3η Εργαστηριακή Άσκηση

Η εικόνα *fsdisk1.img*

1. Τροποποιήστε κατάλληλα το αρχείο utopia.sh ώστε να προσθέσετε στην εικονική μηχανή utopia έναν επιπλέον δίσκο για την εικόνα fsdisk1.img.

Ποια είναι η προσθήκη που κάνατε; Ποια συσκευή στο utopia είναι αυτή που μόλις προσθέσατε;

Προσθέτουμε στo utopia script τα virtual drives με την εξης εντολή:

-drive file=./fsdisk(1/2/3).img,if=virtio,format=raw

Ως αποτέλεσμα προστίθεται στο /dev ένα virtual device στην πρώτη διαθέσιμη vd θέση. Το vda είναι το αρχικό QCOW2\_PRIVATE\_FILE, οπότε οι νέες συσκευές ονομάζονται vdb,vdc,vdd αντίστοιχα.

2. Τι μέγεθος έχει ο δίσκος που προσθέσατε στο utopia;

**Α:** Με lsblk βρίσκουμε size = 50MiB

**Β:** hexdump -C -s 1024 -n 1024 /dev/vdb για να ανοίξουμε τα περιεχόμενα του superblock από όπου βλέπουμε ότι

log2 (block size) - 10 = 0 => block\_size = 1024

Total number of blocks = c800

Οπότε total\_size = #blocks\*block\_size = 50MiB

Βασιστήκαμε στο struct του Superblock όπως φαίνεται [εδώ](https://wiki.osdev.org/Ext2#What_is_a_Block?) και [εδώ](https://www.nongnu.org/ext2-doc/ext2.html#block-group-descriptor-table).

3. Τι σύστημα αρχείων περιέχει;

**Α:** Με blkid βρίσκουμε type=”ext2” (εναλλακτικά lsblk –f ή magic number)

**Β:** magic\_number = 0xef53 που αντιστοιχεί σε σύστημα ext2

4. Πότε ακριβώς δημιουργήθηκε αυτό το σύστημα αρχείων; Δείξτε τη χρονοσφραγίδα [timestamp].

**Α:** Με dumpe2fs: Last mount time: Tue Dec 12 17:23:16 2023

**Β:** Θα αξιοποιήσουμε το γεγονός ότι μαζί με την δημιουργία του ΣΑ, δημιουργήθηκε και το root inode, οπότε θα πάμε να βρούμε στα περιεχόμενα του το πεδίο create\_time.

Ξέρουμε ότι το root inode έχει inode\_number = 2, το οποίο σημαίνει ότι βρίσκεται στο BG #0.

Πηγαίνουμε στον αντίστοιχο Group Descriptor (που ξέρουμε ότι είναι στο 3ο μπλοκ και έχει μέγεθος 32) Από εδώ βρίσκουμε την πληροφορία ότι το inode table του BG #0 βρίσκεται στo block #5

Οπότε τώρα ξέρουμε ότι πρέπει να προσπελάσουμε το δεύτερο inode του block 5. (Τα inodes έχουνε μέγεθος 128 και η αρίθμηση τους ξεκινάει από το 1, οπότε το offset θα είναι 1 inode)

Τέλος θα διαβάσουμε τα byte 8-12 και θα βρούμε την τιμή 65787ae4 (posix) που μεταφράζεται σε

root@utopia:~# date -d @1702394596

Tue Dec 12 17:23:16 EET 2023

5. Πότε ακριβώς προσαρτήθηκε τελευταία φορά; Δείξτε τη χρονοσφραγίδα.

**Α:** Με dumpe2fs: Last mount time: Tue Dec 12 17:23:16 2023

**Β:** Από το superblock βρίσκουμε Tue Dec 12 17:23:16 EET 2023

6. Σε ποιο μονοπάτι προσαρτήθηκε τελευταία φορά;

**Α:** Με dumpe2fs: Last mounted on: /cslab-bunker

**Β:** Από το superblock /cslab-bunker

7. Πότε ακριβώς τροποποιήθηκε τελευταία φορά; Δείξτε τη χρονοσφραγίδα.

**Α:** Με dumpe2fs: Last write time: Tue Dec 12 17:23:17 2023

**Β:** Από το superblock Tue Dec 12 17:23:17 2023

8. Τι είναι το μπλοκ σε ένα σύστημα αρχείων;

Το ΣΑ ext2 διαμερίζει τον δίσκο σε λογικά και συνεχή κομμάτια τα οποία ονομάζονται block.

9. Τι μέγεθος μπλοκ [block size] χρησιμοποιεί αυτό το σύστημα αρχείων;

**Α:** Με dumpe2fs: Block size:               1024 (εναλλακτικά blkid)

**B:** Από το superblock: log2 (block size) - 10 = 0 => 1024

10. Τι είναι το inode σε ένα σύστημα αρχείων;

Το inode είναι μια δομή που αναπαριστά ένα αρχείο (file, δηλαδή τα πάντα) στον δίσκο. Δεν περιέχει τα δεδομένα, παρά μόνο link για την τοποθεσία των block που τα περιέχουν.

11. Τι μέγεθος έχει το inode σε αυτό το σύστημα αρχείων;

**Α:** Με dumpe2fs: Inode size:               128

**Β:** Από το superblock: Size of each inode structure in bytes = 0x80

12. Πόσα διαθέσιμα μπλοκ και πόσα διαθέσιμα inodes υπάρχουν σε αυτό το σύστημα αρχείων;

**Α:** Με dumpe2fs: Free blocks:              49552

Free inodes:              12810

**Β:** Από το superblock: Total number of unallocated blocks: c190 Total number of unallocated inodes: 320a

13. Τι είναι το superblock στο σύστημα αρχείων ext2;

To superblock περιέχει όλες τις πληροφορίες σχετικά με την οργάνωση του ΣΑ (πρακτικά διατηρεί όλα τα metadata του ΣΑ).

14. Πού βρίσκεται μέσα στον δίσκο σε ένα σύστημα αρχείων ext2;

Βρίσκεται στο δεύτερο block ολόκληρου του συστήματος (αμέσως μετά το boot partition).

15. Για ποιο λόγο έχει νόημα να υπάρχουν εφεδρικά αντίγραφα του superblock στο σύστημα αρχείων ext2;

Έτσι ώστε σε περίπτωση που έχουμε corruption του superblock να μην χρειάζεται format ολόκληρος ο δίσκος (κάτι το οποίο προφανώς οδηγεί σε απώλεια δεδομένων).

16. Σε ποια μπλοκ βρίσκονται αποθηκευμένα εφεδρικά αντίγραφα του superblock σε αυτό το σύστημα αρχείων;

**A:** 8193, 16385, 24577, 32769, 40961, 49153, δηλαδή στο 1ο block κάθε block group.

**Β:** Πηγαίνοντας στην αρχή του κάθε Block Group επαληθεύουμε, λόγω της παρουσίας του magic number, πως πράγματι κάθε πρώτο block αποτελεί ένα superblock

Αξίζει να σημειωθεί πως από το dumpe2fs φαίνεται πως το ΣΑ μας είναι το ext2-revision: 1 στο οποίο υπάρχει και η επιλογή του sparse superblock (0, 1 and powers of 3, 5 and 7), κάτι το οποίο όμως δεν έχει εφαρμοστεί εδώ.

17. Τι είναι ένα block group στο σύστημα αρχείων ext2;

Είναι απλά μια σειρά από blocks. Συνήθως το ΛΣ προσπαθεί να αποθηκεύει σχετικά μεταξύ τους δεδομένα στο ίδιο block group (locality) καθώς αυτό βοηθάει στην επίδοση του συστήματος. Περιέχει:

18. Πόσα block groups έχει ένα σύστημα αρχείων ext2 και πώς κατανέμονται;

Ο αριθμός των block group καθορίζεται από το

κάνοντας στρογγυλοποίηση προς τα πάνω. Τις πληροφορίες αυτές τις βρίσκουμε στο superblock.

19. Πόσα block groups περιέχει αυτό το σύστημα αρχείων;

20. Τι είναι ο block group descriptor στο σύστημα αρχείων ext2;

Ο block group descriptor στο σύστημα αρχείων ext2 είναι μια δομή δεδομένων που περιέχει μεταδεδομένα σχετικά με ένα συγκεκριμένο block group. Βοηθά το λειτουργικό σύστημα να εντοπίζει και να διαχειρίζεται τα διάφορα τμήματα ενός block group.

Ο block group descriptor περιλαμβάνει πληροφορίες όπως:

1. Block bitmap location: Την τοποθεσία (offset) του block bitmap, το οποίο δείχνει ποια blocks στο block group είναι κατειλημμένα ή διαθέσιμα.
2. Inode bitmap location: Την τοποθεσία του inode bitmap, το οποίο δείχνει ποιοι inodes είναι κατειλημμένοι ή διαθέσιμοι.
3. Inode table location: Την αρχική τοποθεσία του πίνακα των inodes, όπου βρίσκονται τα μεταδεδομένα των αρχείων.
4. Free blocks count: Τον αριθμό των ελεύθερων blocks στο block group.
5. Free inodes count: Τον αριθμό των ελεύθερων inodes στο block group.
6. Directories count: Τον αριθμό των καταλόγων (directories) που βρίσκονται στο block group.

21. Για ποιο λόγο έχει νόημα να υπάρχουν εφεδρικά αντίγραφα των block group descriptors στο σύστημα αρχείων ext2;

Η ύπαρξη εφεδρικών αντιγράφων των block group descriptors στο σύστημα αρχείων ext2 έχει νόημα για να διασφαλιστεί η αξιοπιστία και η ανάκτηση δεδομένων σε περίπτωση βλάβης ή καταστροφής μέρους του συστήματος αρχείων

22. Σε ποια μπλοκ βρίσκονται αποθηκευμένα εφεδρικά αντίγραφα των block group descriptors σε αυτό το σύστημα αρχείων;

Γενικά το Block Group Descriptor είναι πάντα το block μετά το superblock. Οπότε στην δική μας περίπτωση είναι:

8194, 16386, 24578, 32770, 40962, 49154

Κάνοντας hexdump σε αυτά τα μπλοκ βλέπουμε πως επαληθεύουν την δομή ενός GD

23. Τι είναι το block bitmap και τι το inode bitmap; Πού βρίσκονται μέσα στον δίσκο;

Το block bitmap είναι ένας πίνακας bit που δείχνει ποια blocks ενός block group είναι ελεύθερα (0) και ποια είναι καταχρησμένα (1). Χρησιμοποιείται από το σύστημα αρχείων για να εντοπίσει γρήγορα ελεύθερα blocks για εγγραφή δεδομένων.

Θέση:

Α: Με dumpe2fs: Block bitmap at 3 (+2), δηλαδή σε κάθε block group βρίσκεται στο 3ο block

B: Με hexdump -C -s $((2\*1024)) -n 32 /dev/vdb: Block address of block usage bitmap = 3

Το inode bitmap είναι ένας πίνακας bit που δείχνει ποια inodes ενός block group είναι ελεύθερα (0) και ποια είναι καταχρησμένα (1). Το σύστημα αρχείων το χρησιμοποιεί για την αποδοτική διαχείριση των inodes.

Θέση:

Α: Με dumpe2fs: Block bitmap at 4 (+3), δηλαδή σε κάθε block group βρίσκεται στο 4ο block

Β: Με hexdump -C -s $((2\*1024)) -n 32 /dev/vdb: Block address of inode usage bitmap = 4

25. Τι πεδία περιέχει το κάθε inode; Πού αποθηκεύεται μέσα στον δίσκο;

Τα βασικά πεδία της δομής inode που θα μας απασχολήσουν είναι:

• umode\_t i\_mode;

Τύπος αρχείου (character device, block device, regular file κλπ.) και δικαιώματα πρόσβασης (read, write, execute ως προς owner, group, others).

• kuid\_t i\_uid;

Αναγνωριστικό χρήστη.

• kgid\_t i\_gid;

Αναγνωριστικό group.

• const struct inode\_operations \*i\_op;

Ο δείκτης σε δομή με συλλογή (δεικτών) συναρτήσεων για λειτουργίες πάνω σε inode. Μέσω αυτής της δομής, μεταξύ άλλων, μπορεί το VFS να αναφέρεται στην υλοποίηση του εκάστοτε συστήματος αρχείου.

• struct super\_block \*i\_sb;

Δείκτης στο superblock με το οποίο συσχετίζεται το αρχείο.

• struct address\_space \*i\_mapping;

Δείκτης προς δομή ενδεικτική των σελίδων του inode στην page cache, και με ποιο τρόπο οφείλει να τις διαχειριστεί (μέσω της δομής struct address\_space\_operations, η οποία επίσης περιέχει δείκτες προς συναρτήσεις που υλοποιεί κάθε σύστημα αρχείων ξεχωριστά).

• const unsigned int i\_nlink;

Αριθμός από hard links προς το inode.

• atomic\_t i\_count

Usage counter για το inode. Δείχνει κοινώς πόσες διεργασίες χρησιμοποιούν το αρχείο αυτό μια δεδομένη στιγμή.

• unsigned long i\_ino;

Αναγνωριστικό του inode.

• const struct file\_operations \*i\_fop;

Δείκτης σε δομή file\_operations, η οποία έχει δείκτες συναρτήσεων που υλοποιούν βασικές λειτουργίες πάνω σε αρχεία, πχ. read, write, ioctl κλπ.

Τα inodes αποθηκεύονται στον πίνακα inodes (inode table), ο οποίος υπάρχει σε κάθε block group.

26. Πόσα μπλοκ και πόσα inodes περιέχει το κάθε block group σε αυτό το σύ-

στημα αρχείων;

**Α:** Με dumpe2fs: blocks per group:8192, inodes per group:1832

**Β:** Από το superblock: Number of blocks in each block group = 2000 Number of inodes in each block group = 0728

27. Σε ποιο inode αντιστοιχεί το αρχείο /dir2/helloworld σε αυτό το σύστημα

αρχείων;

**A:** Με stat: 9162

**B:** Αρχικά πρέπει να βρούμε το directory entry block του “/” (root directory)

Ξέρουμε ότι το root inode έχει inode\_number = 2, το οποίο σημαίνει ότι βρίσκεται στο BG #0.

Πηγαίνουμε στον αντίστοιχο Group Descriptor (που ξέρουμε ότι είναι στο 3ο μπλοκ και έχει μέγεθος 32) hexdump -C -s 2048 -n 32 /dev/vdb. Από εδώ βρίσκουμε την πληροφορία ότι το inode table του BG #0 βρίσκεται στo block #5 Οπότε τώρα ξέρουμε ότι πρέπει να προσπελάσουμε το δεύτερο inode του block 5.

Εδώ κοιτάμε το byte #40 (Direct Block Pointer #0, δηλαδή ποιο μπλοκ περιέχει τα περιεχόμενα του directory που εξετάζουμε) και βρίσκουμε: ea=234

Οπότε, τώρα, θα πάμε να δούμε τα περιεχόμενα του block #234 (δηλαδή τα περιεχόμενα του “/”) και βασιζόμενοι στην δομή ενός directory entry βρίσκουμε ότι το inode του dir2 = 23c9 = 9161

Οπότε, = 5, άρα ψάχνουμε στο 5ο BGT. Δηλαδή, πάμε στο BGD (block #3) και βάζουμε ένα offset = 5\*GD\_SIZE

Έτσι, βρίσκουμε πως το inode table του BG #5 βρίσκεται στο block #a005 = 40965

Δηλαδή, βρήκαμε πως στο block #247 περιέχονται τα δεδομένα του dir2.

Τέλος, με hexdump -C -s $((247\*1024)) -n 1024 /dev/vdb βρίσκουμε το αρχείο helloworld και βασιζόμενοι στην τυπική δομή βρίσκουμε inode = 23ca = 9162

28. Σε ποιο block group αντιστοιχεί αυτό το inode;

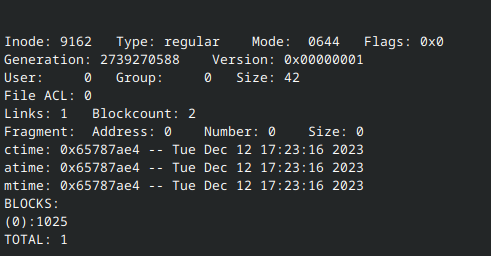
29. Σε ποιο μπλοκ του δίσκου υπάρχει το inode table που περιέχει το παραπάνω inode;

**A:** Από dumpe2fs βρίσκουμε Inode table at 40965-41193 (+4) και εφόσον θέλουμε το 2ο inode του Group θα βρίσκεται στο 40965

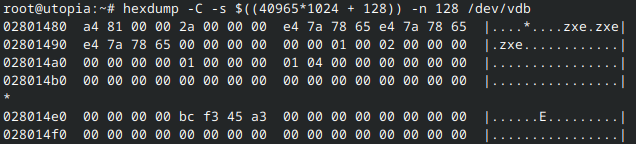
**Β:** Βρήκαμε προηγουμένως πως είναι το 40965

30. Δείξτε όλα τα πεδία αυτού του inode [128 bytes].

**Α:** Με debugfs και stat



B: hexdump -C -s $((40965\*1024 + 128)) -n 128 /dev/vdb



31. Σε ποιο μπλοκ είναι αποθηκευμένα τα δεδομένα αυτού του αρχείου;

**A:** Βρήκαμε προηγουμένως ότι είναι το 1025

**Β:** Από την εικόνα: Byte #40 = 0401 = 1025

32. Τι μέγεθος έχει αυτό το αρχείο;

**Α:** Από το stat φαίνεται 42B

**Β:** Από την εικόνα: size = 2a = 42B

33. Δείξτε τα περιεχόμενα αυτού του αρχείου.

**Α:** Με debugfs και cat ./dir2/helloworld: Welcome to the Mighty World of filesystems

**B:** hexdump -C -s $((1025\*1024)) -n 1024 /dev/vdb

# Η εικόνα *fsdisk2.img*

3. Πέτυχε η εντολή; Αν όχι, τι πρόβλημα υπήρξε;

Η εντολή απέτυχε λόγω έλλειψης χώρου

4. Ποια κλήση συστήματος προσπάθησε να τρέξει η touch, και με ποιον κωδικό λάθους απέτυχε; Υποστηρίξτε την απάντησή σας με χρήση της εντολής strace.

Καλείται η openat η οποία επιστρέφει ENOSPC

5. Πόσα αρχεία και πόσους καταλόγους περιέχει το συγκεκριμένο σύστημα αρχείων;

**A:** Με dumpe2fs βρίσκουμε τα directories στο section του εκάστοτε Group και αθροίζουμε συνολικά 259. Για τα αρχεία

**Β:** Με hexdump εξάγουμε τις ίδιες πληροφορίες από τα Group και το superblock

6. Πόσο χώρο καταλαμβάνουν τα δεδομένα και τα μεταδεδομένα του συγκεκριμένου συστήματος αρχείων;

**Α:** Για τα metadata Άρα, το μέγεθος θα είναι .

Για τα data

(Εναλλακτικά με df)

**Β:** Αρχικά, με hexdump το superblock υπολογίζω τον αριθμό group = 3. Το μέγεθος ενός GD είναι 32 bytes, οπότε και οι 3 χωράνε σε ένα μόνο block. O μέγιστος αριθμός block που μπορεί να απεικονίσει ένα block bitmap είναι =8096 στην δική μας περίπτωση και επίσης blocks per group = 8096. Άρα, και το block bitmap θα καταλαμβάνει ένα block ανά group. Τέλος, ξέρουμε πως δεν υπάρχουν διαθέσιμα inodes (από το sb) και inodes per group = 1712 \* 128 (inode size) και καταλήγουμε στο ίδιο αποτέλεσμα με πριν.

Παρομοίως και για τα data.

7. Πόσο είναι το μέγεθος του συγκεκριμένου συστήματος αρχείων;

Και στις δύο περιπτώσεις από το superblock

8. Πόσα μπλοκ είναι διαθέσιμα/ελεύθερα στο συγκεκριμένο σύστημα αρχείων;

Ισοδύναμα, έχει ελεύθερο χώρο το συγκεκριμένο σύστημα αρχείων;

Και στις δύο περιπτώσεις, από το superblock .

Οπότε, ναι, έχει διαθέσιμο χώρο.

9. Αφού υπάρχουν διαθέσιμα μπλοκ, τι σας αποτρέπει από το να δημιουργήσετε νέο αρχείο;

Δεν υπάρχει διαθέσιμο inode να ανατεθεί στο καινούργιο αρχείο.

# Η εικόνα *fsdisk3.img*

1. Ποιο εργαλείο στο Linux αναλαμβάνει τον έλεγχο ενός συστήματος αρχείων ext2 για αλλοιώσεις;

Το εργαλείο που αναλαμβάνει τον έλεγχο ενός συστήματος αρχείων ext2 για αλλοιώσεις στο Linux είναι το e2fsck

2. Ποιοι παράγοντες θα μπορούσαν δυνητικά να οδηγήσουν σε αλλοιώσεις στο σύστημα αρχείων; Αναφέρετε ενδεικτικά δέκα πιθανές αλλοιώσεις.

1. **Απώλεια δεδομένων λόγω μη ομαλής διακοπής ρεύματος**: Μια ξαφνική διακοπή ρεύματος μπορεί να οδηγήσει σε ατελείς εγγραφές στο σύστημα αρχείων.
2. **Κατεστραμμένα inode**: Τα inodes που περιγράφουν τα αρχεία και τους φακέλους μπορεί να καταστραφούν λόγω αστοχίας δίσκου ή σφάλματος λογισμικού.
3. **Λανθασμένοι δείκτες στα μπλοκ δεδομένων**: Ένα αρχείο μπορεί να δείχνει σε λάθος ή μη διαθέσιμα μπλοκ δεδομένων, προκαλώντας απώλεια ή αλλοίωση δεδομένων.
4. **Διπλή ανάθεση μπλοκ δεδομένων**: Το ίδιο μπλοκ μπορεί να ανατεθεί σε δύο διαφορετικά αρχεία, οδηγώντας σε απρόβλεπτη συμπεριφορά.
5. **Κατεστραμμένη δομή καταλόγων**: Λάθη στη δομή καταλόγων μπορεί να κάνουν ορισμένα αρχεία μη ανιχνεύσιμα ή μη προσβάσιμα.
6. **Σφάλματα κατά τη διαδικασία συγχρονισμού (fsync)**: Αν το σύστημα αποτύχει να συγχρονίσει τις αλλαγές στο δίσκο, τα δεδομένα μπορεί να είναι ασυνεπή.
7. **Καταστροφή του superblock**: Το superblock περιέχει κρίσιμες πληροφορίες για το σύστημα αρχείων (π.χ. τύπο, μέγεθος, δομή) και η αλλοίωσή του μπορεί να καταστήσει το σύστημα μη προσβάσιμο.
8. **Κατεστραμμένα αρχεία ημερολογίου (journaling)**: Στα συστήματα αρχείων με journaling, αν το ημερολόγιο είναι κατεστραμμένο, οι λειτουργίες rollback/recovery μπορεί να αποτύχουν.
9. **Σφάλματα χρήστη (user errors)**: Διαγραφή αρχείων ή καταλόγων κατά λάθος, ιδιαίτερα χωρίς επαρκή δικαιώματα ή αντίγραφα ασφαλείας.
10. **Επιθέσεις κακόβουλου λογισμικού (malware)**: Ένα κακόβουλο λογισμικό μπορεί να αλλοιώσει ή να καταστρέψει δεδομένα στο σύστημα αρχείων.

3. Τρέξτε το εργαλείο αυτό και επιδιορθώστε το σύστημα αρχείων. Αναφέρετε όλες τις αλλοιώσεις που εντοπίσατε, εξαντλητικά.

1. Pass 2: Checking directory structure First entry 'BOO' (inode=1717) in directory inode 1717 (/dir-2) should be '.'

Σε κάθε directory η πρώτη καταχώριση πρέπει να είναι ‘.’, δηλαδή αναφορά στον εαυτό του, κάτι το οποίο δεν ίσχυε.

1. Checking reference counts Inode 3425 ref count is 1, should be 2.

Εδώ το εργαλείο διαπίστωσε πως τα hard links που είναι καταγεγραμμένα στο inode του αρχείου (1) διαφέρουν από τις πραγματικές αναφορές που γίνονται. Δηλαδή εντόπισε δύο άλλα αρχεία που το κάνουν reference στην πραγματικότητα.

1. Checking group summary information Block bitmap differences: +34

Το bitmap είχε λάθος σημειωμένο ένα block (το 34)

1. Free blocks count wrong (926431538, counted=19800).

Τα metadata του superblock είχαν λάθος καταχώριση

4. Επαναφέρετε το δίσκο στην πρότερή του κατάσταση, από την αρχική εικόνα.

Εντοπίστε τις αλλοιώσεις με χρήση της μεθόδου hexedit.

Έχοντας κρατήσει ένα backup του «καθαρού» δίσκου, δημιουργούμε (μέσω xxd) δύο dumps, τα οποία ύστερα συγκρίνουμε μέσω diff για να εντοπίσουμε τις διαφορές.

1. Στα byte 40C-410, δηλαδή 1036-1040 τα οποία εντοπίζονται στο superblock. Συγκεκριμένα, είναι τα byte 12-16 του sb, τα οποία περιέχουν την πληροφορία σχετικά με τα free blocks. Αξίζει να σημειωθεί πως εδώ ακόμα και αφού αλλάξαμε την τιμή ώστε να ισούται με αυτήν της «καθαρής» εικόνας, πάλι υπήρχε διαφορά κατά 1, το οποίο οφείλεται στην λάθος καταγραφή του bitmap.
2. Στο 430, δηλαδή το 48ο byte του sb, το οποίο αντιστοιχεί στο πεδίο last written time, οπότε δεν το πειράζουμε. Αντίστοιχη διαφορά και στο 64ο, last mount time.
3. Στο 570, δηλαδή στο byte 368 του sb, το οποίο σύμφωνα με την δομή του, είναι unused. Οπότε δεν αλλάζουμε κάτι
4. Στο C00, δηλαδή 3072 όπου πλέον βρίσκεται το bitmap. Υπάρχει ένα D το οποίο εμείς θέλουμε να είναι F (δηλαδή στο bitmap είναι καταγεγραμμένο ως άδειο ενώ είναι γεμάτο). Αυτόματα διορθώθηκαν και άλλες διαφορές με αυτό το βήμα.
5. Στο 83700, καταχώριση του directory που είχε το ‘BOO’, την αλλάξαμε σε ‘.’
6. Στο 1001410, τα hard links, από 1 τα κάναμε 2