



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΕΤΑΤΡΟΠΗ ΣΥΣΤΗΜΑΤΟΣ ΑΡΧΕΙΩΝ ΑΠΟ FAT ΣΕ EXT2



Του φοιτητή

Κλάρου Γεώργιου

Αρ. Μητρώου: 01/1659

Επιβλέπων Καθηγητής

Σιδηρόπουλος Αντώνιος

Θεσσαλονίκη 2011

ΠΡΟΛΟΓΟΣ

Η εργασία αυτή ασχολείται με ένα θέμα που δεν έχει αντιμετωπιστεί στην βιβλιογραφία μέχρι σήμερα.

Την μετατροπή μεταξύ δύο τελείως διαφορετικών συστημάτων αρχείων (Σ.Α.), την μετατροπή δηλαδή μεταξύ του FAT (16 ή 32), ενός συστήματος αρχείων της οικογένειας λειτουργικών προγραμμάτων της Microsoft και του Ext2 ενός συστήματος αρχείων του Linux.

Το FAT βασίζεται πάνω σε ομάδες τομέων του δίσκου που συγκροτούν την βασική δομική του μονάδα, το cluster , ενώ το σύστημα αρχείων ext2 βασίζεται πάνω στην λογική των ιδεατών και φυσικών μπλοκ στα οποία χωρίζει τον δίσκο και χρησιμοποιεί κάποιες αρκετά προηγμένες δυνατότητες. Αντίθετα με το ext2, το FAT δεν έχει σχεδόν καμία επιπλέον δυνατότητα, πέραν των απαραίτητων σε ένα σύστημα αρχείων όπως οι λειτουργίες ανάγνωσης, εγγραφής, διαγραφής κτλ.

ΠΕΡΙΛΗΨΗ

Το Σύστημα Αρχείων είναι το απαραίτητο υπόστρωμα των Λειτουργικών Συστημάτων των Υπολογιστών που ασχολείται με την φυσική πρόσβαση των Λειτουργικών Συστημάτων στο μέσο αποθήκευσης (σκληρό δίσκο, CD-ROM, USB disk, DVD-ROM κτλ).

Το Σύστημα Αρχείων είναι υπεύθυνο για όλες τις λειτουργίες πάνω σε αρχεία και φακέλους, όπως αντιγραφή, διαγραφή, άνοιγμα, κλείσιμο, ανάγνωση κτλ. Επίσης είναι υπεύθυνο για την φυσική τοποθέτηση των δεδομένων των αρχείων και των φακέλων στο μέσο αποθήκευσης.

Στην παρούσα εργασία αναλύονται οι διαφορές των δύο Συστημάτων Αρχείων FAT & ext2 και περιγράφεται η προσπάθεια κατασκευής ενός εργαλείου μετατροπής ενός συστήματος αρχείων από FAT σε ext2.

ABSTRACT

This essay deals with a particular subject for first time in the literature.

This is the transformation of the file system of a volume (disk or partition) formatted in FAT (Microsoft file system format), to ext2 format (Linux file system standard format)

Ext2 file system is based on logical and physical blocks for handling the volume, while utilizing some very advanced features like hard or symbolic links, i-nodes and Case Sensitive file storing.

FAT on the other hand is not utilizing any such advanced feature other than the basic and necessary features of a file system such as read, write, delete etc

ΕΥΧΑΡΙΣΤΙΕΣ

Η εργασία αυτή προετοιμάστηκε και υλοποιήθηκε στο ακαδημαϊκό έτος 2010 – 2011 σύμφωνα με τον κανονισμό των πτυχιακών του τμήματος Πληροφορικής ΤΕΙ Θεσσαλονίκης.

Θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Σιδηρόπουλο Αντώνη για την εμπιστοσύνη και την διάθεση για βοήθεια που μου έδειξε κατά την υλοποίηση της πτυχιακής.

Τέλος ένα μεγάλο ευχαριστώ στην οικογένεια μου για την ηθική και οικονομική συμπαράσταση καθ' όλη την διάρκεια των σπουδών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΛΗΨΗ	4
ABSTRACT	5
ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΕΧΟΜΕΝΑ	7
Ευρετήριο σχημάτων	10
Ευρετήριο πινάκων.....	11
ΕΙΣΑΓΩΓΗ.....	12
ΚΕΦΑΛΑΙΟ 1.....	14
ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ.....	14
ΕΙΣΑΓΩΓΗ.....	14
1.1 Τι ονομάζεται Σύστημα Αρχείων (Filesystem)	15
1.2 Αρμοδιότητες του Συστήματος Αρχείων	15
1.3 Ονομασία Αρχείων	16
1.4 Μεταδεδομένα	16
1.5 Δυνατότητες, Αρμοδιότητες του Συστήματος Αρχείων	17
ΕΠΙΛΟΓΟΣ.....	18
ΚΕΦΑΛΑΙΟ 2.....	20
ΤΑ ΣΥΣΤΗΜΑΤΑ ΑΡΧΕΙΩΝ ΣΗΜΕΡΑ	20
ΕΙΣΑΓΩΓΗ.....	20
2.1 Τα Συστήματα Αρχείων Σήμερα	25
2.1.1 Συστήματα Αρχείων της Microsoft.....	25
2.1.1 Συστήματα Αρχείων στο Linux	25

ΕΠΙΛΟΓΟΣ.....	26
ΚΕΦΑΛΑΙΟ 3.....	28
ΤΟ ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ FAT.....	28
ΕΙΣΑΓΩΓΗ.....	28
3.1: Η αρχιτεκτονική του FAT.....	29
3.2: Τα συστατικά του FAT.....	29
3.3: Η φυσική δομή του FAT.....	30
3.3.1: Clusters & Sectors στο FAT.....	30
3.3.2: Αριθμοδότηση Clusters.....	31
3.3.3: Περιορισμοί του μεγέθους του Cluster στο FAT.....	31
3.3.4: Ανώτατα όρια λογικού δίσκου στο FAT.....	32
3.3.5: Φυσική Οργάνωση του FAT.....	34
3.3.6: Τομέας Εκκίνησης & Εκκίνηση Δίσκου.....	35
3.3.7: Περιεχόμενα του Τομέα Εκκίνησης.....	35
3.3.8: Βασικές Διαφορές FAT16 & FAT32.....	36
3.3.9: Διαδικασία Διαμόρφωσης τόμου στο FAT.....	37
3.3.10: Επιπτώσεις του Κατακερματισμού στην Απόδοση.....	37
3.3.12: Ονομασία Αρχείων στο FAT.....	42
3.3.13: Εκτεταμένη Ονοματοδοσία Αρχείων στο FAT.....	42
3.3.14: Η Διαδικασία Προσπέλασης Αρχείων στο FAT.....	44
ΕΠΙΛΟΓΟΣ.....	47
ΚΕΦΑΛΑΙΟ 4.....	49
ΤΟ ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ Ext2.....	49
ΕΙΣΑΓΩΓΗ.....	49
4.1: Το Virtual File System (VFS).....	50
4.2: Η δομή του Ιδεατού Συστήματος Αρχείων.....	51
4.3: Η φυσική δομή του Ext2.....	52

4.4: Βασικά Δομικά Στοιχεία του Ext2	58
4.4.1: Inodes	58
4.4.2: Φάκελοι (Directories).....	59
4.4.3: Δεσμοί (Links)	60
4.4.4: Ειδικά Αρχεία Συσκευής (Special Device Types)	61
4.4.5: Η βιβλιοθήκη του Ext2.....	61
4.4.6: Επιδόσεις του Ext2.....	62
4.4.7: Οι δυνατότητες του Ext2.....	63
4.4.8: Προηγμένες δυνατότητες του Ext2	64
ΕΠΙΛΟΓΟΣ.....	66
ΚΕΦΑΛΑΙΟ 5.....	67
Διαφορές μεταξύ Ext2 και FAT (16/32).....	67
ΕΙΣΑΓΩΓΗ.....	67
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	69
ΒΙΒΛΙΟΓΡΑΦΙΑ	70
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ Partition Surprise	72
ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΟΥΜΕΝΩΝ ΒΗΜΑΤΩΝ ΜΕΤΑΤΡΟΠΗΣ	115

Ευρετήριο σχημάτων

Σχήμα 1: Η φυσική διαστρωμάτωση του FAT (Microsoft, 2003a).....	29
Σχήμα 2: Λογική διάρθρωση ενός τόμου με FAT.....	34
Σχήμα 3: Εκτεταμένη Ονοματοδοσία Αρχείων στο FAT	44
Σχήμα 4: Παράδειγμα Προσπέλασης αρχείων στο FAT	44
Σχήμα 5: Το VFS στο ext2 & στο Linux (Card, Ts'ο, & Tweedie, 1994)	51
Σχήμα 6: Η φυσική δομή του Ext2	52
Σχήμα 7:Αποτύπωση της λογικής διάταξης μπλοκ στο Ext2	53
Σχήμα 8: Το superblock στον δίσκο	55
Σχήμα 9: Το group descriptor στον δίσκο	56
Σχήμα 10: Αναπαράσταση της δομής ενός φακέλου στο Ext2	56
Σχήμα 11: Αναπαράσταση ονομάτων αρχείων στο Est2.....	56
Σχήμα 12: Αναπαράσταση ονομάτων αρχείων στο Est2.....	57
Σχήμα 13: Η δομή ενός inode στο Ext2.....	59
Σχήμα 14: Τυπική καταγραφή ενός φακέλου στο Ext2	60

Ευρετήριο πινάκων

Πίνακας 1: Ιστορική Εξέλιξη Συστημάτων Αρχείων	20
Πίνακας 2: Περιεχόμενα του Σ.Α. FAT (Microsoft, 2003a)	30
Πίνακας 3: Μέγεθος Cluster σε συνάρτηση με το μέγεθος δίσκου.....	32
Πίνακας 4: Περιορισμοί του FAT16.....	33
Πίνακας 5: Περιορισμοί του FAT32.....	34
Πίνακας 6: Περιεχόμενα ενός τόμου διαμορφωμένου με FAT	34
Πίνακας 7: Υποστήριξη εκδόσεων FAT από Λ.Σ. Microsoft (Microsoft, 2003b)....	36
Πίνακας 8: Δομή αποθήκευσης πληροφοριών FAT από Λ.Σ. Microsoft (Microsoft, 2003b)	38
Πίνακας 9: Εκτεταμένο ονοματοδοσία FAT από Λ.Σ. Microsoft (Microsoft, 2003b)	43
Πίνακας 10: Τιμές καταχώρησης FAT	46
Πίνακας 11: Δομή Superblock και Block Group Descriptor.....	53
Πίνακας 12: Επιπλέον δυνατότητες του Ext2 συγκριτικά με το FAT	67
Πίνακας 13: Περιορισμοί στην ονοματοδοσία FAT, Ext2	67

ΕΙΣΑΓΩΓΗ

Στόχος της παρούσας πτυχιακής ήταν η κατασκευή ενός εργαλείου μετατροπής ενός τόμου (δίσκου ή κατάτμησης, partition) διαμορφωμένου με FAT, σε διαμόρφωση συστήματος αρχείων Ext2, προφανώς χωρίς απώλεια των υπαρχόντων δεδομένων.

Η εργασία περιλαμβάνει τα κάτωθι κεφάλαια:

Κεφάλαιο 1 & 2: Όλη η θεωρητική περιγραφή των Συστημάτων Αρχείων

Κεφάλαιο 3: Το FAT16 & FAT32 της Microsoft

Κεφάλαιο 4: Το Ext2

Κεφάλαιο 5: Σύγκριση μεταξύ των δύο αυτών Συστημάτων Αρχείων

ΚΕΦΑΛΑΙΟ 1

ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ

ΕΙΣΑΓΩΓΗ

Το Σύστημα Αρχείων (Σ.Α. , file system ή filesystem) είναι το απαραίτητο υπόστρωμα για κάθε Λειτουργικό Σύστημα (Λ.Σ.) σήμερα.

Βασικές λειτουργίες αρχείων και φακέλων είναι η αποκλειστική αρμοδιότητα του. Επίσης αναλαμβάνει να οργανώνει τα αρχεία και τους φακέλους σε λογικές δομές συνήθως μέσω πινάκων ώστε να είναι εύκολες οι λειτουργίες ανεύρεσης, ταξινόμησης, ιεράρχησης, αλλά και φυσικά λειτουργίες Εισόδου/Εξόδου (In/Out, I/O ή αλλιώς read/write), διαγραφής, μετονομασίας κτλ.

Το Σ.Α. χρησιμοποιείται από το υπερκείμενο Λειτουργικό Σύστημα (Λ.Σ.) κάθε φορά που αυτό χρειάζεται φυσική πρόσβαση σε αρχεία ή φακέλους.

Επίσης το Σ.Α. αναλαμβάνει κατά την διάρκεια διαμόρφωσης του τόμου να εγκαταστήσει όλα τα απαραίτητα αρχεία εκκίνησης στον τόμο, να καθορίσει την γεωμετρία του, και να καθορίσει κατόπιν επιλογής του χρήστη, το μέγεθος της ελάχιστης μονάδας αποθήκευσης (cluster ή block).

Το Σ.Α. χρησιμοποιείται από όλες τις συσκευές αποθήκευσης όπως οι δίσκοι , τα DVD κτλ για να αποθηκεύσουν τα δεδομένα στο φυσικό μέσο. Επίσης παρέχουν την απαραίτητη πρόσβαση σε δεδομένα μέσω δικτύου και των αντίστοιχων πρωτοκόλλων.

1.1 Τι ονομάζεται Σύστημα Αρχείων (Filesystem)

Ένα Σύστημα Αρχείων Σ.Α. (filesystem ή file system) είναι απαραίτητο μέρος ενός λειτουργικού συστήματος (Λ.Σ.). Σε έναν λογικό ή φυσικό χώρο αποθήκευσης καθορίζει πώς ονομάζονται, αποθηκεύονται και οργανώνονται τα αρχεία και οι φάκελοι.

Ένα Σ.Α. διαχειρίζεται τα αρχεία και τους φακέλους, καθώς και τις πληροφορίες που απαιτούνται για τον εντοπισμό και την πρόσβαση αυτών των στοιχείων από τους τοπικούς και απομακρυσμένους χρήστες του συγκεκριμένου μέσου αποθήκευσης, από το ίδιο ή από άλλο Λ.Σ.

Όταν διαμορφώνεται ένα φυσικό μέσο αποθήκευσης (σκληρός δίσκος), ένα Σ.Α. είναι απαραίτητο για την πρόσβαση και την διαχείριση των δεδομένων σε αυτό το μέσο αποθήκευσης.

Αν το μέσο αποθήκευσης διαχωριστεί σε πολλούς λογικούς τόμους (volumes) τότε κάθε τόμος μπορεί να έχει διαφορετικό σύστημα αρχείων.

1.2 Αρμοδιότητες του Συστήματος Αρχείων

Κάνοντας χρήση του φυσικού μέσου αποθήκευσης, το Σ.Α. παρέχει πρόσβαση σε ένα πίνακα τομέων (sectors) καθορισμένου μεγέθους, που είναι συνήθως μεγέθους δύναμης του 2 (π.χ. 512 bytes, μέχρι 4KB). Το Σ.Α. είναι υπεύθυνο για την οργάνωση αυτών των τομέων σε αρχεία ή φακέλους και επίσης αναλαμβάνει να γνωρίζει ποιος από αυτούς τους τομείς ανήκει σε ποιο αρχείο ή φάκελο και ποιος δεν χρησιμοποιείται.

Τα περισσότερα Σ.Α. αποθηκεύουν τα δεδομένα σε προκαθορισμένου μεγέθους λογικές μονάδες, ονομαζόμενες clusters ή blocks, τα οποία με την σειρά τους εμπεριέχουν ένα πλήθος τομέων (sectors) (συνήθως από 1 έως 64 τέτοιους τομείς). Αυτό είναι αναγκαστικά και το μικρότερο μέγεθος αρχείου που μπορεί να υπάρξει στον συγκεκριμένο τόμο.

1.3 Ονομασία Αρχείων

Το όνομα αρχείου (File name ή filename) είναι η ονομασία που ανατίθεται σε ένα αρχείο δεδομένων ώστε να αντιστοιχηθούν μοναδικά τα δεδομένα του με την συγκεκριμένη ονομασία. Το Σ.Α. αναλαμβάνει να οργανώσει και να συντηρεί έναν πίνακα με τις ονομασίες των αρχείων και να τις αντιστοιχίζει σε συγκεκριμένα αρχεία, όπως για παράδειγμα το File Allocation Table του FAT της Microsoft, ή το inode των Unix συγγενών Λειτουργικών Συστημάτων.

Η δομή των φακέλων μέσα στους οποίους μπορούν να περιέχονται τα αρχεία, μπορεί να είναι οριζόντια, ή ιεραρχική (δενδροειδής), δηλαδή ο κάθε φάκελος μπορεί να περιέχει αρχεία, αλλά και άλλους φακέλους, οι οποίοι με την σειρά τους μπορούν να περιέχουν άλλα αρχεία και φακέλους κοκ.

Σε κάποια Σ.Α. οι ονομασίες των αρχείων υπόκεινται σε συγκεκριμένους περιορισμούς όπως κανόνες ονομασίας και extension (π.χ. 8.3 στο MS-DOS) ή μπορεί να είναι απλώς μία συμβολοσειρά (string) και τα όποια μεταδεδομένα σχετικά με το αρχείο να αποθηκεύονται σε ξεχωριστή τοποθεσία.

1.4 Μεταδεδομένα

Απαραίτητη πληροφορία σχετικά με το κάθε αρχείο, θεωρείται όλη αυτή η πληροφορία που αφορά στις λειτουργικές λεπτομέρειες του αρχείου και μεταβάλλονται συνήθως μετά από κάθε επέμβαση πάνω σε αυτό.

Αυτές μπορεί να είναι:

- Μέγεθος αρχείου, που συνήθως αποθηκεύεται ως πλήθος των μπλοκ που καταλαμβάνει στο φυσικό μέσον
- Χρονολογία και ώρα δημιουργίας
- Χρονολογία και ώρα πρόσβασης σε αυτό
- Χρονολογία και ώρα τροποποίησης

- Είδος αρχείου
- Ιδιοκτήτη αρχείου
- Δικαιώματα πρόσβασης
- Είδος αρχείου (εκτελέσιμο, γραφικό κτλ)

Επίσης σε κάποια Σ.Α. μπορούν να ανατεθούν και εκτεταμένες ιδιότητες σε αρχεία ή φακέλους τα οποία διαφέρουν σε δυνατότητες ανάλογα με το Σ.Α.

1.5 Δυνατότητες, Αρμοδιότητες του Συστήματος Αρχείων

Τα παραδοσιακά Σ.Α. προφέρουν όλες τις βασικές δυνατότητες χρήσης πάνω σε αρχεία. Αυτές είναι τουλάχιστον η δημιουργία, μετακίνηση & διαγραφή αρχείων και φακέλων.

Επίσης προσφέρουν δυνατότητες αποκοπής, συμπλήρωσης και μετατροπής αρχείων.

Είναι προφανές ότι κάποια προηγμένα Σ.Α. όπως αυτά που χρησιμοποιούνται από την οικογένεια Unix-Linux χρησιμοποιούν κάποιες προηγμένες δυνατότητες που δεν ανήκουν στο λεγόμενο στάνταρ κορμό των Σ.Α. Π.χ. όλα τους δίνουν την δυνατότητα των hard links, την δυνατότητα δηλαδή παράθεσης του ίδιου αρχείου σε πολλαπλές τοποθεσίες-φακέλους μέσα στο Σ.Α. χωρίς να μεταφέρεται το ίδιο το αρχείο, αλλά απλώς δημιουργώντας έναν δείκτη (hard link) που δείχνει στο ένα αρχικό μοναδικό αρχείο. Κάθε φορά που το αρχείο αυτό μεταβάλλεται, όλες οι τοποθεσίες όπου υπάρχουν δείκτες προς αυτό, βλέπουν αυτόματα το καινούριο αρχείο.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο έγινε μία συνοπτική αναφορά στο Σύστημα Αρχείων γενικά, στις αρμοδιότητες του, καθώς και τις κυριότερες λειτουργίες του. Αναφέρθηκε ότι το Σ.Α. είναι το κατώτερο στρώμα διασύνδεσης του Λειτουργικού Συστήματος με το φυσικό μέσον αποθήκευσης.

Στο επόμενο κεφάλαιο θα γίνει μία μικρή αναφορά σε διάφορα συστήματα αρχείων.

ΚΕΦΑΛΑΙΟ 2

ΤΑ ΣΥΣΤΗΜΑΤΑ ΑΡΧΕΙΩΝ ΣΗΜΕΡΑ

ΕΙΣΑΓΩΓΗ

Σήμερα υπάρχει μία πληθώρα Συστημάτων Αρχείων. Επειδή δεν έχει νόημα να αναφερθούν όλα τα υπάρχοντα Σ.Α. θα γίνει μία περιληπτική αναφορά σε κάποια από αυτά.

Παρακάτω παρατίθεται ένας πίνακας που αναφέρει την ιστορική εξέλιξη όλων των Σ.Α. σύμφωνα με την Wikipedia (Wikipedia, 2011):

Πίνακας 1: Ιστορική Εξέλιξη Συστημάτων Αρχείων

Σύστημα Αρχείων	Δημιουργός	Έτος Δημιουργίας	Αρχικό Λειτουργικό Σύστημα
DECtape	DEC	1964	PDP-6 Monitor
Level-D	DEC	1968	TOPS-10
George 2	ICT (later ICL)	1968	George 2
V6FS	Bell Labs	1972	Version 6 Unix
ODS-1	DEC	1972	RSX-11
RT-11 file system	DEC	1973	RT-11
DOS (GEC)	GEC	1973	Core Operating System
CP/M file system	Gary Kildall	1974	CP/M
OS4000	GEC	1977	OS4000
FAT12	Microsoft	1977	Microsoft Disk BASIC
DOS 3.x	Apple Computer	1978	Apple DOS
Pascal	Apple Computer	1978	Apple Pascal
CBM DOS	Commodore	1978	Microsoft BASIC (for CBM PET)

V7FS	Bell Labs	1979	Version 7 Unix
ODS-2	DEC	1979	OpenVMS
DFS	Acorn Computers Ltd	1982	Acorn BBC Micro MOS
ADFS	Acorn Computers Ltd	1983	Acorn Electron (later Arthur RISC OS)
FFS	Kirk McKusick	1983	4.2BSD
ProDOS	Apple Computer	1983	ProDOS 8
MFS	Apple Computer	1984	Mac OS
Elektronika BK tape format	NPO "Scientific centre" (nowSitronics)	1985	Vilnius Basic, BK monitor program
HFS	Apple Computer	1985	Mac OS
Amiga OFS^[11]	Metacomco for Commodore	1985	Amiga OS
High Sierra	Ecma International	1985	MS-DOS, Mac OS
NWFS	Novell	1985	NetWare 286
FAT16	Microsoft	1987	MS-DOS 3.31
Minix V1 FS	Andrew S. Tanenbaum	1987	Minix 1.0
Amiga FFS	Commodore	1988	Amiga OS 1.3
HPFS	IBM & Microsoft	1988	OS/2
ISO 9660:1988	Ecma International, Microsoft	1988	MS-DOS, Mac OS, and AmigaOS
JFS1	IBM	1990	AIX ^[1]
VxFS	VERITAS	1991	SVR4.0
WAFL	NetApp	1992	Data ONTAP
Minix V2 FS	Andrew S. Tanenbaum	1992	MINIX 1.6 and 2.0
AdvFS	DEC	1993[2]	Digital Unix
NTFS Version 1.0	Microsoft, Tom Miller, Gary Kimura	1993	Windows NT 3.1

LFS	Margo Seltzer	1993	Berkeley Sprite
ext2	Rémy Card	1993	Linux, Hurd
UFS1	Kirk McKusick	1994	4.4BSD
XFS	SGI	1994	IRIX, Linux, FreeBSD
HFS	IBM	1994	MVS/ESA (now z/OS)
Joliet ("CDFS")	Microsoft	1995	Microsoft Windows, Linux, Mac OS, and FreeBSD
UDF	ISO/ECMA/OSTA	1995	-
FAT32	Microsoft	1996	Windows 95b ^[3]
QFS	LSC Inc, Sun Microsystems	1996	Solaris
GPFS	IBM	1996	AIX, Linux, Windows
Be File System	Be Inc., D. Giampaolo, C. Meurillon	1996	BeOS
HFS Plus	Apple Computer	1998	Mac OS 8.1
NSS	Novell	1998	NetWare 5
PolyServe File System (PSFS)	PolyServe	1998	Windows, Linux
ODS-5	DEC	1998	OpenVMS 7.2
ext3	Stephen Tweedie	1999	Linux
ISO 9660:1999	Ecma International, Microsoft	1999	Microsoft Windows, Linux, Mac OS X, FreeBSD, and AmigaOS
JFS	IBM	1999	OS/2 Warp Server for e-business
GFS	Sistina (Red Hat)	2000	Linux
Melio FS	Sanbolic	2001	Windows
NTFS Version 5.1	Microsoft	2001	Windows XP
ReiserFS	Namesys	2001	Linux
zFS	IBM	2001	z/OS (backported to OS/390)
FATX	Microsoft	2002	Xbox

UFS2	Kirk McKusick	2002	FreeBSD 5.0
Lustre	Sun Microsystems/Cluster File Systems	2002	Linux
OCFS	Oracle Corporation	2002	Linux
VMFS2	VMware	2002	VMware ESX Server 2.0
Fossil	Bell Labs	2003	Plan 9 from Bell Labs 4
Google File System	Google	2003	Linux
Reliance^[4]	Datalight	2003	Windows CE, VxWorks, custom ports
ZFS	Sun Microsystems	2004	Solaris, FreeBSD
Reiser4	Namesys	2004	Linux
Non-Volatile File System	Palm, Inc.	2004	Palm OS Garnet
Minix V3 FS	Andrew S. Tanenbaum	2005	MINIX 3
OCFS2	Oracle Corporation	2005	Linux
NILFS	NTT	2005	Linux, NetBSD
VMFS3	VMware	2005	VMware ESX Server 3.0
GFS2	Red Hat	2006	Linux
ext4	various	2006	Linux
exFAT	Microsoft	2006, 2009	Windows CE 6.0, Windows XP SP3, Windows Vista SP1
TexFAT/TFAT	Microsoft	2006	Windows CE 6.0
NTFS Version 6.0	Microsoft	2006	Windows Vista
Btrfs	Oracle Corporation	2007	Linux
HAMMER	Matthew Dillon	2008	Dragonfly BSD
Oracle ACFS	Oracle Corporation	2009	Linux - Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 only
Reliance Nitro^[4]	Datalight	2009	Windows CE, Windows Mobile, VxWorks, Linux, custom ports

LTFS	IBM	2010	Linux, Mac OS X, planned Microsoft Windows,
IlesfayFS	Ilesfay Technology Group	2011	Microsoft Windows, planned Red Hat Enterprise Linux

2.1 Τα Συστήματα Αρχείων Σήμερα

2.1.1 Συστήματα Αρχείων της Microsoft

Η Microsoft χρησιμοποιεί τα δικά της συστήματα αρχείων τις οικογένειες FAT και NTFS.

Αντίθετα από τα περισσότερα Σ.Α. τα Λ.Σ. Windows χρησιμοποιούν ένα γράμμα δίσκου (drive letter) για να παραστήσουν στο επίπεδο χρήστη τον τόμο ή δίσκο και να τους διαχωρίσουν μεταξύ τους.

Η διαδρομή C:\Windows, αναφέρεται στον φάκελο Windows, στην ρίζα του τόμου με την ονομασία C:\.

Επειδή μάλιστα η ονομασία αυτή κατά παράδοση ανατίθεται στον πρώτο (ή μοναδικό) τόμο αποθήκευσης του Λ.Σ. από τον οποίο συνήθως γίνεται η εκκίνηση του Λ.Σ., αυτό έχει οδηγήσει σε αρκετά bugs ειδικά με παλιά προγράμματα που υποθέτουν λανθασμένα σε κάθε σύστημα που θα εγκατασταθούν ότι αυτός είναι ο τόμος εκκίνησης.

2.1.1 Συστήματα Αρχείων στο Linux

Το Linux έχει τα δικά του συστήματα αρχείων, τα XFS, JFS, ReiserFS, btrfs και τα δημοφιλέστερα όλων, την οικογένεια Ext(2-4), που είναι μετεξέλιξη των αρχικών Σ.Α. του Unix.

Το δημοφιλέστερο Σ.Α. στο Linux σήμερα, το Ext2 κατασκευάστηκε αποκλειστικά για το Linux από τον Remy Card, ως αντικατάσταση του extended file system (ext).

Το Ext κατασκευάστηκε τον Απρίλιο του 1992 και ήταν το πρώτο Σ.Α. που δημιουργήθηκε αποκλειστικά για το Linux. Περιείχε δυνατότητα προσάρτησης μεταδεδομένων στο αρχείο (κάτι που λείπει από τα Σ.Α. της Microsoft), ευθεία κληρονομιά από το Unix.

ΕΠΙΛΟΓΟΣ

Σήμερα υπάρχει μία πληθώρα Συστημάτων Αρχείων, τα περισσότερα από αυτά δημιουργημένα για συγκεκριμένο Λειτουργικό Σύστημα. Πολλά από αυτά χρησιμοποιούνται από πολλά άλλα Λειτουργικά Συστήματα και ακόμη περισσότερα μπορούν να αναγνώσουν ή και να τροποποιήσουν δεδομένα κάτω από ακόμη περισσότερα Λειτουργικά Συστήματα.

Στα επόμενα κεφάλαια θα παρατεθούν λεπτομερώς δύο δημοφιλείς οικογένειες Συστημάτων Αρχείων, το FAT(16,32) και το ext(2)

ΚΕΦΑΛΑΙΟ 3

ΤΟ ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ FAT

ΕΙΣΑΓΩΓΗ

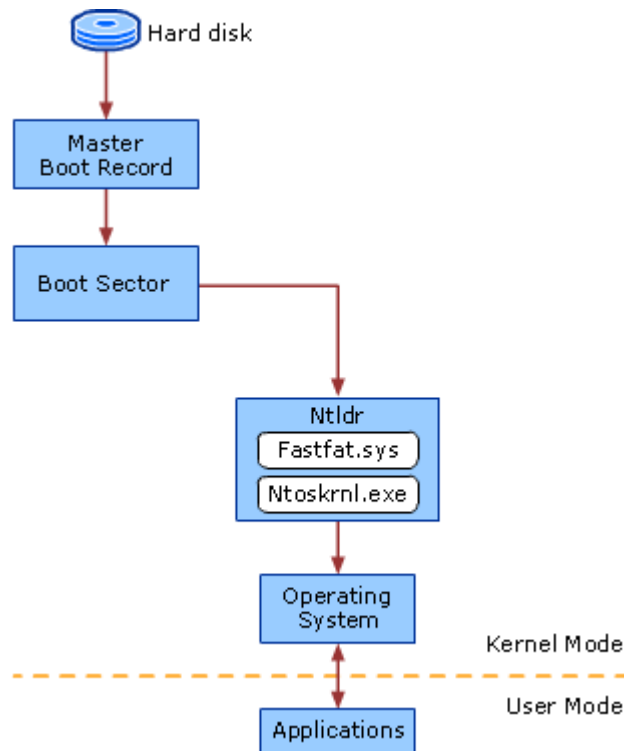
Το FAT16, FAT32 είναι Σ.Α. που χρησιμοποιούνται αποκλειστικά σε Λ.Σ. της Microsoft. Είναι μετεξελίξεις του FAT που δημιουργήθηκε για να υποστηρίξει το MS-DOS.

Γενικώς τα Σ.Α. της Microsoft είναι μάλλον απλοϊκά και τους λείπουν προηγμένες δυνατότητες που έχουν άλλα σύγχρονα Σ.Α. ειδικά στην κατηγορία των μεταδεδομένων.

3.1: Η αρχιτεκτονική του FAT

Κατά τη διάρκεια της διαμόρφωσης και εγκατάστασης ενός συστήματος αρχείων ενός τόμου ενός σκληρού δίσκου, δημιουργείται ένας απαραίτητος τομέας εκκίνησης, ή master boot record (MBR). Το MBR περιέχει ένα μικρό κομμάτι εκτελέσιμου κώδικα που ονομάζεται κώδικας εκκίνησης, καθώς και ένα πίνακα διαμέρισης (κατατμήσεων, partition table) του φυσικού δίσκου.

Όταν ένας τόμος ενός δίσκου είναι προσαρτημένος (mounted) τότε το MBR εκτελεί τον κώδικα εκκίνησης και μεταφέρει τον έλεγχο στον τομέα εκκίνησης του δίσκου, επιτρέποντας έτσι στον Η/Υ να ξεκινήσει το Λ.Σ. του συστήματος αρχείων του συγκεκριμένου τόμου. Το Σχήμα 1 δείχνει την λογική αυτής της διαδικασίας:



Σχήμα 1: Η φυσική διαστρωμάτωση του FAT (Microsoft, 2003a)

3.2: Τα συστατικά του FAT

Τα συστατικά του FAT εμφανίζονται στον παρακάτω πίνακα:

Πίνακας 2: Περιεχόμενα του Σ.Α. FAT (Microsoft, 2003a)

Περιγραφή Περιεχομένων FAT (FAT Component Description)	
Hard disk, Σκληρός Δίσκος	Περιέχει μία ή περισσότερες κατατμήσεις (partitions)
Boot sector, Τομέας Εκκίνησης	Κατάτμηση με δυνατότητα εκκίνησης. Διατηρεί αποθηκευμένη πληροφορία για την διάρθρωση του τόμου, και το Σ.Α. όπως και για τον απαραίτητο κώδικα εκκίνησης (boot code)
Master Boot Record	Περιέχει εκτελέσιμο κώδικα τον οποίο το BIOS φορτώνει στην μνήμη. Ο κώδικας ελέγχει το MBR για να βρει τον πίνακα με τις κατατμήσεις (partition table) ώστε να βρει ποια από αυτές είναι "ενεργή", δηλαδή μπορεί να ξεκινήσει από αυτή κάποιο Λ.Σ.
Ntldr.dll	Αναγκάζει την ΚΜΕ σε ειδική λειτουργία προστασίας (protected mode) εκκινεί το Σ.Α. και μετά διαβάζει τα περιεχόμενα του αρχείου boot.ini. Η πληροφορία που βρίσκει καθορίζει τις επιλογές στην εκκίνηση και την επίδειξη πιθανού μενού επιλογής εκκίνησης (π.χ. επιλογή μεταξύ διαφορετικών Λ.Σ.
Fastfat.sys	Απαραίτητος "οδηγός" συσκευών (system driver) για FAT16 ή FAT32
Ntoskrnl.exe	Επιλεγεί τις πληροφορίες σχετικά με ποιους "οδηγούς συστήματος" (device drivers) θα φορτωθούν και με ποια σειρά
Kernel Mode	Η διαδικασία η οποία επιτρέπει σε κομμάτια κώδικα να έχουν απευθείας πρόσβαση σε όλο το υλικό (hardware) του Η/Υ καθώς και της μνήμης του συστήματος
User Mode	Κατάσταση λειτουργίας κατά την οποία επιτρέπεται η εκκίνηση εφαρμογών (προφανώς από τον χρήστη)

3.3: Η φυσική δομή του FAT

3.3.1: Clusters & Sectors στο FAT

Cluster ονομάζουμε το μικρότερο δυνατό τμήμα ενός δίσκου στο οποίο μπορεί να αποθηκευτεί ένα αρχείο. Η οικογένεια των Λ.Σ. Windows χρησιμοποιεί το μέγεθος του cluster για να προσδιορίσει το πλήθος των sector (Τομέας) που περιέχει το κάθε cluster.

Αν ο δίσκος χρησιμοποιεί τομείς των 512 byte, τότε ένα cluster 512 byte περιέχει (ή πιο σωστά καταλαμβάνει) έναν τομέα, ενώ αντίστοιχα αν ένα cluster εκτείνεται σε 8 τομείς, τότε το cluster καταλαμβάνει 4.048 Byte (δηλαδή 8x512Byte).

Ο Η/Υ κατά την διάρκεια της εκκίνησης προσπελαύνει συγκεκριμένο χώρο στον σκληρό δίσκο (MBR) για να «αποφασίσει» ποιο Λ.Σ. θα εκκινήσει και που βρίσκεται αποθηκευμένη η πληροφορία για την λογική διάρθρωση του δίσκου και των κατατμήσεων του (partition table). Τα δεδομένα που αποθηκεύονται στο MBR διαφέρουν ανάλογα με τα Λ.Σ. αλλά και την πλατφόρμα του Η/Υ.

3.3.2: Αριθμοδότηση Clusters

Το Cluster αριθμοδοτείται διαδοχικά από την αρχή της κατάτμησης. Επειδή στο FAT τα clusters “ξεκινούν» μετά από το τμήμα του δίσκου που είναι αφιερωμένο στις παραμέτρους για το BIOS (BIOS Parameter Block , BPB), δεν μπορεί να ευθυγραμμιστεί η αριθμοδότηση όλων των ίδιων cluster σε φυσικό επίπεδο

3.3.3: Περιορισμοί του μεγέθους του Cluster στο FAT

Επειδή το FAT και το FAT32 χρησιμοποιούν διαφορετικό μέγεθος cluster το οποίο εξαρτάται από το μέγεθος του λογικού δίσκου, κάθε Σ.Α. έχει ένα μέγιστο μέγεθος cluster που μπορεί να υποστηρίξει.

Το μικρότερο μέγεθος cluster έχει αντίκτυπο στην αποτελεσματική χρήση του μεγέθους ενός σκληρού δίσκου αφού λιγότερος χώρος παραμένει ανεκμετάλλετος (αφού το μέγεθος του cluster καθορίζει και τον ελάχιστο χώρο χρήσης για την αποθήκευση ενός αρχείου). Το πλήθος των cluster που μπορεί να υποστηρίξει ένα Σ,Α. έχει άμεσο αντίκτυπο στο μέγεθος ενός λογικού δίσκου που θα μπορεί να χρησιμοποιηθεί από το Λ.Σ.

Όπως αναφέρθηκε, το μέγεθος του cluster είναι υποχρεωτικά πολλαπλάσιο του μεγέθους του τομέα (sector), αφού το cluster είναι το απαραίτητο ελάχιστο μέγεθος προσπέλασης του πλήθους των τομέων σε ένα φυσικό δίσκο.

Το αριθμητικό μέρος στις ονομασίες των FAT16 και FAT32 αναφέρεται στο πλήθος των bit που χρησιμοποιεί για την διευθυνσιοδότηση των clusters ως εξής:

- Το FAT16 χρησιμοποιεί 16-bit μήκος διεύθυνσης στο αντίστοιχο πίνακα διευθύνσεων του Σ.Α. (File Allocation Table) δηλαδή 2¹⁶ clusters.

- Το FAT32 χρησιμοποιεί 32- μήκος διεύθυνσης. Επειδή όμως το FAT32 δεσμεύει τα πρώτα 4 bits της διεύθυνσης στον πίνακα διευθύνσεων, το FAT32 έχει ένα μέγιστο μέγεθος 228 clusters.
- Τα floppy disks πάντα διαμορφώνονται σε FAT

Ο πίνακας που παρατίθεται, αναφέρει τα μεγέθη ενός λογικού δίσκου και τα αντίστοιχα μεγέθη σε FAT & FAT32:

Πίνακας 3: Μέγεθος Cluster σε συνάρτηση με το μέγεθος δίσκου

Μέγεθος Λογικού Τόμου	Μέγεθος FAT16 Cluster	Μέγεθος FAT32 Cluster
7 MB–16 MB	2 KB	Δεν υποστηρίζεται
17 MB–32 MB	512 bytes	Δεν υποστηρίζεται
33 MB–64 MB	1 KB	512 bytes
65 MB–128 MB	2 KB	1 KB
129 MB–256 MB	4 KB	2 KB
257 MB–512 MB	8 KB	4 KB
513 MB–1,024 MB	16 KB	4 KB
1,025 MB–2 GB	32 KB	4 KB
2 GB–4 GB	64 KB	4 KB
4 GB–8 GB	Δεν υποστηρίζεται	4 KB
8 GB–16 GB	Δεν υποστηρίζεται	8 KB
16 GB–32 GB	Δεν υποστηρίζεται	16 KB
32 GB–2 TB	Δεν υποστηρίζεται	Δεν υποστηρίζεται

Εδώ να σημειωθεί ότι το FAT32 δεν υποστηρίζει μέγεθος μεγαλύτερο από 32 GB. Για μεγαλύτερα μεγέθη λογικού δίσκου, το σύστημα αρχείων πρέπει να είναι αναγκαστικά NTFS.

3.3.4: Ανώτατα όρια λογικού δίσκου στο FAT

Αν δεν γίνει επιλογή του μεγέθους του cluster από τον χρήστη κατά την διάρκεια της διαμόρφωσης (format) του λογικού τόμου, θα χρησιμοποιηθούν οι τιμές προεπιλογής. Για την αλλαγή του μεγέθους του cluster, απαιτείται η επαναδιαμόρφωση του λογικού τόμου (με φυσικό επακόλουθο την διαγραφή και απώλεια των δεδομένων).

Τα FAT16 & FAT32 έχουν τους κάτωθι περιορισμούς:

- Λογικά μεγέθη μικρότερα από 16MB αναγκαστικά διαμορφώνονται ως FAT12
- Λογικοί δίσκοι μεγαλύτεροι από 2GB δεν είναι προσπελάσιμοι από Η/Υ με Λ.Σ. MS-DOS, Windows 95, Windows 98, Windows Millennium Edition (Me), αλλά και αρκετά άλλα Λ.Σ. ο περιορισμός αυτός προκύπτει από το ότι αυτά τα Λ.Σ. δεν υποστηρίζουν μέγεθος cluster μεγαλύτερο από 32 KB το οποίο έχει ως αποτέλεσμα το όριο των 2GB
- Θεωρητικά ένας λογικός τόμος με FAT32 μπορεί να έχει μέγεθος μέχρι 8 TeraBytes. Παρόλα αυτά, τα Λ.Σ. της Microsoft λόγω άλλων περιορισμών μπορούν να διαμορφώσουν σε FAT32 μεγέθη μέχρι 32 GB. Μετά πρέπει αναγκαστικά να χρησιμοποιηθεί το NTFS. Να σημειωθεί ότι τα Λ.Σ. της Microsoft μπορούν να προσπελάσουν και να διαχειριστούν μεγαλύτερους λογικούς τόμους που έχουν διαμορφωθεί από άλλα Λ.Σ.
- Το μεγαλύτερο μέγεθος αρχείου για το FAT32 είναι 4GB μείον ένα byte. Το FAT32 περιέχει 4 byte για κάθε cluster στον πίνακα διευθύνσεων. Αντίστοιχα το FAT16 περιέχει 2 byte στον ίδιο πίνακα, ενώ το FAT12 περιέχει 1,5 byte. Ένας λογικός τόμος με FAT32 πρέπει να περιέχει τουλάχιστον 65.527 clusters.

Το FAT16 υποστηρίζει μάξιμουμ 65.527 clusters ανά λογικό τόμο. Ο παρακάτω πίνακας παρουσιάζει τους περιορισμούς του FAT16:

Πίνακας 4: Περιορισμοί του FAT16

Περιγραφή	Περιορισμός
Ανώτερο μέγεθος αρχείου	4 GB μείον 1 byte (232 bytes μείον 1 byte)
Ανώτερο μέγεθος τόμου	4 GB
Αρχεία ανά τόμο	Περίπου 65,536
Ανώτερο πλήθος αρχείων και φακέλων στον γονικό φάκελο (root folder)	512 (Αν χρησιμοποιηθούν μεγάλα ονόματα αρχείων, αυτός ο αριθμός μπορεί να μειωθεί)

Το FAT32 πρέπει να έχει μίνιμουμ 65.527 clusters. Όπως αναφέρθηκε, τα Λ.Σ. της Microsoft μπορούν να διαμορφώσουν λογικούς δίσκους μέχρι 32 GB, μπορούν όμως να διαχειριστούν μεγαλύτερα μεγέθη δίσκων. Οι περιορισμοί του FAT32 περιγράφονται στον παρακάτω πίνακα:

Πίνακας 5: Περιορισμοί του FAT32

Περιγραφή	Περιορισμός
Ανώτερο μέγεθος αρχείου	4 GB μείον 1 byte (232 bytes μείον 1 byte)
Ανώτερο μέγεθος τόμου	32 GB
Αρχεία ανά τόμο	4,177,920
Ανώτερο πλήθος αρχείων και φακέλων σε έναν οποιοδήποτε φάκελο	65,534 (Αν χρησιμοποιηθούν μεγάλα ονόματα αρχείων, αυτός ο αριθμός μπορεί να μειωθεί σημαντικά)

3.3.5: Φυσική Οργάνωση του FAT

Boot Sector	Reserved Sectors	FAT 1	FAT 2 (Duplicate)	Root Folder	Other Folders and All Files
-------------	------------------	-------	-------------------	-------------	-----------------------------

Σχήμα 2: Λογική διάρθρωση ενός τόμου με FAT

Τα περιεχόμενα της λογικής διάρθρωσης του FAT στο Σχήμα 2, περιγράφονται στον παρακάτω πίνακα:

Πίνακας 6: Περιεχόμενα ενός τόμου διαμορφωμένου με FAT

Στοιχείο	Περιγραφή
Τομέας Εκκίνησης Boot Sector	Περιέχει τις παραμέτρους του BIOS που περιγράφουν την διάρθρωση του λογικού τόμου και την δομή του Σ.Α. όπως και τον απαραίτητο κώδικα εκκίνησης (boot code)
Τομείς Προ-κατελλημένοι Reserved Sectors	Το πλήθος των τομέων που προηγούνται του πρώτου διαθέσιμου sector (Σε αυτούς συμπεριλαμβάνεται και ο τομέας εκκίνησης)
FAT 1	Αρχικό FAT.
FAT 2 (Duplicate)	Αντίγραφο ασφαλείας του FAT
Γονικός φάκελος Root folder	Περιγράφει τον γονικό φάκελο στην ρίζα (root) της λογικής κατάτμησης (partition)
Άλλοι φάκελοι και αρχεία Other folders and all files	Περιέχονται όλοι οι φάκελοι και τα αρχεία του Σ.Α. (δηλαδή όλα τα δεδομένα που είναι αποθηκευμένα στον δίσκο) Contains the data for the files and folders within the file system.

3.3.6: Τομέας Εκκίνησης & Εκκίνηση Δίσκου

Σε δίσκους (λογικούς ή φυσικούς) που περιέχουν δυνατότητα εκκίνησης (MBR, Master Boot Record), ο τομέας εκκίνησης που είναι εγκαταστημένος στο πρώτο λογικό τομέα της κάθε κατάτμησης, είναι ένα σημαντικό και απαραίτητο κομμάτι για την εκκίνηση του Η/Υ. εμπεριέχει εκτελέσιμο κώδικα και τα δεδομένα που χρειάζεται αυτός, συμπεριλαμβανομένης και της πληροφορίας που χρησιμοποιεί το Σ.Α. για να προσπελάσει αυτόν τον τόμο. Ο τομέας εκκίνησης δημιουργείται κατά την διάρκεια διαμόρφωσης του τόμου. Στο τέλος του τομέα εκκίνησης βρίσκεται πάντα μία δομή μεγέθους 2 byte, αποκαλούμενη και «λέξη υπογραφής», signature word, η οποία είναι πάντα ίση με 0x55AA.

Εάν υπάρχουν πολλαπλά Λ.Σ. στον ίδιο δίσκο, τότε υπάρχει η δυνατότητα φόρτωσης του λεγόμενου επιλογέα λειτουργικού συστήματος, μέσω του Ntldr.exe. αν δεν υπάρξει παρέμβαση ή επιλογή του χρήστη εντός προαποφασισμένου χρονικού ορίου, το προεπιλεγμένο Λ.Σ. ξεκινάει.

Ο τομέας εκκίνησης ενός δίσκου, αποτελείται από τα εξής στοιχεία:

- Μία εντολή jump για την ΚΜΕ (x86-based CPU jump instruction)
- Το αναγνωριστικό του κατασκευαστή (OEM ID).
- Το BIOS parameter block (BPB)
- Το εκτεταμένο (BPB)
- Τον απαραίτητο κώδικα εκτέλεσης (executable boot code ή bootstrap code) ο οποίος εκκινεί το Λ.Σ.

3.3.7: Περιεχόμενα του Τομέα Εκκίνησης

Ο MBR οδηγεί την ΚΜΕ να εκτελέσει τον κώδικα που βρίσκεται αποθηκευμένος στον τομέα εκκίνησης. Για αυτό τα 3 πρώτα byte του τομέα εκκίνησης πρέπει να είναι σωστά και να είναι έγκυρη εντολή εκτέλεσης x86. Πρέπει απαραίτητα να περιέχει και μία εντολή jump που παραλείπει τα επόμενα byte που δεν είναι εκτελέσιμος κώδικας.

Μετά την εντολή εκτέλεσης βρίσκεται ο κωδικός του κατασκευαστή (OEM ID) μεγέθους 8 byte, μία συμβολοσειρά που αποτυπώνει μοναδικά το όνομα και

την έκδοση του Λ.Σ. το οποίο διαμόρφωσε τον λογικό τόμο. Όσον αφορά στην Microsoft, κάποια Λ.Σ. της καταγράφουν σε αυτό το πεδίο το ίδιο όνομα "MSDOS5.0" για λόγους συμβατότητας. Επίσης μπορούν να καταγράψουν "MSWIN4.0 και "MSWIN4.1" όταν η διαμόρφωση έχει γίνει από Windows 95, και από Windows 98 ή Windows ME αντίστοιχα.

Μετά την συμβολοσειρά OEM υπάρχει το BPB, το οποίο παρέχει πληροφορίες στον εκτελέσιμο κώδικα να ανιχνεύσει το Ntldr. Το BPB ξεκινάει πάντα στο ίδιο offset έτσι ώστε γνωστές εκ των προτέρων παράμετροι να βρίσκονται πάντα στο ίδιο μέρος. Το μέγεθος του δίσκου και η γεωμετρία του είναι ενθυλακωμένα στο BPB. Επειδή το πρώτο μέρος του boot sector είναι μία εντολή jump, το BPB μπορεί να επεκταθεί το μέλλον, περιέχοντας επιπλέον πληροφορίες (και απλώς η εντολή jump θα μεγαλώσει το διάστημα που θα αγνοεί).

3.3.8: Βασικές Διαφορές FAT16 & FAT32

Το FAT16 αποδίδει καλύτερα σε μικρούς δίσκους με απλή διάρθρωση φακέλων.

Επειδή το FAT16 χρησιμοποιεί μεγαλύτερο μέγεθος cluster, ένας τόμος που το χρησιμοποιεί δεν διαχειρίζεται αποτελεσματικά το χώρο του δίσκου όταν υπάρχει πληθώρα μικρών αρχείων αποθηκευμένων σε αυτόν. Είναι προφανές ότι ένα μικρό αρχείο θα παρακρατήσει όλο το μέγεθος ενός cluster ακόμη και αν δεν το χρειάζεται. Για αυτό και το FAT δεν προτείνεται για δίσκους μεγαλύτερους από 511MB, ενώ δεν μπορεί να χρησιμοποιηθεί σε δίσκους μεγαλύτερους των 4GB.

Πίνακας 7: Υποστήριξη εκδόσεων FAT από Λ.Σ. Microsoft (Microsoft, 2003b)

Λειτουργικό Σύστημα (O.S.)	FAT16	FAT32
Windows XP	*	*
Windows Server 2003	*	*
Windows 2000	*	*
Windows NT 4.0	*	
Windows 95 OSR2, Windows 98,	*	*

Windows Me	
Windows 95 (prior to OSR2)	*
MS-DOS	*

Η μεγαλύτερη διαφορά του FAT32 από το FAT16 είναι το απώτερο πλήθος cluster που υποστηρίζει, το οποίο με την σειρά του επιδρά στο απώτερο μέγεθος δίσκου και αποδοτικής αποθήκευσης. Το FAT32 υποστηρίζει δίσκους πολύ μεγαλύτερους των 4GB που υποστηρίζει το FAT16. Επίσης μπορεί να χρησιμοποιήσει πολύ μικρότερο μέγεθος cluster κάνοντας πολύ αποδοτική χρήση του χώρου αποθήκευσης. Π.χ. σε ένα δίσκο (ή τομέα) μεταξύ 2-4GB ενώ το FAT16 χρησιμοποιεί 64-KB cluster, το FAT32 σε δίσκους μεταξύ 16 GB - 32 GB χρησιμοποιεί 16-KB cluster.

Το μεγαλύτερο μέγεθος αρχείου στο FAT32 είναι 4 GB μείον 1 byte.

3.3.9: Διαδικασία Διαμόρφωσης τόμου στο FAT

Κατά την διάρκεια της διαμόρφωσης, είναι δυνατόν να τροποποιηθεί το μέγεθος του cluster. Επίσης γίνεται έλεγχος της ακεραιότητας όλων των τομέων, εκτός και αν έχει επιλεγεί το quick format. σε κάθε περίπτωση δημιουργείται η απαραίτητη δομή του συστήματος αρχείων.

Η διαδικασία μετατροπής από FAT16 σε FAT32 είναι μονής κατεύθυνσης.

3.3.10: Επιπτώσεις του Κατακερματισμού στην Απόδοση

Η έλλειψη χώρου στον δίσκο μπορεί να οδηγήσει σε σημαντική πτώση της απόδοσης, ειδικά όταν πολλοί χρήστες έχουν πρόσβαση στο κοινό μέσο και εργάζονται πάνω σε αρχεία.

Ο κατακερματισμός εμφανίζεται όταν οι cluster που εμπεριέχουν τα περιεχόμενα ενός αρχείου δεν βρίσκονται σε φυσική συνέχεια. Επειδή έτσι το αρχείο βρίσκεται διασκορπισμένο παντού μέσα στο δίσκο, η πρόσβαση σε αυτό καθυστερεί, επηρεάζοντας την απόδοση του Λ.Σ. όταν το Σ.Α. δεν μπορεί να βρει συνεχόμενους clusters για να αποθηκεύσει ένα αρχείο, θα χρησιμοποιήσει τους κοντινότερα διαθέσιμους.

Με τον από-κατακερματισμό, μεταφέρονται τα περιεχόμενα του κάθε αρχείου σε διπλανούς cluster βελτιώνοντας έτσι την απόδοση του συστήματος γενικότερα.

3.3.11: Πίνακας Καταλόγων στο FAT (Root Directory)

Ο πίνακας αρχείων του FAT είναι ο τομέας που αποθηκεύονται, ως μια 32-byte καταχώρηση, κάθε αρχείο ή κατάλογος της κατάτμησης. Κάθε καταχώρηση περιέχει το όνομα, την κατάληξη, τις ιδιότητες (κρυφό, μόνο για ανάγνωση, τόμος, κατάλογος), την ημερομηνία και την ώρα της δημιουργίας του, την διεύθυνση του πρώτου sector του αρχείου ή του καταλόγου και τέλος το μέγεθος του.

Στον πίνακα παρακάτω μπορούμε να δούμε την μορφή αποθήκευσης των πληροφοριών ενός αρχείου ή ενός καταλόγου :

Πίνακας 8: Δομή αποθήκευσης πληροφοριών FAT από Λ.Σ. Microsoft (Microsoft, 2003b)

Byte Offset	Μήκος (bytes)	Περιγραφή
0x00	8	Όνομα αρχείου DOS
		Το πρώτο byte μπορεί να περιέχει τις παρακάτω ειδικές τιμές:
		0x00 Η καταχώρηση είναι κενή και ελεύθερη.
		0x05 Ο αρχικός χαρακτήρας είναι το 0xE5.
		0x2E 'Τελεία'. εναλλακτικά '.' ή '..'
0xE5 Η καταχώρηση ήταν σε χρήση αλλά έχει διαγραφεί και τώρα είναι διαθέσιμη.		

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

0x08	3	Κατάληξη DOS																											
0x0B	1	Ιδιότητες αρχείου																											
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Περιγραφή</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0x01</td> <td>Μόνο για Ανάγνωση</td> </tr> <tr> <td>1</td> <td>0x02</td> <td>Κρυφό</td> </tr> <tr> <td>2</td> <td>0x04</td> <td>Συστήματος</td> </tr> <tr> <td>3</td> <td>0x08</td> <td>Ετικέτα Τόμου</td> </tr> <tr> <td>4</td> <td>0x10</td> <td>Υποκατάλογος</td> </tr> <tr> <td>5</td> <td>0x20</td> <td>Archive</td> </tr> <tr> <td>6</td> <td>0x40</td> <td>Συσκευή (για εσωτερική χρήση μόνο)</td> </tr> <tr> <td>7</td> <td>0x80</td> <td>Αχρησιμοποίητο</td> </tr> </tbody> </table>	Bit	Mask	Περιγραφή	0	0x01	Μόνο για Ανάγνωση	1	0x02	Κρυφό	2	0x04	Συστήματος	3	0x08	Ετικέτα Τόμου	4	0x10	Υποκατάλογος	5	0x20	Archive	6	0x40	Συσκευή (για εσωτερική χρήση μόνο)	7	0x80	Αχρησιμοποίητο
		Bit	Mask	Περιγραφή																									
		0	0x01	Μόνο για Ανάγνωση																									
		1	0x02	Κρυφό																									
		2	0x04	Συστήματος																									
		3	0x08	Ετικέτα Τόμου																									
		4	0x10	Υποκατάλογος																									
		5	0x20	Archive																									
		6	0x40	Συσκευή (για εσωτερική χρήση μόνο)																									
7	0x80	Αχρησιμοποίητο																											
<p>Η τιμή 0x0F χρησιμοποιείται για να υποδείξει μια καταχώρηση αρχείου με μεγάλο όνομα.</p>																													
0x0C	1	Δεσμευμένο, δύο bit χρησιμοποιούνται από την έκδοση NT και όλες τις νεότερες για την κωδικοποίηση πληροφοριών.																											
0x0D	1	Ημερομηνία δημιουργίας με μεγαλύτερη λεπτομέρεια.																											
0x0E	2	<p>Ημερομηνία δημιουργίας. Η ώρα, τα λεπτά και τα δευτερόλεπτα αναπαριστούνται σύμφωνα με τα παρακάτω:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Περιγραφή</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Bits	Περιγραφή																									
Bits	Περιγραφή																												

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

		<table border="1"> <tr> <td>15-11</td> <td>Ωρα (0-23)</td> </tr> <tr> <td>10-5</td> <td>Λεπτά (0-59)</td> </tr> <tr> <td>4-0</td> <td>Δευτερόλεπτα/2 (0-29)</td> </tr> </table> <p>Να σημειωθεί ότι τα δευτερόλεπτα καταγράφονται αν 2. Για αποτελέσματα με μεγαλύτερη ανάλυση απευθυνθείτε στο offset 0x0d.</p>	15-11	Ωρα (0-23)	10-5	Λεπτά (0-59)	4-0	Δευτερόλεπτα/2 (0-29)		
15-11	Ωρα (0-23)									
10-5	Λεπτά (0-59)									
4-0	Δευτερόλεπτα/2 (0-29)									
0x10	2	<p>Ημερομηνία δημιουργίας. Η χρονιά , ο μήνας και η μέρα αναπαριστούνται σύμφωνα με τα παρακάτω:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Περιγραφή</th> </tr> </thead> <tbody> <tr> <td>15-9</td> <td>Χρόνος (0 = 1980, 127 = 2107)</td> </tr> <tr> <td>8-5</td> <td>Μήνας (1 - 12)</td> </tr> <tr> <td>4-0</td> <td>Ημέρα (1 - 31)</td> </tr> </tbody> </table>	Bits	Περιγραφή	15-9	Χρόνος (0 = 1980, 127 = 2107)	8-5	Μήνας (1 - 12)	4-0	Ημέρα (1 - 31)
Bits	Περιγραφή									
15-9	Χρόνος (0 = 1980, 127 = 2107)									
8-5	Μήνας (1 - 12)									
4-0	Ημέρα (1 - 31)									
0x12	2	Τελευταία ημερομηνία πρόσβασης.								
0x14	2	Το ευρετήριο EA (χρησιμοποιείτε στο OS/2 και στο NT) στο FAT12 και FAT16, τα πρώτα 2 bytes του αρχικού cluster στο FAT32								
0x16	2	Τελευταία ώρα τροποποίησης.								
0x18	2	Τελευταία ημερομηνία τροποποίησης.								
0x1A	2	Έναρξη του αρχείου σε clusters στο FAT12 και FAT16. Τα τελευταία 2 bytes του πρώτου cluster στο FAT32. Καταχωρήσεις της σημαίας του ονόματος τόμου, του ριζικού υποκαταλόγου "." στο FAT12/16, και των κενών αρχείων με μηδενικό μέγεθος θα πρέπει να έχουν το πρώτο cluster μηδενικό.								
0x1C	4	Μέγεθος του αρχείου σε bytes. Καταχωρήσεις της ετικέτας τόμου ή της σημαίας								

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

		του υποκαταλόγου θα πρέπει να έχουν μηδενικό μέγεθος.
--	--	---

Παρακάτω βλέπουμε ένα παράδειγμα από μία δισκέτα MSDOS με 6 καταχωρήσεις :

```

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
0009728 49 4f 20 20 20 20 20 20 53 59 53 27 00 00 00 00  IO      .SYS
0009744 00 00 00 00 00 00 08 5d 62 1b 1d 00 16 9f 00 00
0009760 4d 53 44 4f 53 20 20 20 53 59 53 27 00 00 00 00  MSDOS    .SYS
0009776 00 00 00 00 00 00 08 5d 62 1b 6d 00 38 95 00 00
0009792 43 4f 4d 4d 41 4e 44 20 43 4f 4d 20 00 00 00 00  COMMAND  .COM
0009808 00 00 00 00 00 00 07 5d 62 1b b8 00 39 dd 00 00
0009824 44 42 4c 53 50 41 43 45 42 49 4e 27 00 00 00 00  DBLSPACE .BIN
0009840 00 00 00 00 00 00 08 5d 62 1b 27 01 f6 fc 00 00
0009856 4d 53 44 4f 53 20 20 20 20 20 20 28 00 00 00 00  MSDOS
0009872 00 00 00 00 00 00 1a 88 99 1c 00 00 00 00 00 00
0009888 46 44 49 53 4b 20 20 20 45 58 45 20 00 00 00 00  FDISK    .EXE
0009904 00 00 00 00 00 00 36 59 62 1b 02 00 17 73 00 00

```

Για την πρώτη καταχώρηση για παράδειγμα έχουμε :

```

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
0009728 49 4f 20 20 20 20 20 20 53 59 53 27 00 00 00 00  IO      .SYS
      16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0009744 00 00 00 00 00 00 08 5d 62 1b 1d 00 16 9f 00 00

```

Bytes	Περιγραφή
0-7	Όνομα αρχείου (8 bytes) [49 4f 20 20 20 20 20 20] -> IO
8-10	Κατάληξη αρχείου (3 bytes) [53 59 53] -> SYS
11	Ιδιότητες (0x27) ₁₆ => (00100111) ₂ -> <ul style="list-style-type: none"> • 0 byte = Μόνο για ανάγνωση • 1 byte = Κρυφό • 2 byte = Αρχείο συστήματος • 5 byte = Archive
12-21	Δεσμευμένα [00 00 00 00 00 00 00 00 00 00]
22-23	Ωρα (5/6/5 bits, ώρα/λεπτά/(δευτ./2)) (0x5d08) ₁₆ = (01011:101000:01000) ₂ -> 11:40:08
24-25	Ημερομηνία (7/4/5 bits, χρονιά>1980/μήνας/ημέρα) (0x1b62) ₁₆ = (0001101/1011/00010) ₂ -> 1993/11/02
26-27	Αρχικό cluster -> (0x001d) ₁₆ = (11101) ₂
28-31	Μέγεθος αρχείου σε bytes (0000-9f16) ₁₆ = (1001111100010110) ₂ -> 40726 bytes -> 39.77 kbytes
....	
....	

3.3.12: Ονομασία Αρχείων στο FAT

Επιτρεπόμενοι χαρακτήρες είναι όλοι οι παρακάτω :

- Κεφαλαία γράμματα
- Αριθμοί
- Ο κενός χαρακτήρας (θα ήταν προτιμότερο να αποφεύγεται να χρησιμοποιείται στο μέσον μίας λέξης)
- ! # \$ % & ' () - @ ^ _ ` { } ~
- Και οι ASCII τιμές 128–255

Μη επιτρεπόμενοι χαρακτήρες είναι οι παρακάτω :

- " * / : < > ? \ |
 - + , . ; = []
- Δεν επιτρέπονται σε μεγάλα ονόματα αρχείων.
- Τα μικρά γράμματα.
Καταχωρούνται ως κεφαλαία. Επιτρέπονται σε μεγάλα ονόματα αρχείων.
 - Οι χαρακτήρες ελέγχου ASCII 0–31
 - Η ASCII τιμή 127 (DEL)

3.3.13: Εκτεταμένη Ονοματοδοσία Αρχείων στο FAT

Η (εκτεταμένη) ονοματοδοσία των αρχείων στο FAT γίνεται έτσι ώστε να μην επηρεάζεται η αρχική (τυπική) ονοματοδοσία του 8.3 του MS-DOS.

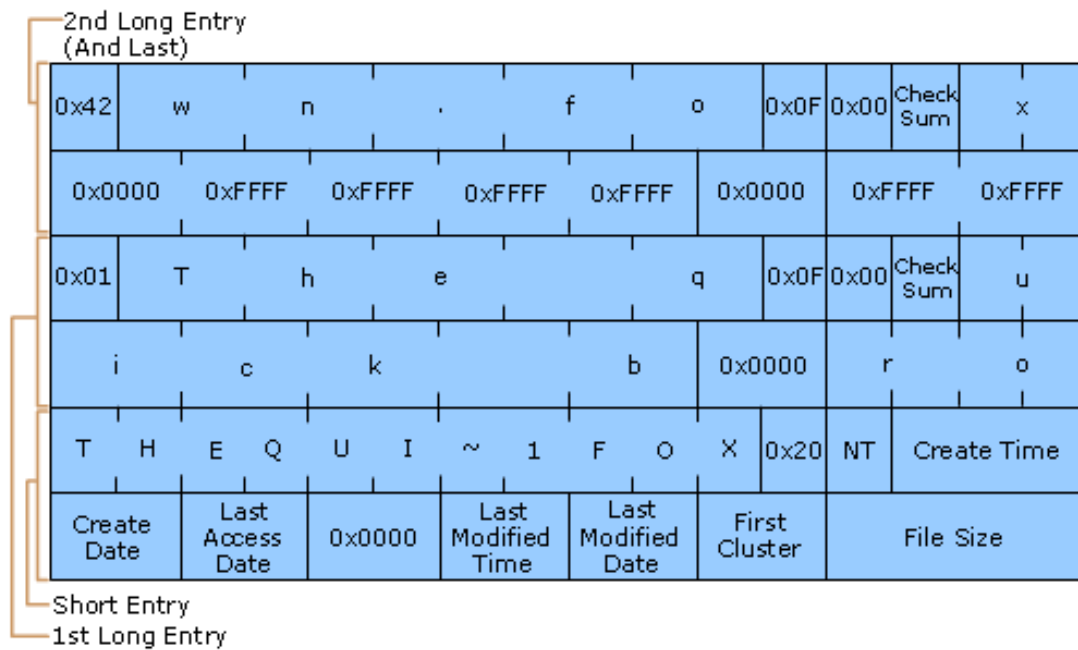
Όταν δημιουργείται ένα αρχείο με «μακρύ» όνομα, δημιουργείται ένα κανονικό όνομα 8.3 και μία ή περισσότερες καταχωρήσεις φακέλου για το αρχείο, οι οποίες έχουν μέγεθος 13 χαρακτήρες η κάθε μία, και αποθηκεύουν έτσι το υπόλοιπο του ονόματος του αρχείου σε Unicode. Οι φάκελοι αυτοί έχουν παραμετροποιημένο το byte ιδιότητας σε ReadOnly Hidden, System & Volume, έτσι ώστε να αγνοούνται από το MS-DOS.

Για αρχεία με εκτεταμένη ονοματοδοσία χρησιμοποιείτε η παρακάτω μορφοποίηση:

Πίνακας 9: Εκτεταμένο ονοματοδοσία FAT από Λ.Σ. Microsoft (Microsoft, 2003b)

Byte Offset	Μήκος (bytes)	Περιγραφή
0x00	1	Αριθμός ακολουθίας
0x01	10	Χαρακτήρες ονόματος (πέντε (5) UTF-16 χαρακτήρες)
0x0B	1	Ιδιότητες
0x0C	1	Δεσμευμένα
0x0D	1	Έλεγχος του ονόματος DOS
0x0E	12	Χαρακτήρες ονόματος (έξι (6) UTF-16 χαρακτήρες)
0x1A	2	Το πρώτο cluster
0x1C	4	Χαρακτήρες ονόματος (δύο (2) UTF-16 χαρακτήρες)

Το MS-DOS προσπελαύνει το αρχείο χρησιμοποιώντας το κλασικό 8.3 όνομα του αρχείου, ενώ νεώτερα Λ.Σ. χρησιμοποιούν την εκτεταμένη ονομασία. Στο Σχήμα 4 φαίνεται η ονομασία ενός αρχείου το οποίο ονομάζεται Thequi~1.fox στο DOS, ενώ έχει το εκτεταμένο όνομα The quick brown.fox. Το εκτεταμένο όνομα χρησιμοποιεί Unicode, δηλαδή 2 byte για κάθε χαρακτήρα. Το byte ιδιότητας για το εκτεταμένο όνομα είναι 0x0F, ενώ για το όνομα 8.3 είναι 0x20.

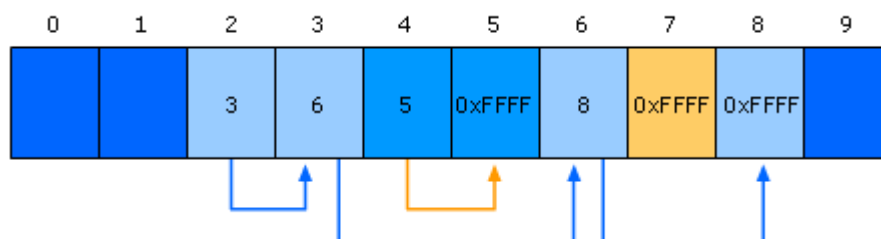


Σχήμα 3: Εκτεταμένη Ονοματοδοσία Αρχείων στο FAT

Να σημειωθεί ότι αν και τα Windows 2003, NT δεν χρησιμοποιούν τον ίδιο αλγόριθμο τροποποίησης ονόματος από εκτεταμένο σε τυπικό με τα Windows 95, 98, Me, παρόλα αυτά όλα τα Λ.Σ. της Microsoft μπορούν να προσπελάσουν αρχεία δημιουργημένα από κάποιο άλλο με εκτεταμένο όνομα.

3.3.14: Η Διαδικασία Προσπέλασης Αρχείων στο FAT

Όταν ένα αρχείο είναι αποθηκευμένο σε έναν τόμο με FAT, η πληροφορία του είναι αποθηκευμένη σε clusters του σκληρού δίσκου. Εάν ένα αρχείο απαιτεί χώρο μεγαλύτερο του ενός cluster, το FAT συνεχίζει να αποθηκεύει πληροφορία στο επόμενο διαθέσιμο cluster μέχρι να ολοκληρωθεί το αρχείο.



Σχήμα 4: Παράδειγμα Προσπέλασης αρχείων στο FAT

Το **Σφάλμα!** Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. περιγράφει ένα τέτοιο παράδειγμα. Στο σχήμα αυτό δέκα cluster περιέχουν

συνολικά τρία αρχεία. Το ένα αρχείο καταλαμβάνει τα cluster 2,3,6,8 ενώ το δεύτερο αρχείο καταλαμβάνει τα cluster 4,5. Το τρίτο αρχείο καταλαμβάνει μόνο το cluster 7. Ο αριθμός του αρχικού cluster είναι η διεύθυνση του πρώτου cluster που χρησιμοποιεί το αρχείο. Κάθε cluster περιέχει έναν δείκτη (Pointer) στο επόμενο cluster του ίδιου αρχείου, ενώ το τέλος του αρχείου δηλώνεται ως περιεχόμενο στο τελευταίο cluster του δεκαεξαδικού 0xFFFF.

Το byte ιδιότητας (Attribute byte) είναι δείκτης των χαρακτηριστικών της καταχώρησης και εξαρτάται από το Λ.Σ. πως θα τα χρησιμοποιήσει.

Τέσσερα από αυτά τα bit μπορούν να χρησιμοποιηθούν για την σήμανση ενός αρχείου (archive, system, hidden, and read-only).

3.3.15: Περιοχή Δεδομένων στο FAT (Data Region)

Όπως είναι γνωστό, μια κατάτμηση διαιρείται σε clusters, τα οποία είναι μικρά συνεχόμενα τμήματα. Το μέγεθος των cluster ποικίλει ανάλογα με τον τύπο του συστήματος αρχείων FAT που χρησιμοποιείται και το μέγεθος του διαμερίσματος και βρίσκεται κάπου μεταξύ 2 kB και 32 kB. Κάθε αρχείο μπορεί να καταλαμβάνει ένα ή περισσότερα από τα cluster ανάλογα με το μέγεθός του, δηλαδή ένα αρχείο αναπαρίστανται από μια λίστα από cluster (συνδεδεμένη λίστα). Ωστόσο αυτά τα clusters δεν αποθηκεύονται το ένα δίπλα στο άλλο αλλά συνήθως βρίσκονται κατακερματισμένα στον δίσκο.

Το File Allocation Table (FAT) αναπαριστά την λίστα των cluster που αντιστοιχούν σε κάθε αρχείο. Κάθε καταχώρηση περιέχει:

- Τον αριθμό του επόμενου cluster στην αλυσίδα
- Μια ειδική καταχώρηση που αντιπροσωπεύει το τέλος της αλυσίδας (EOC)
- Μια ειδική καταχώρηση που καταγράφει ένα κατεστραμμένο cluster
- Μηδέν (0) για cluster που δεν χρησιμοποιείται

Οι δύο πρώτες εγγραφές σε ένα FAT αποθηκεύουν ειδικές τιμές: Η πρώτη εγγραφή περιέχει ένα αντίγραφο του media descriptor (offset 0x15 του τομέα

εκκίνησης). Τα υπόλοιπα 4 bit (αν το σύστημα είναι FAT12), 8 bit (εάν είναι FAT16) ή 20 bit (εάν είναι FAT32) αυτής της καταχώρησης είναι 1.

Η δεύτερη καταχώρηση αποθηκεύει τον δείκτη του τέλους της αλυσίδας. Τα δύο πρώτα bit αυτής της καταχώρησης, στην περίπτωση των FAT16 και FAT32, χρησιμοποιούνται μερικές φορές για την διαχείριση τόμων: το πρώτο (1) bit: ο τελευταίος τερματισμός ήταν επιτυχής; Το επόμενο bit : κατά την διάρκεια του προηγούμενου mount δεν εντοπίστηκε κανένα I/O σφάλμα.

Επειδή οι δύο πρώτες καταχωρήσεις FAT αποθηκεύουν ειδικές τιμές, δεν υπάρχει κανένα cluster με τιμή 0 ή 1. Το πρώτο cluster (μετά το ριζικό κατάλογο για FAT 12/16) είναι το cluster 2.

Πίνακας 10: Τιμές καταχώρησης FAT

FAT12	FAT16	FAT32	Description
0x000	0x0000	0x00000000	Ελεύθερο Cluster — χρησιμοποιείται επίσης ως ριζικός κατάλογος του αρχικού cluster σε ".." καταχωρήσεις for subdirectories του ριζικού καταλόγου (FAT12/16)
0x001	0x0001	0x00000001	Δεσμευμένο, όχι για χρήση
0x002-0xFEf	0x0002-0xFFEF	0x00000002-0x0FFFFFFEF	Cluster σε χρήση
0xFF0-0xFF5	0xFFFF0-0xFFFF5	0xFFFFFFFF0-0xFFFFFFFF5	Δεσμευμένο σε ορισμένα περιβάλλοντα ή χρησιμοποιείται επίσης
0xFF6	0xFFFF6	0xFFFFFFFF6	Δεσμευμένο, όχι για χρήση
0xFF7	0xFFFF7	0xFFFFFFFF7	Κατεστραμμένο sector στο cluster ή δεσμευμένο cluster
0xFF8-0xFFF	0xFFFF8-0xFFFFF	0xFFFFFFFF8-0xFFFFFFFFF	Τελευταίο cluster του αρχείου (EOC)

Σημειώστε ότι το FAT32 χρησιμοποιεί μόνο 28 από τα 32 bit. Τα πρώτα 4 bits είναι συνήθως μηδενικά (0) (όπως αναγράφεται στο πίνακα πιο πάνω) αλλά είναι δεσμευμένα και πρέπει να μείνουν ανέπαφα.

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό έγινε αναλυτική παρουσίαση των Συστημάτων Αρχείων της Microsoft FAT (16, 32).

Επισημάνθηκαν τα κυριότερα ιδιαίτερα σημεία και τεχνολογίες της οικογένειας Σ.Α. καθώς και ο τρόπος πρόσβασης του φυσικού μέσου (τόμου ή δίσκου).

Στο επόμενο κεφάλαιο θα παρουσιαστεί το Σ.Α. Ext2 του Linux.

ΚΕΦΑΛΑΙΟ 4

ΤΟ ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ Ext2

ΕΙΣΑΓΩΓΗ

Το σύστημα αρχείων με την ονομασία Second Extension Filesystem (Ext2) είναι ίσως το ευρύτερα χρησιμοποιούμενο σύστημα αρχείων στο Linux σήμερα. Είναι συμβατό με τις διεργασίες του Unix και προσφέρει αρκετά καινοτόμα χαρακτηριστικά. Επίσης χάρη στις δυνατότητες βελτιστοποίησης που περιέχονται στον πυρήνα (Kernel) είναι σταθερό και προσφέρει εξαιρετικές επιδόσεις (Bryant, Forester, & Hawkes, 2002).

Το Ext2 δημιουργήθηκε με την προοπτική της εξέλιξης, και έτσι περιέχει δυνατότητες αναβάθμισης ή χρήσης για καινούριες προσθήκες.

Αν και πρωτίστως δημιουργήθηκε και εγκαταστάθηκε στο Linux, σήμερα χρησιμοποιείται και σε πολλά άλλα Λ.Σ.

4.1: Το Virtual File System (VFS)

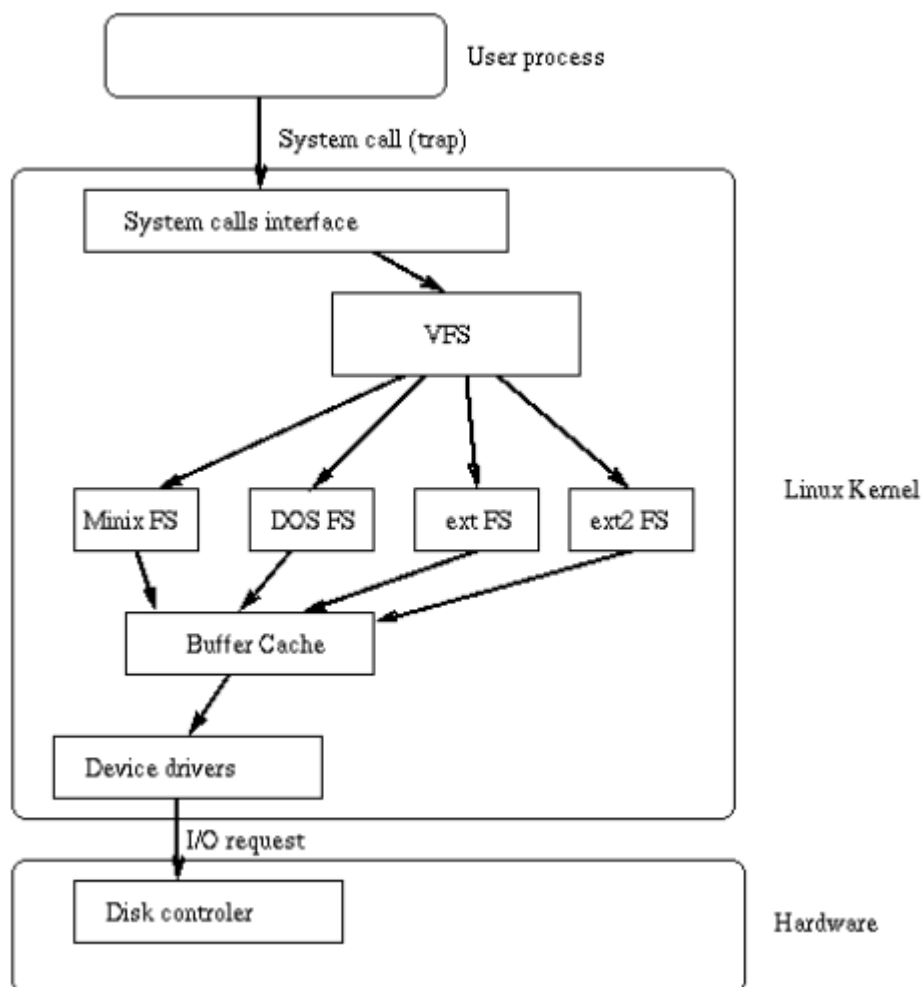
Το Linux ξεκίνησε ως παραλλαγή του Minix και έτσι ήταν ευκολότερο να μοιράζεται κοινή αρχιτεκτονική δίσκων, δηλαδή Σ.Α. από το να αναπτυχθεί καινούριο Σ.Α. αποκλειστικά για αυτό. Έτσι ο Linus Torvads αποφάσισε να υλοποιήσει την υποστήριξη του Σ.Α. του Minix στο Linux. Άλλωστε το Minix είχε την φήμη ενός αποτελεσματικού και σχετικά χωρίς bugs λογισμικό.

Επειδή όμως οι περιορισμοί του Σ.Α. στο Minix ήταν αρκετοί, κάποιοι άρχισαν ήδη να εργάζονται στην κατεύθυνση του καινούριου Σ.Α. αποκλειστικά για το Linux. Για να γίνει πιο εύκολη η ενσωμάτωση του επερχόμενου Σ.Α. στον πυρήνα του Linux, δημιουργήθηκε το Ιδεατό Σύστημα Αρχείων, Ι.Σ.Α (Virtual File System ,VFS).

Το Ι.Σ.Α χρησιμοποιείται από τον πυρήνα κατά την διάρκεια διεργασιών σε αρχεία από το σύστημα. Το Ι.Σ.Α. είναι ένας ενδιάμεσο επίπεδο που χειρίζεται τις κλήσεις που αφορούν σε διεργασίες Ε/Ε σε φυσικό επίπεδο του Σ.Α.

Αυτός ο έμμεσος μηχανισμός χρησιμοποιείται και από πολλά Λ.Σ. παραλλαγές του Unix για καλύτερη ενσωμάτωση διαφορετικών Σ.Α. (Kleiman, 1986).

Όταν μία διεργασία παράγει μία κλήση που αφορά ένα αρχείο, ο πυρήνας καλεί μία διεργασία του Ι.Σ.Α. η διεργασία αυτή χειρίζεται όλες τις εργασίες που δεν έχουν να κάνουν με την φυσική δομή του αρχείου, και μετά κατευθύνει την ροή σε μία διεργασία του πραγματικού Σ.Α. που είναι υπεύθυνο για τον χειρισμό των εργασιών σχετικών με την φυσική δομή (αποθήκευση κτλ) του αρχείου. Οι διεργασίες του πραγματικού Σ.Α. μέσω της προσωρινής μνήμης διαχειρίζονται αποδοτικά όλες τις εντολές Ε/Ε. Η διεργασία φαίνεται παραστατικά στο Σχήμα 5.



Σχήμα 5: Το VFS στο ext2 & στο Linux (Card, Ts'o, & Tweedie, 1994)

4.2: Η δομή του Ιδεατού Συστήματος Αρχείων

Το Ι.Σ.Α. υλοποιεί όλες τις διεργασίες ενός Συστήματος Αρχείων. Περιέχει όλες τις απαραίτητες διεργασίες που σχετίζονται με Σ.Α., συμβολικούς δεσμούς και αρχεία.

Το Ι.Σ.Α. γνωρίζει όλους τους υποστηριζόμενους τύπους αρχείων από τον πυρήνα. Χρησιμοποιεί τον πίνακα που δημιουργήθηκε κατά την αρχική διαμόρφωση του πυρήνα. Κάθε εγγραφή σε αυτόν τον πίνακα περιγράφει ένα τύπο περιεχομένου του Σ.Α.: Περιέχει την ονομασία του Σ.Α. και έναν δείκτη στην διεργασία που υποδείχθηκε κατά την διάρκεια της προσάρτησης (Mount) του τόμου ή δίσκου. Η διεργασία αυτή είναι υπεύθυνη να αναγνώσει το superbloc του δίσκου, αρχικοποιεί τις εσωτερικές μεταβλητές και επιστρέφει μία ονομασία

(descriptor) του προσαρτημένου Σ.Α. στο Ι.Σ.Α. αφού έχει προσαρτηθεί το Σ.Α. , οι διεργασίες του Ι.Σ.Α. μπορούν να χρησιμοποιήσουν αυτή την ονομασία για να προσπελάσουν τις διεργασίες του πραγματικού Σ.Α.

Η ονομασία του πραγματικού Σ.Α. εμπεριέχει και όλη την απαραίτητη πληροφορία προσπέλασης του. Μεταξύ άλλων περιέχει όλες τις κοινές πληροφορίες των Σ.Α., δείκτες προς κάθε διεργασία, και δεδομένα φυσικού επιπέδου. Οι δείκτες σε κάθε διεργασία επιτρέπουν στο Ι.Σ.Α. να καλεί τις διεργασίες αυτές όποτε υπάρχει ανάγκη φυσικής πρόσβασης στον τόμο ή δίσκο.

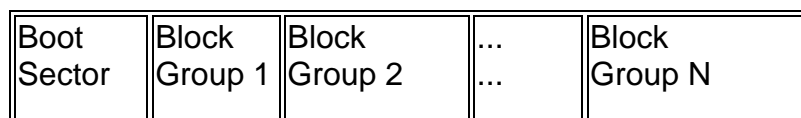
Υπάρχουν και άλλοι δύο τύποι περιγραφής (descriptors) που χρησιμοποιεί το Ι.Σ.Α. : ο περιγραφέας συμβολικού δεσμού (inode descriptor) και ο περιγραφέας ανοιχτού αρχείου (open file descriptor). Και οι δύο δίνουν πρόσβαση σε φυσικό επίπεδο στα αρχεία, με τον πρώτο να περιέχει δείκτες σε διεργασίες που μπορούν να χρησιμοποιηθούν για ενέργειες σε όλα τα αρχεία (π.χ. create, unlink), ενώ οι δεύτεροι περιέχουν δείκτες σε διεργασίες επί ανοιχτών αρχείων (π.χ. read, write).

4.3: Η φυσική δομή του Ext2

Η φυσική δομή του Ext2 έχει επηρεαστεί από την αντίστοιχη του BSD.

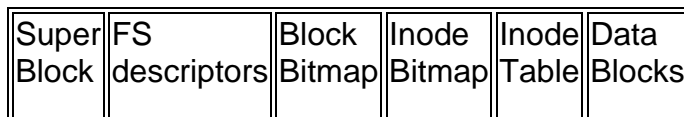
Όπως κάθε Σ.Α. για Linux, το Ext2 υλοποιεί ένα βασικό κορμό κοινών παραμέτρων με το Unix. Τα αρχεία αποτυπώνονται από inodes, φακέλους που τους χειρίζεται σαν απλά αρχεία ή συσκευές, έτσι ώστε να είναι άμεσα προσπελάσιμα μέσω κλήσεων εισόδου/εξόδου (Bach, 1986).

Το Σ.Α. αποτελείται από ομάδες μπλοκ (block group). Αυτές οι λογικές μονάδες, δεν συνδέονται απαραίτητα με τα φυσικά μπλοκ του μέσου αποθήκευσης, αφού τα μοντέρνα μέσα είναι βελτιστοποιημένα για συνεχή πρόσβαση, αποκρύπτοντας την πραγματική τους δομή από το Λ.Σ. (Phillips, 2001). Η φυσική δομή του Ext2 αποτυπώνεται στο Σχήμα 6.



Σχήμα 6: Η φυσική δομή του Ext2

Κάθε ομάδα μπλοκ περιέχει ένα επιπλέον αντίγραφο από σημαντικές πληροφορίες ελέγχου του Σ.Α. (όπως το superblock και τους περιγραφείς του Σ.Α., filesystem descriptors) και μέρος του ίδιου του Σ.Α. δηλαδή ένα γραφικό του block, ένα γραφικό για το inode, ένα κομμάτι του πίνακα του inode και φυσικά τα δεδομένα του μπλοκ. Η δομή μιας λογικής ομάδας μπλοκ φαίνεται στο Σχήμα 7:



Σχήμα 7: Αποτύπωση της λογικής διάταξης μπλοκ στο Ext2

Η χρήση των παραπάνω μπλοκ βοηθάει πολύ στην αξιοπιστία του Σ.Α. Επειδή οι δομές ελέγχου επαναλαμβάνονται σε κάθε ομάδα μπλοκ, είναι πολύ εύκολο να ανανήψει από μία κατάσταση όπου το superblock είναι πιθανότατα κατεστραμμένο. Επίσης βοηθάει πολύ στην απόδοση: με το να μειώνει την απόσταση μεταξύ των μπλοκ των συμβολικών δεσμών (inodes) και των μπλοκ δεδομένων, καθίσταται δυνατή η ελάχιστη μετακίνηση των κεφαλών του δίσκου μεταξύ ενεργειών εισόδου/ εξόδου πάνω στα αρχεία.

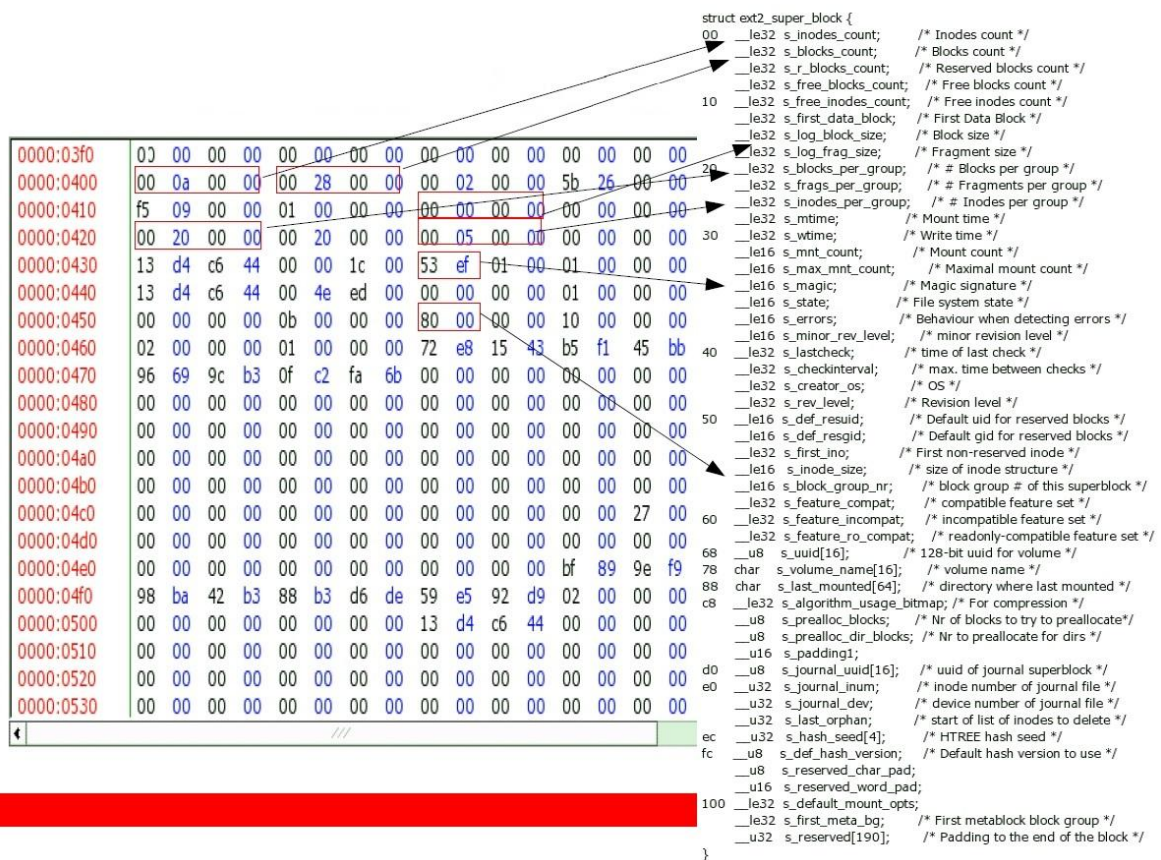
Πίνακας 11: Δομή Superblock και Block Group Descriptor

Superblock		
Offset (bytes)	Size (bytes)	Περιγραφή
0	4	s_inodes_count
4	4	s_blocks_count
8	4	s_r_blocks_count
12	4	s_free_blocks_count
16	4	s_free_inodes_count
20	4	s_first_data_block
24	4	s_log_block_size
28	4	s_log_frag_size
32	4	s_blocks_per_group
36	4	s_frags_per_group
40	4	s_inodes_per_group
44	4	s_mtime
48	4	s_wtime
52	2	s_mnt_count
54	2	s_max_mnt_count
56	2	s_magic
58	2	s_state
60	2	s_errors

62	2	s_minor_rev_level
64	4	s_lastcheck
68	4	s_checkinterval
72	4	s_creator_os
76	4	s_rev_level
80	2	s_def_resuid
82	2	s_def_resgid
EXT2_DYNAMIC_REV Specific		
84	4	s_first_ino
88	2	s_inode_size
90	2	s_block_group_nr
92	4	s_feature_compat
96	4	s_feature_incompat
100	4	s_feature_ro_compat
104	16	s_uuid
120	16	s_volume_name
136	64	s_last_mounted
200	4	s_algo_bitmap
Performance Hints		
204	1	s_prealloc_blocks
205	1	s_prealloc_dir_blocks
206	2	(alignment)
Journaling Support		
208	16	s_journal_uuid
224	4	s_journal_inum
228	4	s_journal_dev
232	4	s_last_orphan
Directory Indexing Support		
236	4x4	s_hash_seed
252	1	s_def_hash_version
253	3	Κενές θέσεις - δεσμευμένο για μελλοντική χρήση
Other options		
256	4	s_default_mount_options
260	4	s_first_meta_bg
264	760	Αχρησιμοποίητο - δεσμευμένο για μελλοντική χρήση

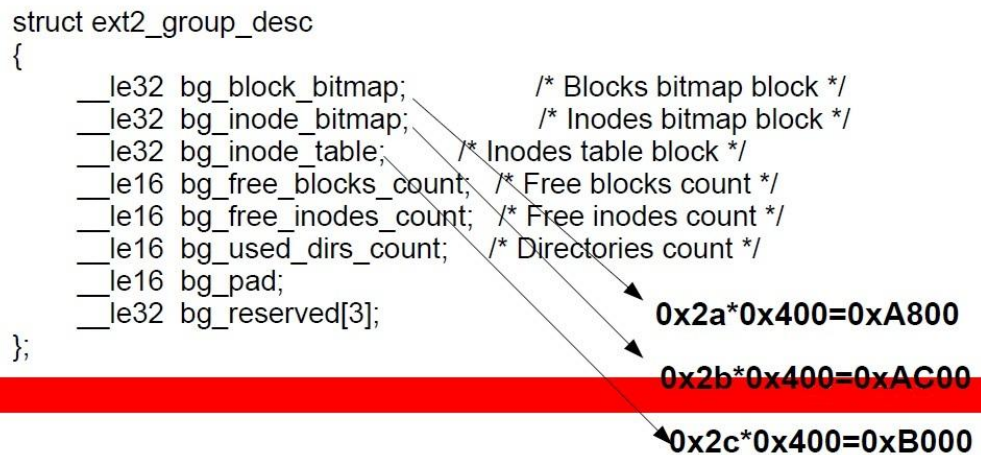
Block Group Descriptor		
Offset (bytes)	Size (bytes)	Description
0	4	bg_block_bitmap
4	4	bg_inode_bitmap
8	4	bg_inode_table
12	2	bg_free_blocks_count

14	2	bg_free_inodes_count
16	2	bg_used_dirs_count
18	2	bg_pad
20	12	bg_reserved



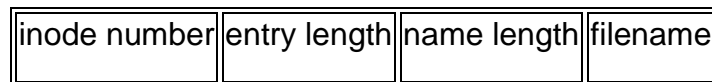
Σχήμα 8: Το superblock στον δίσκο

0000:07f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
0000:0800	2a 00 00 00 2b 00 00 00 2c 00 00 00 27 1f f5 04	* .. + ..
0000:0810	02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
0000:0820	2a 20 00 00 2b 20 00 00 2c 20 00 00 34 07 00 05	* .. + ..
0000:0830	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
0000:0840	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...

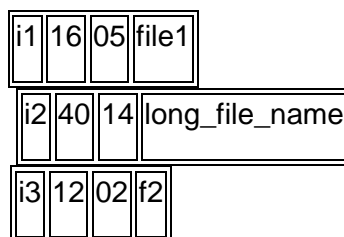


Σχήμα 9: Το group descriptor στον δίσκο

Το Ext2 χειρίζεται τους φακέλους ως συνδεδεμένες λίστες μεταβλητού μήκους. Κάθε τέτοια καταχώρηση περιέχει τον αριθμό του inode, το μήκος της καταχώρησης, το όνομα του αρχείου και το μήκος του. Με την χρήση καταχωρήσεων μεταβλητού μήκους, η χρήση εκτεταμένων ονομάτων δεν κατασπαταλά χώρο στον δίσκο. Η δομή ενός φακέλου αναπαρίσταται στο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..**



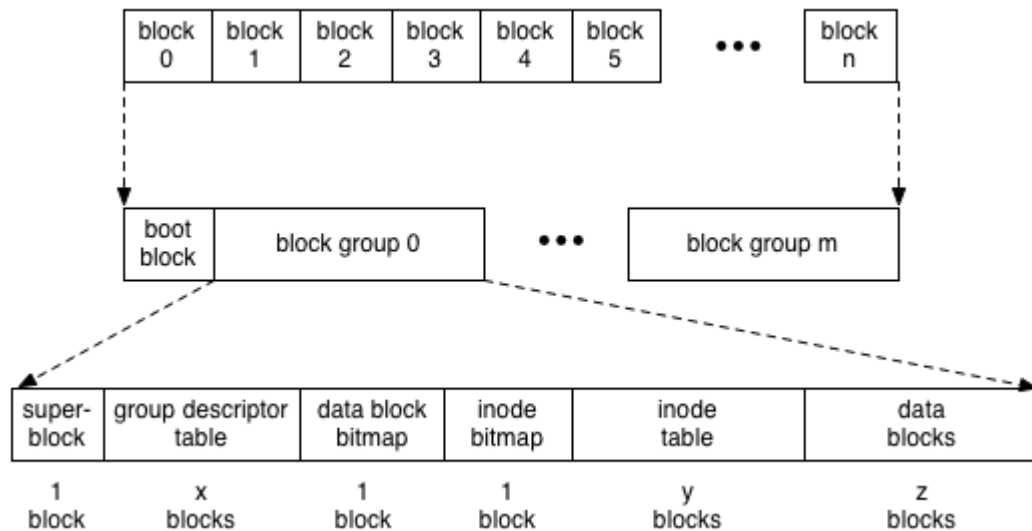
Σχήμα 10: Αναπαράσταση της δομής ενός φακέλου στο Ext2



Σχήμα 11: Αναπαράσταση ονομάτων αρχείων στο Ext2

Στο Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. φαίνεται ένα χαρακτηριστικό παράδειγμα μίας δομής αρχείων και φακέλων στο Ext2, όπου αναπαρίστανται 3 αρχεία με τα ονόματα file1, long_file_name, f2 .

Πιο αναλυτικά :



Σχήμα 12: Αναπαράσταση ονομάτων αρχείων στο Ext2

Με την χρήση της εντολής **mke2fs** σε μία κατάτμηση των 20 MB παίρνουμε τις παρακάτω πληροφορίες.

20 MB Partition meta-data layout

```
offset # of blocks description
-----
0      1      boot record

-- block group 0 --(1024 bytes) 1 superblock

2      1      group descriptors
3      1      block bitmap
4      1      inode bitmap
5      214    inode table
219    7974   data blocks

-- block group 1 --

8193   1      superblock backup
8194   1      group descriptors backup
8195   1      block bitmap
8196   1      inode bitmap
8197   214    inode table
8408   7974   data blocks
```

```
-- block group 2 --  
  
16385 1      block bitmap  
16386 1      inode bitmap  
16387 214    inode table  
16601 3879  data blocks
```

Block group = $s_block_count / s_blocks_per_group = 20480 / 8192 = 2.5$

To Super block ξεκινάει από το offset 1024.

Group desc = $(32 * 2.5) / 1024 = .078$

block bitmap = $(8192 / 8) / 1024 = 1$ block

Άρα ο ελάχιστος αριθμός block που θα χρειαστούμε είναι ένας (1).

inode bitmap = $(1712 / 8) / 1024 = 0.204$ blocks.

inode table = $(1712 * 128) / 1024 = 214$ blocks.

Inodes per group = 1712 και Inode size = 128.

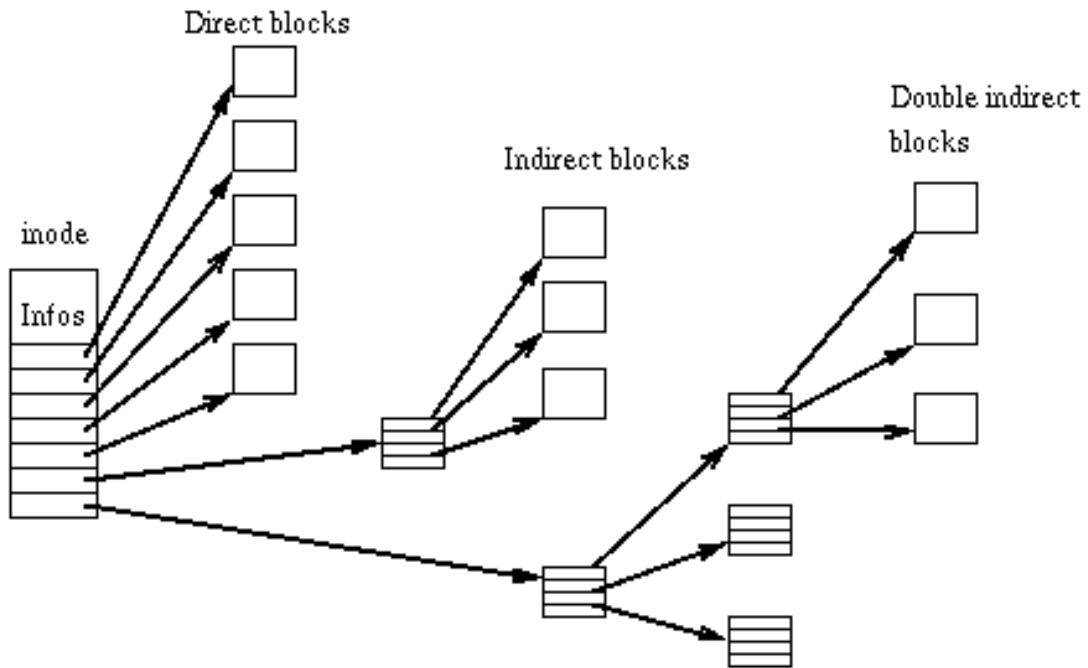
4.4: Βασικά Δομικά Στοιχεία του Ext2

4.4.1: Inodes

Κάθε αρχείο αντιπροσωπεύεται από μία δομή που αποκαλείται inode. Κάθε inode περιέχει την περιγραφή του αρχείου:

- είδος αρχείου
- δικαιώματα πρόσβασης
- ιδιοκτήτης
- χρονική σήμανση
- μέγεθος
- δείκτης στα δεδομένα του αρχείου

Οι διευθύνσεις των μπλοκ δεδομένων είναι αποθηκευμένες στο inode. Όταν κάποιος χρήστης ζητήσει μία λειτουργία E/E στο αρχείο, ο πυρήνας μετατρέπει την τρέχουσα μετατόπιση (offset) σε έναν αριθμό μπλοκ και χρησιμοποιεί τον αριθμό αυτό σαν δείκτη στον πίνακα διευθύνσεων από όπου κατευθύνεται στο φυσικό μπλοκ (Sivathanu, Sundararaman, & Zadok, 2006). Η δομή ενός inode φαίνεται στο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..**



Σχήμα 13: Η δομή ενός inode στο Ext2

Εκτελώντας από τερματικό Linux την εντολή `stat` και ως παράμετρο το όνομα του αρχείου μπορούμε να πάρουμε πληροφορίες σχετικά με το inode όπως φαίνεται παρακάτω :

```
# stat /bin/bash
```

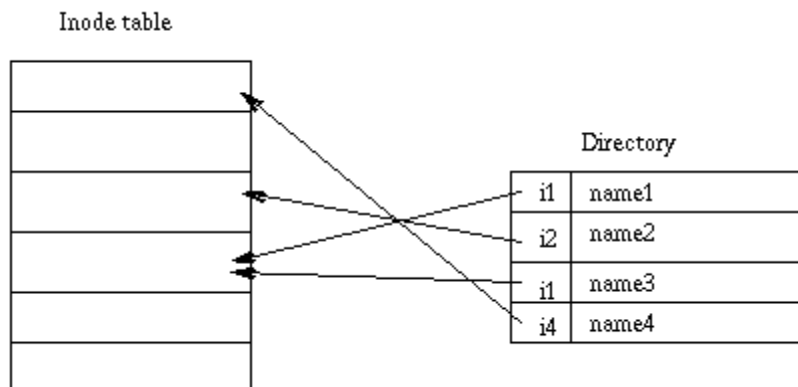
```
File: `/bin/bash'
Size: 722684          Blocks: 1432          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 12799859     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2008-04-06 19:13:50.000000000 -0400
Modify: 2006-07-12 03:11:53.000000000 -0400
Change: 2007-11-22 04:05:30.000000000 -0500
```

4.4.2: Φάκελοι (Directories)

Οι φάκελοι κατανέμονται με ιεραρχική δομή. Κάθε φάκελος μπορεί να περιέχει αρχεία και υποφάκελους.

Οι φάκελοι αποτελούν έναν ειδικό τύπο αρχείου. Στην πραγματικότητα ένας φάκελος είναι ένα αρχείο που περιέχει μία λίστα με καταχωρήσεις. Κάθε καταχώρηση περιέχει έναν αριθμό inode και μία ονομασία αρχείου. Όταν μία διεργασία χρησιμοποιήσει μία διαδρομή αρχείου (path), ο πυρήνας ψάχνει στους

φακέλους να βρει τον αντίστοιχο αριθμό inode. Αφού γίνει η μετατροπή σε έναν αριθμό Inode, το συγκεκριμένο inode φορτώνεται στην μνήμη και χρησιμοποιείται για τις επερχόμενες κλήσεις σε αυτό. Στο Σχήμα 14 αποτυπώνεται μία τέτοια δομή φακέλου.



Σχήμα 14: Τυπική καταγραφή ενός φακέλου στο Ext2

4.4.3: Δεσμοί (Links)

Η λογική των δεσμών πρωτοαναφέρθηκε στο Unix. Με αυτόν τον τρόπο πολλαπλές ονομασίες μπορούν να αναφέρονται στο ίδιο inode. Το inode περιέχει ένα πεδίο που περιέχει έναν αριθμό που σχετίζεται με αυτό το αρχείο. Κάθε φορά που προστίθεται ένας δεσμός που δείχνει στο συγκεκριμένο αρχείο, δημιουργείται μία εγγραφή στο φάκελο, και ο αριθμός του πλήθους των δεσμών που υπάρχει στα πεδία του inode, αυξάνεται κατά ένα. Αντίστοιχα όταν διαγράφεται ένας δεσμός, (όταν χρησιμοποιείται η εντολή `rm` για να διαγραφεί μία ονομασία αρχείου) ο πυρήνας αφαιρεί μία μονάδα από το πλήθος των δεσμών, ή αποδεσμεύει τελείως αυτό το συγκεκριμένο inode, αν αυτός ο αριθμός γίνει μηδέν.

Ο τύπος αυτός των δεσμών ονομάζεται σκληρός δεσμός (hard link) και μπορεί να χρησιμοποιηθεί μόνο εντός του ίδιου Σ.Α. Δεν είναι δυνατόν να χρησιμοποιηθεί μεταξύ διαφορετικών Σ.Α. Επίσης αυτοί οι σκληροί δεσμοί μπορούν να χρησιμοποιηθούν μόνο πάνω σε αρχεία και όχι σε φακέλους (λόγω κυκλικής αναφοράς στους υποφακέλους).

Αντίθετα οι συμβολικοί δεσμοί (symbolic links) είναι απλώς αρχεία που περιέχουν μόνο μία ονομασία αρχείου. Όταν ο πυρήνας συναντά ένα τέτοιο

συμβολικό δεσμό αντικαθιστά την ονομασία του δεσμού με τα περιεχόμενα του, δηλαδή την ονομασία που περιέχει μέσα του το αρχείο του συμβολικού δεσμού, και ξαναρχίζει να διατρέχει το ιεραρχικό δέντρο. Αφού ένας συμβολικός δεσμός δεν δείχνει σε ένα inode είναι δυνατόν να δημιουργηθούν τέτοιοι δεσμοί ανάμεσα σε διαφορετικά Σ.Α. Οι συμβολικοί δεσμοί μπορούν να δείχνουν ακόμη και σε μη υπάρχοντα αρχεία. Είναι πολύ χρήσιμοι γιατί δεν υπόκεινται στους περιορισμούς των σκληρών δεσμών, αν και χρησιμοποιούν παραπάνω χώρο στον δίσκο και δημιουργούν αρκετό πλεονασμό δεδομένων κάθε φορά που ο πυρήνας ξανα-ξεκινάει να διατρέχει την διαδρομή των αρχείων μετά την αντικατάσταση.

4.4.4: Ειδικά Αρχεία Συσκευής (Special Device Types)

Στο Unix οι συσκευές προσπελαύνονται μέσω ειδικών αρχείων (special files). Το αρχείο μίας συσκευής δεν χρησιμοποιεί χώρο στο Σ.Α. είναι μόνο ένα σημείο πρόσβασης προς την συσκευή που δείχνει.

Υπάρχουν δύο ειδών ειδικά αρχεία: αρχεία χαρακτήρα (character) και ειδικά αρχεία μπλοκ (block special files). Το πρώτο επιτρέπει λειτουργίες E/E ενώ το δεύτερο απαιτεί εγγραφή δεδομένων σε μπλοκ μέσω της προσωρινής μνήμης.

Ένα τέτοιο αρχείο χαρακτηρίζεται από τον πρωτεύοντα αριθμό (major number) που δείχνει το είδος της συσκευής και έναν δευτερεύοντα αριθμό που δείχνει την ίδια την συσκευή (Card et al., 1994).

4.4.5: Η βιβλιοθήκη του Ext2

Η βιβλιοθήκη libext2fs δίνει πρόσβαση σε όλα τα προγράμματα χρήστη που απαιτούν πρόσβαση στο Σ.Α. Η βιβλιοθήκη αυτή παρέχει όλες τις απαραίτητες δυνατότητες ανάλυσης, ανάγνωσης και τροποποίησης δεδομένων στο φυσικό επίπεδο του δίσκου.

Τα περισσότερα εργαλεία (utilities) του Ext2 χρησιμοποιούν αυτούσιες αυτές τις δυνατότητες για να εκτελέσουν εργασίες συντήρησης και ανάλυσης του φυσικού μέσου. Αυτό βοηθάει πολύ στην αυτόματη αναβάθμιση αυτών των εργαλείων μέσω της συντήρησης της ίδιας της βιβλιοθήκης. Επίσης αυτό οδηγεί σε

μικρότερα αρχεία μέσω της επαναχρησιμοποίησης του κώδικα της βιβλιοθήκης για όλες τις κοινές λειτουργίες μεταξύ τέτοιων εργαλείων.

Επίσης είναι πολύ εύκολη η συγγραφή καινούριων προγραμμάτων που απαιτούν άμεση πρόσβαση στο φυσικό μέσον (δίσκο, τόμο κτλ) μέσω της χρήσης των αφαιρετικών (abstract) μεθόδων της βιβλιοθήκης.

Το Ext2 προσφέρει πολλαπλές τάξεις πρόσβασης:

- Η πρώτη τάξη είναι οι διεργασίες του Σ.Α. Διεργασίες ανοίγματος/κλεισίματος αρχείου, ανοίγματος/κλεισίματος εικόνας του αρχείου, δημιουργία καινούριου αρχείου. Επίσης υπάρχουν διεργασίες χειρισμού του πίνακα χαλασμένων μπλοκ (bad block table).
- Η δεύτερη τάξη αφορά σε φακέλους. Μία δυνατότητα της βιβλιοθήκης είναι να μπορεί να δημιουργήσει ή να επεκτείνει φακέλους, όπως και να προσθέσει ή να αφαιρέσει καταχωρήσεις μέσα στους φακέλους. Επίσης παρέχει την δυνατότητα αντιστοίχισης μίας διαδρομής σε συγκεκριμένο inode όπως και για το αντίστροφο, δηλαδή την προβολή διαδρομής από τον αριθμό του inode.
- Η τρίτη τάξη διεργασιών αφορά σε inodes. Είναι δυνατόν να διαβαστεί ο πίνακας των inodes, διεργασίες ανάγνωσης/εγγραφής inode, όπως επίσης και σάρωση όλων των μπλοκ που είναι αντιστοιχισμένα σε ένα inode. Επίσης παρέχονται διεργασίες ανάθεσης και απελευθέρωσης (allocate, deallocate) τόσο μπλοκ, όσο και inode (Poirier, 2011).

4.4.6: Επιδόσεις του Ext2

Ο πυρήνας του Ext2 περιέχει πολλές βελτιώσεις όσον αφορά στις επιδόσεις, επιδρώντας στην ταχύτητα των διεργασιών E/E των αρχείων.

Το Ext2 χρησιμοποιεί την μνήμη προσωρινής αποθήκευσης (buffer) κάνοντας συνεχώς “διαβάσματα προς τα εμπρός” (readaheads): όταν ένα μπλοκ πρέπει να διαβαστεί, ο πυρήνας εκτελεί τις διεργασίες E/E σε αρκετά συνεχόμενα επόμενα μπλοκ. Έτσι προσπαθεί να επιτύχει να έχει ήδη φορτώσει στην

προσωρινή μνήμη το επόμενο μπλοκ που πιθανότατα θα ζητηθεί. Αυτό συμβαίνει όταν ζητούνται από το Λ.Σ. διαδοχικά αρχεία. Τότε το Ext2 διαβάζει και φορτώνει στην προσωρινή μνήμη είτε όλο το φάκελο, ή γειτονικούς φακέλους.

Το Ext2 επίσης περιέχει βελτιστοποιήσεις φυσικής τοποθεσίας. Τα λογικά γκρουπ χρησιμοποιούνται ώστε να ομαδοποιούν σχετικούς μεταξύ τους συμβολικούς δεσμούς και δεδομένα. Ο Πυρήνας προσπαθεί συνεχώς να βάλει στο ίδιο λογικό μπλοκ το αρχείο και το συμβολικό δεσμό του. Έτσι ελαχιστοποιούνται οι μετατοπίσεις της κεφαλής ανάγνωσης του δίσκου κατά την διάρκεια ανάγνωσης των δεδομένων ενός αρχείου και του συμβολικού δεσμού του.

Όταν αποθηκεύει τα δεδομένα ενός αρχείου, το Ext2 προ-δεσμεύει μέχρι οκτώ συνεχόμενα επόμενα μπλοκ. Στο 75% των περιπτώσεων αυτοί οι προδεσμευμένοι πόροι χρησιμοποιούνται. Αυτό έχει θετικές επιπτώσεις σε περιπτώσεις βαριάς χρήσης του δίσκου και οδηγεί σε αρχεία που βρίσκονται αποθηκευμένα σε διαδοχικά φυσικά μπλοκ, ώστε να επιτυγχάνονται πολύ καλύτερες ταχύτητες στις μελλοντικές αναγνώσεις αυτών των αρχείων από το Σ.Α. (McKusick, Joy, Leffler, & Fabry, 1984)

Με τις παραπάνω δύο τεχνικές επιτυγχάνονται :

- Γειτονικές (κοντινές) φυσικές αποθηκεύσεις σχετιζόμενων μεταξύ τους αρχείων
- Σχετιζόμενα μεταξύ τους φυσικά μπλοκ μέσω της προδέσμευσης των οκτώ επόμενων μπλοκ.

4.4.7: Οι δυνατότητες του Ext2

Το Ext2 υποστηρίζει τους συνηθισμένους τύπους αρχείων Unix, δηλαδή αρχεία, φακέλους, αρχεία συσκευής (device special files) και συμβολικούς δεσμούς (symbolic links).

Το Ext2 μπορεί να διαχειριστεί Σ.Α. δημιουργημένα σε πολύ μεγάλες κατατμήσεις δίσκου. Αν και η αρχική έκδοση του Kernel υποστήριζε απώτερο μέγεθος 2 GB, σήμερα έχει αυξηθεί σε 4 TB. Έτσι μπορούν να χρησιμοποιηθούν πολύ μεγάλοι δίσκοι, χωρίς την ανάγκη δημιουργίας πολλών κατατμήσεων

Το Ext2 υποστηρίζει εκτεταμένες ονομασίες αρχείων μεταβλητού μήκους. Το απώτερο δυνατό μέγεθος ονόματος αρχείου είναι 255 χαρακτήρες. Αυτό το μέγεθος μπορεί να επεκταθεί σε 1012 αν παραστεί ανάγκη.

Το Ext2 παρακρατεί κάποια blocks για τον υπερχρήστη (super user - root). Συνήθως το 5% είναι δεσμευμένο. Αυτό δίνει την δυνατότητα στον διαχειριστή να επαναφέρει εύκολα το σύστημα από καταστάσεις όπου διεργασίες χρηστών έχουν πρακτικά «γεμίσει» το Σ.Α.

4.4.8: Προηγμένες δυνατότητες του Ext2

Το Ext2 υποστηρίζει και κάποιες επιπλέον των δυνατοτήτων του Unix.

Οι ιδιότητες αρχείου (File attributes) επιτρέπουν στον χρήστη να μετατρέψει την συμπεριφορά του πυρήνα (kernel) όταν επενεργεί πάνω σε μία ομάδα αρχείων. Με αυτή την δυνατότητα, κάθε φορά που δημιουργείται ένα αρχείο στον συγκεκριμένο φάκελο, κληρονομεί τις προκαθορισμένες ιδιότητες του συγκεκριμένου φακέλου.

Δυνατότητες σύγχρονης αναβάθμισης όπως του BSD μπορούν να χρησιμοποιηθούν. Η δυνατότητα προσάρτησης (mount) επιτρέπει στον διαχειριστή να ζητήσει την σύγχρονη εγγραφή των μεταδεδομένων (inodes, bitmap blocks, indirect blocks and directory blocks) στο δίσκο την στιγμή που τροποποιούνται. Αν και αυτό οδηγεί στην ακεραιότητα των μεταδεδομένων, συνήθως έχει επίπτωση στις επιδόσεις και δεν χρησιμοποιείται.

Το Ext2 επιτρέπει στον διαχειριστή να επιλέξει το μέγεθος του λογικού μπλοκ, την ώρα που δημιουργεί το Σ.Α. Τα μεγέθη του λογικού μπλοκ είναι συνήθως 1024, 2048 και 4096 bytes. Η χρήση μεγάλου μεγέθους μπλοκ μπορεί να επιταχύνει τις λειτουργίες εισόδου/εξόδου αφού έτσι παρατηρούνται λιγότερες μετακινήσεις της κεφαλής του δίσκου για την προσπέλαση των αρχείων. Από την άλλη μεριά το μεγάλο μέγεθος μπλοκ επιφέρει σπατάλη χώρου στον δίσκο, ενώ τα περισσότερα από τα πλεονεκτήματα των μεγάλων λογικών μπλοκ μπορούν να υλοποιηθούν από τις τεχνικές preallocation που χρησιμοποιεί το Ext2.

Το Ext2 υλοποιεί πολύ γρήγορους συμβολικούς δεσμούς. Ένας συμβολικός δεσμός δεν χρησιμοποιεί καθόλου μπλοκ δεδομένων στο Σ.Α. Η ονομασία

προορισμού (εκεί που δείχνει) δεν αποθηκεύεται σε ένα μπλοκ δεδομένων αλλά μέσα στο inode. Αυτή η λογική μπορεί να εξοικονομεί χώρο στο δίσκο και επιταχύνει τις λειτουργίες αφού δεν χρειάζεται η προσπέλαση του δίσκου για να «διαβαστεί» ένα τέτοιος δεσμός (link). Προφανώς επειδή ο χώρος του inode είναι συγκεκριμένος, δεν είναι δυνατόν να αποθηκευτεί εκεί ένας συμβολικός δεσμός μεγαλύτερος των 60 χαρακτήρων.

Το Ext2 διατηρεί ιστορικό της κατάστασης του Σ.Α. Για αυτό το σκοπό υπάρχει ένα ειδικό πεδίο στο superblock το οποίο δείχνει αυτή την κατάσταση. Όταν το Σ.Α. προσαρτάται (γίνεται mount) σε κατάσταση read/write, τότε το πεδίο αυτό γίνεται “not Clean”. Αν το Σ.Α. προσαρτηθεί σε κατάσταση “read only” ή από-προσαρτηθεί (unmount) τότε ο διακόπτης αυτός γίνεται “Clean”. Έτσι ο πυρήνας μπορεί να αποφασίσει αν το Σ.Α. χρειάζεται έλεγχο.

Υπάρχει επίσης μία ιδιότητα που επιτρέπει την ασφαλή διαγραφή συγκεκριμένων αρχείων. Όταν ένα τέτοιο αρχείο διαγραφεί, τυχαία δεδομένα γράφονται πάνω στα μπλοκ που ήταν γραμμένο το αρχείο, ώστε να είναι πρακτικά αδύνατη η ανάγνωση των περιεχομένων του διαγραφμένου αρχείου με οποιοδήποτε τρόπο.

Μία δυνατότητα που πήρε από το BSD είναι το ότι τα immutable (αμετάβλητα) αρχεία, δεν μπορούν με κανένα τρόπο να αλλαχθούν τα περιεχόμενα τους, ή να διαγραφούν. Επίσης υπάρχουν τα append-only (συμπλήρωση μόνο) αρχεία, τα οποία μπορούν με να ανοιχθούν σε κατάσταση επεξεργασίας, αλλά τα καινούρια δεδομένα προστίθενται μόνο στο τέλος του αρχείου. Αυτό μπορεί να είναι πολύ χρήσιμο σε αρχεία καταγραφής (log files) τα οποία μπορούν μόνο να μεγαλώσουν με το πέρασμα του χρόνου (Seltzer, Bostic, McKusick, & Staelin, 1993).

Επίσης το ext2 μπορεί να υποστηρίξει την δυνατότητα execute in place μέσω αναβάθμισης (Otte, 2005).

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό περιγράφηκε λεπτομερώς το Σύστημα Αρχείων Ext2. Παρατίθενται οι κυριότερες δυνατότητες του Σ.Α. καθώς και τα χαρακτηριστικά εκείνα που το έχουν κάνει αναμφισβήτητο κυρίαρχο των εγκαταστάσεων των περισσότερων διανομών του Linux.

Στο επόμενο κεφάλαιο παρατίθεται μία σύγκριση των δύο Σ.Α. που περιγράφηκαν προηγουμένως, του FAT και του Ext2

ΚΕΦΑΛΑΙΟ 5

Διαφορές μεταξύ Ext2 και FAT (16/32)

ΕΙΣΑΓΩΓΗ

Εκτός από τις κλασσικές διεργασίες πάνω σε αρχεία και φακέλους που προφανώς υποστηρίζουν και τα δύο συστήματα αρχείων, υπάρχουν κάποιες επιπλέον δυνατότητες που υποστηρίζει μόνο το Ext2, όπως αποτυπώνονται στον Πίνακα 11.

Πίνακας 12: Επιπλέον δυνατότητες του Ext2 συγκριτικά με το FAT

File System	Hard Links	Symbolic Links	Case Sensitive	Case Preserving	XIP ¹
Ext2	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
FAT16/32	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ

Πίνακας 13: Περιορισμοί στην ονοματοδοσία FAT, Ext2

Σύστημα Αρχείων	Μάξιμου μήκος ονόματος	Επιτρεπτοί χαρακτήρες σε καταχωρήσεις φακέλων	Μάξιμου μήκος διαδρομής	Μάξιμου μέγεθος αρχείου	Μάξιμου μέγεθος τόμου
FAT16	8.3 (255 UTF-16 code units LFN)	Όλοι οι Unicode εκτός του κενού (null)	δεν υπάρχει περιορισμός	2 GB	16 MB to 2 GB
FAT32	8.3 (255 UTF-16 code units LFN)	Όλοι οι Unicode εκτός του κενού (null)	δεν υπάρχει περιορισμός	4 GB	512 MB to 8 TB
ext2	255 bytes	Όλοι οι byte εκτός του κενού (null) και του \	δεν υπάρχει περιορισμός	16 GB to 2 TB	2 TB to 32 TB

Το Ext2 υποστηρίζει την δυνατότητα τόσο σκληρών όσο και συμβολικών δεσμών που αναλύθηκε προηγουμένως, όπως και την διαφορετική καταχώρηση ονομάτων αρχείων και φακέλων όσον αφορά τα κεφαλαία ή μη γράμματα (ευθεία κληρονομιά από το Unix). Επίσης μέσω αναβάθμισης υποστηρίζει την εκτέλεση επί τόπου, μία δυνατότητα που δίνει την δυνατότητα στο Σ.Α. να εκτελεί διεργασίες

¹ eXecute In Place

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

χρησιμοποιώντας πόρους του δίσκου ή τόμου στο οποίο βρίσκεται το υπό εκτέλεση αρχείο (Roselli, Lorch, & Anderson, 2000), ("File system guide (Linux Reviews)," 2005), (NISTOR, 2010).

ΣΥΜΠΕΡΑΣΜΑΤΑ

Είναι αναμφισβήτητο γεγονός ότι τα Συστήματα Αρχείων της Microsoft FAT16 & FAT32 είναι πλέον παρωχημένα και ανεπαρκή. Δεν προσφέρουν καμία δυνατότητα εκτός των απολύτως απαραίτητων λειτουργιών που πρέπει να παρέχει ένα Σύστημα Αρχείων. Για αυτό και η Microsoft έχει στραφεί στο NTFS το οποίο είναι αρκετά πιο σύγχρονο.

Αντιθέτως το Ext2 είναι ένα μοντέρνο Σύστημα Αρχείων, με πολλές προηγμένες δυνατότητες και δεν είναι τυχαίο το ότι είναι το δημοφιλέστερο Σ.Α. ανάμεσα σε όλες τις διανομές του Linux. Η αναβάθμιση του μάλιστα στο Ext3 το οποίο θα διασφαλίζει την απόλυτη συμβατότητα προς τα πίσω, θα του προσδώσει πολλές καινούριες δυνατότητες και σίγουρα θα το βοηθήσει να διατηρήσει την πρωτοκαθεδρία του μεταξύ των Σ.Α. σε Linux.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Bach, M. J. (1986). *The design of the UNIX operating system* (Vol. 5). Bell Telephone Laboratories: Prentice-Hall London.
- Bryant, R., Forester, R., & Hawkes, J. (2002). Filesystem performance and scalability in linux 2.4. 17. In *Proceedings of the FREENIX Track: 2002 USENIX Annual Technical Conference* (pp. 259–274).
- Card, R., Ts'o, T., & Tweedie, S. (1994). Design and implementation of the second extended filesystem. In *Proceedings of the First Dutch International Symposium on Linux* (pp. 90–367).
- File system guide (Linux Reviews). (2005). . Retrieved March 19, 2011, from <http://linuxreviews.org/sysadmin/filesystems/>
- Kleiman, S. R. (1986). Vnodes: An architecture for multiple file system types in Sun UNIX. In *USENIX Summer* (pp. 238–247).
- McKusick, M. K., Joy, W. N., Leffler, S. J., & Fabry, R. S. (1984). A fast file system for UNIX. *ACM Transactions on Computer Systems (TOCS)*, 2(3), 181–197.
- Microsoft, T. (2003a). How FAT Works: Local File Systems. Retrieved March 17, 2011, from [http://technet.microsoft.com/en-us/library/cc776720\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc776720(WS.10).aspx)
- Microsoft, T. (2003b). What Is FAT?: Local File Systems. Retrieved March 17, 2011, from [http://technet.microsoft.com/pt-br/library/cc738068\(WS.10\).aspx](http://technet.microsoft.com/pt-br/library/cc738068(WS.10).aspx)
- NISTOR, C. (2010). Open Source–Linux vs. Windows A platform comparison. *Open Source Science Journal*, 2(2).
- Otte, C. (2005). ext2: add execute in place support [LWN.net]. Retrieved March 20, 2011, from <http://lwn.net/Articles/135442/>
- Phillips, D. (2001). A directory index for ext2. In *5th Annual Linux Showcase and Conference* (pp. 173–182).
- Poirier, D. (2011). The Second Extended File System. *The Second Extended File System*. e-Book, . Retrieved March 20, 2011, from <http://www.nongnu.org/ext2-doc/ext2.html>
- Roselli, D., Lorch, J. R., & Anderson, T. E. (2000). A comparison of file system workloads. In *Proceedings of the annual conference on USENIX Annual Technical Conference* (pp. 4–4).
- Seltzer, M., Bostic, K., McKusick, M. K., & Staelin, C. (1993). An implementation of

a log-structured file system for UNIX. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings* (pp. 3–3).

Sivathanu, G., Sundararaman, S., & Zadok, E. (2006). Type-safe disks. In *Proceedings of the 7th symposium on Operating systems design and implementation* (pp. 15–28).

Wikipedia. (2011). Comparison of file systems - Wikipedia, the free encyclopedia. Retrieved March 21, 2011, from http://en.wikipedia.org/wiki/Comparison_of_file_systems

ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ Partition Surprise

1. Εγχειρίδιο χρήστη

1.1 Περιγραφή του έργου

Το 'Partition Surprise' είναι ένας κλώνος του Partition Magic της PowerQuest. Μπορεί και υλοποιεί αλλαγές μεγέθους των κατατμήσεων, μετακινήσεις και μετατροπές. Επίσης περιλαμβάνει και περιορισμένη υποστήριξη λειτουργιών σε χρησιμοποιούμενη κατάτμηση ext2 (mounted).

Χαρακτηριστικά του έργου

Το Partition Surprise περιέχει πλήρη υποστήριξη για την αλλαγή της δομής κατατμήσεων των δίσκων. Μπορεί να υλοποιεί αλλαγή μεγέθους της κατάτμησης, μετακινήσεις αρχείων και μετατροπές. Τα χαρακτηριστικά που υλοποιούνται είναι τα ακόλουθα:

1. **Επεξεργασία δομής της κατάτμησης:** δημιουργία, διαγραφή, καθάρισμα, μετακίνηση και αντιγραφή όλων των τύπων των κατατμήσεων.
2. **Λήψη πληροφοριών συστήματος αρχείων:** λήψη πληροφοριών για συστήματα αρχείων (μέγεθος, παράμετροι,...).
3. **Αλλαγή μεγέθους κατάτμησης:** αυξομείωση των μη επιλεγμένων ext2 κατατμήσεων και αλλαγή μεγέθους αρχείων με βάση το FAT και μετατροπή της κατάτμησης σε ένα βήμα.
4. **Μετατροπή των συστημάτων αρχείων:** μετατροπή του συστήματος αρχείων μεταξύ ext2, fat12, fat16 (v) και (v) fat32 αλλάζοντας τις εσωτερικές παραμέτρους των συστημάτων αρχείων (μέγεθος των συμπλεγμάτων των FAT-συστημάτων αρχείων, τον αριθμό των inodes και blocksize του ext2-συστήματος αρχείων).
5. **On-line υποστήριξη αλλαγής μεγέθους των κατατμήσεων ext2:** την πλήρη υποστήριξη του μεγέθους της επιλεγμένης κατάτμησης ext2, ανάπτυξη του πυρήνα, ενημερωμένες εκδόσεις κώδικα για 2.4.x πυρήνα linux για αυτή τη δυνατότητα.

6. **Διεθνής υποστήριξη:** Με βάση το GNU gettext διεθνή υποστήριξη της αγγλικής και της τσέχικης γλώσσας.
7. **Υποστήριξη LILO:** αυτόματη αλλαγή πληροφοριών LILO μετά από ουσιαστικές τροποποιήσεις για να είστε σε θέση να εκκινήσετε με επιτυχία.
8. **Διαφορετικές διεπιφάνειες χρήστη:**
 - Περιβάλλον χρήστη με γραμμή εντολών,
 - GTK +/- περιβάλλοντος εργασίας Gnome,
 - Περιβάλλον χρήστη Curses/S-Lang.

Θέματα φορητότητας

Ο μεταγλωττιστής GNU C απαιτείται για επιτυχή καθορισμό του συστήματος. Το προϊόν είναι συμβατό με POSIX και συνεπώς θα είναι σε θέση να εκτελούνται διαφορετικά τερματικά unix όπου εκτελείται το gcc.

Μελλοντικές δυνατότητες

Το έργο είναι αρκετά δυναμικό και ευέλικτο και έτσι είναι δυνατόν να προστεθεί υποστήριξη και για άλλο σύστημα αρχείων καθώς και νέα περιβάλλοντα εργασίας χρήστη.

Αδειοδότηση

Το προϊόν κυκλοφορεί υπό την άδεια της GPL καθώς και με SuSE copyrights.

1.2 Προειδοποιήσεις

Το Partition Surprise προϋποθέτει ότι τα δεδομένα στις κατατμήσεις βρίσκονται σε σωστή κατάσταση. Γι' αυτό τον λόγο θα πρέπει να βεβαιωθείτε εκτελώντας την fsck. Αν δεν το πραγματοποιήσετε, υπάρχει φόβος το Partition Surprise να εντοπίσει λάθη και να τερματίσει την διαδικασία. Αφού ελεγχθούν οι κατατμήσεις, πρέπει να αποφορτωθούν (unmounts), αλλιώς το Partition Surprise δεν θα μπορέσει να τις χρησιμοποιήσει.

Όλες μετατροπές εκτελούνται σε δύο φάσεις. Στην πρώτη φάση το νέο Σύστημα Αρχείων δημιουργείται στα ελεύθερα blocks του παλιού – όπου μπορούμε να διακόψουμε την διαδικασία χωρίς κάτι να χαθεί. Στην δεύτερη φάση γίνεται

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

εξαγωγή των blocks του παλιού Συστήματος Αρχείων στην τελική τους θέση. Αυτή η φάση δεν πρέπει ΠΟΤΕ να διακόπτεται, αλλιώς είναι σίγουρη η απώλεια όλων των δεδομένων σε αυτή την κατάτμηση. Συνιστούμε να γίνεται αντίγραφο ασφαλείας των αρχείων πριν εκτελέσετε το Partition Surprise προς αποφυγή όλων των παραπάνω.

Γνωστά προβλήματα

- Επί του παρόντος δεν υποστηρίζεται στο LILO απευθείας μετατροπή με υποστήριξη της επιλογής "*compact*" (Μπορεί να ενεργοποιηθεί από το `/etc/lilo.conf` χωρίς όμως να υποστηρίζεται).

1.3 Συγγραφείς

Jan Kara

jkara8572@users.sourceforge.net

- online μετακίνηση και αλλαγή μεγέθους του ext2 συστήματος αρχείων
- curses client

Daniel Kral

kral@users.sourceforge.net

- cmdline client
- text server
- PART modul
- misc. PTBL types support

Jan Kratochvil

kratochvil@users.sourceforge.net

- gnome client
- Υποστήριξη linux swap
- Απευθείας υποστήριξη LILO

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

- Σύστημα λαθών Surprise
- τμήματα της διαδικασίας (e.g. autogen, RPM packaging)

Robert Spalek

spalek@users.sourceforge.net

- Υποστήριξη παραλλαγών Fat
- Εξαγωγή Ext2
- SGML τεκμηρίωση διεργασίας
- Κώδικας παραμέτρων γενικού συστήματος αρχείων
- Υποστήριξη autoconf/automake

Tomas Tichy

xtom@users.sourceforge.net

- general Conversion module
- εισηγμένο Ext2
- σύστημα προόδου
- Μετάφραση στα Τσέχικα
- ασφαλή διαγραφή δεδομένων

2. Χρήστες Partition Surprise

2.1 Χρήση Γραμμής εντολών

Η εκτέλεση αυτής της λειτουργίας γίνεται με την χρήση της γραμμής εντολών. Ο χρήστης καλείται να εκτελέσει μία εντολή για να εμφανιστεί το αποτέλεσμα. Η κατάσταση της λειτουργίας φαίνεται από μία μπάρα προόδου.

Το Partition Surprise μεταγλωττίζει χρησιμοποιώντας **readline** βιβλιοθήκη ως προεπιλογή, προσφέροντας έτσι μια άνετη γραμμή εισόδου (με γραμμή επεξεργασίας δυνατοτήτων, ιστορικό εντολών,...).

Κατά την εκκίνηση, το Surprise αναζητά όλες τις διαθέσιμες συσκευές και διαβάζει τους πίνακες των κατατμήσεων και τις κεφαλίδες των συστημάτων αρχείων. Ο χρήστης μπορεί να επιλέξει το σύστημα αρχείων και να το επεξεργαστεί. Τέλος μπορεί να επιβεβαιώσει την ακεραιότητα των αλλαγών. Εάν είναι όλα σωστά τότε μπορεί να προχωρήσει στην μετατροπή τους. Αυτό μπορεί να διαρκέσει αρκετή ώρα, γνωρίζοντας βέβαια την πρόοδο της διαδικασίας από την μπάρα.

Το Surprise τυπώνει όλες τις σημαντικές πληροφορίες της διαδικασίας καθώς και όλα τα μηνύματα λάθους.

Βασικές εντολές

quit

Αυτή η εντολή κλείνει το πρόγραμμα της γραμμής εντολών άμεσα, αγνοώντας όλες τις αλλαγές που πραγματοποιήθηκαν χωρίς να ζητά επιβεβαίωση από τον χρήστη.

version

Τυπώνει την έκδοση της βιβλιοθήκης του Surprise.

help

Τυπώνει μια πολύ απλή βοήθεια – μία λίστα όλων των εντολών με τις συντάξεις τους και μία σύντομη περιγραφή.

start

Επανεκκινεί το Partition Surprise. Όλες οι αλλαγές θα παραβλεφθούν και η διαδικασία θα ξεκινήσει από την αρχή. Μπορείτε να ορίσετε συμπληρωματικά ορισμένες πρόσθετες συσκευές ή αρχεία (π.χ. εάν έχετε ένα αντίγραφο ενός δίσκου σε ένα αρχείο) για να χρησιμοποιηθεί από την βιβλιοθήκη.

adddevice name

Προσθέτει ένα συγκεκριμένο αρχείο στην λίστα των συσκευών προς χρήση όταν η διαδικασία start ξεκινήσει.

listdevice

Εκτυπώνει τον κατάλογο με τις συσκευές προς χρήση όταν η διαδικασία start ξεκινήσει.

rmdevice name

Καταργεί ένα συγκεκριμένο αρχείο στην λίστα των συσκευών προς χρήση όταν η διαδικασία start ξεκινήσει.

verify

Συγκρίνει τις παλιές παραμέτρους με τις νέες και ελέγχει για σφάλματα:

- Οι κατατμήσεις ΔΕΝ πρέπει να επικαλύπτονται και πρέπει να χωράνε στον δίσκο.
- Η λογική δομή του πίνακα κατάτμησης(ειδικά του πίνακα της διαμόρφωσης) πρέπει να είναι σωστή.
- Το σύστημα αρχείων δεν μπορεί να επιμηκυνθεί εκτός των ορίων (τα αρχεία πρέπει να χωρούν).
- Οι κατατμήσεις προς αλλαγή ΔΕΝ πρέπει να χρησιμοποιούνται.
- Οι ειδικές παράμετροι του συστήματος αρχείων πρέπει να ρυθμιστούν με τις σωστές τιμές (ελέγχεται από το σύστημα αρχείων).

Εάν εντοπιστεί κάποιο σφάλμα, το Partition Surprise δεν πραγματοποιεί τις αλλαγές και ο χρήστης θα πρέπει να ελέγξει το μήνυμα του λάθους.

confirm

Αυτή η εντολή ελέγχει την τρέχουσα κατάσταση και εφ' όσον η εντολή *verify* δεν έχει εκτελεστεί. Εάν όλα έχουν πάει καλά, τότε εκτελούνται οι αλλαγές στον δίσκο.

Ορισμένες από τις λειτουργίες εκτελούνται πολύ γρήγορα:

- Δημιουργία και διαγραφή κατατμήσεων, έστω και αν περιέχουν ένα σύστημα αρχείων.
- Αλλαγή των ιδιοτήτων superbloc που δεν επηρεάζουν πολύ το σύστημα αρχείων (π.χ. *volume_name*, *max_mount_count*, ...).
- Η μετακίνηση και η αντιγραφή μίας κατάτμησης δεν πραγματοποιείται τόσο γρήγορα, αλλά εξαρτάται από την ταχύτητα του δίσκου και μόνο. Δεν πραγματοποιείται κάποια μετατροπή, το φυσικό block μεταφέρεται σε άλλο μέρος.

Όλες οι υπόλοιπες λειτουργίες εκτελούνται σε πιο αργό τέμπο, λόγω της δημιουργίας του νέου συστήματος αρχείων και της μεταφοράς των αρχείων αυτών:

- Αλλαγή μεγέθους ενός συστήματος αρχείων. Ορισμένες εφαρμογές του συστήματος αρχείων (π.χ. **GNU parted**) υποστηρίζουν γρήγορη μετατροπή του μεγέθους, το Partition Surprise όμως εκτελεί μια αργή ανοικοδόμηση του συστήματος αρχείων τις περισσότερες φορές. Η μόνη διαθέσιμη γρήγορη μετατροπή είναι η online αλλαγή μεγέθους *ext2* συστήματος αρχείων.).
- Αλλαγή σημαντικών χαρακτηριστικών του superbloc (π.χ. *block_size*, *fat_bits*, ...).
- Μετατροπή σε ένα άλλο σύστημα αρχείων.

Ορισμένες λειτουργίες μπορούν να διακοπούν με ασφάλεια. Ο χρήστης μπορεί να πιέσει το **Συνδυασμό πλήκτρων Ctrl + C** οποιαδήποτε στιγμή και το Surprise θα αποφασίσει: Εάν η διακοπή της μετατροπής αφήνει τα δεδομένα σε σωστή κατάσταση, η λειτουργία ακυρώνεται. Εάν όχι, συνεχίζει τη λειτουργία. Εάν ο χρήστης θέλει πραγματικά να διακόψει την διαδικασία, πρέπει να την τερματίσει ο ίδιος **kill -9** και σε αυτήν την περίπτωση θα είναι υπεύθυνος για την απώλεια των δεδομένων. Οι λειτουργίες που μπορούν να ακυρωθούν είναι:

- Μετακίνηση μίας κατάτμησης σε τόπο που δεν επικαλύπτεται από την αρχική του θέση.

- Γενική μετατροπή του συστήματος αρχείων πριν εκτελεστεί η τελική αλλαγή (το Surprise δημιουργεί ένα νέο σύστημα αρχείων στο ελεύθερο block της παλαιάς και αντιγράφει τις πληροφορίες σχετικά με την τοποθέτηση αρχείων σε αυτό. Αφού έχουν ολοκληρωθεί τα πάντα, το παλαιό σύστημα αρχείων καταστρέφεται κατά την τοποθέτηση των τελικών τμημάτων αρχείων του νέου συστήματος.)

Αφού ολοκληρωθούν όλα, το Partition Surprise επανεκκινείται.

Προβολή των πρωτότυπων/αρχικών κατατμήσεων

Οι περισσότερες εντολές λειτουργούν σε μία κατάτμηση. Ο χρήστης μπορεί να χρησιμοποιήσει δύο (2) τρόπους για να καθορίσει μία κατάτμηση / δίσκο. Μπορεί να γράψει μια πλήρη διαδρομή σε μια συσκευή (π.χ. /dev/hdc, /dev/hda3) ή να παραλείψει το πρόθεμα της διαδρομής και απλά να καθορίσει ένα όνομα συσκευής (π.χ. hdc, hda3).

info, info [δίσκος], info [κατάτμηση]

Η πρώτη (γράφοντας info) τυπώνει μια λίστα δίσκων που βρέθηκε στον υπολογιστή.

```
> info
Found 2 device(s): /dev/hda /dev/hdc
```

Πληκτρολογώντας info hda ο χρήστης ενημερώνεται σχετικά με το μοντέλο του δίσκου, την δυναμικότητα, την γεωμετρία, την στοίχιση της κατάτμησης και μια λίστα με τις κατατμήσεις που βρέθηκαν.

```
> info hda
Device: /dev/hda
Model: WDC AC310100B
Size: 19807200 sectors (10141286400 bytes)
```

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

9903600 kB

Heads/sectors/cylinders: 16/63/19650

Sectors reserved: 63

Partitions start/end aligned to 63/1008 sectors

4 partition(s): /dev/hda1 /dev/hda2 /dev/hda5 /dev/hda6

Πληκτρολογώντας `info` ο χρήστης ενημερώνεται σχετικά με την αρχική τοποθέτηση της κατάτμησης (τομείς, KB και την γεωμετρία του δίσκου), τον τύπο της κατάτμησης (όπως φαίνεται στον πίνακα της κατάτμησης), τα συστήματα των αρχείων που εντοπίστηκαν και οι εισηγμένες τιμές των παραμέτρων του αρχείου συστήματος (δείτε εντολή `detected`).

> info hda6

Partition: /dev/hda6

First sector: 5134815

Distance from beginning: 2567407 kB

Length: 14672385 sectors

7336193 kB

Partition entry:

First sector: 63

Length (in sectors): 14672385

First head/sector/cylinder: 0F/FF/FF

Last head/sector/cylinder: 0F/FF/FF

Partition type: 83

Partition boot flag: 00

Detected filesystem: Linux ext2

block_size: 1024

inodes_count: 1827840

bytes_per_inode: 4109

blocks_per_group: 8192

reserved_percent: 3

max_mount_count: 20

check_interval: 15552000

error_handling: continue

volume_name: LINUX

codepage [10]: ISO-8859-1

force_block_size [11]: no

force_inodes [12]: none

Data size: 5855171 kB (11710341 512-byte blocks)

free [δίσκος]

Αυτή η εντολή εκτυπώνει από το συγκεκριμένο δίσκο όλα τα ελεύθερα blocks που βρίσκονται σε αυτόν (ανάγνωση από τον πίνακα της κατάτμησης). Δουλεύει λαμβάνοντας υπ' όψιν του την αρχική κατάσταση της κατάτμησης (αγνοεί οποιαδήποτε μετακίνηση ή αλλαγή μεγέθους). Δεν επεξεργάζεται δεσμευμένα blocks (τα 62/63 ελεύθερα block στο ακόλουθο παράδειγμα δεν μπορούν να χρησιμοποιηθούν, λόγω της ευθυγράμμισης της κατάτμησης).

```
> free hda
```

```
Device: /dev/hda
```

```
Size: 19807200 sectors (10141286400 bytes)
```

```
Sectors reserved: 63
```

```
Partitions start/end aligned to 63/1008 sectors
```

```
Free space on device:
```

Start	End	Length
1	62	62
105840	105902	63
4612608	4612670	63
5134752	5134814	63

state, state [δίσκος], state [κατάτμηση]

Αυτή η εντολή διαβάζει την αρχική και την νέα κατάσταση ,και τις συγκρίνει μεταξύ τους. Ο χρήστης ενημερώνεται σχετικά με την τοποθέτηση της κάθε κατάτμησης. Χωρίς τις παραμέτρους, απλά ενημερώνει για το άθροισμα των κατατμήσεων:

```
> state
```

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

2 device(s) in system

Original state	Planned state
/dev/hda 4 partition(s)	/dev/hda 4 partition(s)
/dev/hdc 3 partition(s)	/dev/hdc 3 partition(s)

Αν δοθεί ως παράμετρος ο δίσκος, εκτυπώνει δύο πίνακες κατατμήσεων. Οι θέσεις καταμετρούνται σε μεγάλα τμήματα των 512b ως προεπιλογή. Εάν δοθεί προαιρετικά ως παράμετρος **-kB**, συνυπολογίζονται σε kilobyte.

```
> state hda
Device /dev/hda
63 block(s) reserved at beginning, partition start/end aligned to
multiples of 63/1008
Original state:
Partition      Start      Length     Filesystem
/dev/hda1      105903     4506705    fat
/dev/hda2           63        105777     Linux ext2
/dev/hda5      4612671    522081     Linux swap
/dev/hda6      5134815    14672385   Linux ext2
Planned state:
Partition      Start      Length     Filesystem
/dev/hda1      105903     4506705    fat
/dev/hda5      4612671    522081     Linux swap
/dev/hda6      5134815    12582801   Linux ext2
/dev/hda7      6500025    1023687    fat
```

Αν δοθεί ως παράμετρος η *κατάτμηση*, πληροφορεί τον χρήστη σχετικά με την τοποθέτηση της σχεδιαζόμενης κατάτμησης και τις τιμές των παραμέτρων του εξαγόμενου συστήματος αρχείων (η έξοδος είναι παρόμοια με την εντολή info, χωρίς τις πληροφορίες για την γεωμετρία).

Εξερεύνηση των γενικών παραμέτρων δίσκου / κατάτμησης

Εάν ο χρήστης θέλει να καθορίσει την τιμή μιας παραμέτρου μπορεί να το κάνει με δύο τρόπους:

1. Από το όνομα της παραμέτρου: απλή αναγραφή του ονόματος, π.χ. *block_size*. Αυτό γίνεται λίγο περίπλοκο όταν η παράμετρος έχει πολύ μεγάλο όνομα. Στην περίπτωση αυτή ο χρήστης πρέπει να εσωκλείσει το όνομα μέσα σε ``` (π.χ. *"Number of bootable partition"*).
2. Με τον αριθμό των παραμέτρων: Σημειώστε τον αριθμό μέσα σε «[]», π.χ. [9]. Αυτός είναι ένας πολύ απλός τρόπος, αλλά ο χρήστης πρέπει να αριθμήσει από την για να τους χρησιμοποιήσει.

gparam

Εξετάζει όλες τις γενικές παραμέτρους του Partition Surprise:

- **On-line μετατροπή:** Εάν επιλεγθεί, το Surprise επιτρέπει στον πυρήνα Linux, να αλλάξει το μέγεθος μίας επιλεγμένης (mounted) κατάτμησης **ext2** on-line. Χρειάζεται ένα ειδικά τροποποιημένο kernel για την διαδικασία αυτή (διανομή με το Partition Surprise).
- **Υποστήριξη Lilo:** Εάν επιλεγθεί, η διαδρομή `lilo.conf` πρέπει να τροποποιηθεί, αφού εκτελεστούν όλες οι αλλαγές. Εάν επιλεγθεί *πλήρη υποστήριξη*, τότε οι εσωτερικές δομές δεδομένων του LILO (που περιέχουν και το αρχείο τοποθέτησης του kernel) τροποποιούνται αυτόματα, χωρίς να χρειάζονται έτσι πρόσθετες κλήσεις του **lilo**.
- **Άδειος χώρος συσκευής:** Εάν επιλεγθεί, ο χώρος που είναι ελεύθερος (εξαιτίας διαγραφής ή μείωσης μεγέθους) διαγράφεται προς αποφυγή ανάγνωσης παλαιών προσωπικών δεδομένων. Παρέχονται 2 αλγόριθμοι για αυτό: *γέμισμα με μηδενικά* και *ασφαλή διαγραφή* (σε πολλές φάσεις).
- **Δοκιμές κατεστραμμένου block:** Εάν επιλεγθεί, οι χαλασμένοι τομείς ελέγχονται πριν εκτελεστεί οποιαδήποτε μετατροπή.

Εάν καλέσουμε την `gparam` χωρίς παραμέτρους, τυπώνεται η τρέχουσα κατάσταση των παραμέτρων. Εάν εισάγεται ένα αναγνωριστικό της παραμέτρου και μία τιμή, η τιμή θα αποδοθεί στην παράμετρο:

```
> gparam
On-line conversion [1]: no
Lilo support [2]: none
Empty device space [3]: no filling
Bad block testing [4]: none

> gparam [2] simple
...
Lilo support [2]: simple
Value of parameter "Lilo support" successfully changed
```

detected [κατάτμηση]

Εξετάζει τις παραμέτρους των νέων συστημάτων αρχείων. Οι περισσότερες από αυτές μπορούν μόνο να προβληθούν (αυτές που δεν έχουν αντιστοιχηθεί με αριθμό), επειδή είναι πλήρως ανιχνεύσιμες (π.χ. *block_size*, *volume_name*, etc...), αλλά μερικές κατεστραμμένες παράμετροι του συστήματος αρχείων δεν μπορούν να αναγνωριστούν το ίδιο εύκολα (π.χ. *fat_bits* στο FAT), μερικά από αυτά χρησιμοποιούνται για να συμπληρώσετε τις πληροφορίες που λείπουν (π.χ. *uid*, *gid*, *mask* στο FAT) και ορισμένα από αυτά που δεν έχουν οριστεί (π.χ. η κωδικοσελίδα που περιέχει την κωδικοσελίδα του ονόματος αρχείου στην FAT/ext2).

Η σύνταξη είναι σχεδόν η ίδια με την εντολή `gparam`:

```
> detected hda1
Partition: /dev/hda1
Detected filesystem: fat
cluster_size: 4096
fat_bits [2]: 32
fats: 2
rootdir_entries: 512
codepage [5]: IBM437
uid [6]: 0
gid [7]: 0
mask [8]: 420
```

```
system_id: MSWIN4.1
volume_name: WINDOWS
fat_name: FAT32
long_name_type [12]: vfat
force_fat_bits [13]: no
save_bootsector [14]: yes

> detected hda1 [2] 17
Partition: /dev/hda1
Detected filesystem: fat
...
Number parameter [2] can have only one of the following value:
    12
    16
    32

> detected hda1 [2] 16
Partition: /dev/hda1
Detected filesystem: fat
...
fat_bits [2]: 16
Value of parameter "fat_bits" successfully changed
```

param [δίσκος], param [κατάτμηση]

Χρησιμοποιείται για να εξερευνήσετε τις παραμέτρους των εξαγόμενων κατατμήσεων / συστημάτων αρχείων. Οι παράμετροι με αντιστοιχισμένο αριθμό, μπορούν να τροποποιηθούν χειροκίνητα, οι υπόλοιποι υπολογίζονται αυτόματα και προσφέρονται μόνο για ανάγνωση (κατόπιν εκτέλεσης της εντολής **verify**).

Οι παράμετροι του συστήματος αρχείων, εξηγούνται στο τμήμα

Παράμετροι των συστημάτων αρχείων , οι παράμετροι της κατάτμησης είναι οι παρακάτω:

- **Τύπος πίνακα κατάτμησης:** Το Partition Surprise υποστηρίζει περισσότερους τύπους πινάκων κατατμήσεων (από ένα παραδοσιακό \$ MBR MSDO): *DOS*, *DOS + BSD*, *Mac*. Μπορεί να

έχει πρόσβαση και στους τρεις (3) τύπους πινάκων των κατατμήσεων καθώς και της μετατροπής μεταξύ τους.

- **Αριθμός των εκκινήσιμων κατατμήσεων:** περιέχει τον αριθμό των κατατμήσεων, που πρόκειται να εκκινηθούν.
- **Αριθμός των κατατμήσεων στο σχήμα PC PTBL:** περιέχει τον αριθμό των κατατμήσεων, των οποίων οι εγγραφές αποθηκεύονται σε σχήμα PC του πίνακα κατάτμησης, δηλαδή αποκλείονται οι κατατμήσεις που είναι αποθηκευμένα στο BSD διαμέρισμα `disklabel`.
- **Αριθμός των BSD κατατμήσεων 'a':** περιέχει τον αριθμό της κατάτμησης στο BSD διαμέρισμα `disklabel` με το όνομα «a».

Η σύνταξη είναι η ίδια με την εντολή `detected`.

Οι εργασίες που υποστηρίζονται από το `Partition Surprise`

Ορισμένες εντολές υποστηρίζουν επιλογές. Συντάσσονται με το σύμβολο της αφαίρεσης στην αρχή, π.χ. `-roundup`, μετά από το όνομα της εντολής και πριν από όλες τις άλλες παραμέτρους που δίνονται στην εντολή.

`new [-renumber] [-override] [-noalign] [-roundup] partition start length[kMG] [type] [hex]`

Αυτή η εντολή δημιουργεί μία νέα κατάτμηση. Εάν χρειάζεται να διευκρινιστεί ο τύπος του συστήματος αρχείων, η εντολή `mkfs` εκτελείται κατόπιν. Οι παράμετροι της εντολής έχουν την ακόλουθη έννοια:

- **partition:** το όνομα της κατάτμησης (π.χ. `hda3`).
- **start:** το αρχικό block (512 bytes μήκος) της κατάτμησης. Πρέπει να είναι ευθυγραμμισμένο στο όριο των κυλίνδρων (κατά το `Surprise`). Εάν δεν καθοριστεί η παράμετρος **-noalign**, θα τυπωθεί ένα μήνυμα προειδοποίησης, αλλιώς θα στρογγυλευτεί στο όριο των κυλίνδρων.
- **length:** Μονάδες μέτρησης μίας κατάτμησης είναι τα `blocks`, `kilobytes`, `Megabytes`, or `Gigabytes` (ανάλογα με το επιθεματικό [kMG]).
- **type:** Τύπος του συστήματος αρχείων. Εάν δεν διευκρινίζεται, δεν θα δημιουργηθεί κανένα σύστημα αρχείων. Ο δεκαεξαδικός κώδικας της

κατάτμησης (στο MBR) αποδίδεται αυτόματα από τον κώδικα του συστήματος αρχείων.

- **hex**: ο δεκαεξαδικός κώδικας της κατάτμησης που αποθηκεύεται στο MBR.
- **-renumber**: Εάν επιλεγθεί, αριθμεί εκ νέου αυτόματα τις άλλες κατατμήσεις για να καταστήσει τα ονόματα των κατατμήσεων μοναδικά.

Οι αρχικές κατατμήσεις (στο σχήμα κατατμήσεων MSDOS) είναι αριθμημένα από το 1 έως το 4. Η εκτεταμένη κατάτμηση αποκρύπτεται από το Partition Surprise, και οι λογικοί τόμοι στη μοναδική εκτεταμένη κατάτμηση αριθμούνται από το 5. Το Partition Surprise αναλαμβάνει γι' αυτό.

- **-override**: Εάν επιλεγθεί, η γεωμετρία θα παρακαμφθεί από το χρήστη, ακόμα και αν προκληθεί σφάλμα. Οι έλεγχοι γίνονται και τα μηνύματα τυπώνονται, αλλά δεν εκτελούνται αλλαγές από το Partition Surprise. Ο χρήστης πρέπει να γνωρίζει τι κάνει.
- **-noalign**: Εάν επιλεγθεί, η αφετηρία της κατάτμησης δεν θα ευθυγραμμίζεται αυτόματα στο μέγεθος του κυλίνδρου από το Surprise.
- **-roundup**: Εάν επιλεγθεί, το τέλος της κατάτμησης θα στρογγυλοποιηθεί προς τα επάνω αντί για προς τα κάτω.

Οι πρωτεύοντες κατατμήσεις μπορούν να τοποθετηθούν οπουδήποτε, αλλά δεν πρέπει να επικαλύπτονται. Οι λογικοί τόμοι πρέπει να τοποθετούνται συνεχώς στην εκτεταμένη κατάτμηση. Το μέγεθος της εκτεταμένης κατάτμησης γίνεται αυτόματα από το Surprise, αλλά ο χρήστης πρέπει να φροντίσει να τοποθετήσει τους λογικούς τόμους συνεχόμενα. Ένα sector δεσμεύεται από το Surprise για κάθε λογικό τόμο και οι κατατμήσεις πρέπει να οριοθετούνται, έτσι ο χρήστης να διατηρεί περίπου 63 sectors (ένα track) εκ του συνόλου μεταξύ των λογικών τόμων, αλλιώς θα προειδοποιηθεί γι' αυτό.

del [-renumber] partition [κατάτμηση...]

Η εντολή αυτή διαγράφει μία κατάτμηση. Ο χώρος που θα δημιουργηθεί θα είναι διαθέσιμος για άλλη κατάτμηση. Εάν οριστεί η γενική παράμετρος **Empty device space**, τότε ο χώρος θα διαγραφεί με ασφάλεια. Εάν οριστεί η γενική παράμετρος **-renumber**, οι άλλες κατατμήσεις με έξυπνο τρόπο επαναριθμούνται (π.χ. λογικοί τόμοι). Μπορούν να διαγραφούν περισσότερες από μία (1) κατατμήσεις τη φορά .

mkfs partition type

Ένα νέο σύστημα αρχείων θα δημιουργηθεί στη κατάτμηση. Το αρχικό σύστημα αρχείων (προαιρετικά) θα καταστραφεί -- αν θέλετε να αφήσετε τα δεδομένα, χρησιμοποιήστε την εντολή **convert**. Μετά την κλήση αυτής της εντολής, ο χρήστης μπορεί να αντικαταστήσει τις παραμέτρους του εξαγόμενου συστήματος αρχείων με την εντολή **param** .

convert partition type

Το αρχικό σύστημα αρχείων, θα μετατραπεί στο σύστημα αρχείων προορισμού (νέο). Τα υπάρχοντα δεδομένα θα παραμείνουν στη σωστή κατάσταση. Μετά την κλήση αυτής της εντολής, ο χρήστης μπορεί να αντικαταστήσει τις παραμέτρους των εξαγόμενων αρχείων με την εντολή **param**. Οι παράμετροι των εισαγόμενων συστημάτων αρχείων είναι προσβάσιμοι από την εντολή **detected**.

move [-override] [-noalign] partition start

Η κατάτμηση θα μετακινηθεί σε άλλο σημείο έναρξης στον ίδιο φυσικό δίσκο. Οι επιλογές **-override**, **-noalign** έχουν την ίδια χρήση όπως στην εντολή **new**. Εάν ο χρήστης θέλει να μετακινήσει την κατάτμηση σε άλλο δίσκο, πρέπει να αντιγράψει την κατάτμηση πρώτα και να διαγράψει την αρχική.

renumber partition newnumber

Η αρίθμηση της κατάτμησης θα αλλάξει όπως έχει προκαθοριστεί, π.χ. `renumber hda2 σε 3` ,πρόκειται να αλλάξετε το όνομα της κατάτμησης

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

του hda2 σε hda3, αλλά η θέση, το μέγεθος, τα δεδομένα και το σύστημα αρχείων της κατάτμησης διατηρούνται . Η κατάτμηση με τον νέο αριθμό δεν πρέπει να προϋπάρχει.

resize [-roundup] [-override] [-noalign] partition length[kMG]

Η κατάτμηση θα αλλάξει το μήκος της σύμφωνα με το μέγεθος που θα δοθεί ως παράμετρος. Εάν περιέχει ένα σύστημα αρχείων, θα γίνει άμεση αλλαγή του μεγέθους (χωρίς απώλεια δεδομένων).. Οι εντολές **-roundup**, **-override**, **-noalign** έχουν την ίδια χρήση όπως και στην εντολή **new**.

copy source-partition destination-partition

Θα αντιγραφούν τα δεδομένα από την πρώτη κατάτμηση ,στην κατάτμηση προορισμού.

2.2 Χρήση Curses

Η χρήση Curses του Partition Surprise είναι το περιβάλλον χρήστη της βιβλιοθήκης Surprise που υλοποιείται με βάση την βιβλιοθήκη ncurses. Είναι περισσότερο φιλική στον χρήστη από την αντίστοιχή της, την γραμμή εντολών. Δεν προορίζεται να χρησιμοποιηθεί από εντελώς αρχάριους χρήστες. Είναι απαραίτητη μια ιδέα του τί είναι κατάτμηση και τί σύστημα αρχείων. Αυτή η χρήση υποστηρίζει αυτήν τη στιγμή:

- δημιουργία κατάτμησης
- διαγραφή κατάτμησης
- αλλαγή μεγέθους κατάτμησης(και συστήματος αρχείων)
- μετακίνηση κατάτμησης
- μετατροπή του συστήματος αρχείων

Επιλογές

Μπορείτε να καθορίσετε μερικές επιλογές στην γραμμή εντολών ,στην χρήση του curses. Αυτές είναι :

-A

Απενεργοποίηση της επιλογής *Align partition start/end*

(Βλέπε ρυθμίσεις χρήσης για την περιγραφή της επιλογής).

-p sectors

Ενεργοποίηση της επιλογής *Minimal size of free space*

(Βλέπε ρυθμίσεις χρήσης για την περιγραφή της επιλογής).

-l

Η χρήση θα τυπώσει ορισμένα μηνύματα στο stderr. Βεβαιωθείτε ότι μπορείτε να ανακατευθύνετε το stderr σε κάποιο αρχείο όταν χρησιμοποιείτε αυτήν την επιλογή. Διαφορετικά το αποτέλεσμα δεν θα φαίνεται πολύ ωραίο στην οθόνη.

-w

Να μην τυπώνεται προειδοποίηση κατά την εκκίνηση προγράμματος.

-v

Ενεργοποίηση της επιλογής *Automatic verification*

(Βλέπε ρυθμίσεις χρήσης για την περιγραφή της επιλογής).

-c

Ενεργοποίηση της επιλογής *Clever size printing*

(Βλέπε ρυθμίσεις χρήσης για την περιγραφή της επιλογής).

Οθόνη

Κατά την εκκίνηση του προγράμματος θα δείτε την οθόνη να χωρίζεται σε τέσσερα μέρη. Η γραμμή στο κάτω μέρος της οθόνης εμφανίζει τις λειτουργίες που έχουν ανατεθεί στο κάθε λειτουργικό πλήκτρο. Η πρώτη στήλη από αριστερά εμφανίζει τη λίστα με τις συσκευές που εντοπίστηκαν στο σύστημα. Η στήλη στα δεξιά

εμφανίζει τη λίστα των κατατμήσεων της επιλεγμένης συσκευής μαζί με το σημείο έναρξης και το μήκος τους. Το μεγαλύτερο μέρος της οθόνης θα χρησιμοποιηθεί για πληροφορίες σχετικά με την τρέχουσα συσκευή ή κατάτμηση.

Λίστα συσκευών

Στη λίστα συσκευών, μπορείτε να δείτε όλες τις συσκευές του συστήματος. Όταν είναι ενεργή η λίστα συσκευής, εμφανίζεται με ανεστραμμένο χρώμα. Αυτή είναι η ενεργή συσκευή. Μπορείτε να αλλάξετε ενεργή συσκευή μετακινώντας στην λίστα με τα πλήκτρα κατεύθυνσης *up* και *down* βέλος, *page up*, *page down*, *home* ή *end*. Πατώντας *right* βέλος ή *enter*, μπορείτε να μετακινήστε στη λίστα των κατατμήσεων.

Λίστα κατάτμησης

Η λίστα κατατμήσεων εμφανίζει τη λίστα με τις κατατμήσεις και τον ελεύθερο χώρο στην τρέχουσα επιλεγμένη συσκευή. Τα ελεύθερα τμήματα εμφανίζονται στη λίστα ως *ΕΛΕΥΘΕΡΕΣ* καταχωρήσεις. Η καταχώρηση της τρέχουσας επιλεγμένης ελεύθερης κατάτμησης ή τμήματος, εμφανίζεται με ανεστραμμένο χρώμα.

Μπορείτε να μετακινήστε στη λίστα με τον ίδιο τρόπο όπως στη λίστα συσκευών.

Όταν πιέσετε το πλήκτρο *LEFT ΒΕΛΟΣ* θα επιστρέψετε πίσω στη λίστα των συσκευών.

Δεξιά από τη λίστα κατάτμησης εμφανίζονται σύντομες πληροφορίες σχετικά με τον “τομέα εκκίνησης” και το μήκος σε megabyte ,που περιέχει κάθε κατάτμηση.

Παράθυρο Πληροφοριών

Σε αυτό το παράθυρο εμφανίζονται πληροφορίες σχετικά με την τρέχουσα συσκευή (όταν υπάρχουν επιλεγμένες συσκευές) ή την τρέχουσα κατάτμηση (όταν υπάρχουν επιλεγμένες κατατμήσεις).

Οι πληροφορίες που παρέχονται σχετικά με τη συσκευή είναι: μοντέλο δίσκου, το μέγεθος της συσκευής σε 512 byte, αναφερθείσα γεωμετρία της συσκευής, ο αριθμός έναρξης της ευθυγραμμισμένης κατάτμησης, ο αριθμός τερματισμού της

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

ευθυγραμμισμένης κατάτμησης και το μήκος των 512 bytes blocks που είχαν δεσμευτεί πριν από το πρώτη κατάτμηση.

Οι πληροφορίες σχετικά με την κατάτμηση είναι: η αρχική θέση της κατάτμησης (σε blocks των 512 byte), το μήκος της κατάτμησης (σε blocks των 512 byte), ο τύπος του συστήματος αρχείου. Όταν η επιλεγμένη καταχώρηση είναι τμήμα του ελεύθερου χώρου ,εμφανίζεται μόνο η θέση της αρχής και το μήκος του τμήματός της. Στο παράθυρο των πληροφοριών εμφανίζονται επίσης οι τρέχουσες παράμετροι του συστήματος αρχείων. Μπορείτε να βρείτε την περιγραφή τους στην τεκμηρίωση του συστήματος αρχείων.

Ρυθμίσεις χρήσης

Η χρήση curses έχει κάποιες επιλογές για να ορίσετε. Μπορούν είτε να οριστούν από τη γραμμή εντολών όταν εκτελείται το πρόγραμμα ,είτε από παράθυρο διαλόγου το οποίο ανοίγει με την επιλογή *Client settings* από το μενού.

Align partition start/end

Όταν επιλεχθεί αυτή η εντολή, τότε το πρόγραμμα στοιχίζει ή δεν στοιχίζει τις κατατμήσεις που έχουν δημιουργηθεί.

Minimal size of free space

Αυτή η επιλογή ορίζει τον ελάχιστο αριθμό των sectors για τον ελεύθερο χώρο που θα εμφανίζονται στη λίστα των κατατμήσεων.

Automatic verification

Εκτελεί αυτόματα έλεγχο μετά από κάθε βήμα. Όταν η επαλήθευση αποτυγχάνει να επιβεβαιώσει την ενέργεια του χρήστη (π.χ.. Δημιουργία κατάτμησης, μετατροπή κ.λπ.) τότε αυτή δεν εκτελείται. Αυτή η δυνατότητα μπορεί να είναι ενοχλητική μερικές φορές καθώς αυτό επιβραδύνει την χρήση αλλά είναι προτιμότερο από τα να έχετε μία εσφαλμένα υλοποιημένη κατάτμηση.

Clever size printing

Μορφοποιεί με έξυπνο τρόπο το μήκος της κατάτμησης. Αυτό σημαίνει ότι αντί να τυπώσει στο παράθυρο διαλόγου το μήκος της κατάτμησης σε sectors, θα το τυπώσει στην πρώτη διαθέσιμη μονάδα με ακρίβεια ενός δεκαδικού ψηφίου. Ορισμένοι μπορεί να θεωρούν αυτή τη συμπεριφορά περισσότερο φιλική προς τον χρήστη. Από την άλλη πλευρά αυτή η εκτύπωση του μήκος της κατάτμησης είναι ανακριβείς και έτσι μπορεί να προκαλέσει ανεπιθύμητη συμπεριφορά σε ορισμένες περιπτώσεις (π.χ. Όταν μετακινείτε μία κατάτμηση).

Λειτουργίες

Μπορείτε να δείτε τις εργασίες που μπορείτε να εκτελέσετε στις κατατμήσεις στο κάτω μέρος της οθόνης. Η φιλοσοφία του Partition Surprise είναι να ενημερωνόμαστε για την τρέχουσα κατάσταση των δίσκων. Μπορούμε να τροποποιούμε οι κατατμήσεις και τα συστήματα αρχείων με τις αλλαγές που θέλουμε και στη συνέχεια να εκτελούμε όλες τις αλλαγές που κάναμε. Αυτό έχει το πλεονέκτημα ότι εάν κατά λάθος διαγράψουμε μία κατάτμηση ή κάνουμε κάποια άλλη ανεπιθύμητη ενέργεια μπορούμε να *κλείσουμε* το πρόγραμμα και οι αλλαγές να μην εκτελεστούν. Ενώ κάνετε αναδιάταξη μπορείτε να ελέγξετε παράλληλα την τρέχουσα κατάσταση εάν είναι σωστή (δηλαδή δεν υπάρχουν επικαλυπτόμενες κατατμήσεις και ούτω καθεξής) εκτελώντας την εντολή *verify*.

Μπορείτε επίσης να προβάλετε την αρχική κατάσταση του δίσκου πιέζοντας το πλήκτρο *Tab* ή επιλέγοντας "View original state" από το μενού. Θα εμφανιστεί παράθυρο παρόμοιο με αυτό του κύριου μενού (θα έχει τίτλο "Original state"). Σε αυτό το παράθυρο μπορείτε να περιηγηθείτε στην αρχική κατάσταση των δίσκων σας. Επίσης, μπορείτε να επεξεργαστείτε το δίσκο και τις εισηγμένες παραμέτρους της κατάτμησης (βλέπε **Εισαγωγή παραμέτρων δίσκου** και **Ενημέρωση του Συστήματος Αρχείων**). Μπορείτε να επιστρέψετε στο κεντρικό παράθυρο πιέζοντας το *Tab* ή επιλέγοντας "View planned state" από το μενού.

Αυτήν τη στιγμή οι υποστηριζόμενες λειτουργίες των κατατμήσεων και του συστήματος αρχείων είναι:

Δημιουργία κατάτμησης(New)

Μπορείτε να δημιουργήσετε μία κατάτμηση και ένα σύστημα αρχείων με αυτή την εντολή. Όταν πιέσετε το F7 για να δημιουργήσετε νέα κατάτμηση θα πρέπει πρώτα να βεβαιωθείτε ότι υπάρχει ελεύθερος χώρος στον σκληρό δίσκο (εάν έχετε επιλέξει από την αρχή δημιουργία νέας κατάτμησης). Τότε θα εμφανιστεί ένα παράθυρο στο οποίο μπορείτε να επεξεργαστείτε τον αριθμό της κατάτμησης, το block εκκίνησης της κατάτμησης, το μήκος της κατάτμησης (Σημειώστε ότι μήκος μπορεί να εισαχθεί επίσης σε kilobyte (π.χ. 10 K), megabytes (π.χ.. 20 M) ή gigabyte (π.χ.. 30 G)) και τον τύπο του συστήματος αρχείων της κατάτμησης. Οι τιμές που είναι ήδη καταχωρημένες είναι οι κατάλληλες για τον ελεύθερο χώρο που έχει επιλεχθεί. Αφού επεξεργαστούμε τις τιμές μπορούμε να επιβεβαιώσουμε την δημιουργία πατώντας το F2 (ή με το κουμπί OK) ή να ακυρώσουμε τη δημιουργία με F10 (ή το κουμπί Cancel). Όταν επιβεβαιώσετε την δημιουργία θα ερωτηθείτε για τις τιμές των παραμέτρων του συστήματος αρχείων. Εφ' όσον οι παράμετροι του συστήματος αρχείων επιβεβαιωθούν (F2 ή κουμπί OK) η κατάτμηση δημιουργείται. Επίσης, μπορείτε να ακυρώσετε τη δημιουργία με το πλήκτρο F10 ή με το κουμπί Cancel.

Διαγραφή κατάτμησης(Del)

Μπορείτε να διαγράψετε την κατάτμηση, απλώς, επιλέγοντάς την και πιέζοντας το πλήκτρο F8.

Μετατροπή κατάτμησης(Conv)

Μπορείτε να μετατρέψετε το σύστημα αρχείων της κατάτμησης, επιλέγοντάς την και πατώντας το πλήκτρο F4. Θα ερωτηθείτε σχετικά με τον τύπο του νέου συστήματος αρχείων. Εφ' όσον επιλέξετε τον τύπο συστήματος αρχείων μπορείτε να πατήσετε το κουμπί F2, για να συνεχίσετε. Στη συνέχεια, θα ερωτηθείτε σχετικά με τις παραμέτρους του συστήματος αρχείων. Εάν το νέο σύστημα αρχείων είναι το ίδιο με το παλιό τότε οι τιμές των παραμέτρων του νέου θα είναι ίδιες με το παλιό. Εάν ο νέος τύπος συστήματος αρχείων είναι διαφορετικός θα συμπληρωθούν

προεπιλεγμένες τιμές. Εάν επιβεβαιωθούν οι τιμές των παραμέτρων τότε θα πραγματοποιηθούν οι αλλαγές.

Αντιγραφή κατάτμησης(Copy)

Μπορείτε να αντιγράψετε δεδομένα από κάποια κατάτμηση, επιλέγοντάς την και πιέζοντας το πλήκτρο *F5*. Θα ερωτηθείτε για την συσκευή προορισμού και την κατάτμηση. Εφ' όσον συμπληρώσετε την συσκευή προορισμού και την κατάτμηση και επιβεβαιώσετε τις τιμές, αντιγράφονται όλα τα δεδομένα από την κατάτμηση του δίσκου προέλευσης στον νέο προορισμό.

Μετακίνηση/αλλαγή μεγέθους κατάτμησης(MvRs)

Μπορείτε να μετακινήσετε ή να αλλάξετε το μέγεθος του συστήματος αρχείων, επιλέγοντάς το και πιέζοντας το πλήκτρο *F6*. Θα ερωτηθείτε για την συσκευή προορισμού και την κατάτμηση, την αφετηρία και το μήκος του συστήματος αρχείων. Καθώς οι τρέχουσες τιμές είναι προεπιλεγμένες ,αρκεί μόνο να επιλέξετε το μήκος αν πρόκειται για αλλαγή μήκους ή τη νέα συσκευή , το σύστημα αρχείων αν πρόκειται για μετακίνηση. Με την επιβεβαίωση των νέων τιμών η διαδικασία ξεκινάει.

Καθορισμός γενικών παραμέτρων

Μπορείτε να ορίσετε παραμέτρους κοινές για όλες τις εργασίες από αυτή τη λειτουργία. Αφού πιέσετε το πλήκτρο *F11* το παράθυρο με τις παραμέτρους αυτές θα εμφανιστεί και έτσι μπορείτε να τις επεξεργαστείτε.

Ρύθμιση εξαγμένων παραμέτρων του δίσκου

Αυτή η λειτουργία (εκτελείται με *F12*) σας επιτρέπει να ορίσετε τις εξαγόμενες παραμέτρους του τρέχοντος δίσκου.

Ρύθμιση εισηγμένων παραμέτρων του δίσκου

Αυτή η λειτουργία (εκτελείται με *F12* από το παράθυρο της αρχικής κατάστασης του δίσκου) σας επιτρέπει να ορίσετε τις εισηγμένες παραμέτρους του τρέχοντος δίσκου.

Ρύθμιση των εισηγμένων παραμέτρων του συστήματος αρχείων

Ορισμένα συστήματα αρχείων (π.χ. FAT) έχουν κάποιες τιμές που σίγουρα δεν είναι δυνατόν να εντοπιστούν από το σύστημα αρχείων στο δίσκο (π.χ. κωδικοσελίδα). Οι τιμές αυτές ονομάζονται εισηγμένες παράμετροι. Η λειτουργία αυτή (Μπορείτε να την ενεργοποιήσετε πατώντας το πλήκτρο *F11* στο παράθυρο της αρχικής κατάστασης του δίσκου) σας επιτρέπει να ρυθμίσετε αυτές τις παραμέτρους για το τρέχον σύστημα αρχείων. Διαβάστε την τεκμηρίωση του συστήματος αρχείων στην **Περιγραφή των παραμέτρων του συστήματος αρχείων**.

Άλλες υποστηριζόμενες λειτουργίες είναι:

Quit

Αυτή η ενέργεια (με το πλήκτρο *F10*) θα τερματίσει την χρήση Curses χωρίς να γίνει οποιαδήποτε αλλαγή.

Menu

Πιέζοντας το πλήκτρο *F9* θα εμφανιστεί το μενού με όλες τις πιθανές λειτουργίες.

Client settings

Όταν επιλέξετε αυτήν την καταχώρηση στο μενού, θα εμφανιστεί ένα παράθυρο διαλόγου για την παραμετροποίηση του προγράμματος του χρήστη (βλέπε Ρυθμίσεις Χρήστη).

Commit

Αυτή η λειτουργία (με το πλήκτρο *F2*) θα εγγράψει όλες τις σχεδιαζόμενες αλλαγές στο δίσκο.

Verify

Αυτή η λειτουργία (με το πλήκτρο *F3*) θα ελέγξει εάν η τρέχουσα κατάσταση του δίσκου είναι σωστή.

2.3 Χρήση Gnome

Η χρήση Gnome του Partition Surprise είναι μια γραφική και άνετη διεπιφάνεια χρήστη που χρησιμοποιεί όλες τις λειτουργίες της βιβλιοθήκης. Χρησιμοποιεί *GTK+* και *Gnome* βιβλιοθήκες, και «τρέχει» σε περιβάλλον *X Window System*.

Διάταξη οθόνης

Στην οθόνη του χρήστη αποτυπώνεται ,στο πάνω μισό μέρος και οριζόντια ,η λίστα με τους σκληρούς δίσκους που βρέθηκαν και οριζόντια πάλι μέσα στο καθένα τα συστήματα αρχείων που περιέχουν. Κάθε δίσκος έχει την δική του οριζόντια μπάρα κύλισης (εφ' όσον αυτό είναι δυνατό και εφ' όσον οι πληροφορίες δεν χωρούν στην οθόνη).

Στο κάτω μισό και στην αριστερή πλευρά εμφανίζονται οι πληροφορίες για τις τρέχουσες παραμέτρους της επιλεγμένης κατάτμησης, αυτή η πλευρά είναι αρχικά κενή ώσπου να επιλεγθεί μία κατάτμηση. Στην δεξιά πλευρά υπάρχουν τα κουμπιά για τις γενικές λειτουργίες και ενέργειες με την ενεργή κατάτμηση.

Διαμερίσματα

Κάθε δίσκος περιέχει ένα κατάλογο με τα "*τμήματα*", κάθε *τμήμα* μπορεί να είναι είτε *κατάτμηση* είτε *ελεύθερος χώρος* εφ' όσον υπάρχουν ορισμένα ελεύθερα τμήματα μεταξύ των κατατμήσεων (ή στο τέλος του δίσκου). Ο *ελεύθερος χώρος* εμφανίζεται μόνο στην περίπτωση που είναι αρκετά μεγάλος ώστε να δέχεται τη δημιουργία νέας κατάτμησης, σύμφωνα με τους κανόνες του εκάστοτε πίνακα κατάτμησης του δίσκου.

Οι αλληλεπιδράσεις με τα *τμήματα* γίνονται συνήθως με την χρήση του ποντικιού:

Αριστερό κλικ ποντικιού

Ενεργοποιεί ένα τμήμα. Οι παράμετροι του εμφανίζονται στην κάτω αριστερή πλευρά της οθόνης. Στον *ελεύθερο χώρο* ενεργοποιείται μόνο το κουμπί **Δημιουργία**, διαφορετικά είναι ενεργοποιημένα τα κουμπιά **Διαγραφή**, **Μετακίνηση**, **Αντιγραφή** και **Μετατροπή**.

Σύρσιμο του αριστερού κουμπιού του ποντικιού

Θέτει την κατάτμηση σε κατάσταση “drag' n 'drop” (τέτοια δυνατότητα δεν είναι εφικτή στον *ελεύθερο χώρο*). Στη συνέχεια μπορείτε να την αποθέσετε σε κάποιο *ελεύθερο χώρο* , σύμφωνα με τους κανόνες μετακίνησης και αντιγραφής. Αυτή η ενέργεια θα μετακινήσει την κατάτμηση στον κενό χώρο, εκτός και εάν χρειάζεστε αντίγραφο οπότε και θα πρέπει να κρατάτε πατημένο το πλήκτρο *Control* κατά την διάρκεια της διαδικασίας.

Δεξί κλικ ποντικιού

Θα ανοίξει το μενού context menu για την συγκεκριμένη κατάτμηση ή *ελεύθερο χώρο*.

Το μενού Context περιέχει πολλές επιλογές που μπορούν να εκτελεστούν (με αριστερό κλικ) επιλέγοντας το τμήμα που επιθυμούμε και στην συνέχεια επιλέγοντας το κατάλληλο διαθέσιμο κουμπί. Οι επιλογές αυτές χωρίζονται σε διάφορες κατηγορίες όπως φαίνεται παρακάτω:

Επιλογές ελεύθερου χώρου:

Create

Θα ανοίξει το παράθυρο διαλόγου με την επιλογή των διαθέσιμων συστημάτων αρχείων.

Επιλογές κατάτμησης:

Delete

Θα απελευθερώσετε το χώρο που δεσμεύει η κατάτμηση. Ο ελεύθερος χώρος θα συνενωθεί με τα περιβάλλοντα κουμπιά *ελεύθερου χώρου* , εφ' όσον υπάρχουν . Δεν απαιτείτε καμία επιβεβαίωση της διαγραφής καθώς μπορείτε εύκολα να ανακαλέσετε πίσω την ενέργεια με το κουμπί **Undo**.

Move

Θα ξεκινήσει η μεταφορά της κατάτμησης. Ο δείκτης του ποντικιού θα αλλάξει και το σύστημα θα περιμένει την επιβεβαίωση του χρήστη (με αριστερό κλικ του ποντικιού) σε οποιοδήποτε κενό χώρο. Αυτή η επιλογή έχει την ίδια λειτουργικότητα με αυτή της μεταφοράς της κατάτμησης (**χωρίς** να πιέζουμε το πλήκτρο *Control*).

Copy

Θα ξεκινήσει επίσης σέρνοντας το με το ποντίκι όπως στην επιλογή **Move**, και το σύστημα θα περιμένει την επιβεβαίωση του χρήστη (με αριστερό κλικ του ποντικιού) σε οποιοδήποτε κενό χώρο της κατάτμησης. Το αρχικό διαμέρισμα θα παραμείνει ανέπαφο ακόμα και αν η κατάτμηση προορισμού υποστεί αλλαγές πριν την επιβεβαίωση των αλλαγών στον δίσκο. Αυτή η επιλογή έχει την ίδια λειτουργικότητα με αυτή της μεταφοράς της κατάτμησης **πιέζοντας** το πλήκτρο *Control*.

Convert

Ανοίγει ένα παράθυρο διαλόγου όπου μπορούμε να επιλέξουμε τον τύπο του συστήματος αρχείων που επιθυμούμε. Ο τύπος του παλιού συστήματος αρχείων θα λείπει από τις διαθέσιμες επιλογές για το νέο σύστημα αρχείων.

Επιλογές δίσκου:

Disk parameters

Θα ανοίξει το παράθυρο διαλόγου με τις παραμέτρους που είναι ειδικά για αυτόν το δίσκο. Κάθε δίσκος θα έχει παραμέτρους για εισαγωγή και εξαγωγή. Όταν αλλάζετε τον τύπο του πίνακα των κατατμήσεων σας συνιστάται να **κάντε κλικ** στο κουμπί **Verify** έτσι ώστε να ενημερωθεί η

λίστα των παραμέτρων για τον επιλεγμένο τύπο. Όλες οι παράμετροι εισαγωγής εντοπίζονται συνήθως από το Partition Surprise και δεν μπορείτε να τις τροποποιήσετε σε καμία περίπτωση.

Γενικές επιλογές:

Global parameters

Ανοίγει το παράθυρο διαλόγου με τις γενικές παραμέτρους. Αυτές οι παράμετροι δεν διαχωρίζονται σε εισηγμένες και εξαγμένες καθώς αυτό δεν εξυπηρετεί σε κάτι.

Αρχείο καταγραφής μηνυμάτων

Η βιβλιοθήκη του Partition Surprise εξάγει πολλές πληροφορίες, ορισμένα μηνύματα προειδοποίησης και σε ακραίες περιπτώσεις δημιουργεί μηνύματα λάθους. Συνήθως, αυτά τα μηνύματα αποθηκεύονται σε αρχείο που καθορίζεται από την επιλογή `--surprise-log` αλλά αυτό δεν ισχύει για αυτή την χρήση Gnome η οποία έχει το δικό της σύστημα καταγραφής μηνυμάτων. Μπορούν να διαβαστούν από το κουμπί **Messages** που βρίσκεται στις γενικές λειτουργίες.

Μπορείτε να φιλτράρετε τα μηνύματα ανάλογα με την σημαντικότητά τους για εσάς με την επιλογή "Minimal level for display". Η ρύθμιση αυτή επηρεάζει μόνο την εμφάνιση στην οθόνη.

Μια άλλη επιλογή εμφάνισης είναι η ενεργοποίηση επιπλέον πεδίων στη λίστα των μηνυμάτων με την επιλογή "Display columns". Αυτές οι επιπρόσθετες στήλες είναι απενεργοποιημένες ως προεπιλογή καθώς χρησιμεύουν περισσότερο στον έλεγχο λαθών.

Και τέλος έχετε την επιλογή να αποκρύψετε όλα τα μηνύματα με το κουμπί **Clear** ή να τα αποθηκεύσετε σε αρχείο με το κουμπί **Save**. Το αρχείο θα αποθηκευτεί ως απλό αρχείο κειμένου.

Αναίρεση

Οι ενέργειες των χρηστών ονομάζονται *λειτουργίες*, οι οποίες ομαδοποιούνται σε μία λίστα ιστορικότητας. Κάθε διαγραφή κατάτμησης, αλλαγή μεγέθους, αλλαγή παραμέτρων κ.λπ. μπορεί να αναθεωρηθεί ή να επαναφερθεί. Θα βρείτε τις επιλογές *Undo/Redo* στο μενού *Edit*. Η επιλογή *Undo* μπορεί να βρεθεί και στα κουμπιά γενικών επιλογών. Όλες οι λειτουργίες είναι **γενικές** και σε ακολουθία: Εάν διαγράψετε μία κατάτμηση στον δίσκο 1 και στην συνέχεια δημιουργήσετε ένα καινούργιο στον δίσκο 2 τότε **δεν θα μπορείτε** να αναθεωρήσετε την διαγραφή χωρίς να χάσετε το νέο διαμέρισμα!

Αρχικά, θα έχετε διαθέσιμο μόνο το στοιχείο ιστορικού "*Initial machine state*" που αντιστοιχεί στην αρχική διάταξη του δίσκου. Κάθε λειτουργία θα προσθέτει την δική της εγγραφή στην λίστα ιστορικότητας με μοναδική αλλαγή κάθε φορά τις τιμές των παραμέτρων: Πολλές αλλαγές της ίδιας παραμέτρου (του ίδιου αντικειμένου) θα συλλέγονται και θα ομαδοποιούνται σε μία εγγραφή στην λίστα.

Τροποποίηση παραμέτρων

Η βιβλιοθήκη του Partition Surprise συσχετίζει το σύνολο των αντικειμένων των δοσμένων παραμέτρων : μερικές από αυτές τις παραμέτρους ενδέχεται να περιέχουν blocks για *ext2* συστήματα αρχείων, μια ετικέτα ενός άλλου τόμου κ.λ.π. Αυτές οι παράμετροι μπορούν να συσχετιστούν με ολόκληρους δίσκους, οι κατατμήσεις, τα υποστηριζόμενα συστήματα αρχείων για νέα διαμερίσματα ή μπορεί να είναι επίσης εντελώς γενικές.

Οι παράμετροι μπορούν να *εισαχθούν* για να καθοριστεί η ανάγνωση ενός αντικειμένου (κατάτμηση) ή μπορούν να *εξάγονται* για την τροποποίηση, τη δημιουργία του νέου συστήματος αρχείων (να θυμάστε: κάθε κατάτμηση μπορεί αναγνωστεί/δημιουργηθεί κατά τη διάρκεια της τροποποίησης από το Partition Surprise!). Επιπλέον οι εισηγμένες παράμετροι μπορούν να εντοπιστούν – χωρίς να είναι ενεργές και χωρίς να είναι δυνατόν να τροποποιηθούν από τον χρήστη. Από την άλλη πλευρά οι *εξαγόμενες* παράμετροι μπορούν να καθορίζονται - η τιμή της καθορίζεται κατά την λειτουργία [Verify](#) και μπορείτε μόνο να την αναθεωρήσετε.

Παράμετροι κατάτμησης

Η λίστα των παραμέτρων αυτών εμφανίζεται στο κάτω αριστερό μέρος της οθόνης για την τρέχουσα επιλεγμένη κατάτμηση. Για κανονικές κατατμήσεις εμφανίζεται μια σελίδα για να εισάγουμε παραμέτρους και μια άλλη σελίδα για να τις εξάγουμε. Όταν κάποια κατάτμηση δημιουργείται εκ νέου από εσάς, θα περιέχει μόνο την σελίδα της εξαγωγής – επειδή δεν υπάρχουν δεδομένα για εισαγωγή.

Παράμετροι δίσκου

Αυτές οι παράμετροι εμφανίζονται πάντα σε ξεχωριστό παράθυρο είτε από την "Disk parameters" μενού του partition-context ή με το κουμπί "Parameters". Οι εισαγμένες παράμετροι του δίσκου είναι συνήθως μη-επεξεργάσιμες (=detected).

Παράμετροι συστήματος αρχείων

Κάθε υποστηριζόμενο σύστημα αρχείων έχει καθορίσει το δικό του προεπιλεγμένο σύνολο παραμέτρων. Ακόμα και αν ο δίσκος σας είναι εντελώς κενός μπορείτε να δημιουργήσετε ένα νέο σύστημα αρχείων με τις δικές του παραμέτρους. Αυτές οι παράμετροι έχουν μόνο εξαγωγή τμήμα καθώς δεν υπάρχει νόημα στο να εισάγουμε δεδομένα σε ένα νέο-δημιουργημένο σύστημα αρχείων.

Γενικές παράμετροι

Ορισμένες παράμετροι δεν είναι δυνατό να συσχετιστούν με κανένα από τα αντικείμενα όπως περιγράφεται πιο πάνω. Δεν υπάρχει καμία σκοπιμότητα να εισαχθούν ή να εξαχθούν καθώς τροποποιούν γενικές ρυθμίσεις του συστήματος. Ένα τέτοιο παράδειγμα είναι το "LILO support".

Ολοκλήρωση των αλλαγών στο δίσκο

Όλες οι ενέργειες του χρήστη τροποποιούν μόνο την προσωρινή διάταξη του δίσκου που είναι φορτωμένη στη μνήμη. Δεν γίνονται αλλαγές στον δίσκο (πριν

πατήσουμε το κουμπί **Commit**). Υπάρχει η εξής επιλογή κουμπιών γενικών εντολών:

Verify

Όλες οι τροποποιήσεις του συστήματος αρχείων θα ελεγχθούν, αν είναι πραγματικά δυνατές. Για παράδειγμα τα μεγέθη των αρχείων και οι δομές του βασικού συστήματος ελέγχονται αλλά χωρίς να γίνεται διεξοδικός έλεγχος στην δομή ή έλεγχος για κατεστραμμένα block των μέσων που χρησιμοποιούνται. Αυτή η λειτουργία δεν τροποποιεί τους δίσκους με κάποιο τρόπο.

Commit

Όλες οι αλλαγές πραγματοποιούνται στο φυσικό μέρος του δίσκου. Η ενέργεια **Verify** περιγράφει τις λειτουργίες που γίνονται πριν την τροποποίηση του δίσκου. Ο χρόνος ολοκλήρωσης της ενέργειας συνυπολογίζεται και εκτιμάται μαζί με τις τρέχουσες ενέργειες σε εξέλιξη. Ο χρήστης μπορεί να διακόψει προσωρινά ή να ακυρώσει τις ενέργειες σε εξέλιξη. Το κουμπί *Cancel* απενεργοποιείται προσωρινά, εφ' όσον πατηθεί, καθώς η διαδικασία ακύρωσης έχει ξεκινήσει. Ο χρόνος του τερματισμού της ενέργειας (μαζί με το κλείσιμο των διαλόγων) εξαρτάται από περίπτωση σε περίπτωση. Εάν βρίσκεστε στο τελικό στάδιο της αλλαγής του τελευταίου block τότε το αίτημα του τερματισμού θα αγνοηθεί.

Cancel

Όλες οι αλλαγές στην οθόνη θα παραβλεφθούν (φυσικά η επιλογή **Undo** θα εξακολουθήσει να είναι ενεργή). Οι πληροφορίες που αφορούν τον κάθε δίσκο θα ξανά διαβαστούν και θα αποτυπωθούν στην οθόνη. Η διαδικασία αυτή γίνεται αυτόματα μετά από κάθε τερματισμό της ενέργειας **Commit**.

3. Παράμετροι συστημάτων αρχείων

3.1 Χρήση συστημάτων αρχείων Linux ext2

Στην εφαρμογή του συστήματος αρχείων χρησιμοποιείται η libe2fs για πρόσβαση στα συστήματα αρχείων, έτσι ώστε να δουλεύει σωστά σε όλα τα συστήματα αρχείων.

Είναι δυνατός ο καθορισμός πολλών παραμέτρων για εξαγωγή. Δεν υπάρχει λόγος ανησυχίας για τις σημαντικές παραμέτρους, διότι σχεδόν όλες ανιχνεύονται αυτόματα. Η μόνη εξαίρεση είναι η παράμετρος της κωδικοσελίδας για εισαγόμενα και εξαγόμενα αρχεία

Παράμετροι Linux ext2

block_size

number 1024|2048|4096, import, export.

Ορίζει το μέγεθος των block του συστήματος αρχείων. Κάθε μη κενό αρχείο θα έχει μέγεθος τουλάχιστον όσο ένα block. Αν θέλετε να εφαρμόσετε την παράμετρο αυτή για εξαγόμενο σύστημα αρχείων, δώστε την τιμή εδώ για να ορίσετε την παράμετρο *force_block_size* αλλιώς θα επιλεγεί αυτόματα η βέλτιστη τιμή. Οι επόμενες δύο (2) παράμετροι μπορούν να εφαρμοσθούν ομοίως.

inodes_count

number, import, export.

Ορίζει τον μέγιστο αριθμό inode που μπορούν να δημιουργηθούν στο σύστημα αρχείων. Αυτή η παράμετρος είναι περιττή λόγω της ύπαρξης της παραμέτρου *bytes_per_inode*. Μπορείτε να επιλέξετε και να χρησιμοποιήσετε οποιαδήποτε από τις δύο θέλετε.

bytes_per_inode

number 1024..256*1024, import, export.

Ορίζει την 'πυκνότητα' των inode. Συνήθως η τιμή κυμαίνεται στο διάστημα 4096/8192.

blocks_per_group

number 256..8block_size*, import, export.**

Μπορείτε να εφαρμόσετε την παράμετρο αυτή θέτοντας την τιμή της ως μη μηδενική, αλλά σας συμβουλεύουμε για το αντίθετο.

reserved_percent

number 0..50, import, export.

Ορίζει την ποσότητα χώρου της κατάτμησης, που είναι δεσμευμένη για τον αρχικό κατάλογο, έτσι ώστε να μην επιτρέπεται στους χρήστες να χρησιμοποιήσουν αυτόν τον χώρο.

max_mount_count

number, import, export.

Ορίζει το πόσο συχνά εφαρμόζεται η *e2fsck*. Επίσης η *max_mount_count* ελέγχει την χρήση του συστήματος αρχείων.

check_interval

time in seconds, import, export.

Σχεδόν ίδια λειτουργία με την παράμετρο *max_mount_count*, με τη διαφορά ότι ο χρόνος μετά την εκτέλεση της *fsck* είναι καθορισμένος.

error_handling

continue|read-only|panic, import, export.

Ορίζει τις ενέργειες που πρέπει το λειτουργικό σύστημα να πραγματοποιήσει στα script αρχικοποίησης, όταν το σύστημα αρχείων είναι κατεστραμμένο.

volume_name

string up to 16 characters long, import, export.

Εδώ μπορείτε να καταχωρήσετε το όνομα του τόμου.

codepage

string, import (must be forced), export.

Ορίζει την κωδικοσελίδα που χρησιμοποιείται για ονόματα αρχείων. Πρέπει να αποδοθεί η κατάλληλη τιμή, για παράδειγμα *iso-8859-2*,... Δυστυχώς το *ext2* δεν υποστηρίζει Unicode ονόματα αρχείων σαν προεπιλογή. Είναι πιθανό να τεθεί η κωδικοσελίδα σε UTF-8, αλλά πολύ λίγες εφαρμογές το υποστηρίζουν. Η προεπιλογή είναι ISO-8859-1

force_block_size

boolean, export.

Αν τεθεί τιμή σε αυτή την παράμετρο, τότε δημιουργείται το σύστημα αρχείων με τιμή της *block_size*. Αν δεν είναι δυνατή αυτή η ενέργεια θα τυπωθεί ένα μήνυμα λάθους. Αν δεν τεθεί τιμή θα γίνει προσπάθεια της εφαρμογής της πρώτης ληφθείσας τιμής και έπειτα άλλων πιθανών τιμών.

force_inodes

none|inodes_count|bytes_per_inode, export.

Εδώ ορίζεται ο τρόπος εφαρμογής μέγιστης αρίθμησης των inodes του δημιουργημένου συστήματος αρχείων.

Οι παράμετροι *reserved_percent*, *max_mount_count*, *check_interval*, *error_handling*, *volume_name* μπορούν να υποστούν επεξεργασία χωρίς να ξαναχτιστεί το σύστημα αρχείων. Αν εντοπισθεί ότι μόνο αυτές οι παράμετροι (συν τις παραμέτρους υπηρεσίας *force_**) αλλάξουν, θα διαφοροποιηθεί μόνο το *superblock*.

Προειδοποιήσεις παραμέτρων Ext2

Μην ξεχνάτε να ορίσετε την παράμετρο της *κωδικοσελίδας* στη σωστή τιμή αλλιώς τα ονόματα των αρχείων σας ,τα οποία περιέχουν εθνικούς χαρακτήρες, θα καταστραφούν.

3.2 Χρήση συστήματος αρχείων FAT

Η εφαρμογή συστήματος αρχείων υποστηρίζει 12/16/32 bit FAT μεταβλητές. Υποστηρίζει short 8+3 MSDOS ονόματα, long VFAT ονόματα και long VFAT ονόματα, συν επιπρόσθετες αποθηκευμένες ιδιότητες των inode (λειτουργίες αρχείων, δικαιώματα,...).

Μπορείτε να ορίσετε πολλές παραμέτρους για εισαγωγή και εξαγωγή. Το κοινό FAT δεν υποστηρίζει ιδιοκτησία αρχείων και ιδιότητες αρχείων Unix , έτσι μπορείτε να τις θέσετε εσείς κατά την εισαγωγή. Η κωδικοσελίδα που χρησιμοποιείται για αποθήκευση των ονομάτων αρχείων, δεν ορίζεται από το σύστημα αρχείων, έτσι πρέπει να ορισθεί χειροκίνητα. Μπορείτε να εφαρμόσετε το *cluster_size* και *fat_bits* για την εξαγωγή, αν θέλετε. Αν δεν εντοπισθούν τα *fat_bits* σωστά, μπορείτε να τα εφαρμόσετε επίσης για εισαγόμενα συστήματα αρχείων.

Παράμετροι FAT

cluster_size

number 512|1024|2048|4096|8192|16384|32768|65536, import, export.

Ορίζει το μέγεθος του cluster του συστήματος αρχείων. Κάθε μη κενός φάκελος θα πρέπει να έχει μέγεθος τουλάχιστον αυτό του ενός cluster. Αυτή η παράμετρος είναι 100% ανιχνεύσιμη από εισαγόμενα συστήματα αρχείων. Αν θέλετε να εφαρμόσετε αυτή την παράμετρο για εξαγόμενα συστήματα αρχείων, ορίστε αυτή την τιμή και την παράμετρο *force_cluster_size* , αλλιώς θα επιλεγθεί η βέλτιστη τιμή. Η επόμενη παράμετρος *fat_bits* μπορεί να εφαρμοσθεί ομοίως.

fat_bits

12|16|32, import (can be forced), export.

Ορίζει τον τύπο του FAT. Μπορεί να εφαρμοσθεί για εξαγόμενα συστήματα αρχείων καθώς και για *cluster_size*.

Πολλές φορές δεν είναι ξεκάθαρο αν ο εισαγόμενος τύπος FAT είναι 12 ή 16. Αν έχετε τη γνώση, μπορείτε να προκαθορίσετε την *fat_bits* τιμή για τα εισαγόμενα συστήματα αρχείων.

fats

number 1..8, import, export.

Ορίζει τον αριθμό αντιγράφων του FAT. Μπορείτε να εφαρμόσετε την τιμή για εξαγόμενα συστήματα αρχείων. Η συνήθης τιμή είναι 2.

rootdir_entries

number 16..65520, import, export.

Ο χώρος που διατίθεται για τις καταχωρίσεις του αρχικού καταλόγου. Αν χρησιμοποιείτε VFAT, κάθε όνομα αρχείου θα μπορούσε να έχει μήκος μίας καταχώρησης! Μη χρησιμοποιείτε πολύ μικρές τιμές, η προεπιλεγμένη τιμή είναι 512. Το FAT32 δεν γνωρίζει αυτήν την παράμετρο, επειδή ο αρχικός κατάλογος είναι αποθηκευμένος σαν απλό αρχείο.

codepage

string, import (must be forced), export.

Ορίζει την χρήση της κωδικοσελίδας που χρησιμοποιείται για μικρά ονόματα αρχείων. Τα μεγάλα ονόματα αρχείων είναι πάντα αποθηκευμένα σε Unicode UCS2, έτσι δεν χρειάζεται να ανησυχείτε αν χρησιμοποιείτε VFAT. Η προεπιλεγμένη τιμή είναι IBM437.

uid

number, import (must be forced).

Ορίζει το uid για κάθε αρχείο στο σύστημα αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε σε αυτό, ή να χρησιμοποιήσετε την προεπιλεγμένη τιμή 0 (root).

gid

number, import (must be forced).

Ορίζει το gid για κάθε αρχείο στο σύστημα αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε σε αυτό ή να χρησιμοποιήσετε την προεπιλεγμένη τιμή 0 (root).

mask

number, import (must be forced).

Ορίζει την μάσκα για κάθε αρχείο στο σύστημα αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε σε αυτό, ή να χρησιμοποιήσετε την προεπιλεγμένη τιμή 0644 σε οκταδικό (rw-r--r--).

system_id

string up to 8 characters long, import, export.

Μία από τις τρεις (3) συμβολοσειρές που είναι αποθηκευμένες στο superblock του συστήματος αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε εκεί.

volume_name

string up to 11 characters long, import, export.

Μία από τις τρεις (3) συμβολοσειρές που είναι αποθηκευμένες στο superblock του συστήματος αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε εκεί.

fat_name

string up to 8 characters long, import, export. Μία από τις τρεις (3) συμβολοσειρές που είναι αποθηκευμένες στο superblock του συστήματος αρχείων. Μπορείτε να αποθηκεύσετε οτιδήποτε εκεί.

long_name_type

none|vfat|vfat+rights, import (can be forced), export.

Ορίζει αν θα χρησιμοποιηθούν RAW msdos 8+3 ονόματα ή vfat long ονόματα αρχείων. Όλα τα εξαγόμενα ονόματα είναι μοναδικά ταξινομημένα στο MSDOS 8+3 dir slot, long ονόματα γράφονται κατόπιν επιλογής.

Εάν έχει επιλεγθεί το **vfat+rights** τότε επιπρόσθετες inode ιδιότητες είναι επίσης αναγνώσιμες/εγγράψιμες (δυστυχώς δεν υποστηρίζει hardlinks). Αυτή η επιλογή χρησιμοποιείται για ασφάλεια, για να προλάβει απώλεια μεταδεδομένων, αν αποφασίσετε να μετατρέψετε ext2 σε FAT και αντίστροφα. Το **vfat+rights** σύστημα ΔΕΝ είναι συμβατό με άλλα συστήματα (όπως UMSDOS) και χρησιμοποιείται μόνο για ασφάλεια.

numtail

boolean, export.

Ορίζει, μικρά ονόματα αρχείων που ανταποκρίνονται σε μεγάλα ονόματα αρχείων, να έχουν πάντα νουμερική ουρά. Εάν δεν οριστεί ,ορίζεται ως όνομα αρχείου ένα χωρίς νουμερική ουρά.

force_cluster_size

boolean, εξαγωγή.

Εάν ρυθμίσετε αυτήν την παράμετρο, το σύστημα αρχείων υποχρεώνεται να δημιουργηθεί με την τιμή της *cluster_size*. Αν το surprise δεν μπορεί να δημιουργήσει το σύστημα αρχείων, θα τυπωθεί ένα μήνυμα λάθους. Αν δεν τεθεί τιμή θα γίνει προσπάθεια της εφαρμογής της πρώτης ληφθείσας τιμής και έπειτα άλλων πιθανών τιμών.

force_fat_bits

boolean, εισαγωγή, εξαγωγή.

Μπορείτε να επιβάλετε εδώ την τιμή των *fat_bits* -- για εξαγόμενα και εισαγόμενα συστήματα αρχείων.

save_bootsector

boolean, εισαγωγή.

Εάν οριστεί, το πρόγραμμα εκκίνησης του λειτουργικού συστήματος θα αποθηκευτεί πριν από την μετατροπή και θα γίνει επαναφορά του μετά. Εφαρμόζεται μόνο όταν το FAT μετατρέπεται σε FAT και όχι από μη-FAT32 σε FAT32 (λόγω της διαφορετικής διάρθρωσης του superblock).

Οι παράμετροι *system_id*, *volume_name*, *fat_name* μπορούν να τροποποιηθούν χωρίς την ανοικοδόμηση ολόκληρου του συστήματος αρχείων. Αν το surprise εντοπίσει πως μόνο αυτές οι παράμετροι έχουν αλλάξει (συν τις παραμέτρους *uid*, *gid*, *mask*, *numtail*, *force_**), απλά θα τροποποιήσει το superblock.

Παράμετροι προειδοποίησης FAT

Εάν μετατρέψετε οποιοδήποτε σύστημα αρχείων Unix σε FAT, σίγουρα θα χάσετε πολλές μετά-πληροφορίες σχετικά με αρχεία, αν δεν χρησιμοποιήσετε *long_name_type=vfat+rights* (δεν είναι προεπιλεγμένο). Τα απλά FAT/VFAT δεν υποστηρίζουν δικαιώματα αρχείων, ιδιοκτησίας, ειδικό κατάλογο καταχωρήσεων (*deviced*, *symlinks*, *hardlinks*, *fifos*,...).

Ρυθμίζοντας **vfat+rights** δεν είναι αρκετό για να αποθηκεύσετε όλα σας τα μετά-δεδομένα, επειδή :

1. Το surprise δεν υποστηρίζει *hardlinks* σε **vfat+rights** ακόμα.
2. Το VFAT δεν σας επιτρέπει να δημιουργήσετε ονόματα αρχείων που διαφέρουν σε μικρούς και μεγάλους χαρακτήρες (*case sensitive*), έτσι λοιπόν τα ονόματα αυτά των αρχείων στο ext2 (όπως είναι το ``hello'` και το

`Hello') ΠΡΕΠΕΙ να μετονομαστούν σε όλες τις περιπτώσεις. Το surprise προσαρτά ένα τυχαίο επίθεμα στο δεύτερο όνομα αρχείου.

Όπως καταλαβαίνετε, κάθε σύστημα αρχείων που μετατρέπεται σε VFAT θα χάσει ορισμένα μετά-δεδομένα. Μην μετατρέπετε σε VFAT εάν δεν γνωρίζετε τι κάνετε.

Τα ονόματα αρχείων που περιέχουν χαρακτήρες ,συνήθως αποθηκεύονται σε longnames. Αν συμβαίνει αυτό, δεν χρειάζεται να ανησυχείτε για αυτά γιατί σίγουρα θα μετατραπούν σωστά. Αν όχι, μπορείτε να τα αποθηκεύσετε διορθώνοντας τις ρυθμίσεις των παραμέτρων της κωδικοσελίδας – εφ' όσον γνωρίζετε, ποια κωδικοποίηση χρησιμοποιεί το λειτουργικό σύστημα για αποθήκευση. Η προεπιλεγμένη τιμή για το IBM437 είναι σίγουρα κακή επιλογή. Μπορείτε να χρησιμοποιήσετε την κωδικοποίηση CP852 (PC-Latin 2) για χαρακτήρες Ανατολικής Ευρώπης. Να είστε προσεκτικοί στον ορισμό των παραμέτρων της κωδικοσελίδας.

3.3 Διαχείριση χρήσης Linux swap

Όλες οι παράμετροι του swap είναι πολύ σημαντικές. Γενικά η χρήση λάθος παραμέτρων μπορεί να προκαλέσει μη εφικτών και «άχρηστων» swap κατατμήσεων. Ευτυχώς εάν εκτελείτε το Partition Surprise στο μηχάνημα (πλατφόρμα) όπου θέλετε να χρησιμοποιήσετε και την κατάτμηση swap, όλες οι σημαντικές παράμετροι εντοπίζονται σωστά αυτόματα.

Παράμετροι swap του Linux

header

Έκδοση 1 εάν είναι δυνατόν| Έκδοση 1| Έκδοση 2, εισαγωγή, εξαγωγή. Επιλέγει την έκδοση της κεφαλίδας που χρησιμοποιείται για το swap σύστημα αρχείων. Το Partition Surprise θα επιλέξει την κεφαλίδα έκδοση 1 για τις κατατμήσεις που ικανοποιούν το μέγιστο όριο κατάτμησης της έκδοσης 1, Αλλιώς θα επιλεγθεί η επικεφαλίδα **έκδοση 2**.

endian

Μικρά endian|Μεγάλα endian, εισαγωγή, εξαγωγή.

Αυτή η παράμετρος εξαρτάται καθαρά από την πλατφόρμα. Πρέπει να επιλεγεί ο τύπος του endian του μηχανήματος το οποίο τελικά θα χρησιμοποιηθεί στην κατάτμηση swar. Ως προεπιλογή επιλέγεται από το Partition Surprise ο εκάστοτε τύπος endian του τρέχοντος μηχανήματος.

pagesize

αριθμός 4096|8192, εισαγωγή, εξαγωγή .

Αυτή η παράμετρος εξαρτάται καθαρά από την πλατφόρμα. Πρέπει να επιλέξετε το μέγεθος της σελίδας του μηχανήματος το οποίο τελικά θα χρησιμοποιεί την κατάτμηση swar. Ως προεπιλογή επιλέγεται από το Partition Surprise το εκάστοτε μέγεθος της σελίδας του τρέχοντος μηχανήματος.

Προειδοποίηση της παραμέτρου της κεφαλίδας στην έκδοση SWAP

Το κύριο πρόβλημα της κατάτμησης swar είναι ότι η κεφαλίδα **έκδοση 1** μπορεί να χρησιμοποιηθεί μόνο για τις κατατμήσεις με μέγιστο μέγεθος περίπου 128 MB (512 MB για πλατφόρμες με μέγεθος σελίδα 8 KB). Κατατμήσεις με κεφαλίδα **έκδοση 2** δεν διαθέτουν αυτό το όριο του μέγιστου μεγέθους αλλά από την άλλη πλευρά υποστηρίζεται μόνο από εκδόσεις πυρήνα Linux από 2.1.117 και άνω. Το Partition Surprise θα επιλέξει την κεφαλίδα **έκδοση 1** για διαμέρισμα με μέγεθος κάτω από αυτό της **έκδοσης 1**, αλλιώς θα επιλέξει την **έκδοση 2**. Παρακαλώ σημειώστε ότι έχει διαπιστωθεί πως η κεφαλίδα **με έκδοση 2** δεν θα είναι συμβατή με παλαιότερες εκδόσεις του πυρήνα Linux, και θα δίνεται προειδοποίηση από το Partition Surprise μόνο όταν η τρέχουσα έκδοσή του πυρήνα σας είναι ακατάλληλη. Η προειδοποίηση αυτή μπορεί να μην είναι τόσο καλή εάν ο σκληρός δίσκος μεταφερθεί σε άλλο υπολογιστή ή γίνει φόρτωση με διαφορετική έκδοση του kernel (πιο παλιά).

ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΟΥΜΕΝΩΝ ΒΗΜΑΤΩΝ ΜΕΤΑΤΡΟΠΗΣ

Βήμα 1, Αρχικό Βήμα

Για να πραγματοποιήσουμε την μετατροπή του συστήματος αρχείων από FAT σε ext2, θα πρέπει πρώτα να εξάγουμε απαραίτητες πληροφορίες.

Πρέπει να εξαχθούν από τον τόμο ή δίσκο υπό μετατροπή, όλες οι πληροφορίες σχετικά με αυτόν. Αυτές είναι:

- Περιγραφή του φυσικού μέσου (μέγεθος, γεωμετρία)
- Περιγραφή του πίνακα της κατάτμησης (είδος, εκτεταμένες πληροφορίες)
- Περιγραφή κάθε λογικής κατάτμησης (partition, μέγεθος, είδος)
- Περιγραφή του Filesystem

Θα χρειαστούμε πληροφορίες του φυσικού μέσου για την κατάτμηση ώστε να ξέρουμε πώς να κατευθυνθούμε, πληροφορίες σχετικά με τον πίνακα της κατάτμησης του τόμου για να ξέρουμε το μέγεθος και την οργάνωση των αρχείων στο φυσικό μέσο καθώς και πληροφορίες σχετικά με το Filesystem της κατάτμησης έτσι ώστε να μπορούμε να προσπελάσουμε τα δεδομένα.

Φυσικά οι παράμετροι αυτές θα ζητούν την επιβεβαίωση του χρήστη, ή θα μπορούν να τροποποιηθούν αν θεωρηθούν λανθασμένες.

Βήμα 2

Εφ' όσον διαβάσαμε το FAT, έχουμε όλες τις απαραίτητες πληροφορίες που χρειαζόμαστε σχετικά με την δομή του δίσκου (track, cluster, sector) και όλες τις πληροφορίες σχετικά με την τοποθεσία των αρχείων και του κενού χώρου.

Αφού έχει γίνει επιβεβαίωση του συστήματος αρχείων μέσω της ανάγνωσης του σχετικού πεδίου στον λογικό δίσκο, μία ρουτίνα γνωρίζοντας πλέον που αρχίζουν τα fat_bits για το FAT (από πού δηλαδή ξεκινούν τα δεδομένα, αφού εκεί

τελειώνουν οι πληροφορίες για το Σ.Α) αρχίζει να επιβεβαιώνει τα δεδομένα αυτά και καλείται το κυρίως σώμα του μετατροπέα.

Βήμα 3, Περιγραφή του Μετατροπέα

Ο μετατροπέας θα ακολουθεί τα εξής βήματα:

Έλεγχος του δίσκου για τυχόν κατεστραμμένα blocks. Στην συνέχεια εύρεση και επιβεβαίωση ελεύθερου χώρου στον δίσκο ή την κατάτμηση. Εάν δεν υπάρχει επαρκής χώρος, τότε η διαδικασία δεν μπορεί να ξεκινήσει. Πρέπει να σημειώσω πως το σημείο αυτό είναι πολύ σημαντικό καθώς το νέο σύστημα αρχείων δημιουργείτε στον κενό χώρο χωρίς να επηρεάζει το παλιό. Έτσι εάν κάτι δεν πάει καλά μέχρι και την ολοκλήρωση αυτού του βήματος, δεν θα δημιουργηθεί κανένα απολύτως πρόβλημα στο παλιό σύστημα αρχείων και στα δεδομένα.

Πραγματοποιείτε δημιουργία του ext2 στον ελεύθερο χώρο του δίσκου ή της κατάτμησης όπως περιγράφεται παρακάτω :

- Προετοιμάζει την δομή των δεδομένων και του superblock.
- Προαιρετικά, ελέγχει εάν η κατάτμηση περιέχει ελαττωματικά block. Εάν ναι, δημιουργεί μια λίστα με τα ελαττωματικά block.
- Για κάθε ομάδα block στον δίσκο, απαιτείται η αποθήκευση του superblock, του πίνακα inode, της περιγραφής της δομής, και δύο (2) εικόνων τους.
- Προετοιμάζει το εικονικό inode και τον εικονικό χάρτη δεδομένων κάθε ομάδας block μέχρι το 0.
- Προετοιμάζει τον πίνακα inode κάθε ομάδας block.
- Δημιουργεί τον ριζικό κατάλογο.
- Δημιουργεί τον κατάλογο **lost+found**, που χρησιμοποιείτε από την **e2fsck** για να συνδέσει τα χαμένα και ανακτημένα ελαττωματικά block.
- Ενημερώνει την εικόνα του inode και την εικόνα του block των αρχείων της ομάδας μπλοκ που έχουν δημιουργηθεί οι δύο προηγούμενοι κατάλογοι.
- Ομαδοποιεί τα ελαττωματικά block (εάν χρειαστεί) στον κατάλογο **lots+found**.

Οι τιμές των παραμέτρων που θα χρησιμοποιηθούν για την κατασκευή ανιχνεύονται αυτόματα έχοντας υπόψη τα στοιχεία από το βήμα 1. Σε κάθε περίπτωση ο χρήστης μπορεί να αλλάξει τις τιμές αυτές με δικιά του ευθύνη. Πρέπει επίσης να δηλωθεί η ύπαρξη του ext2 στον δίσκο στο υπάρχον Σ.Α. δηλαδή στο FAT, μέσω της δήλωσης των συγκεκριμένων τομέων του δίσκου ως κατειλημμένων και μόνο για ανάγνωση.

Δεν υπάρχει ανάγκη μεταφοράς των δεδομένων, δηλαδή copy-paste, καθώς όποια αντιγραφή γίνεται εικονικά.

Αρκεί να μεταφερθεί ο δείκτης σε αυτά τα δεδομένα στο καινούριο Σ.Α. δηλαδή στο ext2. Αν αυτό είναι επιτυχές, τότε προχωράει στα επόμενα δεδομένα. Αν όχι (γιατί π.χ. τα δύο Σ.Α. αντιμετωπίζουν διαφορετικά ένα αρχείο και πρέπει να αποθηκευτεί σε διαφορετικού μεγέθους ή πλήθους blocks) τότε το αρχείο πρέπει να μεταφερθεί (copy –paste).

Η διαδικασία έχει ως εξής:

- Προετοιμασία του παλιού/νέου συστήματος αρχείων για ανάγνωση/εγγραφή.
- Ανάγνωση παλιού συστήματος αρχείων.
 - Σε κάθε κατάλογο γίνεται ανάγνωση της κάθε καταχώρησης.
 - Εάν δεν υπάρχει άλλη καταχώρηση τότε προχωράμε στον επόμενο κατάλογο.
 - Εάν υπάρχει καταχώρηση τότε πρέπει να καταχωρηθεί στο νέο σύστημα αρχείων.
 - Εάν είναι κατάλογος τότε πρέπει να δημιουργηθεί ο αντίστοιχος κατάλογος στο νέο σύστημα αρχείων.
 - Εάν είναι αρχείο πρέπει να γίνει η αντιστοίχιση της θέσης του από το παλιό στο νέο.

Στο τέλος της διεργασίας μεταφοράς/μετατροπής πρέπει να λάβει μέρος και το ουσιαστικότερο κομμάτι:

Βήμα 4, Αλλαγή του Σ.Α

Πρέπει δηλαδή να γραφτούν στο αρχικό κομμάτι της κατάτμησης τα στοιχεία του ext2 και να διαγραφούν τα αντίστοιχα του FAT, ώστε η κατάτμηση (ή ο δίσκος) να γίνεται δεκτή ως «νόμιμη», ότι περιέχει δηλαδή ext2 σύστημα αρχείων. Δυστυχώς το κομμάτι αυτό είναι και το πιο κρίσιμο και δεν πρέπει σε καμία περίπτωση να διακοπεί καθώς μπορεί να προκαλέσει απώλεια των δεδομένων σας.

Οι διεργασίες που θα διαβάζουν τα δεδομένα που υπάρχουν στον δίσκο, δεν πρέπει να ενδιαφέρονται για τα ίδια τα δεδομένα. Η λογική (φυσική) θέση των δεδομένων αυτών, είναι αρκετή, δηλαδή η θέση και το μέγεθος του αρχείου (για αυτό και καλό θα είναι ΠΡΙΝ την μετατροπή να γίνει πλήρης κατακερματισμός (defragment) του τόμου ώστε όλα τα αρχεία να είναι σε μία συνεχή θέση στον τόμο).

Ουσιαστικά κάθε φορά που η ρουτίνα διαβάζει ένα καινούριο αρχείο, αφού βρει την θέση και το μέγεθος, δημιουργεί ένα inode με αυτά τα δεδομένα και αποθηκεύει αυτό το inode.

ΚΟΜΜΑΤΙΑ ΚΩΔΙΚΑ

ΠΛΗΡΟΦΟΡΙΕΣ ΓΕΩΜΕΤΡΙΑΣ ΚΑΙ ΜΕΓΕΘΟΥΣ ΔΙΣΚΟΥ

```
#include "stdafx.h"
#include <windows.h>
#include <winioctl.h>
#include <stdio.h>

BOOL GetDriveGeometry(DISK_GEOMETRY *pdg, LPCWSTR device)
{
    HANDLE hDevice;           // handle to the drive to be examined
    BOOL bResult;            // results flag
    DWORD junk;              // discard results

    hDevice = CreateFile(device, // drive to open
                        0,       // no access to the drive
                        FILE_SHARE_READ | // share mode
                        FILE_SHARE_WRITE,
                        NULL,     // default security attributes
                        OPEN_EXISTING, // disposition
                        0,       // file attributes
                        NULL);    // do not copy file attributes

    if (hDevice == INVALID_HANDLE_VALUE) // cannot open the drive
    {
        printf("CreateFile() failed!\n");
        return (FALSE);
    }

    bResult = DeviceIoControl(hDevice, // device to be queried
                              IOCTL_DISK_GET_DRIVE_GEOMETRY, // operation to perform
                              NULL, 0, // no input buffer
                              pdg, sizeof(*pdg), // output buffer
```

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

```
        &junk,                // # bytes returned
        (LPOVERLAPPED) NULL); // synchronous I/O

    CloseHandle(hDevice);

    return (bResult);
}

int main(int argc, char *argv[])
{
    DISK_GEOMETRY pdg;           // disk drive geometry structure
    BOOL bResult;               // generic results flag
    ULONGLONG DiskSize;        // size of the drive, in bytes

    bResult = GetDriveGeometry (&pdg, TEXT("\\\\.\\F:"));

    if (bResult)
    {
        printf("Cylinders = %I64d\n", pdg.Cylinders);
        printf("Tracks/cylinder = %ld\n", (ULONG) pdg.TracksPerCylinder);
        printf("Sectors/track = %ld\n", (ULONG) pdg.SectorsPerTrack);
        printf("Bytes/sector = %ld\n", (ULONG) pdg.BytesPerSector);

        DiskSize = pdg.Cylinders.QuadPart * (ULONG)pdg.TracksPerCylinder
        * (ULONG)pdg.SectorsPerTrack * (ULONG)pdg.BytesPerSector;

        printf("Disk size = %I64d (Bytes) = %I64d (Gb)\n", DiskSize, DiskSize / (1024
        * 1024 * 1024));
    }
    else
    {
        printf ("GetDriveGeometry() failed. Error %ld.\n", GetLastError ());
    }
    system("Pause");
    return ((int)bResult);
}
```


ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΙΝΑΚΑ ΚΑΤΑΤΜΗΣΗΣ

```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <dos.h>
#include <bios.h>

struct boot
{
    unsigned char jump[3] ;
    char OEMname[8] ;
    shortint bps ;
    unsigned char spc ;
    shortint reservedsec ;
    unsigned char fatcopies ;
    shortint maxdirentries ;
    shortint totalsec ;
    unsigned char mediadesc ;
    shortint secperfat ;
    shortint secpertrack ;
    shortint noofsides ;
    longint hidden ;
    longint hugesec ;
    unsigned char drivenumbr ;
    unsigned char reserved ;
    unsigned char bootsignature ;
    longint volumeid ;
    char volumelabel[11] ;
    char filesystype[8] ;
    unsigned char unused[450] ;
};
```

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

```
struct boot bs ;
char filetypestr[8] ;

void getfat_12 ( unsigned char * ) ;
void read_fat_info ( long ) ;
void fat_info( ) ;

void main( )
{
    char choice ;

    clrscr( ) ;

    printf ( "F. Drive F" ) ;
    //printf ( "\nC. Drive C" ) ;
    printf ( "\n0. Exit" ) ;

    printf ( "\nEnter the drive (F): " ) ;
    scanf ( "%c", &choice ) ;

    if ( absread ( choice - 65, 1, 0, &bs ) == -1 )
    {
        printf ( "Error reading sector" ) ;
        exit ( 0 ) ;
    }
    else
    {
        strcpy ( filetypestr, bs.filesystype ) ;
        filetypestr[6] = '\0' ;
    }

    fat_info( ) ;
}

void getfat_12 ( unsigned char *pfat )
```

```
{
    intvalue ;
    int *fatentry ;
    int i, k ;

    for ( k = 2 ; k < 18 ; k++ )
    {
        i = k * 3 / 2 ;

        fatentry = ( int* ) ( pfat + i ) ;

        if ( ( k % 2 ) == 0 )
            value = ( *fatentry & 0x0fff ) ;
        elsevalue = ( *fatentry >> 4 ) ;

        printf ( "%03x  ", value ) ;
        if ( k % 9 == 0 )
            printf ( "\n" ) ;
    }
}

void read_fat_info ( long fat_num )
{
    int j, i ;

    unsigned char *p ;

    if ( strcmp ( "FAT12", filetypestr, 5 ) == 0 )
    {
        p = ( unsigned char* ) malloc ( bs.bps ) ;
        absread ( 0, 1, fat_num, p ) ;
        getfat_12( p ) ;
    }

    if ( strcmp ( "FAT16", filetypestr, 5 ) == 0 )
```

```
{
    shortint *pfat ;
    p = ( unsigned char* ) malloc ( bs.bps ) ;
    absread ( 2, 1, fat_num, p ) ;
    pfat = ( shortint* ) p ;

    for ( j = 0 ; j < 2 ; j++ )
    {
        printf ( "\n%d  ", j * 8 ) ;
        for ( i = 0 ; i < 8 ; i++ )
        {
            printf ( "%04x  ", *pfat++ ) ;
        }
    }
}

void fat_info( )
{
    longint first_fat, second_fat ;

    first_fat = bs.reservedsec ;
    second_fat = bs.reservedsec + bs.secperfat ;

    printf ( "\n%s Fat Information", filetypestr ) ;
    printf ( "\n-----" ) ;

    printf ( "\nFirst FAT Information\n" ) ;

    read_fat_info ( first_fat ) ;

    printf ( "\n\nSecond FAT Information\n" ) ;

    read_fat_info ( second_fat ) ;

    printf ( "\n-----\n" ) ;
}
```



ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ ΚΑΙ ΤΑ ΑΡΧΕΙΑ

```
using namespace System;
using namespace System::IO;

namespace ConsoleApplication2
{
    public ref class Program
    {
    public:
        static void Main()
        {
            // Loop through all the immediate subdirectories of F.
            for each (String^ entry in Directory::GetDirectories("F:\\"))
            {
                DisplayFileSystemInfoAttributes(gcnew DirectoryInfo(entry));
            }
            // Loop through all the files in F.
            for each (String^ entry in Directory::GetFiles("F:\\"))
            {
                DisplayFileSystemInfoAttributes(gcnew FileInfo(entry));
            }
        }
        static void DisplayFileSystemInfoAttributes(FileSystemInfo^ fsi)
        {
            // Assume that this entry is a file.
            String^ entryType = "Αρχείο";

            // Determine if entry is really a directory
            if ((fsi->Attributes & FileAttributes::Directory) ==
                FileAttributes::Directory)
            {
                entryType = "Φάκελος";
            }
            // Show this entry's type, name, and creation date.
            Console::WriteLine(fsi->FullName);
            Console::WriteLine("Τύπος : {0} ",entryType);
            Console::WriteLine("Δημιουργήθηκε στις {0:D}",fsi->CreationTime);
            Console::WriteLine("Τελευταία τροποποίηση στις {0:D}",fsi->
                LastAccessTime);
            Console::WriteLine("Κατάληξη αρχείου {0}",fsi-> Extension);
            Console::WriteLine("Κατάληξη αρχείου {0}",fsi-> Extension);
        }
    };
};

int main()
{
    ConsoleApplication2::Program::Main();
}
```

ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΚΑΘΕ ΛΟΓΙΚΟ ΤΟΜΟ

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace ConsoleApplication1
{
    class Program
    {
        static void getInfo()
        {
            ArrayList myList = new ArrayList();
            DriveInfo[] allDrives = DriveInfo.GetDrives();

            foreach (DriveInfo dirInfo in allDrives)
            {
                if (dirInfo.IsReady == true &&
                    (Convert.ToString(dirInfo.DriveType) == "Fixed" ||
                     Convert.ToString(dirInfo.DriveType) == "Removable") &&
                    Convert.ToString(dirInfo.DriveFormat) == "FAT32")
                {
                    myList.Add(dirInfo.Name);
                }
            }

            Console.Clear();

            Console.WriteLine("Παρακαλώ επιλέξτε μία από τις παρακάτω κατατιμήσεις  
FAT32 :");
            Console.WriteLine();

            int i = 0;
```

Πτυχιακή εργασία του φοιτητή Κλάρου Γεώργιου

```
        if (myList.Count > 0)
        {
            foreach (string p in myList)
            {
                i++;
                //Console.Write("{0} ", i);

                dr = Convert.ToString(myList[Convert.ToInt32(Console.Write("{0}",
p))]);
                //Console.WriteLine(dr);

                DriveInfo di = new DriveInfo(dr);
                Console.WriteLine("//////////////////////////////////////ΠΛΗΡΟΦΟΡΙΕΣ
ΚΑΤΑΤΜΗΣΗΣ////////////////////////////////////");
                //Console.WriteLine("DRIVE INFO");
                Console.WriteLine("Μέγεθος: " + di.TotalSize + " bytes");
                Console.WriteLine("Διαθέσιμος      Ελεύθερος      Χώρος:      " +
di.AvailableFreeSpace + " bytes");
                Console.WriteLine("Συνολικός      Ελεύθερος      Χώρος:      " +
di.TotalFreeSpace + " bytes");
                Console.WriteLine("Σύστημα Αρχείων: " + di.DriveFormat);
                Console.WriteLine("Τύπος Δίσκου: " + di.DriveType);
                Console.WriteLine("Προσβάσιμος: " + di.IsReady.ToString());
                Console.WriteLine("Όνομα: " + di.Name);
                Console.WriteLine("Πιζική Διαδρομή: " + di.RootDirectory);
                Console.WriteLine("Ετικέτα Τόμου: " + di.VolumeLabel.ToString(),
di.Name);
                //Console.WriteLine("Όνομα: " + di.Name);
                //Console.WriteLine("DRIVE INFO");

                Console.WriteLine("//////////////////////////////////////
////////////////////////////////////");
            }
        }
        static void Main(string[] args)
        {
            Console.WriteLine("Γίνεται αναζήτηση των κατατμήσεων. Παρακαλώ
περιμένετε...");
```



```
        getInfo();  
    }  
}
```

ΑΝΑΓΝΩΣΗ ΔΕΔΟΜΕΝΩΝ ΕΝΟΣ Superblock

```
1.
   #include<linux/ext2_fs.h>
2.   #include<sys/types.h>
3.   #include<sys/stat.h>
4.   #include<stdio.h>
5.   #include<unistd.h>
6.   #include<fcntl.h>
7.   #include<stdlib.h>
8.   #include<string.h>
9.
10.  #define boot_block_size 1024
11.
12.  int main()
13.  {
14.      char *buff = (char *)malloc(sizeof(struct ext2_super_block))
;
15.
16.      struct ext2_super_block * sblock = (struct ext2_super_block
*)malloc(sizeof(struct ext2_super_block));
17.
18.      //open any partition for testing,must be ext2/ext3.
19.      int fd = open("/dev/hda3",O_RDONLY);
20.
21.      //skip the boot block
22.      lseek(fd,boot_block_size,SEEK_CUR);
23.
24.      //read the superblock raw data from disk to buff
25.      read(fd,buff,sizeof(struct ext2_super_block));
26.
27.      //copy buffer to sblock, you can use casting or union for th
is.
28.      memcpy((void *)sblock,(void *)buff,sizeof(struct ext2_super
block));
29.
30.      printf("\nmagic number:%u\n",sblock->s_magic);
31.
32.      close(fd);
33.
34.      return 0;
35.  }
36.
```

ΑΝΑΓΝΩΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ group descriptor

```
1.  #include<linux/ext2_fs.h>
2.  #include<sys/types.h>
3.  #include<sys/stat.h>
4.  #include<stdio.h>
5.  #include<unistd.h>
6.  #include<fcntl.h>
7.  #include<stdlib.h>
8.  #include<string.h>
9.
10. int main()
11. {
12.     char *buff = (char *)malloc(sizeof(struct ext2_group_desc));
13.
14.     struct ext2_group_desc * gdesc = (struct ext2_group_desc *)m
alloc(sizeof(struct ext2_group_desc));
15.
16.     //open any partition for testing,must be ext2/ext3.
17.     int fd = open("/dev/hda3",O_RDONLY);
18.
19.     //skip the boot block and super block
20.     lseek(fd,4096,SEEK_CUR);
21.
22.     //read the raw data from disk to buff
23.     read(fd,buff,sizeof(struct ext2_group_desc));
24.
25.     //copy buffer to gdesc, you can use casting or union for thi
s.
26.     memcpy((void *)gdesc,(void *)buff,sizeof(struct ext2_group_d
esc));
27.
28.     //At this position you can be assured that you read group de
scriptor successfully.
29.     close(fd);
30.
31.     return 0;
32. }
33.
```

