ΣΤΑΒΑΡΑΚΗΣ ΠΑΝΑΓΙΩΤΗΣ 2821

ΛΕΤΣΟΣ ΟΔΥΣΣΕΑΣ 2745

Υλοποίηση αρχείου καταγραφής στο σύστημα αρχείων FAT του Linux.

Στην δεύτερη προγραμματιστική άσκηση στο πλαίσιο του μαθήματος Λειτουργικά Συστήματα μας ζητήθηκε να υλοποιήσουμε ένα αρχείο καταγραφής στο σύστημα αρχείων FAT του Linux.

Αρχικά προκειμένου να μπορέσουμε να υλοποιήσουμε το ζητούμενο αρχείο, βασική προυπόθεση ήταν να καταλάβουμε τον τρόπο με τον οποίο λειτουργεί το σύστημα αρχείων FAT.

Αρχίκα η βασική δομή ενός συστήματος αρχείων αποτελείται από ένα μπλόκ δεδομένων στο οποίο μέσα υπάρχουν τα περιεχόμενα ενός αρχείο , ένα μπλοκ μεταδεδομένων (inodes) , περιέχει πληροφορίες για το αρχείο, όπως το μέγεθος του, τα δικαιώματα πρόσβασης και τους χρόνους δημιουργίας, τροποποίησης και πρόσβασης καθώς επίσης περιέχει αναφορές προς τα μπλοκ που περιέχουν το περιεχόμενο του αρχείου.Τα bitmaps τα οποία μας δείχουν ποια block και inodes είναι ελεύθερα, καθώς επίσης και ***ένα*** Superblock το οποίο ουσιαστικά αποθηκεύεται σε έναν συγκεκριμένο τομέα του δίσκου και περιλαμβάνει δεδομένα όπως το μέγεθος του συστήματος αρχείων, τον τύπο του, τον αριθμό των inodes (δείκτες αρχείων), την έκδοση του συστήματος αρχείων και άλλες παραμέτρους.Κατά την εκκίνηση του συστήματος το Superblock είναι αυτό που θα έρθει σε επαφή με τον πυρήνα του λειτουργικού συστήματος ώστε να του δώσει όλες τις απαραίτητες

πληροφορίες που χρειάζεται.

Στην πράξη επείδη κάθε σύστημα αρχείων υποστηρίζει τις λειτουργείες του με δικό του τρόπο στην επικοινωνία των παραπάνω δομών, κατά την υλοποίηση μιάς εφαρμογής όπως στο παράδειγμα της εκφώνησης η cptfs , χρησιμοποίειται ένα Virtual File System , το οποίο αποτελεί την διεπαφή μεταξύ της εφαρμογής και του εκάστοτε συστήματος αρχείου.Το VFS περιλμβάνει 4 βασικά structs struct file: Αυτό το struct αντιπροσωπεύει ένα ανοικτό αρχείο στο σύστημα. Περιλαμβάνει πληροφορίες όπως τον δείκτη στον inode που αναπαριστά το αρχείο, κατάσταση ανοίγματος, τρέχουσα θέση ανάγνωσης/εγγραφής και άλλες σχετικές παράμετρους.

struct inode: Αυτό το struct αντιπροσωπεύει ένα inode, που είναι μια δομή δεδομένων που περιέχει πληροφορίες για ένα αρχείο ή έναν κατάλογο. Περιλαμβάνει πληροφορίες όπως το μέγεθος του αρχείου, οι αδρές δεδομένων (data blocks) που αντιστοιχούν στο αρχείο, οι δικαιώματα πρόσβασης και άλλα χαρακτηριστικά του αρχείου.

struct dentry: Αυτό το struct αντιπροσωπεύει έναν κατάλογο στο σύστημα αρχείων. Κρατά τις πληροφορίες του ονόματος και του inode που αντιστοιχεί σε έναν κατάλογο, καθως και δείκτες που συνδέουν τον κατάλογο με την ιεραρχία των καταλόγων και αρχείων. Το struct dentry περιέχει πληροφορίες που αφορούν το όνομα του αρχείου ή του καταλόγου, καθώς και δείκτες προς τον ανώτερο κατάλογο (parent directory) και τον αντίστοιχο inode.

struct superblock: Αυτό το struct αντιπροσωπεύει το superblock ενός συστήματος αρχείων. Περιλαμβάνει πληροφορίες για το συγκεκριμένο σύστημα αρχείων, όπως τον τύπο του συστήματος αρχείων, τον αριθμό των inodes που υπάρχουν στο σύστημα, το μέγεθος των data blocks, τις διάφορες παραμέτρους και τις λειτουργίες που αφορούν το σύστημα αρχείων.

Στο πιο πρακτίκο μέρος της κατανόησης της άσκησης , χρείαστηκε να δοκιμάσουμε στην πράξη να υλοποιήσουμε την εφαρμογή cptfs.

Η εφαρμογή αυτή εκτελέστηκε με την παρακάτω εντολή και ουσιαστικά , αυτό που κάνει είναι να αντιγράψει το αρχείο lklfuse.c τοπικά στον στην εικονική μηχανή στο / (υποδηλώνει τον κατάλογο ρίζας του συστήματος)

./cptofs -i /tmp/vfatfile -p -t vfat lklfuse.c /

Φυσικά πριν φτάσουμε στο σημείο να εκτελέσουμε την εφαρμογή cptfs

δημιουργήσαμε εκτελέσιμο πρόγραμμα με όλες τις απαραίτητες εντολές

make που αναγράφονται στην εκφώνηση.

Επιπλέον μελετήσαμε την cptfs που υλοποιείται στο αρχείο cptfs.c , αναλύωντας την main συνάρτηση του αρχείου , παρατηρούμε οτι η διαδικάσια που ακολουθείται στην cptfs είναι η εξής:

Αρχικά παίρνει ως τελευταίο όρισμα το αρχείο που θέλουμε να αντιγράψουμε, έπειτα δεσμεύει χώρο στον δίσκο , προκειμένου να αποθηκευτεί αυτο το νεο αρχείο. (*disk.fd = open(cla.fsimg_path, cptofs ? O_RDWR : O_RDONLY);*)

Μετά δημιουργεί ένα mount point και ουσιαστικά επικοινωνεί με την διεπαφή lkl (vfat)προκειμένου να προσαρμώσει την λειτουργεια στον συστημα αρχειών μας,

(ret = lkl_mount_dev(disk_id, cla.part, cla.fsimg_type,

cptofs ? 0 : LKL_MS_RDONLY,

NULL, mpoint, sizeof(mpoint));)

Έπειτα αντιγράφει μέσω κάποων συναρτήσεων που καλούνται, μέσα στον προωρισμό που έχουμε δώσει ως δέυτερο όρισμα το αρχείο ή αρχεία που του έχουμε δώσει ως τελευταίο όρισμα.

(for (i = 0; i < cla.npaths - 1; i++) {

ret = copy_one(cla.paths[i], mpoint, cla.paths[cla.npaths - 1]);

if (ret)

break;

})

και τελικά αποδεσμεύει την μνήμη που είχε εξ αρχής δεσμεύσει και κλείνει τον δίσκο.

(ret = lkl_umount_dev(disk_id, cla.part, 0, 1000);

out_close:

close(disk.fd);)

.

Προκειμένου να αναπτύξουμε μια καλύτερη εικόνα του τι συμβαίνει κατά την εκτέλεση της cptfs μέσα στο σύστημα αρχείου μας , εμπλουτίσαμε τον κώδικα , έτσι ώστε να μπορούμε να εντοπίσουμε ποιές λειτουργίες και ποιές δομές από τις παραπάνω που αναφέραμε 'επισκέπτεται' ο κώδικας κατά την cptfs.

Χρησιμοποιήσαμε την εντολή printk() μια σύναρτηση που υλοποιεί ο πυρήνας του λειτουργικού συστήματος με τα επιθυμητά αποτελέσματα σαν την printf() η οποία δεν μπορούσε να εκτελεστει καθώς χρειάζεται τις βιβλιοθήκες της C για να λειτουργήσει που δεν είναι διαθέσιμες σε επίπεδο πυρήνα., .

Αρχίκα βάλαμε κάποιες τυχαιές εντολές printk() που τελικά δεν μας εξυπηρέτησαν τόσο στην κατανόηση και έτσι , εντοπίσαμε από την εκφώνηση , ποιές λειτουργίες είναι σε ποία δομή και της κάναμε print έτσι ωστε να δούμε πως αλληλεπιδρούν οι δομές μεταξύ τους και αν υπαρχόυν διαφορές ανάλογα με το αν αντιγράφουμε κάποιο αρχείο ή κάποιο κατάλογο αρχείων καθώς και αν επιρρεάζονται τα αποτελέσματα από το μέγεθος των αρχείων.

ΟΛΕΣ ΟΙ ΑΛΛΑΓΕΣ ΕΧΟΥΝ ΓΙΝΕΙ ΣΤΑ ΑΡΧΕΊΑ ΜΕΣΑ **:**

/home/myy601/lkl/lkl-source/fs/fat/

Δοκιμή 1 lklfuse.c

Εδώ παρατηρούμε ότι οι πρώτες λειτουργίες που γίνονται κατά την εφαρμογή είναι αυτές που έχουν να κάνουν με το Superblock.(fat_alloc_inode).

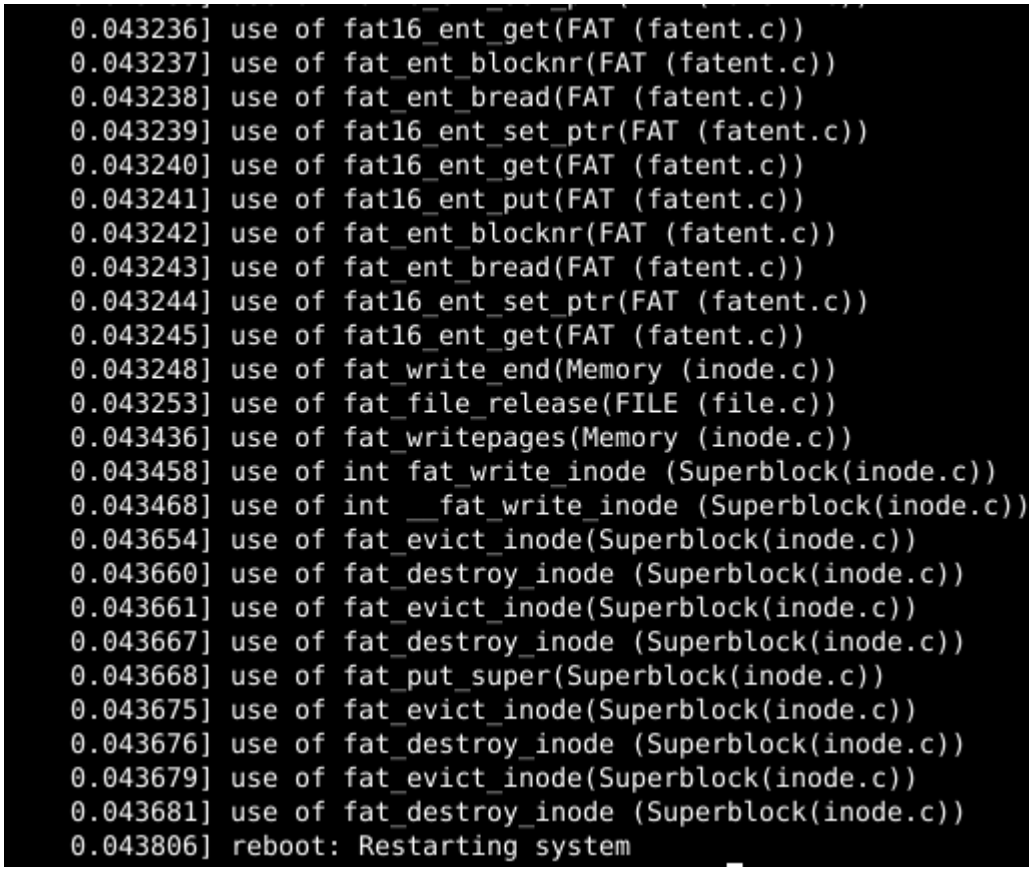
Έπειτα ξεκινάει η διαδικασία που γίνεται ουσιαστικά η αντιγραφή.

Πρώτα εντοπίζεται το αρχείο(vfat_lookup,fat_setatrr) και επειτα με επαναλήψεις των λειτουργιών του FAT , ξεκινάει η αντιγραφή του αρχείου.(fat16_ent_get, fat16_ent_put, fat16_ert_set , etc...)

```
[    0.042090] This architecture does not have kernel memory protec
[    0.042483] use of fat_alloc_inode (Superblock(inode.c))
[    0.042491] use of fat_alloc_inode (Superblock(inode.c))
[    0.042493] use of fat_alloc_inode (Superblock(inode.c))
[    0.043027] use of vfat_lookup(FOLDER (namei_vfat.c))
[    0.043034] use of fat_alloc_inode (Superblock(inode.c))
[    0.043037] use of fat_settattr(INODE  (file.c))
[    0.043040] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043042] use of fat_ent_bread(FAT (fatent.c))
[    0.043047] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043049] use of fat16_ent_get(FAT (fatent.c))
[    0.043050] use of fat16_ent_put(FAT (fatent.c))
[    0.043051] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043052] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043053] use of fat16_ent_get(FAT (fatent.c))
[    0.043054] use of fat16_ent_put(FAT (fatent.c))
[    0.043055] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043056] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043056] use of fat16_ent_get(FAT (fatent.c))
[    0.043057] use of fat16_ent_put(FAT (fatent.c))
[    0.043058] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043059] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043060] use of fat16_ent_get(FAT (fatent.c))
[    0.043061] use of fat16_ent_put(FAT (fatent.c))
[    0.043062] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043063] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043064] use of fat16_ent_get(FAT (fatent.c))
[    0.043064] use of fat16_ent_put(FAT (fatent.c))
[    0.043065] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043066] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043091] use of fat16_ent_get(FAT (fatent.c))
[    0.043093] use of fat16_ent_put(FAT (fatent.c))
[    0.043094] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043095] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043096] use of fat16_ent_get(FAT (fatent.c))
[    0.043097] use of fat16_ent_put(FAT (fatent.c))
[    0.043105] use of fat_write_begin(Memory (inode.c))
[    0.043108] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043109] use of fat_ent_bread(FAT (fatent.c))
[    0.043110] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043111] use of fat16_ent_get(FAT (fatent.c))
[    0.043112] use of fat16_ent_next(FAT (fatent.c))
[    0.043113] use of fat16_ent_get(FAT (fatent.c))
[    0.043113] use of fat16_ent_next(FAT (fatent.c))
[    0.043114] use of fat16_ent_get(FAT (fatent.c))
[    0.043115] use of fat16_ent_next(FAT (fatent.c))
[    0.043116] use of fat16_ent_get(FAT (fatent.c))
[    0.043117] use of fat16_ent_put(FAT (fatent.c))
[    0.043119] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043120] use of fat_ent_bread(FAT (fatent.c))
[    0.043121] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043122] use of fat16_ent_get(FAT (fatent.c))
```

```
[    0.043123] use of fat16_ent_put(FAT (fatent.c))
[    0.043124] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043125] use of fat_ent_bread(FAT (fatent.c))
[    0.043126] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043127] use of fat16_ent_get(FAT (fatent.c))
[    0.043128] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043129] use of fat_ent_bread(FAT (fatent.c))
[    0.043130] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043131] use of fat16_ent_get(FAT (fatent.c))
[    0.043132] use of fat16_ent_put(FAT (fatent.c))
[    0.043133] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043134] use of fat_ent_bread(FAT (fatent.c))
[    0.043135] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043135] use of fat16_ent_get(FAT (fatent.c))
[    0.043140] use of fat_write_end(Memory (inode.c))
[    0.043143] use of fat_write_begin(Memory (inode.c))
[    0.043146] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043147] use of fat_ent_bread(FAT (fatent.c))
[    0.043148] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043149] use of fat16_ent_get(FAT (fatent.c))
[    0.043150] use of fat16_ent_put(FAT (fatent.c))
[    0.043151] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043152] use of fat_ent_bread(FAT (fatent.c))
[    0.043153] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043154] use of fat16_ent_get(FAT (fatent.c))
[    0.043154] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043155] use of fat_ent_bread(FAT (fatent.c))
[    0.043156] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043157] use of fat16_ent_get(FAT (fatent.c))
[    0.043158] use of fat16_ent_put(FAT (fatent.c))
[    0.043159] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043160] use of fat_ent_bread(FAT (fatent.c))
[    0.043161] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043162] use of fat16_ent_get(FAT (fatent.c))
[    0.043163] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043164] use of fat_ent_bread(FAT (fatent.c))
[    0.043165] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043166] use of fat16_ent_get(FAT (fatent.c))
[    0.043167] use of fat16_ent_put(FAT (fatent.c))
[    0.043168] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043169] use of fat_ent_bread(FAT (fatent.c))
[    0.043170] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043171] use of fat16_ent_get(FAT (fatent.c))
[    0.043172] use of fat_ent_blocknr(FAT (fatent.c))
[    0.043172] use of fat_ent_bread(FAT (fatent.c))
[    0.043173] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.043174] use of fat16_ent_get(FAT (fatent.c))
[    0.043175] use of fat16_ent_put(FAT (fatent.c))
```

```
    0.043175] use of fat16_ent_put(FAT (fatent.c))
    0.043176] use of fat_ent_blocknr(FAT (fatent.c))
    0.043177] use of fat_ent_bread(FAT (fatent.c))
    0.043178] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043180] use of fat16_ent_get(FAT (fatent.c))
    0.043184] use of fat_write_end(Memory (inode.c))
    0.043186] use of fat_write_begin(Memory (inode.c))
    0.043188] use of fat_ent_blocknr(FAT (fatent.c))
    0.043189] use of fat_ent_bread(FAT (fatent.c))
    0.043190] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043191] use of fat16_ent_get(FAT (fatent.c))
    0.043192] use of fat16_ent_put(FAT (fatent.c))
    0.043193] use of fat_ent_blocknr(FAT (fatent.c))
    0.043194] use of fat_ent_bread(FAT (fatent.c))
    0.043195] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043196] use of fat16_ent_get(FAT (fatent.c))
    0.043197] use of fat_ent_blocknr(FAT (fatent.c))
    0.043198] use of fat_ent_bread(FAT (fatent.c))
    0.043199] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043200] use of fat16_ent_get(FAT (fatent.c))
    0.043200] use of fat16_ent_put(FAT (fatent.c))
    0.043202] use of fat_ent_blocknr(FAT (fatent.c))
    0.043203] use of fat_ent_bread(FAT (fatent.c))
    0.043203] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043204] use of fat16_ent_get(FAT (fatent.c))
    0.043206] use of fat_ent_blocknr(FAT (fatent.c))
    0.043206] use of fat_ent_bread(FAT (fatent.c))
    0.043207] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043208] use of fat16_ent_get(FAT (fatent.c))
    0.043209] use of fat16_ent_put(FAT (fatent.c))
    0.043210] use of fat_ent_blocknr(FAT (fatent.c))
    0.043211] use of fat_ent_bread(FAT (fatent.c))
    0.043212] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043213] use of fat16_ent_get(FAT (fatent.c))
    0.043214] use of fat_ent_blocknr(FAT (fatent.c))
    0.043215] use of fat_ent_bread(FAT (fatent.c))
    0.043216] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043217] use of fat16_ent_get(FAT (fatent.c))
    0.043217] use of fat16_ent_put(FAT (fatent.c))
    0.043219] use of fat_ent_blocknr(FAT (fatent.c))
    0.043220] use of fat_ent_bread(FAT (fatent.c))
    0.043220] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043221] use of fat16_ent_get(FAT (fatent.c))
    0.043225] use of fat_write_end(Memory (inode.c))
    0.043227] use of fat_write_begin(Memory (inode.c))
    0.043229] use of fat_ent_blocknr(FAT (fatent.c))
    0.043230] use of fat_ent_bread(FAT (fatent.c))
    0.043231] use of fat16_ent_set_ptr(FAT (fatent.c))
    0.043232] use of fat16_ent_get(FAT (fatent.c))
```



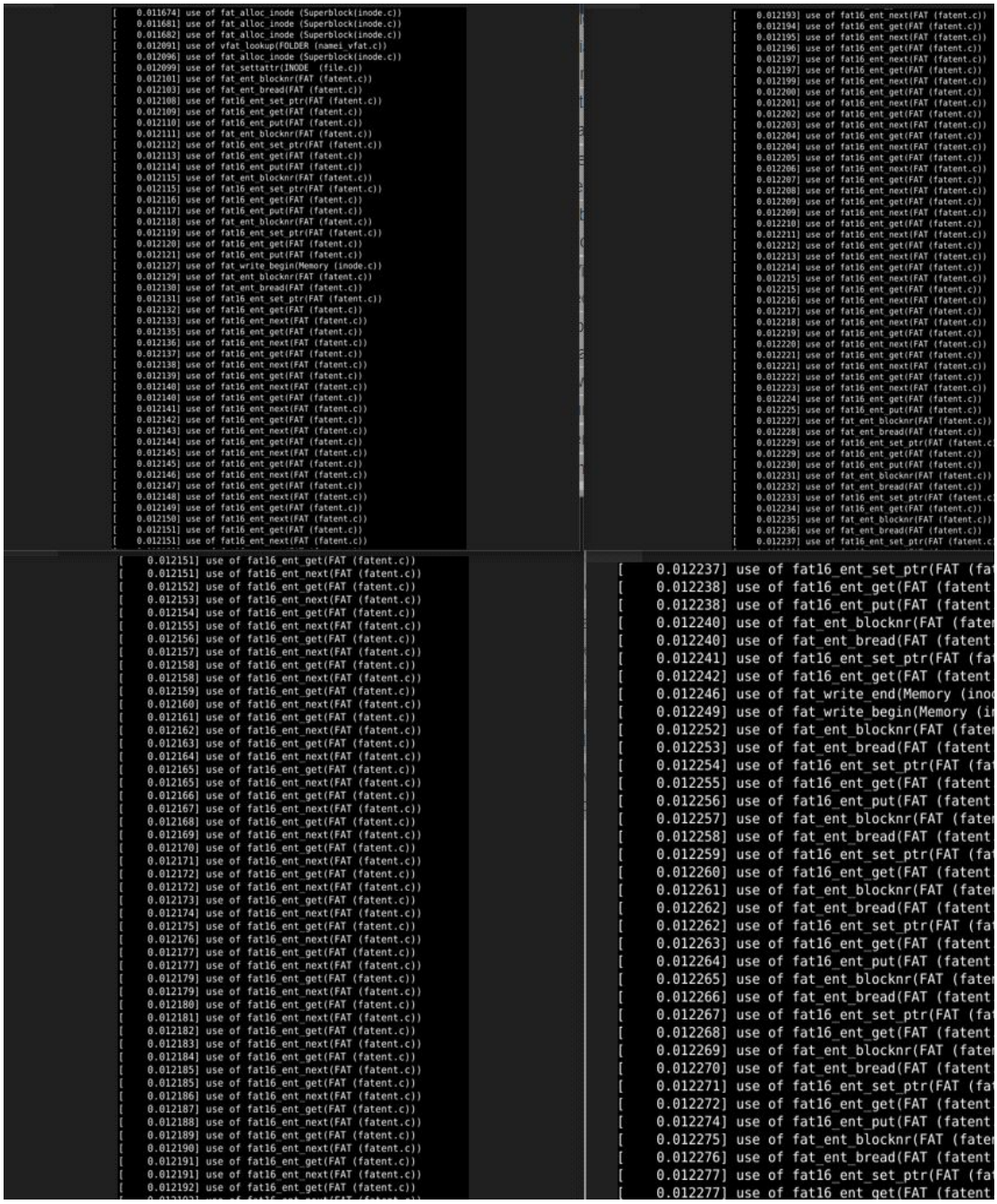
Τελικά με την χρήση της fat_write_pages που είναι υπευθυνη να διαχειριστεί τα παραπάνω inodes , και να μεταφέρει τα δεδομένα από την μνήμη στον δίσκο, ενημερώνοντας και τον FAT.

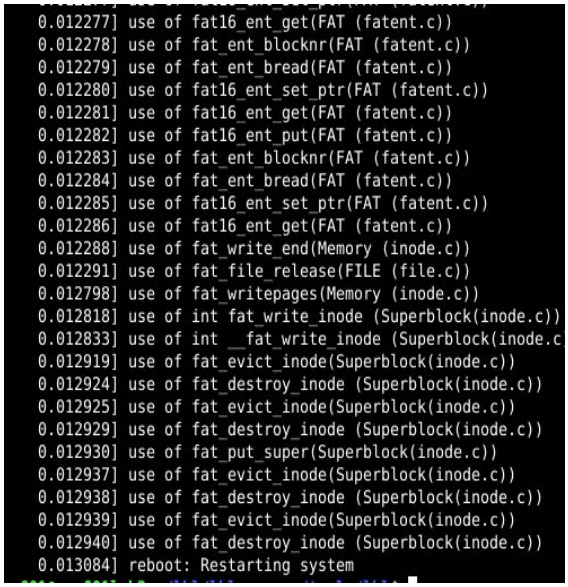
Ακολουθούν οι λειτουργίες fat_evict_inode , fat destroy inode, που αποδευσμεύουν τους πόρους που έχουμε δευσμεύσει.

ΔΟΚΙΜΗ 2

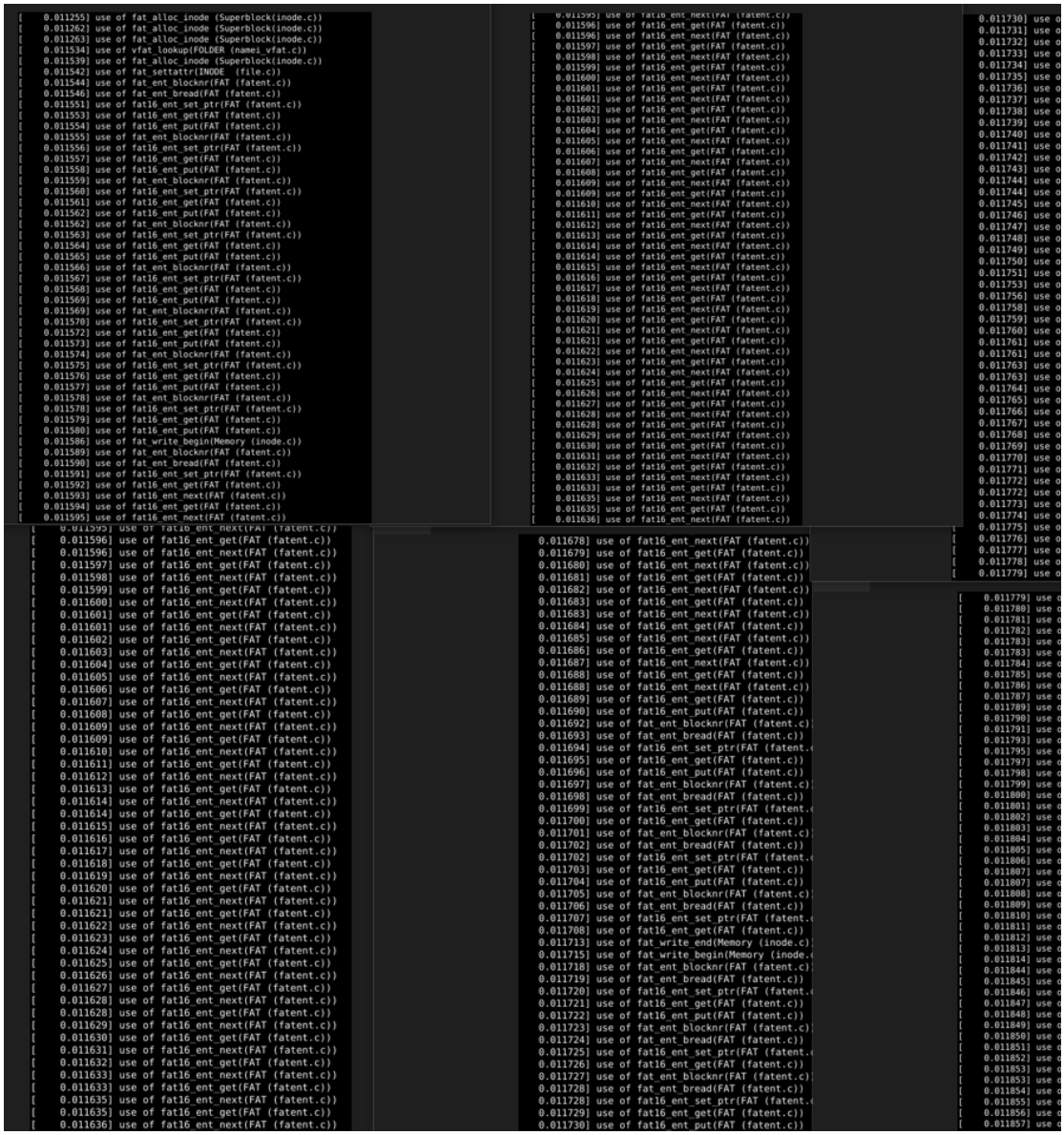
ΔΙΚΟ ΜΑΣ ΑΡΧΕΙΟ info.txt σε σύγκριση με το info2.txt που περιλαμβάνει το περιεχόμενο του αρχείου info.txt 2 φορές.

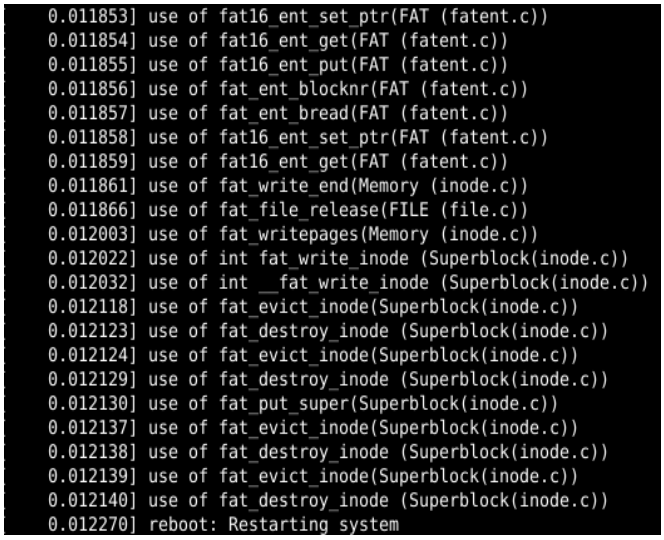
ΕΚΤΕΛΕΣΗ info.txt


[0.011674] use of fat_alloc_inode (Superblock(inode.c))
[0.011681] use of fat_alloc_inode (Superblock(inode.c))
[0.011682] use of fat_alloc_inode (Superblock(inode.c))
[0.012091] use of vfat_lookup(FOLDER (namei_vfat.c))
[0.012096] use of fat_alloc_inode (Superblock(inode.c))
[0.012099] use of fat_setattr(INODE (file.c))
[0.012101] use of fat_ent_blocknr(FAT (fatent.c))
[0.012103] use of fat_ent_bread(FAT (fatent.c))
[0.012108] use of fat16_ent_set_ptr(FAT (fatent.c))
[0.012109] use of fat16_ent_get(FAT (fatent.c))
[0.012110] use of fat16_ent_put(FAT (fatent.c))
[0.012111] use of fat_ent_blocknr(FAT (fatent.c))
[0.012112] use of fat16_ent_set_ptr(FAT (fatent.c))
[0.012113] use of fat16_ent_get(FAT (fatent.c))
[0.012114] use of fat16_ent_put(FAT (fatent.c))
[0.012115] use of fat_ent_blocknr(FAT (fatent.c))
[0.012115] use of fat16_ent_set_ptr(FAT (fatent.c))
[0.012116] use of fat16_ent_get(FAT (fatent.c))
[0.012117] use of fat16_ent_put(FAT (fatent.c))
[0.012118] use of fat_ent_blocknr(FAT (fatent.c))
[0.012119] use of fat16_ent_set_ptr(FAT (fatent.c))
[0.012120] use of fat16_ent_get(FAT (fatent.c))
[0.012121] use of fat16_ent_put(FAT (fatent.c))
[0.012127] use of fat_write_begin(Memory (inode.c))
[0.012129] use of fat_ent_blocknr(FAT (fatent.c))
[0.012130] use of fat_ent_bread(FAT (fatent.c))
[0.012131] use of fat16_ent_set_ptr(FAT (fatent.c))
[0.012132] use of fat16_ent_get(FAT (fatent.c))
[0.012133] use of fat16_ent_next(FAT (fatent.c))
[0.012135] use of fat16_ent_get(FAT (fatent.c))
[0.012136] use of fat16_ent_next(FAT (fatent.c))
[0.012137] use of fat16_ent_get(FAT (fatent.c))
[0.012138] use of fat16_ent_next(FAT (fatent.c))
[0.012139] use of fat16_ent_get(FAT (fatent.c))
[0.012140] use of fat16_ent_next(FAT (fatent.c))
[0.012140] use of fat16_ent_get(FAT (fatent.c))
[0.012141] use of fat16_ent_next(FAT (fatent.c))
[0.012142] use of fat16_ent_get(FAT (fatent.c))
[0.012143] use of fat16_ent_next(FAT (fatent.c))
[0.012144] use of fat16_ent_get(FAT (fatent.c))
[0.012145] use of fat16_ent_next(FAT (fatent.c))
[0.012145] use of fat16_ent_get(FAT (fatent.c))
[0.012146] use of fat16_ent_next(FAT (fatent.c))
[0.012147] use of fat16_ent_get(FAT (fatent.c))
[0.012148] use of fat16_ent_next(FAT (fatent.c))
[0.012149] use of fat16_ent_get(FAT (fatent.c))
[0.012150] use of fat16_ent_next(FAT (fatent.c))
[0.012151] use of fat16_ent_get(FAT (fatent.c))
[0.012151] use of fat16_ent_next(FAT (fatent.c))
[0.012193] use of fat16_ent_next(FAT (fatent.c))
[0.012194] use of fat16_ent_get(FAT (fatent.c))
[0.012195] use of fat16_ent_next(FAT (fatent.c))
[0.012196] use of fat16_ent_get(FAT (fatent.c))
[0.012197] use of fat16_ent_next(FAT (fatent.c))
[0.012197] use of fat16_ent_get(FAT (fatent.c))
[0.012199] use of fat16_ent_next(FAT (fatent.c))
[0.012200] use of fat16_ent_get(FAT (fatent.c))
[0.012201] use of fat16_ent_next(FAT (fatent.c))
[0.012202] use of fat16_ent_get(FAT (fatent.c))
[0.012203] use of fat16_ent_next(FAT (fatent.c))
[0.012204] use of fat16_ent_get(FAT (fatent.c))
[0.012204] use of fat16_ent_next(FAT (fatent.c))
[0.012205] use of fat16_ent_get(FAT (fatent.c))
[0.012206] use of fat16_ent_next(FAT (fatent.c))
[0.012207] use of fat16_ent_get(FAT (fatent.c))
[0.012208] use of fat16_ent_next(FAT (fatent.c))
[0.012209] use of fat16_ent_get(FAT (fatent.c))
[0.012209] use of fat16_ent_next(FAT (fatent.c))
[0.012210] use of fat16_ent_get(FAT (fatent.c))
[0.012211] use of fat16_ent_next(FAT (fatent.c))
[0.012212] use of fat16_ent_get(FAT (fatent.c))
[0.012213] use of fat16_ent_next(FAT (fatent.c))
[0.012214] use of fat16_ent_get(FAT (fatent.c))
[0.012215] use of fat16_ent_next(FAT (fatent.c))
[0.012215] use of fat16_ent_get(FAT (fatent.c))
[0.012216] use of fat16_ent_next(FAT (fatent.c))
[0.012217] use of fat16_ent_get(FAT (fatent.c))
[0.012218] use of fat16_ent_next(FAT (fatent.c))
[0.012219] use of fat16_ent_get(FAT (fatent.c))
[0.012220] use of fat16_ent_next(FAT (fatent.c))
[0.012221] use of fat16_ent_get(FAT (fatent.c))
[0.012221] use of fat16_ent_next(FAT (fatent.c))
[0.012222] use of fat16_ent_get(FAT (fatent.c))
[0.012223] use of fat16_ent_next(FAT (fatent.c))
[0.012224] use of fat16_ent_get(FAT (fatent.c))
[0.012225] use of fat16_ent_put(FAT (fatent.c))
[0.012227] use of fat_ent_blocknr(FAT (fatent.c))
[0.012228] use of fat_ent_bread(FAT (fatent.c))
[0.012229] use of fat16_ent_set_ptr(FAT (fatent.c
[0.012229] use of fat16_ent_get(FAT (fatent.c))
[0.012230] use of fat16_ent_put(FAT (fatent.c))
[0.012231] use of fat_ent_blocknr(FAT (fatent.c))
[0.012232] use of fat_ent_bread(FAT (fatent.c))
[0.012233] use of fat16_ent_set_ptr(FAT (fatent.c
[0.012234] use of fat16_ent_get(FAT (fatent.c))
[0.012235] use of fat_ent_blocknr(FAT (fatent.c))
[0.012236] use of fat_ent_bread(FAT (fatent.c))
[0.012237] use of fat16_ent_set_ptr(FAT (fatent.c
[0.012237] use of fat16_ent_set_ptr(FAT (fa
[0.012238] use of fat16_ent_get(FAT (fatent
[0.012238] use of fat16_ent_put(FAT (fatent
[0.012240] use of fat_ent_blocknr(FAT (fate
[0.012240] use of fat_ent_bread(FAT (fatent
[0.012241] use of fat16_ent_set_ptr(FAT (fa
[0.012242] use of fat16_ent_get(FAT (fatent
[0.012246] use of fat_write_end(Memory (ino
[0.012249] use of fat_write_begin(Memory (i
[0.012252] use of fat_ent_blocknr(FAT (fate
[0.012253] use of fat_ent_bread(FAT (fatent
[0.012254] use of fat16_ent_set_ptr(FAT (fa
[0.012255] use of fat16_ent_get(FAT (fatent
[0.012256] use of fat16_ent_put(FAT (fatent
[0.012257] use of fat_ent_blocknr(FAT (fate
[0.012258] use of fat_ent_bread(FAT (fatent
[0.012259] use of fat16_ent_set_ptr(FAT (fa
[0.012260] use of fat16_ent_get(FAT (fatent
[0.012261] use of fat_ent_blocknr(FAT (fate
[0.012262] use of fat_ent_bread(FAT (fatent
[0.012262] use of fat16_ent_set_ptr(FAT (fa
[0.012263] use of fat16_ent_get(FAT (fatent
[0.012264] use of fat16_ent_put(FAT (fatent
[0.012265] use of fat_ent_blocknr(FAT (fate
[0.012266] use of fat_ent_bread(FAT (fatent
[0.012267] use of fat16_ent_set_ptr(FAT (fa
[0.012268] use of fat16_ent_get(FAT (fatent
[0.012269] use of fat_ent_blocknr(FAT (fate
[0.012270] use of fat_ent_bread(FAT (fatent
[0.012271] use of fat16_ent_set_ptr(FAT (fa
[0.012272] use of fat16_ent_get(FAT (fatent
[0.012274] use of fat16_ent_put(FAT (fatent
[0.012275] use of fat_ent_blocknr(FAT (fate
[0.012276] use of fat_ent_bread(FAT (fatent
[0.012277] use of fat16_ent_set_ptr(FAT (fa
[0.012277] use of fat16_ent_get(FAT (fatent



εκτέλεση info2.txt





Με την εκτέλεση της συγκεκριμένης δοκιμής , διαπιστώνουμε οτι όντως για την αντιγραφή μεγαλύτερου αρχείου(info2.txt) οι λειτουργίες που αναφέραμε παραπανω, εκτελούνται περισσότες φορές από οτι σε ενα μικροτερο αρχειο(info.txt)

Αυτό συμβαίνει γιατί η fat_ent_put κάνει ουσιαστικά περισσότερες καταγραφές στον FAT , αντιστοιχα ηfat_ent_get περισσότερες αναγνώσεις, και η fat_ent_set περισσοτερες τροποποιησεις τιμών σε θέσεις του FAT.

ΔΟΚΙΜΗ 3 ΜΕ ΚΑΤΑΛΟΓΟΥΣ

ΚΑΤΑΛΟΓΟΣ ΜΕ 2 ΑΡΧΕΙΑ ΤΟ INFO3.TXT ΚΑΙ ΤΟ INFO4.TXT

```
[    0.012702] This architecture does not have kernel memory protection.
[    0.013578] use of fat_alloc_inode (Superblock(inode.c))
[    0.013586] use of fat_alloc_inode (Superblock(inode.c))
[    0.013587] use of fat_alloc_inode (Superblock(inode.c))
[    0.013736] use of vfat_lookup(FOLDER (namei_vfat.c))
[    0.013747] use of vfat_mkdir(FOLDER (namei_vfat.c))
[    0.013749] use of fat_ent_blocknr(FAT (fatent.c))
[    0.013750] use of fat_ent_bread(FAT (fatent.c))
[    0.013755] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.013756] use of fat16_ent_get(FAT (fatent.c))
[    0.013772] use of fat16_ent_next(FAT (fatent.c))
[    0.013775] use of fat16_ent_get(FAT (fatent.c))
[    0.013776] use of fat16_ent_next(FAT (fatent.c))
[    0.013777] use of fat16_ent_get(FAT (fatent.c))
[    0.013778] use of fat16_ent_next(FAT (fatent.c))
[    0.013779] use of fat16_ent_get(FAT (fatent.c))
[    0.013780] use of fat16_ent_next(FAT (fatent.c))
[    0.013780] use of fat16_ent_get(FAT (fatent.c))
[    0.013781] use of fat16_ent_next(FAT (fatent.c))
[    0.013782] use of fat16_ent_get(FAT (fatent.c))
[    0.013783] use of fat16_ent_next(FAT (fatent.c))
[    0.013784] use of fat16_ent_get(FAT (fatent.c))
[    0.013785] use of fat16_ent_next(FAT (fatent.c))
[    0.013786] use of fat16_ent_get(FAT (fatent.c))
[    0.013787] use of fat16_ent_next(FAT (fatent.c))
[    0.013788] use of fat16_ent_get(FAT (fatent.c))
[    0.013788] use of fat16_ent_next(FAT (fatent.c))
[    0.013789] use of fat16_ent_get(FAT (fatent.c))
[    0.013790] use of fat16_ent_next(FAT (fatent.c))
[    0.013791] use of fat16_ent_get(FAT (fatent.c))
[    0.013794] use of fat16_ent_next(FAT (fatent.c))
[    0.013795] use of fat16_ent_get(FAT (fatent.c))
[    0.013796] use of fat16_ent_next(FAT (fatent.c))
[    0.013797] use of fat16_ent_get(FAT (fatent.c))
[    0.013798] use of fat16_ent_next(FAT (fatent.c))
[    0.013799] use of fat16_ent_get(FAT (fatent.c))
```

```
    0.013799] use of fat16_ent_next(FAT (fatent.c))
    0.013800] use of fat16_ent_get(FAT (fatent.c))
    0.013801] use of fat16_ent_next(FAT (fatent.c))
    0.013816] use of fat16_ent_get(FAT (fatent.c))
    0.013817] use of fat16_ent_next(FAT (fatent.c))
    0.013818] use of fat16_ent_get(FAT (fatent.c))
    0.013819] use of fat16_ent_next(FAT (fatent.c))
    0.013820] use of fat16_ent_get(FAT (fatent.c))
    0.013821] use of fat16_ent_next(FAT (fatent.c))
    0.013822] use of fat16_ent_get(FAT (fatent.c))
    0.013823] use of fat16_ent_next(FAT (fatent.c))
    0.013824] use of fat16_ent_get(FAT (fatent.c))
    0.013837] use of fat16_ent_next(FAT (fatent.c))
    0.013838] use of fat16_ent_get(FAT (fatent.c))
    0.013839] use of fat16_ent_next(FAT (fatent.c))
    0.013840] use of fat16_ent_get(FAT (fatent.c))
    0.013841] use of fat16_ent_next(FAT (fatent.c))
    0.013842] use of fat16_ent_get(FAT (fatent.c))
    0.013842] use of fat16_ent_next(FAT (fatent.c))
    0.013843] use of fat16_ent_get(FAT (fatent.c))
    0.013844] use of fat16_ent_next(FAT (fatent.c))
    0.013845] use of fat16_ent_get(FAT (fatent.c))
    0.013846] use of fat16_ent_next(FAT (fatent.c))
    0.013847] use of fat16_ent_get(FAT (fatent.c))
    0.013848] use of fat16_ent_next(FAT (fatent.c))
    0.013849] use of fat16_ent_get(FAT (fatent.c))
    0.013850] use of fat16_ent_next(FAT (fatent.c))
    0.013851] use of fat16_ent_get(FAT (fatent.c))
    0.013851] use of fat16_ent_next(FAT (fatent.c))
    0.013852] use of fat16_ent_get(FAT (fatent.c))
    0.013853] use of fat16_ent_next(FAT (fatent.c))
    0.013854] use of fat16_ent_get(FAT (fatent.c))
    0.013855] use of fat16_ent_next(FAT (fatent.c))
    0.013856] use of fat16_ent_get(FAT (fatent.c))
    0.013857] use of fat16_ent_next(FAT (fatent.c))
    0.013858] use of fat16_ent_get(FAT (fatent.c))
    0.013859] use of fat16_ent_next(FAT (fatent.c))
    0.013859] use of fat16_ent_get(FAT (fatent.c))
    0.013860] use of fat16_ent_next(FAT (fatent.c))
    0.013861] use of fat16_ent_get(FAT (fatent.c))
    0.013862] use of fat16_ent_next(FAT (fatent.c))
    0.013863] use of fat16_ent_get(FAT (fatent.c))
    0.013864] use of fat16_ent_next(FAT (fatent.c))
    0.013865] use of fat16_ent_get(FAT (fatent.c))
    0.013866] use of fat16_ent_next(FAT (fatent.c))
    0.013867] use of fat16_ent_get(FAT (fatent.c))
    0.013868] use of fat16_ent_next(FAT (fatent.c))
    0.013868] use of fat16_ent_get(FAT (fatent.c))
```

```
[    0.013866] use of fat16_ent_next(FAT (fatent.c))
[    0.013867] use of fat16_ent_get(FAT (fatent.c))
[    0.013868] use of fat16_ent_next(FAT (fatent.c))
[    0.013868] use of fat16_ent_get(FAT (fatent.c))
[    0.013869] use of fat16_ent_next(FAT (fatent.c))
[    0.013870] use of fat16_ent_get(FAT (fatent.c))
[    0.013871] use of fat16_ent_next(FAT (fatent.c))
[    0.013872] use of fat16_ent_get(FAT (fatent.c))
[    0.013873] use of fat16_ent_next(FAT (fatent.c))
[    0.013874] use of fat16_ent_get(FAT (fatent.c))
[    0.013875] use of fat16_ent_next(FAT (fatent.c))
[    0.013876] use of fat16_ent_get(FAT (fatent.c))
[    0.013877] use of fat16_ent_next(FAT (fatent.c))
[    0.013877] use of fat16_ent_get(FAT (fatent.c))
[    0.013878] use of fat16_ent_next(FAT (fatent.c))
[    0.013879] use of fat16_ent_get(FAT (fatent.c))
[    0.013880] use of fat16_ent_next(FAT (fatent.c))
[    0.013881] use of fat16_ent_get(FAT (fatent.c))
[    0.013882] use of fat16_ent_next(FAT (fatent.c))
[    0.013883] use of fat16_ent_get(FAT (fatent.c))
[    0.013884] use of fat16_ent_next(FAT (fatent.c))
[    0.013885] use of fat16_ent_get(FAT (fatent.c))
[    0.013886] use of fat16_ent_next(FAT (fatent.c))
[    0.013888] use of fat16_ent_get(FAT (fatent.c))
[    0.013889] use of fat16_ent_next(FAT (fatent.c))
[    0.013890] use of fat16_ent_get(FAT (fatent.c))
[    0.013891] use of fat16_ent_next(FAT (fatent.c))
[    0.013892] use of fat16_ent_get(FAT (fatent.c))
[    0.013893] use of fat16_ent_next(FAT (fatent.c))
[    0.013894] use of fat16_ent_get(FAT (fatent.c))
[    0.013894] use of fat16_ent_next(FAT (fatent.c))
[    0.013896] use of fat16_ent_get(FAT (fatent.c))
[    0.013896] use of fat16_ent_next(FAT (fatent.c))
[    0.013897] use of fat16_ent_get(FAT (fatent.c))
[    0.013898] use of fat16_ent_next(FAT (fatent.c))
[    0.013899] use of fat16_ent_get(FAT (fatent.c))
[    0.013900] use of fat16_ent_next(FAT (fatent.c))
[    0.013901] use of fat16_ent_get(FAT (fatent.c))
[    0.013902] use of fat16_ent_next(FAT (fatent.c))
[    0.013903] use of fat16_ent_get(FAT (fatent.c))
[    0.013904] use of fat16_ent_next(FAT (fatent.c))
[    0.013905] use of fat16_ent_get(FAT (fatent.c))
[    0.013905] use of fat16_ent_next(FAT (fatent.c))
[    0.013906] use of fat16_ent_get(FAT (fatent.c))
[    0.013907] use of fat16_ent_next(FAT (fatent.c))
[    0.013908] use of fat16_ent_get(FAT (fatent.c))
[    0.013909] use of fat16_ent_next(FAT (fatent.c))
[    0.013910] use of fat16_ent_get(FAT (fatent.c))
[    0.013911] use of fat16_ent_next(FAT (fatent.c))
[    0.013912] use of fat16_ent_get(FAT (fatent.c))
[    0.013913] use of fat16_ent_next(FAT (fatent.c))
[    0.013914] use of fat16_ent_get(FAT (fatent.c))
```

```
[    0.013962] use of fat_ent_blocknr(FAT (fatent.c))
[    0.013963] use of fat_ent_bread(FAT (fatent.c))
[    0.013964] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.013965] use of fat16_ent_get(FAT (fatent.c))
[    0.013967] use of (fat_validate_dir)
[    0.013993] use of vfat_lookup(FOLDER (namei_vfat.c))
[    0.013995] use of vfat_create(FOLDER (namei_vfat.c))
[    0.013998] use of fat_alloc_inode (Superblock(inode.c))
[    0.014006] use of fat_write_begin(Memory (inode.c))
[    0.014026] use of fat_ent_blocknr(FAT (fatent.c))
[    0.014028] use of fat_ent_bread(FAT (fatent.c))
[    0.014029] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.014030] use of fat16_ent_get(FAT (fatent.c))
[    0.014031] use of fat16_ent_put(FAT (fatent.c))
[    0.014047] use of fat_write_end(Memory (inode.c))
[    0.014054] use of fat_file_release(FILE (file.c))
[    0.014067] use of vfat_lookup(FOLDER (namei_vfat.c))
[    0.014069] use of vfat_create(FOLDER (namei_vfat.c))
[    0.014072] use of fat_alloc_inode (Superblock(inode.c))
[    0.014077] use of fat_write_begin(Memory (inode.c))
[    0.014080] use of fat_ent_blocknr(FAT (fatent.c))
[    0.014081] use of fat_ent_bread(FAT (fatent.c))
[    0.014081] use of fat16_ent_set_ptr(FAT (fatent.c))
[    0.014082] use of fat16_ent_get(FAT (fatent.c))
[    0.014083] use of fat16_ent_put(FAT (fatent.c))
[    0.014087] use of fat_write_end(Memory (inode.c))
[    0.014089] use of fat_file_release(FILE (file.c))
[    0.014257] use of int fat_write_inode (Superblock(inode.c))
[    0.014270] use of int __fat_write_inode (Superblock(inode.c))
[    0.014272] use of int fat_write_inode (Superblock(inode.c))
[    0.014274] use of int __fat_write_inode (Superblock(inode.c))
[    0.014284] use of fat_writepages(Memory (inode.c))
[    0.014291] use of int fat_write_inode (Superblock(inode.c))
[    0.014303] use of int __fat_write_inode (Superblock(inode.c))
[    0.014305] use of fat_writepages(Memory (inode.c))
[    0.014317] use of int fat_write_inode (Superblock(inode.c))
[    0.014319] use of int __fat_write_inode (Superblock(inode.c))
[    0.015150] use of fat_evict_inode(Superblock(inode.c))
[    0.015163] use of fat_destroy_inode (Superblock(inode.c))
[    0.015165] use of fat_evict_inode(Superblock(inode.c))
[    0.015166] use of fat_destroy_inode (Superblock(inode.c))
[    0.015168] use of fat_evict_inode(Superblock(inode.c))
[    0.015181] use of fat_destroy_inode (Superblock(inode.c))
[    0.015191] use of fat_evict_inode(Superblock(inode.c))
[    0.015194] use of fat_destroy_inode (Superblock(inode.c))
[    0.015195] use of fat_put_super(Superblock(inode.c))
[    0.015203] use of fat_evict_inode(Superblock(inode.c))
[    0.015213] use of fat_destroy_inode (Superblock(inode.c))
[    0.015214] use of fat_evict_inode(Superblock(inode.c))
[    0.015216] use of fat_destroy_inode (Superblock(inode.c))
[    0.015319] reboot: Restarting system
```

Τέλος έγινε μια δοκιμή αντιγραφής ολόκληρου καταλόγου που περιείχε 2 αρχεία μέσα.

Παρατηρήθηκε λοιπόν οτι πράγματι αρχικά έγινε η δημιουργία του καταλόγου με την λειτουργία vfat_mkdir και έπειτα δημιουργήθηκαν τα 2 αρχεία , αφού υπάρχουν 2 φορές οι λειτουργιές vfat_look_up και vfat_create για την δημιουργία των αρχείων που περιέχονται στον κατάλογο.

Έπειτα μας ζητήθηκε να εντοπίσουμε τις αλλαγές στις βασικές δομές του συστήματος αρχείων , και να τις καταγράψουμε σε ένα αρχείο journal , το οποίο θα αποθηκευόταν στο τοπίκο σύστημα αρχείων της εικόνικής μηχανής που χρησιμοποιήσαμε χρησιμοποιώντας κανονικές κλήσεις συστήματος open , write.

Εντοπίσαμε κάποιες από τις σημαντικές αλλαγές κυρίως στα πεδία της δομής Superblock , παρόλα αυτά , δεν δημιουργήσαμε το αρχείο καθώς όταν πήγαμε να κάνουμε #include <fcntl.h> απαραίτητη βιβλιοθήκη για την χρήση open(), write() λάβαμε το παρακάτω σφάλμα.

```
fs/fat/inode.c:12:10: fatal error: fcntl.h: No such file or directory
 #include <fcntl.h>
```

Τελικά τις αλλαγές που εντοπίσαμε τις αποθηκεύσαμε σε ένα αρχείο journal αποθηκευμένο στο ίδιο το σύστημα FAT.

Αρχικά, αρχικοποιήσαμε έναν ακέραιο προκειμένου να χρησιμοποιήθει για το άνοιγμα του αρχείου journal.

Η αρχικοποίηση έγινε στο struct του Superblock και συγκερκιμένα στο fat.h ως εξης:

```
105         int arxeio_katagrafis;
```

Έπειτα χρησιμοποιήσαμε την συνάρτηση sys_open() προκείμενου να ανοίξουμε το αρχείο, και μετά σε κάθε σημαντική αλλαγή συναντάμε στις βασικές δομές του συστήματος να τις καταγράφουμε με sys_write.

Για το άνοιγμα του αρχείου χρησιμοποιήσαμε την εντολή :

```
sbi->arxeio_katagrafis = sys_open("journal",O_APPEND|O_CREAT|O_RDWR,0644)
```

στο αρχείο inode.c στην λειτουργία fat_fill_super. ,

O_APPEND δηλώνει ότι τα δεδομένα θα εγγράφονται στο τέλος του αρχείου

O_CREAT δηλώνει ότι το αρχείο θα δημιουργηθεί αν δεν υπάρχει ήδη

O_RDWR δηλώνει ότι το αρχείο θα ανοίξει για ανάγνωση και εγγραφή.

Οι άδειες **0644**: το αρχείο θα έχει δικαιώματα εγγραφής και ανάγνωσης.

Έπειτα επιλέξαμε κάποιες από τις βασικές μεταβολές στις δομές του συστήματος και τις αντιγράψαμε στο αρχείο journal χρησιμοποιώντας την sys_write με ορίσματα:

fd: Ο αριθμός αρχείου περιγραφέα (file descriptor) που αναφέρεται στο ανοιχτό αρχείο.

buf: Δείκτης σε έναν χώρο μνήμης που περιέχει τα δεδομένα που πρόκειται να εγγραφούν.

count: Ο αριθμός των bytes που πρόκειται να εγγραφούν.

Για τον buffer χρησιμοποιήσαμε την **kzalloc** αντί για την malloc :

```
1642
1643        char *temp = kzalloc(sizeof(struct msdos_sb_info), GFP_KERNEL);
```

Για την χρήση της sys_write() έγινε include η βιβλιοθήκη
#include<linux/syscalls.h>

Αποθηκεύτηκαν αναλυτικά οι παρακάτω αλλαγές:

Για το SuperBlock: (inode.c)

```
1732
1733        sbi->cluster_size = sb->s_blocksize * sbi->sec_per_clus;
1734        counter+=1;
1735
1736        sprintf(temp, "%u: sb info cluster_size change to :  %u",counter, sbi->cluster_si
1737        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1738        sys_fsync(sbi->arxeio_katagrafis);
1739        sys_fdatasync(sbi->arxeio_katagrafis);
1740        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1741
```

sbi->cluster_size: Το cluster_size αναφέρεται στο μέγεθος του cluster στο αρχείο συστήματος, που είναι μια μονάδα αποθήκευσης όπου τα δεδομένα αρχείων αποθηκεύονται στον δίσκο.

```
1742        sbi->cluster_bits = ffs(sbi->cluster_size) - 1;
1743        counter+=1;
1744
1745        sprintf(temp, "%u: sb_info cluster_bits change to:  %u",counter,sbi->cluster_bits);
1746        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1747        sys_fsync(sbi->arxeio_katagrafis);
1748        sys_fdatasync(sbi->arxeio_katagrafis);
1749        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1750
1751
1752        sbi->fats = bpb.fat_fats;
1753        counter+=1;
1754
1755        sprintf(temp, "%u: sb_info fats change to:  %u",counter, sbi->fats);
1756        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1757        sys_fsync(sbi->arxeio_katagrafis);
1758        sys_fdatasync(sbi->arxeio_katagrafis);
1759        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1760
```

sbi->cluster_bits: αναφέρεται στον αριθμό των bits που απαιτούνται για να αναπαραστήσουν ένα cluster του αρχειου συστήματος

sbi->fats:αναφέρεται στον αριθμό των File Allocation Tables (FATs) στο αρχείο συστήματος

```
1762        sbi->fat_bits = 0;              /* Don't know yet */
1763        counter+=1;
1764
1765        sprintf(temp, "%u: sb_info fat_bits change to:  %u",counter, sbi->fat_bits);
1766        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1767        sys_fsync(sbi->arxeio_katagrafis);
1768        sys_fdatasync(sbi->arxeio_katagrafis);
1769        printk(KERN_INFO "this is about to be written to journal:%s\n\n",temp);
1770
1771
1772        sbi->fat_start = bpb.fat_reserved;
1773        counter+=1;
1774
1775        sprintf(temp, "%u: sb_info fat_start change to:  %u", counter, sbi->fat_start);
1776        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1777        sys_fsync(sbi->arxeio_katagrafis);
1778        sys_fdatasync(sbi->arxeio_katagrafis);
1779        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1780
1781
1782        sbi->fat_length = bpb.fat_fat_length;
1783        counter+=1;
1784
1785        sprintf(temp, "%u: sb_info fat_length change to:  %lu", counter , sbi->fat_length);
1786        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1787        sys_fsync(sbi->arxeio_katagrafis);
1788        sys_fdatasync(sbi->arxeio_katagrafis);
1789        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1790
1791
1792        sbi->root_cluster = 0;
1793
1794        counter+=1;
1795        sprintf(temp, "%u: sb_info root_cluster change to:  %lu",counter, sbi->root_cluster);
1796        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1797        sys_fsync(sbi->arxeio_katagrafis);
1798        sys_fdatasync(sbi->arxeio_katagrafis);
1799        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1800
1801
1802        sbi->free_clusters = -1;        /* Don't know yet */
1803        counter+=1;
1804
1805        sprintf(temp, "%u: sb_info fats change to:  %u",counter, sbi->free_clusters);
1806        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1807        sys_fsync(sbi->arxeio_katagrafis);
```

sbi->fat_bits: Αναφέρεται στον αριθμό των bits που απαιτούνται για να αναπαραστήσουν μια εγγραφή στον πίνακα FAT

sbi->fat_start: Αναφέρεται στην αρχική θέση του πίνακα FAT στο αρχείο συστήματος.

sbi->fat_length: Αναφέρεται στο μήκος του πίνακα FAT στο αρχείο συστήματος.

sbi->root_cluster: Αναφέρεται στον αριθμό του root directory cluster στο αρχείο στο αρχείο συστήματος

sbi->free_clusters: αναφέρεται στον αριθμό των ελεύθερων clusters που υπάρχουν στο αρχείο συστήματος.

```
1814
1815        sprintf(temp, "%u: sb_info free_clus_valid change to:  %u",counter, sbi->free_clus_valid);
1816        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1817        sys_fsync(sbi->arxeio_katagrafis);
1818        sys_fdatasync(sbi->arxeio_katagrafis);
1819        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1820
1821
1822        sbi->prev_free = FAT_START_ENT;
1823        counter+=1;
1824        sprintf(temp, "%u: sb_info prev_free change to:  %u",counter,sbi->prev_free);
1825        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1826        sys_fsync(sbi->arxeio_katagrafis);
1827        sys_fdatasync(sbi->arxeio_katagrafis);
1828        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
1829
```

sbi->free_clus_valid: Αυτό το πεδίο υποδεικνύει εάν ο αριθμός των ελεύθερων cluster είναι έγκυρος και ενημερωμένος

sbi->prev_free:Αυτό το πεδίο περιέχει τον αριθμό του προηγούμενου ελεύθερου cluster που βρέθηκε κατά την αναζήτηση ελεύθερου χώρου αποθήκευσης

```
1920        sbi->max_cluster = total_clusters + FAT_START_ENT;
1921        counter+=1;
1922
1923        sprintf(temp, "%u: sb_info MAX_CLUSTER change to:  %u",counter, sbi->max_cluster);
1924        sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
1925        sys_fsync(sbi->arxeio_katagrafis);
1926        sys_fdatasync(sbi->arxeio_katagrafis);
1927        printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
```

sbi-> max_cluster αναφέρεται σε μια μεταβλητή που κατέχει τον μέγιστο αριθμό cluster στο αρχείο συστήματος FAT

```
buf->f_bfree = sbi->free_clusters;
counter+=1;
sprintf(temp, "%u: file system free blocks avail. f_bsize change to :  %u",counter,buf->f_bfree );
sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
```

buf->f_bfree αναπαριστά πληροφορίες για το αρχείο συστήματος, τελίκα κατά την εφαρμογή cptfs δεν χρησιμοποιούμε την λειτουργεια fat_statfs επομένως δεν γράφεται στο αρχείο journal σην συγκεκριμένη διαδικασία αντιγραφής

```
846
847         buf->f_bsize = sbi->cluster_size;
848         counter+=1;
849         sprintf(temp, "%u: file system blocksize f_bsize change to :  %u",counter,buf->f_bsize );
850         sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
851         sys_fsync(sbi->arxeio_katagrafis);
852         sys_fdatasync(sbi->arxeio_katagrafis);
853         printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
854
```

buf->f_bsize αντιπροσωπεύει το μέγεθος των μπλοκ (block size) στο αρχείο συστήματος

fatent.c

```
304 void fat_ent_access_init(struct super_block *sb)
305 {
306         struct msdos_sb_info *sbi = MSDOS_SB(sb);
307
308         mutex_init(&sbi->fat_lock);
309         char *temp = kzalloc(sizeof(MSDOS_SB(sb)), GFP_KERNEL);
310         switch (sbi->fat_bits) {
311         case 32:
312                 sbi->fatent_shift = 2;
313                 counterss+=1;
314                 printk(KERN_INFO"change made->");
315                 sprintf(temp, "%u: sb info fatent_shift change to :  %u",counterss, sbi->fatent_shift);
316                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
317                 sys_fsync(sbi->arxeio_katagrafis);
318                 sys_fdatasync(sbi->arxeio_katagrafis);
319                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
320
321                 sbi->fatent_ops = &fat32_ops;
322                 counterss+=1;
323                 printk(KERN_INFO"change made->");
324                 sprintf(temp, "%u: sb info fatent_ops change to :  %p",counterss, sbi->fatent_ops);
325                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
326                 sys_fsync(sbi->arxeio_katagrafis);
327                 sys_fdatasync(sbi->arxeio_katagrafis);
328                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
329
330                 break;
```

sbi->fatent_shift:κρατά τη μεταβλητή που καθορίζει τον αριθμό των bits που απαιτούνται για να αναπαρασταθεί ένας καταχωρητής στον πίνακα FAT.

sbi->fatent_ops: περιέχει δείκτες σε συναρτήσεις που υλοποιούν τις λειτουργίες που μπορούν να εκτελεστούν στους καταχωρητές του FAT.

Ομοια για περιπτώσεις 12 και 16 bit:

```
331         case 16:
332                 sbi->fatent_shift = 1;
333                 counterss+=1;
334                 printk(KERN_INFO"change made->");
335                 sprintf(temp, "%u: sb info fatent_shift change to :  %u",counterss, sbi->fatent_shift);
336                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
337                 sys_fsync(sbi->arxeio_katagrafis);
338                 sys_fdatasync(sbi->arxeio_katagrafis);
339                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
340
341                 sbi->fatent_ops = &fat16_ops;
342                 counterss+=1;
343                 printk(KERN_INFO"change made->");
344                 sprintf(temp, "%u: sb info fatent_ops change to :  %p",counterss, sbi->fatent_ops);
345                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
346                 sys_fsync(sbi->arxeio_katagrafis);
347                 sys_fdatasync(sbi->arxeio_katagrafis);
348                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
349
```

```
351         case 12:
352                 sbi->fatent_shift = -1;
353                 counterss+=1;
354                 printk(KERN_INFO"change made->");
355                 sprintf(temp, "%u: sb info fatent_shift change to :  %u",counterss, sbi->fatent_shift);
356                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
357                 sys_fsync(sbi->arxeio_katagrafis);
358                 sys_fdatasync(sbi->arxeio_katagrafis);
359                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
360
361                 sbi->fatent_ops = &fat12_ops;
362                 counterss+=1;
363                 printk(KERN_INFO"change made->");
364                 sprintf(temp, "%u: sb info fatent_ops change to :  %p",counterss, sbi->fatent_ops);
365                 sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
366                 sys_fsync(sbi->arxeio_katagrafis);
367                 sys_fdatasync(sbi->arxeio_katagrafis);
368                 printk(KERN_INFO "this is about to be written to journal: %s\n\n",temp);
369
370                 break;
371         }
372         kfree(temp);
373 }
```

Τέλος κρατήσαμε κάποιες αλλαγές και για το namei_vfat.c:

Για τον File Descriptor:

τελικά και εδώ κατα την εκτέλεση του cptfs.c δεν έγινε κάποια εγγραφή στο journal καθώς δεν χρησιμοποιήθηκε η συγκεκριμένη λειτουργία.
vfat_build_slots

```
652         /* build the entry of 8.3 alias name */
653         (*nr_slots)++;
654         memcpy(de->name, msdos_name, MSDOS_NAME);
655         sprintf(temp, "FILE DE change: de->name= %s\n", de->name); //grafoume to onoma
656         sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
657         sys_fsync(sbi->arxeio_katagrafis);
658         sys_fdatasync(sbi->arxeio_katagrafis);
659
660         de->attr = is_dir ? ATTR_DIR : ATTR_ARCH;
661         de->lcase = lcase;
662         fat_time_unix2fat(sbi, ts, &time, &date, &time_cs);
663         de->time = de->ctime = time;
664         de->date = de->cdate = de->adate = date;
665         de->ctime_cs = time_cs;
666         fat_set_start(de, cluster);
667         de->size = 0;
668         sprintf(temp, "FILE D change :de->size= %d",de->size); //grafoume to megethos
669         sys_write(sbi->arxeio_katagrafis, temp , sizeof(temp));
670         sys_fsync(sbi->arxeio_katagrafis);
671         sys_fdatasync(sbi->arxeio_katagrafis);
672
```

de->name :αναφέρεται στο πεδίο "name" του καταλόγου σε ένα αρχείο συστήματος

de.->size : αναφέρεται στο πεδίο "size" του καταλόγου σε ένα αρχείο συστήματος

Παρακάτω παραθέτουμε μια δοκιμή εκτέλεσης του κώδικα με την λειτουργία cptfs.

Έπειτα από κάθε εγγραφή στο αρχείο journal έχει προστεθεί μια printk() που εμφανίζει τον buffer : temp που χρησιμοποιήσαμε για την χρήση της sys_write προκειμένου να εμφανιστεί η αλλαγή που θελουμε να εγγραφεί στο αρχείο journal καθώς δεν μπορέσαμε να το εντοπίσουμε για να ελεγξουμε οτι όντως έχουν γίνει οι εγγραφές.

Παρακάτω παραθέτουμε ένα παράδειγμα του τελικού κωδικα με την εμφάνιση των αλλαγών που εγγραφησαν στο journal για την εκτέλεση της cptfs με τελευταίο όρισμα έναν κατάλογο που περιέχει 2 αρχειά:

```
    0.013176]
    0.013178] this is about to be written to journal: 2: sb_info cluster_bits change to:  11
    0.013178]
    0.013179] this is about to be written to journal: 3: sb_info fats change to:  2
    0.013179]
    0.013181] this is about to be written to journal:4: sb_info fat_bits change to:  0
    0.013181]
    0.013183] this is about to be written to journal: 5: sb_info fat_start change to:  4
    0.013183]
    0.013185] this is about to be written to journal: 6: sb_info fat_length change to:  200
    0.013185]
    0.013186] this is about to be written to journal: 7: sb_info root_cluster change to:  0
    0.013186]
    0.013188] this is about to be written to journal: 8: sb_info fats change to:  4294967295
    0.013188]
    0.013190] this is about to be written to journal: 9: sb_info free_clus_valid change to:  0
    0.013190]
    0.013191] this is about to be written to journal: 10: sb_info prev_free change to:  2
    0.013191]
    0.013203] this is about to be written to journal: 11: sb_info MAX_CLUSTER change to:  51093
    0.013203]
    0.013206] this is about to be written to journal: 12: sb_info free_clusters change to:  4294967295
    0.013206]
    0.013208] change made->
    0.013209] this is about to be written to journal: 1: sb info fatent_shift change to :  1
    0.013209]
    0.013212] change made->
    0.013214] this is about to be written to journal: 2: sb info fatent_ops change to :  000055b1e164be60
```

Εδώ παρατηρούμε ότι οι αλλαγές στα πεδία του superblock όντως έχουν καταγραφεί στο αρχείο.

Σημαντική βοήθεια για να εντοπίσουμε κάποιες λειτουργίες, αλλαγές που γίνονται καθώς και στην ευρύτερη κατανόηση της συνοχής του συστήματος αρχείων υπήρξε το lkl-doc που μας δόθηκε στην εκφώνηση.