

Εργαστήριο Λειτουργικών Συστημάτων

Εργαστηριακή Άσκηση 3

ΝΤΟΝΤΟΡΟΣ ΗΛΙΑΣ

el19206

ΑΣΚΗΣΗ 1

1. Η προσθήκη που έκανα ήταν η παρακάτω γραμμή στο utopia.sh και στο utopia εμφανίζεται ως η συσκευή vdb στον φάκελο /dev.

```
-drive file="./fsdisk1-c8e29319b.img",format=raw,if=virtio \
```

2. Εκτελώντας την εντολή lsblk μπορούμε να δούμε ότι ο δίσκος vdb έχει μέγεθος 52428800 bytes, δηλαδή 50 megabytes.

```
vdb    254:16    0    52428800    0 disk
```

Επίσης με την εντολή hexdump -s 1024 -n 1024 -C /dev/vdb μπορούμε να δούμε τα περιεχόμενα του superbloc όπου το block count είναι 51200 και το block size είναι 1024 άρα το συνολικό μέγεθος είναι 52428800 bytes.

```
root@utopia:/dev# hexdump -s 1024 -n 1024 -C /dev/vdb
00000400  18 32 00 00 00 c8 00 00 00 0a 00 00 90 c1 00 00 |.2.....|
00000410  0a 32 00 00 01 00 00 00 00 00 00 00 00 00 00 00 |.2.....|
00000420  00 20 00 00 00 20 00 00 28 07 00 00 6d 73 9c 63 |. . . .(.ms.c|
00000430  6d 73 9c 63 01 00 ff ff 53 ef 01 00 01 00 00 00 |ms.c....S.....|
00000440  6d 73 9c 63 00 00 00 00 00 00 00 00 01 00 00 00 |ms.c.....|
00000450  00 00 00 00 0b 00 00 00 80 00 00 00 00 00 00 00 |.....|
00000460  00 00 00 00 00 00 00 00 22 6f d1 2b 12 52 40 58 |....."o.+R@X|
00000470  9f 5f 48 e0 ef f7 41 f0 66 73 64 69 73 6b 31 2e |. _H...A.fsdisk1.|
00000480  69 6d 67 00 00 00 00 00 2f 68 6f 6d 65 2f 6a 69 |img..../home/ji|
00000490  6d 73 69 61 6b 2f 63 73 6c 61 62 2d 67 69 74 2f |mslak/cslab-git/|
000004a0  66 73 2f 6d 6e 74 00 00 00 00 00 00 00 00 00 00 |fs/mnt.....|
000004b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000004e0  00 00 00 00 00 00 00 00 00 00 00 00 ba 15 e3 b8 |.....|
000004f0  3d 92 4b 5e a6 29 8f 95 29 d2 28 0a 01 00 00 00 |=.K^.)..).(....|
00000500  0c 00 00 00 00 00 00 00 6d 73 9c 63 00 00 00 00 |.....ms.c....|
00000510  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000560  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000570  00 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 |.....|
00000580  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000800
```

3. Εκτελώντας την εντολή lsblk -f μπορούμε να δούμε ότι περιέχει σύστημα αρχείων ext2. Επίσης από το superbloc μπορούμε να βρούμε στο byte 56-57 μπορούμε να δούμε ότι υπάρχει το id για το ext2: ef53
4. Με την εντολή dumpe2fs vdb μπορούμε να δούμε το Filesystem created

```
Filesystem created:      Fri Dec 16 15:32:29 2022
Last mount time:         Fri Dec 16 15:32:29 2022
Last write time:         Fri Dec 16 15:32:29 2022
```

MISSING HEXDUMP

5. Με την παραπάνω εντολή φαίνεται το Last mount time. Επίσης στο superbloc στα byte 44-47 φαίνεται η τιμή 6399c736d η οποία είναι σε posix time και αν την μετατρέψουμε σε κανονική είναι η ίδια με την παραπάνω.

6. Με την εντολή `dumpe2fs vdb` μπορούμε να δούμε το path που προσαρτήθηκε τελευταία φορά:
- ```
Last mounted on: /home/jimslak/cslab-git/fs/mnt
```
- ενώ το ίδιο φαίνεται στο superblock στα byte 136-199.
7. Με την εντολή `dumpe2fs vdb` μπορούμε να δούμε το Last write time όπως φαίνεται και στην εικόνα της ερώτησης 4. Επίσης στο superblock στα byte 48-51 είναι φαίνεται η ημερομηνία που τροποποιήθηκε τελευταία φορά σε posix time.
8. Το ext2 χωρίζει τον χώρο του δίσκου σε λογικά κομμάτια συνεχόμενου χώρου σταθερού μήκους. Κάθε τέτοιο κομμάτι ονομάζεται Block. [\(πηγή\)](#)
9. Με την εντολή `dumpe2fs vdb` μπορούμε να δούμε το Block size :
- ```
Block size: 1024
```
- Επίσης στο superblock στα byte 24-27 βλέπουμε ότι η τιμή είναι 00000000 δηλαδή πρέπει να κάνουμε shift left 0 φορές το 1024 για να βρούμε το block size άρα τελικά το block size είναι 1024 bytes.
10. Το inode είναι ένα structure που αντιπροσωπεύει ένα αρχείο ή ένα φάκελο ή ένα symbolic link. Τα inodes δεν περιέχουν δεδομένα του στοιχείου που αντιπροσωπεύουν αλλά είναι σύνδεσμοι για το block που περιέχει το στοιχείο που αντιπροσωπεύουν και έτσι με αυτόν τον τρόπο μπορούν να έχουν σταθερό μέγεθος και για κάθε block group να είναι αποθηκευμένα σε ένα indexed array για γρήγορη πρόσβαση. [\(πηγή\)](#)
11. Με την εντολή `dumpe2fs vdb` εμφανίζεται το inode size:
- ```
Inode size: 128
```
- Επίσης, στο superblock στα byte 88-89 φαίνεται ότι το μέγεθος του inode είναι 0x80 δηλαδή 128 bytes.
12. Με την εντολή `dumpe2fs vdb` εμφανίζονται τα Free inodes και free blocks:
- ```
Free blocks: 49552
Free inodes: 12810
```
- Επίσης στο superblock στα byte 12-15 εμφανίζεται ο αριθμός των ελεύθερων block (0xc190=49552) και στα byte 16-19 εμφανίζεται ο αριθμός των ελεύθερων inodes (0x320a=12810).
13. Το superblock είναι ένα block το οποίο περιέχει πληροφορίες για το σύστημα αρχείων όπως τα παραπάνω και διάφορες προαιρετικές λειτουργίες που υπάρχουν.
14. Το superblock βρίσκεται πάντα στο byte 1024 και έχει μέγεθος 1024 bytes.
15. Γιατί το superblock είναι πολύ σημαντικό και χωρίς αυτό δεν μπορεί να γίνει προσάρτηση του συστήματος αρχείων. Με τα εφεδρικά αντίγραφα μπορεί εύκολα να γίνει επαναφορά του χωρίς την απώλεια δεδομένων.
16. Αντίγραφα του superblock υπάρχουν στην αρχή κάθε block group όπως σε κάθε σύστημα αρχείων ext2 (revision 0). Μπορούμε να το επιβεβαιώσουμε με την εντολή `dumpe2fs /dev/vdb | grep super :`

```

root@utopia:/dev# dumpe2fs vdb | grep super
dumpe2fs 1.46.2 (28-Feb-2021)
Primary superblock at 1, Group descriptors at 2-2
Backup superblock at 8193, Group descriptors at 8194-8194
Backup superblock at 16385, Group descriptors at 16386-16386
Backup superblock at 24577, Group descriptors at 24578-24578
Backup superblock at 32769, Group descriptors at 32770-32770
Backup superblock at 40961, Group descriptors at 40962-40962
Backup superblock at 49153, Group descriptors at 49154-49154

```

ή με την εντολή `hexdump -s $((8193*1024)) -n 1024 -C /dev/vdb` όπου εμφανίζεται το ίδιο superblock στο block 8193:

```

root@utopia:/dev# hexdump -s $(( 8193*1024)) -n 1024 -C /dev/vdb
00800400 18 32 00 00 00 c8 00 00 00 0a 00 00 93 c1 00 00 |.2.....|
00800410 0d 32 00 00 01 00 00 00 00 00 00 00 00 00 00 00 |.2.....|
00800420 00 20 00 00 00 20 00 00 28 07 00 00 00 00 00 00 |. ... ..(.....|
00800430 6d 73 9c 63 00 00 ff ff 53 ef 00 00 01 00 00 00 |ms.c....S.....|
00800440 6d 73 9c 63 00 00 00 00 00 00 00 00 01 00 00 00 |ms.C.....|
00800450 00 00 00 00 0b 00 00 00 80 00 01 00 00 00 00 00 |.....|
00800460 00 00 00 00 00 00 00 00 22 6f d1 2b 12 52 40 58 |....."o+.R@X|
00800470 9f 5f 48 e0 ef f7 41 f0 66 73 64 69 73 6b 31 2e |. _H...A.fsdisk1.|
00800480 69 6d 67 00 00 00 00 00 2f 63 73 6c 61 62 2d 62 |img..../cslab-b|
00800490 75 6e 6b 65 72 00 00 00 00 00 00 00 00 00 00 |unker.....|
008004a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
008004e0 00 00 00 00 00 00 00 00 00 00 00 00 ba 15 e3 b8 |.....|
008004f0 3d 92 4b 5e a6 29 8f 95 29 d2 28 0a 01 00 00 00 ||=K^.)..).(.....|
00800500 0c 00 00 00 00 00 00 00 6d 73 9c 63 00 00 00 00 |.....ms.C....|
00800510 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00800560 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00800570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*

```

17. Τα block groups είναι ομάδες συνεχόμενων blocks και inodes τα οποία ομαδοποιούνται για να ελαττωθεί το fragmentation και να υπάρχει πιο γρήγορη πρόσβαση σε πολλά συνεχόμενα δεδομένα. ([πηγή](#))
18. Το μέγιστο μέγεθος κάθε block group είναι 8 φορές το μέγεθος του block καθώς κάθε block group έχει ένα block usage bitmap μεγέθους 1 block. Άρα ο αριθμός των block groups σε ένα σύστημα αρχείων είναι όσο το μέγεθος του συστήματος αρχείων σε blocks διά το μέγεθος του block group στρογγυλοποιημένο προς τα πάνω. ([πηγή](#))
19. Για να βρούμε τον αριθμό των block groups μπορούμε να πάρουμε το μέγεθος του συστήματος αρχείων και να το διαιρέσουμε με το block size και να βρούμε το block count (51200) (μπορούμε να το βρούμε και πιο γρήγορα με την εντολή `dumpe2fs vdb` ή διαβάζοντας το superblock). Έπειτα παίρνουμε το αποτέλεσμα και το πολλαπλασιάζουμε το 8 και έχουμε το block group size (8192). Τέλος διαιρούμε το block count με το block group size και το στρογγυλοποιούμε προς τα πάνω. Άρα στο συγκεκριμένο σύστημα αρχείων έχουμε 7 block groups.
20. Ο block group descriptor βρίσκεται μέσα στο block group descriptor table και περιέχει πληροφορίες για κάθε block group που αντιπροσωπεύει, όπως για παράδειγμα την διεύθυνση του block usage bitmap. ([πηγή](#))
21. Όπως και το superblock κάθε block group descriptor είναι πολύ σημαντικός για το σύστημα αρχείων και διατηρούνται αντίγραφα σε κάθε block group (αμέσως μετά το αντίγραφο του superblock) σε περίπτωση που υπάρχει αλλοίωση του πρωτότυπου. ([πηγή](#))
22. Στο συγκεκριμένο σύστημα αρχείων υπάρχει αντίγραφο του block group descriptor table στην αρχή του κάθε block group αμέσως μετά το αντίγραφο του superblock. Αυτό μπορούμε να το δούμε και με την

εντολή `dump2efs vdb` και αν εμφανίσουμε το δεύτερο block του κάθε block group με την εντολή `hexdump`.

23. Το block bitmap και το inode bitmap αντίστοιχα είναι blocks που βρίσκονται στην αρχή κάθε block group μετά το block group descriptor table τα οποία αποτελούνται από μία αλληλουχία από bits και κάθε ένα bit αντιπροσωπεύει ένα block (inode) και η τιμή του bit δηλώνει ότι το block (inode) είναι ελεύθερο (για τιμή 0) ή δεσμευμένο (για τιμή 1).([πηγή](#))
24. Τα inode tables είναι μια σειρά από blocks που βρίσκονται σε κάθε block group μετά από το inode bitmap και περιέχουν inodes για τα blocks που υπάρχουν μέσα στο block group.([πηγή](#))
25. Τα πεδία του inode φαίνονται στον παρακάτω πίνακα και κάθε inode βρίσκεται στο inode table του block group στο οποίο βρίσκεται το block που αντιπροσωπεύει.([πηγή](#))

Starting Byte	Ending Byte	Size in Bytes	Field Description
0	1	2	Type and Permissions (see below)
2	3	2	User ID
4	7	4	Lower 32 bits of size in bytes
8	11	4	Last Access Time (in POSIX time ↗)
12	15	4	Creation Time (in POSIX time ↗)
16	19	4	Last Modification time (in POSIX time ↗)
20	23	4	Deletion time (in POSIX time ↗)
24	25	2	Group ID
26	27	2	Count of hard links (directory entries) to this inode. When this reaches 0, the data blocks are marked as unallocated.
28	31	4	Count of disk sectors (not Ext2 blocks) in use by this inode, not counting the actual inode structure nor directory entries linking to the inode.
32	35	4	Flags (see below)
36	39	4	Operating System Specific value #1
40	43	4	Direct Block Pointer 0
44	47	4	Direct Block Pointer 1
48	51	4	Direct Block Pointer 2
52	55	4	Direct Block Pointer 3
56	59	4	Direct Block Pointer 4
60	63	4	Direct Block Pointer 5
64	67	4	Direct Block Pointer 6
68	71	4	Direct Block Pointer 7
72	75	4	Direct Block Pointer 8
76	79	4	Direct Block Pointer 9
80	83	4	Direct Block Pointer 10
84	87	4	Direct Block Pointer 11
88	91	4	Singly Indirect Block Pointer (Points to a block that is a list of block pointers to data)
92	95	4	Doubly Indirect Block Pointer (Points to a block that is a list of block pointers to Singly Indirect Blocks)
96	99	4	Triply Indirect Block Pointer (Points to a block that is a list of block pointers to Doubly Indirect Blocks)
100	103	4	Generation number (Primarily used for NFS)
104	107	4	In Ext2 version 0, this field is reserved. In version >= 1, Extended attribute block (File ACL).
108	111	4	In Ext2 version 0, this field is reserved. In version >= 1, Upper 32 bits of file size (if feature bit set) if it's a file, Directory ACL if it's a directory
112	115	4	Block address of fragment
116	127	12	Operating System Specific Value #2

26. Με την εντολή `dumpe2fs vdb` μπορούμε να δούμε το blocks per group και το inodes per group:

```
Blocks per group:      8192
Fragments per group:   8192
Inodes per group:      1832
```

Επίσης η ίδια πληροφορία βρίσκεται και στο superblock στα byte 32-32 (blocks) και 40-43 (inodes).

27. Για να βρούμε το inode του συγκεκριμένου αρχείου μπορούμε να προσαρτήσουμε το σύστημα αρχείων με την εντολή `mount /dev/vdb /mnt` και μετά να εκτελέσουμε την εντολή `stat /mnt/dir2/helloworld` όπου θα

εμφανίσει το inode του συγκεκριμένου αρχείου:

```
root@utopia:/mnt/dir2# stat /mnt/dir2/helloworld
File: /mnt/dir2/helloworld
Size: 42          Blocks: 2          IO Block: 1024   regular file
Device: fe10h/65040d Inode: 3666          Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2023-01-15 17:36:50.000000000 +0200
Modify: 2022-12-16 15:32:29.000000000 +0200
Change: 2022-12-16 15:32:29.000000000 +0200
Birth: -
```

Επίσης μπορούμε να μπορούμε να βρούμε το inode ακολουθώντας τα εξής βήματα: Βρίσκουμε το inode table του block group 0 και στα byte 8-11 βλέπουμε την τιμή 0x00000005 που αυτή είναι ο αριθμός του πρώτου block του inode table του block group 0.

```
root@utopia:/mnt/dir2# hexdump -s $((1024*2)) -n 32 -C /dev/vdb
00000800  03 00 00 00 04 00 00 00 05 00 00 00 0a 1f 1d 07 |.....|
00000810  02 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000820
```

Το inode του root directory βρίσκεται πάντα στο inode #2 άρα στα byte 128-256 του block #5 όπου βρίσκουμε το directory entry block του root directory ("/") είναι το block #234.

```
root@utopia:/mnt/dir2# hexdump -s $((1024*5+128)) -n 128 -C /dev/vdb
00001480  ed 41 00 00 00 04 00 00 83 1d c4 63 6d 73 9c 63 |.A.....cms.c|
00001490  6d 73 9c 63 00 00 00 00 00 00 05 00 02 00 00 00 |ms.c.....|
000014a0  00 00 00 00 02 00 00 00 ea 00 00 00 00 00 00 00 |.....|
000014b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
```

Διαβάζοντας τα περιεχόμενα του block #234 βλέπουμε τα περιεχόμενα του root directory και βρίσκουμε ότι το inode του directory dir2 είναι το inode #3665 το οποίο είναι το 1^ο inode στο block group #2

```
root@utopia:/mnt/dir2# hexdump -s $((1024*234)) -n 1024 -C /dev/vdb
0003a800  02 00 00 00 0c 00 01 00 2e 00 00 00 02 00 00 00 |.....|
0003a810  0c 00 02 00 2e 2e 00 00 0b 00 00 00 14 00 0a 00 |.....|
0003a820  6c 6f 73 74 2b 66 6f 75 6e 64 00 00 c9 23 00 00 |lost+found...#..|
0003a830  0c 00 04 00 64 69 72 31 51 0e 00 00 c8 03 04 00 |...dir1Q.....|
0003a840  64 69 72 32 00 00 00 00 00 00 00 00 00 00 00 00 |dir2.....|
0003a850  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
```

Από το block group descriptor βρίσκουμε το πρώτο block του inode table του block group #2 το οποίο είναι το block #16389

```
root@utopia:/mnt/dir2# hexdump -s $((1024*2+64)) -n 32 -C /dev/vdb
00000840  03 40 00 00 04 40 00 00 05 40 00 00 17 1f 26 07 |.@...@...@...&.|
00000850  01 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000860
```

Διαβάζουμε το πρώτο inode του block group #2 στο block #16389 και βρίσκουμε το directory entry block του directory /dir2 το οποίο είναι το block #33002

```
root@utopia:/mnt/dir2# hexdump -s $((1024*16389)) -n 128 -C /dev/vdb
01001400  ed 41 00 00 00 04 00 00 8a 1d c4 63 6d 73 9c 63 |.A.....cms.c|
01001410  6d 73 9c 63 00 00 00 00 00 00 02 00 02 00 00 00 |ms.c.....|
01001420  00 00 00 00 02 00 00 00 ea 80 00 00 00 00 00 00 |.....|
01001430  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
```

Διαβάζοντας το block #33002 βρίσκουμε ότι το αρχείο helloworld έχει inode το inode #3666.

```
oot@utopia:/mnt/dir2# hexdump -s $((1024*33002)) -n 1024 -C /dev/vdb
203a800  51 0e 00 00 0c 00 01 00 2e 00 00 00 02 00 00 00 |Q.....|
203a810  0c 00 02 00 2e 2e 00 00 52 0e 00 00 e8 03 0a 00 |.....R.....|
203a820  68 65 6c 6c 6f 77 6f 72 6c 64 00 00 00 00 00 00 |helloworld.....|
203a830  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
203ac00
```


28. Κάθε block group έχει 1832 inodes άρα το inode #3666 είναι το 2° inode του block group #2
29. Το inode table του block group #2 μπορούμε να το δούμε με την εντολή `dumpe2fs /dev/vdb` ή όπως το βρήκαμε παραπάνω στο block #16389.
30. Μπορούμε να δούμε τα πεδία του inode #3666 με την εντολή `debugfs -R 'stat <3666>' /dev/vdb`

```
Inode: 3666   Type: regular   Mode:  0644   Flags: 0x0
Generation: 969212841   Version: 0x00000001
User:      0   Group:      0   Size: 42
File ACL: 0
Links: 1   Blockcount: 2
Fragment: Address: 0   Number: 0   Size: 0
ctime: 0x639c736d -- Fri Dec 16 15:32:29 2022
atime: 0x63c41d92 -- Sun Jan 15 17:36:50 2023
mtime: 0x639c736d -- Fri Dec 16 15:32:29 2022
BLOCKS:
(0):33793
TOTAL: 1
```

Ή με την εντολή `hexdump -s $((1024*16389)) -n 128 /dev/vdb`

```
root@utopia:/mnt/dir2# hexdump -s $((1024*16389+128)) -n 128 -C /dev/vdb
01001480 a4 81 00 00 2a 00 00 00 92 1d c4 63 6d 73 9c 63 |....*.....cms.c|
01001490 6d 73 9c 63 00 00 00 00 00 00 01 00 02 00 00 00 |ms.C.....|
010014a0 00 00 00 00 01 00 00 00 01 84 00 00 00 00 00 00 |.....|
010014b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
010014e0 00 00 00 00 a9 03 c5 39 00 00 00 00 00 00 00 00 |.....9.....|
010014f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
01001500
```

Type and permissions : 0x81a4

User ID : 0x0000

Lower 32 bits of size in bytes :0x0000002a

Last access time : 0x63c41d92

Creation time : 0x639c736d

Last modification time : 0x639c736d

Deletion time : 0x00000000

Group id : 0x0000

Count of hard links : 0x0001

Count of disks sectors : 0x00000002

Flags : 0x00000000

Operation system specific value #1 : 0x00000001

Direct block pointer 0 : 0x00008401

Direct block pointer 1-11: 0x00000000

Singly indirect block pointer : 0x00000000

Doubly indirect block pointer : 0x00000000

Triply indirect block pointer : 0x00000000

Generation number : 0x39c503a9

Τα byte 104-111 είναι reserved στο ext2 revision 0 και είναι όλα 0

Block address of fragment : 0x000000

Operating system specific value #2 : 0

31. Είναι αποθηκευμένα στο block που δείχνει ο direct block pointer 0 δηλαδή στο block #33793 όπως φαίνεται και παραπάνω στα δεδομένα του inode #3666
32. Όπως φαίνεται και παραπάνω με τις εντολές `debugfs -R 'stat <3666>' /dev/vdb` και `hexdump -s $((1024*16389)) -n 128 /dev/vdb` το μέγεθος του αρχείου είναι 42 bytes.
33. Με την εντολή `cat /mnt/dir2/helloworld` μπορούμε να δούμε τα περιεχόμενα του αρχείου

```
root@utopia:/mnt/dir2# cat /mnt/dir2/helloworld
Welcome to the Mighty World of Filesystemsroot@utopia:/mnt/dir2#
```

Επίσης μπορούμε να δούμε τα περιεχόμενα του block #33793 τα οποίο είναι το block που περιέχει τα δεδομένα του αρχείου helloworld.

```
root@utopia:/mnt/dir2# hexdump -s $((1024*33793)) -n 1024 -C /dev/vdb
02100400  57 65 6c 63 6f 6d 65 20  74 6f 20 74 68 65 20 4d  |Welcome to the M|
02100410  69 67 68 74 79 20 57 6f  72 6c 64 20 6f 66 20 46  |ighty World of F|
02100420  69 6c 65 73 79 73 74 65  6d 73 00 00 00 00 00 00  |ilesystems.....|
02100430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
02100800
```

ΑΣΚΗΣΗ 2

1. Για να συνδέσουμε την εικόνα του δίσκου `fdisk2.img` στο κατάλογο `/mnt` εκτελούμε την εντολή `mount /dev/vdc /mnt`.
2. Για να δημιουργήσουμε ένα κενό αρχείο με την εντολή `touch` εκτελούμε την εντολή `touch /mnt/file1`.
3. Η εντολή δεν εκτελέστηκε και βγάζει μήνυμα ότι δεν υπάρχει ελεύθερος χώρος στην συσκευή.

```
root@utopia:~# touch /mnt/file1
touch: cannot touch '/mnt/file1': No space left on device
```

4. Η εντολή `touch` προσπάθησε να καλέσει την `openat` αλλά αυτή επέστρεψε -1 (ENOSPC)

```
openat(AT_FDCWD, "file1", O_WRONLY|O_CREAT|O_NOCTTY|O_NONBLOCK, 0666) = -1 ENOSPC (No space left on device)
```

5. Με την εντολή `find . -type f | wc -l` μπορούμε να δούμε τον αριθμό των αρχείων μέσα στον φάκελο `mnt`

```
root@utopia:/mnt# find . -type f | wc -l
4868
```

Επίσης μπορούμε να δούμε από το `superblock` τα διαθέσιμα inodes τα οποία είναι 0 και το συνολικό αριθμό των inodes στο σύστημα αρχείων που είναι 5136 αν από αυτά αφαιρέσουμε τα πρώτα 11 που είναι

δεσμευμένα και προσθέσουμε το inode για το root directory και για το lost+found τότε τα συνολικά inodes είναι 5127. Αν τώρα διαβάσουμε τους block group descriptors μπορούμε να βρούμε το συνολικό αριθμό των directories στα block groups τα οποία είναι 259. Αφαιρούμε λοιπόν τα inodes που είναι δεσμευμένα από τα directories και βρίσκουμε ότι το συνολικός αριθμός των inodes που είναι δεσμευμένα από αρχεία είναι 4868 άρα ο συνολικός αριθμός αρχείων είναι 4868.

```
root@utopia:/dev# hexdump -s 1024 -n 1024 -C /dev/vdc
00000400  10 14 00 00 00 50 00 00 00 00 00 00 63 4c 00 00 |.....P.....CL..|
00000410  00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000420  00 20 00 00 00 20 00 00 b0 06 00 00 b3 3e c5 63 |. . . . .>.C|
00000430  b3 3e c5 63 02 00 ff ff 53 ef 00 00 01 00 00 00 |.>.C....S.....|
00000440  6e 73 9c 63 00 00 00 00 00 00 00 00 01 00 00 00 |ns.C.....|
00000450  00 00 00 00 0b 00 00 00 80 00 00 00 00 00 00 00 |.....|
00000460  00 00 00 00 00 00 00 00 33 3a 01 a6 6b b5 4d 3b |.....3:...k.M;|
00000470  92 22 0c 66 ca fd 28 9b 66 73 64 69 73 6b 32 2e |. ".f..(.fsdisk2.|
00000480  69 6d 67 00 00 00 00 00 2f 68 6f 6d 65 2f 6a 69 |img..../home/jl|
00000490  6d 73 69 61 6b 2f 63 73 6c 61 62 2d 67 69 74 2f |msiak/cslab-git/|
000004a0  66 73 2f 6d 6e 74 00 00 00 00 00 00 00 00 00 00 |fs/mnt.....|
000004b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000004e0  00 00 00 00 00 00 00 00 00 00 00 00 0a 40 a8 44 |.....@.D|
000004f0  67 47 49 94 81 d1 a7 f0 fd e5 43 e4 01 00 00 00 |gGI.....C.....|
00000500  0c 00 00 00 00 00 00 00 6e 73 9c 63 00 00 00 00 |.....ns.C....|
00000510  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000560  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000570  00 00 00 00 00 00 00 00 98 03 00 00 00 00 00 00 |.....|
00000580  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000800
```

6. Τα συνολικά blocks του συστήματος είναι 20480 από τα οποία χρησιμοποιούνται τα 19555 άρα τα δεδομένα καταλαμβάνουν 925 blocks δηλαδή 947200 bytes. Την πληροφορία αυτή μπορούμε να την βρούμε και από την εντολή `dumpe2fs /dev/vdc` και κοιτώντας τις πληροφορίες μέσα στο superblock. Τα μεταδεδομένα είναι ο όγκος που καταλαμβάνουν τα inode στο σύστημα αρχείων δηλαδή $32 \cdot 4868$ δηλαδή 155776 bytes.
7. Το συνολικό μέγεθος του συστήματος αρχείων είναι όσο τα blocks επί το block size δηλαδή 2097152 bytes.
8. Το σύστημα αρχείων έχει 19555 blocks ακόμα διαθέσιμα κάτι το οποίο μπορούμε να δούμε με την εντολή `dumpe2fs /dev/vdc` και από το superblock.
9. Στο συγκεκριμένο σύστημα αρχείων υπάρχουν αρκετά διαθέσιμα blocks και θα μπορούσαμε θεωρητικά να δημιουργήσουμε πολλά αρχεία αλλά όπως φαίνεται και παραπάνω δεν υπάρχουν διαθέσιμα inodes ώστε να μπορέσουν να αποθηκευτούν οι πληροφορίες για το νέο αρχείο.

ΑΣΚΗΣΗ 3

1. Στα linux μπορούν να χρησιμοποιηθούν και το `fsck` και το `e2fsck` (το οποίο είναι αποκλειστικό για τα ext2, ext3 και ext4) για να ελέγξουν ένα σύστημα αρχείων ext2 για αλλοιώσεις.
2. 10 λόγοι που προκαλούν αλλοιώσεις στο σύστημα αρχείων είναι :
 - 1) Λάθος λογισμικού δηλαδή κάποιο λογισμικό προσπάθησε να γράψει κάτι σε εάν σύστημα αρχείων άλλαξε κάποια άλλη σημαντική οντότητα στο δίσκο (π.χ. το superblock).

- 2) Αστοχία hardware , για παράδειγμα ένα δίσκος μπορεί να μην λειτουργεί σωστά με αποτέλεσμα να γραφεί δεδομένα σε λάθος block.
- 3) Λάθος από την μεριά του χρήστη, δηλαδή μπορεί ο χρήστης να αλλάξει κάτι στο δίσκο που να μην έπρεπε να αλλαχτεί (π.χ. το group descriptor table).
- 4) Ιοί οι οποίοι καταστρέφουν τις απαραίτητες δομές για την λειτουργία του συστήματος αρχείων.
- 5) Απότομη παύση του συστήματος την ώρα που γίνονται διεργασίες πάνω στο σύστημα αρχείων.
- 6) Χρήση μη συμβατού driver ή driver με λάθη στον κώδικά του.
- 7) Χρήση λειτουργικού συστήματος που δεν υποστηρίζει το σύστημα αρχείων.
- 8) Αλλαγή στις σημαντικές δομές του δίσκου από εξωτερικούς παράγοντες όπως για παράδειγμα από υπέρταση ή έκθεση σε έντονα ηλεκτρομαγνητικά πεδία.
- 9) Σφάλμα κατά την δημιουργία του συστήματος αρχείων το οποίο να μην εμφανίσει αμέσως επιπτώσεις.
- 10) Απότομη παύση της διαδικασίας μεταφοράς αρχείων στο σύστημα αρχείων.

Την δεύτερη αστοχία μπορούμε να την δούμε στο inode #3435 το οποίο είναι το πρώτο inode του block group #2 και βρίσκεται στο block #16389.

```
root@utopia:~# hexdump -s $((16389*1024)) -n 128 -C /dev/vdd
01001400  ed 41 00 00 00 04 00 00  71 73 9c 63 71 73 9c 63  |.A.....qs.cqs.c|
01001410  71 73 9c 63 00 00 00 00  00 00 01 00 02 00 00 00  |qs.C.....|
01001420  00 00 00 00 04 00 00 00  e8 00 00 00 00 00 00 00  |.....|
01001430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
01001460  00 00 00 00 d3 4c 70 ef  00 00 00 00 00 00 00 00  |....Lp.....|
01001470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
01001480
```

Την Τρίτη αστοχία μπορούμε να την δούμε κοιτώντας στο block bitmap όπου το bit #34 έχει τιμή 0 ενώ θα έπρεπε να είναι 1.

```
root@utopia:/dev# hexdump -s $((3*1024)) -n 1024 -C /dev/vdd
00000c00  ff ff ff ff fd ff ff ff  ff ff ff ff ff ff ff ff  |.....|
00000c10  ff ff ff ff ff ff ff ff  ff ff ff ff ff 00 00 00  |.....|
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
00000c40  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
00000c50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
00001000
```

Την τέταρτη αστοχία μπορούμε να την δούμε στο superbloc που τα bytes 11-15 είναι 0x37303539 δηλαδή 925906233.

```
root@utopia:/dev# hexdump -s 1024 -n 1024 -C /dev/vdd
00000400  10 14 00 00 00 50 00 00  00 04 00 00 39 35 30 37  |....P.....9507|
00000410  f9 13 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |.....|
00000420  00 20 00 00 00 20 00 00  b0 06 00 00 71 73 9c 63  |. . . . .qs.c|
00000430  71 73 9c 63 01 00 ff ff  53 ef 01 00 01 00 00 00  |qs.C....S.....|
00000440  71 73 9c 63 00 00 00 00  00 00 00 00 01 00 00 00  |qs.C.....|
00000450  00 00 00 00 0b 00 00 00  80 00 00 00 00 00 00 00  |.....|
00000460  00 00 00 00 00 00 00 00  82 fb 64 4f d6 66 48 68  |.....d0.fHh|
00000470  94 22 60 e5 7c f3 9e 56  66 73 64 69 73 6b 33 2e  |."|.Vfsdisk3.|
00000480  69 6d 67 00 00 00 00 00  2f 68 6f 6d 65 2f 6a 69  |img..../home/ji|
00000490  6d 73 69 61 6b 2f 63 73  6c 61 62 2d 67 69 74 2f  |mslak/cslab-git/|
000004a0  66 73 2f 6d 6e 74 00 00  00 00 00 00 00 00 00 00  |fs/mnt.....|
000004b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
000004e0  00 00 00 00 00 00 00 00  00 00 00 00 23 e1 8f 01  |.....#...|
000004f0  6d e1 44 ed 81 23 b1 26  6e 12 bd d7 01 00 00 00  |m.D..#.&n.....|
00000500  0c 00 00 00 00 00 00 00  71 73 9c 63 00 00 00 00  |.....qs.C....|
00000510  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
00000560  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
00000570  00 00 00 00 00 00 00 00  1f 00 00 00 00 00 00 00  |.....|
00000580  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
00000800
```

5. Η πρώτη επιδιόρθωση που κάνουμε είναι τα byte 0x837006-0x83700A από 03 00 42 4F 4F τα κάνουμε 01 00 2E 00 00. Η δεύτερη επιδιόρθωση που κάνουμε είναι το byte 0x100141A από 01 σε 02. Η τρίτη επιδιόρθωση που κάνουμε είναι να αλλάξουμε την τιμή του bit που αναπαριστά την διαθεσιμότητα του block #34 στο block bitmap από 0 σε 1, δηλαδή αλλάζουμε την τιμή του byte 0xC04 από FD σε FF. Την τέταρτη επιδιόρθωση που κάνουμε είναι να αλλάξουμε τα byte 0x40C - 0x40F από 39 35 30 37 σε 58 4D 00 00. Με αυτές τις αλλαγές όταν εκτελούμε την εντολή fsck -n /dev/vdd φαίνεται πως δεν υπάρχει κάποιο λάθος στο σύστημα αρχείων.