Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

«Оптимизация метода сжатия страниц памяти с использованием подсчета информационной энтропии»

Студент ИУ7-83Б _____ Р. Р. Хамзина
(Группа) (Подпись, дата) (И. О. Фамилия)

Руководитель ВКР _____ А. А. Оленев
(Подпись, дата) (И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

1 Аналитический раздел

1.1 Управление памятью в операционной системе

Для хранения данных запущенных в операционной системе процессов, находящихся в состоянии выполнения, ожидания или блокировки, используется оперативная память. Функции оперативной памяти реализует оперативное запоминающее устройство (ОЗУ). В многозадачных системах память должна быть распределена для размещения нескольких процессов. Для управления памятью в системе используется абстракция адресного пространства. У каждого процесса имеется собственное адресное пространство — набор адресов, который может быть использован процессом для обращения к памяти. Если объем оперативной памяти, необходимый для размещения данных всех процессов, превышает объем ОЗУ, то возникает перегрузка памяти. Одним из способов решения проблемы является виртуальная память.

1.1.1 Виртуальная память

Механизм виртуальной памяти предполагает разделение логической памяти и физической памяти. У каждого процесса имеется собственное виртуальное адресное пространство. При использовании страничной организации памяти, виртуальное адресное пространство делится на блоки фиксированного размера, называемые страницами. Физическая память делится на блоки байт фиксированного размера, называемые физическими страницами, страничным блоком или кадром. Страницы виртуального адресного пространства и физические страницы имеют одинаковые размеры от 512 байт до 1 гигабайта.

При использовании виртуальной памяти процессы работают не с физическими адресами, а с виртуальными. Виртуальный адрес — это адрес, присвоенный местоположению в виртуальной памяти, который позволяет обращаться к данному местоположению так, как если бы это была часть физической памяти. Отображение виртуального адреса на физический адрес памяти выполняется диспетчером памяти с использованием таблиц страниц.

Схема работы виртуальной памяти показана на рисунке 1.1.

Применение виртуальной памяти позволяет реализовать механизм подкачки страниц.

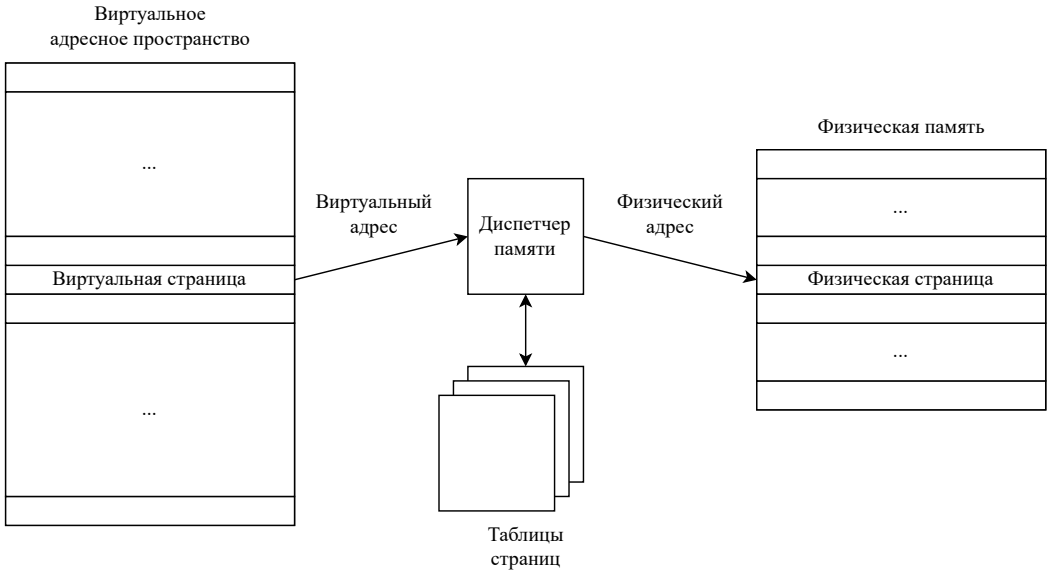

Рисунок 1.1 – Схема работы виртуальной памяти

1.1.2 Подкачка страниц

Идея подкачки страниц заключается в хранении некоторых физических страниц во вторичном хранилище, а не в оперативной памяти, и загрузке страницы в память только тогда, когда она необходима. Присутствие страницы отслеживается с помощью бита присутствия-отсутствия.

Если процесс попытается получить доступ к отсутствующей в памяти странице, возникает системное прерывание (page fault). Операционная система находит свободный страничный кадр или выбирает редко используемую физическую страницу и сбрасывает ее содержимое во вторичное хранилище. Затем она перемещает нужную физическую страницу в свободный страничный кадр.

Схема работы подкачки страниц при возникновении page fault приведена на рисунке 1.2.

1.2 Управление памятью в ядре Linux

Операционная система Linux является системой с поддержкой виртуальной памяти. В данной работе будет рассматриваться ядро Linux, так как оно является программным обеспечением с открытым исходным кодом.

TODO: написать про PAGE_SIZE.

TODO: описание struct page.

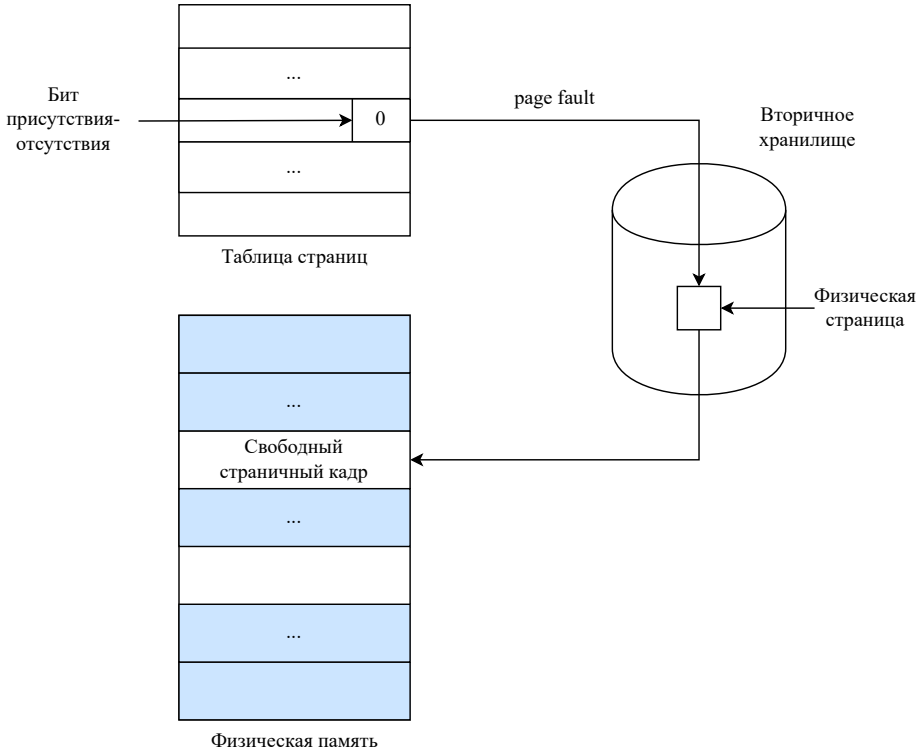

Рисунок 1.2 – Схема работы подкачки страниц при возникновении page fault

1.3 Сжатие данных

Неслучайные данные имеют некоторую структуру. Наличие у данных структуры, которую можно использовать для уменьшения их размера путем достижения такого представления данных, в котором никакая структура не выделяется, называют избыточностью. Сжатие данных — это процесс преобразования исходных данных в их компактную форму путем распознавания и использования избыточности данных [1]. Процесс сжатия состоит из двух этапов:

1. Этап моделирования, который включает в себя распознавание избыточности для построения модели. Модель представляет собой набор данных и правил, используемых для обработки входных символов.

2. Этап кодирования данных с использованием модели.

Описанные этапы сжатия данных представлены на рисунке 1.3.

В результате сжатия исходных данных X получается их представление $X_{\text{сж}}$. При восстановлении сжатые данные $X_{\text{сж}}$ преобразуются в представление Y. На основании требований к восстановлению выделяют [2]:

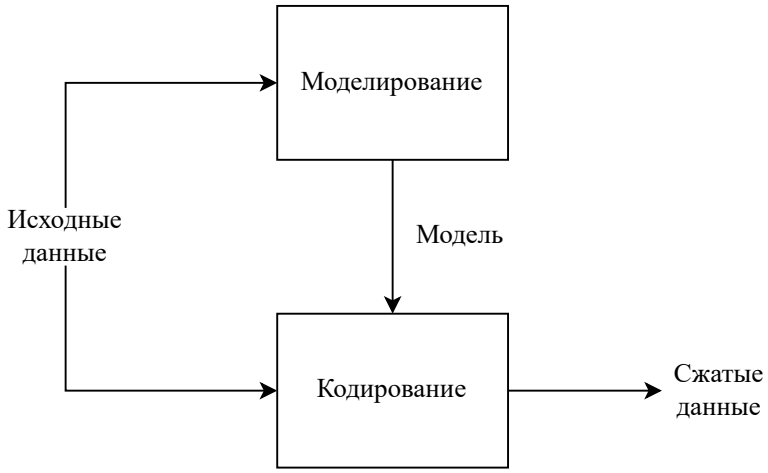

Рисунок 1.3 – Этапы сжатия данных

— сжатие данных без потерь, при котором $Y = X$;

— сжатие данных с потерями, при котором $Y \neq X$.

Схемы сжатия данных без потерь и с потерями показаны на рисунке 1.4.

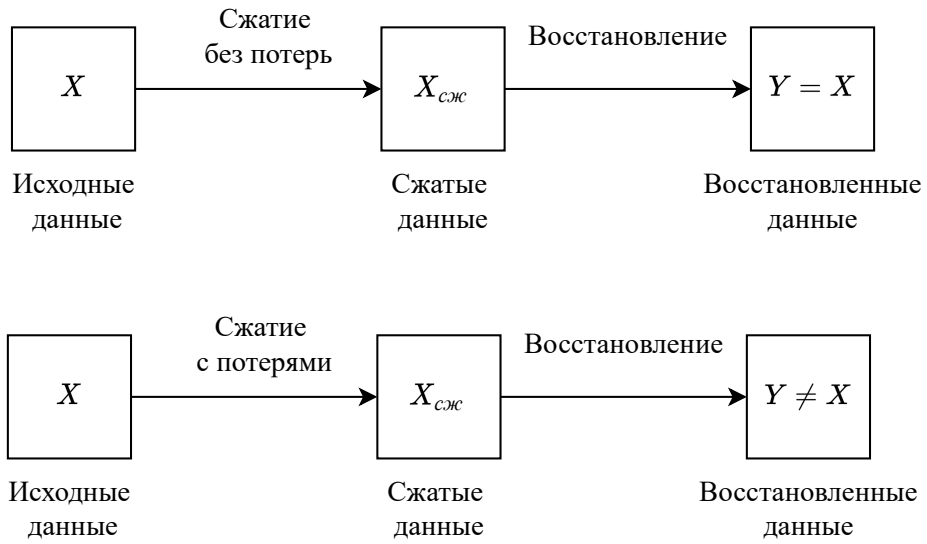

Рисунок 1.4 – Схемы сжатия данных без потерь и с потерями

Для измерения производительности сжатия используют характеристику, которую называют коэффициентом сжатия и определяют следующим образом [3]:

$$K_{\text{сж}} = \frac{L_{\text{исх}}}{L_{\text{сж}}}, \tag{1.1}$$

где $L_{\text{исх}}$ — объем исходных данных, $L_{\text{сж}}$ — объем сжатых данных.

1.4 Сжатие данных в ядре Linux

TODO: Описание модуля zram.

1.5 Информационная энтропия

Под информацией понимают сведения, которые являются объектом хранения, передачи и обработки [4]. Формой представления информации является сообщение. Физическую величину, отображающую сообщение, называют сигналом. Передача информации осуществляется следующим образом [5]:

1. Источник информации создает случайное сообщение. В теории информации любой источник информации является стохастическим, его можно описать измеряемыми вероятностными категориям.

2. Сообщение поступает в систему передачи, в которой выполняется кодирование — преобразование сообщения с целью согласования источника информации с каналом связи для увеличения скорости передачи информации или обеспечения заданной помехоустойчивости [4]. Кодирование состоит из шифрования, сжатия и защиты от шума, в результате которых формируется сигнал.

3. Сигнал проходит через канал — среду передачи информации [5]. В канале могут возникать помехи, создаваемые источником шума.

4. Сигнал подается на вход системе приема, которая выполняет декодирование — восстановление исходного сообщения [4].

5. Исходное сообщение передается получателю.

Описанная схема передачи информации показана на рисунке 1.5.

Сообщение содержит сведения о некоторой физической системе X, которая случайным образом может перейти в какое-либо состояние x_i из конечного множества состояний $x_1, x_2, \ldots, x_n$ с вероятностями $p_1, p_2, \ldots, p_n$, где $n \in \mathbb{N}$,

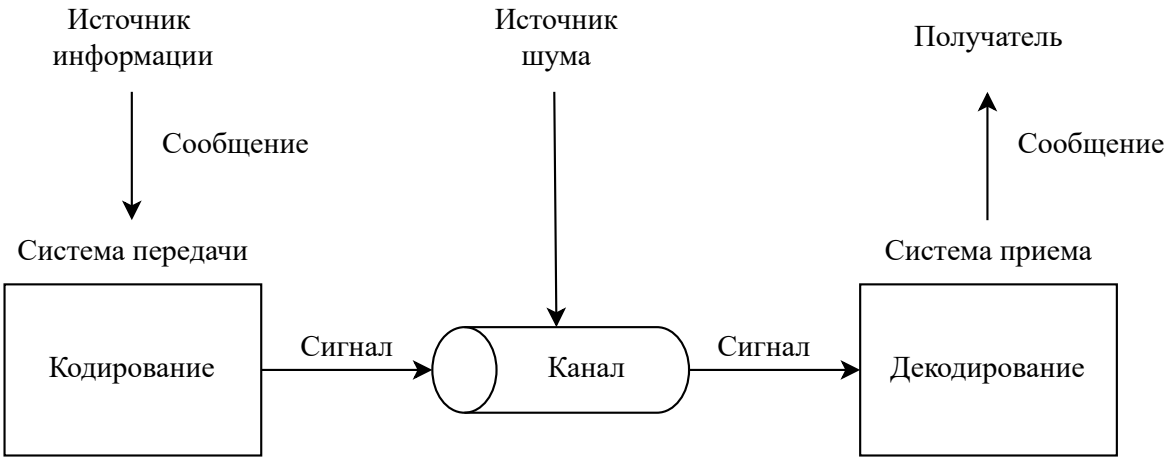

Рисунок 1.5 – Схема передачи информации

$p_i = P(X \sim x_i)$ и $\sum_{i=1}^{n} p_i = 1$. То есть, для такой системы существует степень неопределенности, которая описывается числом ее возможных состояний и их вероятностями. Сведения из принятого сообщения тем ценнее, чем больше была неопределенность системы до получения сообщения. Специальную характеристику, используемую в качестве меры неопределенности системы, называют информационной энтропией [6]. Информационная энтропия конечной вероятностной схемы определяется по формуле Шеннона:

$$H(X) = -\sum_{i=1}^{n}(p_i \cdot \log_a p_i), \tag{1.2}$$

где $p_i \in [0, 1]$, $a > 1$.

Основание логарифма определяет единицы измерения информационной энтропии: при $a = 2$ энтропия измеряется в битах, при $a = 3$ — в тритах, при $a = e$ — в натах.

1.5.1 Свойства информационной энтропии

Информационная энтропия обладает следующими свойствами [7]:

1. Энтропия всегда неотрицательна. Значения $\log_a p_i$ в формуле (1.2) принимают неположительные значения, так как $p_i \in [0, 1]$. Поэтому

$$H(X) = -\sum_{i=1}^{n}(p_i \cdot \log_a p_i) \geqslant 0. \tag{1.3}$$

2. Энтропия равна нулю, если состояние системы в точности известно заранее. Если известно состояние x_k, в которое перейдет система X, то вероятность этого состояния p_k равна единице, вероятности других состояний равны нулю. Тогда

$$p_k \cdot \log_a p_k = 1 \cdot \log_a 1 = 1 \cdot 0 = 0. \tag{1.4}$$

В связи с тем, что $\lim\limits_{p \to 0}(p \cdot \log_a p) = 0$, другие слагаемые суммы в формуле (1.2) равны нулю. В этом случае

$$H(X) = -\sum_{i=1}^{n}(p_i \cdot \log_a p_i) = 0. \tag{1.5}$$

3. Энтропия принимает наибольшее значение при условии, что все состояния равновероятны, то есть, $p_1 = p_2 = \cdots = p_n = \frac{1}{n}$. Тогда

$$H(X) = -\sum_{i=1}^{n}(p_i \cdot \log_a p_i) = -\sum_{i=1}^{n}(\frac{1}{n} \cdot \log_a \frac{1}{n}) = -\log_a \frac{1}{n} = \log_a n. \tag{1.6}$$

1.5.2 Связь информационной энтропии и коэффициента сжатия

Данные, обладающие предсказуемой структурой, сокращают неопределенность системы меньше, чем сведения, в которых никакая структура не выделяется. Так как информационная энтропия является мерой неопределенности системы, то данные с выделяемой структурой, имеют низкое значение энтропии. Сведения, в которых закономерности не определяются, имеют высокое значение энтропии [8]. Так, чем меньше избыточность данных, тем выше значение их энтропии. То есть, информационная энтропия сжатых данных выше, чем ее значение до сжатия.

Согласно теореме Шеннона об источнике шифрования сигнал, облада-

ющий размером S и информационной энтропией H, не может быть сжат менее, чем до $S \cdot H$ битов без потери точности информации. Таким образом, на основании информационной энтропии исходных данных определяется теоретическая граница коэффициента сжатия [9].

В связи с применением операций сложения, умножения и логарифмирования при вычислении энтропии по формуле (1.2), число которых растет с увеличением объема данных, встает задача выбора метода подсчета. Использование метода влияет на скорость и время определения информационной энтропии.

1.6 Методы подсчета информационной энтропии

1.6.1 Метод скользящего окна

При решении задач, связанных со сжатием, данные X представляют собой массив байтов размером N [10]. Байт состоит из восьми битов, каждый из которых кодирует одно из значений множества $\{0, 1\}$. Поэтому один байт может принимать значения из интервала от 0 до 255 включительно в десятичной системе счисления.

В методе скользящего окна под окном понимают рассматриваемую на текущем этапе подпоследовательность данных размером n [11]. При подсчете энтропии данным методом необходима дополнительная память размером 2^n для хранения числа вхождений подпоследовательностей данных. Так как с увеличением размера окна, растет объем дополнительной памяти, в качестве окна выбирается минимально адресуемая единица памяти, которой является байт [12]. В связи с тем, что на каждом этапе окно смещается на следующие восемь битов, как показано на рисунке 1.6, оно называется скользящим.

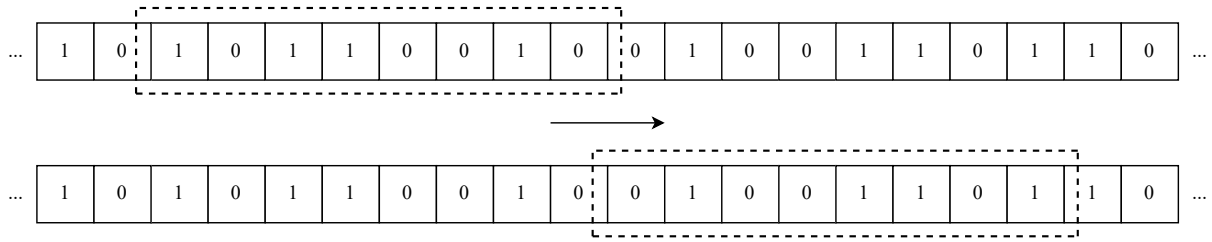

Рисунок 1.6 – Проход по массиву байтов методом скользящего окна

Вычисление информационной энтропии методом скользящего окна вклю-

чает в себя два шага:

1. Для каждого возможного значения байта подсчитывается число его вхождений k_i в массив байтов, где $i = \overline{0, 255}$.

2. С учетом того, что вероятность появления байта в массиве $p_i = \frac{k_i}{N}$, информационная энтропия вычисляется по следующей формуле:

$$H(X) = -\sum_{i=0}^{255}(p_i \cdot \log_2 p_i). \tag{1.7}$$

Так как первый шаг метода предполагает проход по массиву байтов размером N, а второй шаг — проход по массиву числа вхождений подпоследовательностей данных размером 2^n, временная сложность метода скользящего окна — $O(N + 2^n)$.

Информационная энтропия, подсчитанная методом скользящего окна, принимает значения из интервала $[0, 8]$ битов. При этом согласно свойству 2 из подраздела 1.5.1 информационная энтропия принимает нулевое значение в случае, когда массив данных состоит из одинаковых байтов, и в соответствии со свойством 3 из подраздела 1.5.1 максимальное значение, равное восьми, если все байты в массиве различны.

1.6.2 Биномиальный метод

В биномиальном методе вычисления информационной энтропии [13] данные X рассматриваются как последовательность сообщений, генерируемых бернуллиевским источником — источником информации, порождающим символы из алфавита $\{0; 1\}$ с вероятностями $1 - p$ и p соответственно, причем $p \in (0, 1)$ и может быть неизвестно [14]. То есть, сообщение представляет собой последовательность битов длины n.

Вероятность того, что сообщение содержит k единиц, где $k = \overline{0, n}$, вычисляется следующим образом:

$$P_k = p^k \cdot (1 - p)^{(n-k)}. \tag{1.8}$$

Количество возможных сообщений, содержащих k единиц, определяется как биномиальный коэффициент:

$$C_n^k = \frac{n!}{k! \cdot (n-k)!}. \tag{1.9}$$

В связи с тем, что k может принимать значения из интервала $[0, n]$, число биномиальных коэффициентов, подсчитанных по формуле (1.9), равно $n+1$. Это означает, что сообщения можно разбить на $n+1$ классов эквивалентности. Тогда информационная энтропия вычисляется так:

$$H(X) = -\sum_{k=0}^{n}(C_n^k \cdot P_k \cdot \log_2 P_k). \tag{1.10}$$

В соответствии с формулой (1.10) биномиальный метод подсчета информационной энтропии состоит из следующих этапов:

1. Исходные сообщения разбиваются на $n+1$ классов эквивалентности, сообщения которых содержат $k = \overline{0, n}$ единиц.

2. Для каждого класса эквивалентности рассчитывается биномиальный коэффициент C_n^k.

3. Определяются вероятности p появления единицы в сообщениях.

4. Вычисляются вероятности P_k по формуле (1.8).

5. Суммируются произведения биномиальных коэффициентов C_n^k, вероятностей P_k и логарифмов вероятностей P_k для всех $n+1$ значений k.

Схема определения биномиальных коэффициентов C_n^k и вероятностей P_k представлена на рисунке 1.7.

Для определения вероятности p появления единицы в сообщениях на третьем этапе биномиального метода необходима дополнительная память размером $n+1$. При этом данный этап подразумевает проход по массиву байтов размером N. Последний этап вычисления предполагает проход по массиву, содержащему значения вероятностей p появления единицы в сообщениях. Тогда временная сложность биномиального метода подсчета информационной энтропии — $O(N+n)$.

В биномиальном методе определяются вероятности появления не подпоследовательности, а единицы в подпоследовательностях. Кроме того, расчет

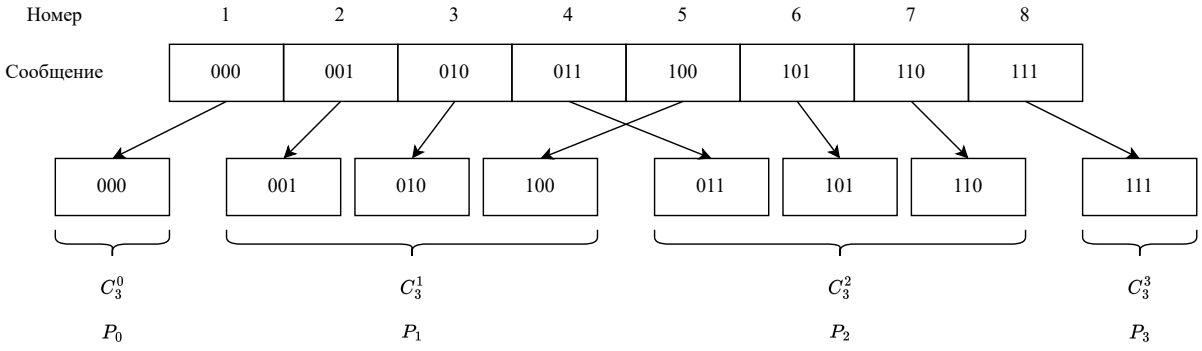

Рисунок 1.7 – Определение биномиальных коэффициентов C_n^k и вероятностей P_k

ускоряется за счет разделения исходной последовательности на классы эквивалентности, что приводит к сокращению операций сложения.

При рассмотрении бернуллиевского источника информации предполагается, что вероятность появления единицы в подпоследовательности не зависит от вероятностей появления нуля или единицы в битах предыдущей подпоследовательности. При наличии такой зависимости вычисленное значение энтропии будет завышено. Так как рассматриваемое представление данных — последовательность битов, то вероятности появления каждого значения бита независимы.

Недостатком данного метода является трудоемкость вычисления факториала при определении биномиальных коэффициентов с увеличением длины подпоследовательности. Для снижения времени подсчета биномиальных коэффициентов можно хранить их значения в дополнительном массиве. Так как для сообщения длины n количество требуемых для определения энтропии биномиальных коэффициентов равно $n + 1$, то для их хранения потребуется память размером $n + 1$.

1.6.3 Сравнение существующих методов подсчета

При вычислении информационной энтропии методом скользящего окна и биномиальным методом операции сложения, умножения и логарифмирования применяются к целым числам и числам с плавающей запятой.

Для сравнения методов подсчета информационной энтропии были выделены следующие критерии оценки:

— К1 — временная сложность;

— К2 — необходимость вычисления факториала;

— К3 — возможность распараллеливания вычислений;

— К4 — объем требуемой дополнительной памяти.

Результаты сравнения представлены в таблице 1.1.

Таблица 1.1 – Сравнение методов подсчета информационной энтропии

Метод	**К1**	**К2**	**К3**	**К4**
Скользящего окна	$O(N + 2^n)$	$-$	$+$	2^n
Биномиальный	$O(N + n)$	$+$	$+$	$2 \cdot (n + 1)$

Таким образом, биномиальный метод подсчета информационной энтропии требует меньших вычислительных затрат по времени и по памяти.

1.7 Постановка задачи

TODO

2 Конструкторский раздел

3 Технологический раздел

4 Исследовательский раздел

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Uthayakumar J.*, *Vengattaraman T.*, *Dhavachelvan P.* A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications // Journal of King Saud University — Computer and Information Sciences. — 2021. — 119—140 с.
2. *Пантелеев Е. Р.*, *Алыкова А. Л.* Алгоритмы сжатия данных без потерь: учебное пособие для вузов // 2-е изд., стер. — СПб.: Лань. — 2022. — 172 с.
3. *Gupta A.*, *Nigam S.* A Review on Different Types of Lossless Data Compression Techniques // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. — 2021. — 50—56 с.
4. *Березкин Е. Ф.* Основы теории информации и кодирования: учебное пособие // 3-е изд., стер. — СПб.: Лань. — 2022. — 320 с.
5. *Rodrigues M.* Information-Theoretic Methods in Data Science // Cambridge: Cambridge University Press. — 2021. — 43 с.
6. *Попов И. Ю.*, *Блинова И. В.* Теория информации // 3-е изд., стер. — СПб.: Лань. — 2022. — 160 с.
7. *Осокин А. Н.*, *Мальчуков А. Н.* Теория информации: учебное пособие для вузов // М.: Издательство Юрайт. — 2022. — 205 с.
8. *Zbili M.*, *Rama S.* A Quick and Easy Way to Estimate Entropy and Mutual Information for Neuroscience // Frontiers in Neuroinformatics. — 2021.
9. *Cheng X.*, *Li Z.* How does Shannon's source coding theorem fare in prediction of image comrpession ratio with current algorithms? // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. — 2020. — 1313—1319 с.
10. *Ryabko B.* Time-Universal Data Compression // Algorithms. — 2019.
11. *Guo H.* File Entropy Signal Analysis Combined With Wavelet Decomposition for Malware Classification // IEEE Access. — 2020. — 158961—158971 с.
12. *Пухальский Г. И.*, *Новосельцева Т. Я.* Проектирование цифровых устройств: учебное пособие для вузов // СПб.: Лань. — 2022. — 896 с.

13. *Borysenko O.* On the binomial method for calculation of entropy // Grail of Science. — 2022. — 113—118 с.

14. *Рябко Б. Я.*, *Фионов А. Н.* Криптография в информационном мире // М.: Горячая линия-Телеком. — 2018. — 300 с.

ПРИЛОЖЕНИЕ А