



Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ «Программной инженерии и компьютерной техники»

ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА «Проектирование и разработка систем больших данных»

НАПРАВЛЕНИЕ ПОДГОТОВКИ (СПЕЦИАЛЬНОСТЬ) «09.04.04 Программная инженерия»

ОТЧЕТ

по лабораторной работе №4
по курсу «Хранение и алгоритмы сжатия данных»
на тему: «Сравнение алгоритмов сжатия данных»

Обучающийся: Постнов С. А., Р4135
(Ф.И.О, № группы)

Преподаватель Бабаянц А. А., преподаватель
(Ф.И.О, должность)

2025 г.

СОДЕРЖАНИЕ

1	Теоретическая часть	3
1.1	Алгоритм сжатия Хаффмана	3
1.2	Библиотека zlib	3
2	Практическая часть	4
2.1	Реализация программ	4
2.2	Результат работы	4
3	Вывод	6

1 Теоретическая часть

1.1 Алгоритм сжатия Хаффмана

Алгоритм Хаффмана — это метод энтропийного сжатия данных без потерь, разработанный Дэвидом Хаффманом в 1952 году. Он основан на построении префиксного кода, где более частые символы получают более короткие коды.

Основные шаги:

- 1) Подсчёт частоты каждого символа в данных.
- 2) Построение дерева Хаффмана: символы с наименьшей частотой объединяются в узлы до получения одного корневого узла.
- 3) Присвоение кодов: левым рёбрам — 0, правым — 1.
- 4) Кодирование данных с использованием полученных кодов.

Декодирование происходит с помощью того же дерева.

1.2 Библиотека `zlib`

`zlib` — это библиотека для сжатия данных, реализующая алгоритм DEFLATE, который сочетает LZ77 и код Хаффмана. DEFLATE использует словарное сжатие (LZ77) для поиска повторяющихся последовательностей и динамическое кодирование Хаффмана для сжатия.

Особенности:

- 1) Эффективно сжимает текстовые и бинарные данные.
- 2) Поддерживает уровни сжатия от 0 (без сжатия) до 9 (максимальное сжатие).
- 3) Широко используется в форматах ZIP, PNG, HTTP.

2 Практическая часть

В лабораторной работе реализована программа на C++ для сравнения эффективности алгоритма сжатия Хаффмана (собственная реализация) и библиотеки `zlib`. Программа тестирует обе реализации алгоритма на различных типах файлов: текстовом (`file.txt`), PDF (`file.pdf`), DOC (`file.doc`), ZIP (`file.zip`) и BMP (`file.bmp`).

2.1 Реализация программ

- Реализация алгоритма Хаффмана: построение дерева частот, генерация кодов, кодирование и декодирование.
- Замеры времени кодирования и декодирования при помощи реализованного алгоритма и `zlib`.
- Расчёт коэффициента сжатия как отношение размера сжатых данных к оригинальным.
- Вывод результатов в CSV-файл для дальнейшего анализа.

2.2 Результат работы

Результаты сравнения представлены в таблице 2.1 и на графике 2.1.

Таблица 2.1 – Результаты сравнения алгоритмов сжатия

Алгоритм	Файл	Коэффициент сжатия	Время сжатия (мс)	Время разжатия (мс)
Huffman	data/file.txt	2.608696	0.174500	0.095292
zlib	data/file.txt	6.666667	0.101958	0.064083
Huffman	data/file.pdf	0.999635	1420.730750	764.234750
zlib	data/file.pdf	1.042540	151.265916	114.895542
Huffman	data/file.doc	1.572061	18.582125	8.246333
zlib	data/file.doc	2.738191	2.636458	1.003500
Huffman	data/file.zip	0.998123	275.354208	147.683166
zlib	data/file.zip	1.000645	24.997625	21.995875
Huffman	data/file.bmp	1.049384	73.750625	38.998917
zlib	data/file.bmp	1.181160	8.448833	5.912958

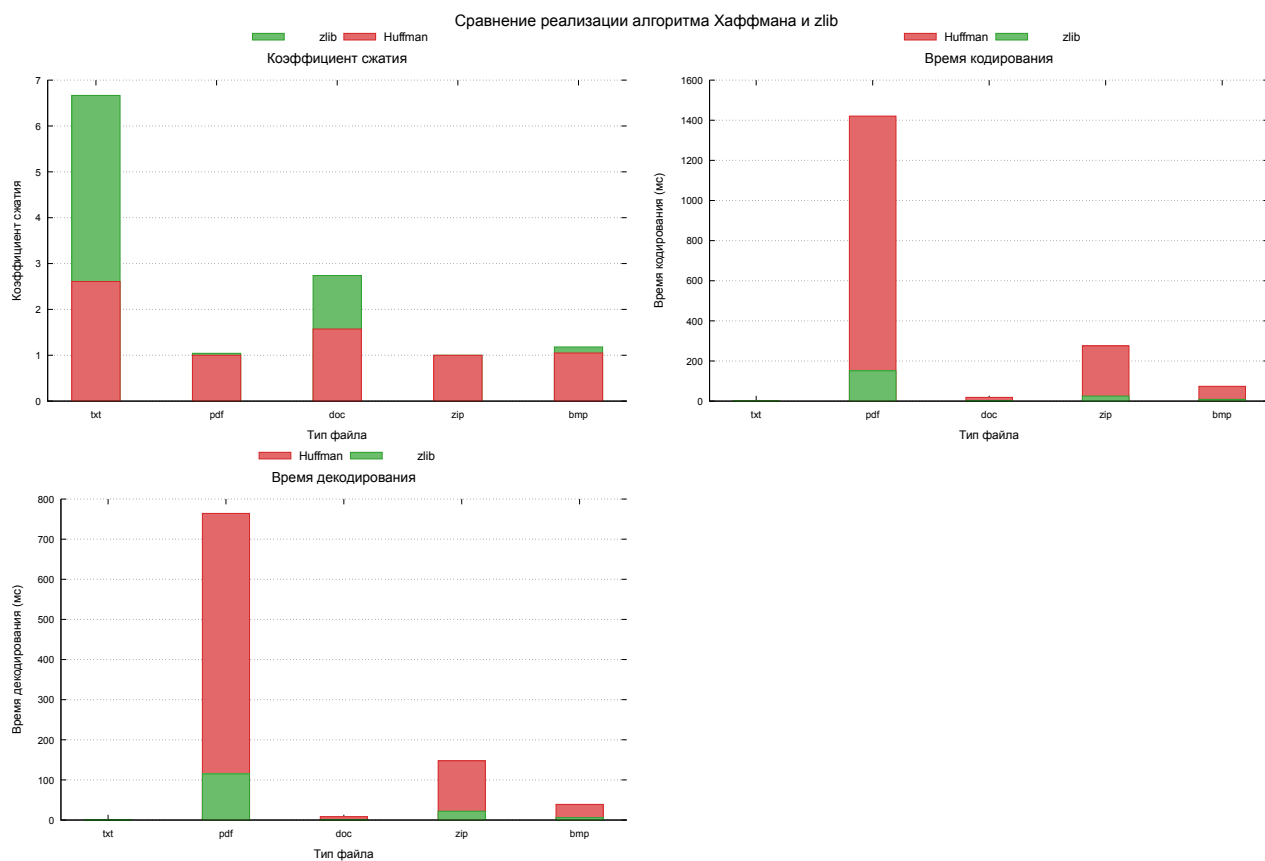


Рисунок 2.1 – Результаты сравнения реализаций алгоритмов сжатия

3 Вывод

Исходя из результатов сравнения, можно сделать следующие выводы:

- 1) **zlib** показывает лучший коэффициент сжатия для большинства файлов (текст, PDF, DOC, BMP), за исключением уже сжатых (ZIP), где оба алгоритма близки.
- 2) Время кодирования и декодирования у **zlib** значительно быстрее, особенно для больших файлов (PDF).
- 3) Алгоритм Хаффмана эффективен для простых случаев, но **zlib** превосходит по скорости и степени сжатия благодаря комбинации LZ77 и динамического кодирования.