



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ
ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

Εργαστήριο Λειτουργικών Συστημάτων

Εργαστηριακή Άσκηση 3 Συστήματα Αρχείων σε Περιβάλλον Linux

Σταύρος Λαζόπουλος 03120843, Ομάδα 66

2.3 Η εικόνα fsdisk1.img

1) Έγινε η παρακάτω προσθήκη στο utopia.sh

```
127 exec $QEMU -enable-kvm -M pc -m $UTOPIA_MEMORY_MB \  
128 -smp 2 -drive file=$QCOW2_PRIVATE_FILE,if=virtio \  
129 -net nic -net user,hostfwd=tcp:$UTOPIA_SSH_INTERFACE:$UTOPIA_SSH_PORT-:22 \  
130 -vnc 127.0.0.1:0 \  
131 -nographic -monitor /dev/null \  
132 -chardev socket,id=sensors0,host=lunix.cslab.ece.ntua.gr,port=49152,ipv4=on \  
133 -device isa-serial,chardev=sensors0,id=serial1,index=1 \  
134 -fsdev local,id=fsdev0,path="$SHARED_FS_DIR",security_model=none \  
135 -device virtio-9p-pci,fsdev=fsdev0,mount_tag=shared \  
136 -drive file=fsdisk1-7c2b425a6.img,format=raw,if=virtio \  
137 -drive file=fsdisk2-a0173283d.img,format=raw,if=virtio \  
138 -drive file=fsdisk3-982902777.img,format=raw,if=virtio \  
139 $*
```

Στην πρώτη σειρά προσθέσαμε τον δίσκο fdisk1.img. Αν κάνουμε blkid βλέπουμε ότι το όνομα του δίσκου στο utopia είναι vdb

```
root@utopia:~# blkid  
/dev/vda1: UUID="ce8cc13a-d361-471d-adac-3061bf165cf4" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="46795e59-01"  
/dev/vdb: LABEL="fsdisk1.img" UUID="c63028e5-711b-410d-a263-e7ca2b15a8d3" BLOCK_SIZE="1024" TYPE="ext2"  
/dev/vdc: LABEL="fsdisk2.img" UUID="d1266ad1-dae1-4275-8136-a29a4dfc9d1f" BLOCK_SIZE="1024" TYPE="ext2"  
/dev/vdd: LABEL="fsdisk3.img" UUID="19032143-52ce-4917-8ec5-991c89ee421b" BLOCK_SIZE="1024" TYPE="ext2"
```

2) Κάνουμε lsblk και βλέπουμε ότι το μέγεθος του δίσκου vdb είναι 50mB

```
user@utopia:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0          2:0    1    4K  0 disk
sr0         11:0    1 1024M  0 rom
vda         254:0    0   11G  0 disk
└─vda1      254:1    0   11G  0 part /
vdb         254:16   0    50M  0 disk
vdc         254:32   0    20M  0 disk
vdd         254:48   0    20M  0 disk
```

Κάνουμε και hexdump με offset 1024 και μέγεθος 1024 για να δούμε το superblock (βλέπουμε παρακάτω στα metadata του fs το μέγεθος του δίσκου)

```
root@utopia:/# hexdump /dev/vdb -s 1024 -n 1024 -C
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 8e 2f 99 65 |. ... ..(....|.e|
00000430  8e 2f 99 65 02 00 ff ff 53 ef 00 00 01 00 00 00 |./..e....S.....|
00000440  e4 7a 78 65 00 00 00 00 00 00 00 00 01 00 00 00 |.zxe.....|
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 c6 30 28 e5 71 1b 41 0d |.....0(.q.A.|
00000470  a2 63 e7 ca 2b 15 a8 d3 66 73 64 69 73 6b 31 2e |.c..+...fsdisk1.|
00000480  69 6d 67 00 00 00 00 00 2f 63 73 6c 61 62 2d 62 |img..../cslab-b|
00000490  75 6e 6b 65 72 00 00 00 00 00 00 00 00 00 00 00 |unker.....|
000004a0  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 a7 f1 37 f1 |.....7.|
000004f0  07 8d 4b b9 9b 0d e1 b1 22 bc 50 38 01 00 00 00 |..K.....".P8....|
00000500  0c 00 00 00 00 00 00 00 e4 7a 78 65 00 00 00 00 |.....zxe....|
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 0d 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
```

Το block size όπως είπαμε είναι 1024 το οποίο φαίνεται και από τα bytes 24-27 (00 00 00 00), και έχουμε συνολικά 0x c800 = 51200 blocks το οποίο φαίνεται από τα bytes 4-7 (00 00 c8 00), άρα συνολικά έχουμε μέγεθος 52428800 bytes ή 50Mb

3) Περιέχει σύστημα αρχείων ext2 όπως φαίνεται παραπάνω στην εντολή blkid.

Στο superblock του ext2 υπάρχει ext2 signature (0xef53) στα bytes 56-57

4) Βλέπουμε τα metadata του fs μέσω της dumpe2fs

```
root@utopia:/# dumpe2fs /dev/vdb
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   fsdisk1.img
Last mounted on:          /cslab-bunker
Filesystem UUID:          c63028e5-711b-410d-a263-e7ca2b15a8d3
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      (none)
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              12824
Block count:              51200
Reserved block count:    2560
Free blocks:              49552
Free inodes:              12810
First block:              1
Block size:               1024
Fragment size:            1024
Blocks per group:         8192
Fragments per group:     8192
Inodes per group:         1832
Inode blocks per group:   229
Filesystem created:       Tue Dec 12 17:23:16 2023
Last mount time:          Tue Dec 12 17:23:16 2023
Last write time:          Tue Dec 12 17:23:17 2023
```

Αυτό το σύστημα αρχείων δημιουργήθηκε Tue Dec 12 17:23:16 2023

Το ext2 θεωρητικά δεν προβλέπει την ύπαρξη πληροφορίας σχετικά με την ώρα δημιουργίας του. Όμως αφήνει 788 unused bytes από το offset 264. Το ext4 αποθηκεύει στα bytes 264-267 την ημερομηνία δημιουργίας του σε δευτερόλεπτα σε unix time(seconds since epoch midnight 1/1/1970 or posix time or unix timestamp). Πολλά συστήματα αποθηκεύουν στα unused bytes του ext2 πληροφορίες σχετικές με το ext4. Έτσι και εμείς στα bytes 264-267 ή 0x108-10B (e4 7a 78 65) βλέπουμε σε little endian μορφή το posix time Tue Dec 12 17:23:16 2023

5) Προσαρτήθηκε τελευταία φορά Tue Dec 12 17:23:16 2023

Στα bytes 44-47 (e4 7a 78 65) βλέπουμε σε little endian μορφή το posix time Tue Dec 12 17:23:16 2023

6) Προσαρτήθηκε τελευταία φορά στο /cslab-bunker

Στο superblock στα bytes 136-199 αποθηκεύεται σε μορφή string το path του καταλόγου αυτού. Φαίνεται και στο ASCII representation δεξιά του hexdump.

7) Τροποποιήθηκε τελευταία φορά Tue Dec 12 17:23:17 2023

Στα bytes 48-51 (e4 7a 78 65) βλέπουμε σε little endian μορφή το posix time Tue Dec 12 17:23:17 2023

8) Σε ένα σύστημα αρχείων το block είναι η μεγαλύτερη ποσότητα χώρου στον δίσκο που μπορεί να γίνει allocated για ένα αρχείο και η μέγιστο μέγεθος πληροφορίας που μπορεί να μεταφερθεί σε μία λειτουργία I/O ([IBM](#)).

9) Το block size καθορίζει το μέγιστο μέγεθος μιας έτησης read ή write που το σύστημα αρχείων στέλνει στο driver της συσκευής I/O ([IBM](#)).

10) Το inode (index node) σε ένα σύστημα αρχείων είναι μια δομή δεδομένων που περιγράφει ένα αρχείο του συστήματος. Κάθε inode αποθηκεύει τα χαρακτηριστικά και τις τοποθεσίες των disk block των δεδομένων του αρχείου. Τα χαρακτηριστικά του αρχείου στο σύστημα αρχείων μπορεί να περιλαμβάνουν μεταδεδομένα, καθώς και δεδομένα ιδιοκτήτη και δικαιωμάτων. Δεν περιλαμβάνει το αρχείο αυτό καθαυτό αλλά δείκτες προς τα blocks που το περιέχουν

11) Από την dumpe2fs βλέπουμε ότι το inode size είναι 128.

```
First inode:      11
Inode size:       128
```

Στο superblock βλέπουμε το μέγεθος του inode στα bytes 88-89 (80 00 00 00) σε little endian.

12) Από την dumpe2fs βλέπουμε ότι το inode count είναι 12824 και το Block count 51200

```
Inode count:      12824
Block count:      51200
```

13) Το superblock είναι ουσιαστικά μεταδεδομένα του συστήματος αρχείων και καθορίζει τον τύπο, το μέγεθος, την κατάσταση του συστήματος αρχείων και πληροφορίες σχετικά με άλλες δομές μεταδεδομένων.

14) Το superblock λόγω της ανάγκης για redundancy αποθηκεύεται στην αρχή κάθε block group. Το πρώτο βρίσκεται πάντα στο block 1 του δίσκου (δεύτερο block)

15) Χρειάζεται να έχουμε εφεδρικά αντίγραφα του superblock διότι σε περίπτωση που αυτό χανόταν δεν θα είχαμε την δυνατότητα να διαβάσουμε αρχεία από το fs.

16)

```
Group 1: (Blocks 8193-16384)
Backup superblock at 8193, Group descriptors at 8194-8194
Block bitmap at 8195 (+2)
Inode bitmap at 8196 (+3)
Inode table at 8197-8425 (+4)
7958 free blocks, 1831 free inodes, 1 directories
Free blocks: 8427-16384
Free inodes: 1834-3664
Group 2: (Blocks 16385-24576)
Backup superblock at 16385, Group descriptors at 16386-16386
Block bitmap at 16387 (+2)
Inode bitmap at 16388 (+3)
Inode table at 16389-16617 (+4)
7959 free blocks, 1832 free inodes, 0 directories
Free blocks: 16618-24576
Free inodes: 3665-5496
Group 3: (Blocks 24577-32768)
Backup superblock at 24577, Group descriptors at 24578-24578
Block bitmap at 24579 (+2)
Inode bitmap at 24580 (+3)
Inode table at 24581-24809 (+4)
7959 free blocks, 1832 free inodes, 0 directories
Free blocks: 24810-32768
Free inodes: 5497-7328
Group 4: (Blocks 32769-40960)
Backup superblock at 32769, Group descriptors at 32770-32770
Block bitmap at 32771 (+2)
Inode bitmap at 32772 (+3)
Inode table at 32773-33001 (+4)
7959 free blocks, 1832 free inodes, 0 directories
Free blocks: 33002-40960
Free inodes: 7329-9160
Group 5: (Blocks 40961-49152)
Backup superblock at 40961, Group descriptors at 40962-40962
Block bitmap at 40963 (+2)
Inode bitmap at 40964 (+3)
Inode table at 40965-41193 (+4)
7959 free blocks, 1830 free inodes, 1 directories
Free blocks: 41194-49152
Free inodes: 9163-10992
Group 6: (Blocks 49153-51199)
Backup superblock at 49153, Group descriptors at 49154-49154
Block bitmap at 49155 (+2)
Inode bitmap at 49156 (+3)
Inode table at 49157-49385 (+4)
1814 free blocks, 1832 free inodes, 0 directories
Free blocks: 49386-51199
Free inodes: 10993-12824
```

17) Το block group είναι μία σειρά από data blocks μαζί με τα inodes τους το, inode bitmap, το block bitmap, το block group descriptor, και το superblock.

19) Το fs μας έχει 7 groups (group 0 - group 6) το καθένα από τα οποία έχει μέγεθος 8192 groups ξεκινώντας από το group 1.

Από το superblock μπορούμε να το καταλάβουμε αυτό γνωρίζοντας ότι το inode count είναι 12824 και βλέποντας από τα bytes 40-43 (28 07 00 00) ότι τα συνολικά inodes ανά block group είναι $0x0728 = 1832$ άρα έχουμε $12824 / 1832 = 7$ groups.

20) Το Block group descriptor περιέχει την διεύθυνση του block usage bitmap και inode usage bitmap, την αρχική διεύθυνση του inode table, τους αριθμούς των unallocated blocks και inodes και τον αριθμό των καταλόγων στο group

21) έτσι ώστε αν χαθεί ένα block group descriptor να μην χαθεί πληροφορία απαραίτητη για την λειτουργία του fs





22) Τα group descriptors βρίσκονται στα blocks 2, 8194, 16386, 24578, 32770, 40962, 49154

23) Το block bitmap και το inode bitmap είναι bitmaps στα οποία το 1 σημαίνει ότι το αντίστοιχο block η inode είναι allocated. Στο ext2 fs που εξετάζουμε βρίσκονται 2 και 3 blocks αντίστοιχα μετά το superblock.

24) Τα inode tables αποτελούνται από συνεχόμενα blocks το καθένα από τα οποία περιέχει έναν προκαθορισμένο αριθμό από inodes. Στο ext2 τα inodes έχουν μέγεθος 128 bytes, έτσι και το block με μέγεθος 1024 περιέχει 8 inodes. βρίσκονται 4 blocks μετά το superblock

25) Παρακάτω φαίνεται το inode data structure ([link](#))

Inode Data Structure

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

Τα inodes βρίσκονται στο inode table και υπάρχει pointer στο πρώτο από αυτά στο block group descriptor.

26) Από τα bytes 40-43 (28 07 00 00) βλέπουμε ότι τα συνολικά inodes ανά block group είναι 0x0728= 1832 και από τα bytes 32-35 (00 20 00 00) βλέπουμε ότι τα συνολικά blocks ανά block group είναι 0x2000=8192

27) Αρχικά πρέπει να κάνουμε mount το fs.

```
root@utopia:~# mount /dev/vdb /mnt
root@utopia:~#
```

Χρησιμοποιούμε την stat για το αρχείο

```
root@utopia:/mnt/dir2# stat helloworld
  File: helloworld
  Size: 42          Blocks: 2          IO Block: 1024   regular file
Device: fe10h/65040d Inode: 9162        Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2024-01-07 19:48:49.000000000 +0200
Modify: 2023-12-12 17:23:16.000000000 +0200
Change: 2023-12-12 17:23:16.000000000 +0200
 Birth: -
```

Βλέπουμε ότι αντιστοιχεί στο inode 9162

Για να κάνουμε το ίδιο με hexdump:

Το inode του root directory είναι πάντα το inode #2. Βρίσκεται στο Block Group #0 (διαβάσαμε το inodes per group από το superblock = 1832)

Βρίσκουμε σε ποιο block είναι το inode table του BG #0 (από το GDT που βρίσκεται στο BG #0 -> block #2)

Βρήκαμε το inode table του BG #0 στο block #5 (0x05)

Διαβάζουμε το inode #2 από το inode table που ξεκινάει στο block #5

Στο byte #40 του inode βρίσκεται το Direct Block Pointer 0
Βρήκαμε το directory entry block του "/" block #234(0xEA)

```

root@utopia:/mnt/dir2# export blocksiz=1024
root@utopia:/mnt/dir2# export GDT_OFFSET=$((2*blocksiz))
root@utopia:/mnt/dir2# export GD_SIZE=32
root@utopia:/mnt/dir2# hexdump -s$((GDT_OFFSET)) -n$((GD_SIZE)) -C /dev/vd
vda vda1 vdb vdc vdd
root@utopia:/mnt/dir2# hexdump -s$((GDT_OFFSET)) -n$((GD_SIZE)) -C /dev/vdb
00000800 03 00 00 00 04 00 00 00 05 00 00 00 08 1f 1d 07 |.....|
00000810 02 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000820
root@utopia:/mnt/dir2# hexdump -s$((5*blocksiz+^C
root@utopia:/mnt/dir2# export INODE_SIZE=128
root@utopia:/mnt/dir2# hexdump -s$((5*blocksiz+INODE_SIZE)) -n$((INODE_SIZE)) -C /dev/vdb
00001480 ed 41 00 00 00 04 00 00 5d e3 9a 65 e4 7a 78 65 |.A.....].e.zxe|
00001490 e4 7a 78 65 00 00 00 00 00 00 05 00 02 00 00 00 |.zxe.....|
000014a0 00 00 00 00 02 00 00 00 0a 00 00 00 00 00 00 00 |.....|
000014b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00001500

```

Διαβάζουμε το directory entry block #234. Ψάχνουμε μέσα σε αυτό το όνομα dir2. Στα 8-5 bytes πριν από αυτό έχουμε το inode του /dir2 (inode #9161). Το inode #9161 είναι το 1ο inode του BG #5 (1832 inodes per BG) (τα inodes ξεκινάνε απο το #1)

```
root@utopia:/mnt/dir2# hexdump -s$((234*blocksize)) -n$((128)) -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 29 07 00 00 |lost+found..)|
0003a830 0c 00 04 00 64 69 72 31 c9 23 00 00 c8 03 04 00 |....dir1.#....|
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 |.....|
*
```

Διαβάζουμε απο το BGD του BG #5 το block του inode table. Inode table του BG #5 στο block #40965 (0xA005)

```
root@utopia:/mnt/dir2# hexdump -s$((GDT_OFFSET+5*GD_SIZE)) -n$((GD_SIZE)) -C /dev/vd
hexdump: /dev/vd: No such file or directory
hexdump: stdin: Bad file descriptor
root@utopia:/mnt/dir2# hexdump -s$((GDT_OFFSET+5*GD_SIZE)) -n$((GD_SIZE)) -C /dev/vdb
000008a0  03 a0 00 00 04 a0 00 00  05 a0 00 00 17 1f 26 07  |.....&.|
000008b0  01 00 04 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
000008c0
```

Διαβάζουμε το 1ο inode του inode table και βρίσκουμε το block του directory entry για το /dir2
Είναι το block #247 (0xf7)

```
root@utopia:/mnt/dir2# hexdump -s$((4096*blocksize)) -n$((INODE_SIZE)) -C /dev/vdb
02801400  ed 41 00 00 00 04 00 00  5d e3 9a 65 e4 7a 78 65  |.A.....]..e.zxe|
02801410  e4 7a 78 65 00 00 00 00  00 00 02 00 02 00 00 00  |.zxe.....|
02801420  00 00 00 00 02 00 00 00  f7 00 00 00 00 00 00 00  |.....|
02801430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
*
02801460  00 00 00 00 c5 b2 72 92  00 00 00 00 00 00 00 00  |.....r.....|
02801470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.....|
02801480
```


Διαβάζουμε το directory entry block #247. Ψάχνουμε μέσα σε αυτό το όνομα helloworld. Βρήκαμε το inode του /dir2/helloworld (inode #9162)

```
root@utopia:/mnt/dir2# hexdump -s$((247*blocksize)) -n$((64)) -C /dev/vdb
0003dc00 c9 23 00 00 0c 00 01 00 2e 00 00 00 02 00 00 00 |.#.....|
0003dc10 0c 00 02 00 2e 2e 00 00 ca 23 00 00 e8 03 0a 00 |.....#.....|
0003dc20 68 65 6c 6c 6f 77 6f 72 6c 64 00 00 00 00 00 00 |helloworld.....|
0003dc30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
0003dc40
root@utopia:/mnt/dir2#
```

28) Αφού κάθε group έχει 1832 inodes το inode 9162 αντιστοιχεί στο group 5

29) Στο block 40965

30)

```
root@utopia:/mnt/dir2# hexdump -s$((40965*blocksize)) -n$((INODE_SIZE)) -C /dev/vdb
02801400 ed 41 00 00 00 04 00 00 5d e3 9a 65 e4 7a 78 65 |.A.....]..e.zxe|
02801410 e4 7a 78 65 00 00 00 00 00 00 02 00 02 00 00 00 |.zxe.....|
02801420 00 00 00 00 02 00 00 00 f7 00 00 00 00 00 00 00 |.....|
02801430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
02801460 00 00 00 00 c5 b2 72 92 00 00 00 00 00 00 00 00 |.....r.....|
02801470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
02801480
```

31) Διαβάζουμε το inode του αρχείου (2ο inode του 5ου block group)

```
root@utopia:/mnt/dir2# hexdump -s$((40965*blocksize+INODE_SIZE)) -n$((INODE_SIZE)) -C /dev/vdb
02801480 a4 81 00 00 2a 00 00 00 01 e4 9a 65 e4 7a 78 65 |....*......e.zxe|
02801490 e4 7a 78 65 00 00 00 00 00 00 01 00 02 00 00 00 |.zxe.....|
028014a0 00 00 00 00 01 00 00 00 01 04 00 00 00 00 00 00 |.....|
028014b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
028014e0 00 00 00 00 bc f3 45 a3 00 00 00 00 00 00 00 00 |.....E.....|
028014f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
02801500
```

32)

```
root@utopia:/mnt/dir2# stat helloworld
  File: helloworld
  Size: 42          Blocks: 2          IO Block: 1024   regular file
Device: fe10h/65040d Inode: 9162         Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2024-01-07 19:48:49.000000000 +0200
Modify: 2023-12-12 17:23:16.000000000 +0200
Change: 2023-12-12 17:23:16.000000000 +0200
 Birth: -
```

Έχει μέγεθος 42 bytes

33)

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

root@utopia:/mnt/dir2# hexdump -s$((1025*blocksize)) -n$((64)) -C /dev/vdb
00100400  57 65 6c 63 6f 6d 65 20 74 6f 20 74 68 65 20 4d |Welcome to the M|
00100410  69 67 68 74 79 20 57 6f 72 6c 64 20 6f 66 20 46 |lighty World of F|
00100420  69 6c 65 73 79 73 74 65 6d 73 00 00 00 00 00 00 |ilesystems.....|
00100430  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00100440
```

2.4 Η εικόνα fsdisk2.img

1) Κάνουμε mount τον εικονικό δίσκο στο /mnt

```
user@utopia:~$ su -
Password:
root@utopia:~# mount /dev/vdc /mnt
root@utopia:~# cd /mnt/
root@utopia:/mnt# ls
dir-0  dir-11  dir-3  dir-6  dir-9
dir-1  dir-12  dir-4  dir-7  lost+found
dir-10 dir-2   dir-5  dir-8
root@utopia:/mnt#
```

2) Προσπαθούμε να δημιουργήσουμε το κενό αρχείο file1 στον /mnt.

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

3) Παρατηρούμε ότι η δημιουργία αποτυγχάνει επειδή δεν υπάρχει ελεύθερος χώρος στον δίσκο.

4) Τρέχουμε την ίδια εντολή μέσω strace.

```
root@utopia:/mnt# strace touch file1
openat(AT_FDCWD, "file1", O_WRONLY|O_CREAT|O_NOCTTY|O_NONBLOCK, 0666) = -1 ENOSPC
(No space left on device)
```

Η κλήση συστήματος που προσπάθησε να τρέξει η touch και απέτυχε είναι η openat και ο κωδικός λάθους είναι ENOSPC.

5) Από την dumpe2fs μπορούμε να δούμε ότι έχουμε 1712 inodes ανά group και 3 groups

```
Inodes per group:      1712
```

και 3 groups με κανένα free inode

```

Group 0: (Blocks 1-8192)
  Primary superblock at 1, Group descriptors at 2-2
  Block bitmap at 3 (+2)
  Inode bitmap at 4 (+3)
  Inode table at 5-218 (+4)
  7787 free blocks, 0 free inodes, 84 directories
  Free blocks: 406-8192
  Free inodes:
Group 1: (Blocks 8193-16384)
  Backup superblock at 8193, Group descriptors at 8194-8194
  Block bitmap at 8195 (+2)
  Inode bitmap at 8196 (+3)
  Inode table at 8197-8410 (+4)
  7891 free blocks, 0 free inodes, 83 directories
  Free blocks: 8494-16384
  Free inodes:
Group 2: (Blocks 16385-20479)
  Backup superblock at 16385, Group descriptors at 16386-16386
  Block bitmap at 16387 (+2)
  Inode bitmap at 16388 (+3)
  Inode table at 16389-16602 (+4)
  3877 free blocks, 0 free inodes, 92 directories
  Free blocks: 16603-20479
  Free inodes:

```

Άρα έχουμε συνολικά $1712 \times 3 = 5136$ inodes. Επίσης βλέπουμε ότι έχουμε $84 + 83 + 92 = 259$ directories.

Ο αριθμός των αρχείων είναι ο αριθμός των inodes μείων τα inodes που αντιστοιχούν σε directories, μείων τα 10 reserved inodes, συν 1 για να μην αφαιρέσουμε 2 φορές το inode 2 που είναι reserved για το root folder του fs. άρα συνολικά $5136 - 259 - 10 + 1 = 4868$ αρχεία.

Αντίστοιχα με την `dumpe2fs` χρησιμοποιούμε την `hexdump` για να διαβάσουμε το superblock. Βλέπουμε από τα bytes 40-43 ότι έχουμε `0x06b0=1712` inodes ανά block.

```

root@utopia:/mnt# hexdump /dev/vdc -s 1024 -n 1024 -C
00000400  10 14 00 00 00 50 00 00  00 00 00 00 63 4c 00 00 |.....P.....cL..|
00000410  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00 |.....|
00000420  00 20 00 00 00 20 00 00  b0 06 00 00 8e 11 a5 65 |.  ... .....e|
00000430  8e 11 a5 65 02 00 ff ff  53 ef 00 00 01 00 00 00 |...e....S.....|
00000440  e5 7a 78 65 00 00 00 00  00 00 00 00 01 00 00 00 |.zxe.....|
00000450  00 00 00 00 00 0b 00 00  80 00 00 00 00 00 00 00 |.....|
00000460  00 00 00 00 00 00 00 00  d1 26 6a d1 da e1 42 75 |.....&j ... Bu|
00000470  81 36 a2 9a 4d fc 9d 1f  66 73 64 69 73 6b 32 2e |.6..M...fsdisk2.|
00000480  69 6d 67 00 00 00 00 00  2f 63 73 6c 61 62 2d 62 |img...../cslab-b|
00000490  75 6e 6b 65 72 00 00 00  00 00 00 00 00 00 00 00 |unker.....|
000004a0  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 e0 1a 1f 28 |.....|

```

Διαβάζουμε επίσης τα Block Group Descriptors των 3ων block groups.

```

root@utopia:/mnt# hexdump /dev/vdc -s$((GDT_OFFSET)) -n$((GD_SIZE*3)) -C
00000800 03 00 00 00 04 00 00 00 05 00 00 00 6b 1e 00 00 |.....k...|
00000810 54 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |T.....|
00000820 03 20 00 00 04 20 00 00 05 20 00 00 d3 1e 00 00 |. ... ..|
00000830 53 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |S.....|
00000840 03 40 00 00 04 40 00 00 05 40 00 00 25 0f 00 00 |.ᄁ...ᄁ...ᄁ..%...|
00000850 5c 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |\.....|
00000860

```

Και βλέπουμε από τα bytes 14-15 ότι δεν υπάρχει κανένα unallocated inode στα group και από τα byte 16-17 ότι έχουμε 0x54, 0x53 και 0x5c (=84, 83, 92) directories ανά group.

6) Μέσω της df βλέπουμε ότι τα δεδομένα καταλαμβάνουν 270KB.

```

root@utopia:/mnt# df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            493484          0    493484   0% /dev
tmpfs           99980          452    99528   1% /run
/dev/vda1      11192088    2477832    8121260  24% /
tmpfs          499880          0    499880   0% /dev/shm
tmpfs          5120           0     5120   0% /run/lock
shared         1935295488 1008621656  925937096  53% /home/user/shared
tmpfs          99976          0     99976   0% /run/user/1000
/dev/vdc        19825         270     19555   2% /mnt

```

Τα μεταδεδομένα απαρτίζονται από :

- 1 block για primary superblock
- 2 blocks για backups του superblock
- 3 blocks για group descriptors
- 3 blocks για block bit map
- 3 blocks για inode bit map
- 5136 inodes * 128 bytes inode size / 1024 bytes block size = 642 blocks για inodes

Συνολικά δηλαδή 654 blocks = 654KB

Διαβάζοντας τα group descriptors μέσω της hexdump, βλέπουμε στα bytes 12-13 ότι υπάρχουν 0x1e6b, 0x1ed3 και 0x0f25 (=7787, 7891, 3877) free blocks σε κάθε group αντίστοιχα.

```

root@utopia:/mnt# hexdump /dev/vdc -s$((GDT_OFFSET)) -n$((GD_SIZE*3)) -C
00000800 03 00 00 00 04 00 00 00 05 00 00 00 6b 1e 00 00 |.....k...|
00000810 54 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |T.....|
00000820 03 20 00 00 04 20 00 00 05 20 00 00 d3 1e 00 00 |. ... ..|
00000830 53 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |S.....|
00000840 03 40 00 00 04 40 00 00 05 40 00 00 25 0f 00 00 |.ᄁ...ᄁ...ᄁ..%...|
00000850 5c 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |\.....|
00000860

```

Επίσης από τα byte 4-7 του superblock βλέπουμε ότι έχουμε 0x5000 = 20480 blocks στο σύστημα.

```

root@utopia:/mnt# hexdump /dev/vdc -s 1024 -n 1024 -C
00000400 10 14 00 00 00 50 00 00 00 00 00 00 63 4c 00 00 |.....P.....cL..|

```

Άρα τα συνολικά blocks δεδομένων θα είναι 20480 - 654 - 7787 - 7891 - 3877 - 1 (boot sector) = 270 blocks = 270KB

7) Το μέγεθός του είναι 20480 blocks, από τα οποία $20480 - 654 - 1 = 19825$ μπορούν να περιέχουν δεδομένα αρχείων.

8) Όπως φαίνεται και από τα παραπάνω έχουμε $7787 + 7891 + 3877 = 19555$ free blocks. Οπότε υπάρχει ελεύθερος χώρος.

9) Δεν μπορούμε να γράψουμε επειδή είναι γεμάτα τα inode tables.

2.5 Η εικόνα fsdisk3.img1

1) Το e2fsck (πρέπει να γίνει ως root).

2) Πιθανοί λόγοι για corruption αρχείων είναι :

- Διακοπή ρεύματος
- Malware
- Αστοχία Hardware
- Κακή διαχείριση δεδομένων από drivers / Η γενικά λογισμικό που γράφει απευθείας στον δίσκο

10 πιθανές αλλοιώσεις του fs:

- 1) Αλλοίωση του boot sector με αποτέλεσμα να μην μπορεί να ξεκινήσει το λειτουργικό
- 2) Λάθος μέγεθος block στο superblock
- 3) Αλλαγή των αριθμών των block στο superblock
- 4) Λανθασμένα pointers στο block group descriptor (για block / inode bitmap ή inode table)
- 5) Λάθος block bitmap με αποτέλεσμα να υπάρχει κίνδυνος να γράψουμε σε λάθος block
- 6) Λάθος inode bitmap
- 7) Λάθος type σε inode
- 8) Λάθος permissions στο inode
- 9) Λάθος count των links στο inode (πιθανή διαγραφή χωρίς να έχουν διαγραφεί όλα τα links ή αποτυχία διαγραφής ενώ δεν υπάρχουν links)
- 10) Λάθος block pointer στο inode

3) Προσπαθούμε να επιδιορθώσουμε το σύστημα αρχείων με το e2fsck

```
root@utopia:~# e2fsck /dev/vdd
e2fsck 1.46.2 (28-Feb-2021)
fsdisk3.img contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
```

Η πρώτη αλλοίωση που εντοπίζεται είναι ότι στο directory με inode 1717 το πρώτο entry το όνομα αντί για "." είναι "BOO"

```
Pass 2: Checking directory structure
First entry 'BOO' (inode=1717) in directory inode 1717 (/dir-2) should be '.'
Fix<y>? yes
```

Η δεύτερη αλλοίωση που εντοπίζεται είναι ότι το reference count του inode 3425 είναι 1 ενώ θα έπρεπε να είναι 2 (Είναι το inode του dir-3 άρα μάλλον λείπει το reference από το root του filesystem)

```
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Inode 3425 ref count is 1, should be 2.  Fix<y>? yes

root@utopia:/mnt# ls -li
1713 dir-1  1717 dir-2  3425 dir-3    11 lost+found
```

Η τρίτη αλλοίωση που εντοπίζεται είναι ότι στο block bitmap το block 34 είναι λανθασμένα ορισμένο ως ελεύθερο

```
Pass 5: Checking group summary information
Block bitmap differences:  +34
Fix<y>? yes
```

Το επιβεβαιώνουμε ελέγχοντας την διορθωμένη και την αλλοιωμένη έκδοση του δίσκου

```
root@utopia:/mnt# xxd -s$((3*BLOCK_SIZE)) -l 1024 -b /dev/vdd
00000c00: 11111111 11111111 11111111 11111111 11111111 11111111 .....
00000c06: 11111111 11111111 11111111 11111111 11111111 11111111

[root@garuda images]# xxd -s$((3*1024)) -l 1024 -b fsdisk3-982902777.img
00000c00: 11111111 11111111 11111111 11111111 11111111 11111111 .....
00000c06: 11111111 11111111 11111111 11111111 11111111 11111111
```

Η τέταρτη αλλοίωση που εντοπίζεται είναι ότι το free block count του superbloc είναι 926431538 αλλά το πρόγραμμα μέτρησε 19800.

```
Free blocks count wrong (926431538, counted=19800).
Fix<y>? yes
```

Και τέλος ολοκληρώνεται η εκτέλεση του εργαλείου

```
fsdisk3.img: ***** FILE SYSTEM WAS MODIFIED *****
fsdisk3.img: 23/5136 files (0.0% non-contiguous), 680/20480 blocks
```

4 - 5) Για τη πρώτη αλλοίωση πρέπει πρώτα να βρούμε το Data Block στο οποίο είναι γραμμένα τα Directory Entries. 1712 inodes ανά group και ψάχνουμε το inode 1717 άρα πρέπει να βρούμε το 5 inode του group 2.

```
root@utopia:~# hexdump /dev/vdd -s$((GDT_OFFSET+GD_SIZE)) -n$((GD_SIZE)) -C
00000820  03 20 00 00 04 20 00 00 05 20 00 00 1b 1f a8 06 |. ... ..|
00000830  02 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000840

root@utopia:~# hexdump /dev/vdd -s$((8197*BLOCK_SIZE+4*INODE_SIZE)) -n$((INODE_SIZE))
00801600  ed 41 00 00 00 04 00 00 e9 7a 78 65 e9 7a 78 65 |.A.....zxe.zxe|
00801610  e9 7a 78 65 00 00 00 00 00 00 02 00 02 00 00 00 |.zxe.....|
00801620  00 00 00 00 04 00 00 00 dc 20 00 00 00 00 00 00 |.....|
00801630  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
```


0x20dc = 8412

```
root@utopia:~# hexdump /dev/vdd -s$((8412*BLOCK_SIZE)) -n$((BLOCK_SIZE)) -C
00837000 b5 06 00 00 0c 00 03 00 42 4f 4f 00 02 00 00 00 |.....BOO....|
00837010 0c 00 02 00 2e 2e 00 00 b6 06 00 00 10 00 06 00 |.....|
00837020 66 69 6c 65 2d 31 00 00 b7 06 00 00 10 00 06 00 |file-1.....|
00837030 66 69 6c 65 2d 32 00 00 b8 06 00 00 c8 03 06 00 |file-2.....|
00837040 66 69 6c 65 2d 33 00 00 00 00 00 00 00 00 00 00 |file-3.....|
00837050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
```

Κανουμε search για το 42 4f 4f για να βρούμε το BOO

```
00836FE4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00836FF8 00 00 00 00 00 00 00 00 B5 06 00 00 0C 00 03 00 42 4F 4F 00 .....BOO.
0083700C 02 00 00 00 0C 00 02 00 2E 2E 00 00 B6 06 00 00 10 00 06 00 .....
```

Και το αντικαθιστούμε με 2e και το μέγεθος του ονόματος με 1

```
00836FE4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00836FF8 00 00 00 00 00 00 00 00 B5 06 00 00 0C 00 01 00 2E 00 00 00 .....
0083700C 02 00 00 00 0C 00 02 00 2E 2E 00 00 B6 06 00 00 10 00 06 00 .....
```

Και αν τρέξουμε το e2fsck με -n ώστε να μην κάνει αλλαγές βλέπουμε ότι δεν υπάρχει πια το σφάλμα.

```
fsdisk3.img contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
```

Για τη δεύτερη αλλοίωση ανοίγουμε το inode 3425 που είναι το 1ο του 3ου block group

```
root@utopia:~# hexdump /dev/vdd -s$((GDT_OFFSET+2*GD_SIZE)) -n$((GD_SIZE)) -C
00000840 03 40 00 00 04 40 00 00 05 40 00 00 25 0f ac 06 |.᠔...᠔...᠔..%...|
00000850 01 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000860
root@utopia:~# hexdump /dev/vdd -s$((16389*BLOCK_SIZE)) -n$((BLOCK_SIZE)) -C
01001400 ed 41 00 00 00 04 00 00 e9 7a 78 65 e9 7a 78 65 |.A.....zxe.zxe|
01001410 e9 7a 78 65 00 00 00 00 00 00 01 00 02 00 00 00 |.zxe.....|
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 |.....|
```

Θέλουμε να αλλάξουμε το 01 σε 02

```
000012C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
New position ? 0x01001410
00001310 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
01001404 00 04 00 00 E9 7A 78 65 E9 7A 78 65 E9 7A 78 65 00 00 00 00 .....zxe.zxe.zxe...
01001418 00 00 01 00 02 00 00 00 00 00 00 00 04 00 00 00 .....
01001430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences: +34
```

Για τη τρίτη αλλοίωση γράφουμε στο 34 bit του bitmap 1 (το fd το κάνουμε ff)

```
0000BF4  00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FD FF FF FF
0000C08  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

```
00000BF4  00 00 00 00  00 00 00 00  00 00 00 00  FF FF FF FF  FF FF FF FF
00000C08  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
```

```
fsdisk3.img contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong (926431538, counted=19800).
```

Για τη τρίτη αλλοίωση γράφουμε στα 4 bytes $1024+(12-15)$ τον σωστό αριθμό blocks $19800 = 0x4d58$

```
000003E8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003FC  00 00 00 00 10 14 00 00 00 50 00 00 00 04 00 00  32 39 38 37
00000410  F9 13 00 00 01 00 00 00 00 00 00 00 00 00 00 00  00 20 00 00
```

```
0000003E8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000003FC 00 00 00 00 10 14 00 00 00 50 00 00 00 04 00 58 4D 00
```

```

** vdd = 0x405/0x1400000 = 0%

```

```
root@utopia:~# e2fsck -n /dev/vdd
e2fsck 1.46.2 (28-Feb-2021)
fsdisk3.img: clean, 23/5136 files, 680/20480 blocks
```