БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Реферат. Хэш-таблицы.**

**По:** Основы алгоритмизации и программирования

**Выполнил:** Корнелюк

Валентин Владимирович

1 курс 4 группа ПОИТ

**Преподаватель:**

Белодед Николай Иванович

**г. Минск**

2023 г.

**Оглавление**

[Пример дека на основе двунаправленного списка. Построение дека, вставка и удаление элементов из дека. 3](#_Toc132752315)

## **Введение.**

Цель реферата.

Хэш-функция является одним из ключевых понятий современной информационной технологии, широко используемым для обеспечения безопасности, целостности и быстроты обработки данных. Целью данного реферата является изучение различных типов хэш-функций и их особенностей.

Определение понятия “Хэш-функция”.

Хэш-функция представляет собой математическую функцию, которая преобразует входные данные фиксированной длины в выходные данные фиксированной длины. В отличие от криптографических алгоритмов, которые используют ключ для защиты данных, хэш-функции не используют ключи и работают быстро и эффективно на больших объемах данных. Основной задачей хэш-функций является обеспечение уникальности для каждого входного значения, то есть невозможности получения одного и того же хэша для различных входных данныхю

Значимость хэш-функций в современных информационных технологиях.

Хэш-функции широко применяются в современный информационных технологиях, включая базы данных, поисковые системы, криптографию и блокчейн-технологии. Они используются для обеспечения безопасности данных и защиты от взлома паролей, проверки целостности данных, идентификации вредоносных программ, ускорения процесса поиска и многих других задач. В современном мире без использования хэш-функций невозможно представить безопасное хранение и передачу данных.

Понимание и использование различных типов хэш-функций является критически важным в современной информационной технологии. Данный реферат предоставит полезную информацию о различных типах хэш-функций и их особенностях, что позволит улучшить безопасность и эффективность обработки данных.

## **1. Традиционные хэш-функции.**

Особенности.

Традиционные хэш-функции - это хэш-функции, которые были разработаны до появления криптографических хэш-функций. Их особенность заключается в том, что они были разработаны для работы с данными фиксированного размера и были использованы в основном для ускорения обработки данных и оптимизации использования памяти. В отличие от криптографических хэш-функций, которые используются для обеспечения безопасности данных, традиционные хэш-функции используются в основном для поиска и индексации данных.

Алгоритмы и примеры традиционных хэш-функций.

Наиболее известными алгоритмами традиционных хэш-функций являются:

1. CRC (Cyclic Redundancy Check) - это алгоритм, который используется для проверки целостности данных. Он был разработан в 1961 году и был использован во многих стандартах, таких как Ethernet, USB, SATA и других.

2. MD5 (Message-Digest algorithm 5) - это алгоритм, который создает хэш-значение фиксированной длины 128 бит. Он был разработан в 1991 году и широко использовался в качестве алгоритма хэширования паролей и проверки целостности данных. Однако, MD5 считается небезопасным и в настоящее время не рекомендуется к использованию.

3. SHA-1 (Secure Hash Algorithm 1) - это алгоритм, который создает хэш-значение фиксированной длины 160 бит. Он был разработан в 1995 году и широко использовался для цифровой подписи, проверки целостности данных и других задач. Однако, SHA-1 также считается небезопасным и был заменен на более безопасные криптографические хэш-функции, такие как SHA-256 и SHA-3.

Применение традиционных хэш-функций.

Традиционные хэш-функции используются во многих областях, таких как базы данных, поисковые системы, сетевые протоколы и другие. Например, CRC используется для проверки целостности данных во многих протоколах передачи данных, таких как Ethernet, Wi-Fi и других. MD5 был широко использован в качестве алгоритма хэширования паролей и проверки целостности данных, но в настоящее время считается небезопасным. SHA-1 также широко использовался для цифровой подписи, проверки целостности данных и других задач, но сейчас также считается небезопасным.

Традиционные хэш-функции также могут быть использованы для сжатия данных. Например, алгоритм SHA-1 используется в сжатии ZIP-архивов для уменьшения объема передаваемых данных.

Еще одним примером применения традиционных хэш-функций является использование их в системах контроля версий. Контроль версий - это процесс отслеживания изменений в коде программного обеспечения. Хэш-функции используются для создания уникальных идентификаторов для каждой версии кода. Это позволяет быстро определить, какие изменения были внесены в каждую версию, и восстановить предыдущие версии при необходимости.

Традиционные хэш-функции также широко используются в базах данных для поиска и индексации данных. Они позволяют быстро находить данные в больших объемах информации, оптимизируя процесс обработки данных и ускоряя поиск.

Таким образом, традиционные хэш-функции имеют широкое применение в различных областях информационных технологий, обеспечивая оптимизацию процессов обработки данных и безопасность информации. Однако, с развитием технологий и появлением криптографических хэш-функций, традиционные хэш-функции все больше замещаются более безопасными и эффективными алгоритмами.

## **2. Криптографические хэш-функции.**

Особенности.

Криптографические хэш-функции - это хэш-функции, которые обладают специальными свойствами безопасности и используются в криптографии для целей аутентификации, цифровой подписи, шифрования и других задач, где требуется высокая степень защиты данных. Криптографические хэш-функции обладают следующими особенностями:

1. Сопротивление коллизиям. Криптографические хэш-функции должны быть устойчивы к коллизиям, то есть к ситуации, когда два разных входных сообщения имеют одинаковый хэш-код.

2. Односторонняя функция. Криптографическая хэш-функция должна быть односторонней, то есть невозможно восстановить исходное сообщение по его хэш-коду.

3. Случайность. Криптографические хэш-функции должны обладать свойством случайности, то есть каждый хэш-код должен быть уникальным и непредсказуемым.

Алгоритмы и примеры криптографических хэш-функций.

Существует множество алгоритмов криптографических хэш-функций, включая MD5, SHA-1, SHA-2 и другие. Некоторые из них были разработаны более десяти лет назад, и с тех пор были обнаружены уязвимости. Например, MD5 и SHA-1 считаются устаревшими и небезопасными для использования, в то время как более новые алгоритмы, такие как SHA-256 и SHA-3, считаются более безопасными.

Применение криптографических хэш-функций.

Криптографические хэш-функции широко применяются в сфере информационной безопасности. Они используются для генерации дайджестов сообщений, которые позволяют эффективно проверять целостность сообщений и файлов. Криптографические хэш-функции также используются для хеширования паролей и других конфиденциальных данных.

Примером криптографической хэш-функции, которая получила широкое распространение, является MD5. Она используется для хеширования паролей и других конфиденциальных данных. Однако, MD5 считается устаревшей и небезопасной в связи с возможностью коллизий, когда два разных сообщения могут дать одинаковый хэш.

Криптографические хэш-функции также используются для создания цифровых подписей. Цифровая подпись - это математический способ связывания сообщения с отправителем, который гарантирует его подлинность и невозможность подделки.

Интересный пример использования криптографической хэш-функции - блокчейн технология, которая используется в криптовалютах. Криптографические хэш-функции обеспечивают безопасность транзакций, путем генерации уникальных идентификаторов для каждой транзакции.

В заключение, криптографические хэш-функции играют важную роль в обеспечении безопасности информации в цифровом мире. Они используются в широком спектре приложений, включая проверку целостности файлов, защиту паролей и создание цифровых подписей.

## **3. Современные хэш-функции.**

Особенности.

Современные хэш-функции отличаются от традиционных и криптографических тем, что они являются не только простыми и безопасными функциями хеширования, но и обладают дополнительными функциями, такими как индексирование, поиск и сжатие данных. Они могут использоваться для хранения паролей, проверки целостности данных, а также для создания цифровых подписей. Современные хэш-функции обладают высокой производительностью и могут обрабатывать большие объемы данных.

Алгоритмы и примеры современных хэш-функций.

Существует множество современных хэш-функций, некоторые из них являются открытыми стандартами, например, SHA-3, BLAKE2, Skein и другие. Другие функции были разработаны для конкретных задач и применений. Например, хэш-функция MurmurHash была создана для использования в распределенных системах.

SHA-3 - это открытый стандарт, который был выбран в 2012 году Национальным институтом стандартов и технологий (NIST) в качестве нового стандарта для криптографических хэш-функций. SHA-3 основан на алгоритме Keccak и обладает высокой стойкостью к атакам и хорошей производительностью.

BLAKE2 - это современная хэш-функция, которая обеспечивает высокую скорость хеширования и высокую стойкость к атакам. Она является улучшенной версией BLAKE, обладает большим числом режимов работы и поддерживает как 32-битные, так и 64-битные платформы.

MurmurHash - это хэш-функция, которая была разработана для использования в распределенных системах. Она обладает высокой производительностью и хорошей стойкостью к коллизиям.

Применение современных хэш-функций.

Современные хэш-функции нашли широкое применение в различных областях информационных технологий. Некоторые примеры их использования включают:

1. Безопасное хранение паролей: современные хэш-функции, такие как Argon2, используются для безопасного хранения паролей в базах данных. Эти функции обладают высокой стойкостью к атакам типа "перебора пароля" и "словарных атак".

2. Цифровые подписи: криптографические хэш-функции, такие как SHA-3 и BLAKE2, используются для создания цифровых подписей, которые обеспечивают аутентификацию и целостность данных.

3. Блокчейн: современные хэш-функции, такие как SHA-3, используются в технологии блокчейн для создания уникальных идентификаторов транзакций, блоков и цепочек блоков. Это обеспечивает надежность и невозможность изменения данных в блокчейн-сети.

4. Контроль целостности данных: современные хэш-функции используются для проверки целостности данных в хранилищах, копировании файлов, а также в других случаях, когда необходимо убедиться, что данные не были изменены после их создания.

5. Различные алгоритмы: современные хэш-функции применяются в различных алгоритмах, таких как алгоритмы поиска, алгоритмы сравнения, алгоритмы индексации и другие.

Стоит отметить, что современные хэш-функции не являются универсальными решениями для всех задач, связанных с хэшированием данных. В некоторых случаях для решения конкретных задач могут быть использованы специализированные хэш-функции, разработанные с учетом определенных требований к скорости работы, объему данных и другим параметрам.

## **4. Сравнение и анализ различных типов хэш-функций.**

Хэш-функции - важный инструмент в современной информационной технологии, используемый для решения различных задач, таких как аутентификация, хранение паролей, проверка целостности данных и других. Существует несколько типов хэш-функций, каждый из которых имеет свои особенности и применения.

Критерии сравнения.

Критерии сравнения:

• Безопасность: насколько сложно подобрать такой входной набор данных, при котором хэш-функция вернет такое же значение, как и для другого набора данных. Хэш-функции, используемые в криптографии, должны быть устойчивы к атакам подбора.

• Скорость: время, затрачиваемое на вычисление хэш-значения.

• Размер выходных данных: размер хэш-значения, который может быть от 128 до 512 бит.

Сильные и слабые стороны каждого типа хэш-функций.

Традиционные хэш-функции: Традиционные хэш-функции, такие как MD5, SHA-1 и SHA-2, имеют размер выходных данных от 128 до 512 бит, что делает их хорошим выбором для задач, требующих высокой скорости и эффективности. Однако эти функции могут быть подвержены коллизиям, что означает, что два разных входных значения могут иметь одно и то же хэш-значение.

Криптографические хэш-функции: Криптографические хэш-функции, такие как SHA-1 и SHA-2, разработаны для защиты данных от подделки и несанкционированного доступа. Они обеспечивают более высокий уровень безопасности, чем традиционные хэш-функции, и могут использоваться в качестве ключевых компонентов криптографических протоколов. Однако, из-за их увеличенного размера выходных данных, они могут быть менее эффективны в задачах, где требуется высокая скорость.

Современные хэш-функции, такие как BLAKE2 и SipHash, разработаны с учетом последних научных достижений в области криптографии и могут обеспечить более высокую скорость работы и безопасность, чем традиционные хэш-функции и криптографические хэш-функции.

Таким образом, при выборе хэш-функции необходимо учитывать конкретные задачи и требования к безопасности и скорости работы, а также принимать во внимание сильные и слабые стороны каждого типа хэш-функций.

Применение каждого типа хэш-функций в конкретных задачах.

Рассмотрим применение различных типов хэш-функций в конкретных задачах.

Традиционные хэш-функции часто используются в различных алгоритмах сжатия данных, таких как алгоритмы сжатия без потерь LZ77 и LZ78, а также алгоритмы сжатия с потерями, например, JPEG и MPEG. Они также находят применение в хранилищах данных и базах данных, где могут использоваться для ускорения поиска и сравнения записей. Также традиционные хэш-функции могут использоваться для проверки целостности данных, например, при загрузке файлов или передаче данных по сети.

Криптографические хэш-функции нашли широкое применение в области криптографии и безопасности информации. Они используются для обеспечения целостности данных, аутентификации и защиты от подделки данных. Криптографические хэш-функции используются в электронной подписи, цифровом сертификате, SSL/TLS-соединениях, а также в различных протоколах и приложениях, связанных с безопасностью данных.

Современные хэш-функции находят широкое применение в различных областях, включая информационную безопасность, биоинформатику, компьютерное зрение и машинное обучение. В биоинформатике, например, современные хэш-функции используются для сравнения и анализа геномов, а в компьютерном зрении - для распознавания образов и обработки изображений.

При выборе хэш-функции для конкретной задачи следует учитывать ее свойства и особенности, а также соответствие требованиям безопасности и целостности данных. Важно также учитывать производительность и эффективность хэш-функции в конкретных условиях использования.

## **Заключение.**

В данном реферате были рассмотрены различные типы хэш-функций и их особенности. Введение пояснило, что такое хэш-функция и почему она является важной в современной информационной технологии.

Затем были описаны традиционные хэш-функции, их алгоритмы и применение. Криптографические хэш-функции были рассмотрены отдельно, так как они используются в криптографии для защиты данных и подписи сообщений.

Также были представлены современные хэш-функции, которые имеют более высокий уровень безопасности и производительности, чем традиционные хэш-функции.

В разделе сравнения и анализа были рассмотрены критерии сравнения, сильные и слабые стороны каждого типа хэш-функций, а также применение каждого типа хэш-функций в конкретных задачах.

В заключении подведены итоги рассмотренных типов хэш-функций, их применения и важности в современной информационной технологии. Отмечены перспективы развития хэш-функций, которые будут обладать еще более высоким уровнем безопасности и производительности.

Таким образом, хэш-функции являются неотъемлемой частью современной информационной технологии и играют важную роль в обеспечении безопасности и целостности данных, а также в решении различных задач, связанных с хранением, передачей и обработкой информации.

## **Список литературы.**

В рамках данного реферата были использованы следующие источники:

1. Ахо, А. В., Хопкрофт, Дж. Е., Ульман, Дж. Дж. Структуры данных и алгоритмы. — 2-е издание. — М.: Вильямс, 2019. — 992 с.

2. Wikipedia. Hash function. — https://en.wikipedia.org/wiki/Hash\_function