Raport 4: Systemy operacyjne – systemy plików

Wstęp

To laboratorium ma na celu zaznajomienie się z podstawowymi koncepcjami na temat systemów plików na przykładzie systemu plików ext4. W ramach trzech zadań wykonamy następujące operacje:

1. Stworzymy plik o wielkości 100 MB, który będzie „udawał” obraz partycji dysku.

2. W tym pliku stworzymy system plików ext4 z domyślnymi parametrami.

3. Przebadamy stworzony, „dziewiczy” system plików za pomocą komendy dumpe2fs oraz edytora szesnastkowego. Poznamy dzięki temu praktycznie strukturę superbloku.

4. Nauczymy się tworzyć tzw. loop device i montować systemy plików pod Linuksem.

5. Zmodyfikujemy system plików tworząc w nim jeden plik.

6. Spróbujemy w strukturach systemu pliku odnaleźć informacje o tym pliku

Zadanie 1

Wykonano komendę

$dd if=/dev/zero of=moj-fs.raw count=100 bs=1M

Co ona robi?

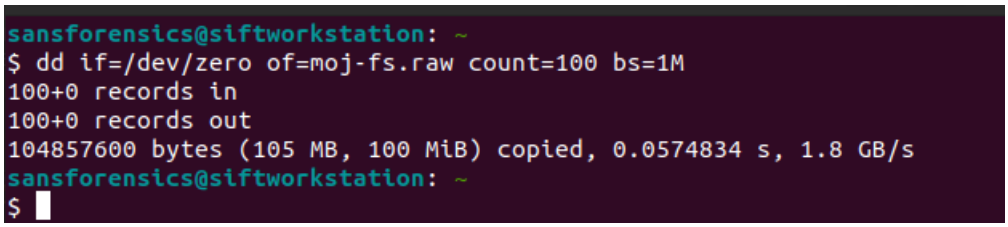
• dd to komenda do kopiowania plików – ale pozwala na kopiowanie dowolnej części ich zawartości

• if=/dev/zero mówi, że źródłowym plikiem jest File /dev/zero – jest to plik specjalny (urządzenie znakowe), które zwraca tylko zera

• of=moj-fs.raw mówi, że plikiem docelowym (wynikowym) jest File moj-fs.raw

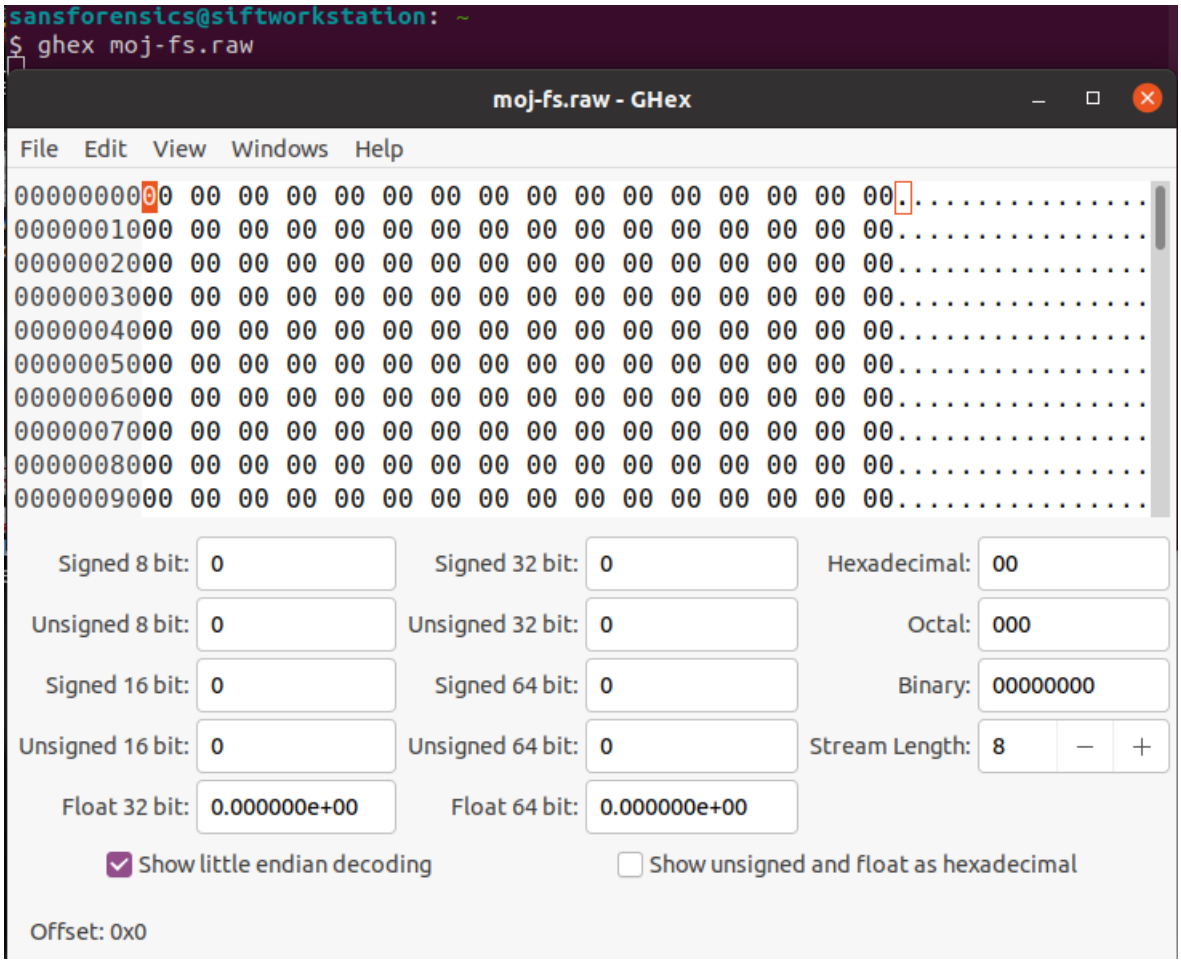
• count=100 określa, że kopiujemy 100 bloków, których rozmiar podany jest jako bs

• bs=1M określa, że operujemy na blokach o wielkości 1 MiB



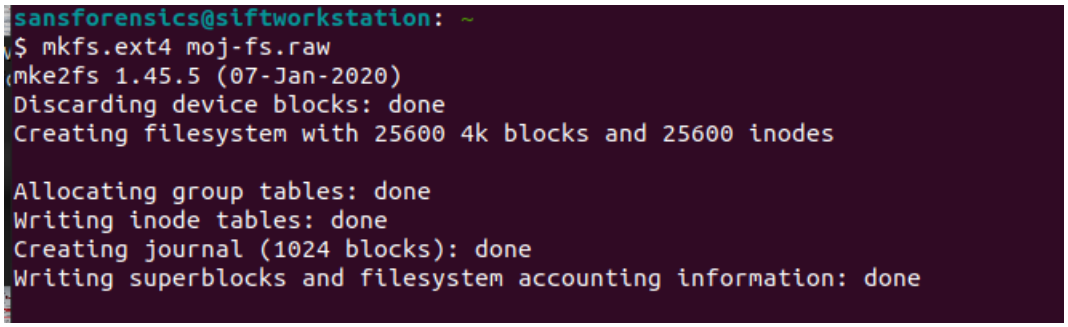
Na tym etapie możesz zweryfikować, że plik składa się z samych zer za pomocą xxd albo ghex lub innego edytora szesnastkowego:

Wpisano komendę: *$ghex moj-fs.raw*



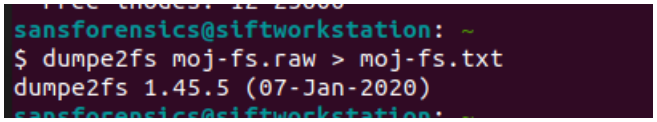
Stwórz system plików w tak stworzonym pliku:

$ mkfs.ext4 moj-fs.raw



Tak właśnie – komenda mkfs.ext4 tworzy system plików ext4. Wykonanie jej na dużym pliku powoduje, że tworzy ona nowy system plików w tym pliku – zupełnie tak, jakbyśmy wykonali ją na fizycznej partycji na dysku. Na tym etapie będziemy chcieli ustalić parametry naszego systemu plików i zrobimy to na dwa sposoby. Pierwszy sposób – prosty. Uruchom komendę:

$ dumpe2fs moj-fs.raw



Dane uzyskane z komendy:

```
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          f8812850-e085-48b8-a34b-bd99a2321b55
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index
filetype extent 64bit flex_bg sparse_super large_file huge_file
dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              25600
Block count:              25600
Reserved block count:     1280
Free blocks:              23754
Free inodes:              25589
First block:              0
Block size:               4096
Fragment size:            4096
Group descriptor size:    64
Reserved GDT blocks:      12
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         25600
Inode blocks per group:   800
Flex block group size:    16
Filesystem created:       Mon Mar 28 11:36:30 2022
Last mount time:          n/a
Last write time:          Mon Mar 28 11:36:30 2022
Mount count:              0
Maximum mount count:      -1
Last checked:             Mon Mar 28 11:36:30 2022
Check interval:           0 (<none>)
Lifetime writes:          65 kB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:              128
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      c49b4947-92e5-4a0f-a5bb-2c3517d9439c
Journal backup:           inode blocks
Checksum type:            crc32c
Checksum:                 0x965b528f
Journal features:         (none)
Journal size:             4096k
Journal length:           1024
Journal sequence:         0x00000001
Journal start:            0


Group 0: (Blocks 0-25599) csum 0xb7f5 [ITABLE_ZEROED]
  Primary superblock at 0, Group descriptors at 1-1
  Reserved GDT blocks at 2-13
  Block bitmap at 14 (+14), csum 0xedc12315
  Inode bitmap at 30 (+30), csum 0x0a40f45f
```

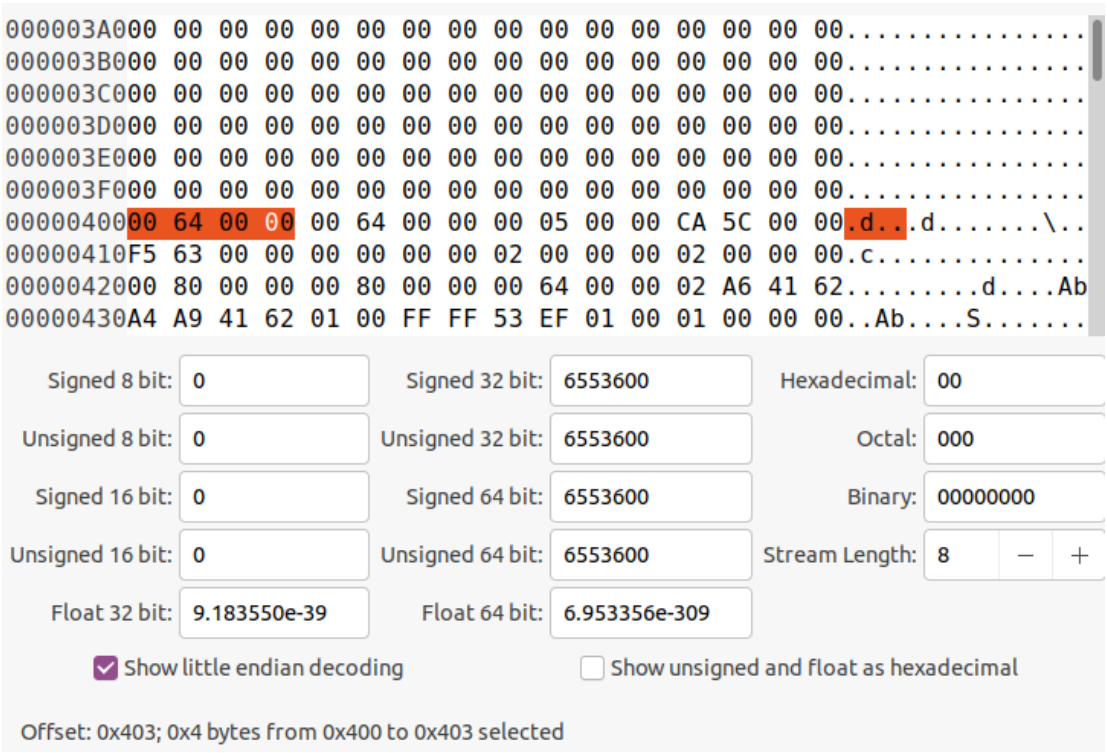
```
  Inode table at 46-845 (+46)
  23754 free blocks, 25589 free inodes, 2 directories, 25589 unused
inodes
  Free blocks: 1846-25599
  Free inodes: 12-25600
```

Tabela wartości, które zostały podane na wykładzie:

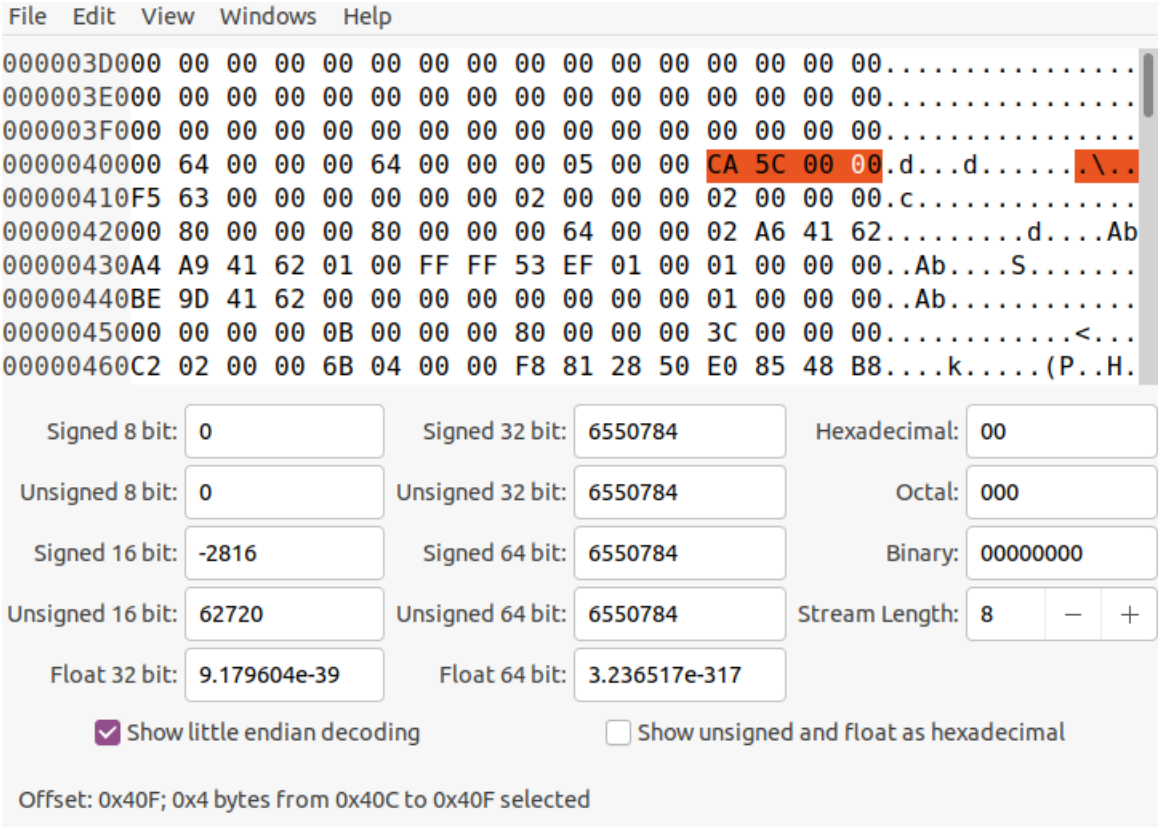
Offset	Rozmiar (B)	Opis
0x00	4	Liczba i-węzłów
0x04	4	Liczba bloków
0x08	4	Liczba bloków zarezerwowanych dla roota
0x0C	4	Liczba wolnych bloków
0x10	4	Liczba wolnych i-węzłów
0x14	4	Pierwszy blok z danymi
0x18	4	Rozmiar bloku
0x1C	4	Rozmiar klastra
0x20	4	Liczba bloków w grupie

Drugi sposób – trudny. Wybierz cztery parametry, które znajdują się w superbloku (mogą być nieomówione na wykładzie) i odnajdź je i ich wartości analizując zapis szesnastkowy w edytorze szesnastkowym (ghex ,xxd).

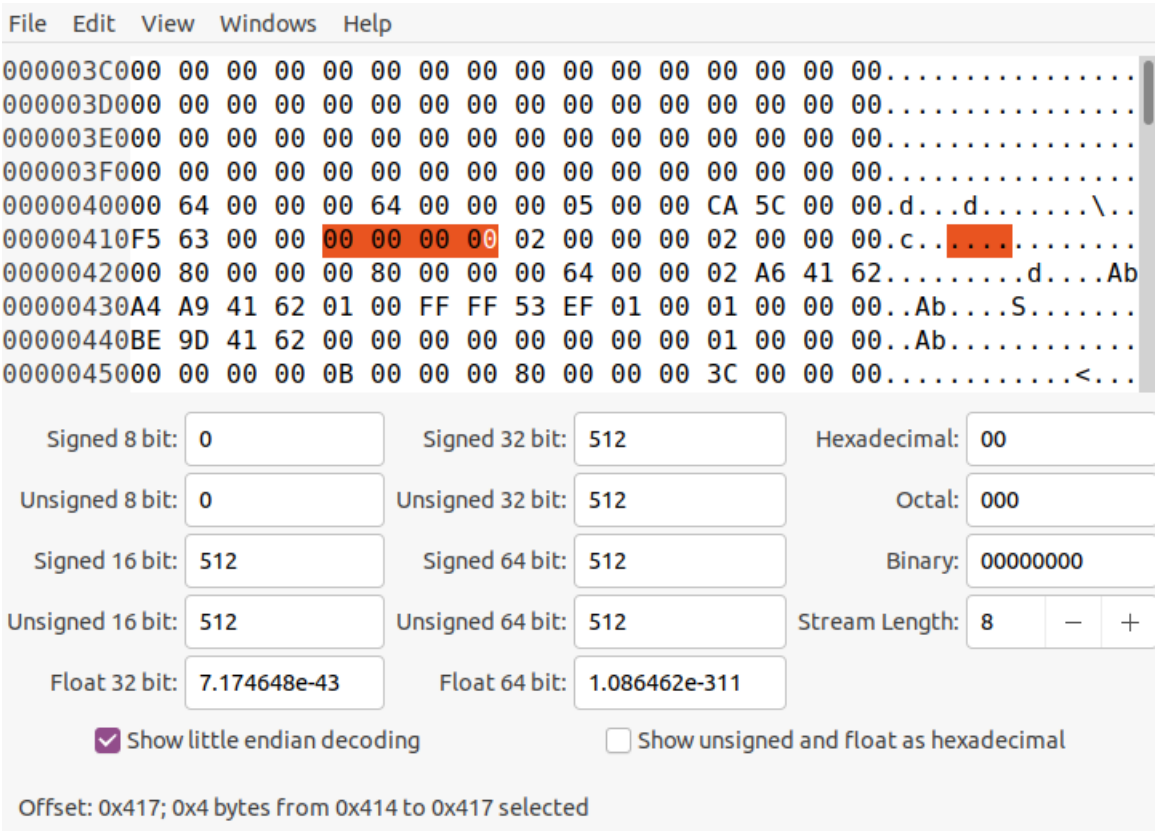
Parametr 1: Liczba i-węzłów, offset 0x00, wartość: 0 x 00 64 00 00, DEC: 25 600



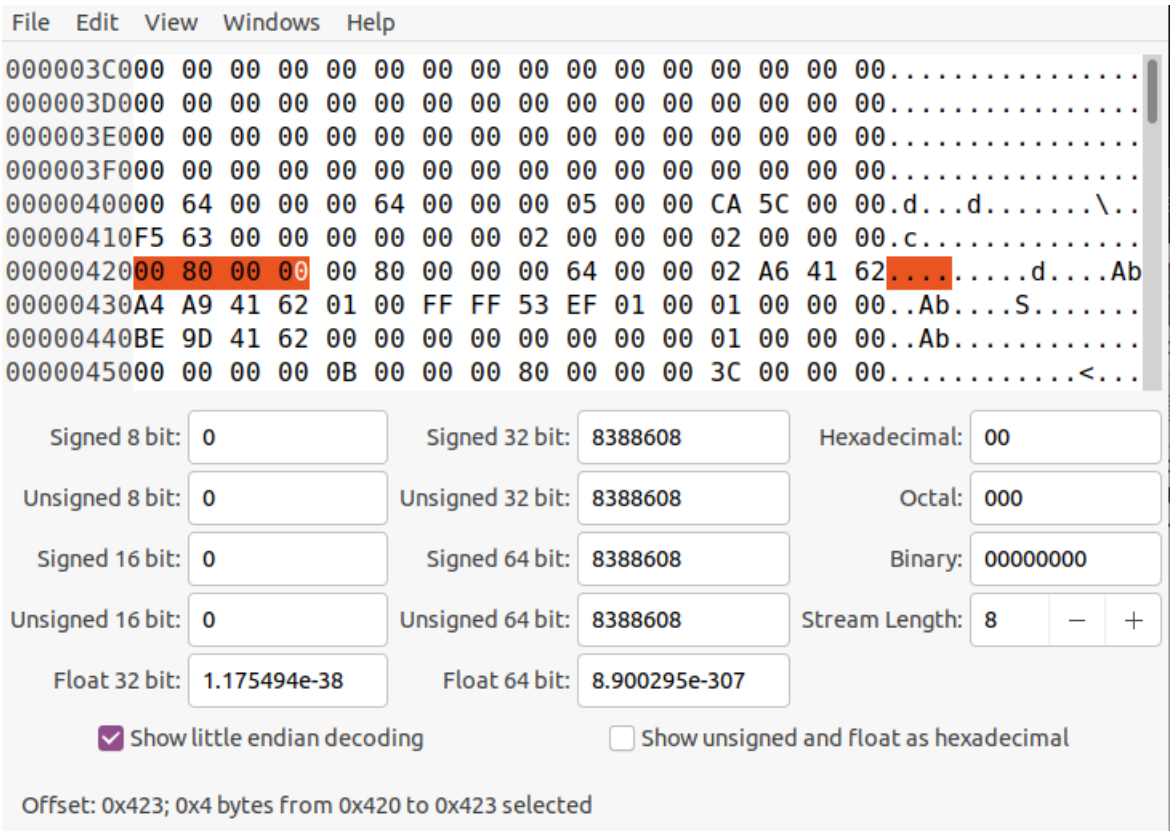
Parametr 2: Liczba i-węzłów, offset 0x0C, wartość: 0 x 00 00 5C CA, DEC: 23 754



Parametr 3: Pierwszy blok z danymi, offset = 0x14, wartość: 0 x 00 00 00 00, DEC: 0



Parametr 4: Liczba bloków w grupie, offset = 0x20, wartość: 0x 00 00 80 00, DEC: 32 768



Zadanie 2

Teraz zamontujemy nasz stworzony system plików. Najpierw stwórzmy urządzenie blokowe wskazujące na plik z systemem plików:

$ sudo losetup --find --show moj-fs.raw

```
sansforensics@siftworkstation: ~
$ sudo losetup --find --show moj-fs.raw
/dev/loop7
```

Wypisaną przez system wartość należy zapamiętać i użyć w kolejnej komendzie. Teraz wykonujemy właściwe montowanie:

$ sudo mount /dev/loop18 /mnt/usb

```
sansforensics@siftworkstation: ~
$ sudo mount /dev/loop7 /mnt/usb
sansforensics@siftworkstation: ~
```

Poprawne wykonanie powyższych komend możemy z grubsza zweryfikować następująco:

$ ls -li /mnt/usb/

```
sansforensics@siftworkstation: ~
$ ls -li /mnt/usb/
total 16
11 drwx------ 2 root root 16384 Mar 28 11:36 lost+found
```

Wypisanie parametrów tego pliku:

$ ls -l /dev/loop18

```
sansforensics@siftworkstation: ~
$ ls -l /dev/loop7
brw-rw---- 1 root disk 7, 7 Mar 28 12:10 /dev/loop7
```

Wyjście z komendy *$ lsblk -f*

```
sansforensics@siftworkstation: ~
$ lsblk -f
NAME FSTYPE LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
loop0
     squash                                                  0   100% /snap/codi
loop1
     squash                                                  0   100% /snap/core
loop2
     squash                                                  0   100% /snap/snap
loop3
     squash                                                  0   100% /snap/core
loop4
     ext4         f8812850-e085-48b8-a34b-bd99a2321b55
loop5
     ext4         f8812850-e085-48b8-a34b-bd99a2321b55
loop6
     ext4         f8812850-e085-48b8-a34b-bd99a2321b55
loop7
     ext4         f8812850-e085-48b8-a34b-bd99a2321b55   85.8M     0% /mnt/usb
sda
├─sda1
     swap         18ce61b9-bd58-4c0c-9b73-c0ddd2e83f5e                [SWAP]
└─sda2
     ext4         8097329e-0ecb-45ce-85c7-2fb00e14a44a  443.9G     2% /
sansforensics@siftworkstation: ~
```

Komenda **lsblk** wypisuje informacje o wszystkich dostępnych lub podanych urządzeniach blokowych.

Z kolei **-f** odpowiada za wypisanie informacji o systemach plików.

Czym jest katalog **lost+found** i dlaczego jego i-węzeł ma numer 11?

Jest to katalog, w którym można znaleźć odzyskane kawałki uszkodzonych plików np. w sytuacji, gdy użytkownik nagle zamknie komputer w czasie, gdy jest on uruchomiony, a pliki są zapisywane na dysku twardym.

Ma numer 11, ponieważ jest to pierwszy i-węzeł dostępny dla użytkownika. W katalogu tym system przechowuje pliki odnalezione podczas wykonywania testów dysku – są to pliki oraz fragmenty „ocalonych” lub uszkodzonych plików w systemie plików.

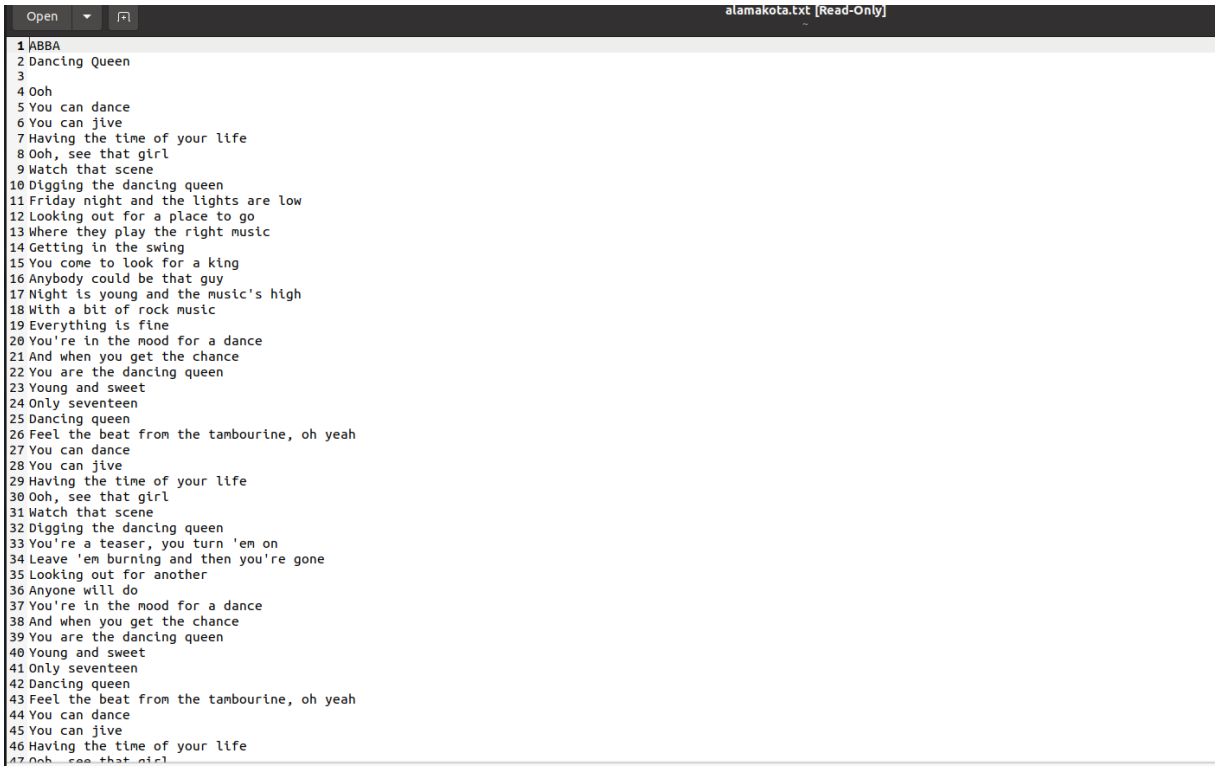
Zadanie 3

Wejdź do katalogu/mnt/usb (tam, gdzie w poprzednim zadaniu zamontowaliśmy system plików). Stwórz w nim plik tekstowy (czysty tekst) – np. za pomocą komendy:

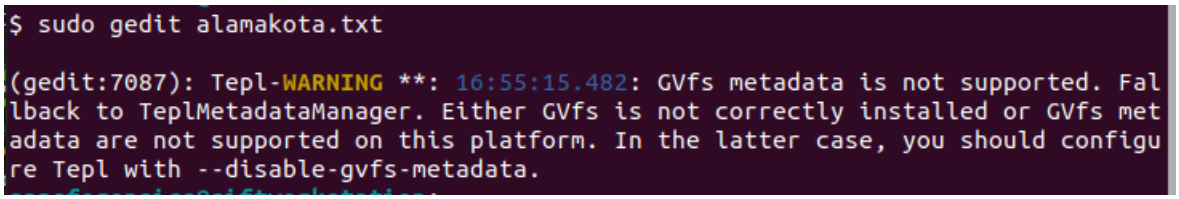
$ sudo gedit alamakota.txt

```
sansforensics@siftworkstation: ~
$ sudo gedit alamakota.txt
```

W pliku tym wprowadź jakiś długi tekst (np. skopiuj treść jakiejś strony internetowej). Zapisz plik i wyjdź z katalogu.

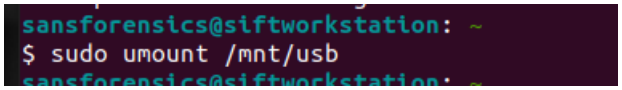


Przy zapisywaniu pliku pojawiło się ostrzeżenie:



Odmontuj system plików:

$ sudo umount /mnt/usb



Po odmontowaniu użyj komendy *debugfs,* aby poznać szczegóły informacje o stworzonym pliku:

Wartości uzyskane po wpisaniu komendy:

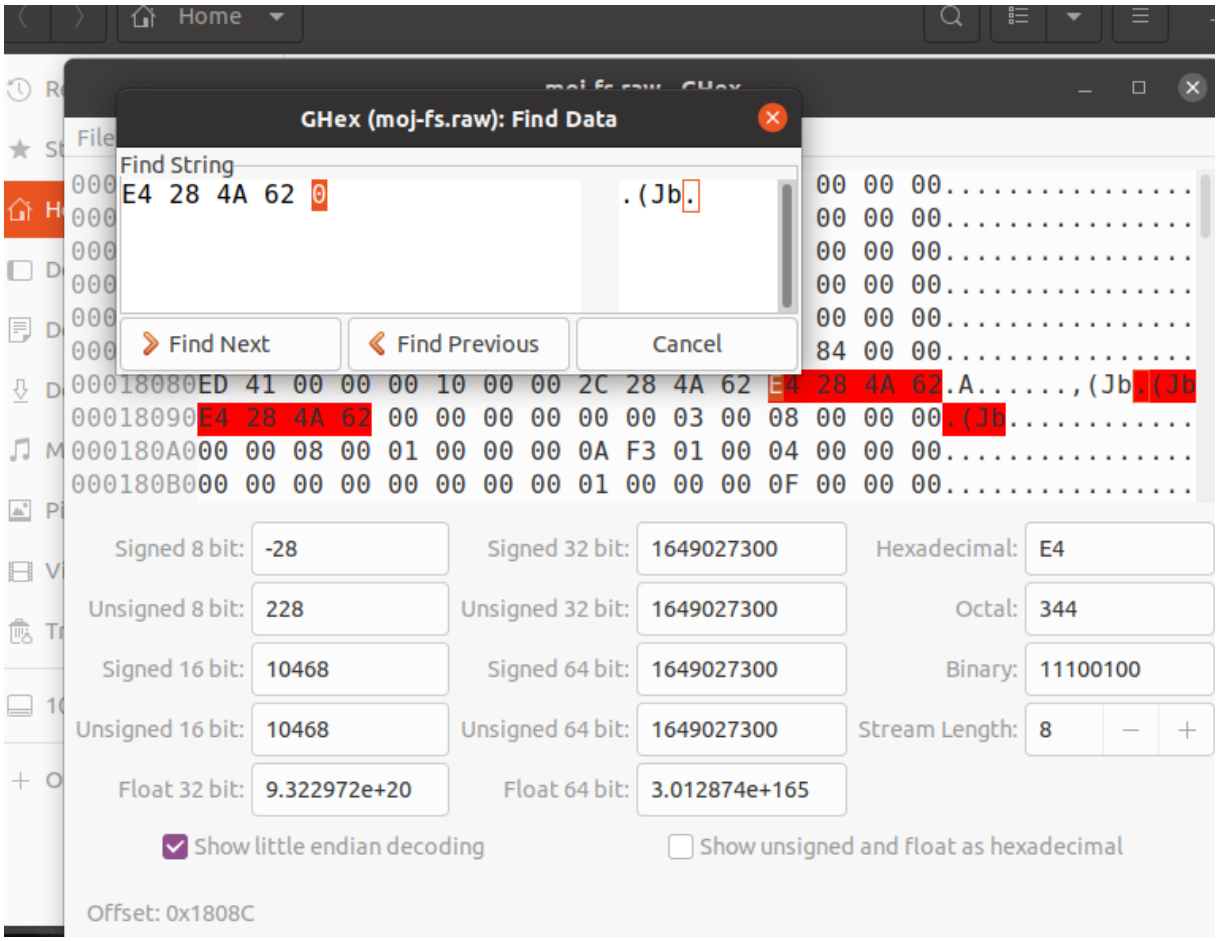
```
Inode: 12   Type: regular    Mode:  0644   Flags: 0x80000
Generation: 1390409572    Version: 0x00000001
User:     0   Group:     0   Size: 1094
File ACL: 0
Links: 1   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x624a28e4 -- Sun Apr  3 23:08:20 2022
atime: 0x624a28f6 -- Sun Apr  3 23:08:38 2022
mtime: 0x624a28e4 -- Sun Apr  3 23:08:20 2022
Inode checksum: 0x00005d84
EXTENTS:
(0):1846
```

Wyświetli to informacje o stworzonym przez Ciebie pliku (np. numer i-węzła). Otwórz ponownie plik moj-fs.raw za pomocą ghex lub innego edytora szesnastkowego. Postaraj się zlokalizować i-węzeł opisujący stworzony przez Ciebie plik.

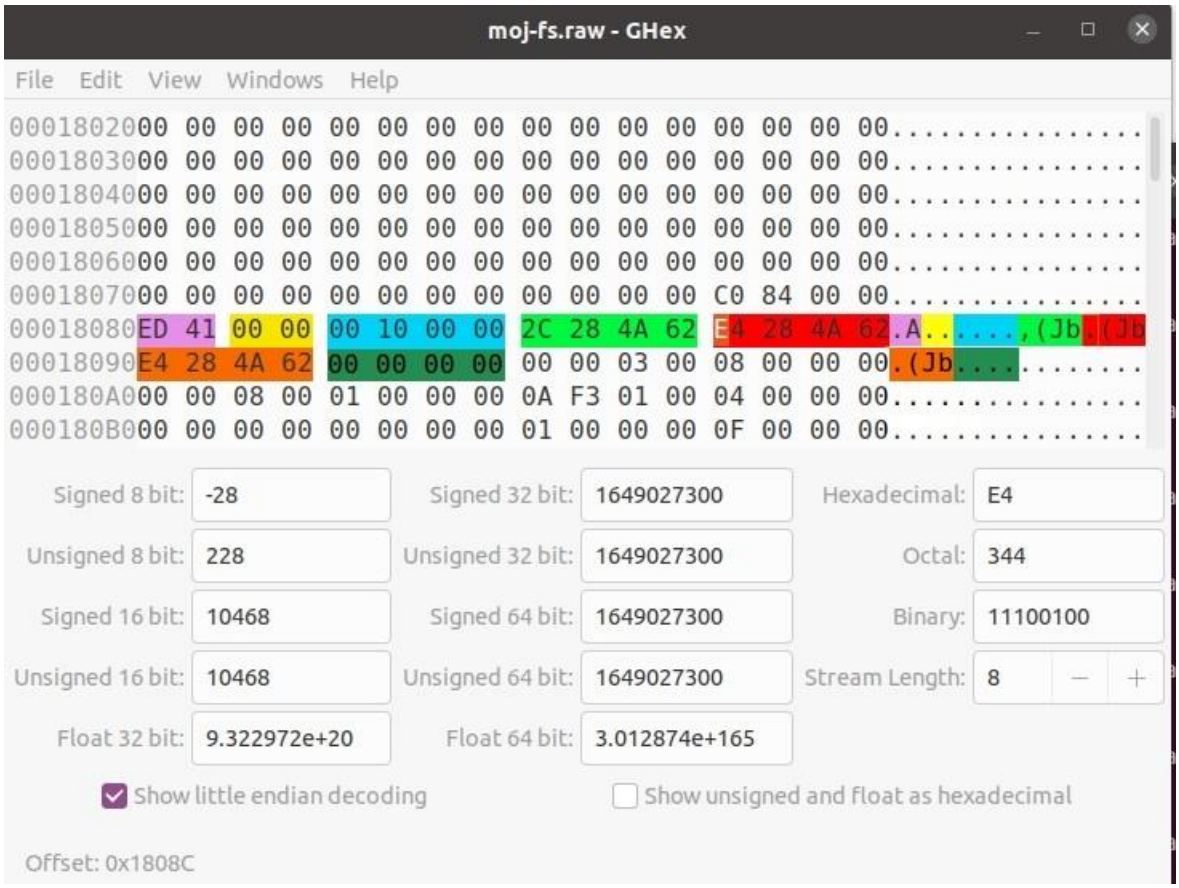
Należy znaleźć i-węzeł Skorzystano z jedynej możliwej danej jaką jest wyświetlana data utworzenia / dostępu / modyfikacji, która została wywołana w komendzie $ debugfs moj-fs.raw, ma ona wartość 0x624a28e4.

Następnie skonwertowano wartość, aby uzyskać niezbędny format, który następnie zostanie użyty w edytorze szesnastkowym (e4 28 4a 62).

Zlokalizowanie i-węzła:



Zaznaczenie odpowiednich bajtów:



Różowy – 2 bajty – tryb pliku

Żółty – 2 bajty – niższe 16 bitów UID właściciela

Niebieski – 4 bajty – niższe 32 bity rozmiaru w bajtach

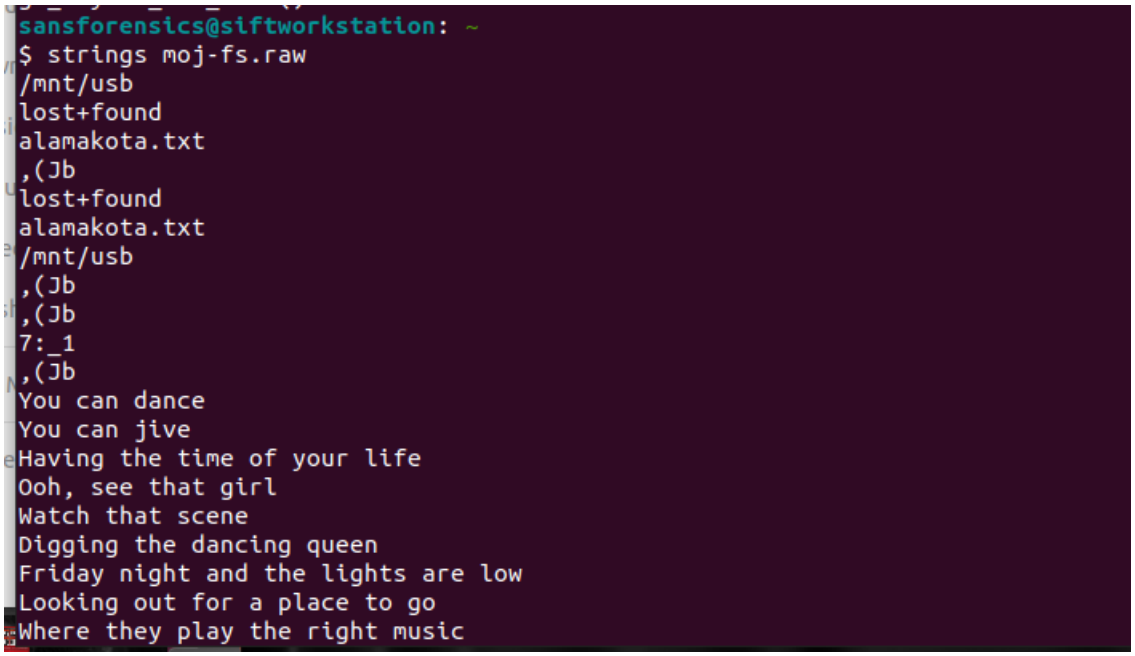
Jasnozielony – 4 bajty – czas dostępu (access time)

Czerwony – 4 bajty – czas zmiany (change time)

Pomarańczowy – 4 bajty – czas modyfikacji (modification time)

Ciemnozielony – 4 bajty – czas usunięcia (deletion time)

Zrzut ekranu z komendy *$strings*


sansforensics@siftworkstation: ~
$ strings moj-fs.raw
/mnt/usb
lost+found
alamakota.txt
,(Jb
lost+found
alamakota.txt
/mnt/usb
,(Jb
,(Jb
7:_1
,(Jb
You can dance
You can jive
Having the time of your life
Ooh, see that girl
Watch that scene
Digging the dancing queen
Friday night and the lights are low
Looking out for a place to go
Where they play the right music