208
Μέγεθος του Πίνακα Καταχώρησης	~128KB	~2MB

Πίνακας 3.3: Διαφορές FAT16-FAT32

Αν αυξήσουμε το μέγεθος του διαμερίσματος από 2 GB σε 8 GB, το μέγεθος του πίνακα καταχώρησης θα αυξηθεί από 2 MB σε 8 MB. Έχοντας όμως ένα πίνακα αυξημένο στο μέγεθος, θα επηρεάσει αρνητικά την ταχύτητα του συστήματος.

Ουσιαστικά, κάθε σύστημα χρησιμοποιεί κρυφή μνήμη στο δίσκο (disk caching), για να κρατήσει στη μνήμη δομές του δίσκου που προσπελούνται συχνά, όπως είναι ο πίνακας καταχώρησης αρχείων. Η κρυφή μνήμη του δίσκου χρησιμοποιεί μέρος της κύριας μνήμης για να κρατάει πληροφορίες, έτσι ώστε να κερδίσει χρόνο διαβάζοντάς τες από εκεί, καθώς η μνήμη είναι πολύ πιο γρήγορη από το δίσκο. Όταν ο πίνακας καταχώρησης αρχείων είναι μικρός, όπως ο 128 KB πίνακας που χρησιμοποιείται από το FAT16, ολόκληρος ο πίνακας μπορεί να κρατηθεί εύκολα στη μνήμη και κάθε φορά που χρειαζόμαστε κάτι από αυτόν να μπορούμε να τον προσπελάσουμε αρκετά γρήγορα. Όταν ο πίνακας αυξάνει σε μέγεθος, στα 8 MB για παράδειγμα, το σύστημα είναι αναγκασμένο να διαλέξει ανάμεσα σε δύο κακές εναλλακτικές λύσεις: να χρησιμοποιήσει ένα μεγάλο μέρος της μνήμης για να κρατήσει τον πίνακα ή να μην τον κρατήσει στη μνήμη.

Γι' αυτό το λόγο είναι σημαντικό να περιορίσουμε το μέγεθος του πίνακα καταχώρησης αρχείων σε ένα λογικό μέγεθος. Στην ουσία, στις περισσότερες περιπτώσεις είναι θέμα ισορροπίας μεταξύ των μεγεθών της συστοιχίας και του πίνακα καταχώρησης. Μια καλή λύση είναι η επιλογή του μεγέθους της συστοιχίας να γίνεται από το ίδιο το FAT32. Έτσι, το FAT32 χρησιμοποιεί συστοιχίες μεγέθους 4 KB, μόνο για διαμερίσματα με μέγεθος μέχρι 8 GB και από εκεί και πάνω αυξάνει το μέγεθος των συστοιχιών, όπως δείχνει ο πίνακας 3.4:

Μέγεθος συστοιχίας	«Ελάχιστο» μέγεθος διαμερίσματος	«Μέγιστο» μέγεθος διαμερίσματος
4KB	0.5GB	8GB
8KB	8GB	16GB
16KB	16GB	32GB
32KB	32GB	64GB

Πίνακας 3.4: Συστοιχίες και αντίστοιχα μεγέθη διαμέρισης όσον αφορά το FAT32

Διαμέριση για δίσκους μεγαλύτερους από 2GB

Αν έχουμε ένα σκληρό δίσκο μεγαλύτερο από 2GB σε μέγεθος και δε χρησιμοποιούμε FAT32, πρέπει να χωρίσουμε το δίσκο σε διαμερίσματα, που το καθένα δε θα είναι μεγαλύτερο από 2GB. Αν δεν το κάνουμε αυτό, θα έχουμε πρόσβαση μόνο στα πρώτα 2GB του δίσκου.[22]

Χαρακτηριστικά FAT32

- Το FAT32 χρησιμοποιεί μικρές συστοιχίες με αποτέλεσμα την αποδοτικότερη χρήση του χώρου σε σχέση με το fat16
- Το FAT32 υποστηρίζει σκληρό δίσκο κάτω από 2 terabytes στο μέγεθος
- Το FAT32 είναι πιο ασφαλές. Χρησιμοποιεί κάποιο εφεδρικό αντίγραφο του πίνακα κατανομής αρχείων σε περίπτωση που το πρωτότυπο καταστραφεί.[23]

3.3 Το σύστημα αρχείων VFAT

Το VFAT ονομαζόταν στο παρελθόν xmsdos και μετονομάστηκε σε VFAT. Πολλές εκδόσεις χρησιμοποιούν ακόμα το όνομα xmsdos. Η Microsoft ενσωμάτωσε διάφορες βελτιώσεις στις δυνατότητες διαχείρισης του δίσκου στα Windows 95. Η πρόσβαση στο σύστημα αρχείων μπορεί να γίνει χρησιμοποιώντας υψηλής ταχύτητας, 32-bit οδηγούς, ή για συμβατότητα τις παλιότερες 16-bit ρουτίνες του DOS. Προστέθηκε επίσης υποστήριξη για μεγάλα ονόματα αρχείων (Μεγάλα ονόματα αρχείων μέχρι 255 χαρακτήρες ανά αρχείο, μπορεί να αντιστοιχούν σε κάποιο αρχείο) 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 το VFAT ως σύστημα αρχείων, είναι βασικά το ίδιο με το FAT. Οι περισσότερες από τις νέες ικανότητες σχετίζονται με το πως χρησιμοποιείται το σύστημα αρχείων. Το VFAT διαχειρίζεται τυπικά FAT16 διαμερίσματα, καθώς και διαμερίσματα FAT32, κάτω από Windows 95 ή μεταγενέστερα. [24]

3.4 Το σύστημα αρχείων NTFS

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. [25]

Χαρακτηριστικά γνωρίσματα του NTFS

Κάποια χαρακτηριστικά γνωρίσματα του συστήματος αρχείων NTFS είναι τα ακόλουθα:

- Το NTFS δίνει σημαντικά πλεονεκτήματα στους διαχειριστές του συστήματος, καθώς μπορούν να καθορίσουν ποιοι χρήστες θα έχουν δικαιώματα στα αρχεία
- Το σύστημα αρχείων NTFS προσφέρει καλύτερη συμπίεση αρχείων και καταλόγων για αποθήκευση δεδομένων
- Το NTFS υποστηρίζει μεγάλα μεγέθη αρχείων
- Το NTFS χρησιμοποιεί τη μέθοδο κατάτμησης του διαστήματος, χώρου. Δηλαδή ο δίσκος είναι χωρισμένος σε μικρά κομμάτια για την εξοικονόμηση χώρου όσον αφορά την αποθήκευση των αρχείων
- Δεν παρουσιάζει κάποιο πρόβλημα όσον αφορά το μέγεθος των συστοιχιών και το μέγεθος του σκληρού δίσκου
- Το NTFS επιτρέπει ονόματα αρχείων πάνω από 255 χαρακτήρες.
- Πλήρη έλεγχο πρόσβασης
- Συνδέσεις αρχείων-καταλόγων[26]

NTFS ιδιότητες αρχείων

-Όνομα αρχείων (FN): Καταχωρεί ένα όνομα σε ένα αρχείο

-Ιδιότητες ρίζας ευρετηρίων: Περιέχει την εύρεση των αρχείων που περιλαμβάνονται μέσα σε έναν κατάλογο .

-Περιγραφέας (SD) ασφάλειας: Περιέχει τις πληροφορίες ασφάλειας που ελέγχουν την πρόσβαση σε ένα αρχείο ή έναν κατάλογο αρχείων.

-Πρότυπες πληροφορίες (SI): Περιέχει τις "πρότυπες πληροφορίες" για όλα τα αρχεία και τους καταλόγους, όπως η ημερομηνία/ time-stamps για όταν δημιουργήθηκε το αρχείο, τροποποιήθηκε και προσπελάστηκε.

-Όνομα, πληροφορίες και έκδοση όγκου: Αυτές οι τρεις ιδιότητες καταχωρούν το βασικό όνομα, την έκδοση και άλλες πληροφορίες για τον όγκο του δίσκου του NTFS.[27]

3.5 Κρυπτογραφία στο NTFS

Στο NTFS μόνο οι εξουσιοδοτημένοι χρήστες έχουν την δυνατότητα να έχουν πρόσβαση ή έλεγχο στο σύστημα. Αυτό το σύστημα λειτουργεί μόνο όταν οι χρήστες βρίσκονται μέσα στο σύστημα. Δυστυχώς όμως υπάρχει πάντα πιθανότητα πρόσβασης στο σύστημα από έναν κακόβουλο χρήστη, ο οποίος μπορεί να παρακάμπτει τις μεθόδους ασφαλείας.

Προκειμένου λοιπόν να λυθεί αυτό το πρόβλημα η Microsoft εισήγαγε μια ικανότητα κρυπτογράφησης, την Encrypting File System ή EFS. Το EFS λειτουργεί ως εξής:

Κάθε χρήστης έχει ένα δημόσιο και ένα ιδιωτικό κλειδί. Το δημόσιο κλειδί μπορεί να είναι γνωστό στους άλλους, ενώ το ιδιωτικό είναι φυσικά γνωστό μόνον σε αυτόν που κάνει την κρυπτογράφηση. Όταν ένα αρχείο κρυπτογραφείται αυτό γίνεται φυσικά χρησιμοποιώντας το δημόσιο κλειδί. Για να αποκρυπτογραφηθεί το αρχείο πρέπει να είναι γνωστό το ιδιωτικό κλειδί. Το EFS εξασφαλίζει ότι μόνο το άτομο που κρυπτογραφεί το αρχείο μπορεί και να το αποκρυπτογραφήσει. Εάν οποιοσδήποτε προσπαθήσει να ανοίξει, να αντιγράψει, να κινήσει ή να μετονομάσει οποιοδήποτε από τα κρυπτογραφημένα αρχεία, θα σταματήσει από ένα μήνυμα που θα λέει ότι δεν επιτρέπεται η πρόσβαση. Όταν κρυπτογραφούμε ένα φάκελο, όλα τα αρχεία και οι υποφάκελοι που βρίσκονται στον συγκεκριμένο φάκελο κρυπτογραφούνται και αυτά.

Αρχές ασφαλείας στο NTFS

Μια από τις αρχές ασφαλείας στο NTFS είναι η ανάθεση δικαιωμάτων σε χρήστες ή ομάδα χρηστών. Σε ένα δίκτυο υποθέτουμε ότι είναι συνδεδεμένες πολλές μηχανές πελατών(client machines). Οποιοσδήποτε χρήστης καθίσει σε μία από αυτές τις θέσεις

μπορεί να συνδεθεί στο server, αλλά πρέπει να κάνει Login στον server για να μπορεί να έχει πρόσβαση σε αυτόν.

Ένας χρήστης για να χρησιμοποιήσει το δίκτυο πρέπει να έχει λογαριασμό. Το ίδιο ισχύει φυσικά και για τις ομάδες, στις οποίες υπάγονται οι χρήστες. Οι ομάδες αυτές δίνουν δικαιώματα στους χρήστες τους να μοιράζονται τις ίδιες διεργασίες. Για παράδειγμα οι χρήστες που ανήκουν στο ίδιο τμήμα ενός οργανισμού μοιράζονται τις ίδιες διεργασίες καθώς ανήκουν στην ίδια ομάδα. Αν κάποιος χρήστης δεν έχει λογαριασμό στο δίκτυο τότε έχει τη δυνατότητα να χρησιμοποιήσει το λογαριασμό της ομάδας του, ειδάλλως χρησιμοποιεί έναν λογαριασμό επισκέπτη αλλά σαφώς με λιγότερα δικαιώματα σε σχέση με τους υπόλοιπους χρήστες που έχουν λογαριασμό. Αν κάποιος δεν έχει ούτε λογαριασμό επισκέπτη τότε αυτό σημαίνει ότι δεν έχει δυνατότητα να μπει στα server του δικτύου.

Ένας χρήστης μπορεί να είναι μέλος διαφορετικών ομάδων. Στο σύστημα υπάρχουν αρκετές ομάδες οι οποίες έχουν συγκεκριμένες ιδιότητες. Μια από αυτές τις ιδιότητες είναι ο διαχειριστής ομάδας, που έχει πρόσβαση σχεδόν παντού και όπως λέει και η λέξη διαχειρίζεται, ελέγχει τους χρήστες που ανήκουν σε αυτή την ομάδα. Δύο άλλες αρχές ασφάλειας στο NTFS είναι η ιδιοκτησία(ownership) και auditing(ακρόαση). Το ownership είναι ένα συγκεκριμένο δικαίωμα που δίνει δυνατότητα στους ιδιοκτήτες να παρέχουν άδειες, δικαιώματα σε άλλους. Τέλος, το auditing επιτρέπει στους διαχειριστές να παρακολουθούν τις αλλαγές που γίνονται στα αρχεία και τους καταλόγους

Άδειες στο NTFS

Η λίστα ελέγχου πρόσβασης χρησιμοποιείται για να διαχειριστεί ποιος χρήστης και ποια ομάδα χρηστών επιτρέπεται να έχουν πρόσβαση στα αρχεία ή τους φακέλους. Πιο συγκεκριμένα η λίστα ελέγχου πρόσβασης καθορίζει τι δικαιώματα έχει κάθε χρήστης ή ομάδα.

Για το NTFS υπάρχουν 6 διαφορετικοί τύποι αδειών. Κάθε ένας από αυτούς τους τύπους αδειών έχει διαφορετικό τύπο πρόσβασης σε ένα αντικείμενο και έχουν ένα συγκεκριμένο γράμμα που τους καθορίζει. Ο πίνακας 3.5 δείχνει τα διαφορετικά δικαιώματα και πως εφαρμόζονται στα αρχεία και στους φακέλους.

Δικαιώματα	Γράμμα συντόμευσης	Άδειες που παραχωρούνται στα αρχεία	Άδειες που παραχωρούνται στους καταλόγους
Ανάγνωση	R	Ανάγνωση περιεχομένων αρχείων	Ανάγνωση περιεχομένων καταλόγου
Εγγραφή	W	Αλλαγή περιεχομένων αρχείων	Αλλαγή περιεχομένων καταλόγου (δημιουργία νέων αρχείων ή καταλόγων)
Εκτέλεση	X	Εκτέλεση(τρέξιμο)ενός προγράμματος αρχείων	Διείσδυση και χρησιμοποίηση υποκαταλόγου ή φακέλου
Διαγραφή	D	Διαγραφή αρχείων	Διαγραφή καταλόγου
Αλλαγή δικαιωμάτων	P	Αλλαγή ρυθμίσεων δικαιωμάτων αρχείων	Αλλαγή ρυθμίσεων δικαιωμάτων φακέλων
Ιδιοκτησία	O	Ιδιοκτησία αρχείου	Ιδιοκτησία καταλόγου

Πίνακας 3.5: Διάφορα δικαιώματα και πως εφαρμόζονται στα αρχεία και στους φακέλους.[28]

3.6 Μεταδεδομένα στο NTFS

Το σύστημα αρχείων NTFS αποθηκεύει όλες τις πληροφορίες στα αρχεία. Οι πιο σημαντικές από αυτές τις πληροφορίες είναι ένα σύνολο ειδικών συστημάτων αρχείων που ονομάζονται metadata files(μεταδεδομένα αρχείων). Τα μεταδεδομένα αρχείων είναι αρχεία που περιλαμβάνουν πληροφορίες για τα δεδομένα όπως όνομα δεδομένων, χρόνος δημιουργίας τους κτλ. Αυτά τα μεταδεδομένα αρχείων δημιουργούνται αυτόματα από το σύστημα όταν το NTFS γίνεται Format. Το πως δουλεύουν αυτά αρχεία είναι κάπως περίπλοκο. Μαζί με τα υπόλοιπα αρχεία απαιτείται και ένα NTFS Master File Table (MFT). Το MFT περιλαμβάνει περιγραφές όχι μόνο για τα αρχεία αλλά και για τα άλλα

μεταδεδομένα αρχείων. Είναι το μέρος του συστήματος αρχείων που δίνει πληροφορίες για τα μεταδεδομένα, τα αρχεία και τους καταλόγους που είναι αποθηκευμένα στο δίσκο. Το MFT περιλαμβάνει μια εγγραφή που περιγράφει κάθε αρχείο και κατάλογο στο NTFS. Αν τα αρχεία είναι αρκετά μικρά, τα περιεχόμενά τους μπορούν να αποθηκευτούν στο MFT. Τα μεταδεδομένα αρχείων στο NTFS έχουν εγγραφές στο MFT. Για την ακρίβεια, οι 16 πρώτες εγγραφές στο MFT αποτελούνται από αρχεία μεταδεδομένων. Μπορούμε να δούμε τα μεταδεδομένα πληκτρολογώντας <<dir /ah filename>>στη ρίζα οδηγού εκκίνησης(π.χ ex: dir /a:h \$BOOT).Ο πίνακας 3.6 παρέχει σημαντικές πληροφορίες για τα αρχεία μεταδεδομένων, περιλαμβάνοντας τα αγγλικά τους ονόματα, τα ονόματα αρχείων, τον αριθμό εγγραφής τους στο MFT και μια μικρή περιγραφή για το καθένα.

Όνομα αρχείων μεταδεδομένων	Όνομα αρχείων	MFT Εγγραφή #	Περιγραφή
Master File Table (MFT)	\$MFT	0	Αυτή η πρώτη MFT εγγραφή περιλαμβάνει πληροφορίες για το MFT.
Master File Table 2 (MFT2) or Master File Table Mirror	\$MFTMirr	1	Αυτό είναι ένας καθρέφτης των 16 εγγραφών του Master File Table. Ο καθρέφτης είναι ένα αντίγραφο που χρησιμοποιείται για να διαβεβαιώσει ότι οι πρώτες εγγραφές του MFT, που περιγράφουν τα μεταδεδομένα, μπορεί να γίνουν αντικείμενο πρόσβασης σε περίπτωση λάθους στο δίσκο.
Log File	LogFile	2	Εκτέλεση αρχείου πρόσβασης.
Volume Descriptor	\$Volume	3	Περιλαμβάνει πληροφορίες όπως όνομα, χρόνος δημιουργίας, κτλ
Attribute Definition Table	\$AttrDef	4	Αυτός ο πίνακας περιλαμβάνει τα ονόματα και τα χαρακτηριστικά του <u>NTFS</u> .
Root Directory /Folder	"." (single period)	5	Αυτός είναι ένας δείκτης στον root(ρίζας) κατάλογο ή στο φάκελο.

Cluster Allocation Bitmap	\$Bitmap	6	Δείχνει ποια συστοιχία χρησιμοποιείται και ποια είναι διαθέσιμη για χρήση.
Volume Boot Code	\$Boot	7	Αυτή η εγγραφή περιλαμβάνει ένα αντίγραφο του κώδικα εκκίνησης (ή ένα δείκτη σε αυτό).
Bad Cluster File	\$BadClus	8	Μια λίστα όλων των συστοιχιών που έχουν μαρκαριστεί σαν "άσχημα" (σημαίνει ότι ανιχνεύθηκε ένα λάθος κάπου σε αυτές τις συστοιχίες, έτσι το σύστημα αρχείων θέλει να βεβαιωθεί ώστε να μη τις χρησιμοποιήσει ξανά.)
Quota Table	\$Quota	9	Πίνακας που περιλαμβάνει quota πληροφορίες(ποσοστό δίσκου που εκχωρείται σε κάθε χρήστη)
Upper Case Table	\$UpCase	10	Ο πίνακας περιλαμβάνει πληροφορίες για ονομασία συστήματος αρχείων
NTFS Extensions	\$Extend	11-15	Διατηρημένα αλλά μη χρησιμοποιημένα. Οι εγγραφές 11 με 15 στο MFT κρατώνται για μελλοντικά μεταδεδομένα αρχείων.

Πίνακας 3.6: Πληροφορίες για τα αρχεία μεταδεδομένων [29,30]

ΚΕΦΑΛΑΙΟ 4 Το σύστημα αρχείων EXT2

Ιστορική παρουσίαση

Βασικές αρχές υλοποίησης

Σύγκριση με FAT

Εργαλεία διαχείρισης του EXT2

Κρυπτογραφία στο EXT2

Ασφάλεια στο EXT2

Συμπίεση αρχείων

Μειονεκτήματα του EXT2

Επιδόσεις του EXT2

Περίληψη

Στο κεφάλαιο αυτό γίνεται αναφορά γύρω από το EXT2 σύστημα αρχείων. Πιο συγκεκριμένα, αναφέρονται τα χαρακτηριστικά και τα εργαλεία διαχείρισης του EXT2, η κρυπτογραφία και η ασφάλεια που υπάρχει στο υπάρχον σύστημα αρχείο και τέλος τα πλεονεκτήματα και τα μειονεκτήματα του.

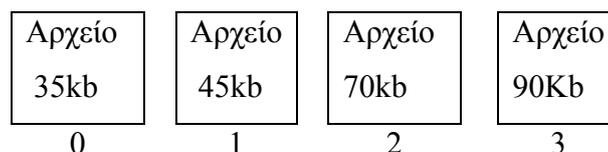
Στόχος του Ext2 είναι η εξασφάλιση ενός συστήματος αρχείου όπου θα προσφέρει ανεπτυγμένα χαρακτηριστικά, όπως είναι: τα μεγάλα ονόματα αρχείων, τα εργαλεία διαχείρισης και η ασφάλεια. Το μέγιστο μέγεθος του συστήματος αρχείων είναι 4TB, το μέγεθος των μπλοκ είναι 1- 4KB και το μέγιστο μέγεθος των αρχείων είναι 2GB.[31]

4.1 Ιστορική παρουσίαση

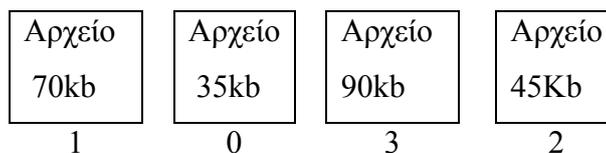
Το EXT2 σχεδιάστηκε από τον Wayne Davidson[32] για να επιλύσει κάποια από τα προβλήματα που εμφανίστηκαν στο EXT. Μέχρι την πρώτη έκδοση (0.02) το 1992 το Linux βασιζόταν στο λειτουργικό σύστημα Minix. Οι περιορισμοί όμως του Minix συστήματος αρχείων ήταν πολλοί. Το αποτέλεσμα λοιπόν ήταν οι σχεδιαστές να οδηγηθούν στην ανάπτυξη ενός στρώματος VFS, το οποίο περιγράφεται σαν Virtual File System. Μετά την ενσωμάτωση του VFS στον πυρήνα, ένα νέο σύστημα καλούμενο Extended File System σχεδιάστηκε τον Απρίλιο του 1992 από τον Remy Card[33] και προστέθηκε στο Linux.

Αυτό το σύστημα αρχείων απομάκρυνε τους δύο μεγάλους περιορισμούς του Minix: το μέγιστο μέγεθος του δίσκου που ήταν 2 giga bytes και το μέγιστο μήκος του ονόματος των αρχείων που ανερχόταν στους 255 χαρακτήρες. Ήταν μια βελτίωση του Minix συστήματος αρχείων αλλά κάποια προβλήματα συνέχιζαν να υπάρχουν και σε αυτό. Μερικά από αυτά τα προβλήματα ήταν ότι δεν παρείχε υποστήριξη για ξεχωριστή πρόσβαση των χρηστών στα δεδομένα από διάφορα τερματικά παρά μόνο από ένα τερματικό, τροποποίηση των i-node και τροποποίηση δεδομένων. Το σύστημα αρχείων χρησιμοποιούσε λίστες για να διατηρήσει τα ίχνη, το ιστορικό των ελεύθερων block και i-node, κάτι το οποίο παρείχε λανθασμένα αποτελέσματα καθώς όσο το σύστημα αρχείων χρησιμοποιούνταν, οι λίστες αναταξινομούσαν. Στο σχήμα 4.1 παρατηρούμε μια ταξινομημένη και μια μη ταξινομημένη λίστα με βάση το μέγεθος των αρχείων:

Ταξινομημένη λίστα



Μη ταξινομημένη λίστα



Σχήμα 4.1: Ταξινομημένη και μη ταξινομημένη λίστα με βάση το μέγεθος των αρχείων

Έτσι λοιπόν το 1993 δύο νέα συστήματα αρχείων έκαναν την εμφάνισή τους: το σύστημα αρχείων Xia και το Second Extended File System. Το Xia σύστημα αρχείων βασιζόταν στο Minix σύστημα αρχείων και απλά προστέθηκαν κάποιες βελτιώσεις σε αυτό, όπως μεγάλα ονόματα αρχείων, υποστήριξη μεγάλων διαμερίσεων και υποστήριξη για τα timestamps⁹. Από την άλλη μεριά, το EXT2 βασίστηκε στο EXT, είναι η συνέχεια του EXT με πολλές αναδιοργανώσεις και βελτιώσεις και παρείχε δυνατότητες για μελλοντικές βελτιώσεις. Αρχικά τα δύο νέα συστήματα (Xia-ext2) παρείχαν τα ίδια χαρακτηριστικά. Αργότερα όμως όσο τα συστήματα αρχείων χρησιμοποιούνταν όλο και πιο πολύ, στο Ext2 προστέθηκαν πολλές βελτιώσεις και νέα χαρακτηριστικά. Ο πίνακας 4.1 περιλαμβάνει κάποια από τα χαρακτηριστικά διαφορετικών συστημάτων αρχείων:

	Minix FS	Ext FS	Ext2 FS	Xia FS
Μέγιστο μέγεθος FS	64 MB	2 GB	4 TB	2 GB
Μέγιστο μέγεθος αρχείου	64 MB	2 GB	2 GB	64 MB
Μέγιστο όνομα αρχείου	16/30 c	255 c	255 c	248 c

Πίνακας 4.1: Χαρακτηριστικά διάφορων συστημάτων αρχείων [34]

⁹ Timestamps: αποτυπώσεις χρόνου-ίχνη (Συμβολοσειρά που έχει ημερομηνία και ώρα τελευταίας τροποποίησης αρχείου)

4.2 Βασικές αρχές υλοποίησης

Κάθε σύστημα αρχείων στο Linux έχει κάποιες βασικές αρχές υλοποίησης οι οποίες στο συγκεκριμένο σύστημα αρχείων συνίσταται σε κόμβους, καταλόγους, συνδέσεις και ειδικά αρχεία συσκευών.

Κόμβοι: Κάθε αρχείο αποτελείται από μια δομή που ονομάζεται κόμβος. Κάθε κόμβος περιλαμβάνει την περιγραφή ενός αρχείου: τύπο αρχείου, δικαιώματα πρόσβασης, ιδιοκτήτες, μέγεθος.

Κατάλογοι: Οι κατάλογοι ταξινομούνται σε ιεραρχικό δέντρο. Κάθε κατάλογος μπορεί να συμπεριλάβει αρχεία και άλλους καταλόγους.

Συνδέσεις: Αρκετά ονόματα μπορούν να συνδεθούν με έναν κόμβο.

Ειδικά αρχεία συσκευών: Είναι ένας δείκτης πρόσβασης στον οδηγό συσκευής. [35]

4.3 Σύγκριση με FAT

Καταρχήν, το Fat έχει μέγιστο διαθέσιμο χώρο όσον αφορά τις διευθύνσεις 2TB ενώ το Ext2 4TB. Επίσης, στο Ext2 σύστημα αρχείων τα αρχεία μπορούν να έχουν μεγάλα ονόματα μέχρι 255 χαρακτήρες. Τα ονόματα αρχείων μπορούν να δημιουργηθούν από οποιοδήποτε γράμμα, αριθμό ή κάποιο άλλο χαρακτήρα. Τα κεφαλαία και τα μικρά γράμματα διαφέρουν. Στα ονόματα αρχείων μπορεί να υπάρξουν και τελείες ενώ οι επεκτάσεις(Like.exe)δεν έχουν καμία σημασία. Ένα πρόγραμμα λέγεται απλά με το όνομά του χωρίς την ύπαρξη επέκτασης. Αν είναι απαραίτητο μπορούν να χρησιμοποιηθούν σε ένα όνομα αρχείου χαρακτήρες όπως `*`, `?`, `|`, `+` και `>`. Η μόνη εξαίρεση είναι το «/».

Στο Fat από την άλλη μεριά τα κεφαλαία και τα μικρά γράμματα δεν διαφέρουν, δεν μπορούν να υπάρξουν τελείες στα ονόματα αρχείων και οι επεκτάσεις είναι απαραίτητες.

Στο παρακάτω παράδειγμα βλέπουμε κάποιες από τις διαφορές μεταξύ του fat και του ext2 ως προς τα ονόματα των αρχείων: [36]

Fat: `\dir1\dir2\filename.ext`,

Ext2: `/home/itsme/testfile`

4.4 Εργαλεία διαχείρισεως του Ext2

Τα εργαλεία διαχείρισεως χρησιμοποιούνται για να δημιουργήσουν, τροποποιήσουν και διορθώσουν οποιαδήποτε ασυνέπεια στο Ext2 σύστημα αρχείων.

- Το εργαλείο [mke2fs](#) χρησιμοποιείται για να αρχικοποιήσει μια διαμέριση ώστε να δημιουργήσει ένα Ext2 σύστημα αρχείων σε αυτή.
- Το εργαλείο [Tune2fs](#) μας επιτρέπει να ορίσουμε συμπεριφορά για τυχόν σφάλματα, μέγιστο έλεγχο των συστημάτων αρχείων, αριθμό block κτλ.
- Το [E2fsck](#) προορίζεται για να επιδιορθώσει τις ασυνέπειες του συστήματος αρχείων μετά από ένα unclean κλείσιμο του συστήματος και όχι μόνον. Η παρούσα έκδοση του e2fsck σε σχέση με την αρχική έκδοση, που βασιζόταν στο πρόγραμμα του Linus Torvalds[37] fsck για το Minix σύστημα αρχείων, είναι πολύ γρηγορότερη και μπορεί να διορθώσει περισσότερες ασυνέπειες του συστήματος αρχείων.
- Το [Debugfs](#) είναι ένα ισχυρό πρόγραμμα που μπορεί να χρησιμοποιηθεί για να εξετάσει και να αλλάξει την κατάσταση ενός συστήματος αρχείων. Το Debugfs μπορεί να χρησιμοποιηθεί για να εξετάσει τις εσωτερικές δομές ενός συστήματος αρχείων, να επισκευάσει ένα σύστημα αρχείων κτλ. Το debugfs προκειμένου να μην υπάρξει κίνδυνος καταστροφής ενός συστήματος αρχείων, ανοίγει τα συστήματα αρχείων μόνο για πρόσβαση ανάγνωσης.[38]
- [dumpe2fs](#):Αυτό το εργαλείο καταγράφει τη θέση του ext2 συστήματος αρχείων στο δίσκο. Χρησιμοποιείται για την ανάλυση των προβλημάτων καθώς επίσης και στις γενικές πληροφορίες για τη χρήση του filesystem.
- [badblocks](#):Αυτό το εργαλείο ψάχνει το δίσκο για να ελέγξει ποιες ομάδες δεδομένων έχουν πάθει ζημιά.[39]
- [chattr](#):αλλάζει τα χαρακτηριστικά των αρχείων σε ένα ext2 σύστημα αρχείων.
- [e2label](#):σε περίπτωση που σε μια συσκευή, σε ένα σύστημα γίνει αλλαγή ως προς το σύστημα αρχείων το εργαλείο αυτό θα προσθέσει και την ανάλογη ετικέτα που θα το χαρακτηρίζει.
- [lsattr](#):βάζει σε μια λίστα τα χαρακτηριστικά του συστήματος αρχείων Ext2.[40]

4.5 Κρυπτογραφία στο Ext2

Υπάρχουν διάφοροι μέθοδοι δημιουργίας κρυπτογραφημένου συστήματος αρχείων. Μία από αυτές τις μεθόδους είναι η “Loopback device”, η οποία είναι γρήγορη και αξιόπιστη. Η λειτουργία της είναι η ακόλουθη:

1.Ο χρήστης δημιουργεί ένα αρχείο με n bytes(οποιοσδήποτε αριθμός θέλει ο χρήστης) σε ένα σύστημα αρχείων που ήδη υπάρχει.

if=/dev/zero of=/myefs bs=1k count=100

(dd=δημιουργία αρχείου, myefs=όνομα αρχείου)

2.Στη συνέχεια ο χρήστης οργανώνει τη “Loopback device”(κάνει Mount ένα αρχείο σαν να είναι διαμέριση) να δείχνει στο αρχείο χρησιμοποιώντας serpent κρυπτογράφηση:

losetup -e serpent /dev/loop0 /myefs

3.Ακολούθως ο χρήστης δημιουργεί ένα καινούριο σύστημα αρχείων με την εντολή mkfs για πρώτη φορά μέσα στη loopback Device(έτσι θα δημιουργηθεί το νέο σύστημα αρχείων μέσα στο αρχείο των n bytes).Το σύστημα αρχείων κρυπτογραφείται αυτόματα καθώς δημιουργείται:

mkfs -t ext2 /dev/loop0 100

(100=μέγεθος συστήματος αρχείων. Έχει το ίδιο μέγεθος με το dd)

4.Στη συνέχεια ο χρήστης προσαρτεί το κρυπτογραφημένο σύστημα αρχείων. Έτσι ο χρήστης αποκτά πρόσβαση στο κρυπτογραφημένο σύστημα αρχείων το οποίο βρίσκεται στο αρχείο των N bytes.:

mount -t ext2 /dev/loop0 /mnt

Αυτή η εντολή λέει στον πυρήνα ότι η συσκευή **/dev/loop0** έχει ένα Ext2 σύστημα αρχείων.

Οποιοδήποτε αρχείο προσαρτηθεί στο κρυπτογραφημένο σύστημα αρχείων θα κρυπτογραφηθεί αυτόματα ενώ αντίστοιχα θα αποκρυπτογραφηθεί κατά την ανάγνωσή του.

5. Τέλος απενεργοποιούμε, απελευθερώνουμε την loopback συσκευή.

umount /dev/loop0

losetup -d /dev/loop0

Η μέθοδος αυτή έχει τα εξής πλεονεκτήματα: Καταρχήν είναι απλή καθώς δεν υπάρχει πολύς κώδικας. Επίσης, είναι δύσκολο να το σπάσει κάποιος hacker καθώς περιέχεται αποκλειστικά σε ένα αρχείο για το οποίο δε γνωρίζει τίποτα.[41]

4.6 Ασφάλεια στο Ext2

Ο πυρήνας του Linux αποτελείται από βοηθήματα, εργαλεία που αναπτύσσονται για να βελτιώσουν την ασφάλεια του λειτουργικού συστήματος παρέχοντας ελέγχους πρόσβασης σε αυτό. Στόχος της ασφάλειας είναι να περιοριστεί η πιθανή ζημιά που μπορεί να προκληθεί μέσω της παραβίασης μιας διαδικασίας από την εισαγωγή χρηστών χωρίς δικαίωμα πρόσβασης.[42]

Οι **κανόνες** όταν γίνεται πρόσβαση στα δεδομένα είναι οι ακόλουθοι:

-Εάν η διαδικασία έχει root δικαιώματα , η πρόσβαση επιτρέπεται

-Εάν η πρόσβαση πρόκειται να γίνει από τον ιδιοκτήτη ή τον δημιουργό του UID γίνεται έλεγχος για να εξακριβωθεί αν έχει δοθεί το δικαίωμα πρόσβασης σε αυτόν.

-Εάν η πρόσβαση πρόκειται να γίνει από την ομάδα που δημιούργησε το GID, τότε γίνεται έλεγχος αν έχει δοθεί το δικαίωμα πρόσβασης σε αυτή την ομάδα

-Διαφορετικά γίνεται έλεγχος αν οι υπόλοιποι χρήστες έχουν το δικαίωμα πρόσβασης.[43]

Ένα παράδειγμα με δικαιώματα πρόσβασης είναι το ακόλουθο:

```
rwxr-xr-x 1 root root 8632 May 9 1998 /usr/bin/id
```

Εδώ βλέπουμε ότι το αρχείο είναι αναγνώσιμο, εγγράψιμο και εκτελέσιμο από τον ιδιοκτήτη, από την ομάδα είναι αναγνώσιμο και εκτελέσιμο και για όλους τους άλλους είναι μόνο εκτελέσιμο.

Ο **αλγόριθμος** που χρησιμοποιείται για πρόσβαση σε ένα αρχείο είναι ο ακόλουθος:

-ο έλεγχος όσον αφορά τα δικαιώματα που δίνονται στους χρήστες ξεκινάει από την ρίζα, συνεχίζει στους πατρικούς καταλόγους και καταλήγει στο αρχείο. Εάν η πρόσβαση αμφισβητηθεί από τη ρίζα ή από τον πατρικό κατάλογο ή από το αρχείο, τότε δεν επιτρέπεται η πρόσβαση του χρήστη.[44]

Στο Ext2 σύστημα αρχείων υπάρχει η εντολή dump. Εδώ καλούμαστε να ορίσουμε αν θα ζητήσουμε να γίνει backup του συστήματος αρχείου κατά την εκκίνηση. Οι επιλογές είναι δύο: η πρώτη επιλογή είναι μηδέν(0) και η δεύτερη είναι ένα (1). Με μηδέν το σύστημα αρχείων δεν θα γίνει backup, ενώ με ένα (1) θα γίνει ένα Backup.[45]

Η ασφάλεια των δεδομένων είναι πρωταρχικής σημασίας για το σύστημα, και γι' αυτό δίνεται η δυνατότητα για:

-Password protection(προστασία κωδικού πρόσβασης)

-Encryption(κρυπτογράφηση)

-Password protection and Encryption(προστασία κωδικού πρόσβασης και κρυπτογράφηση)
[46]

4.7 Συμπίεση αρχείων

To e2compr

Το E2compr είναι ένα πρόγραμμα που επιτρέπει την συμπίεση και την αποσυμπίεση αρχείων. Συμπιέζει μόνο τα κανονικά αρχεία. Τα αρχεία: superbblock, inodes, directory files κτλ δεν συμπιέζονται(κυρίως για λόγους ασφάλειας). Οι κατάλογοι μπορούν επίσης να συμπιεστούν, κάτι που σημαίνει ότι κάθε νέο δημιουργημένο αρχείο στον κατάλογο θα συμπιεστεί αυτόματα. Μπορούμε να συμπιέσουμε κάθε αρχείο ξεχωριστά στον κατάλογο αλλά για λόγους διευκόλυνσης έχουμε τη δυνατότητα να συμπιέσουμε τον κατάλογο ο οποίος και περιέχει αυτά τα αρχεία. Με αυτόν τον τρόπο συμπιέζονται αυτόματα όλα τα αρχεία που περιέχονται σε αυτόν. [47]

4.8 Μειονεκτήματα του Ext2

-Το Ext2 σύστημα αρχείων δεν παρέχει καλή διαχείριση των τομέων που βασίζονται στην εγγραφή και την διαγραφή. Για να διαγράψεις ένα απλό Byte σε έναν τομέα στο Ext2, ολόκληρος ο τομέας θα πρέπει να αντιγραφεί στη μνήμη RAM, να διαγραφεί και να επανεγγραφεί.

-Η διαδικασία Fscck(έλεγχος συστημάτων αρχείων που χρησιμοποιείται από ext2) επιβραδύνει την εκκίνηση των συστημάτων.

-Δεν είναι αξιόπιστο σε περίπτωση αποτυχίας(chrash). [48]

4.9 Επιδόσεις του Ext2

Ο κώδικας πυρήνων Ext2fs περιέχει πολλές βελτιστοποιήσεις σχετικά με την απόδοση, οι οποίες τείνουν να βελτιώσουν την ταχύτητα κατά την ανάγνωση και το γράψιμο των αρχείων.

Το Ext2fs περιέχει επίσης πολλές βελτιστοποιήσεις όσον αφορά την κατανομή. Οι ομάδες block χρησιμοποιούνται για να συνδυάσουν μεταξύ τους i-nodes και δεδομένα: ο κώδικας πυρήνων προσπαθεί πάντα να καταναίμει τις ομάδες Block για ένα αρχείο στην ίδια ομάδα με το i-node του. Αυτό μειώνει τις αναζητήσεις δίσκων που δημιουργούνται όταν ο πυρήνας διαβάζει ένα i-node και τις ομάδες των Block του.

Περιγραφή των συγκριτικών μετρήσεων επιδόσεων

Έχουν γίνει διάφορες δοκιμές όσον αφορά τις επιδόσεις ενός filesystem. Οι δοκιμές των επιδόσεων έχουν γίνει σε ένα PC , βασισμένο σε έναν επεξεργαστή i486DX2, χρησιμοποιώντας 16 MB μνήμης και δύο δίσκους 420 MB IDE. Οι δοκιμές έτρεξαν σε Ext2 fs και Xia fs .

Έχουν τρέξει δύο διαφορετικές δοκιμές όσον αφορά την επίδοση. Η δοκιμή Bonnie εξετάζει την ταχύτητα σε ένα μεγάλο αρχείο (το μέγεθος του αρχείου ήταν 60 MB κατά τη διάρκεια των δοκιμών), γράφει τα δεδομένα στο αρχείο, επανεγγράφει τα περιεχόμενα ολόκληρου του αρχείου, γράφει δεδομένα χρησιμοποιώντας block, διαβάζει τα αρχεία, και κάνει αναζητήσεις στο αρχείο. Η δοκιμή των επιδόσεων του Andrew αναπτύχθηκε στο πανεπιστήμιο Carnegie Mellon και έχει χρησιμοποιηθεί στο πανεπιστήμιο του Μπέρκλεϋ.

Τρέχει σε πέντε φάσεις: δημιουργεί μια ιεραρχία καταλόγου, κάνει ένα αντίγραφο των δεδομένων, εξετάζει τη κατάσταση κάθε αρχείου, εξετάζει κάθε Byte κάθε αρχείου, και μεταγλωττίζει αρκετά από τα αρχεία.

Αποτελέσματα της δοκιμασίας των επιδόσεων Bonnie

Τα αποτελέσματα της δοκιμασίας των επιδόσεων Bonnie παρουσιάζονται στον πίνακα 4.2:

	Εγγραφή δεδομένων (KB/s)	Εγγραφή δεδομένων με χρήση Block (KB/s)	Επανεγγραφή δεδομένων (KB/s)	Ανάγνωση δεδομένων (KB/s)
Ext2 fs	452	1237	536	397
Xia fs	440	704	380	366

Πίνακας 4.2: Αποτελέσματα της δοκιμασίας των επιδόσεων Bonnie

Αποτελέσματα της δοκιμασίας επιδόσεων του Andrew

Τα αποτελέσματα της δοκιμασίας των επιδόσεων του Andrew παρουσιάζονται στον πίνακα 4.3:

	P1 Δημιουργία (ms)	P2 Αντιγραφή δεδομένων (ms)	P3 εξέταση κατάστασης αρχείου (ms)	P4 εξέταση Byte αρχείου (ms)	P5 Μεταγλώττιση αρχείων (ms)
Ext2 fs	790	4791	7235	11685	63210
Xia fs	934	5402	8400	12912	66997

Πίνακας 4.3 : Αποτελέσματα της δοκιμασίας επιδόσεων του Andrew [49]

Τελειώνοντας, ο πίνακας 4.4 δείχνει την κατάσταση του ext2:

Μέγιστο μέγεθος συστήματος αρχείων	4TB (Terabytes)
Μέγεθος Block	1KB to 4KB
Μέγιστο μέγεθος αρχείων	2GB
Ονόματα αρχείων	255 χαρακτήρες
Χρόνος επανεκκίνησης μετά από ένα λάθος	Πολύς
Ανάκτηση δεδομένων σε περίπτωση λάθους	Καλή αλλά υπάρχει ο κίνδυνος μερικής ή ολοκληρωτικής απώλειας δεδομένων.

Πίνακας 4.4: Κατάσταση Ext2 [50]

Επίσης:

-Ο διαχειριστής μπορεί να επιλέξει το λογικό μέγεθος των block κατά τη δημιουργία του συστήματος αρχείων(τα χαρακτηριστικά μεγέθη είναι 1024...2048 και 4096 bytes). [51]

ΚΕΦΑΛΑΙΟ 5 Το σύστημα αρχείων EXT3

Ιστορική παρουσίαση

Βασικές αρχές υλοποίησης

Επιδόσεις EXT3

Εργαλεία διαχείρισης EXT3

Σύγκριση με EXT2

Από το EXT2 στο EXT3 και αντίστροφα

Περίληψη

Το κεφάλαιο αναφέρεται στο Ext3 και πιο συγκεκριμένα στους τύπους journaling, στις βασικές αρχές υλοποίησης, στις επιδόσεις και στα εργαλεία διαχείρισης του Ext3 και τέλος στις ομοιότητες και τις διαφορές του με το Ext2 όπως επίσης και στη μετάβαση από το ένα σύστημα αρχείων στο άλλο.

Σαν ισχυρό λειτουργικό σύστημα καθώς είναι το Linux, παρέχει τη δυνατότητα ανάπτυξης του συστήματος και του λογισμικού. Ένα από τα αποτελέσματα της ανάπτυξης του συστήματος του Linux ήταν και το ext3 σύστημα αρχείων, το οποίο έχει ενσωματωθεί στον πυρήνα του Linux από την έκδοση 2.2.19.

Το Ext3, το οποίο δημιουργήθηκε από τον Stephen Tweedie[52], είναι ένα journaling(η δυνατότητα journaling αποτρέπει την αλλοίωση των μεταδεδομένων) σύστημα αρχείων που είναι συμβατό με το ext2 σύστημα αρχείων, το οποίο είναι το προκαθορισμένο σύστημα αρχείων που χρησιμοποιείται από τα συστήματα Linux τα τελευταία έτη.[53]

Εκτός από τη συμβατότητά του ext3 με το ext2 σύστημα αρχείων και την ευκολία με την οποία μπορούμε να μετατρέψουμε το ext2 σύστημα αρχείων σε ext3, το ext3 σύστημα αρχείων προσφέρει επίσης διάφορους τύπους journaling, τους οποίους μπορούμε να ενεργοποιήσουμε στο /etc/fstab για ένα ext3 σύστημα αρχείων. Πιο συγκεκριμένα το αρχείο /etc/fstab χρησιμοποιείται για να καθορίσει πως, πότε και πού πρέπει να προσαρτηθούν τα συστήματα αρχείων.

Ένα παράδειγμα χρησιμοποίησης της etc/fstab για τη καταχώρηση του τύπου Journaling, writeback, είναι το ακόλουθο:

```
/dev/hda5 /opt ext3 data=writeback 1 0
```

Οι τύποι journaling είναι οι ακόλουθοι:

journal-Αυτός ο τύπος απαιτεί το σύστημα αρχείων να γράφει κάθε αλλαγή που γίνεται στα δεδομένα και στα μεταδεδομένα δύο φορές, μια φορά στο σύστημα αρχείων και μια φορά στο τύπο journal. Κάτι τέτοιο μπορεί να μειώσει την απόδοση του συστήματος αλλά ελαχιστοποιεί την πιθανότητα λαθών στα αρχεία μας.

ordered-Καταχωρεί στο σύστημα αρχείων μόνο τις αλλαγές που γίνονται στα μεταδεδομένα, εφόσον πρώτα τα αρχεία δεδομένων ενημερώσουν τον δίσκο για τις αλλαγές που πρόκειται να υπάρξουν στα σχετικά μεταδεδομένα. Στον τύπο αυτό οι αλλαγές των μεταδεδομένων δεν εγγράφονται και στο σύστημα αρχείων και στο journal. Είναι επομένως γρηγορότερος σε σχέση με τον τύπο Journal. Αυτός είναι ο καθορισμένος τύπος στο Ext3.

writeback-Η χρήση του τρίτου τύπου, writeback, είναι γρηγορότερη από τους άλλους δύο τύπους καθώς καταγράφει τις αλλαγές μόνο στα μεταδεδομένα Filesystem και δε περιμένει

τα αρχεία δεδομένων να ενημερώσουν τον δίσκο για τις αλλαγές που πρόκειται να συμβούν στα μεταδεδομένα . [54]

5.1 Ιστορική παρουσίαση

Το ext3 προέρχεται άμεσα από τον πρόγονό του ext2 και έχει το χαρακτηριστικό ότι είναι απόλυτα συμβατό προς τα πίσω με το ext2, δεδομένου ότι είναι ακριβώς ένα ext2 σύστημα αρχείων με την επιπλέον ιδιότητα journaling. Το ext3 από τη στιγμή που είναι απόλυτα συμβατό με το ext2 σημαίνει ότι περιέχει τόσο τα πλεονεκτήματα όσο και τα μειονεκτήματα του ext2.

5.2 Βασικές αρχές υλοποίησης

Το ext3 έχει τις ακόλουθες αρχές υλοποίησης:

- 1.Metadata-μεταδεδομένα
- 2.Descriptor blocks-block περιγραφής
- 3.Header blocks-block επικεφαλίδας

Το **journal metadata block** περιλαμβάνει το Block που περιέχει τα μεταδεδομένα στο σύστημα αρχείων.

Το **Descriptor blocks** περιγράφει το journal metadata block. Πιο συγκεκριμένα περιέχει πληροφορίες για το journal metadata block όπως πότε δημιουργήθηκε ή πότε τροποποιήθηκε. Τα data descriptor block περιέχουν τα data headers block.

Το **header block** δίνει πληροφορίες για το descriptor block. [55]

5.3 Επιδόσεις ext3

Στον πίνακα 5.1 βλέπουμε τα χαρακτηριστικά του ext3:

Μέγιστο μέγεθος συστήματος αρχείων	4TB
Μέγεθος block	1KB με 4KB
Μέγιστο μέγεθος αρχείων	2GB
Χρόνος επανεκκίνησης μετά από ένα λάθος	Γρήγορος
Διόρθωση δεδομένων μετά από κάποιο σφάλμα	Δεν είναι διαθέσιμη

Πίνακας 5.1: Χαρακτηριστικά Ext3[56]

Επίσης, το Ext3 έχει τα εξής χαρακτηριστικά:

- Στο σύστημα αρχείων παρέχει χαρακτηριστικά Journaling γράφοντας τις αλλαγές σε δύο μέρη του δίσκου. Αυτό σημαίνει ότι παρέχει μεγαλύτερη ασφάλεια καθώς τα δεδομένα είναι αποθηκευμένα στο δίσκο τουλάχιστον μια φορά.
- Μετά από κάποιο σφάλμα του συστήματος τα δεδομένα μας δε χάνονται αλλά ούτε και μπερδεύονται με άλλα δεδομένα που βρίσκονται σε άλλο μέρος του δίσκου.
- Οι χρήστες επιλέγουν το είδος της ασφάλειας των δεδομένων τους.[57]

5.4 Εργαλεία διαχείρισης Ext3

- [mke3fs](#).

Αυτό το εργαλείο δημιουργεί νέα Ext3 συστήματα αρχείων με τον ίδιο τρόπο όπως το mke2fs δημιουργεί νέα Ext2 συστήματα αρχείων.

- [quota utils](#)

Τα εργαλεία quota θέτουν όρια στο χώρο που καταλαμβάνει κάθε χρήστης στο δίσκο.[58]

- [acl](#)

Αυτό το εργαλείο χρησιμοποιείται για να εκτελεί ελέγχους κατά την πρόσβαση στις λίστες αρχείων και καταλόγων στο σύστημα αρχείων

- **attr**

Αυτό το εργαλείο παρέχει τη δυνατότητα στους χρήστες να συνδέσουν το όνομα και τις ιδιότητες(αρχεία, καταλόγους) του συστήματος αρχείων με το αντίστοιχο σύστημα αρχείων.

- Σε περίπτωση που θελήσουμε να αναπτύξουμε το λογισμικό το οποίο θα χρησιμοποιήσει τα εργαλεία attr και acl θα χρειαστούμε τα εργαλεία **attr-devel** και **acl-devel** [59]

5.5 Σύγκριση με ext2

Υπάρχουν πολλές διαφορές και ομοιότητες μεταξύ των συστημάτων αρχείων ext2 και ext3. Οι σημαντικότερες από αυτές είναι οι ακόλουθες:

- Το Ext3 προσφέρει μεγαλύτερη ασφάλεια. Στην περίπτωση σφάλματος(απότομου κλεισίματος του H/Y) ο υπολογιστής θα ανανεώσει το σύστημα αρχείων και θα διατηρήσει όλες τις αλλαγές που κάναμε στα δεδομένα πριν κλείσουμε τον υπολογιστή. [60]
- Η επαναφορά των αρχείων ύστερα από ένα απροσδόκητο κλείσιμο του υπολογιστή είναι πολύ πιο γρήγορη στο ext3 από ότι στο ext2. Ο χρόνος να ανακτηθεί ένα ext3 σύστημα αρχείων μετά από ένα λανθασμένο κλείσιμο συστημάτων δεν εξαρτάται ούτε από το μέγεθος του συστήματος αρχείων αλλά ούτε και από τον αριθμό των αρχείων όπως συμβαίνει στο ext2, αλλά εξαρτάται από το μέγεθος του "journal"(αρχείο που καταγράφει τις αλλαγές στο σύστημα αρχείων). Το μέγεθος του journal χρειάζεται περίπου ένα δευτερόλεπτο για να ανακτηθεί.
- **Διαθεσιμότητα:** Το ext3 δεν απαιτεί να γίνει έλεγχος των συστημάτων αρχείων, ακόμα και μετά από ένα λανθασμένο κλείσιμο του συστήματος. Αυτό είναι επειδή σε περίπτωση σφάλματος(απότομου κλεισίματος του H/Y) ο υπολογιστής διατηρεί όλες τις αλλαγές που κάναμε στα δεδομένα.
- **Ακεραιότητα δεδομένων:** Όσον αφορά την ακεραιότητα των στοιχείων τώρα η χρησιμοποίηση του ext3 συστήματος αρχείων μπορεί να παρέχει τις ισχυρότερες εγγυήσεις για την ακεραιότητα των δεδομένων σε περίπτωση λανθασμένου κλεισίματος των συστημάτων. Και λέγοντας εγγυήσεις για την ακεραιότητα των δεδομένων φυσικά εννοούμε την αποφυγή αλλοίωσης των δεδομένων σε περίπτωση κάποιου λάθους όπως αυτό του απροσδόκητου κλεισίματος του συστήματος. Μπορούμε να επιλέξουμε από το /etc/fstab τον τύπο και το επίπεδο προστασίας που το στοιχείο μας, τα δεδομένα μας θα λάβουν. Μπορούμε να επιλέξουμε δηλαδή να κρατήσουμε το σύστημα αρχείων συνεπές,

αλλά να επιτρέψουμε τη ζημιά, την αλλοίωση των δεδομένων όσον αφορά το σύστημα αρχείων στην περίπτωση του λανθασμένου κλεισίματος του συστήματος. Εναλλακτικά, μπορούμε να κρατήσουμε το σύστημα αρχείων συνεπές χωρίς να επιτρέψουμε την αλλοίωση των δεδομένων του συστήματος στην περίπτωση του λανθασμένου κλεισίματος του συστήματος. Τέτοια παραδείγματα είναι τα ακόλουθα:

```
cat > /etc/fstab
/dev/hda5 /opt ext3 data=writeback 1 0
```

```
cat > /etc/fstab
/dev/hda5 /opt ext3 data=journal 1 0
```

```
cat > /etc/fstab
/dev/hda5 /opt ext3 data=ordered 1 0
```

➤ **Ταχύτητα:** το Ext3 είναι ταχύτερο από τα άλλα συστήματα αρχείων. Για την βελτίωση της ταχύτητας μπορούμε να επιλέξουμε ένα από τους τρεις τρόπους στους οποίους αναφερθήκαμε και παραπάνω:

-data=writeback (γρήγορη επαναφορά αλλά περιορίζεται η ακεραιότητα των δεδομένων)

-data=ordered (υπάρχει συμβατότητα χρόνου και ακεραιότητας δεδομένων)

-data=journal (χρειάζεται χρόνο για να ανακτηθούν τα δεδομένα ύστερα από ένα λανθασμένο κλείσιμο του συστήματος)

➤ **Εύκολη μετάβαση:** Όσον αφορά την εύκολη μετάβαση τώρα(εύκολη μετατροπή του συστήματος αρχείων),δεν είναι δύσκολο να γίνει αλλαγή από ext2 σε ext3. Υπάρχουν δύο τρόποι να γίνει η μεταφορά από ext2 σε ext3:

Το λειτουργικό σύστημα Red Hat Linux, που είναι ένα από τα προϊόντα της Red Hat(εταιρεία λογισμικού στις ΗΠΑ), θα μεταφέρει τα αρχεία του συστήματός μας αυτόματα όταν αναβαθμίσουμε το σύστημά μας από ext2 σε ext3.

Το πρόγραμμα Tune2fs μπορεί να προσθέσει ένα journal σε ένα υπάρχον ext2 σύστημα αρχείων. Το tune2fs χρησιμοποιείται για την μετατροπή του ext2 σε ext3. Ένα παράδειγμα είναι το ακόλουθο:

```
# /tune2fs -j /dev/hda5
```

Οποιοσδήποτε χρήστης επιθυμεί να μεταβεί από το ext3 στο ext2 κάνει unjournal το σύστημα αρχείων.

➤ Το Ext3 μοιράζεται την δυνατότητα ότι τα νέα χαρακτηριστικά που προστίθενται στο ext2 μπορούν εύκολα να μεταφερθούν στο ext3. Όταν για παράδειγμα κάποιες ιδιότητες προστίθονται στο ext2 είναι εύκολο να προστεθούν και στο Ext3.

➤ Το ext3 χρησιμοποιεί τον ίδιο κώδικα e2fsck με το Ext2 για τη διόρθωση, επαναφορά του συστήματος αρχείων μετά από ένα απροσδόκητο λάθος έχοντας έτσι την ίδια δύναμη με το ext2 ενάντια στην απώλεια δεδομένων. Δηλαδή το E2fsck αναφέρει όλες τις βλάβες λογισμικού ή υλικού που μπορούν να αλλοιώσουν ένα σύστημα καθώς επίσης και όλες τις μεθόδους που χρησιμοποιήθηκαν για την αποκατάστασή τους.

Βέβαια όλα αυτά δεν ισχύουν μονάχα για το Ext3. Πολλά από τα παραπάνω χρησιμοποιούνται και ισχύουν και σε άλλα συστήματα αρχείων. Αλλά σαν σύνολο ισχύουν μόνο στο ext3.[61]

5.6 Από το ext2 στο ext3 και αντίστροφα

Το ext2 έχει τη δυνατότητα να αναβαθμιστεί σε ext3, το οποίο στο μόνο που διαφέρει από το ext2 είναι η προσθήκη του journal. Είναι δηλαδή ίδιο με το ext2 απλά έχει παραπάνω τη δυνατότητα journal. Το ext3 κληρονομεί όλες τις ιδιότητες του ext2. Αυτό σημαίνει ότι κληρονομεί και τα πλεονεκτήματά και τα μειονεκτήματα του ext2(π.χ αριθμός αρχείων και μέγεθος κάθε αρχείου). Έχουμε τη δυνατότητα λοιπόν χωρίς κάποια περίπλοκη διαδικασία να μεταφερθούμε εύκολα από το ext2 στο ext3 και αντίστροφα. Το ext3 μπορεί να εκτελέσει όλες τις εφαρμογές που εκτελεί το ext2. Για να πάμε από το ext2 στο ext3 χρησιμοποιούμε απλά την εντολή:

```
# tune2fs -j <partition-name>
```

Ένα παράδειγμα μετατροπής του ext2 που βρίσκεται στο /dev/hda5 σε Ext3 γίνεται με την παρακάτω εντολή:

```
# /tune2fs -j /dev/hda5
```

Η εντολή `tune2fs -j` δημιουργεί το `ext3 journal` σε ένα υπάρχον `Ext2` σύστημα αρχείων. Εφόσον γίνει η μετατροπή από `Ext2` σε `ext3` πρέπει να ενημερώσουμε το αρχείο `/etc/fstab` έτσι ώστε το σύστημα αρχείων να καταλάβει ότι έγινε η μετατροπή σε `ext3`. Το παρακάτω παράδειγμα δείχνει πως ήταν το σύστημα αρχείων πριν και μετά την αλλαγή από `Ext2` σε `ext3`:

```
Πριν: /dev/hda5 /opt ext2 defaults 1 2
```

```
Μετά: /dev/hda5 /opt ext3 defaults 1 0
```

Το τελευταίο πεδίο του `/etc/fstab` καθορίζει το στάδιο κατά τη διάρκεια της εκκίνησης του υπολογιστή στο οποίο πρέπει να επαληθευθεί η συνέπεια του συστήματος αρχείου από το `fsck` πρόγραμμα. Όταν χρησιμοποιούμε το `ext3` σύστημα αρχείων μπορούμε να θέσουμε αυτό το πεδίο=0. Αυτό σημαίνει ότι το `fsck` πρόγραμμα δε θα ελέγξει ποτέ τη συνέπεια του συστήματος αρχείων, καθώς η συνέπεια του είναι εγγυημένη.[62]

Για τη δημιουργία νέου `ext3` συστήματος αρχείων απλά δίνουμε την εντολή: `mke2fs -j /dev/hdXX`

Κάνοντας αλλαγές μεταξύ ext2-ext3:

Γενικά, πριν προσαρτήσουμε μια διαμέριση σαν `ext3` πρέπει πρώτα να δημιουργήσουμε ένα `Journal` πάνω σε αυτή. Αυτό γίνεται με την εντολή: `tune2fs -j /dev/hdaX`. Στη συνέχεια έχουμε τη δυνατότητα να προσαρτήσουμε το σύστημα αρχείων σαν `ext3`:

```
mount -t ext3 /dev/hdaX /mnt/somewhere
```

Πώς μπορώ τώρα να προσαρτήσω μια `ext3` διαμέριση σε `ext2`; Αν και αυτό συνήθως δε γίνεται, σε περίπτωση που καθίσταται απαραίτητο δίνουμε την παρακάτω εντολή σε μια διαμέριση:

```
tune2fs -O ^has_journal /dev/hdaX
```

Ακολουθώντας για μεγαλύτερη ασφάλεια δίνουμε την εντολή:

```
fsck.ext2 -f /dev/hdaX
```

Στη συνέχεια διαγράφουμε το αρχείο `.journal` [63]

ΚΕΦΑΛΑΙΟ 6 Το σύστημα αρχείων REISERFS

Ιστορική αναδρομή

Εργαλεία διαχείρισεως

Επιδόσεις

Περίληψη

Το κεφάλαιο αυτό μας παρέχει πληροφορίες για το REISERFS και πιο συγκεκριμένα για τα εργαλεία διαχείρισεως που χρησιμοποιεί και τις επιδόσεις-χαρακτηριστικά του.

Το Reiserfs είναι ένα σύστημα αρχείων, το οποίο βασίζεται σε *ισορροπημένους αλγόριθμους δέντρων*¹⁰. Στην ουσία, το Reiserfs μεταχειρίζεται έναν ολόκληρο δίσκο σαν ένα πίνακα βάσεων δεδομένων. Οι ισορροπημένοι αλγόριθμοι δέντρων είναι και γίνονται όλο και πιο αποδοτικοί σε σχέση με τους παραδοσιακούς αλγόριθμους στους οποίους βασίζονται συστήματα αρχείων όπως το ext2fs, nss, hfs, spiralog.

Πιο συγκεκριμένα στα συστήματα αρχείων που μόλις αναφέρθηκαν, η ανάγνωση και το γράψιμο των μεγάλων αρχείων, όπως οι εικόνες CD-ROM ή ISO Images, περιορίζονται συχνά από την ταχύτητα των δίσκων. Αλλά και από την άλλη μεριά, η πρόσβαση στα μικρά αρχεία όπως τα αρχεία εντολών περιορίζεται συχνά από την αποδοτικότητα του συστήματος αρχείων. Κύριος λόγος είναι ότι για να ανοιχτεί ένα αρχείο πρέπει πρώτα από όλα το σύστημα να εντοπίσει αυτό το αρχείο, κάτι που σημαίνει ότι πρέπει να ανοίξει και να διαβάσει όλους τους καταλόγους που περιέχουν τα αρχεία στο δίσκο. Επιπλέον, το σύστημα πρέπει να εξετάσει εάν ο χρήστης έχει τη δικαιοδοσία να έχει πρόσβαση στο αρχείο, κάτι που σημαίνει επιπλέον διαδικασίες. Το σύστημα μπορεί κυριολεκτικά να ξοδέψει περισσότερο χρόνο αποφασίζοντας εάν και που θα επιτρέψει την πρόσβαση, έπειτα να εντοπίσει το αρχείο, και από εκεί να διαβάσει τις πληροφορίες που περιέχονται σε αυτό.

Το Reiserfs από την άλλη μεριά χρησιμοποιεί τα ισορροπημένα δέντρα(περιέχουν τους καταλόγους, τα αρχεία, και τα μεταδεδομένα τα οποία είναι οργανωμένα σε μια δομή δεδομένων) για την εύρεση των αρχείων και την εξασφάλιση της ασφάλειάς τους, ενώ παράλληλα στοχεύει στην μείωση του χρόνου που απαιτείται από τα άλλα συστήματα αρχείων γι' αυτή τη διαδικασία. Για τα μικρά αρχεία, τα δεδομένα του αρχείου μπορούν να καταχωρηθούν κοντά στα μεταδεδομένα του αρχείου, έτσι ώστε και τα δύο να μπορούν να ανακτηθούν εύκολα. Εάν μια εφαρμογή πρέπει να ανοίξει πολλά μικρά αρχεία γρήγορα, αυτή η προσέγγιση βελτιώνει σημαντικά την απόδοση.

Σε ένα παραδοσιακό filesystem όπως ext2, το διάστημα στο δίσκο χωρίζεται σε Block που κυμαίνονται σε μέγεθος από 512 bytes σε 4096 bytes. Εάν το μέγεθος ενός αρχείου συμβεί να είναι διαφορετικό από το μέγεθος των block, ο χώρος θα σπαταληθεί. Παραδείγματος χάριν, ας υποθέσουμε ότι το μέγεθος των block είναι 1024 bytes αλλά πρέπει να καταχωρήσουμε ένα αρχείο που είναι 8195 bytes . Οκτώ block είναι 8192. Τα bytes που

¹⁰ Ισορροπημένοι Αλγόριθμοι Δέντρων: Μια δομή δεδομένων που επιτρέπει την γρήγορη εύρεση, διαγραφή, εισαγωγή πληροφοριών

απομένουν έχουν το δικό τους block, το οποίο είναι σχεδόν άδειο. Το διάστημα που δαπανάται άσκοπα (ως ποσοστό) είναι μικρότερο εάν τα αρχεία είναι μεγαλύτερα και αντιστρόφως.

Στο Reiserfs δεν συμβαίνει κάτι τέτοιο. Το reiserfs στηριζόμενο στη δενδρική δομή παρακολουθεί το μέγεθος των αρχείων που βρίσκονται σε αυτή τη δομή και τα τοποθετεί στο ανάλογο μπλοκ έτσι ώστε να εξοικονομείται περισσότερος χώρος. Ένα παράδειγμα είναι τα μικρά αρχεία, όπου η διαδικασία αυτή μπορεί να εξοικονομήσει πολύ χώρο αποθήκευσης, καθώς τα αρχεία τοποθετούνται κοντά το ένα με τ' άλλο και το σύστημα είναι σε θέση να τα τοποθετεί σε ένα μπλοκ και να τ' ανοίγει και να τα διαβάζει με μια μόνο κίνηση, κάτι το οποίο βελτιώνει την απόδοση.

Παρατηρούμε λοιπόν ότι το Reiserfs είναι ένα σύστημα αρχείων όπως τα υπόλοιπα συστήματα αρχείων xfs, ext3, και JFS, τα οποία εκτελούν τις ίδιες διαδικασίες με έναν όμως διαφορετικό τρόπο λιγότερο ή περισσότερο αποδοτικό. [64]

6.1 Ιστορική αναδρομή

Το Reiserfs ονομάστηκε έτσι από αυτόν που το δημιούργησε και το ανέπτυξε, τον Hans Reiser[65]. Το Reiserfs αρχικά ονομαζόταν "TreeFS" αλλά επειδή υπήρχε μια διαφωνία ως προς το όνομα με άλλους προγραμματιστές πήρε την ονομασία του δημιουργού του, Hans Reiser. Το Reiserfs έχει αυξημένη δημοτικότητα και μπορούν να το χρησιμοποιήσουν πολλά συστήματα για εγκατάσταση. Η υποστήριξη του Reiserfs υπάρχει στους πυρήνες της έκδοσης 2.4 και πιο συγκεκριμένα είναι ενσωματωμένη από την έκδοση του πυρήνα του Linux 2.4.1. [66]

6.2 Εργαλεία διαχείρισεως

Τα εργαλεία διαχείρισεως που περιλαμβάνονται στο reiserfs είναι τα ακόλουθα:

I. mkreiserfs: Δημιουργεί ένα σύστημα αρχείων ReiserFS. Είναι ισοδύναμο με το mke2fs.

II. reiserfsck: Εκτελεί έλεγχο για το σύστημα αρχείων ReiserFS. Είναι ισοδύναμο με το e2fsck στο ext2.

III. resize reiserfs: Χρησιμοποιείται για την αλλαγή μεγέθους σε συστήματα αρχείων Reiserfs.

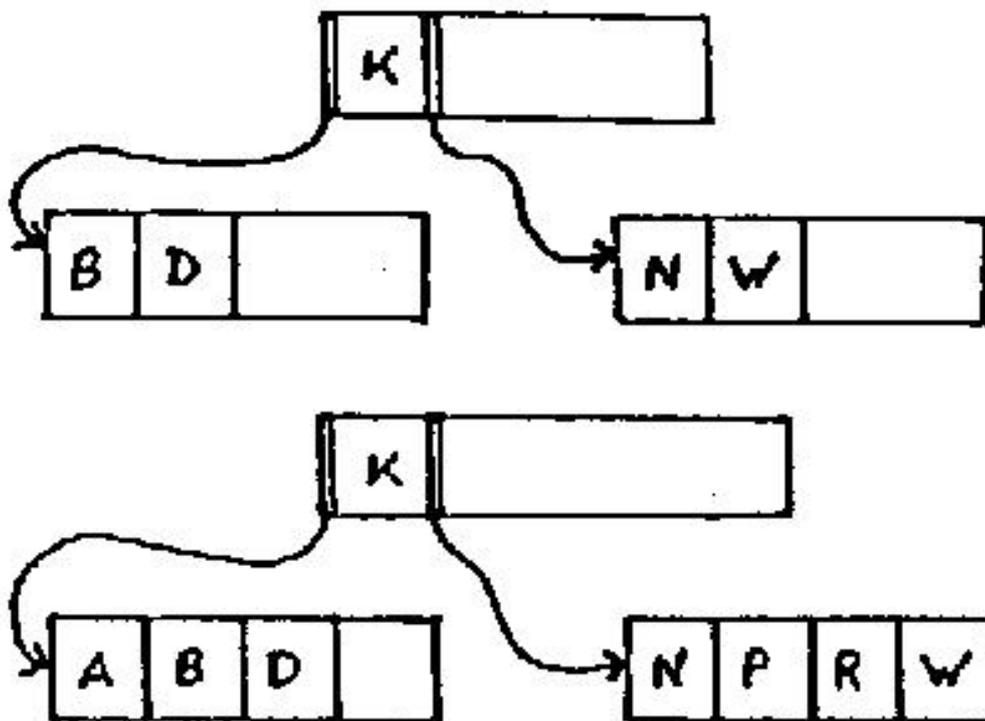
IV. debugreiserfs: Βοηθάει στην επίλυση προβλημάτων με συστήματα αρχείων reiserfs.

V. reiserfstune: Αυτό το εργαλείο χρησιμοποιείται για να ρυθμίσει το journal(αρχείο που καταγράφει τις αλλαγές στο σύστημα αρχείων) του Reiserfs. Μπορεί να αλλάξει το μέγεθος του journal και το μέγεθος της συναλλαγής(οτιδήποτε περικλείει τη μεταφορά δεδομένων όπως για παράδειγμα γράψιμο ή ανάγνωση από αρχεία).[67]

6.3 Επιδόσεις

-Το Reiserfs έχει γρήγορο journaling, το οποίο σημαίνει ότι όταν «κρεμάσει» για παράδειγμα το σύστημα μας δεν χρειάζεται πολύ χρόνο για να κάνει έλεγχο της κατάστασης του συστήματος σε αντίθεση με άλλα συστήματα τα οποία χρειάζονται αρκετό χρόνο για να κάνουν τον έλεγχο. Το reiserfs χρειάζεται μόνο λίγα δευτερόλεπτα. Για παράδειγμα το ReiserFS είναι περίπου *οκτώ έως δεκαπέντε φορές* γρηγορότερο από το ext2 κατά το χειρισμό των αρχείων που είναι μικρότερα από 1K στο μέγεθος.

-Το Reiserfs είναι βασισμένο στα ισορροπημένα δέντρα (σχήμα 6.1). Τα ισορροπημένα δέντρα είναι αποδοτικά σε σχέση με άλλα συστήματα αρχείων. Εάν για παράδειγμα βάλουμε 100.000 αρχεία σε έναν κατάλογο, αυτό σημαίνει ότι θα χρειαστεί αρκετός χρόνος για να γίνει κάτι τέτοιο στα άλλα συστήματα αρχείων σε αντίθεση με το reiserfs



Σχήμα 6.1: Ισοροπημένα δέντρα

- Το Reiserfs είναι πιο αποδοτικό όσον αφορά την εξοικονόμηση χώρου. Αν θέλουμε να εισάγουμε πολλά αρχεία στο σύστημα, το reiserfs θα τοποθετήσει πολλά από αυτά μέσα σε ένα Block. Από την άλλη μεριά άλλα συστήματα αρχείων τοποθετούν κάθε αρχείο σε διαφορετικό Block. Αυτό σημαίνει ότι στα υπόλοιπα συστήματα αρχείων σπαταλάμε περισσότερο χώρο σε σχέση με το Reiserfs όπου τοποθετούμε όλα τα μικρά αρχεία μαζί.
- Οι διαμερίσεις μπορούν να αλλάξουν μέγεθος κατά τη διάρκεια της χρησιμοποίησής τους.

Πιο ειδικά:

Μέγιστος αριθμός αρχείων	$2^{32} \Rightarrow 4 \text{ G}$
Μέγιστος αριθμός αρχείων που μπορεί να έχει ένας κατάλογος	$2^{31} \Rightarrow 2 \text{ G}$
Μέγιστος αριθμός υποκαταλόγων σε έναν κατάλογο	$2^{16} - 1 \text{K} \Rightarrow 64.536 \text{ K}$
Μέγιστος αριθμός συνδέσεων σε ένα	$2^{15} \Rightarrow 32.768 \text{ K}$

αρχείο	
Μέγιστο μέγεθος συστήματος αρχείων	2^{32} (4K) blocks => 17.6 Tbytes
Μέγεθος Block	4KB μέχρι 64KB
Χρόνος επανεκκίνησης μετά από ένα λάθος	Πολύ γρήγορος
Ανάκτηση δεδομένων στην περίπτωση σφάλματος	Πολύ καλή. Η ολοκληρωτική απώλεια δεδομένων είναι πολύ σπάνια

Πίνακας 6.1: Ιδιότητες Reiserfs

Βέβαια το Reiserfs έχει και κάποια μειονεκτήματά. Δύο από αυτά είναι τα παρακάτω:

-Το λογισμικό είναι σχετικά καινούριο.

-Δεν είναι δυνατόν να κάνουμε αλλαγές από το Reiserfs στο ext2 και αντιστρόφως. Δεν μπορούμε να πάμε δηλαδή από το reiserfs στο ext2 όπως γίνεται με το ext3 σύστημα αρχείων. Για να γίνει κάτι τέτοιο πρέπει πρώτα απ' όλα να κρατήσουμε αντίγραφο των δεδομένων μας, να δημιουργήσουμε το νέο σύστημα αρχείων και στη συνέχεια να επαναφέρουμε τα δεδομένα μας.[68]

ΚΕΦΑΛΑΙΟ 7 Το σύστημα αρχείων XFS

Ιστορική αναδρομή

Εργαλεία διαχείρισεως

Επιδόσεις

Περίληψη

Το κεφάλαιο μας δίνει πληροφορίες για το XFS και πιο συγκεκριμένα για τα εργαλεία διαχείρισεως που χρησιμοποιεί και τις επιδόσεις-χαρακτηριστικά του.

Το XFS είναι ένα σύστημα αρχείων "καταγραφής". Αυτό σημαίνει ότι το σύστημα αρχείων καταγράφει τις δραστηριότητές του, κάνοντας κατά συνέπεια την επαναφορά από ένα 'κρέμασμα' του συστήματος πολύ γρηγορότερη. Αυτό ισχύει καθώς στην περίπτωση διακοπής ρεύματος, το σύστημα αρχείων έχει μια εικόνα του τι έκανε εκείνη τη στιγμή. Αυτό σημαίνει ότι το σύστημα αρχείων δεν χρειάζεται να κάνει έλεγχο συνοχής fsck στο Linux. Η λειτουργία καταγραφής είναι περίπου η εξής: όταν πρόκειται να συμβεί μία εγγραφή αρχείου, το σύστημα αρχείων πρώτα γράφει τη λεγόμενη «καταχώρηση πρόθεσης αλλαγής» στο σύστημα. Έπειτα, γίνεται η καθεαυτή εγγραφή του αρχείου. Τέλος, η προηγούμενη καταχώρηση σημειώνεται ως πλήρης και διαγράφεται. Αντί λοιπόν να ελεγχθεί όλο το σύστημα αρχείων, ελέγχεται μόνο η καταχώρηση, και εάν βρεθεί μία καταχώρηση πρόθεσης αλλαγής που δεν έχει ολοκληρωθεί, η δομή των αρχείων την καταχωρεί. Το XFS έχει επίσης και άλλα πλεονεκτήματα να προσφέρει στο Linux. Το XFS είναι ένα 64-bit σύστημα αρχείων, κάτι που σημαίνει ότι μπορεί να υποστηρίξει μεγάλα αρχεία (9 εκατομμύρια terabytes), και ακόμα μεγαλύτερα συστήματα αρχείων (18 εκατομμύρια terabytes). [69]

Ένα XFS σύστημα αρχείων αποτελείται από τουλάχιστον τρία τμήματα: το data section, το log section και το real-time section.

Το data section(τμήμα δεδομένων) περιέχει όλα τα μεταδεδομένα του filesystem (inodes, καταλόγους, blocks), τα αρχεία των χρηστών και την περιοχή log (εάν το log περιλαμβάνεται στο τμήμα δεδομένων καθώς δεν είναι απαραίτητο να υπάρχει). Το τμήμα των δεδομένων διαιρείται σε κατανεμημένες, διαμοιρασμένες ομάδες(allocation groups) ίσου μεγέθους. Ο αριθμός και το μέγεθος των ομάδων επιλέγονται από τα mkfs έτσι ώστε να υπάρχει ένας μικρός αριθμός ίσων σε μέγεθος ομάδων. Ο αριθμός των ομάδων δεν πρέπει να είναι πολύ μεγάλος καθώς κάτι τέτοιο μπορεί να προκαλέσει τα συστήματα αρχείων να χρησιμοποιήσουν πολύ χρόνο από την ΚΜΕ για να πραγματοποιήσουν έλεγχο της συνέπειας των δεδομένων(όσο πιο λίγες είναι ομάδες τόσο πιο γρήγορα πραγματοποιείται ο έλεγχος τους. Το ίδιο συμβαίνει και με τα περιεχόμενα των ομάδων. Όσο λιγότερα δεδομένα έχει μια ομάδα τόσο λιγότερο χρόνο χρειάζεται για να ελεγχθούν).

Το Log section(τμήμα ή περιοχή εάν υπάρχει στο τμήμα δεδομένων) χρησιμοποιείται για να καταχωρήσει τις αλλαγές στα μεταδεδομένα του filesystem. Όταν γίνεται προσάρτηση ενός συστήματος αρχείου και γίνει κάποιο λάθος, διαβάζεται το τμήμα log έτσι ώστε να ολοκληρώσει τις λειτουργίες οι οποίες βρισκόταν σε εξέλιξη εκείνο το διάστημα.

Το τμήμα real-time χρησιμοποιείται για να αποθηκεύσει τα δεδομένα. Κάθε κατανεμημένη ομάδα περιέχει το superblock(το superblock είναι απλά ένα αντίγραφο των δεδομένων), πληροφορίες για blocks και inodes μέσα στην ομάδα, καθώς επίσης και ομάδες δεδομένων για τον εντοπισμό ελεύθερων block και inodes. [70]

7.1 Ιστορική αναδρομή

Το XFS αναπτύχθηκε από την Silicon Graphics στις αρχές τις δεκαετίας του 90. Η SGI κατάλαβε ότι έπρεπε να δημιουργήσει κάτι προκειμένου να αντεπεξέλθει στις νέες απαιτήσεις που αφορούσαν κατά κύριο λόγο την αύξηση της χωρητικότητας των δίσκων. Η SGI αποφάσισε λοιπόν να σχεδιάσει ένα νέο υψηλής απόδοσης σύστημα αρχείων 64 bit, το λεγόμενο XFS. Σκοπός της SGI ήταν ο χειρισμός των νέων εφαρμογών με την παροχή υποστήριξης σε κάποιες βασικές περιοχές. Αυτές οι περιοχές περιλάμβαναν τα συστήματα αρχείων, τη γρήγορη και αξιόπιστη αποκατάσταση σφαλμάτων και τέλος τους μεγάλους καταλόγους και τα αρχεία. Το XFS είναι διαθέσιμο από το 1994 και χρησιμοποιείται μέχρι και σήμερα. Ένα παράδειγμα χρησιμοποίησής του είναι στο Linux.[71]

7.2 Εργαλεία διαχείρισεως

Το XFS, όπως και τα άλλα συστήματα αρχείων, έχει εργαλεία τα οποία και το ενισχύουν. Τα εργαλεία αυτά είναι τα ακόλουθα:

- ❖ Το **mkfs** εργαλείο χρησιμεύει στη δημιουργία του συστήματος αρχείων
- ❖ Τα εργαλεία **dump** και **restore** για να κρατάμε αντίγραφα ασφαλείας. Η εντολή dump εξετάζει τα αρχεία στο σύστημα αρχείων, καθορίζει ποια από αυτά πρέπει να γίνουν Back up και αντιγράφει αυτά τα αρχεία σε ένα συγκεκριμένο δίσκο, ταινία ή άλλο αποθηκευτικό μέσο.

Η εντολή restore εκτελεί την αντίστροφη λειτουργία από την εντολή Dump. Μπορεί να επαναφέρει ένα πλήρες αντίγραφο ασφαλείας ενός συστήματος αρχείων

- ❖ Το εργαλείο **xfs_db** για διόρθωση των προγραμματιστικών λαθών(κάνει debug).
- ❖ Το εργαλείο **xfs_check** για έλεγχο του συστήματος αρχείων
- ❖ Το εργαλείο **xfs_repair** για διόρθωση του συστήματος αρχείων
- ❖ Το εργαλείο **xfs_fsr** για ανασυγκρότηση του XFS

❖ Το εργαλείο `xfs_bmap` δείχνει σε ποιο σημείο του δίσκου βρίσκονται τα αρχεία του συστήματος αρχείων. [72]

7.3 Επιδόσεις

Ο πίνακας 7.1 παρουσιάζει τα χαρακτηριστικά γνωρίσματα του XFS:

Τεχνολογία	64 bit σύστημα αρχείων
Μέγιστο μέγεθος συστήματος αρχείων	18 εκατομμύρια TB
Μέγεθος Block στο Filesystem	4KB
Μέγεθος τομέα δίσκων	512 Bytes
Μνήμη	64 MB
Υψηλή απόδοση	ανεξάρτητα από το μέγεθος
Ανά σύστημα αρχείων	εκατομμύρια αρχεία
Ανά κατάλογο	εκατομμύρια αρχεία
Γρήγορη αποκατάσταση	συστημάτων αρχείων

Πίνακας 7.1: Χαρακτηριστικά γνωρίσματα XFS [73]

ΚΕΦΑΛΑΙΟ 8 Σύγκριση των journaling filesystems

Ταχύτητα

Ασφάλεια

Συμβατότητα

Συμπύεση

Ανάκτηση Δεδομένων

Μέγεθος Κατατμήσεων

Περίληψη

Στο κεφάλαιο αυτό γίνεται σύγκριση των συστημάτων αρχείων που αναφέρθηκαν στα προηγούμενα κεφάλαια για μια πιο λεπτομερειακή προσέγγισή τους.

8.1 Ταχύτητα

FAT:

Το fat32 είναι πιο αποδοτικό και γρήγορο όταν χρησιμοποιεί μεγάλους σκληρούς δίσκους. Το fat16 είναι ικανοποιητικό στην ταχύτητα όταν ο όγκος των αρχείων είναι μικρότερος του 256MB

Το NTFS έχει περίπου την ίδια ταχύτητα με το Fat32 αλλά όσο μεγαλώνει το μέγεθος των δίσκων το χάσμα αυξάνει. Η απόδοση NTFS είναι υψηλότερη εάν ένας κατάλογος περιέχει χιλιάδες αρχεία. Σε αυτήν την περίπτωση η ταχύτητα του NTFS είναι δέκα φορές μεγαλύτερη από του FAT και FAT32 . [74]

Παρακάτω βλέπουμε έναν αντιπροσωπευτικό πίνακα που συγκρίνει τις επιδόσεις κάποιων από τα συστήματα αρχείων που αναφέρθηκαν στα παραπάνω κεφάλαια. Στον πίνακα 8.1 βλέπουμε πόσο γρήγορα επιτεύχθηκε η πρόσβαση σε ένα αρχείο, η δημιουργία, ανάγνωση και διαγραφή ενός αρχείου.

	Sequential Create (Διαδοχική δημιουργία)							Random Create (Τυχαία δημιουργία)						
	Αρχεία	Δημιουργία		Ανάγνωση		Διαγραφή		Δημιουργία		Ανάγνωση		Διαγραφή		
		/ sec	% CPU	/ sec	% CPU	/ sec	% CPU	/ sec	% CPU	/ sec	% CPU	/ sec	% CPU	
EXT2	16	94	99	278	99	492	97	95	99	284	100	93	41	
EXT3	16	89	98	274	100	458	96	93	99	288	99	97	45	
XFS	16	92	99	251	96	436	98	91	99	311	99	90	41	
REISERFS	16	1307	100	8963	100	1914	99	1245	99	9316	100	1725	100	

Πίνακας 8.1: Σύγκριση συστημάτων αρχείων ως προς τη ταχύτητα

8.2 Ασφάλεια

Συστήματα αρχείων	Παροχή Ασφάλειας
FAT	-
NTFS	Κάθε χρήστης έχει δικαίωμα πρόσβασης σε συγκεκριμένα αρχεία του συστήματος
EXT2 - EXT3	Δεδομένων(δεν χάνονται τα δεδομένα) και γρήγορη ανάκτησή τους μετά από ένα σφάλμα του συστήματος.
REISERFS	Δεδομένων αλλά όχι απόλυτη, δηλαδή 100%. Το σύστημα για ασφάλεια εξετάζει να δει αν ο χρήστης έχει δικαιώματα πρόσβασης στο αρχείο. Το Reiserfs εγγυάται την ακεραιότητα των δεδομένων αλλά για μεγαλύτερη ασφάλεια χρησιμοποιούνται κρυπτογραφημένα συστήματα αρχείων για την κρυπτογράφηση των δεδομένων
XFS	Παρέχει λίστες ελέγχου πρόσβασης των αρχείων όπως επίσης και δυνατότητα για αντίγραφα ασφαλείας

Πίνακας 8.2: Σύγκριση συστημάτων αρχείων ως προς την ασφάλεια [76]

8.3 Συμβατότητα

Συστήματα Αρχείων	Συμβατότητα
FAT	Το Fat είναι συμβατό με το NTFS
NTFS	-
EXT2 - EXT3	Το ext2 είναι συμβατό με το Ext3 και αντίστροφα
REISERFS	-
XFS	-

Πίνακας 8.3: Σύγκριση συστημάτων αρχείων ως προς τη συμβατότητα [77]

8.4 Συμπίεση

Συστήματα Αρχείων	Συμπίεση
FAT	Υποστηρίζει συμπίεση δίσκων
NTFS	-
EXT2	Το e2compr είναι μια μέθοδος συμπίεσης δίσκων όπως επίσης και συμπίεσης-αποσυμπίεσης αρχείων. Συμπιέζει μόνο τα κανονικά αρχεία. Τα αρχεία: superbloc, inodes, directory files κτλ δεν συμπιέζονται(κυρίως για λόγους ασφάλειας). Οι κατάλογοι μπορούν επίσης να συμπιεστούν, κάτι που σημαίνει ότι κάθε νέο δημιουργημένο αρχείο στον κατάλογο θα συμπιεστεί αυτόματα
EXT3	Το ext3 είναι επέκταση του ext2 και στηρίζεται στα χαρακτηριστικά και τις ιδιότητες του ext2. Πιο συγκεκριμένα το ext3 εφαρμόζει τα χαρακτηριστικά του ext2. Ένα από αυτά λοιπόν είναι και η εφαρμογή της συμπίεσης αρχείων

REISERFS	-
XFS	Υποστηρίζει γρήγορη αλλά μειωμένη απόδοση ως προς τη συμπίεση αρχείων

Πίνακας 8.4: Σύγκριση συστημάτων αρχείων ως προς τη συμπίεση [78]

8.5 Ανάκτηση δεδομένων

Συστήματα Αρχείων	Ανάκτηση Δεδομένων
FAT	Όταν τα δεδομένα μας αποθηκευτούν στο δίσκο, ο υπολογιστής χρειάζεται ένα τρόπο εύρεσης των δεδομένων και ανάκτησής τους όταν είναι απαραίτητο. Αυτό γίνεται με το FAT. Το FAT λέει στον υπολογιστή που θα βρει κάθε αρχείο
NTFS	Το NTFS χρησιμοποιεί έναν πίνακα για να αποθηκεύσει τα αρχεία στον σκληρό δίσκο. Εάν κάτι συμβεί γράφοντας στο αρχείο, οι χαμένες πληροφορίες μπορούν να ανακτηθούν αυτόματα.
EXT2 – EXT3	Χρησιμοποιούν κόμβους-δ
REISERFS	Το reiserfs χρησιμοποιεί τα ισορροπημένα δέντρα(balanced trees) για να βελτιώσει τη διαδικασία εύρεσης αρχείων και ανάκτησης των μεταδεδομένων. Για εξαιρετικά μικρά αρχεία, τα δεδομένα μπορούν να αποθηκευτούν κοντά στα μεταδεδομένα, έτσι ώστε να ανακτηθούν μαζί με αυτά. Αυτό έχει σαν αποτέλεσμα τη βελτίωση της απόδοσης
XFS	Το xfs χρησιμοποιεί ισορροπημένα δέντρα για την εύρεση καταλόγων, αρχείων, εντοπισμού ελεύθερου χώρου κτλ. Αυτό έχει σαν αποτέλεσμα η ανάγνωση των

	συστημάτων αρχείων και η ανάκτηση πληροφοριών να γίνεται μέσα σε πολύ μικρό χρονικό διάστημα
--	--

Πίνακας 8.5: Σύγκριση συστημάτων αρχείων ως προς την ανάκτηση δεδομένων [79]

8.6 Μέγεθος κατατμήσεων

FAT: Μέσα στο σκληρό δίσκο υπάρχουν clusters(συστοιχίες). Όσο μεγαλύτερη είναι η διαμέριση, τόσο μεγαλύτερο είναι το μέγεθος των συστοιχιών. Οι πίνακες 8.6 και 8.7 δείχνουν τη διαφορά που υπάρχει μεταξύ του Fat16 και του Fat32 ως προς το μέγεθος της διαμέρισης και το μέγεθος των συστοιχιών:

Fat16	
Μέγεθος συστοιχιών	Μέγεθος διαμέρισης
2KB	128MB
4KB	256MB
8KB	512MB
16KB	1GB
32KB	2GB

Πίνακας 8.6: Μεγέθη διαμέρισης και συστοιχιών FAT16

Fat32	
Μέγεθος συστοιχιών	Μέγεθος διαμέρισης
2KB	<260MB
4KB	260MB-8GB
8KB	8GB-16GB
16KB	16GB-32GB
32KB	32GB<

Πίνακας 8.7: Μεγέθη διαμέρισης και συστοιχιών FAT32

Βλέπουμε λοιπόν ότι το μεγαλύτερο μέγεθος διαμέρισης στο fat16 είναι 2 GB ενώ στο fat32 είναι πάνω από 32GB μέχρι και 128GB. Επίσης, το μέγεθος των συστοιχιών στο fat16 για 2GB διαμέριση είναι 32KB ενώ για διαμέριση μεταξύ 260MB και 8GB στο fat32 το μέγεθος των συστοιχιών είναι 4KB σώζοντας 8 φορές περισσότερο χώρο συστοιχιών από μια διαμέριση που είναι κάτω από fat16.[80]

Στον πίνακα 8.8 βλέπουμε το μέγεθος της διαμέρισης για το NTFS για διάφορες συστοιχίες

Μέγεθος συστοιχίας	Ελάχιστο μέγεθος διαμέρισης	Μέγιστο μέγεθος διαμέρισης
0.5 KB	0 GB	0.5 GB
1 KB	0.5 GB	1 GB
2 KB	1 GB	2 GB
4 KB	2 GB	4 GB
8 KB	4 GB	8GB
16 KB	8 GB	16 GB
32 KB	16 GB	32 GB
64 KB	32 GB	64 GB

Πίνακας 8.8: Διαμέριση NTFS[81]

Συστήματα Αρχείων	Μέγεθος Κατατμήσεων
EXT2	Το Ext2 υποστηρίζει διαμερίσεις πάνω από 4 Terabytes σε μέγεθος ενώ ένα απλό αρχείο μπορεί να είναι πάνω από 2 Gbytes
EXT3	Μια Ext3 διαμέριση μπορεί να προσαρτηθεί από ένα παλιότερο πυρήνα που δεν υποστηρίζει Ext3. Είναι όπως η Ext2 διαμέριση. Πιο συγκεκριμένα η Ext2 διαμέριση μπορεί να αναβαθμιστεί σε Ext3 χωρίς να υπάρξει απώλεια δεδομένων.
REISERFS	Το επιθυμητό μέγεθος των διαμερίσεων στο reiserfs πρέπει να είναι πάνω από 32MB.
XFS	Το Xfs είναι ένα 64-μπιτο σύστημα αρχείων, κάτι που σημαίνει ότι οι περιορισμοί του μεγέθους απέρχονται σε εκατομμύρια Terabytes

Πίνακας 8.9: Σύγκριση συστημάτων αρχείων ως προς το μέγεθος κατάτμησης [82]

Συμπεράσματα – Προτάσεις

Η Πτυχιακή Εργασία που παρουσιάστηκε είχε σαν αντικείμενο την μελέτη των Journaling Filesystems – Συστήματα αρχείων, τα οποία αποτελούν το μέρος του λειτουργικού συστήματος που ασχολείται με τα αρχεία. Ένα σύστημα αρχείων αποτελεί τη μέθοδο ονομασίας, αποθήκευσης και οργάνωσης των αρχείων στον υπολογιστή μας. Το σύστημα αρχείων βελτιώνει την ασφάλεια, επιτρέποντας μας να κρυπτογραφούμε αρχεία και φακέλους και να περιορίζουμε την πρόσβαση σε αρχεία.

Αυτή η εργασία παρουσίασε κάποιες βασικές έννοιες γύρω από τα αποθηκευτικά μέσα, τους τρόπους προσπελάσεως και υλοποιήσεως συστημάτων αρχείων. Επίσης, ανέφερε διάφορων ειδών αρχεία όπως τα αρχεία χαρακτήρα και αρχεία μπλοκ, την προσθήκη διαφορετικών συστημάτων αρχείων σε ένα σύστημα.

Τέλος, αναφέρθηκαν τα συστήματα αρχείων FAT, EXT2, EXT3, REISERFS, XFS και πιο συγκεκριμένα παρουσιάστηκαν τα εργαλεία διαχειρίσεως που χρησιμοποιήθηκαν, οι επιδόσεις τους, η σύγκριση μεταξύ τους και άλλα θέματα που είχαν σαν σκοπό την βαθύτερη ενημέρωση μας γύρω από τα λειτουργικά συστήματα όσον αφορά τα Journaling Filesystems αυτή τη περίοδο.

Ηλεκτρονική Βιβλιογραφία:

1. http://www.agoraste.gr/business_forum/art_ststore.php
2. <http://vrl.thestep.gr/tele/kef13.htm>
3. <http://www.medialab.ntua.gr/multinew/left.htm>,
<http://www.medialab.ntua.gr/multi/chap6.htm>
4. <http://www.microsoft.com/hellas/windows/windowsxp/pro/evaluation/overviews/filesystem.asp>
5. A.S. TANENBAUM: Σύγχρονα λειτουργικά συστήματα. Τόμος Α' Αθήνα 1993(Σελ.207)
6. A.S. TANENBAUM: Σύγχρονα λειτουργικά συστήματα. Τόμος Α' Αθήνα 1993(Σελ.223),
<http://cgi.di.uoa.gr/~os1/syllabus/fs/>
7. <http://www.ukoln.ac.uk/metadata/>,
<http://www.fgdc.gov/metadata/metadata.html>
8. <http://www.cs.adfa.oz.au/teaching/studinfo/osrts/Lectures/node108.html>
9. <http://docsrv.caldera.com:8457/en/SecureProg/files.html>
10. <http://mm.iit.uni-miskolc.hu/Data/texts/Linux/SAG/node84.html>,
<http://web.mit.edu/tytso/www/linux/ext2intro.html>
11. <http://www.tldp.org/LDP/tlk/fs/filesystem.html>
12. A.S TANENBAUM: Σύγχρονα Λειτουργικά Συστήματα, Τόμος Α', Αθήνα 1993, Σελ.23,204,401
13. A.S TANENBAUM: Σύγχρονα Λειτουργικά Συστήματα, Τόμος Α', Αθήνα 1993, Σελ.214
<http://www.spinellis.gr/os1/files/mmap.htm>
14. <http://www.yggdrasil.com/bible/bible-src/khg-0.6/khg/node23.html>.
15. <http://www.tldp.org/HOWTO/mini/Hard-Disk-Upgrade/mount.html>
16. <http://www.science.unitn.it/~fiorella/guidelinux/tlk/node94.html>
17. <http://www.techteam.gr/Leksiko/f.html>
18. http://www.it.uom.gr/mycomputer/storage/file_sys/info/dom_diam.html
19. <http://members.aol.com/mbs1058/waste.html>.
20. http://www.pcguides.com/ref/hdd/file/part_Sizes.htm
21. http://www.it.uom.gr/mycomputer/storage/file_sys/info/general.html#FAT32
22. http://www.it.uom.gr/mycomputer/storage/file_sys/info/m_s_fat.html
23. <http://support.microsoft.com/default.aspx?scid=KB;EN-US;q154997>
24. http://www.it.uom.gr/mycomputer/storage/file_sys/info/general.html,
http://www.pcguides.com/ref/hdd/file/file_VFAT.htm
25. http://www.it.uom.gr/mycomputer/storage/file_sys/info/general.html
26. <http://www.pcguides.com/ref/hdd/file/ntfs/index-c.html>,
<http://support.microsoft.com/default.aspx?scid=KB;EN-US;q100108>,
<http://www.microsoft.com/hellas/windows/windowsxp/pro/evaluation/overviews/filesystem.asp>
<http://www.microsoft.com/hwdev/tech/storage/ntfs-preinstall.asp>
27. <http://www.pcguides.com/ref/hdd/file/ntfs/files.htm>
28. <http://www.pcguides.com/ref/hdd/file/ntfs/sec.htm>
29. http://www.garykessler.net/library/nt_crypto.html
30. <http://www.pcguides.com/ref/hdd/file/ntfs/archFiles-c.html>
31. <http://www.extremetech.com/article/0,3396,s=1027&a=19913&app=1&ap=2,00.asp>
32. mike.davidson@wayne.kiz.in.us
33. Remy.Card@linux.org
34. <http://e2fsprogs.sourceforge.net/ext2intro.html#section:ext2fs>
35. <http://e2fsprogs.sourceforge.net/ext2intro.html>
36. http://panic.et.tudelft.nl/debian/handleiding/en/Noel_Harte/Section-2.5.html

37. torvalds@transmeta.com,
<http://www.tuxedo.org/~esr/faqs/linux/>
38. <http://e2fsprogs.sourceforge.net/ext2intro.html>
39. <http://www.freeos.com/articles/3912/>
40. <http://qslinux.org/docs/snmc/ext2fs/>
41. <http://weigand.home.texas.net/efs.html>
42. <http://www.nsa.gov/selinux/faq.html>
43. <http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/sysv-ipc.html>
44. <http://trustees.sourceforge.net/>
45. http://magaz.hellug.gr/13/04_lin_for_dummies2-2.html
46. <http://mm.iit.uni-miskolc.hu/Data/texts/Linux/SAG/node88.html>
47. http://216.239.51.100/search?q=cache:P47LLi8qnRoC:cvs.bofh.asn.au/e2compr/manual-0.4/e2compr-0.4_3.html+compression%2Bext2&hl=el&ie=UTF8
48. <http://www-106.ibm.com/developerworks/library/l-embdev.html?dwzone=wireless>
49. <http://e2fsprogs.sourceforge.net/ext2intro.html>
50. <http://www.extremetech.com/article2/0,3973,51322,00.asp>
51. <http://www-106.ibm.com/developerworks/library/l-embdev.html?dwzone=wireless>
52. <http://www.linuxsymposium.org/2002/keynote.php>
53. <http://www.linuxplanet.com/linuxplanet/reports/4136/1/>
54. <http://www.linuxplanet.com/linuxplanet/reports/4136/5/>
55. <http://bulmalug.net/body.phtml?nIdNoticia=1154&nIdPage=7>
56. <http://www.extremetech.com/article2/0,3973,51322,00.asp>
57. <http://www.europe.redhat.com/products/linux/featuresFunctionsBenefits.php3>
58. <http://www.beta.redhat.com/index.cgi?action=ext3>
59. <http://acl.bestbits.at/download.html#E2fsprogs>
60. <http://www.linuxworld.com.au/article.php3?aid=214&tid=8>
61. <http://www.redhat.com/support/wpapers/redhat/ext3/>,
<http://www.linuxhq.com/kernel/v2.0/doc/initrd.txt.html>,
<http://www.redhat.com/support/wpapers/redhat/ext3/#advantages>,
<http://bulmalug.net/body.phtml?nIdNoticia=1154&nIdPage=7>,
<http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/custom-guide/buildkernel-initrd.html>,
http://www.humbug.org.au/talks/fstab/fstab_structure.html
62. <http://www.linuxplanet.com/linuxplanet/reports/4136/4/>
63. <http://people.spoiled.org/jha/ext3-faq.html>
64. http://www.icewalkers.com/softlib/app/app_01171.html,
<http://www.ukuug.org/events/linux2000/speaker-HR.shtml>,
<http://www.linuxplanet.com/linuxplanet/tutorials/2926/4/>
http://certcities.com/certs/linux_unix/columns/story.asp?EditorialsID=8,
<http://www-106.ibm.com/developerworks/library/l-fs.html>
65. http://www.idiom.com/~beverly/hans_resume.html
66. <http://steven.haryan.to/interview-hans-reiser.html>,
<http://www.debianplanet.org/node.php?id=386>,
67. http://networking.earthweb.com/netos/article/0,,12083_625421,00.html,
http://linuxcommand.org/man_pages/mkreiserfs8.html,
http://linuxcommand.org/man_pages/reiserfsck8.html,
http://www.namesys.com/resize_reiserfs.html,
<http://www.namesys.com/debugreiserfs.html>,
<http://www.namesys.com/reiserfstune.html>.

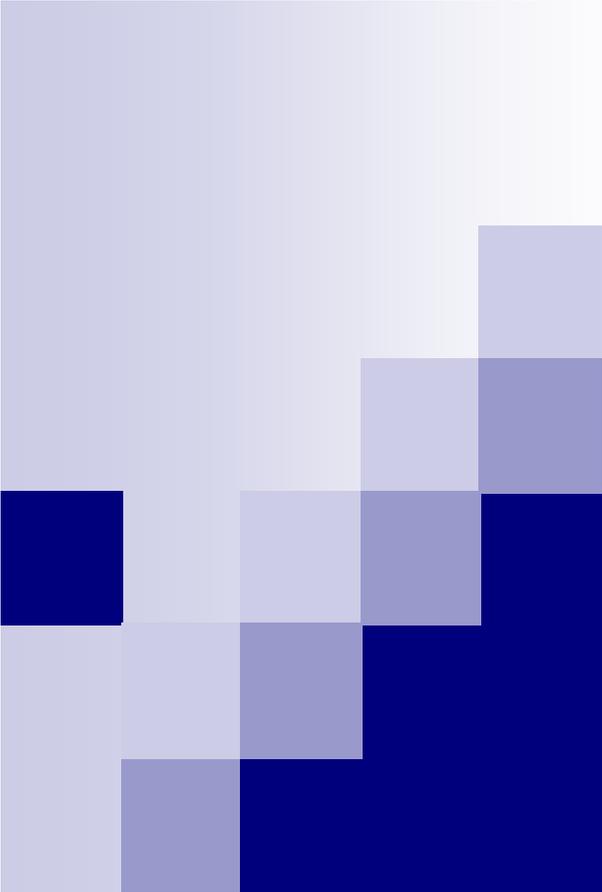
-
68. <http://www.namesys.com/>,
http://networking.earthweb.com/netos/article/0,,12083_625421,00.html,
<http://www.extremetech.com/article2/0,3973,51322,00.asp>,
<http://escher.cs.ucdavis.edu:1024/ECS165A/lectures/lect13.html>
69. http://magaz.hellug.gr/17/03_xfs.html,
<http://www.sgi.com/software/xfs/>,
http://www.linux-mag.com/2000-08/journaling_04.html
70. [http://conx.bu.edu/cgi-bin/perl/manscript?xfs\(4\)](http://conx.bu.edu/cgi-bin/perl/manscript?xfs(4))
71. <http://www-106.ibm.com/developerworks/linux/library/l-fs9.html>,
http://networking.earthweb.com/netos/article/0,,12284_623661,00.html.
72. http://networking.earthweb.com/netos/article/0,,12284_623661,00.html
73. <http://oss.sgi.com/projects/xfs/features.html>,
http://networking.earthweb.com/netos/article/0,,12284_623661,00.html
74. <http://www.freeaccess.com.au/pcproj/spring!/SPRING.HTM>,
<http://www.swynk.com/friends/zandri/70-270-6-1.asp>,
<http://www.freeaccess.com.au/pcproj/spring!/SPRING.HTM>,
<http://checknow.virtualave.net/catg/hard/hdd/1hdd1.html>,
<http://www.digit-life.com/articles/ntfs/index3.html>
75. <http://www.linuxgazette.com/issue68/dellomodarme.html>
76. <http://securityresponse.symantec.com/avcenter/venc/data/trojan.fatkill.html>,
<http://www.informationweek.com/837/langa.htm>,
support.microsoft.com/support/kb/articles/q80/5/70.asp,
<http://shop.barnesandnoble.com/booksearch/isbninquiry.asp?btob=Y&isbn=0471978442>,
http://education.hp.com/itrc/news/micro_art1.html,
<http://online.securityfocus.com/infocus/1407>,
http://www.linuxsecurity.com/articles/host_security_article-1966.html,
<http://www.luv.asn.au/overheads/security/files.html>,
<http://www.theorygroup.com/Theory/FAQ/Secure-Programs-HOWTO-3.html>,
<http://www.nsa.gov/selinux/faq.html>,
<http://www.insecure.org/tools.html>,
<http://www.cbbrowne.com/info/secunix.html>,
<http://rr.sans.org/linux/SELinux.php>,
<http://gd.tuwien.ac.at/opsys/linux/lasg-www/file/>,
[http://www.amazon.co.uk/exec/obidos/tg/stores/detail/reviews/- /software /B00005RCWH /ref=ed_ec_h_cs_6_1/](http://www.amazon.co.uk/exec/obidos/tg/stores/detail/reviews/-/software/B00005RCWH/ref=ed_ec_h_cs_6_1/),
<http://www.darpa.mil/ipto/psum2001/M133-0.html>,
<http://lwn.net/2001/0524/security.php3>,
<http://lwn.net/2000/0420/security.php3>,
http://www.msclinux.com/software/msclinux/release_guide_v2002mar.html,
http://networking.earthweb.com/netsecur/article/0,,12084_625191,00.html,
<http://www.pcguides.com/ref/hdd/file/ntfs/sec-c.html>
77. http://www.internetguideandmore.com/Windows_95-98_Glossary.htm,
<http://www.project9.com/fat32/>,
<http://www.computerhope.com/fat32.htm>,
<http://evms.sourceforge.net/faq.html>,
<http://lwn.net/2001/0412/a/index-format.php3>,
<http://www.nsa.gov/selinux/0faq.html>,
<http://www.linuxplanet.com/linuxplanet/reports/4136/6/>,
<http://www.redhat.com/support/wpapers/redhat/ext3/why.html>,

- <http://linux.org.mt/article/filesystems>,
<http://intimate.handhelds.org/news.shtml>,
<http://lwn.net/2001/0301/kernel.php3>,
<http://www-106.ibm.com/developerworks/linux/library/l-fs7/>,
<http://www.sgi.com/software/xfs/overview.html>,
<http://www.uoks.uj.edu.pl/resources/flugor/IRIX/xfs-whitepaper.html>,
<http://www.uoks.uj.edu.pl/resources/flugor/IRIX/XFS.html>,
http://www.experts-exchange.com/Operating_Systems/Q_20281807.html,
<http://builder.com.com/article.jhtml?id=r00320020528tcc01.htm>
78. <http://www.comptechdoc.org/os/windows/win2k/win2kfiles.html>,
<http://www.linuxrouter.org/archives/linux-router-devel/2001-02-01/msg00011.html>,
http://cvs.bofh.asn.au/e2compr/manual-0.4/e2compr-0.4_32.html,
<http://www.linuxgazette.com/issue18/e2compr.html>,
<https://lists.dulug.duke.edu/pipermail/dulug/2000-November/001864.html>,
http://www.freeswan.org/freeswan_trees/freeswan-1.95/doc/performance.html,
<http://mail.nl.linux.org/linux-crypto/2001-06/msg00008.html>,
<http://playground.sun.com/pub/nfsv4/nfsv4-wg-archive/2001/0078.html>,
<https://lists.dulug.duke.edu/pipermail/dulug/2000-November/001867.html>,
http://bfm.org/~aa089/c_ntfs.htm
79. <http://www.infopackets.com/freenewsarticles/backup.htm>,
<http://www.pcbroker.co.uk/dataremoval.htm>,
http://www.simsmith5.fsnet.co.uk/Hardware/Drives_hard.htm,
<http://www.seniorcentre.ca/dc/components/storage.html>,
http://securityuniversity.net/courses_reaction_aft.htm,
<http://parlweb.parl.clemson.edu/pvfs/desc.html>,
<http://www.linuxplanet.com/linuxplanet/tutorials/2926/4/>,
<http://freshmeat.net/articles/view/305/>,
<http://www.geocrawler.com/archives/3/2311/2000/11/50/4705605/>,
http://networking.earthweb.com/netos/article/0,,12284_623661,00.html,
<http://lasr.cs.ucla.edu/awang/papers/usenix2002/camera2.htm>,
<http://acl.bestbits.at/pipermail/acl-devel/2001-February/000509.html>,
<http://support.packardbell-europe.com/support/faq/article/TPC01447.asp?c=ap>
80. http://www.activewin.com/faq/faq_2.shtml,
<http://www.scitrav.com/labtips/diskpart.htm>,
<http://www.firmware.com/support/bios/cluster.htm>,
<http://www.dewassoc.com/support/win98/fat.htm>
81. <http://web.ukonline.co.uk/cook/Clustsize.htm>
82. <http://linux.org.mt/article/filesystems>,
<http://cambuca.ldhs.cetuc.puc-rio.br/>,
<http://www.linuxplanet.com/linuxplanet/tutorials/2926/4/>,
<http://lists.suse.com/archive/suse-linux-e/2001-Jul/0617.html>,
<http://www.jsifaq.com/SUBB/tip0800/rh0862.htm>

Βιβλία-Περιοδικά-Σημειώσεις:

- ✓ Σύγχρονα Λειτουργικά Συστήματα - TANENBAUM A.S., Εκδόσεις Παπασωτηρίου Τόμος Α', 1993
- ✓ Σύγχρονα Λειτουργικά Συστήματα - TANENBAUM A.S., Εκδόσεις Παπασωτηρίου Τόμος Β', 1994

- ✓ Αρχές Λειτουργικών Συστημάτων - Silberschatz Abraham, Εκδόσεις Παπασωτηρίου 5^η Έκδοση, 1999
- ✓ Science & Engineering - Περιοδικό Linux Journal, Έκδοση 51, Ιούλιος 1998
- ✓ Σημειώσεις Λειτουργικών - Στεργίου Ελευθέριος, ΤΕΙ ΑΡΤΑΣ 1999



Θεωρητική Μελέτη των “Journaling Filesystems”

Της σπουδάστριας
Ποτσίκα Πολυξένης

❖ **Βασικές Έννοιες**

- Αποθηκευτικά Μέσα
- Τι είναι τα συστήματα αρχείων και τρόποι προσπέλασής τους
- Τρόποι υλοποίησης συστημάτων αρχείων
- Μεταδεδομένα συστημάτων αρχείων

❖ **VFS(Virtual Filesystem)**

- Προσάρτηση συστήματος αρχείων
- Είδη αρχείων
- Συστήματα αρχείων στο Linux

❖ **FAT(File Allocation Table)**

- Συστήματα αρχείων FAT16 - FAT32 – VFAT - NTFS

❖ **EXT2 - EXT3 - REISERFS - XFS**

- Ιστορική παρουσίασή
- Εργαλεία Διαχείρισεως
- Επιδόσεις

❖ **Σύγκριση συστημάτων αρχείων**

- Ταχύτητα – Ασφάλεια – Συμβατότητα – Συμπύεση – Ανάκτηση Δεδομένων – Μέγεθος Κατάτμησης

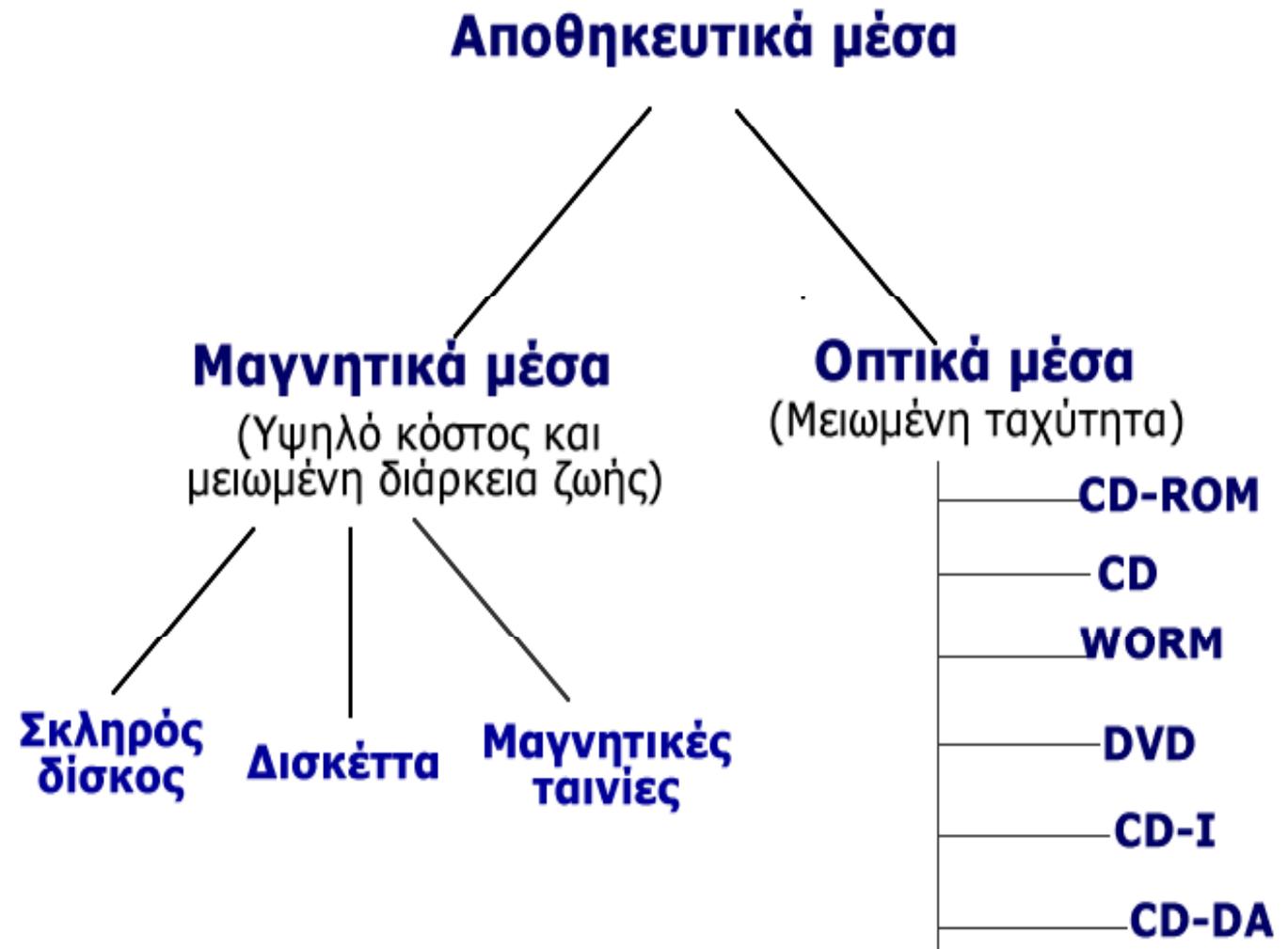
Παράγοντες για την επιλογή αποθηκευτικού μέσου

- Ποσότητα που θα αποθηκευτεί και απαιτούμενος χρόνος προσπέλασης δεδομένων

- Είδος πληροφορίας που θα αποθηκευτεί: κείμενο, εικόνα ήχος

- Μεταβλητότητα πληροφορίας

- Ασφάλεια πληροφοριών



Ορισμός συστήματος αρχείων

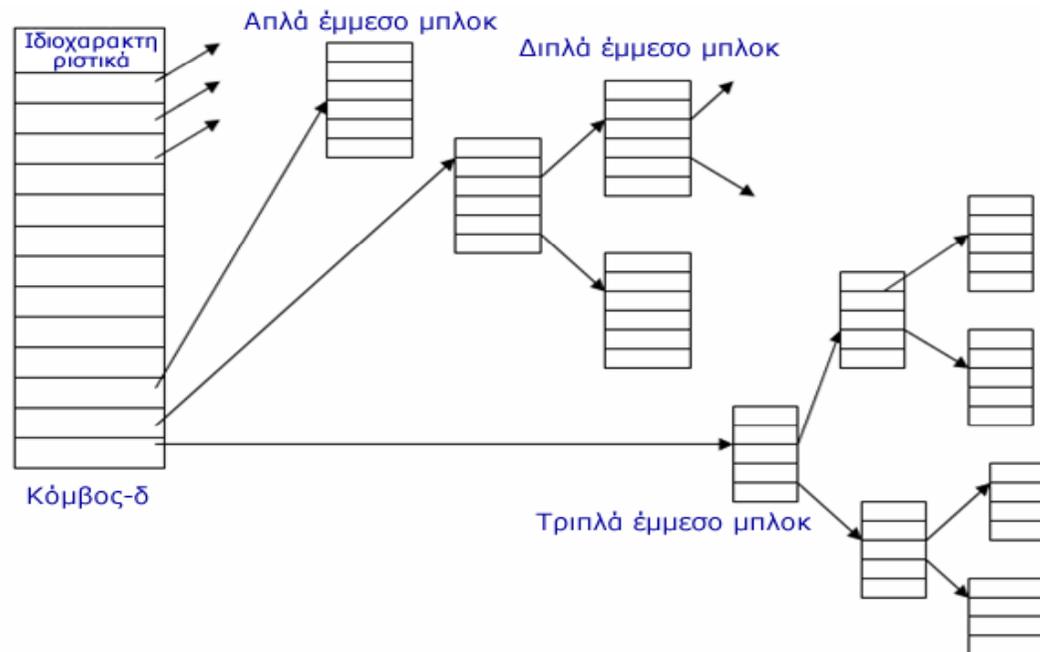
Ένα σύστημα αρχείων περιλαμβάνει μεθόδους:

- ονομασίας των αρχείων
- αποθήκευσης των αρχείων
- οργάνωσης των αρχείων
- ασφάλειας των αρχείων
- κρυπτογράφησης των αρχείων

Τρόποι Προσπέλασης συστήματος αρχείων

- Σειριακή Προσπέλαση: Μια διαδικασία διαβάζει όλα τα Bytes ή τις εγγραφές ενός αρχείου στη σειρά, ξεκινώντας από την αρχή. Δεν μπορεί όμως να παραλείπει μερικά ή να διαβάζει εκτός σειράς
- Τυχαία Προσπέλαση: Τα Bytes και οι εγγραφές ενός αρχείου διαβάζονται χωρίς συγκεκριμένη σειρά. Είναι απαραίτητα για εφαρμογές όπως βάσεις δεδομένων.

- ✓ **Συνεχής Κατανομή:** αποθηκεύουμε κάθε αρχείο σαν ένα συνεχόμενο μπλοκ δεδομένων πάνω στο δίσκο.
- ✓ **Κατανομή συνδεδεμένης λίστας:** τα αρχεία πρέπει να τα διατηρούμε ως μια συνδεδεμένη λίστα από μπλοκ δίσκου. Το πρώτο τμήμα του κάθε μπλοκ χρησιμοποιείται ως δείκτης στο επόμενο. Το υπόλοιπο του μπλοκ είναι δεδομένα.
- ✓ **Κατανομή συνδεδεμένης λίστας με χρήση δείκτη:** παίρνουμε τη λέξη με το δείκτη από κάθε μπλοκ και την τοποθετούμε σε έναν πίνακα στη μνήμη.
- ✓ **Κόμβος-δ:** συσχετίζουμε κάθε αρχείο με έναν μικρό πίνακα ο οποίος περιέχει τα χαρακτηριστικά και τις διευθύνσεις των αρχείων στο δίσκο(σχήμα 1.1).



Σχήμα 1.1

Τα μεταδεδομένα ή metadata δίνουν πληροφορίες για τα ίδια τα δεδομένα, όπως για παράδειγμα: **το όνομα και τις ιδιότητες κάθε αρχείου και κάθε καταλόγου**

Τα περισσότερα συστήματα αρχείων καταχωρούν ιδιότητες όπως:

- Ιδιοκτήτης -προσδιορίζει τον ιδιοκτήτη και το group του filesystem
- permission bits-Bits δικαιωμάτων
- timestamps- χρόνοι πρόσβασης και τροποποίησης

- ❖ Το VFS γνωρίζει τα συστήματα αρχείων που υπάρχουν στο σύστημα χρησιμοποιώντας πίνακα.
- ❖ Κάθε είσοδος σε αυτόν τον πίνακα περιγράφει ένα σύστημα αρχείων περιέχοντας:
 - το όνομα του συστήματος αρχείων
 - μία λίστα από λειτουργίες στο σύστημα αρχείων έτσι ώστε να μπορούμε να χρησιμοποιούμε τις ίδιες λειτουργίες I/O για κάθε σύστημα αρχείων
- ❖ Το VFS επιτρέπει την προσάρτηση πολλών διαφορετικών συστημάτων αρχείων ταυτόχρονα. Για την προσάρτηση ενός συστήματος αρχείου χρησιμοποιούμε την εντολή **mount**. Ένα παράδειγμα σε ένα Unix σύστημα είναι το ακόλουθο:

```
mount -t ext2 /dev/hdb1 /new-disk
```

❖ Είδη αρχείων:

- ✓ **Αρχεία χαρακτήρων:** χρησιμοποιούνται για να μοντελοποιήσουν συσκευές I / O.
- ✓ **Αρχεία μπλοκ:** χρησιμοποιούνται για την μοντελοποίηση δίσκων.
- ✓ **Αρχεία απεικονιζόμενα στη μνήμη**

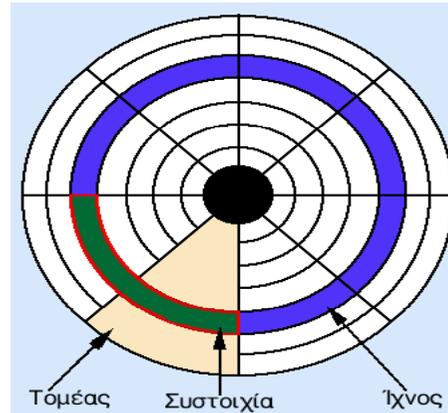
❖ Τα συστήματα αρχείων στο Linux

Η υλοποίηση του Vfs στο Linux μεταξύ άλλων συστημάτων αρχείων υποστηρίζει και αυτά: ext, ext2, ext3, reiserfs, xfs, xia, minix, umsdos, MS-dos, vfat, proc, smb, ncp, iso9660, sysv, hpfs, affs και ufs.

- FAT = File Allocation Table(Πίνακες που αποθηκεύονται στο σκληρό δίσκο, στον οποίο υπάρχει καταχωρημένη η ακριβής θέση κάθε αρχείου)
- Πρώτη υλοποίηση ήταν το FAT 12 που χρησιμοποιείται σε δισκέτες.
- Αργότερα εμφανίστηκε το FAT 16 όπου χρησιμοποιείται σε σκληρούς δίσκους κάτω από 2GB.
- Στο FAT 16 κάθε cluster = 8 Kb. Αυτό σημαίνει ότι ακόμα και αν το μέγεθος ενός αρχείου είναι ας πούμε 2 Kb, αυτό θα καταλάβει ένα ολόκληρο cluster δηλαδή 8 Kb.
- ❖ **Αποτέλεσμα:** απώλεια σημαντικής χωρητικότητας του σκληρού δίσκου.
- Το FAT 32 είναι κατάλληλο για σκληρούς δίσκους άνω των 2GB.
- Στο Fat 32 κάθε cluster = 4 Kb. Έτσι το αρχείο που έχει μέγεθος 2Kb πιάνει αυτή τη φορά 4 Kb και χάνονται μόνο 2 Kb αντί για 6 Kb που θα χάνονταν με το FAT 16.
- ❖ **Αποτέλεσμα:** αξιοποιείται καλύτερα ο σκληρός δίσκος.

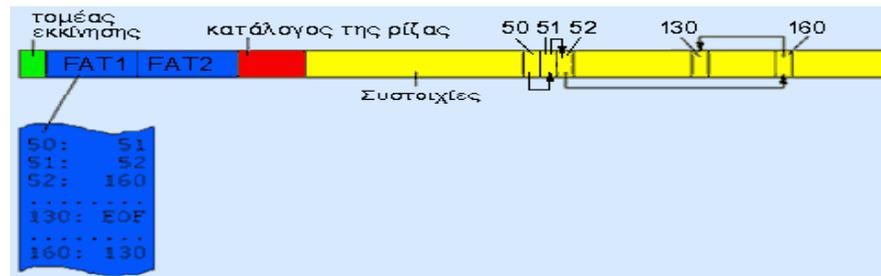
Κατανομή χώρου σκληρού δίσκου

Ο σκληρός δίσκος είναι χωρισμένος σε συστοιχίες, τομείς και ίχνη.



Σχήμα 3.1: Οργάνωση επιφάνειας δίσκου: ίχνη / τομείς / συστοιχίες

Οι πίνακες κατανομής αρχείων(FAT) είναι τοποθετημένοι στην περιοχή του δίσκου αμέσως μετά τον τομέα εκκίνησης του διαμερίσματος. Κάθε διαμέρισμα περιέχει δύο ίδια αντίγραφα του FAT. Το δεύτερο αντίγραφο χρησιμοποιείται ως εφεδρικό, σε περίπτωση που συμβεί κάποια ζημιά στο πρώτο.



Σχήμα 3.2: Πίνακας Κατανομής Αρχείων(FAT)

Τύπος Πίνακα Καταχώρησης	FAT16	FAT32
Μέγεθος συστοιχίας	32KB	4KB
Αριθμός Καταχωρήσεων του Πίνακα	65525	524208
Μέγεθος του Πίνακα Καταχώρησης	~128KB	~2MB

Πίνακας 3.1: διαφορές Fat16-Fat32

Το VFAT ονομαζόταν στο παρελθόν xmsdos και μετονομάστηκε σε VFAT. Πολλές εκδόσεις χρησιμοποιούν ακόμα το όνομα xmsdos. Παρέχει υποστήριξη για μεγάλα ονόματα αρχείων(μέχρι 255 χαρακτήρες) και καλύτερο έλεγχο σε θέματα όπως το κλείδωμα του δίσκου. Παρά το νέο όνομα το VFAT ως σύστημα αρχείων, είναι βασικά το ίδιο με το FAT.

Χαρακτηριστικά γνωρίσματα του NTFS

- Προσφέρει καλύτερη συμπίεση αρχείων και καταλόγων για αποθήκευση δεδομένων
- Υποστηρίζει μεγάλα μεγέθη αρχείων
- Ο δίσκος είναι χωρισμένος σε μικρά κομμάτια για την εξοικονόμηση χώρου όσον αφορά την αποθήκευση των αρχείων
- Επιτρέπει ονόματα αρχείων πάνω από 255 χαρακτήρες.
- Παρέχει έλεγχο ασφάλειας και πρόσβασης σε ένα κατάλογο ή αρχείο

NTFS ιδιότητες αρχείων

- ❖ **Όνομα αρχείων (FN):** καταχωρεί ένα όνομα σε ένα αρχείο
- ❖ **Ιδιότητες ρίζας ευρετηρίων:** περιέχει την εύρεση των αρχείων που περιλαμβάνονται μέσα σε έναν κατάλογο .
- ❖ **Περιγραφέας (SD) ασφάλειας:** περιέχει τις πληροφορίες ασφάλειας που ελέγχουν την πρόσβαση σε ένα αρχείο ή έναν κατάλογο αρχείων.
- ❖ **Πρότυπες πληροφορίες (SI):** Περιέχει πληροφορίες για όλα τα αρχεία και τους καταλόγους, όπως η ημερομηνία/ time-stamps για όταν δημιουργήθηκε το αρχείο, τροποποιήθηκε και προσπελάστηκε.

Λειτουργία EFS

Το EFS εξασφαλίζει ότι μόνο το άτομο που κρυπτογραφεί το αρχείο μπορεί και να το αποκρυπτογραφήσει.

Ιδιότητες ομάδων

- **διαχειριστής ομάδας**: έχει πρόσβαση παντού και διαχειρίζεται τους χρήστες που ανήκουν σε αυτή την ομάδα
- **ιδιοκτησία**: οι ιδιοκτήτες παρέχουν δικαιώματα σε άλλους
- **auditing**(ακρόαση): οι διαχειριστές παρακολουθούν τις αλλαγές που γίνονται στα αρχεία και τους καταλόγους

❖ Σχεδιάστηκε από τον Wayne Davidson

Το Linux αρχικά βασιζόταν στο Minix. Οι περιορισμοί όμως του Minix, όπως το μέγιστο μέγεθος του δίσκου που ήταν 2 GB και το μέγιστο μέγεθος του ονόματος των αρχείων που ανερχόταν στους 255 χαρακτήρες, οδήγησαν στη δημιουργία του Ext. Αλλά και το Ext είχε κάποια προβλήματα όπως για παράδειγμα δεν παρείχε τροποποίηση δεδομένων. Έτσι το 1993 εμφανίστηκε το Ext2, το οποίο παρείχε ακόμα περισσότερες βελτιώσεις όπως φαίνεται στον πίνακα 4.1

Μέγεθος συστήματος αρχείων	4TB
Μέγεθος Block	1 KB - 4 KB
Μέγιστο μέγεθος αρχείων	2 GB
Ονόματα αρχείων	255 χαρακτήρες

Πίνακας 4.1: Κατάσταση Ext2

Εργαλεία διαχείρισεως Ext2

- **mke2fs:** αρχικοποιεί μια διαμέριση για να δημιουργήσει ένα Ext2 σύστημα αρχείων σε αυτή
- **tune2fs:** ορίζουμε ρυθμίσεις όπως έλεγχο του συστήματος αρχείων, συμπεριφορά για τυχόν σφάλματα.
- **e2fsck:** επιδιορθώνει τις ασυνέπειες του συστήματος μετά από ένα λανθασμένο κλείσιμο του συστήματος
- **debugfs:** πρόγραμμα που εξετάζει την κατάσταση του συστήματος αρχείων.
- **dumpe2fs:** καταγράφει τη θέση του Ext2 στο δίσκο. Αναλύει τα προβλήματα και δίνει γενικές πληροφορίες για τη χρήση του filesystem
- **badblocks:** ελέγχει στο δίσκο ποιες ομάδες δεδομένων έχουν πάθει ζημιά
- **chattr:** αλλάζει τα χαρακτηριστικά των αρχείων σε ένα ext2 σύστημα αρχείων
- **e2label:** ετικέτα που χαρακτηρίζει το σύστημα αρχείων
- **lsattr:** βάζει σε λίστα τα χαρακτηριστικά του Ext2

Δημιουργία κρυπτογραφημένου συστήματος αρχείων με χρήση της «Loopback device»

1. Δημιουργία αρχείου με N bytes σε ένα ήδη υπάρχον σύστημα αρχείων

```
if = / dev/zero of=/myefs bs=1k count=100
```

2. Ο χρήστης οργανώνει την «loopback device»(κάνει mount ένα αρχείο σαν να είναι διαμέριση) να δείχνει στο αρχείο χρησιμοποιώντας κρυπτογράφηση serpent:

```
losetup -e serpent /dev/loop0/ myefs
```

- 3.Ο χρήστης δημιουργεί ένα καινούριο σύστημα αρχείων με την εντολή mkfs στη loopback device. Το σύστημα αρχείων κρυπτογραφείτε αυτόματα καθώς δημιουργείτε:

```
mkfs -t ext2 /dev/loop0 100
```

4. Ο χρήστης προσαρτεί το κρυπτογραφημένο σύστημα αρχείων, το οποίο βρίσκεται στο αρχείο των n bytes και αποκτά πρόσβαση σε αυτό:

```
mount -t ext2 /dev/loop0/mnt
```

5. Απενεργοποιούμε την loopback device:

```
umount /dev/loop0
```

```
losetup -d/dev/loop0
```

Ασφάλεια στο Ext2

Στόχος: περιορισμός ζημιάς που μπορεί να προκληθεί μέσω παραβίασης μιας διαδικασίας από την εισαγωγή χρηστών χωρίς δικαίωμα πρόσβασης

Κανόνες στη πρόσβαση των δεδομένων

- ❖ Εάν η διαδικασία έχει root δικαιώματα η πρόσβαση επιτρέπεται
- ❖ Εάν η πρόσβαση πρόκειται να γίνει από τον ιδιοκτήτη του αρχείου γίνεται έλεγχος να εξακριβωθεί αν έχει δοθεί το δικαίωμα πρόσβασης σε αυτό
- ❖ Εάν η πρόσβαση πρόκειται να γίνει από την ομάδα που δημιούργησε το αρχείο γίνεται πάλι έλεγχος για το αν έχει το δικαίωμα πρόσβασης
- ❖ Διαφορετικά γίνεται έλεγχος αν οι υπόλοιποι χρήστες έχουν δικαίωμα πρόσβασης

Παράδειγμα:

```
rwxr-xr-x 1 root root 8632 May 9 1998/usr/bin/id
```

Το αρχείο είναι αναγνώσιμο, εγγράψιμο και εκτελέσιμο από τον ιδιοκτήτη, από την ομάδα είναι αναγνώσιμο και εκτελέσιμο και για τους υπόλοιπους είναι μόνο εκτελέσιμο

Το **e2compr**

Το **e2compr** είναι ένα πρόγραμμα που επιτρέπει την συμπίεση και την αποσυμπίεση των κανονικών αρχείων. Τα αρχεία: superbloc, inodes, directory files δεν συμπιέζονται για λόγους ασφάλειας. Οι κατάλογοι μπορούν επίσης να συμπιεστούν που σημαίνει ότι συμπιέζονται αυτόματα και τα περιεχόμενά του

Μειονεκτήματα του Ext2

- ❖ Δεν είναι αξιόπιστο σε περίπτωση αποτυχίας (crash)
- ❖ Η διαδικασία fsck (έλεγχος συστήματος) επιβραδύνει την εκκίνηση του συστήματος
- ❖ Δεν παρέχει καλή διαχείριση τομέων που βασίζονται στην εγγραφή και διαγραφή. Για να διαγραφεί ένα byte, ολόκληρος ο τομέας θα πρέπει να αντιγραφεί στη μνήμη Ram, να διαγραφεί και επανεγγραφεί.

- ❑ Το Ext2 περιέχει βελτιώσεις όσον αφορά την ταχύτητα κατά την ανάγνωση και την εγγραφή αρχείων.
- ❑ Το Ext2 περιέχει βελτιώσεις όσον αφορά την κατανομή. Οι ομάδες Block είναι έτσι κατανομημένες ώστε ένα αρχείο να είναι στην ίδια ομάδα με το I-node του. Έτσι μειώνονται οι αναζητήσεις στο δίσκο.

Περιγραφή μέτρησης επιδόσεων

Οι δοκιμές των επιδόσεων του Ext2 έχουν γίνει σε ένα PC, με επεξεργαστή I486dx2, χρησιμοποιώντας 16 MB μνήμης και δύο δίσκους 429 MB IDE. Έχουν τρέξει δύο διαφορετικές δοκιμές όσον αφορά την επίδοση.

Η δοκιμή Bonnie γράφει τα δεδομένα στο αρχείο, επανεγγράφει τα περιεχόμενα ολόκληρου του αρχείου, γράφει δεδομένα χρησιμοποιώντας block, διαβάζει τα περιεχόμενα.

Η δοκιμή των επιδόσεων Andrew τρέχει σε 5 φάσεις: δημιουργεί μια ιεραρχία καταλόγου, κάνει ένα αντίγραφο των δεδομένων, εξετάζει την κατάσταση κάθε αρχείου, εξετάζει κάθε byte κάθε αρχείου και μεταγλωττίζει αρκετά από τα αρχεία

Αποτελέσματα δοκιμασίας επιδόσεων Bonnie όσον αφορά το Ext2

	Εγγραφή δεδομένων (KB/s)	Εγγραφή δεδομένων με χρήση block(KB/s)	Επανεγγραφή δεδομένων (KB/s)	Ανάγνωση δεδομένων (KB/s)
Ext2 fs	452	1237	536	397

Πίνακας 4.2: Αποτελέσματα δοκιμασίας επιδόσεων Bonnie όσον αφορά το Ext2

Αποτελέσματα δοκιμασίας επιδόσεων Andrew όσον αφορά το Ext2

	P1 Δημιουργία (ms)	P2 Αντιγραφή δεδομένων (ms)	P3 Εξέταση κατάστασης αρχείου (ms)	P4 Εξέταση byte αρχείου (ms)	P5 Μεταγλώττιση αρχείων (ms)
Ext2 fs	790	4791	7235	11685	63210

Πίνακας 4.3: Αποτελέσματα δοκιμασίας επιδόσεων Andrew όσον αφορά το Ext2

Το Ext3 δημιουργήθηκε από τον Stephen Tweedie. Είναι ένα journaling σύστημα αρχείων που είναι συμβατό με το Ext2.

Τύποι journaling

- Journal: Το σύστημα αρχείων γράφει κάθε αλλαγή που γίνεται στα δεδομένα και στα μεταδεδομένα δύο φορές, μια φορά στο σύστημα αρχείων και μια φορά στον τύπο Journal. Κάτι τέτοιο όμως οδηγεί στην μείωση της απόδοσης του συστήματος αλλά ελαχιστοποιεί την πιθανότητα λαθών στα αρχεία μας.
- Ordered: Καταχωρεί στο σύστημα αρχείων μόνο τις αλλαγές που γίνονται στα μεταδεδομένα, εφόσον τα αρχεία δεδομένων ενημερώσουν το δίσκο για τις αλλαγές που πρόκειται να υπάρξουν στα μεταδεδομένα. Είναι γρηγορότερος από τον τύπο Journal
- Writeback: Ο τρίτος αυτός τύπος είναι γρηγορότερος από τους άλλους δύο καθώς καταγράφει τις αλλαγές μόνο στα μεταδεδομένα και δεν περιμένει τα αρχεία δεδομένων να ενημερώσουν τον δίσκο για τις αλλαγές που πρόκειται να συμβούν στα μεταδεδομένα

Τους τύπους Journaling μπορούμε να ενεργοποιήσουμε στο /etc/fstab (καθορίζει πως, πότε και πού πρέπει να προσαρτηθούν τα συστήματα αρχείων)

Παράδειγμα:

```
/dev/hda5 /opt ext3 data=writeback 1 0
```

Βασικές αρχές υλοποίησης

- Journal Metadata Block (μεταδεδομένα): περιλαμβάνει το block που περιέχει τα μεταδεδομένα στο σύστημα αρχείων
- Descriptor Blocks: αναφέρει πότε δημιουργήθηκε, τροποποιήθηκε κτλ. το journal metadata block. Τα data descriptor block περιέχουν τα data header block
- Header block: δίνει πληροφορίες για το data descriptor

Εργαλεία διαχείρισης Ext3

- ❖ **mke3fs**: Δημιουργεί νέα Ext3 συστήματα αρχείων
- ❖ **quota utils**: Θέτουν όρια στο χώρο που καταλαμβάνει κάθε χρήστης στο δίσκο
- ❖ **attr**: Οι χρήστες συνδέουν τα αρχεία, τους καταλόγους του συστήματος αρχείων με το αντίστοιχο σύστημα αρχείων
- ❖ **acl**: Εκτελεί ελέγχους κατά την πρόσβαση στις λίστες αρχείων και καταλόγων
- ❖ **attr-devel και acl-devel**: Για ανάπτυξη του λογισμικού το οποίο θα χρησιμοποιήσει τα εργαλεία attr και acl

Χαρακτηριστικά Ext3

Μέγιστο μέγεθος συστήματος αρχείων	4 TB
Μέγεθος block	1 KB – 4 KB
Μέγιστο μέγεθος αρχείων	2 GB
Χρόνος επανεκκίνησης μετά από ένα λάθος	Γρήγορος
Ασφάλεια	Οι χρήστες επιλέγουν το είδος της ασφάλειας
Συνέπεια δεδομένων	Τα δεδομένα δε χάνονται μετά από κάποιο σφάλμα

Πίνακας 5.1: Χαρακτηριστικά Ext3

- ✓ Το Ext3 προσφέρει μεγαλύτερη ασφάλεια. Σε περίπτωση σφάλματος ο H/Y θα ανανεώσει και θα διατηρήσει τις αλλαγές που κάναμε στα δεδομένα πριν το κλείσιμο του υπολογιστή
- ✓ Η επαναφορά των αρχείων ύστερα από ένα απροσδόκητο κλείσιμο του H/Y είναι πιο γρήγορη στο Ext3 από ότι στο Ext2
- ✓ Ακεραιότητα δεδομένων: Το Ext3 μας δίνει τη δυνατότητα να επιλέξουμε μεταξύ 1) αλλοίωσης των δεδομένων σε περίπτωση σφάλματος και 2) αποτροπή αλλοίωσης των δεδομένων σε περίπτωση σφάλματος
- ✓ Ταχύτητα: Για την βελτίωση της ταχύτητας μπορούμε να επιλέξουμε μεταξύ των τριών τρόπων που αναφέρθηκαν παραπάνω.
- ✓ Εύκολη μετάβαση: το Ext2 μπορεί να μεταβεί εύκολα σε Ext3 με το πρόγραμμα **tune2fs** το οποίο προσθέτει ένα journal σε ένα υπάρχον Ext2 σύστημα αρχείων.

Παράδειγμα:

```
# /tune2fs -j /dev / hda5
```

- ✓ Τα νέα χαρακτηριστικά που προστίθενται στο Ext2 μπορούν εύκολα να μεταφερθούν στο Ext3
- ✓ Το Ext3 όπως και το Ext2 χρησιμοποιούν τον ίδιο κώδικα e2fsck για τη διόρθωση και την επαναφορά του συστήματος αρχείων

Το Reiserfs στηρίζεται στους ισορροπημένους αλγόριθμους δέντρων (δομή δεδομένων που επιτρέπει την γρήγορη εύρεση, διαγραφή και εισαγωγή πληροφοριών). Μεταχειρίζεται έναν ολόκληρο δίσκο σαν ένα πίνακα βάσεων δεδομένων. Το Reiserfs παρακολουθεί το μέγεθος των αρχείων που βρίσκονται σε αυτή τη δενδρική δομή και τα τοποθετεί στο ανάλογο μπλοκ για εξοικονόμηση χώρου.

Παράδειγμα:

Στα μικρά αρχεία η διαδικασία αυτή εξοικονομεί πολύ χώρο αποθήκευσης, καθώς τα αρχεία τοποθετούνται κοντά το ένα με το άλλο.

Ιστορική αναδρομή

Το Reiserfs ονομάστηκε έτσι από αυτόν που το δημιούργησε, τον Hans Reiser. Αρχικά ονομαζόταν «TreeFS». Το Reiserfs είναι ενσωματωμένο από την έκδοση του πυρήνα του Linux 2.4.1

- ❑ **mkreiserfs:** δημιουργεί ένα σύστημα αρχείων
- ❑ **reiserfsck:** εκτελεί έλεγχο για το σύστημα αρχείων reiserfs
- ❑ **resize_reiserfs:** χρησιμοποιείται για την αλλαγή μεγέθους σε συστήματα αρχείων reiserfs
- ❑ **debugreiserfs:** χρησιμοποιείται για την επίλυση προβλημάτων
- ❑ **reiserfstune:** μπορεί να αλλάξει το μέγεθος journal

Μέγιστο μέγεθος αρχείων	4 G
Αριθμός αρχείων που έχει ένας κατάλογος	2 G
Αριθμός υποκαταλόγων σε ένα κατάλογο	64.536 K
Αριθμός συνδέσεων σε ένα αρχείο	32.768 K
Μέγιστο μέγεθος συστήματος αρχείων	17.6 T
Μέγεθος Block	4 KB
Χρόνος επανεκκίνησης μετά από ένα λάθος	Γρήγορος
Ανάκτηση δεδομένων στην περίπτωση σφάλματος	Πολύ καλή. Η ολοκληρωτική απώλεια δεδομένων είναι πολύ σπάνια

Πίνακας 6.1: Επιδόσεις Reiserfs

Το Xfs είναι ένα σύστημα αρχείων «καταγραφής». Αυτό σημαίνει ότι το σύστημα αρχείων έχει μια εικόνα του τι έκανε εκείνη τη στιγμή. Κατά συνέπεια η επαναφορά του συστήματος ύστερα από κάποιο κρέμασμα είναι πολύ γρήγορη.

Λειτουργία καταγραφής

Όταν πρόκειται να συμβεί μια εγγραφή αρχείου, το σύστημα αρχείων πρώτα γράφει τη λεγόμενη «καταχώρηση πρόθεσης αλλαγής» και έπειτα γίνεται η καθεαυτή εγγραφή του αρχείου. Αντί λοιπόν να ελεγχθεί όλο το σύστημα αρχείων ελέγχεται μόνο η καταχώρηση, και αν βρεθεί μια καταχώρηση πρόθεσης αλλαγής που δεν έχει ολοκληρωθεί, η δομή των αρχείων την καταχωρεί.

Το XFS αποτελείται από τρία τμήματα

- Data section (τμήμα δεδομένων): περιλαμβάνει όλα τα μεταδεδομένα και τα αρχεία.
- Log section: χρησιμοποιείται για να καταχωρήσει τις αλλαγές στα μεταδεδομένα του Filesystem.
- Real time: χρησιμοποιείται για να αποθηκεύσει τα δεδομένα.

Το XFS αναπτύχθηκε από την Silicon Graphics στις αρχές τις δεκαετίας του 90. Σκοπός της SGI ήταν η δημιουργία νέων εφαρμογών με την παροχή υποστήριξης σε περιοχές όπως η γρήγορη και αξιόπιστη αποκατάσταση των σφαλμάτων. Το XFS είναι διαθέσιμο από το 1994 και χρησιμοποιείται μέχρι και σήμερα. Ένα παράδειγμα χρησιμοποίησής του είναι στο Linux.

Εργαλεία Διαχειρίσεως

- ✓ **mkfs**: χρησιμεύει στη δημιουργία του συστήματος αρχείων
- ✓ **dump – restore**: κρατάμε αντίγραφα ασφαλείας. Η εντολή dump καθορίζει ποια αρχεία πρέπει να γίνουν back up και τα αντιγράφει σε ένα συγκεκριμένο δίσκο, ταινία ή άλλο αποθηκευτικό μέσο. Η εντολή restore κάνει το αντίστροφο από την εντολή dump. Μπορεί να επαναφέρει ένα πλήρες αντίγραφο ασφαλείας ενός συστήματος αρχείων
- ✓ **xfs_db**: διορθώνει τα προγραμματιστικά λάθη(κάνει debug)
- ✓ **xfs_check**: ελέγχει το σύστημα αρχείων
- ✓ **xfs_repair**: διορθώνει το σύστημα αρχείων
- ✓ **xfs_fsr**: χρησιμεύει στην ανασυγκρότηση του XFS
- ✓ **xfs_bmap**: δείχνει σε ποιο σημείο του δίσκου βρίσκονται τα αρχεία

Τεχνολογία	64 bit σύστημα αρχείων
Μέγιστο μέγεθος συστήματος αρχείων	18 εκατομμύρια TB
Μέγεθος Block	4 KB
Μέγεθος τομέα δίσκων	512 Bytes
Μνήμη	64 MB
Ανά σύστημα αρχείων	Εκατομμύρια αρχεία
Ανά κατάλογο	Εκατομμύρια αρχεία
Γρήγορη αποκατάσταση	Συστήματος αρχείων

Πίνακας 7.1: Χαρακτηριστικά γνωρίσματα XFS

Ταχύτητα

Σύστημα αρχείων	Αρχεία	Sequential Create (Διαδοχική δημιουργία)						Random Create (Τυχαία δημιουργία)					
		Δημιουργία		Ανάγνωση		Διαγραφή		Δημιουργία		Ανάγνωση		Διαγραφή	
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU
Ext2	16	94	99	278	99	492	97	95	99	284	100	93	41
Ext3	16	89	98	274	100	458	96	93	99	288	99	97	45
Xfs	16	92	99	251	96	436	98	91	99	311	99	90	41
Reisefs	16	1307	100	8963	100	1914	99	1245	99	9316	100	1725	100

Πίνακας 8.1: Σύγκριση συστημάτων αρχείων ως προς την ταχύτητα

Ασφάλεια

Συστήματα αρχείων	Παροχή ασφάλειας
FAT	-
NTFS	Κάθε χρήστης έχει δικαίωμα πρόσβασης σε συγκεκριμένα αρχεία του συστήματος
EXT2 - EXT3 -REISERFS	Δεδομένων
XFS	Αντίγραφα ασφαλείας

Πίνακας 8.2: Σύγκριση συστημάτων αρχείων ως προς την ασφάλεια

Συμβατότητα

Συστήματα αρχείων	Συμβατότητα
FAT	Είναι συμβατό με το NTFS
NTFS	-
EXT2	Είναι συμβατό με το Ext3
EXT3	Είναι συμβατό με το Ext2
REISERFS	-
XFS	-

Πίνακας 8.3: Σύγκριση συστημάτων αρχείων ως προς τη συμβατότητα

Συμπίεση

Συστήματα αρχείων	Συμπίεση
FAT	Υποστηρίζει συμπίεση δίσκων
NTFS	-
EXT2-EXT3	Χρησιμοποιεί τη μέθοδο συμπίεσης και αποσυμπίεσης αρχείων: e2compr
REISERFS	-
XFS	Γρήγορη αλλά μειωμένη απόδοση ως προς τη συμπίεση αρχείων

Πίνακας 8.4: Σύγκριση συστημάτων αρχείων ως προς τη συμπίεση

Ανάκτηση δεδομένων

Συστήματα αρχείων	Ανάκτηση δεδομένων
FAT - NTFS	Χρησιμοποιούν πίνακα για να αποθηκεύσουν τα αρχεία στον σκληρό δίσκο.
EXT2-EXT3	Χρησιμοποιούν τους κόμβους-δ
REISERFS-XFS	Η ανάγνωση και ανάκτηση δεδομένων γίνεται πολύ γρήγορα λόγω της χρήσης ισορροπημένων δέντρων για την εύρεση αρχείων, καταλόγων κτλ.

Πίνακας 8.5: Σύγκριση συστημάτων αρχείων ως προς τη ανάκτηση δεδομένων

Μέγεθος Κατατμήσεων

Συστήματα Αρχείων	Ελάχιστο Μέγεθος Διαμέρισης	Μέγιστο Μέγεθος Διαμέρισης
FAT16	128 MB	2 GB
FAT32	<260 MB	128GB
NTFS	0 GB	64 GB
EXT2	<1.4 MB	4 TB
EXT3	<1.4 MB	4TB
REISERFS	32 MB	17,6 TB
XFS	0 GB	18 εκατομμύρια TB

Πίνακας 8.6: Μεγέθη διαμέρισης και συστοιχιών συστημάτων αρχείων

Σε αυτήν την ομιλία αναφερθήκαμε στα Συστήματα Αρχείων,
στα Journaling Filesystems και συγκρίναμε διάφορα
συστήματα αρχείων μεταξύ τους

ΕΥΧΑΡΙΣΤΩ