Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №2**

**«Элементы теории информации. Информативность данных в различных кодировках»**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

Цель: приобретение практических навыков трансформации данных и сопоставление энтропийных свойств используемых при этом алфавитов.

Задачи:

1. Закрепить теоретические знания по взаимной конвертации данных, представленных в кодах ASCII и base64.

2. Разработать приложение для конвертации произвольного документа в формат base64 и обратно.

3. Исследовать энтропийные характеристики используемых в конвертерах алфавитов.

4. Изучить особенности практической реализации операции XOR над данными, представленными в разных форматах.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**2. Теоретические сведения**

Энтропия максимальна при равномерном появлении букв на любом месте сообщения. Для характеристики источника сообщений с различным алфавитом представляет интерес сравнение фактической энтропии источника с максимально возможной. В этом смысле введено понятие избыточности источника сообщений, или избыточности алфавита.

Избыточностью алфавита называют уменьшение информационной нагрузки на один символ вследствие разной вероятности и взаимозависимости появления его символов в сообщениях.

В наиболее общем виде избыточность алфавита R можно оценить отношением энтропии по Хартли и по Шеннону:

Утверждение восьмибитных кодировок (ASCII) как стандарта принесло некоторые проблемы. К этому моменту уже существовала определенная инфраструктура, использующая семибитные кодировки. Известны проблемы с «обрезанием восьмого бита» в системе электронной почты.

Одним из решений проблемы стала кодировка (а точнее – алгоритм) base64. В PGP алгоритм base64 используется для кодирования бинарных данных.

Кодирование base64 разработано для представления произвольных последовательностей октетов в форме, позволяющей использовать строчные и прописные буквы. Используется 65-символьное подмножество набора символов US-ASCII, обеспечивающее представление одним печатным символом 6 битов данных (дополнительный 65-й символ используется для обозначения функции специальной обработки).

Процесс кодирования представляет группу из 24 последовательных битов в форме строки из 4 символов. Обработка выполняется слева направо, а 24-битная исходная группа образуется конкатенацией трех 8-битных групп (байтов). Данные 24 бита после этого трактуются как 4 сцепленных группы по 6 битов, каждая из которых транслируется в один символ алфавита base64. Каждая 6-битная группа используется в качестве индекса массива из 64 печатных символов. Символы алфавита, соответствующие индексу, помещаются в выходную строку [6]. Кодирование base64 с безопасным алфавитом используется для представления URL и имен файлов

Если на входе доступно менее 24 битов, входная группа дополняется (справа) нулями до формирования целого числа 6-битных групп. Заполнение в конце данных осуществляется как раз с использованием символа «=». Поскольку входная информация base64 всегда включает целое число октетов, возможны лишь перечисленные ниже случаи:

• размер финального блока кодирования на входе кратен 24 битам, кодированный результат будет содержать целое число 4-символьных групп без заполнения символами «=»;

• размер финального блока кодирования на входе составляет 8 битов, выходной блок будет представлять 2 символа, дополненные последовательностью из двух символов заполнения «==»;

• размер финального блока кодирования на входе составляет 16 битов, выходной блок будет представлять 3 символа, дополненные символом заполнения «=».

**3. Практическая часть**

**Практическое задание:**

1. Создать собственное приложение (приветствуется!) или воспользоваться Base64-онлайн-кодировщиком, с помощью которого конвертировать произвольный документ (а) на латинице (можно использовать документ из лабораторной работы № 1) в документ (б) формата base64. В качестве входных данных можно использовать указанный преподавателем вариант из списка:

• входные параметры;

• текстовый файл (\*.txt);

• документ Word (\*.doc);

• документ Word (\*.docx);

• документ PowerPoint (\*.ppt, \*.pptx);

• архив (\*.zip);

• текстовая строка;

• случайное число (от 999999);

• PDF-файл;

• архив (\*.rar);

• архив (\*.7z).

2. С помощью приложения, созданного в лабораторной работе № 1, получить распределение частотных свойств алфавитов по документам (а) и (б). Вычислить энтропию Хартли и Шеннона, а также избыточность алфавитов. Объяснить полученный результат.

3. Написать функцию, которая принимает в качестве аргументов два буфера (а и b) одинакового размера и возвращает XOR (собственная фамилия (а) и имя (b); при разной длине меньшую дополнить нулями). Входные аргументы представлять: 1) в кодах ASCII; 2) в кодах base64. Что будет результатом операции аXORbXORb?

При написании не использовать стандартные функции языка программирования. Итоговые данные сравнить с результатами использования стандартных функций языка программирования (если они есть).

4. Результаты оформить в виде отчета по установленным правилам.

**Ход работы**

Для создания собственного приложения, с помощью которого можно конвертировать произвольный документ, была разработана функция encodeFile, представленная на рисунке 3.1.



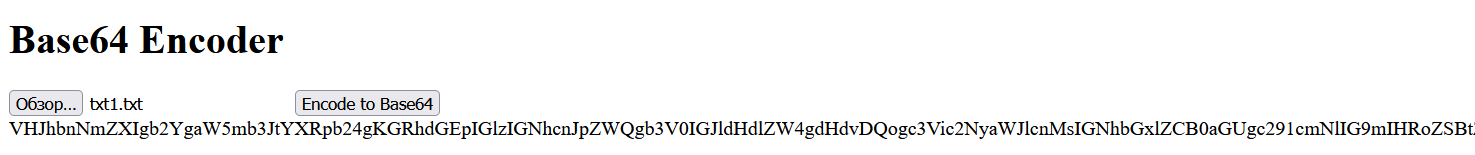
Рисунок 3.1 – Функция для кодирования в base64

Для сохранения закодированного текста в файл была разработана функция saveToFile представленная на рисунке 3.2. В качестве параметров функция принимает закодированную строку и имя нового файла.



Рисунок 3.2 – Функция для сохранения закодированного файла в файл

Результат работы функции представлен на рисунке 3.3.

 Рисунок 3.3 – Результат работы функции encodeFile

Чтобы получить распределение частотных свойств алфавитов по документам использовалась функция Get, изображенная на рисунке 3.4.

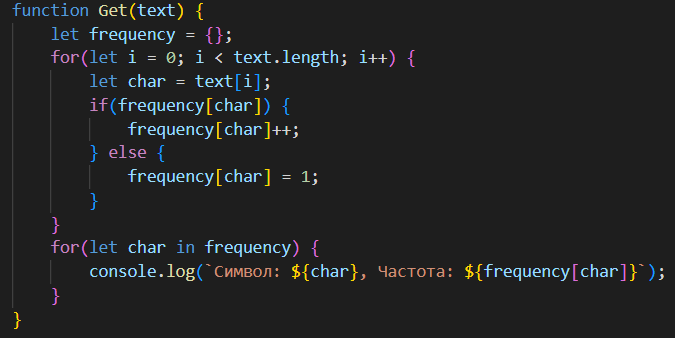


Рисунок 3.4 – Код функции Get для получения распределения частотных свойств алфавитов

Для вычисления энтропии Хартли и Шеннона использовались разработанные во 2 лабораторной работе функции GetEntropyShanon и GetEntropyHartley. Для расчета избыточности алфавита была написана функция calculateRedundancy(рисунок 3.5).

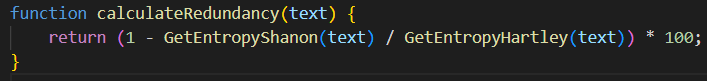


Рисунок 3.5 – Код функции calculateRedundancy

Результат работы функций GetEntropyShanon, GetEntropyHartley и calculateRedundancy представлен на рисунке 3.6.

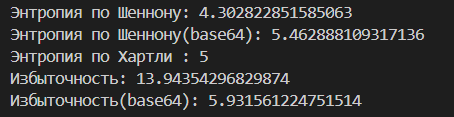
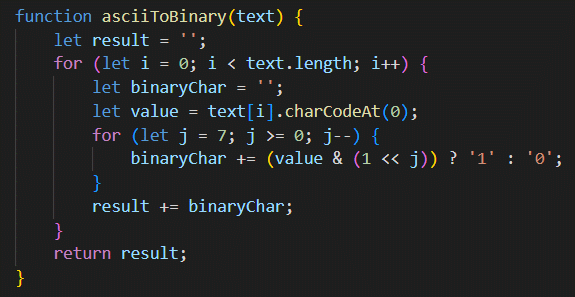


Рисунок 3.7 – Результат работы функций GetEntropyShanon, GetEntropyHartley и calculateRedundancy

Перед написанием функции, которая принимает в качестве аргументов два буфера (а и b) одинакового размера и возвращает XOR было разработано 2 функции: asciiToBinary для перевода текста ascii в двоичный формат и base64ToBinary для перевода текста base64в двоичный формат.

На рисунке 3.8 приведен код функции asciiToBinary.



На рисунке 3.9 приведен код функции base64ToBinary.

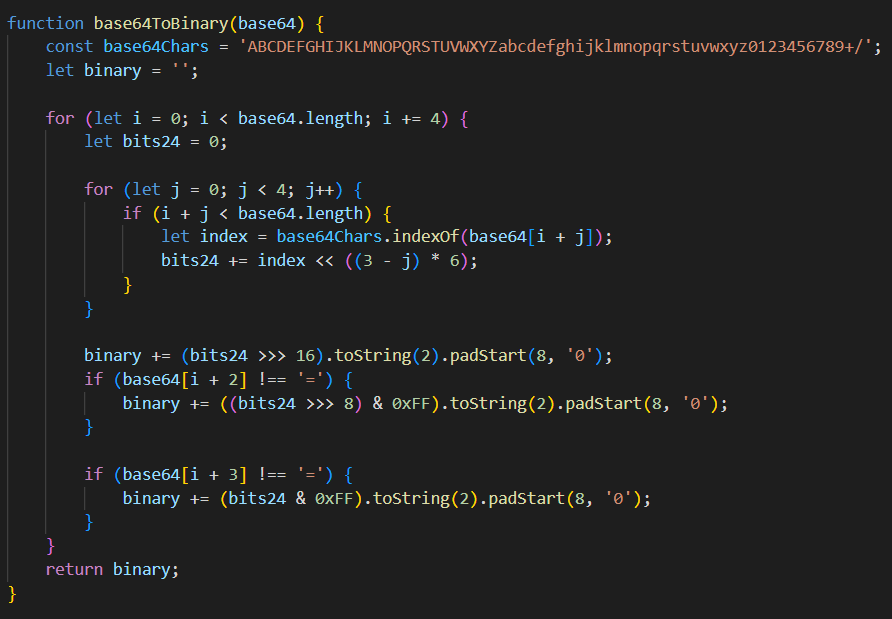


Рисунок 3.9 – Код функции base64ToBinary

Реализована функция для расчета XOR двух текстов в ASCII и Base64 форматах, представленная на рисунке 3.10.

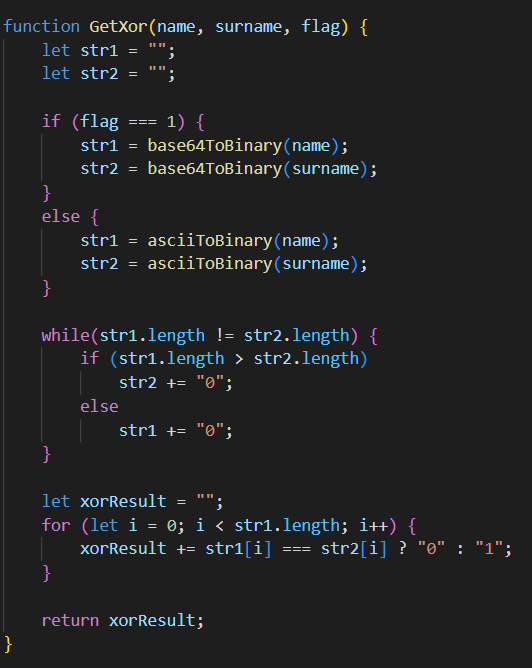


Рисунок 3.10 – Код функции GetXOR

Функция **GetXor** демонстрирует процесс побитового сравнения данных в различных кодировках (ASCII и Base64), что позволяет анализировать их бинарные представления и особенности. Это способствует более глубокому пониманию информационных характеристик и различий между этими кодировками, что является ключевым аспектом теории информации и информационной безопасности.

**Вывод**

В ходе работы были рассмотрены различные аспекты информативности данных в различных кодировках. ASCII текст имеет энтропию по Шеннону, которая отражает среднюю неопределенность (информацию) каждого символа в тексте, тогда как Base64 текст обычно имеет меньшую энтропию по Шеннону из-за увеличенной длины и уменьшенной уникальности символов. Энтропия по Хартли измеряет количество информации на символ, исходя из количества уникальных символов в тексте. ASCII текст демонстрирует более высокую энтропию по Хартли по сравнению с Base64 текстом, так как последний использует ограниченный набор символов. Избыточность ASCII текста показывает процент повторяющейся или неинформативной информации, тогда как избыточность Base64 текста выше из-за его кодировки, которая увеличивает количество символов. Анализ частотных свойств символов выявил, что некоторые символы в ASCII тексте встречаются чаще других, что влияет на энтропию, в то время как в Base64 тексте частотное распределение символов более равномерное из-за природы кодировки. Преобразование текстов в бинарный формат позволяет выполнять побитовые операции, такие как XOR. Расчет XOR показал различия в бинарных представлениях ASCII и Base64 текстов, что важно для задач криптографии и кодирования. Работа продемонстрировала различные аспекты информативности данных в различных кодировках и способы их количественного анализа. Результаты показывают, что выбор кодировки влияет на информационные характеристики текста, такие как энтропия и избыточность, что важно учитывать при обработке и передаче данных.