



МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))

Институт транспортной техники и систем управления
Кафедра «Управление и защита информации»

Отчет по лабораторной работе №1
«Сравнение энтропии распакованных и упакованных файлов»
по дисциплине
«Технология реверс-инжиниринга»

Выполнили: студенты ТКИ-341
Козлов А. Д. и Дьячков Д.О.
Проверили: Профессор “УиЗИ” Сидоренко В.Г.
Профессор “УиЗИ” Сафонов А. И.

Москва 2025 г.

ОГЛАВЛЕНИЕ

ЦЕЛЬ РАБОТЫ.....	3
ПОСТАВЛЕННЫЕ ЗАДАЧИ	3
ПРЕДСТАВЛЕНИЯ РАЗЛИЧНЫХ ФАЙЛОВ И ПРОГРАММ В НЕХ-РЕДАКТОРЕ	4
СРАВНЕНИЕ ЭНТРОПИИ РАСПАКОВАННЫХ И УПАКОВАННЫХ ИСПОЛНЯЕМЫХ ФАЙЛОВ ПРОГРАММ	7
ГИСТОГРАММЫ И ЭНТРОПИЯ С++	7
ГИСТОГРАММЫ И ЭНТРОПИЯ С#.....	10
ГИСТОГРАММЫ И ЭНТРОПИЯ РУТНОН.....	12
СРАВНЕНИЕ ЭНТРОПИИ АУДИОФАЙЛОВ РАЗНЫХ ФОРМАТОВ .	15
СРАВНЕНИЕ ЭНТРОПИИ ФАЙЛОВ ИЗОБРАЖЕНИЙ РАЗНЫХ ФОРМАТОВ	18
СРАВНЕНИЕ ЭНТРОПИИ ТЕКСТОВЫХ ФАЙЛОВ РАЗНЫХ ФОРМАТОВ.....	21
СРАВНЕНИЕ ЭНТРОПИИ ФАЙЛОВ ТАБЛИЦ РАЗНЫХ ФОРМАТОВ	24
ТАБЛИЦЫ СРАВНЕНИЙ ЭНТРОПИИ	28
ВЫВОД.....	30

Цель работы

Исследование файлов разных форматов и разной степенью сжатия с целью изучения их структуры и определения влияния сжатия на энтропию этих файлов.

Поставленные задачи

1. Анализ файлов в hex-редакторе. Выбрать несколько файлов разных типов близкого размера (изображение, музыка, таблица, текстовый файл, тексты программ, реализующих одни и те же действия, написанные на разных языках программирования, и их исполняемые файлы), просмотреть их в hex-редакторе.
2. Сравнение энтропии распакованных и упакованных исполняемых файлов программ, а также сравнение файлов одного типа в разных форматах данных. Упаковать выбранный файл с использованием UPX, просмотреть его в hex-редакторе, сравнить результаты, полученные в п.п.1 и 2. При помощи программы radare2, NIST, или другого ПО рассчитать энтропию исходного и упакованного файлов, провести анализ полученных результатов.

Представления различных файлов и программ в hex-редакторе

000000000	7F 45 4C 46 02 01 01 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ELF.....
000000010	03 00 3E 00 01 00 00 00	C0 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.>....L".....
000000020	40 00 00 00 00 00 00 00	28 56 00 00 00 00 00 00 00 00 00 00 00 00 00 00	@.....(V.....
000000030	00 00 00 00 40 00 38 00	0D 00 40 00 20 00 1F 00@.8...@....
000000040	06 00 00 00 04 00 00 00	40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00@.....
000000050	40 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	@.....@.....
000000060	D8 02 00 00 00 00 00 00	D8 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00	+.+.+.+
000000070	08 00 00 00 00 00 00 00	03 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00
000000080	18 03 00 00 00 00 00 00	18 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000090	18 03 00 00 00 00 00 00	1C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000A0	1C 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000B0	01 00 00 00 04 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Рис 1. Представление исполняемого файла c++ в hex-редакторе

000000000	7F 45 4C 46 02 01 01 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ELF.....
000000010	03 00 3E 00 01 00 00 00	50 23 2D 00 00 00 00 00 00 00	.>....P#-.....
000000020	40 00 00 00 00 00 00 00	90 97 A7 00 00 00 00 00 00 00	@.....Éù°.....
000000030	00 00 00 00 40 00 38 00	0C 00 40 00 26 00 24 00@.8...@.&.\$
000000040	06 00 00 00 04 00 00 00	40 00 00 00 00 00 00 00 00 00@.....
000000050	40 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00 00 00	@.....@.....
000000060	A0 02 00 00 00 00 00 00	A0 02 00 00 00 00 00 00 00 00	á.....á.....
000000070	08 00 00 00 00 00 00 00	03 00 00 00 04 00 00 00 00 00
000000080	E0 02 00 00 00 00 00 00	E0 02 00 00 00 00 00 00 00 00	α.....α.....
000000090	E0 02 00 00 00 00 00 00	1C 00 00 00 00 00 00 00 00 00	α.....
0000000A0	1C 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00 00 00
0000000B0	01 00 00 00 04 00 00 00	00 00 00 00 00 00 00 00 00 00

Рис 2. Представление исполняемого файла c# в hex-редакторе

000000000	7F 45 4C 46 02 01 01 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ELF.....
000000010	02 00 3E 00 01 00 00 00	40 1B 40 00 00 00 00 00 00 00	.>....@. @.....
000000020	40 00 00 00 00 00 00 00	D0 84 71 00 00 00 00 00 00 00	@.....äq.....
000000030	00 00 00 00 40 00 38 00	09 00 40 00 1D 00 1C 00@.8...@.....
000000040	06 00 00 00 04 00 00 00	40 00 00 00 00 00 00 00 00 00@.....
000000050	40 00 40 00 00 00 00 00	40 00 40 00 00 00 00 00 00 00	@. @.....@. @.....
000000060	F8 01 00 00 00 00 00 00	F8 01 00 00 00 00 00 00 00 00	°.....°.....
000000070	08 00 00 00 00 00 00 00	03 00 00 00 04 00 00 00 00 00
000000080	38 02 00 00 00 00 00 00	38 02 40 00 00 00 00 00 00 00	8.....8. @.....
000000090	38 02 40 00 00 00 00 00	1C 00 00 00 00 00 00 00 00 00	8. @.....
0000000A0	1C 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00 00 00
0000000B0	01 00 00 00 05 00 00 00	00 00 00 00 00 00 00 00 00 00

Рис 3. Представление исполняемого файла python в hex-редакторе

00000000	52 49 46 46 E4 24 EF 03	57 41 56 45 66 6D 74 20	RIFF\$n.WAVEfmt
00000010	10 00 00 00 01 00 02 00	44 AC 00 00 10 B1 02 00D\.....
00000020	04 00 10 00 64 61 74 61	C0 24 EF 03 3B 08 D5 0Adata\\$n.;.F.
00000030	0E 05 EF 04 58 09 35 07	9C 06 72 05 DD 05 A0 08	.n.X.5.E.r. .á.
00000040	E5 4C E7 4E 83 56 08 58	3B 1B 63 26 54 88 C5 95	σLτNâV.X;.c&Tê ò
00000050	25 82 71 89 27 6A C7 66	8A 7E 8A 7E A8 28 E5 27	%éqë'j fè~è~{σ'
00000060	C1 84 7F 81 76 81 76 81	F2 F6 90 EB F2 7D 9D 7C	Łäöüñvñvü÷ÉDz}¥
00000070	87 7E 85 7E 94 6D E9 68	99 54 6D 62 07 20 33 32	ç~à~ömθhÖTmb. 32
00000080	C9 E0 9A ED 7A 81 6D 82	75 81 75 81 81 81 22 83	¶aÜφzüméuüüüü"å
00000090	35 8A 9B 85 78 A4 A0 9C	08 B2 BD AF 46 CF 9D C6	5ètàxñá£. »F ¥
000000A0	68 F0 03 E5 91 4F D0 44	E1 65 55 67 93 40 70 43	h=.σæO DBeUgô@pC
000000B0	CF 54 BE 53 32 4C 72 56	8B 48 86 54 27 47 C6 4E	ŁT S2LrViHåT'G N

Рис 4. Представление аудиофайла в hex-редакторе

00000000	89 50 4E 47 0D 0A 1A 0A	00 00 00 0D 49 48 44 52	éPNG.....IHDR
00000010	00 00 09 60 00 00 06 40	08 06 00 00 00 FD A9 79	...`....@.....²γy
00000020	B1 00 00 00 04 67 41 4D	41 00 00 B1 8F 0B FC 61gAMA. Å.na
00000030	05 00 00 00 01 73 52 47	42 00 AE CE 1C E9 00 00sRGB. «†.0..
00000040	00 06 62 4B 47 44 00 FF	00 FF 00 FF A0 BD A7 93	..bKGD. . . á°ôô
00000050	00 00 00 09 70 48 59 73	00 00 0B 12 00 00 0B 12pHYs.....
00000060	01 D2 DD 7E FC 00 00 00	01 6F 72 4E 54 01 CF A2	.T~n....orNT.Łó
00000070	77 9A 00 00 00 25 74 45	58 74 64 61 74 65 3A 63	wÜ...%tEXtdate:c
00000080	72 65 61 74 65 00 32 30	32 35 2D 30 32 2D 31 37	reate.2025-02-17
00000090	54 31 38 3A 35 38 3A 31	39 2B 30 30 3A 30 30 B5	T18:58:19+00:00
000000A0	40 3F 15 00 00 00 25 74	45 58 74 64 61 74 65 3A	@?....%tEXtdate:
000000B0	6D 6F 64 69 66 79 00 32	30 32 35 2D 30 32 2D 31	modify.2025-02-1

Рис 5. Представление файла изображения в hex-редакторе

Рис 6. Представление текстового файла формата .docx в hex-редакторе

00000000	25	50 44 46 2D 31 2E 35	0D 0A 25 B5 B5 B5 B5 0D	%PDF-1.5..%+==.
00000010	0A	31 20 30 20 6F 62 6A	0D 0A 3C 3C 2F 54 79 70	.1 0 obj..<</Typ
00000020	65	2F 43 61 74 61 6C 6F	67 2F 50 61 67 65 73 20	e/Catalog/Pages
00000030	32	20 30 20 52 2F 4C 61	6E 67 28 65 6E 2D 55 53	2 0 R/Lang(en-US
00000040	29	20 2F 53 74 72 75 63	74 54 72 65 65 52 6F 6F) /StructTreeRoo
00000050	74	20 31 30 20 30 20 52	2F 4D 61 72 6B 49 6E 66	t 10 0 R/MarkInf
00000060	6F	3C 3C 2F 4D 61 72 6B	65 64 20 74 72 75 65 3E	o<</Marked true>
00000070	3E	3E 0D 0A 65 6E 64	6F 62 6A 0D 0A 32 20 30	>>..endobj..2 0
00000080	20	6F 62 6A 0D 0A 3C 3C	2F 54 79 70 65 2F 50 61	obj..<</Type/Pa
00000090	67	65 73 2F 43 6F 75 6E	74 20 31 2F 4B 69 64 73	ges/Count 1/Kids
000000A0	5B	20 33 20 30 20 52 5D	20 3E 0D 0A 65 6E 64	[3 0 R] >>..end
000000B0	6F	62 6A 0D 0A 33 20 30	20 6F 62 6A 0D 0A 3C 3C	obj..3 0 obj..<<

Рис 7. Представление текстового файла формата .pdf в hex-редакторе

00000000	49	27 6D 20 74 65 78 74	20 66 69 6C 65 2E 20 46	I'm text file. F
00000010	69	6E 64 20 6D 79 20 65	6E 74 72 6F 70 79 21 +	ind my entropy!

Рис 8. Представление текстового файла формата .txt в hex-редакторе

00000000	D0	CF 11 E0 A1 B1 1A E1	00 00 00 00 00 00 00 00 00ai.....B.....
00000010	00	00 00 00 00 00 00 00 00	3E 00 03 00 FE FF 09 00>.... . .
00000020	06	00 00 00 00 00 00 00 00	00 00 00 00 01 00 00 00
00000030	35	00 00 00 00 00 00 00 00	00 10 00 00 FE FF FF FF	5.....
00000040	00	00 00 00 FE FF FF FF	00 00 00 00 34 00 00 00
00000050	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF	5.....
00000060	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
00000070	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
00000080	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
00000090	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
000000A0	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
000000B0	FF	FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF

Рис 9. Представление файла таблицы формата xls в hex-редакторе

Сравнение энтропии распакованных и упакованных исполняемых файлов программ

Для анализа исполняемых файлов были подготовлены и скомпилированы программы, написанные на языках c++, c# и python, а также были подготовлены упакованные версии этих программ с использованием упаковщика UPX и GZEXE. С использованием hex-редактора и утилиты ENT мы получили сравнили энтропию файлов программ и их запакованных версий. Данное сравнение помогает нам представить степень сжатия данных, а также насколько снижена предсказуемость данных.

Построенные ниже диаграммы позволяют наблюдать за изменением распределения частот значений байтов в файле.

Гистограммы и энтропия C++

```
#include <iostream>
#include <string>

int main() {
    std::string correctPassword = "secret123";
    std::string inputPassword;

    std::cout << "Введите пароль: ";
    std::cin >> inputPassword;

    if (inputPassword == correctPassword) {
        std::cout << "Доступ разрешен\n";
    } else {
        std::cout << "Доступ запрещен\n";
    }

    return 0;
}
```

Рис 10. Листинг программы C++

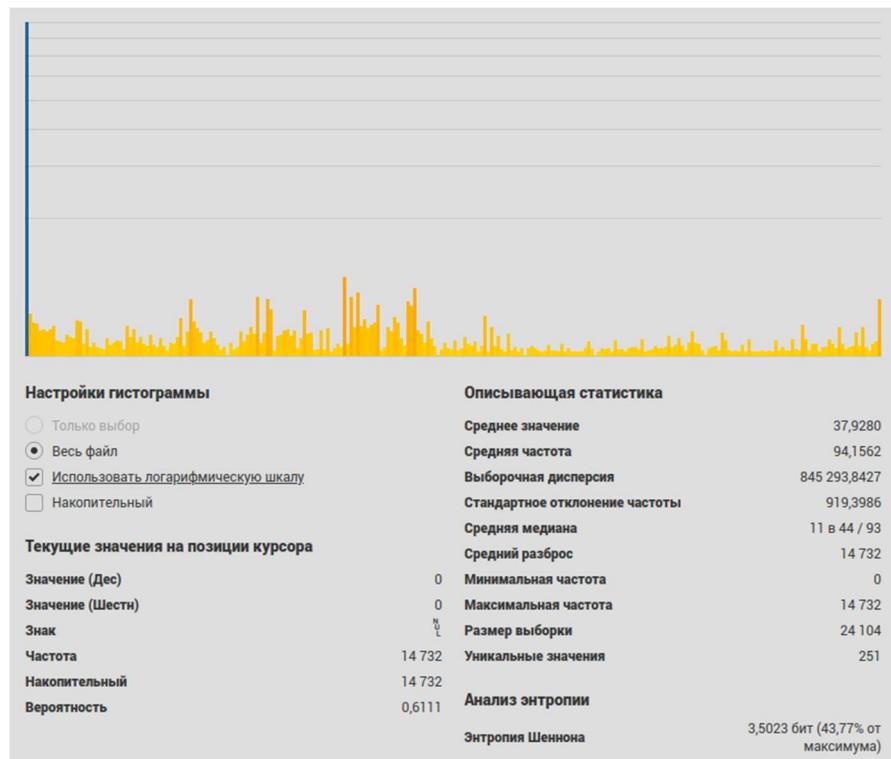


Рис 11. Гистограмма и энтропия исполняемого файла C++ (без упаковки)

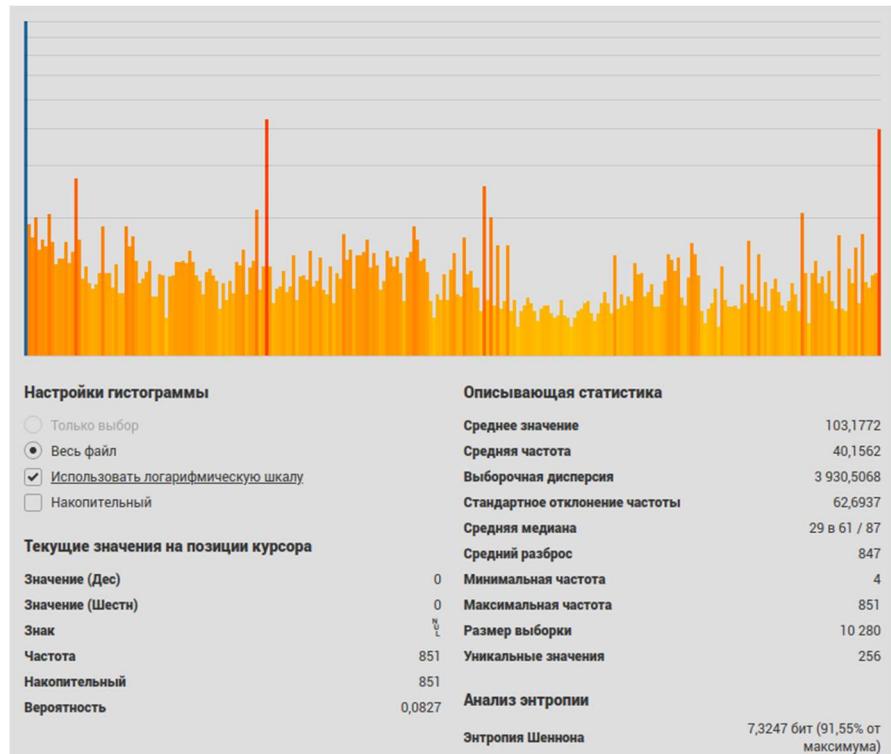


Рис 12. Гистограмма и энтропия исполняемого файла C++ (UPX)

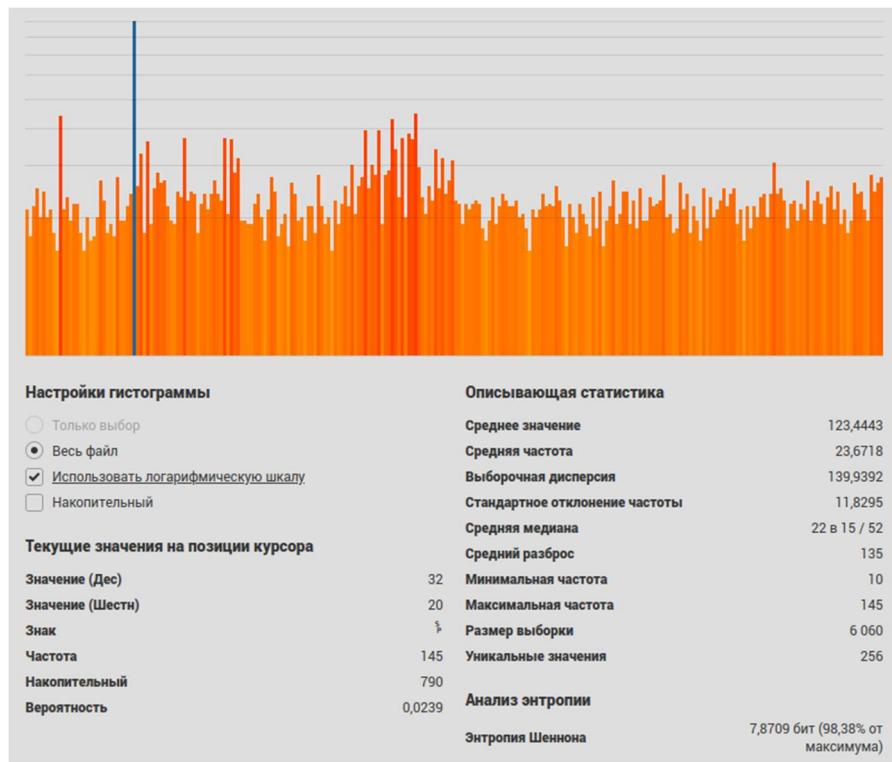


Рис 13. Гистограмма и энтропия исполняемого файла C++ (GZEXE)

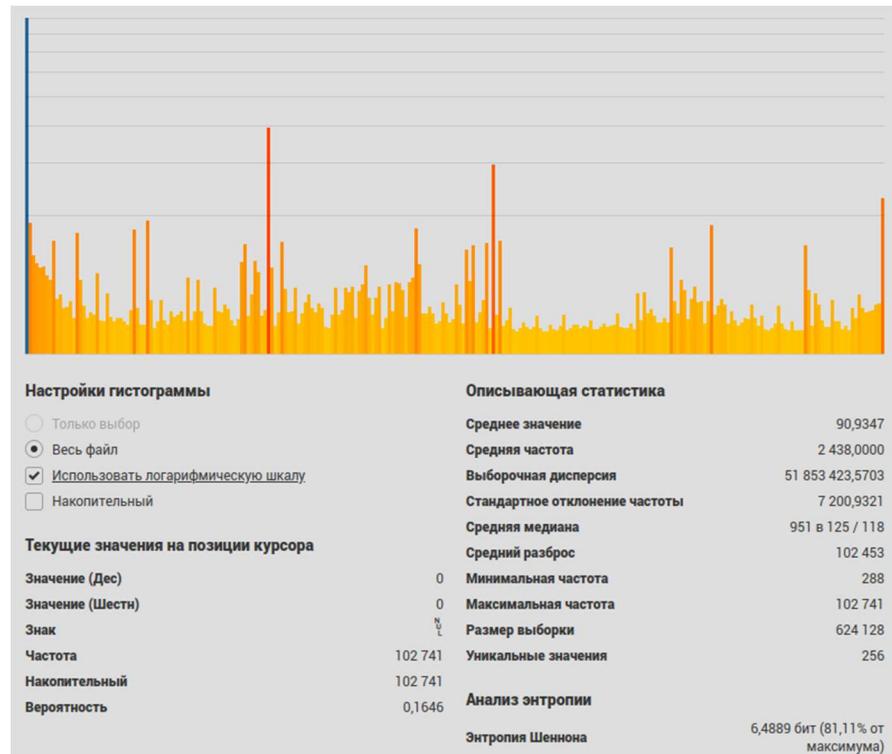


Рис 14. Гистограмма и энтропия исполняемого файла C++ (упаковщик Перепелкина Н.)

Гистограммы и энтропия C#

```
using System;

class Program {
    static void Main() {
        string correctPassword = "secret123";

        Console.WriteLine("Введите пароль: ");
        string inputPassword = Console.ReadLine();

        if (inputPassword == correctPassword) {
            Console.WriteLine("Доступ разрешен");
        } else {
            Console.WriteLine("Доступ запрещен");
        }

        Console.ReadKey();
    }
}
```

Рис 15. Листинг программы C#

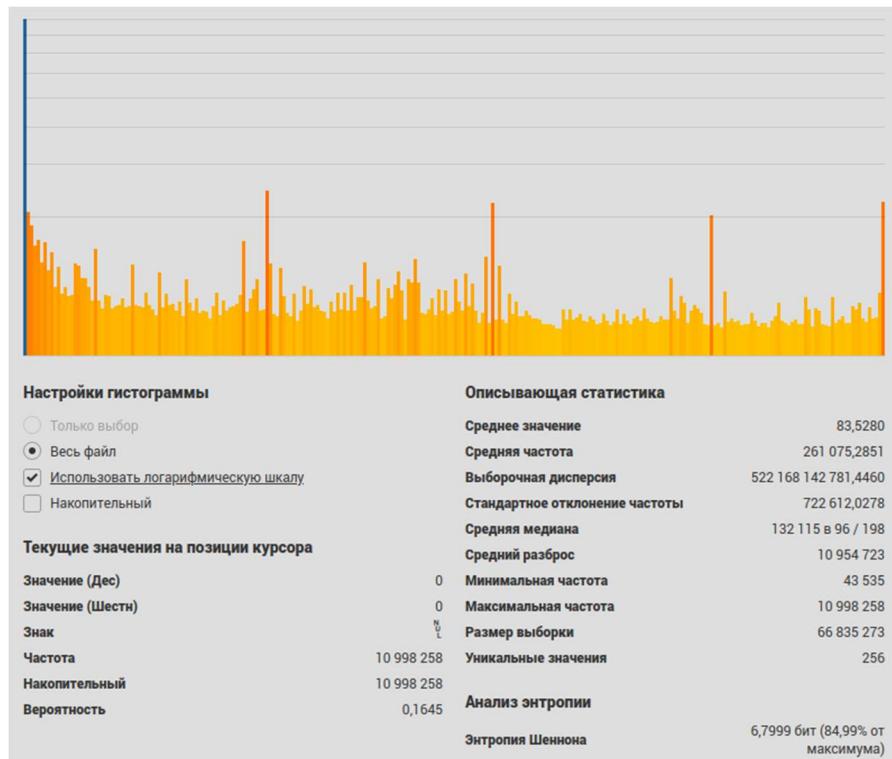


Рис 16. Гистограмма и энтропия исполняемого файла C# (без упаковки)

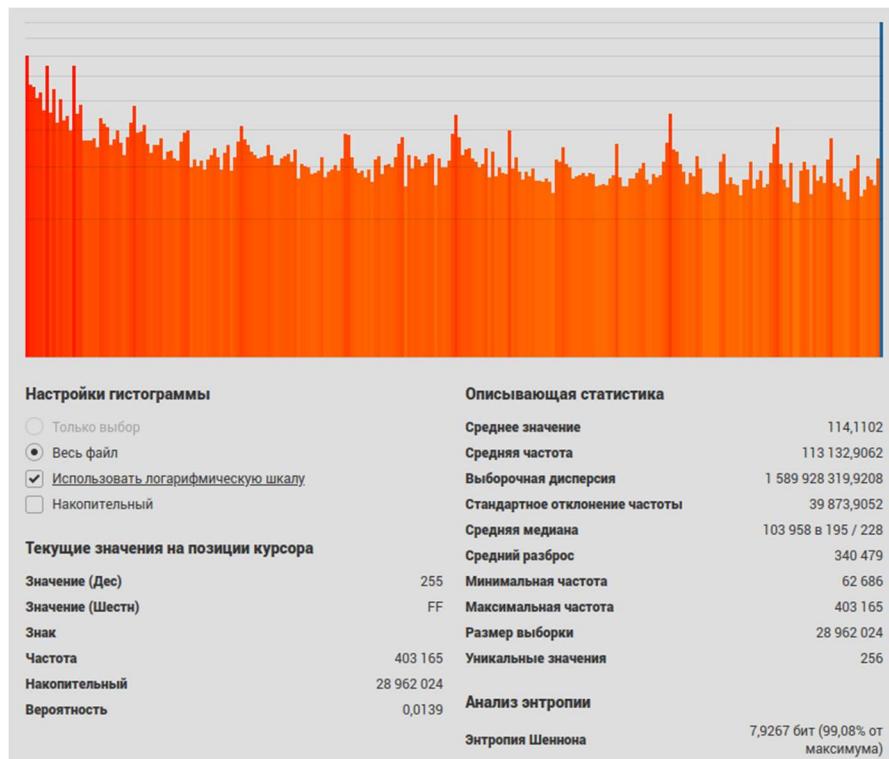


Рис 17. Гистограмма и энтропия исполняемого файла C# (UPX)



Рис 18. Гистограмма и энтропия исполняемого файла C# (GZEXE)

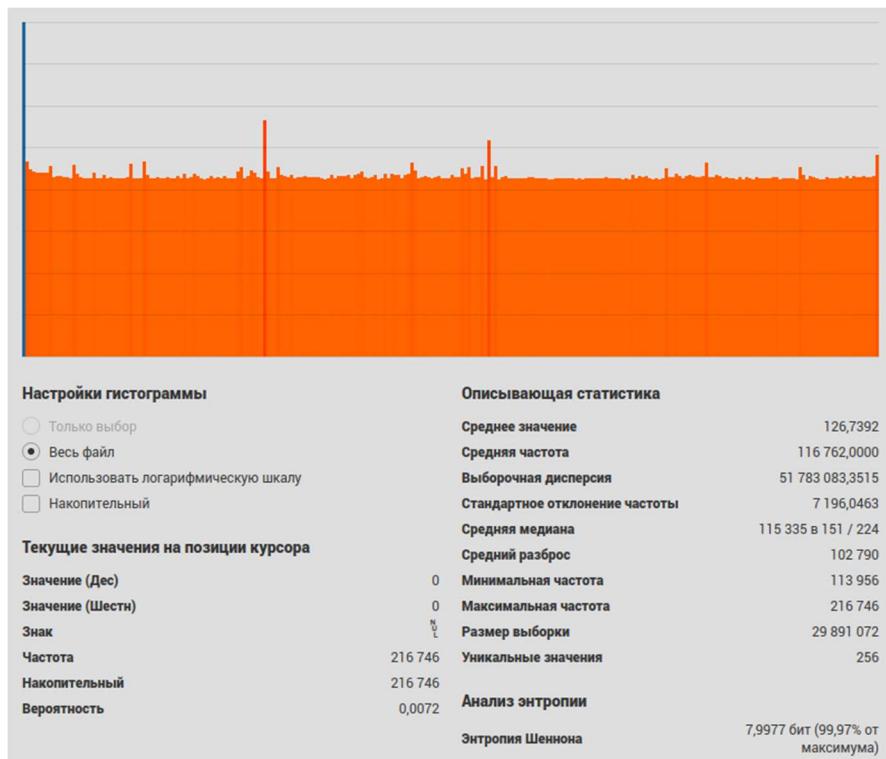


Рис 19. Гистограмма и энтропия исполняемого файла C# (упаковщик Перепелкина Н.)

Гистограммы и энтропия Python

```

correct_password = "secret123"

input_password = input("Введите пароль: ")

if input_password == correct_password:
    print("Доступ разрешен")
else:
    print("Доступ запрещен")

input()

```

Рис 20. Листинг программы Python



Рис 21. Гистограмма и энтропия исполняемого файла Python (без упаковки)



Рис 22. Гистограмма и энтропия исполняемого файла Python (UPX)

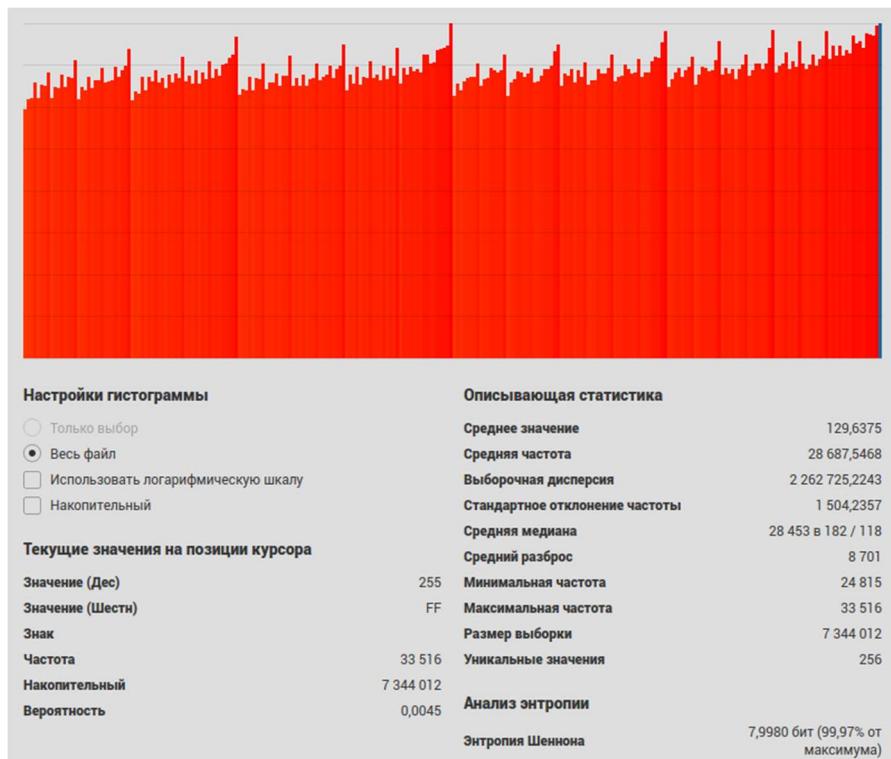


Рис 23. Гистограмма и энтропия исполняемого файла Python (GZEXE)

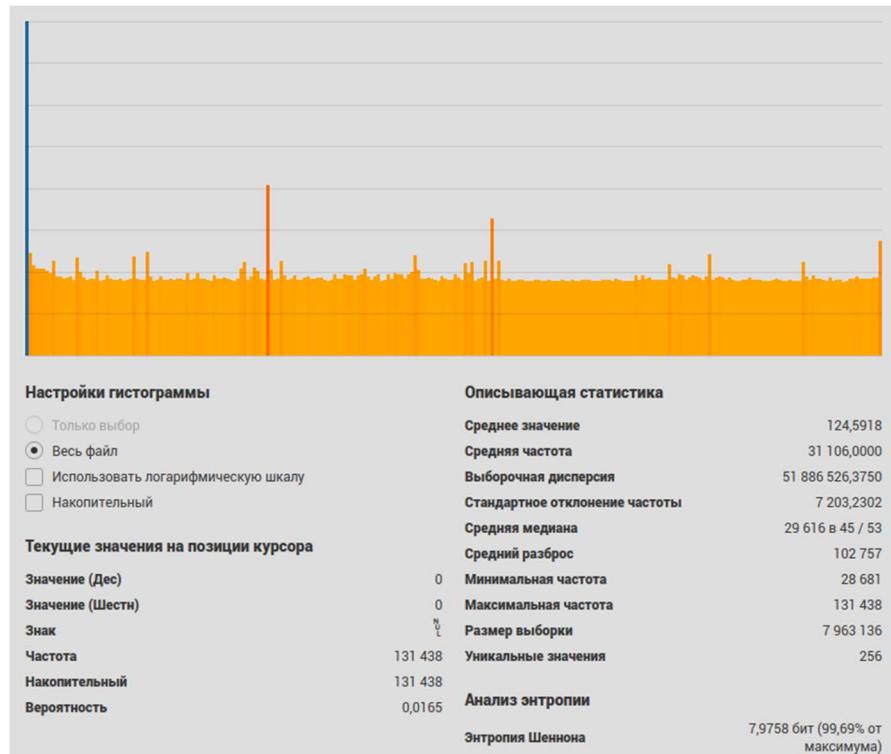


Рис 24. Гистограмма и энтропия исполняемого файла Python (упаковщик Перепелкина Н.)

Сравнение энтропии аудиофайлов разных форматов

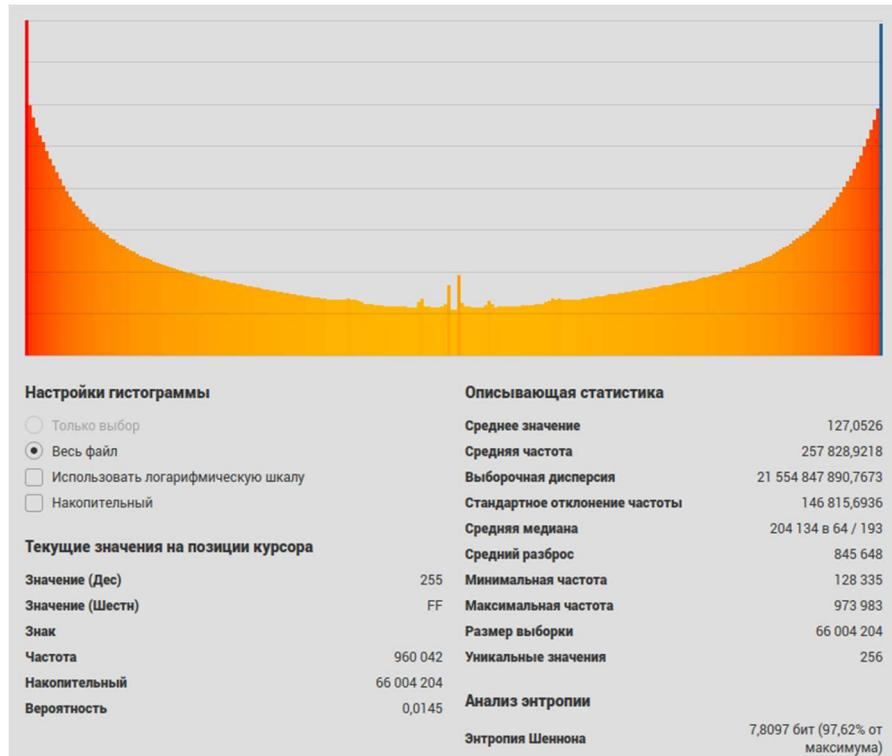


Рис 25. Гистограмма и энтропия аудиофайла формата .wav

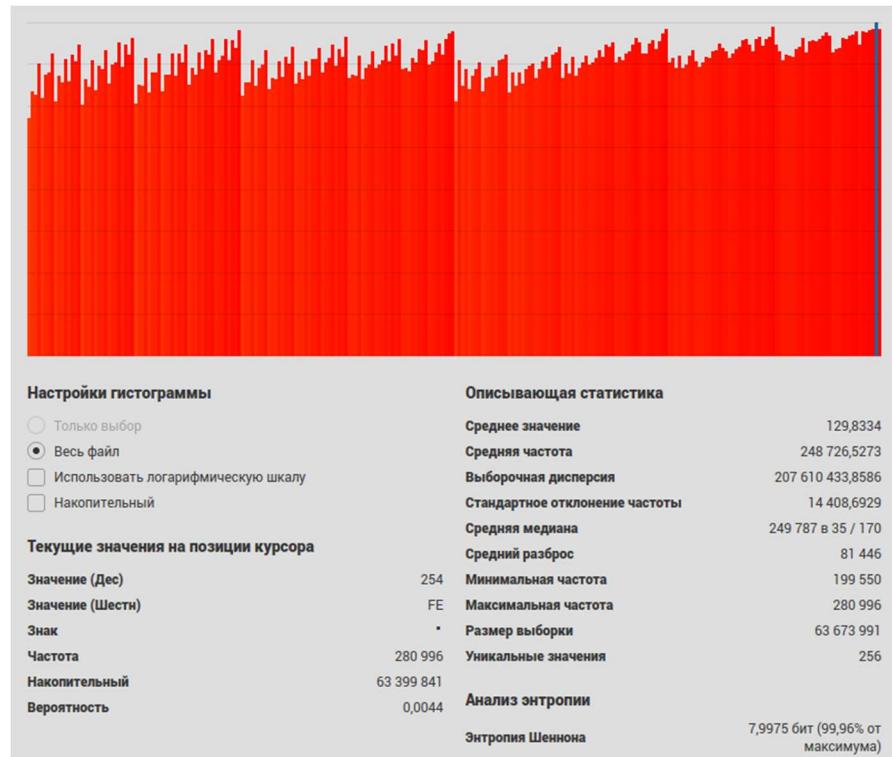


Рис 26. Гистограмма и энтропия аудиофайла формата .wav (zip)



Рис 27. Гистограмма и энтропия аудиофайла формата .mp3



Рис 28. Гистограмма и энтропия аудиофайла формата .mp3 (zip)



Рис 29. Гистограмма и энтропия аудиофайла формата .aac



Рис 30. Гистограмма и энтропия аудиофайла формата .aac (zip)

Сравнение энтропии файлов изображений разных форматов



Рис 31. Используемое для анализа энтропии изображение

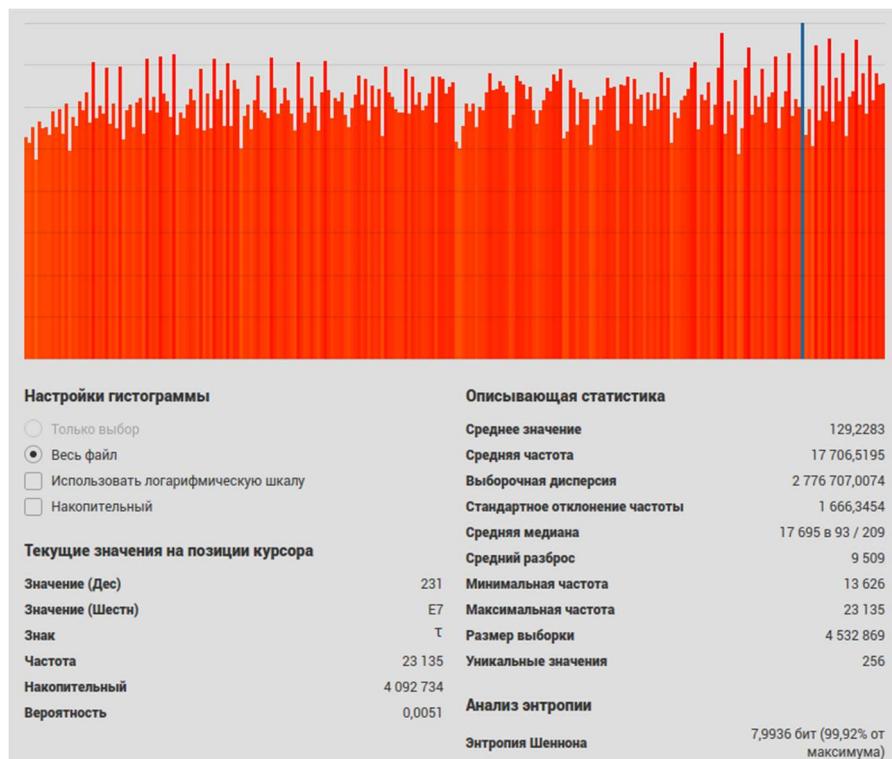


Рис 32. Гистограмма и энтропия файла изображения формата .bmp

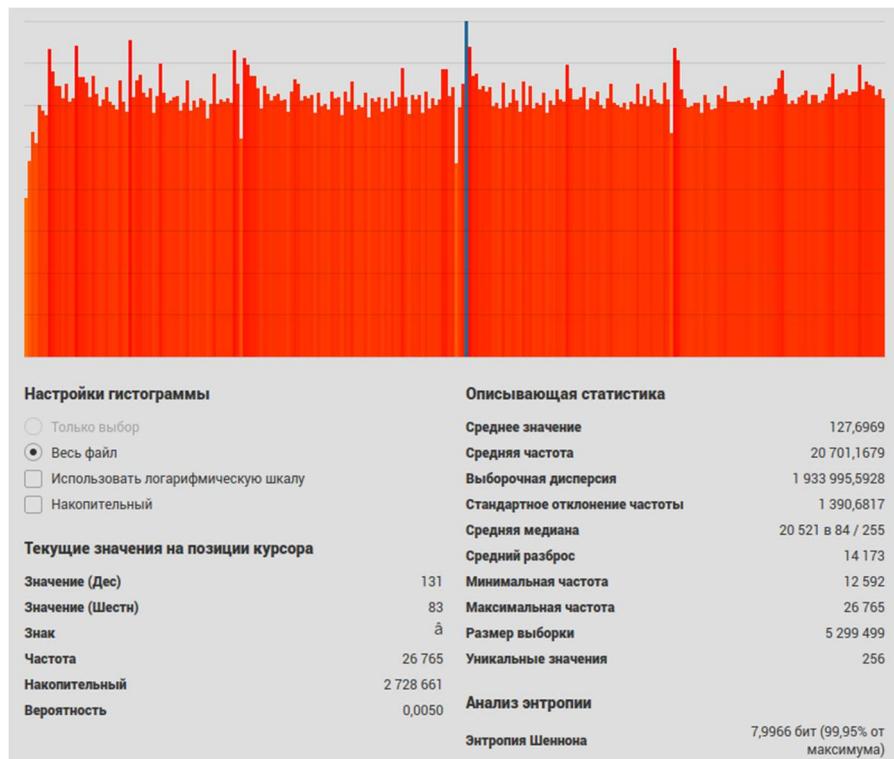


Рис 33. Гистограмма и энтропия файла изображения формата .bmp (zip)

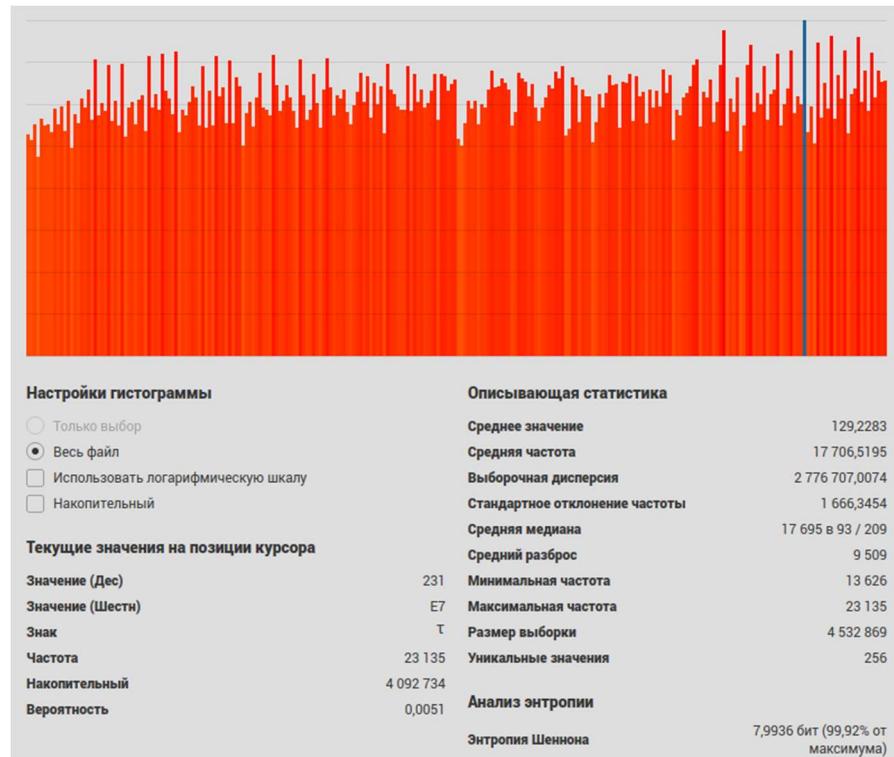


Рис 34. Гистограмма и энтропия файла изображения формата .png

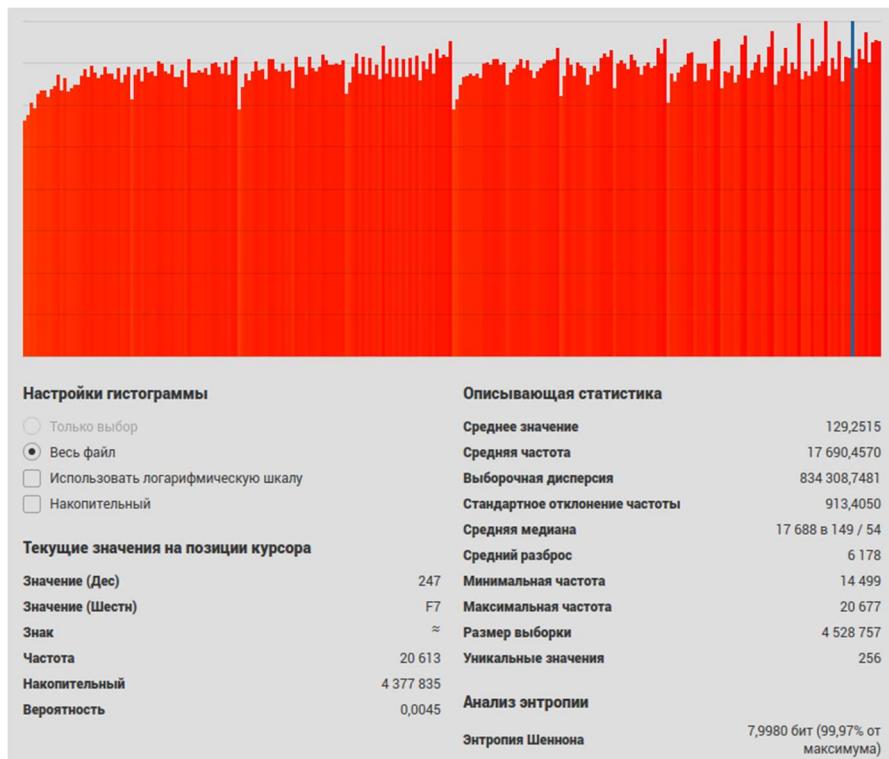


Рис 35. Гистограмма и энтропия файла изображения формата .png (zip)



Рис 36. Гистограмма и энтропия файла изображения формата .jpeg

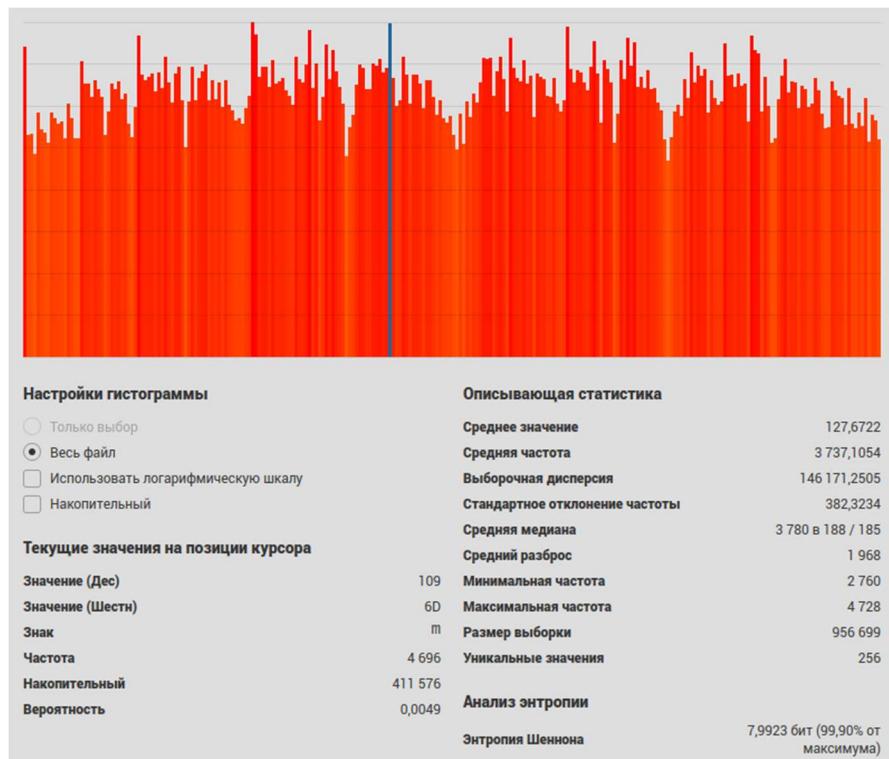


Рис 37. Гистограмма и энтропия файла изображения формата .jpeg (zip)

Сравнение энтропии текстовых файлов разных форматов

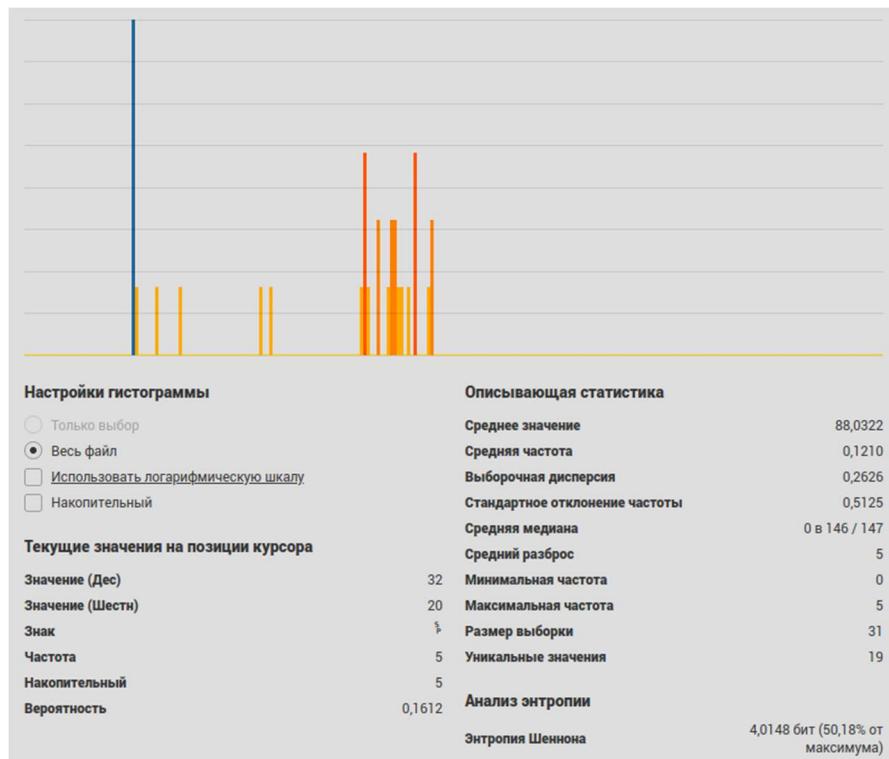


Рис 38. Гистограмма и энтропия текстового файла формата .txt



Рис 39. Гистограмма и энтропия текстового файла формата .txt (zip)

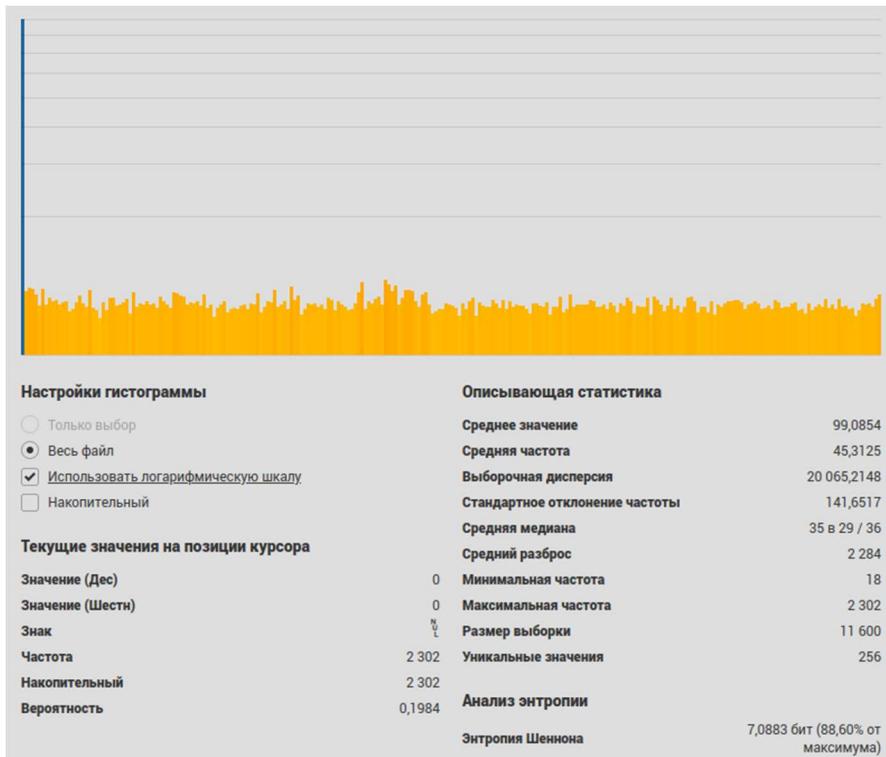


Рис 40. Гистограмма и энтропия текстового файла формата .docx



Рис 41. Гистограмма и энтропия текстового файла формата .docx (zip)



Рис 42. Гистограмма и энтропия текстового файла формата .pdf

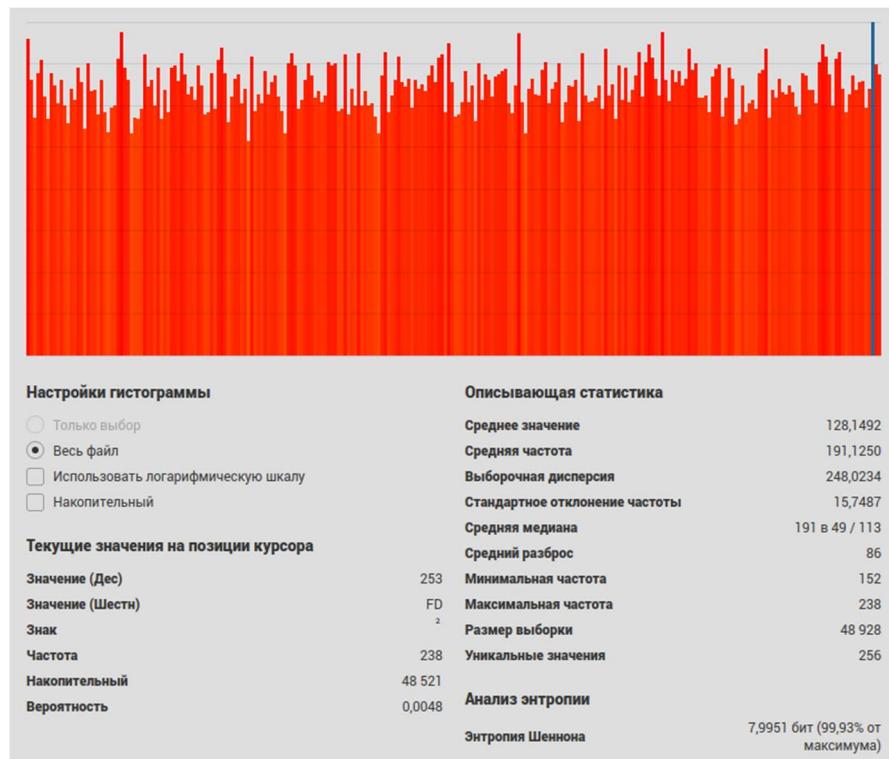


Рис 43. Гистограмма и энтропия текстового файла формата .pdf (zip)

Сравнение энтропии файлов таблиц разных форматов

A	B	C	D	E
№ п/п	Наименование товара	Количество	Цена за единицу (₽)	Стоимость (₽)
1				
2	1 Блокнот	10	150	1500
3	2 Ручка	20	50	1000
4	3 Карандаш	15	30	450
5	4 Ластик	12	20	240
6	5 Линейка	8	80	640
7	6 Тетрадь	25	40	1000
8	7 Маркер	5	120	600
9	8 Скрепки	50	10	500
10	9 Папка для документов	3	200	600
11	10 Конверт	30	15	450

Рис 44. Представление используемой таблицы

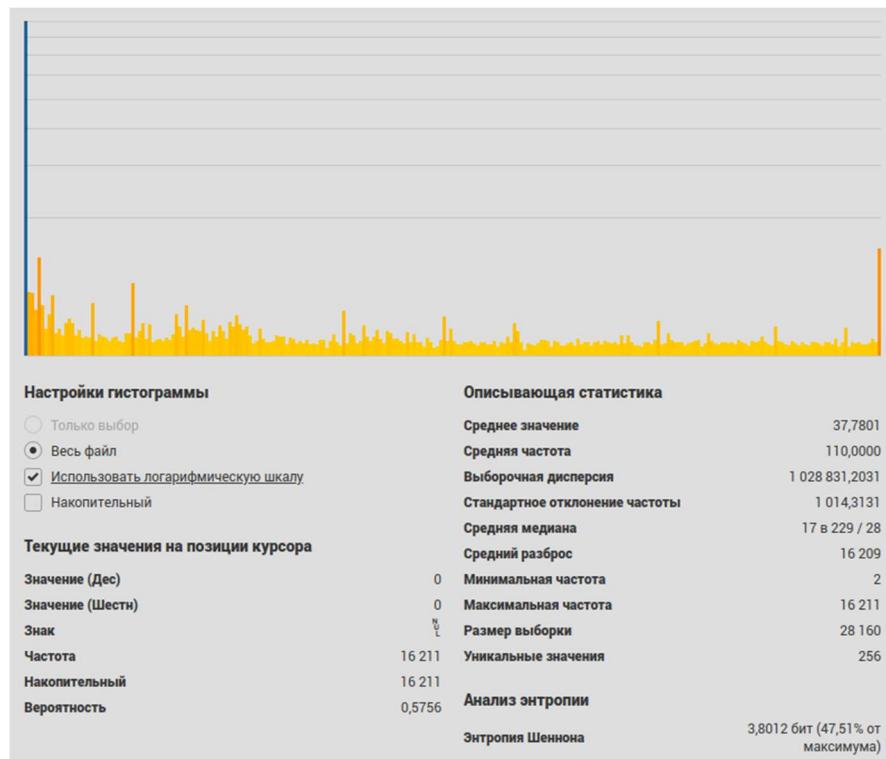


Рис 45. Гистограмма и энтропия текстового файла таблицы формата xls



Рис 46. Гистограмма и энтропия текстового файла таблицы формата xls (zip)

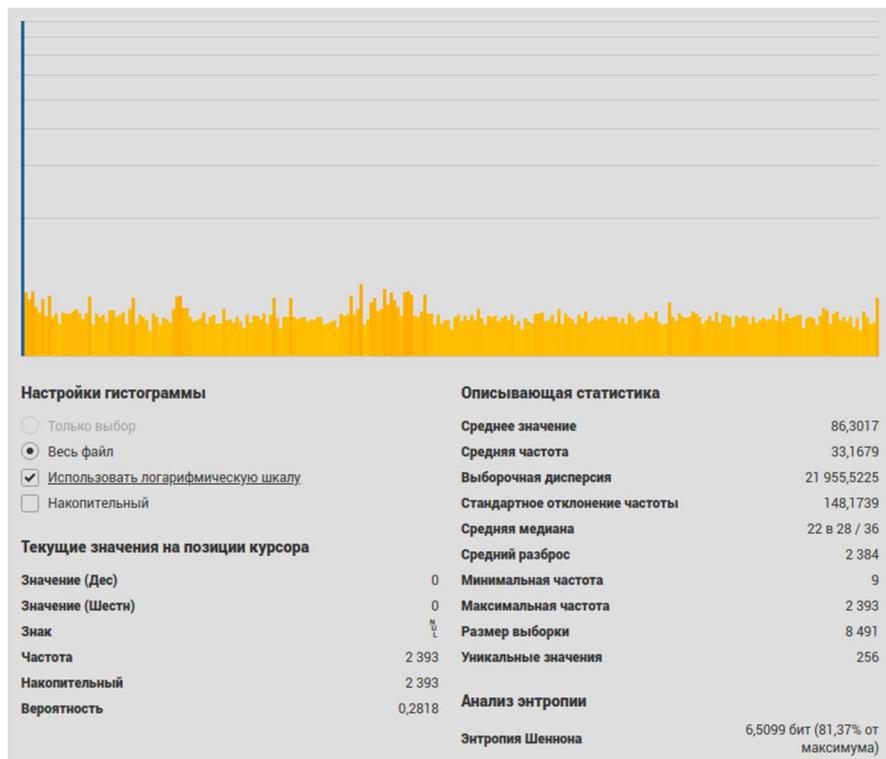


Рис 47. Гистограмма и энтропия текстового файла таблицы формата xlbs



Рис 48. Гистограмма и энтропия текстового файла таблицы формата xlbs (zip)

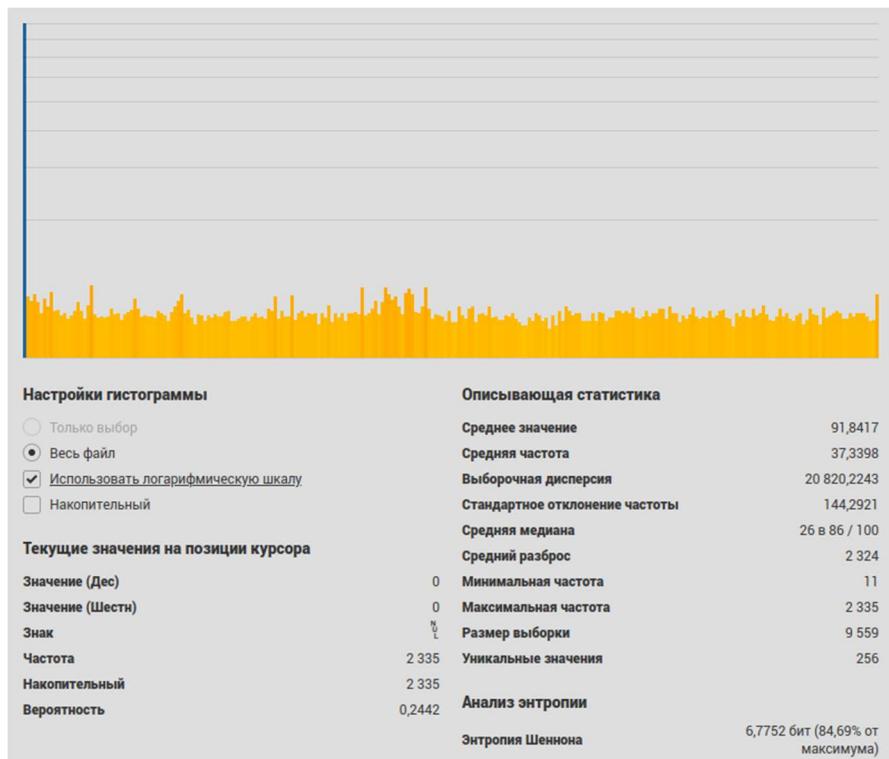


Рис 49. Гистограмма и энтропия текстового файла таблицы формата XLSX



Рис 50. Гистограмма и энтропия текстового файла таблицы формата XLSX (zip)

Таблицы сравнений энтропии

Исполняемые файлы		
Формат	Размер (кб)	Энтропия
C++	24	3.5023
C++ (UPX)	11	7.3247
C++ (GZEXE)	6	7.8709
C++ (упаковщик Перепелкина Н.)	610	6.4889
C#	65269	6.7999
C# (UPX)	28284	7.9267
C# (GZEXE)	28583	7.9969
C# (упаковщик Перепелкина Н.)	29191	7.9977
Pyhon	7268	7.9933
Pyhon (UPX)	7238	7.9938
Pyhon (GZEXE)	7172	7.9980
Pyhon (упаковщик Перепелкина Н.)	7777	7.9758

Аудиофайлы		
Формат	Размер (мб)	Энтропия
wav	64.458	7.8097
wav (zip)	62.182	7.9975
mp3	11.522	7.9646
mp3 (zip)	11.485	7.9983
aac	5.992	7.9813
aac (zip)	5.913	7.9992

Файлы изображений		
Формат	Размер (мб)	Энтропия
bmp	11.254	7.6335
bmp (zip)	5.176	7.9966
png	4.427	7.9936
png (zip)	4.423	7.9980
jpeg	0.941	7.9762
jpeg (zip)	0.935	7.9923

Текстовые файлы		
Формат	Размер (кб)	Энтропия
txt	1	4.0148
txt (zip)	1	4.5879
docx	12	7.0883
docx (zip)	9	7.9733
pdf	50	7.9769
pdf (zip)	48	7.9951

Табличные файлы		
Формат	Размер (кб)	Энтропия
xls	28	3.8012
xls (zip)	7	7.9582
xlsb	9	6.5099
xlsb (zip)	6	7.9593
XLSX	10	6.7752
XLSX (zip)	7	7.9625

Вывод

В ходе работы было исследовано изменение энтропии различных типов файлов, а также влияние упаковки исполняемых файлов на их значение энтропии. Проведённый анализ показал, что энтропия файла во многом зависит от его формата и структуры данных.

- Исполняемые файлы c++ и c# в исходном виде имели средний уровень энтропии, однако после упаковки их энтропия значительно увеличивалась, приближаясь к максимальному значению (около 8 бит на байт), что затрудняет анализ их содержимого. Программа на языке Python изначально имеет высокую энтропию, это означает, что при сборке программы в исполняемый файл уже использовался упаковщик и дальнейшая упаковка не приносит ощутимого прироста энтропии и уменьшения размера файла.
- Изображения, аудиофайлы и другие мультимедийные данные имели более высокую энтропию, особенно если использовали сжатые форматы (JPEG, MP3), что свидетельствует о минимизации избыточности данных. Упаковка же данных файлов ощутимых результатов не приносит.
- Текстовый формат txt продемонстрировал низкую энтропию из-за высокой повторяемости символов и предсказуемости структуры. Однако остальные выбранные форматы показали достаточно высокий уровень энтропии, поэтому можем предположить, что в этих форматах используется упаковка.

Таким образом, энтропийный анализ позволяет не только различать типы файлов, но и выявлять факт сжатия или шифрования данных. Эти знания могут быть полезны в области реверс-инжиниринга, информационной безопасности и анализа вредоносного ПО.