Laboratorium 3 – Szymon Szkarłat

Zadanie 1. – Base64 jako narzędzie do kodowania i dekodowania
Zadanie wstępne.
Kodowanie ASII odbywa się na 8 bitach.
Tabela kodowania ASCII.

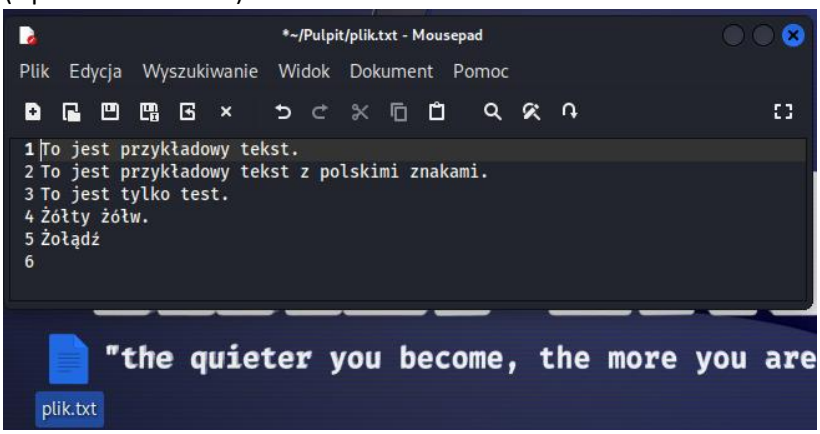
	32	!	33	"	34	#	35	$	36	%	37
&	38	’	39	(	40	)	41	*	42	+	43
,	44	-	45	.	46	/	47	0	48	1	49
2	50	3	51	4	52	5	53	6	54	7	55
8	56	9	57	:	58	;	59	<	60	=	61
>	62	?	63	@	64	A	65	B	66	C	67
D	68	E	69	F	70	G	71	H	72	I	73
J	74	K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84	U	85
V	86	W	87	X	88	Y	89	Z	90	[	91
\	92	]	93	^	94	_	95	‘	96	a	61
b	98	c	99	d	100	e	101	f	102	g	103
h	104	i	105	j	106	k	107	l	108	m	109
n	110	o	111	p	112	q	113	r	114	s	115
t	116	u	117	v	118	w	119	x	120	y	121
z	122	{	123	|	124	}	125	~	126		

Na jej podstawie, oraz postaci binarnej poszczególnych liczb udało mi się dojść do tzw. tekstu jawnego, którym jest „AGH1”.
Kodowanie BASE64 przyjmuje ciąg binarny tekstu jawnego, a następnie dzieli zadany ciąg na segmenty po 6 bitów, które kolejno są szyfrowane, jeżeli liczba bitów podanych na wejściu nie jest podzielna przez 6, to należy na końcu podanego ciągu bitów dodać bity o wartości 0.
Po podzieleniu na 6-bitowe segmenty przechodzimy na system dziesiętny, a następnie przy pomocy tabeli, która przedstawiłem poniżej, odczytujemy znaki.

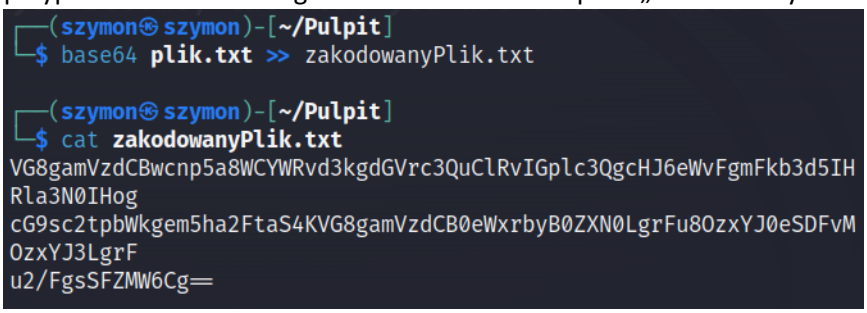
Kod	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Znak	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Kod	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Znak	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
Kod	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
Znak	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
Kod	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
Znak	w	x	y	z	0	1	2	3	4	5	6	7	8	9	+	/

Otrzymany szyfrogram to: „QUdIMQ”, potwierdza to screen umieszczony na kolejnej stronie raportu.

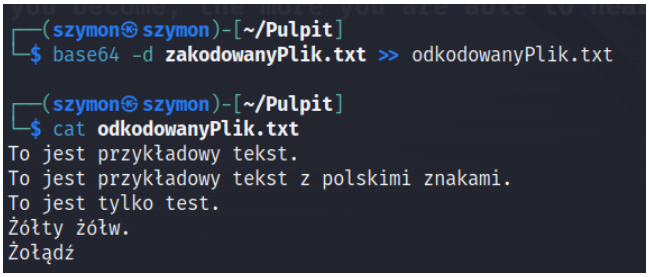
Dalsze zadania.
Utworzenie na pulpicie pliku z rozszerzeniem .txt oraz zamieszczenie w nim kilku przykładowych zdań (z polskimi znakami).



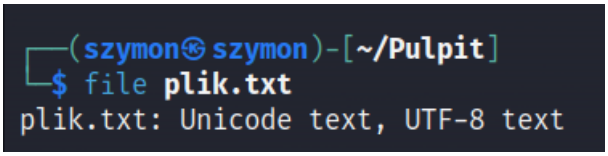
Użycie narzędzia base64 do zakodowania utworzonego pliku tekstowego o nazwie „plik.txt” i przypisanie zakodowanego łańcucha znaków do pliku „zakodowanyPlik.txt”.



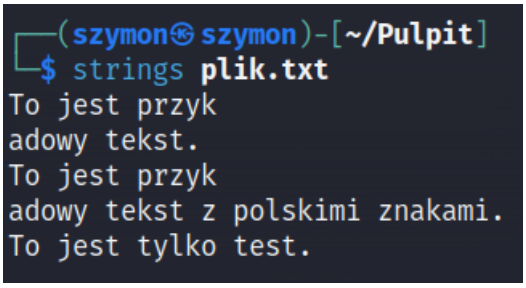
Użycie polecenia: base64 -d do odkodowania zawartości pliku „zakodowanyPlik.txt”. Jak przedstawiono na poniższymi sreenie otrzymaliśmy ten sam tekst co przed zakodowaniem.



System kodowania pliku „plik.txt” to UTF-8 (polecenie: *file plik.txt*)

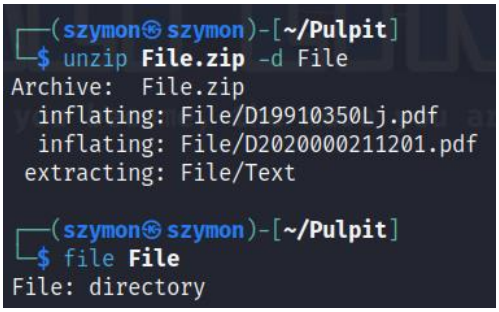


Użycie polecenia *strings.*

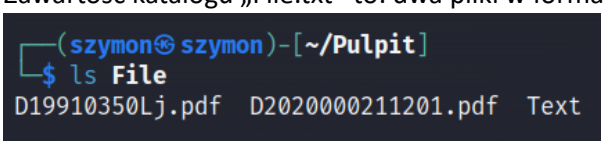


Polecenie *strings* służy do wyświetlenia zawartości pliku o nazwie „plik.txt”, nie wyświetlane są polskie znaki. Natomiast komenda *file* umożliwia sprawdzenie kodowania pliku, tj. UTF-8 (które to w teorii obsługuje polskie znaki). Dlatego to zastały one w sposób prawidłowy zapisane. Jednak jak wspomniano wcześniej, *strings* nie wyświetla polskich znaków.

Zadanie 2. – W zakładce pliki do przedmiotu Informatyka Śledcza znajduje się katalog „File.zip”, który zawiera przykładowe pliki o różnych rozszerzeniach.
Rozpakowanie katalogu „File.zip” w systemie Linux oraz ustalenie zawartości wypakowanego archiwum do katalogu „File”.



Zawartość katalogu „File.txt” to: dwa pliki w formacie PDF oraz plik o nazwie „Text”.



Instalacja pdfinfo

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo apt-get install poppler-utils
Czytanie list pakietów... Gotowe
```

Użycie programu pdfinfo

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ pdfinfo File/D19910350Lj.pdf
Title:          Akt prawny
Author:         Władysław Baksza
Creator:        Microsoft® Word 2013
Producer:       Microsoft® Word 2013
CreationDate:   Tue Oct 12 13:08:08 2021 CEST
ModDate:        Tue Oct 12 13:08:08 2021 CEST
Custom Metadata: no
Metadata Stream: no
Tagged:         yes
UserProperties: no
Suspects:       no
Form:           none
JavaScript:     no
Pages:          351
Encrypted:      no
Page size:      595.32 x 841.92 pts (A4)
Page rot:       0
File size:      2601654 bytes
Optimized:      no
PDF version:    1.5
```

Informacje o plikach PDF:

a) D19910350Lj.pdf

- tytuł wiadomości: „Akt prawny”
- data utworzenia pliku: 12.10.2021 13:08 (wtorek)
- liczba stron: 351
- wielkość stron: 595.32 x 841.92 pts (A4)
- plik nie zawiera Java Scipt
- Autor: Władysław Baksza
- Użyte oprogramowanie: Microsoft® Word 2013

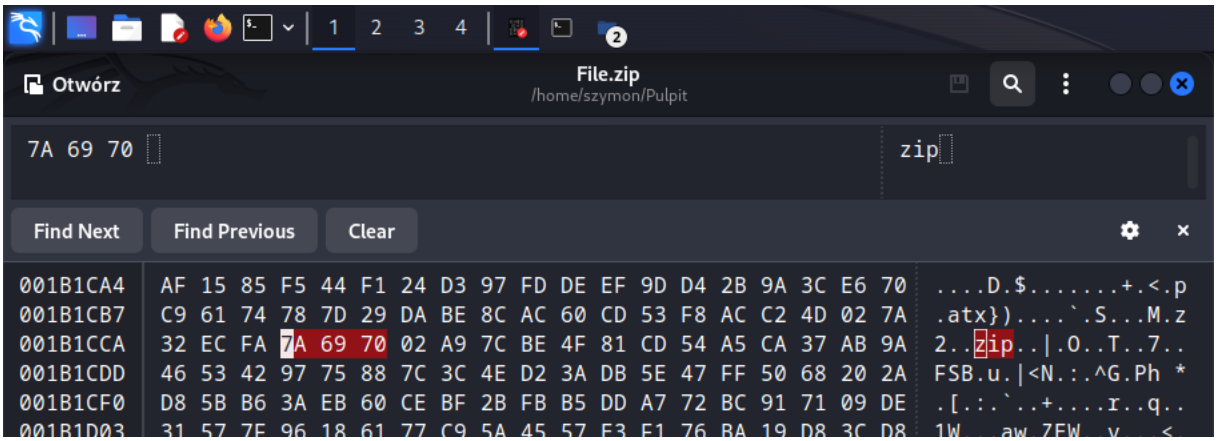
b) D2020000211201.pdf

- tytuł wiadomości: „Ustawa z dnia 28 października 2020 r. o zmianie niektórych ustaw w związku z przeciwdziałaniem sytuacjom kryzysowym związanym z wystąpieniem COVID-19”
- data utworzenia pliku: 28.11.2020 18:40 (sobota)
- liczba stron: 18
- wielkość stron: 595.32 x 841.92 pts (A4)
- plik nie zawiera Java Scipt
- Autor: RCL
- Użyte oprogramowanie: Microsoft® Word 2010

Zadanie 3. – Właściwości narzędzia GHex

Zainstalowanie narzędzia GHex w terminalu Linux

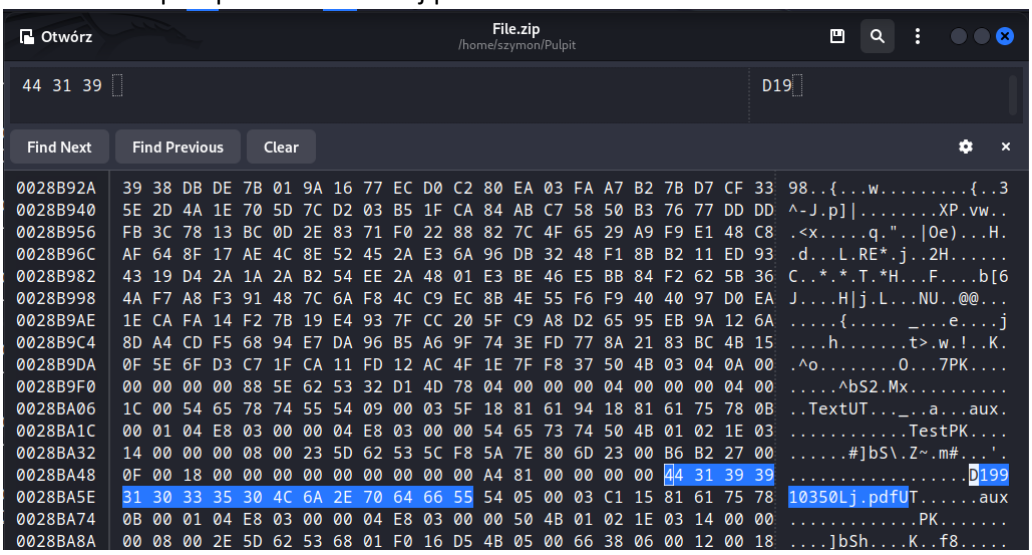
```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo apt-get install ghex
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
```



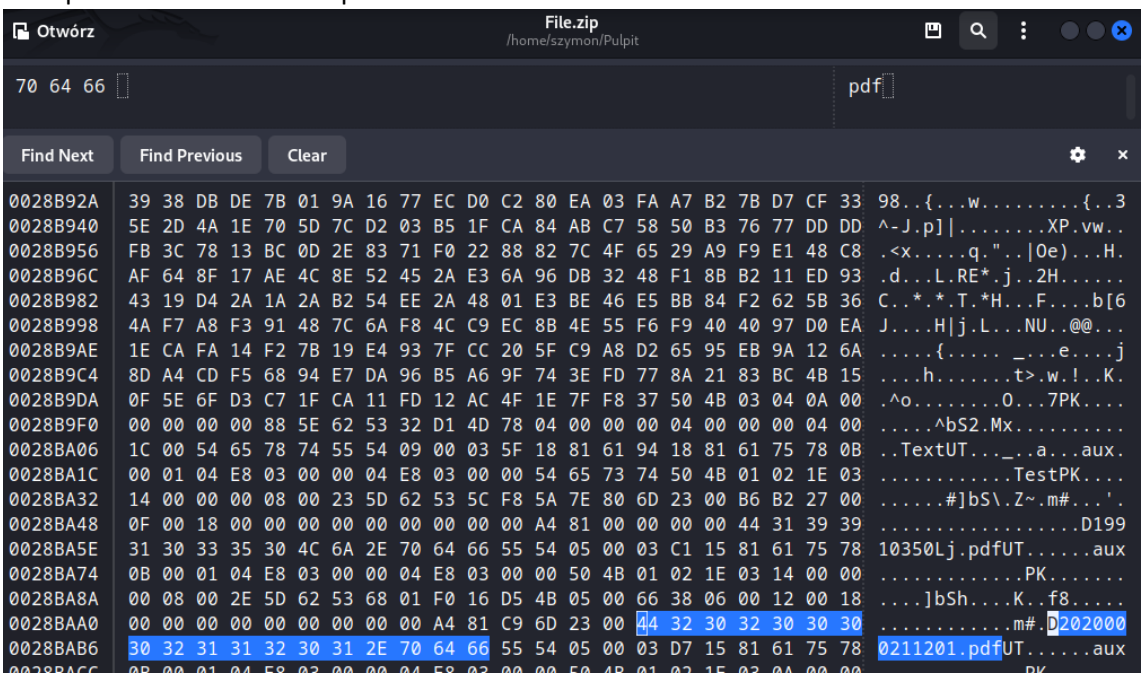
Informacje o pliku zawiera zaznaczony fragment wiersza, sygnatura "7A 69 70", czyli zip co potwierdza, że jest to rozszerzenie .zip archiwum.

7A, 69, 70 to w kodowaniu ASCII odpowiednio z, i, p.

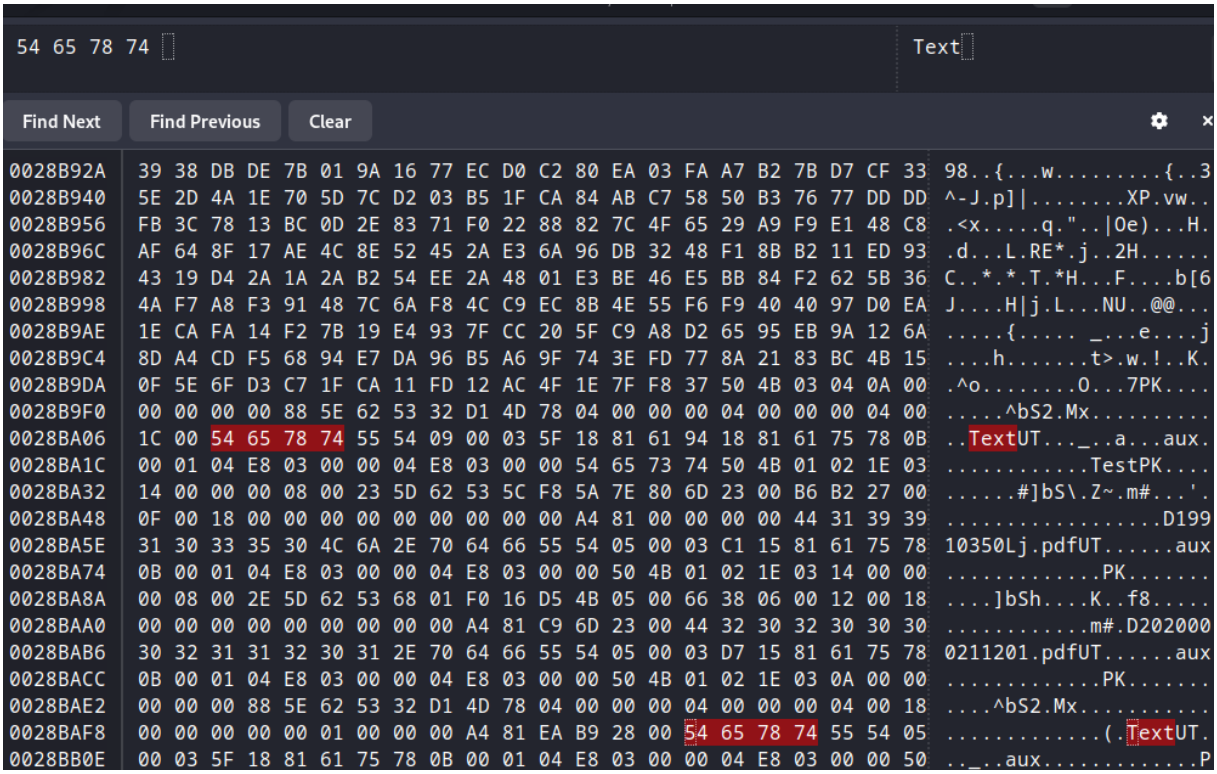
Odnalezione pliki pdf: D19910350Lj.pdf



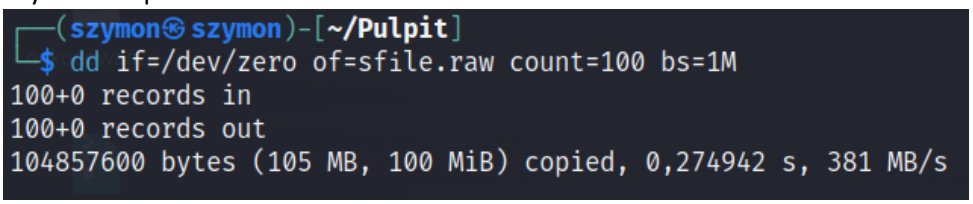
Oraz plik: D2020000211201.pdf



Onalezionie również informacji o pliku „Text”.



Zadanie 4. – Różnice pomiędzy systemami plików?
Wykonanie polecenia.



Po zweryfikowaniu zawartości pliku sfile.raw przy pomocy narzędzia ghex.

Plik ten ma same sygnatury 0, jest pusty (jeśli chodzi o zawartość).
Natomiast plik poprzednio badany miał sygnatury., nie był pusty (jeśli chodzi o zawartość).
Po wykonaniu komendy, nastąpiła wyraźna zmiana w pliku sfile.raw.

Wykonanie polecenia *mkfs.fat*.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ mkfs.fat sfile.raw
mkfs.fat 4.2 (2021-01-31)

┌──(szymon㉿szymon)-[~/Pulpit]
└─$ ghex sfile.raw

(process:49452): Gtk-WARNING **: 16:45:29.857: Theme parser error: gtk.css:3977:3-22: No prop
erty named "-gtk-outline-radius"
```

Zmiany w pliku po wykonaniu polecenia mkfs.fat na pliku „sfile.raw” wyświetlone za pomocą narzędzia ghex.

Otwórz — sfile.raw — /home/szymon/Pulpit

```
00000000  EB 3C 90 6D 6B 66 73 2E 66 61 74 00 02 04 04 00 02  .<.mkfs.fat......
00000011  00 02 00 00 F8 C8 00 20 00 08 00 00 00 00 00 00 20  ....... ........
00000022  03 00 80 00 29 C3 69 A1 00 4E 4F 20 4E 41 4D 45 20  ....).i..NO NAME
00000033  20 20 20 46 41 54 31 36 20 20 20 0E 1F BE 5B 7C AC     FAT16   ...[|.
00000044  22 C0 74 0B 56 B4 0E BB 07 00 CD 10 5E EB F0 32 E4  ".t.V.......^..2.
00000055  CD 16 CD 19 EB FE 54 68 69 73 20 69 73 20 6E 6F 74  ......This is not
00000066  20 61 20 62 6F 6F 74 61 62 6C 65 20 64 69 73 6B 2E   a bootable disk.
00000077  20 20 50 6C 65 61 73 65 20 69 6E 73 65 72 74 20 61    Please insert a
00000088  20 62 6F 6F 74 61 62 6C 65 20 66 6C 6F 70 70 79 20   bootable floppy
00000099  61 6E 64 0D 0A 70 72 65 73 73 20 61 6E 79 20 6B 65  and..press any ke
000000AA  79 20 74 6F 20 74 72 79 20 61 67 61 69 6E 20 2E 2E  y to try again ..
000000BB  2E 20 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 00  . ...............
000000CC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Kolejno zmiana w utworzonym pliku na system plików ext4.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ mkfs.ext4 sfile.raw
mke2fs 1.46.6 (1-Feb-2023)
sfile.raw contains a vfat file system
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: 999f5b48-c5b4-4c83-b0a2-a84b1ae81c59
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Gdzie utworzono superbloki?

Superbloki utworzono w blokach: 8193, 24577, 40961, 57345, 73729.

Użycie narzędzia dumpe2fs.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ dumpe2fs sfile.raw
dumpe2fs 1.46.6 (1-Feb-2023)
Filesystem volume name:   <none>
```

Wyświetlenie najważniejszych informacji:

- „magiczny numer” badanego obrazu: 0xEF53
- numer UUID: 999f5b48-c5b4-4c83-b0a2-a84b1ae81c59
- wielkość bloku: 1024
- liczba wolnych bloków: 90319
- checksum type: crc32c
- liczba wolnych bloków, w grupie nr 12: 4094, bo przedział 98305-102399

```
Group 12: (Blocks 98305-102399) csum 0x59e1 [INODE_UNINIT, ITABLE_ZEROED]
  Block bitmap at 271 (bg #0 + 270), csum 0x3b046629
  Inode bitmap at 284 (bg #0 + 283), csum 0x00000000
  Inode table at 6189-6680 (bg #0 + 6188)
  4095 free blocks, 1968 free inodes, 0 directories, 1968 unused inodes
  Free blocks: 98305-102399
  Free inodes: 23617-25584
```

Użycie komendy *fsck* w celu sprawdzenia ilości zużytych bloków.
Można to sprawdzić, przy pomocy komendy: *fsck.ext4 sfile.raw.*

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ fsck.ext4 sfile.raw
e2fsck 1.46.6 (1-Feb-2023)
sfile.raw: clean, 11/25584 files, 12081/102400 blocks
```

Lub użyć komendy: *fsck -fvy sfile.raw*

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ fsck.ext4 -fvy sfile.raw
e2fsck 1.46.6 (1-Feb-2023)
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

        12 inodes used (0.05%, out of 25584)
         0 non-contiguous files (0.0%)
         0 non-contiguous directories (0.0%)
           # of inodes with ind/dind/tind blocks: 0/0/0
           Extent depth histogram: 4
     12081 blocks used (11.80%, out of 102400)
         0 bad blocks
         0 large files
```

Procent zużytych bloków to 11.8%.

Zadanie 5. – Znaczenie superbloków w odzyskiwaniu danych.
Zamontowanie w systemie przygotowanego obrazu .raw

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo losetup --find --show sfile.raw
/dev/loop0
```

Zamontowanie pliku.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo mount /dev/loop0 /mnt/hgfs
```

Sprawdzenie uprawnień, przy pomocy komendy *ls -la*.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ ls -li /mnt/hgfs
razem 12
11 drwx------ 2 root root 12288 11-10 19:02 lost+found
```

Pełne uprawnienia jedynie dla twórcy katalogu (właściciela).
Katalog lost+found służy do odzyskiwanie plików, które nie zostały prawidłowo zamknięte z powodu chociażby awarii. Każdy plik do odzyskiwania jest przechowywyany w tym folderze. Do odzyskiwania tych plików służy polecenie *fsck* (file system check).

Posiadając informację na temat nr super bloków utworzenie w zamontowanym obrazie nowego pliku, który kolejno będziemy odzyskiwać.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ cd /mnt/hgfs/

┌──(szymon㉿szymon)-[/mnt/hgfs]
└─$ sudo touch newitem

┌──(szymon㉿szymon)-[/mnt/hgfs]
└─$ ls
lost+found  newitem
```

Wykorzystanie polecenia dd do wyczyszczenia danych i zastąpienia ich zerami

```
┌──(szymon㉿szymon)-[/mnt/hgfs]
└─$ sudo dd if=/dev/zero of=/dev/loop0 count=1 bs=1024 seek=1
1+0 records in
1+0 records out
1024 bytes (1,0 kB, 1,0 KiB) copied, 0,00173302 s, 591 kB/s
```

Nowo utworzony plik nie znajduje się w katalogu, katalog z zamontowanym obrazem jest pusty.

```
┌──(szymon㉿szymon)-[/mnt/hgfs]
└─$ ls -la
razem 0
```

Odmontowanie obrazu, problem nie wystąpił.

```
┌──(szymon㉿szymon)-[/mnt/hgfs]
└─$ cd ~/Pulpit

┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo umount /dev/loop0
```

Odmontowanie zakończone sukcesem.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ df -k
System plików     1K-bl    użyte dostępne %uż. zamont. na
udev             4010348        0  4010348   0% /dev
tmpfs             810188     1496   808692   1% /run
/dev/sda1       29801344 14440244 13821928  52% /
tmpfs            4050932        0  4050932   0% /dev/shm
tmpfs               5120        0     5120   0% /run/lock
tmpfs             810184       80   810104   1% /run/user/1000
```

W trakcie ponownego montowania pojawił się problem.

```
┌──(szymon㉿szymon)-[/mnt]
└─$ sudo mount /dev/loop1 /mnt/hgfs
mount: /mnt/hgfs: wrong fs type, bad option, bad superblock on /dev/loop1, missing codepage o
r helper program, or other error.
       dmesg(1) may have more information after failed mount system call.
```

Problem ten udało się rozwiązać, poprzez sprawdzenie kopii zapasowej superbloków.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo dumpe2fs -o superblock=8193 /dev/loop1 | grep "Backup superblock"

dumpe2fs 1.46.6 (1-Feb-2023)
  Backup superblock at 8193, Group descriptors at 8194-8194
  Backup superblock at 24577, Group descriptors at 24578-24578
  Backup superblock at 40961, Group descriptors at 40962-40962
  Backup superblock at 57345, Group descriptors at 57346-57346
  Backup superblock at 73729, Group descriptors at 73730-73730
```

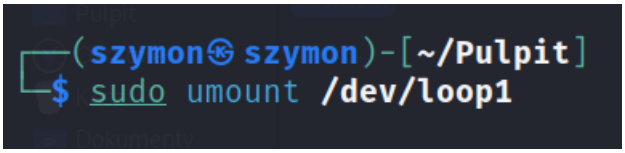
Oraz zamontowanie w oparciu o nr konkretnego superbloku, który posłużył jako kopia zapasowa.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo mount -o sb=8193 /dev/loop1 /mnt/hgfs

┌──(szymon㉿szymon)-[~/Pulpit]
└─$ df -k
System plików     1K-bl    użyte dostępne %uż. zamont. na
udev             4010352        0  4010352   0% /dev
tmpfs             810188     1496   808692   1% /run
/dev/sda1       29801344 14446016 13816156  52% /
tmpfs            4050936        0  4050936   0% /dev/shm
tmpfs               5120        0     5120   0% /run/lock
tmpfs             810184       88   810096   1% /run/user/1000
/dev/loop1         94429       14    89295   1% /mnt/hgfs
```

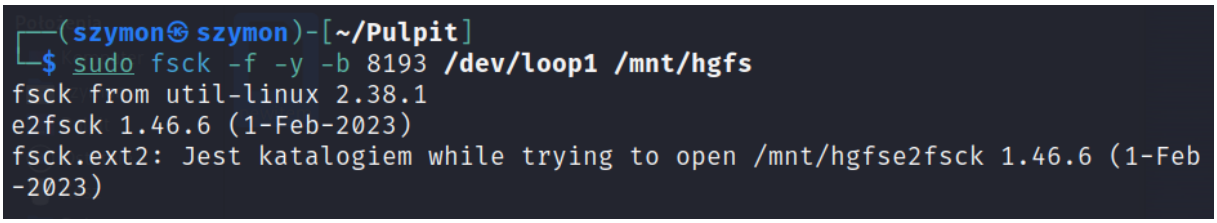
Polecenie nie działa prawidłowo, należy odmontować.

```
┌──(szymon㉿szymon)-[~/Pulpit]
└─$ sudo fsck -f -y -b 8193 /dev/loop1 /mnt/hgfs
fsck from util-linux 2.38.1
e2fsck 1.46.6 (1-Feb-2023)
e2fsck 1.46.6 (1-Feb-2023)
fsck.ext2/dev/loop1 is mounted.
: e2fsck: Cannot continue, aborting.


Jest katalogiem while trying to open /mnt/hgfs

The superblock could not be read or does not describe a valid ext2/ext3/ext4
filesystem.  If the device is valid and it really contains an ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>
 or
    e2fsck -b 32768 <device>
```

Odmontowanie i ponowne wykonanie polecenia *fsck*.



Ponowne wykonanie polecenia



Wyświetlenie rezultatu z odzyskanych danych. Jak przedstawiono na poniższym screenie, udało się odzyskać skasowana dane.