Содержание

Введение 4
1 Лекция № 1. Введение 5
2 Лекция №2. Основные понятия и преобразования информации 11
3 Лекция №3. Меры дискретной информации 15
4 Лекция №4. Алфавитное кодирование информации 23
5 Лекция № 5. Представление числовой информации в цифровых устройствах 60
6 Лекция № 6. Эффективное кодирование 66
7 Лекция №7. Передача информации 86
8 Лекция №8. Кодирование и шифрование информации 98
Список литературы 106

Введение

Задачами изучения дисциплины являются приобретение студентами знаний в области:

– мер информации;

– преобразования и передачи информации;

– кодирования информации;

– систем счисления и форм представления информации в цифровых устройствах;

– применения теории информации в системах информационной безопасности.

В результате изучения дисциплины ИОЗИ студент должен обладать знаниями и освоить практически основные технологии информационных основ защиты информации в телекоммуникационных системах, базовые методы применяемые в информационных основах защиты информации, уметь применять на практике базовые технологии информационных основ защиты информации и показать необходимость использования информационных основ защиты информации при изучении вопросов обеспечения защиты и безопасности телекоммуникационных систем и сетей, выработать умение использовать нормативные и правовые акты, содержащие нормы безопасности защиты информации, а также осуществлять контроль за их соблюдением.

1 Лекция № 1. Введение

Цель лекции: изучить основные понятия, методы и средства защиты информации.

Содержание лекции: определение цели предмета ИОЗИ. Методы и средства защиты информации. Роль защиты информации в современном мире.

Информационная безопасность является одним из важнейших аспектов любого уровня безопасности – национального, отраслевого, корпоративного или персонального. Основная проблема ИБ состоит в том, что она является составной частью информационных технологий. Технологии программирования не позволяют создавать безошибочные программы, что не может обеспечить информационную безопасность. То есть существует необходимость создавать надежные системы ИБ с использованием ненадежных программ. Такая необходимость требует соблюдения архитектурных принципов и контроля защищенности при использовании ИС. Также с развитием информационно-коммуникационных технологий, с постоянным использованием сетей значительно выросло количество атак. Но это нельзя назвать самой большой проблемой информационной безопасности, так как гораздо значительнее проблемы, связанные с постоянным обнаружением новых уязвимых мест в программном обеспечении и, как следствие, появления новых видов атак. Над уничтожением таких уязвимых мест трудятся разработчики абсолютно всех разновидностей операционных систем, поскольку новые ошибки начинают активно использоваться злоумышленниками.

В таких случаях системы ИБ должны иметь средства противостояния разнообразным атакам. Атаки могут длиться как доли секунды, так и несколько часов, медленно прощупывая уязвимые места (в таком случае вредоносная активность практически незаметна). Действия злоумышленников могут быть нацелены на нарушение как отдельно взятых, так и всех составляющих ИБ – доступности, целостности и конфиденциальности.

Целями защиты являются: предотвращение утечки, хищения, утраты, искажения, подделки, несанкционированных действий по уничтожению, модификации, копированию, блокированию документированной информации и иных форм незаконного вмешательства в информационные системы.

Информация может существовать в виде документа (бумажного), в виде физических полей и сигналов (электромагнитных, акустических, тепловых и т. д.), в виде биологических полей (память человека).

Природа воздействия на ИО может быть двух видов: непреднамеренной (стихийные бедствия, отказы, ошибки персонала и т. д.); преднамеренной (действия злоумышленников). Все воздействия могут привести к последствиям (ущербу) трех видов: нарушению конфиденциальности; нарушению целостности; нарушению доступности.

Нарушение конфиденциальности – нарушение свойства информации быть известной только определенным субъектам.

Нарушение целостности – несанкционированное изменение, искажение, уничтожение информации.

Нарушение доступности (отказ в обслуживании) – нарушается доступ к информации, нарушается работоспособность объекта, доступ в который получил злоумышленник.

В отличие от разрешенного (санкционированного) доступа к информации в результате преднамеренных действий злоумышленник получает несанкционированный доступ (НСД). Суть НСД состоит в получении нарушителем доступа к объекту в нарушении установленных правил.

Целью защиты ИО является противодействие угрозам безопасности.

Защищенный ИО – объект со средствами защиты, которые успешно и эффективно противостоит угрозам безопасности.

Комплексная защита ИО – совокупность методов и средств (правовых, организационных, физических, технических, программных).

Политика безопасности – совокупность норм, правил, рекомендаций, регламентирующих работу средств защиты ИО от заданного множества угроз безопасности.

1.1 Методы защиты информации

Для обеспечения безопасности информации в ИС используются следующие методы:

– препятствие;
– управление доступом;
– методы криптографии;
– противодействие атакам вредоносных программ;
– регламентация;
– принуждение;
– побуждение.

Препятствие – физическое преграждение пути к защищаемой информации (к техническому оборудованию, носителям информации и т.д.).

Управление доступом – методы защиты информации через регулирование использования ресурсов информационных технологий и информационной системы. Управление доступом должно препятствовать абсолютно всем возможным путям несанкционированного доступа к защищаемой информации.

Защита информации с помощью управления доступом происходит через:

– идентификацию пользователей и персонала (присвоение персонального идентификатора);

– опознание объекта по идентификатору;

– проверку полномочий доступа к информации или объекту;

– регистрацию обращений к информации;

– реагирование (сигнализация, отключение, задержка работ, отказ в запросе и т.п.) при попытках несанкционированных действий.

Криптографические методы защиты – шифрование информации. Методы шифрования широко применяются при обработке и хранении информации. Особенно надежным данный метод является при передаче информации по сети.

Защиту от атак вредоносных программ призван обеспечить комплекс различных методов организационного характера и использование антивирусных программ, результатом чего является снижение вероятности заражения ИС, определение фактов инфицирования системы; снижение или предотвращение последствий информационных заражений, уничтожение вирусов; последующее восстановление информации.

Регламентация – ограничение времени работы, ограниченный доступ людей к информации, ограничение доступа по определенным дням, времени суток, часам и т.п. Создание таких условий работы с защищаемой информацией нормы и стандарты по защите будут выполняются в наибольшей степени.

Принуждение – метод защиты информации, при котором пользователи и персонал ИС соблюдают правила работы с защищаемой информацией под угрозой ответственности (материальной, административной или уголовной).

Побуждение – метод, который побуждает (за счет соблюдения уже сложившихся морально-этических норм) субъектов ИС не нарушать установленные порядки.

Технические средства защиты информации делятся на аппаратные и физические. К аппаратным средствам относятся устройства, которые встраиваются непосредственно в техническое оборудование ИС или связываются с ним по стандартному интерфейсу. К физическим средствам относят инженерные устройства и сооружения, которые препятствуют физическому проникновению на объекты защиты и осуществляют защиту персонала, материальных, информационных и других ценностей (например, решетки, замки, сейфы, сигнализация и т.п.). Также широко используются программные средства, предназначенные для защиты информации в ИС. К ним относятся программы паролирования, антивирусные программы, программы ограничения доступа, программы шифрования (криптографии). Организационные средства обеспечивают мероприятия, которые делают невозможными или затрудняют разглашение, утечку, несанкционированный доступ к информации на нормативно-правовой основе. Законодательные средства защиты регламентируют правила работы с информацией и устанавливают порядок ответственности за их нарушение. Законодательные средства защиты определяются законодательными актами страны. Морально-этические средства защиты включают нормы поведения, которые могут быть

неписанными (например, честность) или оформленными в виде правил и предписаний. Как правило, они не утверждены законодательством, но считаются обязательными для исполнения. Примером таких правил может быть свод этических правил общения в сети и т.п.

1.2 Средства защиты информации

Для целей разработки концепций ИБ-защиты средства защиты информации принято делить на нормативные (неформальные) и технические (формальные).

Неформальные средства защиты – это документы, правила, мероприятия, формальные – это специальные технические средства и программное обеспечение. Разграничение помогает распределить зоны ответственности при создании ИБ-систем: при общем руководстве защитой административный персонал реализует нормативные способы, а IT-специалисты, соответственно, технические.

Основы информационной безопасности предполагают разграничение полномочий не только в части использования информации, но и в части работы с ее охраной. Подобное разграничение полномочий требует и нескольких уровней контроля.

Физические средства защиты. Это механические, электрические, электронные механизмы, которые функционируют независимо от информационных систем и создают препятствия для доступа к ним. Замки, в том числе электронные, экраны, жалюзи призваны создавать препятствия для контакта дестабилизирующих факторов с системами. Группа дополняется средствами систем безопасности, например, видеокамерами, видеорегистраторами, датчиками, выявляющие движение или превышение степени электромагнитного излучения в зоне расположения технических средств снятия информации, закладных устройств.

Аппаратные средства защиты. Это электрические, электронные, оптические, лазерные и другие устройства, которые встраиваются в информационные и телекоммуникационные системы. Перед внедрением аппаратных средств в информационные системы необходимо удостовериться в совместимости.

Программные средства – это простые и системные, комплексные программы, предназначенные для решения частных и комплексных задач, связанных с обеспечением ИБ. Примером комплексных решений служат DLP-системы и SIEM-системы: первые служат для предотвращения утечки, переформатирования информации и перенаправления информационных потоков, вторые – обеспечивают защиту от инцидентов в сфере информационной безопасности. Программные средства требовательны к мощности аппаратных устройств, и при установке необходимо предусмотреть дополнительные резервы.

К специфическим средствам информационной безопасности относятся различные криптографические алгоритмы, позволяющие шифровать информацию на диске и перенаправляемую по внешним каналам связи. Преобразование информации может происходить при помощи программных и аппаратных методов, работающих в корпоративных информационных системах.

Все средства, гарантирующие безопасность информации, должны использоваться в совокупности, после предварительной оценки ценности информации и сравнения ее со стоимостью ресурсов, затраченных на охрану. Поэтому предложения по использованию средств должны формулироваться уже на этапе разработки систем, а утверждение должно производиться на том уровне управления, который отвечает за утверждение бюджетов.

В целях обеспечения безопасности необходимо проводить мониторинг всех современных разработок, программных и аппаратных средств защиты, угроз и своевременно вносить изменения в собственные системы защиты от несанкционированного доступа. Только адекватность и оперативность реакции на угрозы поможет добиться высокого уровня конфиденциальности в работе компании.

Неформальные средства защиты группируются на нормативные, административные и морально-этические. На первом уровне защиты находятся нормативные средства, регламентирующие информационную безопасность в качестве процесса в деятельности организации.

Нормативные средства. та категория средств обеспечения информационной безопасности представлена законодательными актами и нормативно-распорядительными документами, которые действуют на уровне организации.

В мировой практике при разработке нормативных средств ориентируются на стандарты защиты ИБ, основный – ISO/IEC 27000. Стандарт создавали две организации:

- ISO – Международная комиссия по стандартизации, которая разрабатывает и утверждает большинство признанных на международном уровне методик сертификации качества процессов производства и управления;
- IEC – Международная энергетическая комиссия, которая внесла в стандарт свое понимание систем ИБ, средств и методов ее обеспечения

Актуальная версия ISO/IEC 27000-2016 предлагают готовые стандарты и опробованные методики, необходимые для внедрения ИБ. По мнению авторов методик, основа информационной безопасности заключается в системности и последовательной реализации всех этапов от разработки до пост-контроля.

Для получения сертификата, который подтверждает соответствие стандартам по обеспечению информационной безопасности, необходимо внедрить все рекомендуемые методики в полном объеме. Если нет необходимости получать сертификат, в качестве базы для разработки

собственных ИБ-систем допускается принять любую из более ранних версий стандарта, начиная с ISO/IEC 27000-2002, или российских ГОСТов, имеющих рекомендательный характер.

По итогам изучения стандарта разрабатываются два документа, которые касаются безопасности информации. Основной, но менее формальный – концепция ИБ предприятия, которая определяет меры и способы внедрения ИБ-системы для информационных систем организации. Второй документ, которые обязаны исполнять все сотрудники компании, – положение об информационной безопасности, утверждаемое на уровне совета директоров или исполнительного органа.

Кроме положения на уровне компании должны быть разработаны перечни сведений, составляющих коммерческую тайну, приложения к трудовым договорам, закрепляющий ответственность за разглашение конфиденциальных данных, иные стандарты и методики. Внутренние нормы и правила должны содержать механизмы реализации и меры ответственности. Чаще всего меры носят дисциплинарный характер, и нарушитель должен быть готов к тому, что за нарушением режима коммерческой тайны последуют существенные санкции вплоть до увольнения.

В рамках административной деятельности по защите ИБ для сотрудников служб безопасности открывается простор для творчества. Это и архитектурно-планировочные решения, позволяющие защитить переговорные комнаты и кабинеты руководства от прослушивания, и установление различных уровней доступа к информации. Важными организационными мерами станут сертификация деятельности компании по стандартам ISO/IEC 27000, сертификация отдельных аппаратно-программных комплексов, аттестация субъектов и объектов на соответствие необходимым требованиям безопасности, получений лицензий, необходимых для работы с защищенными массивами информации.

С точки зрения регламентации деятельности персонала важным станет оформление системы запросов на допуск к интернету, внешней электронной почте, другим ресурсам. Отдельным элементом станет получение электронной цифровой подписи для усиления безопасности финансовой и другой информации, которую передают государственным органам по каналам электронной почты.

Морально-этические меры определяют личное отношение человека к конфиденциальной информации или информации, ограниченной в обороте. Повышение уровня знаний сотрудников касательно влияния угроз на деятельность компании влияет на степень сознательности и ответственности сотрудников. Чтобы бороться с нарушениями режима информации, включая, например, передачу паролей, неосторожное обращение с носителями, распространение конфиденциальных данных в частных разговорах, требуется делать упор на личную сознательность сотрудника. Полезным будет установить показатели эффективности персонала, которые будут зависеть от отношения к корпоративной системе ИБ.

2 Лекция №2. Основные понятия и преобразования информации

Цель лекции: изучить основные понятия об информации, их формах и преобразованиях.

Содержание лекции: понятие об информации, источнике информации и информационном сообщении. Непрерывная и дискретная форма представления информации. Преобразование информации, квантование по уровню, дискретизация по времени. Теорема Котельникова.

2.1 Понятие информации

Понятие информации предполагает наличие материального носителя информации, источника информации, передатчика информации, приемника и канала связи между источником и приемником. Понятие информации используется во всех сферах: науке, технике, культуре, социологии и повседневной жизни. Конкретное толкование элементов, связанных с понятием информации, зависит от метода конкретной науки, цели исследования или просто от наших представлений.

Термин «информация» происходит от латинского informatio – разъяснение, изложение, осведомленность. Информация – общенаучное понятие, включающее обмен сведениями между людьми, человеком и автоматом, обмен сигналами в животном и растительном мире. Более узкое определение дается в технике, где это понятие включает в себя все сведения, являющиеся объектом хранения, передачи и преобразования.

С понятием информации связаны такие понятия, как сигнал, сообщение и данные.

Сигнал (от латинского signum – знак) представляет собой любой процесс, несущий информацию.

Сообщение – это информация, представленная в определенной форме и предназначенная для передачи.

Данные – это информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами, например, ЭВМ.

В простейшем бытовом понимании с термином «информация»обычно ассоциируются некоторые сведения, данные, знания и т.п. Информация передается в виде сообщений, определяющих форму и представление передаваемой информации. Примерами сообщений являются музыкальное произведение; телепередача; текст, распечатанный на принтере; данные, полученные в результате работы составленной программы и т.д. При этом предполагается, что имеются «источник информации» и «получатель информации».

Процессы, связанные с поиском, хранением, передачей, обработкой и использованием информации, называются информационными процессами.

Информационные процессы всегда играли важную роль в науке, технике и жизни общества. В ходе эволюции человечества просматривается устойчивая тенденция к автоматизации этих процессов, хотя их внутреннее содержание по существу осталось неизменным.

Основные информационные процессы:

1) Сбор информации – это деятельность субъекта, в ходе которой он получает сведения об интересующем его объекте. Сбор информации может производиться или человеком, или с помощью технических средств и систем – аппаратно. Например, пользователь может получить информацию о движении поездов или самолетов сам, изучив расписание, или же от другого человека непосредственно, либо через какие-то документы, составленные этим человеком, или с помощью технических средств (автоматической справки, телефона и т. д.). Задача сбора информации не может быть решена в отрыве от других задач, – в частности, задачи обмена информацией (передачи).

2) Обмен информацией – это процесс, в ходе которого источник информации ее передает, а получатель – принимает. Если в передаваемых сообщениях обнаружены ошибки, то организуется повторная передача этой информации. Обмен информации производится с помощью сигналов, являющихся ее материальным носителем. Источниками информации могут быть любые объекты реального мира, обладающие определенными свойствами и способностями. Если объект относится к неживой природе, то он вырабатывает сигналы, непосредственно отражающие его свойства. Если объектом-источником является человек, то вырабатываемые им сигналы могут не только непосредственно отражать его свойства, но и соответствовать тем знакам, которые человек вырабатывает с целью обмена информацией. Принятую информацию получатель может использовать неоднократно. С этой целью он должен зафиксировать ее на материальном носителе (магнитном, фото, кино и др.).

3) Накопление информации – это процесс формирования исходного, несистематизированного массива информации. Среди записанных сигналов могут быть такие, которые отражают ценную или часто используемую информацию.

4) Хранение информации – это процесс поддержания исходной информации в виде, обеспечивающем выдачу данных по запросам конечных пользователей в установленные сроки. Способ хранения информации зависит от ее носителя (книга- библиотека, картина- музей, фотография- альбом). ЭВМ предназначен для компактного хранения информации с возможностью быстрого доступа к ней. Информационная система - это хранилище информации, снабженное процедурами ввода, поиска и размещения и выдачи информации.

5) Обработка информации – это упорядоченный процесс ее преобразования в соответствии с алгоритмом решения задачи. После решения задачи обработки информации результат должен быть выдан конечным

пользователям в требуемом виде. Эта операция реализуется в ходе решения задачи выдачи информации. Выдача информации, как правило, производится с помощью внешних устройств ЭВМ в виде текстов, таблиц, графиков и пр.

Деятельность людей всегда связана с передачей информации.

В процессе передачи информация может теряться и искажаться: искажение звука в телефоне, атмосферные помехи в радио, и тд. Эти помехи, или, как их называют специалисты, шумы, искажают информацию. К счастью, существует наука, разрабатывающая способы защиты информации - криптология.

2.2 Лекция №3 Формы представления информации

Различают две формы представления информации – непрерывную и дискретную.

Чтобы сообщение было передано от источника к получателю, необходима некоторая материальная субстанция – носитель информации. Сообщение, передаваемое с помощью носителя, назовем сигналом. В общем случае сигнал – это изменяющийся во времени физический процесс. Такой процесс может содержать различные характеристики (например, при передаче электрических сигналов могут изменяться напряжение и сила тока). Та из характеристик, которая используется для представления сообщений, называется параметром сигнала.

В случае когда параметр сигнала принимает последовательное во времени конечное число значений (при этом все они могут быть пронумерованы), сигнал называется дискретным, а сообщение, передаваемое с помощью таких сигналов -дискретным сообщением. Информация, передаваемая источником, в этом случае также называется дискретной.

Если же источник вырабатывает непрерывное сообщение (соответственно параметр сигнала – непрерывная функция от времени), соответствующая информация называется непрерывной.

Пример дискретного сообщения – процесс чтения книги, информация в которой представлена текстом, т.е. дискретной последовательностью отдельных значков (букв). Примером непрерывного сообщения служит человеческая речь, передаваемая модулированной звуковой волной; параметром сигнала в этом случае является давление, создаваемое этой волной в точке нахождения приемника – человеческого уха.

2.2.1 Непрерывная форма представления информации.

Непрерывные (аналоговые) информационные сигналы могут принимать любые значения из всех возможных в рамках заданного интервала. В качестве примера непрерывных величин можно рассматривать такие физические величины как температура, расстояние, масса. Как правило, численные значения непрерывных величин, выражаются не целыми, а дробными числами (имеют знаки после запятой).

Всегда можно определить промежуточное значение между двумя соседними значениями аналоговых (непрерывных) величин, то есть такие величины могут уточняться до бесконечности.

Аналоговые процессы основаны на непрерывных сигналах. Например, процессы, происходящие в природе, имеют аналоговое происхождение: звучание пения птиц, изменение температуры окружающего воздуха. Некоторые технические процессы имеют также аналоговую природу, например аналоговое телевидение и телефония.

В аналоговых системах, изменение величин сигналов происходит плавно, не наблюдается каких-либо резких скачков.

2.2.2 Дискретная форма представления информации.

В дискретной форме представления информации величины могут принимать лишь отдельные, неделимые значения и не могут принимать значения промежуточные между ними. Дискретные величины выражаются целыми числами, используемыми при пересчете предметов. Примерами дискретных величин могут быть число букв в алфавите, число учеников в классе.

Представление информационных сообщений с помощью знаковых систем также носит дискретный характер. Это может быть информация, записанная с помощью алфавита естественного языка, например русского.

В цифровых (дискретных) системах процессы основаны на дискретных сигналах. Примерами цифровых систем являются системы цифровых телекоммуникаций, цифровые телевидение и телефония.

2.2.3 Изменение формы представления информации.

При выполнении повседневных дел человек часто сталкивается с задачей преобразования информации из одной формы представления в другую. Например, напечатанный текст представлен в дискретной форме, а при произнесении его вслух он представляется в аналоговой форме в виде звукового сигнала.

Дискретная форма сигнала гораздо проще для обработки ее на ЭВМ, передачи по цифровым коммуникационным каналам. Поэтому решение задачи трансформации непрерывного сигнала в дискретный вид имеет большое значение.

Процесс перехода аналогового сигнала в дискретную форму называется дискретизацией сигнала.

Перевод информации в дискретную форму заключается в кодировании ее символами естественного или формального языка. Каждый язык — разговорный или язык программирования имеет свой алфавит.

Алфавитом называется законченный набор знаков, применяемых для кодирования информации. Число символов, составляющих алфавит, называется мощностью алфавита.

Информационный сигнал в ЭВМ представляется в двоичном формате, то есть, закодирован с помощью алфавита, мощность которого равна 2. Двоичный алфавит состоит только из двух знаков: единицы и нуля.

3 Лекция №4. Меры дискретной информации

3.1 Знаки, сигналы и сообщения

Сообщение – в теории коммуникации – предназначенные для передачи высказывание, текст, изображение, физический предмет или поступок. Сообщения состоят из словесных или невербальных сигналов. Одиночный сигнал не может содержать много информации, поэтому для передачи информации используется ряд следующих друг за другом сигналов. Последовательность сигналов и называется сообщением.

Таким образом, от источника к приемнику информация передается в виде сообщений. Можно сказать, что сообщение выступает в качестве материальной оболочки для представления информации при передаче. Следовательно, сообщение служит переносчиком информации, а информация является содержанием сообщения.

Соответствие между сообщением и содержащейся в нем информацией называется правилом интерпретации сообщения. Это соответствие может быть однозначным и неоднозначным. В первом случае сообщение имеет лишь одно правило интерпретации. Во втором случае соответствие между сообщением и информацией возможно в двух вариантах: 1) одна и та же информация может передаваться различными сообщениями (в частности, новости могут быть получены по радио, из газеты, по телефону и пр.); 2) одно и то же сообщение может содержать различную информацию для разных приемников (скажем, падение курса акций на бирже для одних катастрофа, а для других – возможность обогащения).

Поскольку последовательность сигналов есть сообщение, качество прерывности-непрерывности сигналов переносится и на сообщение. Существуют такие понятия, как непрерывное (аналоговое), дискретное, квантованное и цифровое сообщение. Заметим, что информация данным качеством не обладает, так как информация – категория нематериальная и не может обладать свойством дискретности или непрерывности. Хотя одна и та же информация может быть представлена посредством различных сообщений, в том числе и отличающихся характером сигналов. В информатике иногда используются словосочетания "непрерывная информация" и "дискретная информация". Они являются результатом сокращения таких терминов, как информация, представленная посредством непрерывных сигналов, и информация, представленная посредством дискретных сигналов. Поэтому, когда речь идет о видах информации, правильнее говорить о формах ее представления в сообщении или о видах сообщений.

При формировании сообщения, наряду с сигналом, используются и такие понятия, как знак, буква и символ. Ниже показаны отличия между ними.

Знак – это элемент некоторого конечного множества отличных друг от друга сущностей. Природа знака может любой – жест, рисунок, буква, сигнал светофора, определенный звук и т.д. и определяется как носителем сообщения, так и формой представления информации в сообщении. Все

множество знаков, используемых для представления дискретной информации, называется **набором знаков.** Набор есть дискретное множество знаков.

Набор знаков, в котором установлен порядок их следования, называется алфавитом. **Алфавит** – это упорядоченная совокупность знаков. Порядок следования знаков в алфавите называется **лексикографическим** и предоставляет возможность устанавливать отношения "**больше – меньше":** для двух знаков Г < Д, если порядковый номер у Г в алфавите меньше, чем у Д.

Знаки, используемые для обозначения фонем в устной речи, называются **буквами,** а их совокупность – алфавитом языка.

Сами по себе знак или буква не несут никакого смыслового содержания. Однако такое содержание им может быть приписано, в этом случае знак будет называться **символом.**

Например, электрическое напряжение в физике принято обозначать буквой **U,** и, следовательно, **U** в формулах является символом физической величины "электрическое напряжение". Другим примером символов могут служить пиктограммы, обозначающие в компьютерных программах объекты или действия.

Таким образом, понятия "знак", "буква" и "символ" нельзя считать тождественными, хотя весьма часто различия между ними не проводят; поэтому в информатике существуют понятия "символьная переменная", "кодировка символов алфавита", "символьная информация", во всех приведенных примерах вместо термина "символьный" более корректным было бы использовать "знаковый" или "буквенный".

Представляется важным еще раз подчеркнуть, что понятия знака и алфавита можно отнести только к **дискретным сообщениям.**

Сигналы

Термин "информация" па бытовом уровне и в многочисленных научных дисциплинах ассоциируется с такими понятиями, как сведения, знания, данные, известие, сообщение, управление и др. Если допустить, что информация – категория нематериальная, то для ее существования и распространения в нашем материальном мире она должна быть обязательно связана с какой-либо материальной основой (носителем) – без нее информация не может проявляться, передаваться и сохраняться.

Материальным носителем называется материальный объект или среда, которые служат для представления или передачи информации. Материальным носителем информации может быть бумага, пергамент, шелк, камень, лазерный диск, а также воздух, вода, электромагнитное поле, луч света и пр. Следует отметить, что хранение информации связано с характеристикой носителя, которая не меняется с течением времени; а передача информации, напротив, – с характеристикой, которая изменяется с течением времени. Другими словами, хранение информации связано с фиксацией состояния носителя, а передача – с процессом, который протекает в носителе. Состояния и процессы могут иметь физическую, химическую, биологическую или иную основу;

главное, что они материальны. Ниже рассмотрена разница между такими понятиями, как сигнал, сообщение, знак, буква, символ, данные и знания.

Сигнал (от лат. signum – знак) – физический процесс или явление, несущее сообщение о каком-то событии, состоянии объекта, либо передающий команды управления. Таким образом, изменение характеристики носителя, которое используется для представления информации, называется сигналом, а значение этой характеристики, отнесенное к некоторой шкале измерений, называется параметром сигнала.

Приведем пример. Процессы для передачи информации – волновые (звук, радио, свет), параметры сигнала – частота, амплитуда и фаза волны (например, высота, громкость и фаза звука).

Различают аналоговые, дискретные, квантованные и цифровые сигналы, которые, в свою очередь, могут быть синхронными и асинхронными.

Аналоговый сигнал – это сигнал, величина которого непрерывно изменяется во времени (рис. 1). Аналоговый сигнал обеспечивает передачу данных путем непрерывного изменения во времени амплитуды, частоты либо фазы. Аналоговые сигналы описываются непрерывными функциями времени, поэтому аналоговый сигнал иногда называют непрерывным сигналом. Аналоговые сигналы описываются некоторой математической функцией времени.

Гармоническое колебание – явление периодического изменения какой-либо величины, при котором зависимость от аргумента имеет характер функции синуса или косинуса. Например, гармонически колеблется величина, изменяющаяся во времени следующим образом:

$x(t) = A\cos(\omega t + \varphi)$,

где x – значение изменяющейся величины; t – время; A – амплитуда колебаний; ω – циклическая частота колебаний; $(\omega t + \varphi)$ – полная фаза колебаний; φ – начальная фаза колебаний.

Гармонический сигнал – это гармонические колебания, со временем распространяющиеся в пространстве, которые несут в себе информацию или какие-то данные и описываются уравнением

$$S(t) = A\cos(\omega t + \varphi_0),$$

где A – амплитуда сигнала; $\varphi = (\omega t + \varphi_0)$ – фаза гармонического сигнала; t – время; ω – циклическая частота сигнала; φ0 – начальная фаза сигнала.

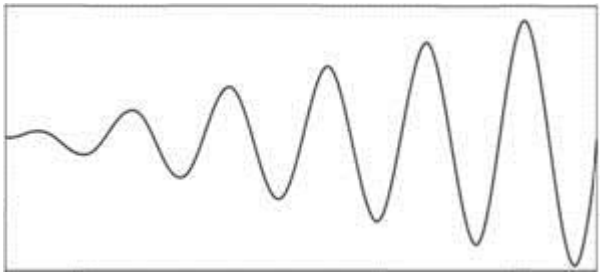


Рисунок 1 - Непрерывный гармонический сигнал

К особым свойствам непрерывного сигнала относят отсутствие избыточности. Из непрерывности пространства значений сигнала следует, что любая помеха, внесенная в сигнал, неотличима от самого сигнала и, следовательно, исходная амплитуда не может быть восстановлена. Хотя в принципе возможна фильтрация помех, если известна дополнительная информация о самом сигнале. Аналоговые сигналы используются для представления непрерывно изменяющихся каких-либо физических величин.

Так, аналоговый электрический сигнал, снимаемый с термопары, несет информацию об изменении температуры. Сигнал с микрофона – о мгновенных изменениях давления в звуковой волне.

Дискретный сигнал. Дискретизация – это процесс преобразования аналогового сигнала в дискретный сигнал (рис. 2). Процесс, обратный этому, называется восстановлением. Непрерывный аналоговый сигнал заменяется здесь последовательностью коротких импульсов – отсчетов, величина которых равна значению сигнала в текущий момент времени S(ti). Дискретизация аналогового сигнала состоит в том, что сигнал представляется в виде последовательности значений, взятых в дискретные моменты времени $t_i = i\Delta t$, Эти значения сигнала называются отсчетами S_i, а $\Delta t = \text{const}$ – интервалом дискретизации.

Возможность точного воспроизведения такого представления зависит от интервала времени Δt между отсчетами Si Согласно теореме Котельникова сигнал может быть точно воспроизведен, если

$$\Delta t \le \frac{1}{2F_{\max}},$$

где Fmax – наибольшая частота спектра сигнала.

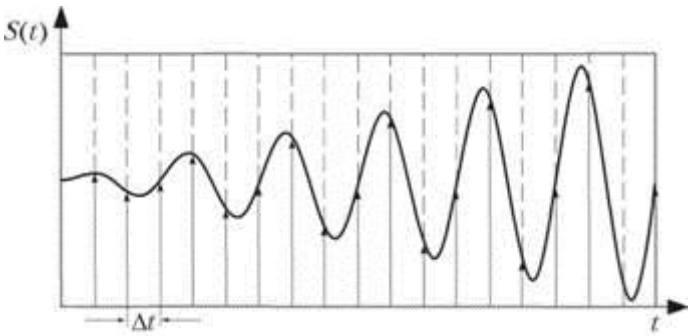


Рисунок 2 - Дискретный сигнал

Квантованный сигнал. При квантовании вся область значений сигнала, $S(t)$ разбивается на уровни, количество которых должно быть представлено в числах заданной разрядности (рис. 3). Расстояния между этими уровнями называется шагом квантования Δ. Число этих уровней равно N. Каждому уровню присваивается некоторое число $m \in 0$; N – 1. Отсчеты S_i фиксируются, когда значение сигнала 5(í) совпадает с соответствующим уровнем квантования $m_i \Delta_i = S_i$. Каждый уровень квантования mi кодируется двоичным числом с п разрядами. Аналоговый сигнал $S(t)$ заменяется набором пар

чисел $(\Delta t_i; m_i \Delta)$. Причем отметим, что $\Delta t_i \neq \text{const}$. Число уровней квантования N и число разрядов п двоичных чисел, кодирующих эти уровни, связаны соотношением $n \geq \log_2 N$

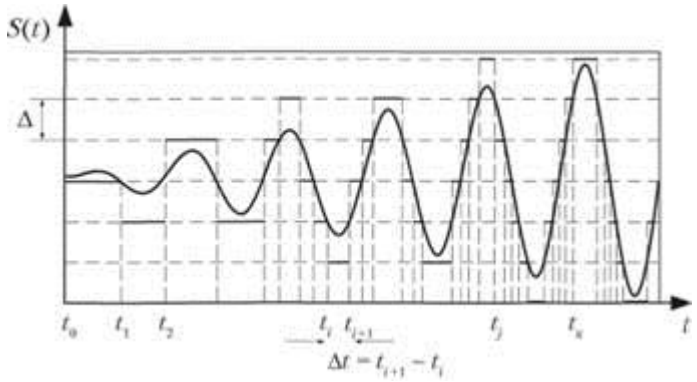


Рисунок 3 - Квантованный сигнал

Цифровой сигнал. Чтобы представить аналоговый сигнал последовательностью чисел конечной разрядности, его следует сначала превратить в дискретный сигнал (Δt = const), а затем подвергнуть квантованию (Δ = const). В результате сигнал будет представлен так, что на каждом заданном промежутке времени известно приближенное (квантованное) значение сигнала, которое можно записать целым числом (рис. 4). Если записать эти целые числа в двоичной системе, получится последовательность нулей и единиц, которая и будет являться цифровым сигналом.

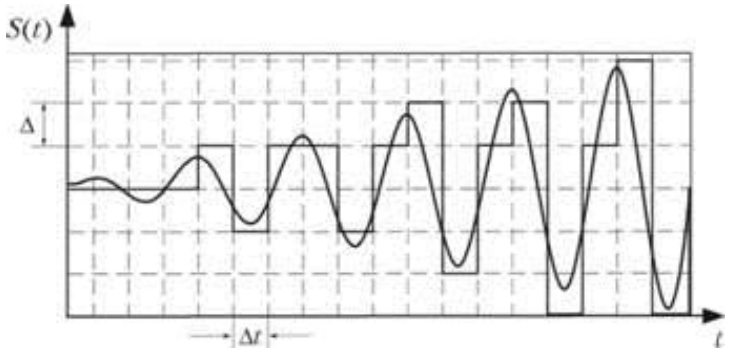


Рисунок 4 - Цифровой сигнал

Синхронный сигнал – это сигнал, значения которого могут изменяться только в моменты, определяемые тактами. Например, ежедневный выстрел из пушки в полдень.

Асинхронный сигнал – это сигнал, изменение значения которого может происходить в любое время.

3.1.1 Аспекты информационных мер

В структурном аспекте рассматривается строение массивов информации и их измерение простым подсчетом информационных элементов или комбинаторным методом. Структурный подход применяется для оценки возможностей информационных систем вне зависимости от условий их применения.

Информация всегда представляется в виде сообщения. Элементарная единица сообщений – символ. Символы, собранные в группы, - слова. Сообщение, оформленное в виде слов или отдельных символов, всегда передается в материально-энергетической форме (электрический, световой, звуковой сиг. и т. д.).

Различают информацию непрерывную и дискретную. При использовании структурных мер информации учитывается только дискретное строение сообщения, количество содержащихся в нем информационных элементов, связей между ними.

При структурном подходе различаются:

– геометрическая;
– комбинаторная;
– аддитивная меры информации.

Геометрическая мера предполагает измерение параметра геометрической модели информационного сообщения (длины, площади, объема и т. п.) в дискретных единицах.

Максимально возможное количество информации в заданных структурах определяет информационную емкость модели (системы), которая определяется как сумма дискретных значений по всем измерениям (координатам).

В комбинаторной мере количество информации определяется как число комбинаций элементов (символов). Возможное количество информации совпадает с числом возможных сочетаний, перестановок и размещений элементов. Комбинирование символов в словах, состоящих только из 0 и 1, меняет значения слов. Рассмотрим две пары слов 1001 10 и 001101, 011101 и 111010. В них проведена перестановка крайних разрядов (изменено местоположение знакового разряда в числе – перенесен слева направо).

Аддитивная мера (мера Хартли), в соответствии с которой количество информации измеряется в двоичных единицах – битах, - наиболее распространена. Вводятся понятия глубины q и длины п числа.

Глубина q числа – количество символов (элементов), принятых для представления информации, В каждый момент времени реализуется только один какой-либо символ. (По позже понятие глубины числа будет трансформировано в понятие основания системы счисления).

Длина п числа – количество позиций, необходимых и достаточных для представления чисел заданной величины.

При заданных глубине и длине числа количество чисел, которое можно представить, $N = q^n$. Величина N не удобна для оценки информационной емкости. Введем логарифмическую меру, позволяющую, вычислять количество информации, – бит.

При статистическом подходе используется понятие энтропии как меры неопределенности, учитывающей вероятность появления и информативность того или иного сообщения. Статистический подход учитывает конкретные условия применения информационных систем.

В статистической теории информации вводится более общая мера количества информации, в соответствии с которой рассматривается не само событие, а информация о нем. Этот вопрос глубоко проработан К. Шенноном в работе «Избранные труды по теории информации». Если появляется сообщение о часто встречающемся событии, вероятность появления которого близка к единице, то такое сообщение для получателя малоинформативно. Столь же малоинформативны сообщения о событиях, вероятность появления которых близка к нулю.

События можно рассматривать как возможные исходы некоторого опыта, причем все исходы этого опыта составляют ансамбль, или полную группу событий. К. Шеннон ввел понятие неопределенности ситуации, возникающей в процессе опыта, назвав ее энтропией (обозначают обычно буквой *И)*. *Энтропия* ансамбля есть количественная мера его неопределенности и, следовательно, информативности, количественно выражаемая как средняя функция множества вероятностей каждого из возможных исходов опыта.

Семантический подход позволяет выделить полезность или ценность информационного сообщения.

Вычислительные машины обрабатывают и преобразуют информацию разного содержания – от числовых данных до сочинения музыки и стихов. Вся эта информация изображается соответствующими символами. Оценка содержания разнохарактерной информации – весьма сложная проблема.

Среди семантических мер наиболее распространены:

- содержательность;
- логическое количество;
- целесообразность;
- существенность информации.

3.1.2 Аддитивная мера информации по Хартли Классификация настольных приложений.

В 1928 году Хартли предложил меру для *измерения информации дискретных сообщений*.

Аддитивная мера (мера Хартли) использует понятия глубины А и длины n числа.

Глубина числа – количество символов (элементов), принятых для представления информации. В каждый момент времени реализуется только один какой-либо символ.

Длина n числа – количество позиций, необходимых и достаточных для представления чисел заданной величины.

Эти понятия могут быть распространены и на вариант нечислового сообщения. В этом случае глубина числа тождественна размеру алфавита, а длина числа – разрядности слова при передаче символьного сообщения.

Если сообщение – число, понятие глубины числа будет трансформировано в понятие основания системы счисления. При заданных глубине и длине числа количество чисел, которое можно представить, $\boldsymbol{N = A^n}$. Очевидно, что $\boldsymbol{N}$ однозначно характеризует степень исходной неопределенности. Исходная неопределенность по Хартли определяется:

$$H_1 = \log_a N. \qquad (3.1)$$

Неопределенность после получения сообщения, остаточная неопределенность:

$$H_2 = \log_a N^*, \qquad (3.2)$$

где N^* – число возможных значений принятого слова после получения сообщения.

Основание логарифма в (3.1) определяет только единицы измерения неопределенности. При $a=2$ это двоичная единица информации, называемая бит. При $a = 10$ десятичная (*дит*), при $a = e$ натуральная (*нат*). Далее мы будем всегда пользоваться двоичной единицей.

N* равно единице, если после получения информации нет неопределенности, т.е. получатель гарантировано получил то сообщение, которое было передано. Если получателю приходится после приема информации выбирать сообщения из некоторого множества, а это происходит тогда, когда в канале связи за счет влияния помех возникают искажения переданного сигнала, то характеризует число возможных сообщений при выборе. Таким образом, если передается символ некоторого алфавита, N^* определяет возможную неоднозначность приема символа за счет искажений в канале связи. В случае измерительного опыта, число N^* – характеризует число возможных значений величины после измерения и определяет погрешность измерения.

Очевидно, что должно быть $N^* < N$, а $N^* = 1$ только в идеальном случае передачи сообщения без потери информации или, что то же самое, измерения некоторой физической величины без ошибок. Количество информации по Хартли оценивается как:

$$I=H_1 – H_2 = \log_a N - \log a\ N^*\ n = \log_a N/ N^* . \qquad (3.3)$$

Логарифмическая мера, позволяющая, вычислять количество информации, содержащейся в сообщении, переданном числом длиной *n* и глубиной *A*:

$$I(q) = \log_2 N = n \log_2 A, \textit{ бит}. \quad (3.4)$$

Следовательно, *1 бит* информации соответствует одному элементарному событию, которое может произойти или не произойти. Такая мера количества информации удобна тем, что она обеспечивает возможность оперировать мерой как числом.

Логарифмическая мера для неопределенности и информации выбрана не случайно. Она оказывается удобной при описании сложных опытов. Допустим, что задача состоит в одновременном приеме информации от двух источников, не зависящих друг от друга. При этом N_1 и n_1 – число возможных сообщений до и после приема информации от первого источника, а – N_2 и n_2 от второго. Пусть H_{11} и H_{12} – исходная неопределенность знания первого и второго сообщения, соответственно, первого и второго источника. Естественно потребовать, чтобы общая неопределенность знания о двух сообщениях определялась суммой неопределенностей каждого, т.е. мера должна обладать свойством аддитивности:

$$H = H_{11} + H_{12}. \quad (3.5)$$

Число возможных сочетаний двух независимых величин из множеств $N_1 N_2 N = N_1 N_2$.

Тогда исходная неопределенность $H = H_{11} + H_{12}$, , аналогично остаточная неопределенность $H = H_{21} + H_{22}$.

При наличии нескольких источников информации общее количество информации:

$$I(q_1, q_2, ..., q_n) = I(q_1) + I(q_2) + ... + I(q_k), \quad (3.6)$$

где $I(q_k)$ — количество информации от источника *k*.

Логарифмическая мера информации позволяет измерять количество информации и широко используется на практике. Однако всегда надо учитывать, что все сообщения в этой мере полагаются равновероятными и независимыми.

3.2 Лекция №5 Оценка количества информации по Шеннону

Существует два подхода к анализу явлений статический и динамический. Статический подход встречается чаще, поскольку проведение его проще и не требует использования сложных

математических методик. **Первая формула вероятностного подхода к измерению количества информации была предложена в 1928 г. Ральфом Хартли, вторая формула - в 1948 году - Клодом Шенноном.**

При **статическом вероятностном подходе** получение конкретного количества информации рассматривается как результат определенного выбора среди возможных сообщений. Получатель информации может заранее знать или угадать ее часть.

При статическом вероятностном подходе получение конкретного количества информации рассматривается как **результат определенного выбора среди возможных сообщений.** Получатель информации может заранее знать или угадать ее часть. Когда приходит сообщение о часто происходящих событиях, вероятность появления которых P стремится к единице, то такое сообщение малоинформативно. Столь же малоинформативны в среднем сообщения о событиях, вероятности которых стремятся к нулю, т.е. о почти невозможных событиях, поскольку сообщения о таких событиях поступают чрезвычайно редко.

События можно рассматривать как возможные исходы некоторого опыта. Все исходы составляют полную группу событий, или ансамбль.

Ансамбль характеризуется тем, что сумма вероятностей всех сообщений в нем равна единице, то есть

$$P_1 + P_2 + \ldots + P_k = \sum_{i=1}^{k} P_i = 1 .$$

Рассмотрим сложные сообщения, составляемые из n элементов, каждый из которых является независимым и выбирается из алфавита, содержащего m букв, с вероятностями выбора элементов $P_1, P_2, \ldots, P_m$ соответственно. Предположим, что в некоторое сообщение вошло n_1 элементов X_1 алфавита, n_2 элементов X_2 и т.д. Такое сообщение характеризуется таблицей (табл. 1.1).

Таблица 1.1

Тип элемента				..		..	
Число элементов	n_1	n_2	n_3	...	n_i	...	n_m
Вероятности выбора элементов	P_1	P_2	P_3	...	P_i	...	P_m

Вероятность того, что в сообщение войдут n_i элементов X_i равняется $P_i^{n_i}$, а вероятность образования сообщения из $n_1, n_2, n_3, ..., n_i, ..., n_m$ элементов будет равна

$$P = P_1^{n_1} P_2^{n_2} \dots P_i^{n_i} \dots P_m^{n_m} \quad . (1.1)$$

При большой длине n источником будут формироваться типичные сообщения, в которых относительная частота появления отдельных элементов X_i стремится к вероятности появления этих элементов, то есть

$$\frac{n_i}{n} = \frac{n_i}{\sum_{i=1}^{m} n_i} = P_i \quad , \qquad (1.2)$$

а вероятности появления типичных сообщений P будут одинаковы и могут быть найдены из (1.1), (1.2):

$$P = \prod_{i=1}^{m} P_i^{n_i P_i} \quad . \qquad (1.3)$$

Определим число типичных сообщений:

$$L = \frac{1}{P} = \frac{1}{\prod_{i=1}^{m} P_i^{n_i P_i}} \quad , \qquad (1.4)$$

так как суммарная вероятность всех типичных сообщений стремится к единице при увеличении длины сообщений.

Хотя число возможных сообщений m^n, источник практически будет вырабатывать только L типичных сообщений, а вероятность появления остальных сообщений стремится к нулю.

Найдем количество информации I, содержащейся в одном сообщении:

$$I = \log L = -\log \prod_{i=1}^{m} P_i^{n_i P_i} = -\sum_{i=1}^{m} n_i P_i \log P_i = -n \sum_{i=1}^{m} P_i \log P_i \quad . (1.5)$$

Данное выражение (формула Шеннона) дает более полное представление об источнике информации, чем аддитивная мера (мера Хартли). Поясним это на следующем примере. Если мы подбрасываем монету, то получим сообщение из двух возможных состояний (орел или решка), то есть, алфавит сообщений из двух букв. Если подбрасываем кубик, одна грань

которого голубая, а остальные грани окрашены в розовый цвет, то здесь также имеем алфавит из двух букв (голубой или розовый). Чтобы записать полученный текст (сообщение), в обоих случаях достаточно одной двоичной цифры на букву (n=1, m=2).

По Хартли здесь в обоих случаях

$$I = n \cdot \log_2 m = 1 \cdot \log_2 2 = 1.$$

Но мы знаем, что в первом случае вероятность каждого исхода опыта равна 0,5 ($P_1 = P_2$=0,5). А во втором случае $P_1 = \frac{1}{6}$ и $P_2 = \frac{5}{6}$ соответственно. Мера Хартли не учитывает этого.

При равновероятности символов (частный случай) формула Шеннона вырождается в формулу Хартли:

$$I = -n \sum_{i=1}^{m} P_i \log_2 P_i = -n \sum_{i=1}^{m} \frac{1}{m} \log_2 \frac{1}{m} = n \cdot \log_2 m.$$

Для случая с монетой:

$$I = -1 \left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) = 1.$$

Для случая с кубиком:

$$I = -1 \left(\frac{1}{6} \log_2 \frac{1}{6} + \frac{5}{6} \log_2 \frac{5}{6} \right) = 0{,}195.$$

Количество информации, приходящейся на один элемент сообщения, называется **удельной информативностью**или **энтропией**.

$$H = -\sum_{i=1}^{m} P_i \log P_i = \sum_{i=1}^{m} P_i \log \frac{1}{P_i}. \quad (1.6)$$

Количество информации и энтропия являются логарифмическими мерами и измеряются в одних и тех же единицах. Основание логарифма определяет единицу измерения количества информации и энтропии. Двоичная единица соответствует основанию логарифма, равному двум, и называется битом. Один бит - это количество информации в сообщении в одном из двух равновероятностных исходов некоторого опыта.

. Аналогичными единицами пользуются и при оценке количества информации с помощью меры Хартли.

Из формулы Шеннона следует, что количество информации, содержащейся в сообщении, зависит от числа элементов сообщения n, алфавита m и вероятностей выбора элементов P_i.

Зависимость I от n является *линейной*.

3.2.1 Свойства энтропии

Любое сообщение в теории информации описывает состояние какой-то физической системы, которая случайным образом может оказаться в том или ином состоянии. Поэтому в качестве объекта, о котором передается информация, в теории информации рассматривается физическая система, которой присуща степень неопределенности. **Степень неопределенности физической системы определяется числом ее возможных состояний и вероятностями состояний.** В качестве меры априорной неопределенности системы в теории информации применяется характеристика, называемая энтропией.

Понятие энтропии как меры необратимого рассеяния энергии впервые было введено в термодинамике **в 1865 году немецким физиком Р. Клаузиусом, который показал, что каждому состоянию термодинамической системы соответствует определенное значение энтропии.** В статистической физике энтропия S, согласно концепции Л. Больцмана (1872 год), связывается с термодинамической вероятностью **W макроскопического состояния соотношением**

S = klnW,

где k – постоянная Больцмана. В свою очередь, в теории информации энтропия по К. Шеннону определяется **как мера неопределенности опыта с разными исходами.** Последние две трактовки энтропии имеют весьма глубокую связь: на базе информационной энтропии, например, выводятся канонические гиббсовские распределения статистической физики.

Раскроем на простых примерах случайного выбора в конечных схемах. Случайные события, как известно, характеризуются тем, что у нас нет полной уверенности в их наступлении, то есть мы имеем некую неопределенность при изучении опытов, связанных с такими событиями. **Понятно, что степень этой неопределенности в различных случаях может быть разной.**

Для практики необходимо уметь численно выражать степень неопределенности различных опытов, чтобы сравнивать их посредством такой характеристики между собой. В качестве меры неопределенности случайного объекта (системы) с конечным множеством возможных состояний A1 , A2 , …, An с соответствующими вероятностями p1 , p2 , …, pn или дискретной случайной величины X, принимающей значения X1 , X2 , … …, Xn с теми же вероятностями, Клод Шеннон предложил использовать функционал [1]

$$S(x) = -\sum_{i=1}^{n} p(i)\log_2 p(i) = \sum_{i=1}^{n} p(i)\log_2 \frac{1}{p(i)}$$

Эта величина также называется *средней энтропией* сообщения.

Энтропия — это то, как много информации вам не известно о системе

Например, если вы спросите меня, где я живу, и я отвечу: в Казахстане, то моя энтропия для вас будет высока, всё-таки Казахстан большая страна. Если же я назову вам свой почтовый индекс: 520000 то моя энтропия для вас понизится, поскольку вы получите больше информации.

Почтовый индекс содержит шесть цифр, то есть я дал вам шесть символов информации. Энтропия вашего знания обо мне понизилась приблизительно на 6 символов. (На самом деле, не совсем, потому что некоторые индексы отвечают большему количеству адресов, а некоторые — меньшему, но мы этим пренебрежём).**Энтропия в формуле Шеннона является средней характеристикой – математическим ожиданием распределения случайной величины** $i_1, i_2, \ldots i_n$**.**
Например, в последовательности букв, составляющих какое-либо предложение на русском языке, разные буквы появляются с разной частотой, поэтому **неопределённость появления для некоторых букв меньше, чем для других.**

1. Энтропия является величиной вещественной, ограниченной и неотрицательной, то есть H>0. Это свойство следует из выражения (1.6).

2. Энтропия минимальна и равна нулю, если сообщение известно заранее, то есть если P_i=1, а

$$P_1 = P_2 = \ldots = P_{i-1} = P_{i+1} = \ldots = P_n = 0.$$

3. Энтропия максимальна, если все состояния элементов сообщений равновероятны.

H=$H_{\max}$, если $P_1 = P_2 = \ldots = P_i = \ldots = P_m = P = \frac{1}{m}$. (1.7)

Величину максимальной энтропии найдем при использовании (1.6) и (1.7):

$$H_{\max} = -\sum_{i=1}^{m} \frac{1}{m} \log \frac{1}{m} = \log m.$$

4. Энтропия бинарных (двоичных) сообщений может изменяться от нуля до единицы (двоичный алфавит, следовательно, т=2.)

$$H = -\sum_{i=1}^{2} P_i \log P_i = -P_1 \log P_1 - P_2 \log P_2.$$

Используя условие $\sum_{i=1}^{m} P_i = 1$ и обозначив P_1=P, получим $P_2 = 1 - P_1 = 1 - P$, а энтропия определится выражением

H= -P logP - (1 - P)log(1 - P) .

Энтропия достигает максимума, равного единице, при $P_1 = P_2$ =0,5.

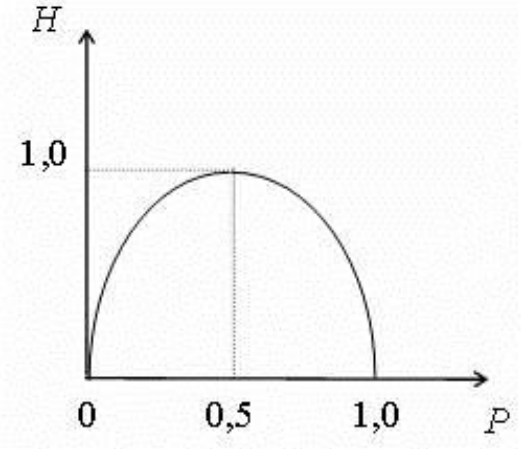


Рис. 1.3 - График функции H= -P logP - (1 - P)log(1 - P)

3.2.2 Семантические меры информации

Семантические меры информации оценивают смысл, содержание информации, ее целесообразность и существенность.

Целесообразность, полезность информации для решения какой-то задачи можно оценить по эффекту, который оказывает полученная информация на решение задачи. Если вероятность достижения цели увеличивается, то информацию следует считать полезной.

Рассмотрим два подхода к построению семантической меры информации, определяющей ее количество с точки зрения смысла или содержания.

Один из подходов построения такой меры информации базируется на математической (символической) логике и состоит в использовании простого формализованного языка, включающего конечное число имен единичных предметов (индивидов) и логических сказуемых (предикатов). В этом языке выделяются определенные предложения, которые называются описаниями состояния. Каждому такому описанию состояния сопоставляется некоторое положительное число, которое называется мерой. Мера рассматривается как логическая вероятность m(i), определяющая, в какой степени подтверждается истинность выдвинутой гипотезы i. Для этого устанавливается связь содержания h гипотезы с начальным достоверным знанием e (эмпирическими данными). Значение m(i) максимально и равно единице, если гипотеза полностью вытекает из эмпирических данных, и нулю

в противном случае. По логической вероятности, или знаниям h, содержащимся в гипотезе i относительно достоверного знания e, можно судить о величине семантической информации **I**c (h/e). В качестве семантической меры принимают количество информации, которое определяется следующим выражением [24]:

$$I_c = \log_2 [1/m(i)]. \quad (1.7)$$

Из выражения (1.7) следует, что, количество информации / = 0, если логическая вероятность m(i) = 1; с уменьшением логической вероятности m(i), или степени подтверждения гипотезы, количество семантической информации увеличивается.

Недостаток меры (1.7) состоит в том, что достоверные знания не несут информации.

Рассмотрим поход, лишенный указанного недостатка. Согласно ему информацией обладают любые сообщения, которые изменяют запас знаний, сведений, которыми располагает приемник (пользователь) информации. **Этот запас знаний, сведений называется тезаурусом**. Его можно представить в виде словаря пользователя, в котором отражены все его знания, а количество семантической информации, содержащееся в сообщении, – степенью пополнения словаря. После приема сообщения возможны два исхода, когда пользователь не включает принятую информацию в свой словарь:

• пользователь не понимает и не воспринимает поступающую информацию ($S_П = 0$);

• пользователь все знает ($S_П \to \infty$).

В тех случаях, когда поступающая информация понятна пользователю и несет ему ранее неизвестные сведения, он включает ее (полностью или частично) в свой словарь. Максимальное количество семантической информации I_c соответствуют согласованию ее смыслового содержания **S** со словарем (тезаурусом) $S_П$ пользователя, т.е. $I_c = S = S_{П\,опт}$ [1].

Следовательно, количество семантической информации в сообщении, количество новых знаний, получаемых пользователем, является величиной относительной. Одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным (семантический шум) для некомпетентного. Таким образом, тезаурусная мера связывает семантические свойства информации с индивидуальными способностями пользователя принимать поступившее сообщение. Смысловое содержание S принятого сообщения может быть понятно одному пользователю и непонятно другому.

Для количественной оценки семантической информации также используют коэффициент содержательности С, представляющий собой отношение количества семантической информации к ее объему

$$C = I_c / V_д.$$

Так, школьник, читая учебник, изменяет (увеличивает) свой запас знаний. Изменение запаса знаний можно определенным способом измерить и определить количество семантической информации. Однако студент, прочитав тот же учебник, не приобретет знаний, так как сведения, содержащиеся в учебнике средней школы, ему известны. В этом случае он не получает семантической информации, так как не изменяет запаса своих знаний.

Основная литература

1. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. - Москва: Наука, 2015. - 552 с.
2. Бабаш, А. В. Информационная безопасность (+ CD-ROM) / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. - М.: КноРус, 2013. - 136 с.
3. Васильков, А. В. Безопасность и управление доступом в информационных системах / А.В. Васильков, И.А. Васильков. - М.: Форум, 2015. - 368 с.
4. Гафнер, В. В. Информационная безопасность / В.В. Гафнер. - М.: Феникс, 2014. - 336 с.
5. Информационная безопасность открытых систем. В 2 томах. Том 1. Угрозы, уязвимости, атаки и подходы к защите / С.В. Запечников и др. - Москва: Машиностроение, 2016. - 536 с.
6. Информационная безопасность открытых систем. В 2 томах. Том 2. Средства защиты в сетях / С.В. Запечников и др. - Москва: СПб. [и др.] : Питер, 2014. - 560 с.
7. Мельников, В. П. Информационная безопасность / В.П. Мельников, С.А. Клейменов, А.М. Петраков. - М.: Academia, 2017. - 336 с.
8. Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников. - М.: Академия (Academia), 2016. - 282 с.
9. Партыка, Т. Л. Информационная безопасность / Т.Л. Партыка, И.И. Попов. - М.: Форум, Инфра-М, 2016. - 368 с.
10. Степанов, Е.А. Информационная безопасность и защита информации. Учебное пособие / Е.А. Степанов, И.К. Корнеев. - М.: ИНФРА-М, 2017. - 304 с.
11. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей / В.Ф. Шаньгин. - М.: Форум, Инфра-М, 2017. - 416 с.
12. Ярочкин, В. Безопасность информационных систем / В. Ярочкин. - М.: Ось-89, 2016. - 320 с.

Интернет Ресурсы

13. Информационная безопасность - http://protect.htmlweb.ru
14. Информационная безопасность - http://wikipedia.org
15. Информационная безопасность и защита информации. Учебное пособие – М.: 2004 – 82 с. http://bezopasnik.org›article/book/23.pdf
16. https://moodle.tsu.ru/course/view.phpid=1557
17. http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000534758
18. http://www.intuit.ru/studies/courses/2259/155/info
19. http://www.intuit.rU/studies/courses/l0/10/info

Дополнительная

20. Девянин, П.Н. Анализ безопасности управления доступом и информационными потоками в компьютерных системах / П.Н. Девянин. - М.: Радио и связь, 2013. - 176 с.
21. Айтхожаева Е.Ж., Шерстобитова Т.М. Информатика: учебное пособие.- Алматы: КазНТУ, 2012.

4 Лекция №6. Алфавитное кодирование информации

Цель лекции: изучить теорию кодирования, изучить алфавитное кодирование информации, стандартный код и схемы кодирования.

Содержание лекции: теория кодирования. Алфавитное представление информации. кодирование, взаимно-однозначное кодирование, схемы кодирования. Префикс и постфикс слова. Таблица кодов.

4.1 Теория кодирования, как информационная основа цифровых устройств.

Одним из важнейших разделов теории информации является теория кодирования, основы которой были заложены Хэммингом.

Шеннон установил, что по каналу связи информация может передаваться безошибочно в том случае, если скорость передачи не превышает его пропускной способности. Однако доказательство Шеннона носило неконструктивный характер. Более поздние его исследования и другого американского ученого С. О. Райса показали, что практически любой случайно выбранный код позволяет достичь теоретического предела помехоустойчивости приёма сообщений. Однако такой код имел высокую сложность декодирования: число операций, необходимых для декодирования принятой кодовой комбинации, возрастал экспоненциально росту его длины.

Хэмминг был первым, кто предложил конструктивный метод построения кодов с избыточностью и простым декодированием. Его труд предопределил направление большинства работ в этой области, последовавших позже.

Для теории и техники связи теорема Котельникова и коды Хэмминга имеют исключительное значение, поскольку именно благодаря им перед инженерами открылась ясная перспектива создания цифровых систем, которые в конце XX века произвели революцию в электросвязи и поэтому их с полным основанием называют именами этих учёных.

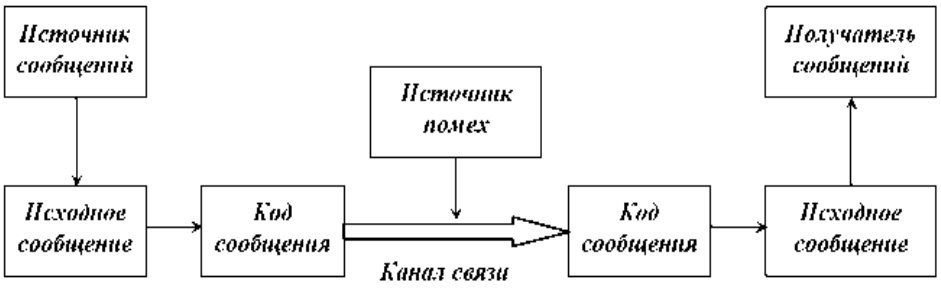


Рисунок 1. Общая схема передачи информации

Цифровые устройства – это устройства для обработки информации, представленной в доступной для компьютера форме."

Кодирование и обработка звуковой информации

Звуковая информация. Звук представляет собой распространяющуюся в воздухе, воде или другой среде волну с непрерывно меняющейся **интенсивностью** и **частотой**.

Человек воспринимает звуковые волны (колебания воздуха) с помощью слуха в форме звука различных **громкости** и **тона**. Чем больше интенсивность звуковой волны, тем громче звук, чем больше частота волны, тем выше тон звука (рис.2).

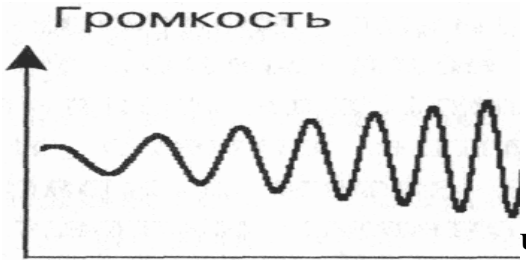


Рисунок 2. Зависимость громкости и высоты тона звука от интенсивности и частоты звуковой волны

Человеческое ухо воспринимает звук с частотой от 20 колебаний в секунду (низкий звук) до 20 000 колебаний в секунду (высокий звук).

Человек может воспринимать звук в огромном диапазоне интенсивностей, в котором максимальная интенсивность больше минимальной в 10^{14} раз (в сто тысяч миллиардов раз). Для измерения громкости звука применяется специальная **единица "децибел"** (дбл) (табл. 5.1). Уменьшение или увеличение громкости звука на 10 дбл соответствует уменьшению или увеличению интенсивности звука в 10 раз.

Таблица 5.1. Громкость звука

Звук	**Громкость в децибелах**
Нижний предел чувствительности человеческого уха	**0**
Шорох листьев	**10**
Разговор	**60**
Гудок автомобиля	**90**
Реактивный двигатель	**120**
Болевой порог	**140**

Временная дискретизация звука. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму с помощью временной дискретизации.

Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука.

Таким образом, непрерывная зависимость громкости звука от времени A(t) заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность "ступенек" (рисунок.2).

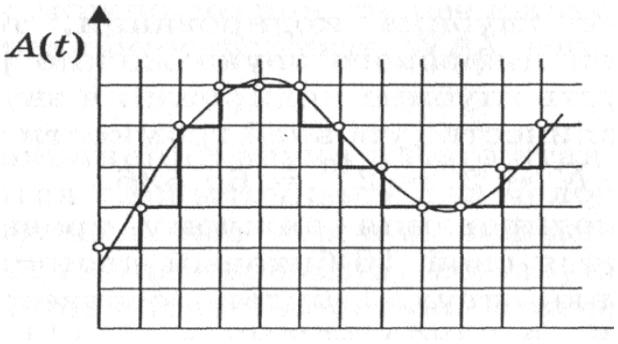


Рисунок.2. Временная дискретизация звука

Частота дискретизации. Для записи аналогового звука и этого преобразования в цифровую форму используется микрофон, подключенный к звуковой плате. **Качество полученного цифрового звука зависит от количества измерений уровня громкости звука в единицу времени**, т. е. **частоты дискретизации**. Чем большее количество измерений производится за I секунду (чем больше частота дискретизации), тем точнее "лесенка" цифрового звукового сигнала повторяет кривую диалогового сигнала.

Частота дискретизации звука - *это количество измерений громкости звука за одну секунду.*

Частота дискретизации звука может лежать в диапазоне от 8000 до 48 000 измерений громкости звука за одну секунду.

Глубина кодирования звука. Каждой "ступеньке" присваивается определенное значение уровня громкости звука. Уровни громкости звука можно рассматривать как набор возможных состояний N, для кодирования которых необходимо определенное количество информации I, которое называется глубиной кодирования звука.

Глубина кодирования звука - *это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.*

Если известна глубина кодирования, то количество уровней громкости цифрового звука можно рассчитать по формуле $N = 2^I$. Пусть глубина кодирования звука составляет 16 битов, тогда количество уровней громкости звука равно:

$N = 2^I = 2^{16} = 65\ 536$.

В процессе кодирования каждому уровню громкости звука присваивается свой 16-битовый двоичный код, наименьшему уровню звука

будет соответствовать код 0000000000000000, а наибольшему - 1111111111111111.

Качество оцифрованного звука. Чем больше частота и глубина дискретизации звука, тем более качественным будет звучание оцифрованного звука. Самое **низкое качество** оцифрованного звука, соответствующее качеству телефонной связи, получается при частоте дискретизации 8000 раз в секунду, глубине дискретизации 8 битов и записи одной звуковой дорожки (режим "моно"). Самое высокое качество оцифрованного звука, соответствующее качеству аудио-CD, достигается при частоте дискретизации 48 000 раз в секунду, глубине дискретизации 16 битов и записи двух звуковых дорожек (режим "стерео").

Необходимо помнить, что чем выше качество цифрового звука, тем больше информационный объем звукового файла. Можно оценить информационный объем цифрового стереозвукового файла длительностью звучания 1 секунда при среднем качестве звука (16 битов, 24 000 измерений в секунду). Для этого глубину кодирования необходимо умножить на количество измерений в 1 секунду и умножить на 2 (стереозвук):

16 бит × 24 000 × 2 = 768 000 бит = 96 000 байт = 93,75 Кбайт.

Звуковые редакторы. Звуковые редакторы позволяют не только записывать и воспроизводить звук, но и редактировать его. Оцифрованный звук представляется в звуковых редакторах в наглядной форме, поэтому операции копирования, перемещения и удаления частей звуковой дорожки можно легко осуществлять с помощью мыши. Кроме того, можно накладывать звуковые дорожки друг на друга (микшировать звуки) и применять различные акустические эффекты (эхо, воспроизведение в обратном направлении и др.).

Звуковые редакторы позволяют изменять качество цифрового звука и объем звукового файла путем изменения частоты дискретизации и глубины кодирования. Оцифрованный звук можно сохранять без сжатия в звуковых файлах **в универсальном формате WAV** или в **формате со сжатием MP3.**

При сохранении звука в форматах со сжатием отбрасываются "избыточные" для человеческого восприятия **звуковые частоты с малой интенсивностью, совпадающие по времени со звуковыми частотами с большой интенсивностью.** Применение такого формата позволяет сжимать звуковые файлы в десятки раз, однако приводит к необратимой потере информации (файлы не могут быть восстановлены в первоначальном виде).

Вывод. В процессе кодирования непрерывного звукового сигнала производится его временная дискретизация. Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, причем для каждого такого участка устанавливается определенная величина амплитуды.

Т.о. при двоичном кодировании непрерывного звукового сигнала он заменяется последовательностью дискретных уровней сигнала.

Рассмотрим пример задачи.

Какой объем будет иметь звуковой монофонический файл содержащий звук, если длительность звука 1 минута, глубина кодирования 8 бит, а частота дискретизации 22050Гц?

Зная частоту дискретизации и длительность звука легко установить количество измерений уровня сигнала за все время. Если частота дискретизации 22050Гц — значит за 1 секунду происходит 22050 измерений, а за минуту таких измерений будет 22050*60=1 323 000.

На одно измерение требуется 8 бит памяти, следовательно на 1 323 000 измерений потребуется 1 323 000*8 = 10 584 000 бит памяти. Разделив полученное число на 8 получим объем файла в байтах — 10584000/8=1 323 000 байт. Далее, разделив полученное число на 1024 получим объем файла в килобайтах — 1 291,9921875 Кбайт. А разделив полученное число еще раз на 1024 и округлив до сотых получим размер файла в мегабайтах — 1 291,9921875/1024=1,26Мбайт.

ЗАДАЧА 2.

Можно оценить информационный объем стереоаудиофайла длительностью звучания 1 секунда при высоком качестве звука (16 битов, 48 кГц). Для этого количество битов, приходящихся на одну выборку, необходимо умножить на количество выборок в 1 секунду и умножить на 2 (стерео):

Решение: 16 бит • 48 000 • 2 = 1 536 000 бит = 192 000 байт = 187,5 Кбайт.

ЗАДАЧА 3.

Оценить информационный объем цифрового стерео звукового файла длительностью звучания 1 минута при среднем качестве звука (16 битов, 24 кГц).

Решение: 16 бит × 24 000 × 2 × 60 = 46 080 000 бит = 5 760 000 байт = 5 625 Кбайт ≈ 5,5 Мбайт

4.1.1 Алфавитное представление информации, отображение информации

С появлением языка, а затем и знаковых систем расширились возможности общения между людьми. Это позволило хранить идеи, полученные знания и любые данные, передавать их различными способами на расстояние и в другие времена — не только своим современникам, но и будущим поколениям. До наших дней дошли творения предков, которые с помощью различных символов увековечили себя и свои деяния в памятниках и надписях. Наскальные рисунки (петроглифы) до сих пор служат загадкой для ученых. Возможно, таким способом древние люди хотели вступить в контакт с нами, будущими жителями планеты и сообщить о событиях их жизни.

<u>При алфавитном подходе к измерению информации количество информации зависит не от содержания, а от размера текста и мощности алфавита.</u>

При использовании двоичной системы (алфавит состоит из двух знаков: 0 и 1) каждый двоичный знак несет 1 бит информации. Интересно, что сама единица измерения информации «бит» получила свое название от английского сочетания «**bi**nary digi**t**» - «двоичная цифра».

Применение алфавитного подхода удобно прежде всего при использовании технических средств работы с информацией. В этом случае теряют смысл понятия «новые — старые», «понятные — непонятные» сведения. Алфавитный подход является объективным способом измерения информации в отличие от субъективного содержательного подхода.

Удобнее всего измерять информацию, когда размер алфавита N равен целой степени двойки. Например, если N=16, то каждый символ несет 4 бита информации потому, что $2^4 = 16$. А если N =32, то один символ «весит» 5 бит.

Ограничения на максимальный размер алфавита теоретически не существует. Однако есть алфавит, который можно назвать достаточным. С ним мы скоро встретимся при работе с компьютером. Это алфавит мощностью 256 символов. В алфавит такого размера можно поместить все практически необходимые символы: латинские и русские буквы, цифры, знаки арифметических операций, всевозможные скобки, знаки препинания....

Поскольку $256 = 2^8$, то один символ этого алфавита «весит» 8 бит. Причем 8 бит информации — это настолько характерная величина, что ей даже присвоили свое название — *байт*.

1 байт = 8 бит.

Сегодня очень многие люди для подготовки писем, документов, статей, книг и пр. используют компьютерные текстовые редакторы. Компьютерные редакторы, в основном, работают с алфавитом размером 256 символов.

В этом случае легко подсчитать объем информации в тексте. Если 1 символ алфавита несет 1 байт информации, то надо просто сосчитать количество символов; полученное число даст информационный объем текста в байтах.

Пусть небольшая книжка, сделанная с помощью компьютера, содержит 150 страниц; на каждой странице — 40 строк, в каждой строке — 60 символов. Значит страница содержит 40x60=2400 байт информации. Объем всей информации в книге: 2400 x 150 = 360 000 байт.

4.2 Лекция 7. Алфавиты и слова, кодирование, взаимно-однозначное кодирование, схемы кодирования

Множество символов, с помощью которых записывается текст, называется *алфавитом*.

Число символов в алфавите – это его *мощность*.

Формула определения количества информации: $N = 2\ b$,

где N – мощность алфавита (количество символов),

b – количество бит (информационный вес символа).

В алфавит мощностью 256 символов можно поместить практически все необходимые символы. Такой алфавит называется *достаточным.*

Т.к. $256 = 2^{8}$, то вес 1 символа – 8 бит.

Единице измерения 8 бит присвоили название *1 байт:*

Двоичный код каждого символа в компьютерном тексте занимает 1 байт памяти.

А теперь познакомимся с другим способом измерения информации. Этот способ не связывает количество информации с содержанием сообщения, и называется он алфавитным подходом.

При алфавитном подходе к определению количества информации отвлекаются от содержания информации и рассматривают информационное сообщение как последовательность знаков определенной знаковой системы.

Проще всего разобраться в этом на примере текста, написанного на каком-нибудь языке. Для нас удобнее, чтобы это был русский язык.

Все множество используемых в языке символов будем традиционно называть алфавитом. Обычно под алфавитом понимают только буквы, но поскольку в тексте могут встречаться знаки препинания, цифры, скобки, то мы их тоже включим в алфавит. В алфавит также следует включить и пробел, т.е. пропуск между словами.

Полное количество символов алфавита принято называть мощностью алфавита. Будем обозначать эту величину буквой N. Например, мощность алфавита из русских букв и отмеченных дополнительных символов равна 54.

Представьте себе, что текст к вам поступает последовательно, по одному знаку, словно бумажная ленточка, выползающая из телеграфного аппарата. Предположим, что каждый появляющийся на ленте символ с **одинаковой вероятностью** может быть любым символом алфавита. В действительности это не совсем так, но для упрощения примем такое предположение.

В каждой очередной позиции текста может появиться любой из N символов. Тогда, согласно известной нам формуле, каждый такой символ несет I бит информации, которое можно определить из решения уравнения: $2^{I} = 54$. Получаем: **I = 5.755 бит**. В одном символе

Вот сколько информации несет один символ в русском тексте! А теперь для того, чтобы найти количество информации во всем тексте, нужно посчитать число символов в нем и умножить на I.

Посчитаем количество информации на одной странице книги. Пусть страница содержит 50 строк. В каждой строке — 60 символов. Значит, на странице умещается 50x60=3000 знаков. Тогда объем информации будет равен: 5,755 x 3000 = 17265 бит.

4.2.1 Кодирование

Кодирование, т. е. перевод информации из одной знаковой системы в другую, производится с помощью таблиц соответствия знаковых систем, которые устанавливают взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем. Пример такой таблицы — таблица кодов ASCII (американский стандартный код обмена информацией), устанавливающая соответствие между интернациональными знаками алфавита и их числовыми компьютерными кодами.

символ	10-й код	2-й код	символ	10-й код	2-й код	символ	10-й код	2-й код	символ	10-й код	2-й код
Ђ	128	10000000		160	10100000	А	192	11000000	а	224	11100000
Ѓ	129	10000001	Ў	161	10100001	Б	193	11000001	б	225	11100001
‚	130	10000010	ў	162	10100010	В	194	11000010	в	226	11100010
ѓ	131	10000011	Ј	163	10100011	Г	195	11000011	г	227	11100011
„	132	10000100	¤	164	10100100	Д	196	11000100	д	228	11100100
…	133	10000101	Ґ	165	10100101	Е	197	11000101	е	229	11100101
†	134	10000110	¦	166	10100110	Ж	198	11000110	ж	230	11100110
‡	135	10000111	§	167	10100111	З	199	11000111	з	231	11100111
€	136	10001000	Ё	168	10101000	И	200	11001000	и	232	11101000
‰	137	10001001	©	169	10101001	Й	201	11001001	й	233	11101001
Љ	138	10001010	Є	170	10101010	К	202	11001010	к	234	11101010
‹	139	10001011	«	171	10101011	Л	203	11001011	л	235	11101011
Њ	140	10001100	¬	172	10101100	М	204	11001100	м	236	11101100
Ќ	141	10001101	-	173	10101101	Н	205	11001101	н	237	11101101
Ћ	142	10001110	®	174	10101110	О	206	11001110	о	238	11101110
Џ	143	10001111	Ї	175	10101111	П	207	11001111	п	239	11101111
ђ	144	10010000	°	176	10110000	Р	208	11010000	р	240	11110000
‘	145	10010001	±	177	10110001	С	209	11010001	с	241	11110001
’	146	10010010	І	178	10110010	Т	210	11010010	т	242	11110010
“	147	10010011	і	179	10110011	У	211	11010011	у	243	11110011
”	148	10010100	ґ	180	10110100	Ф	212	11010100	ф	244	11110100
•	149	10010101	µ	181	10110101	Х	213	11010101	х	245	11110101
–	150	10010110	¶	182	10110110	Ц	214	11010110	ц	246	11110110
—	151	10010111	·	183	10110111	Ч	215	11010111	ч	247	11110111
□	152	10011000	ё	184	10111000	Ш	216	11011000	ш	248	11111000
™	153	10011001	№	185	10111001	Щ	217	11011001	щ	249	11111001
ѓ	154	10011010	є	186	10111010	Ъ	218	11011010	ъ	250	11111010
›	155	10011011	»	187	10111011	Ы	219	11011011	ы	251	11111011
ѝ	156	10011100	ј	188	10111100	Ь	220	11011100	ь	252	11111100
ќ	157	10011101	Ѕ	189	10111101	Э	221	11011101	э	253	11111101
ћ	158	10011110	ѕ	190	10111110	Ю	222	11011110	ю	254	11111110
џ	159	10011111	ї	191	10111111	Я	223	11011111	я	255	11111111

Кодирование информации в компьютере

Вся информация , которую обрабатывает компьютер, должна быть представлена двоичным кодом с помощью двух цифр — 0 и 1. Эти два символа принято называть двоичными цифрами, или битами. С помощью двух цифр 1 и 0 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организовано два важных процесса:

- кодирование, которое обеспечивается устройствами ввода при преобразовании входной информации в форму, воспринимаемую компьютером, то есть в двоичный код;
- декодирование, которое обеспечивается устройствами вывода при преобразовании данных из двоичного кода в форму, понятную человеку.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов. Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

- 0 — отсутствие электрического сигнала или сигнал имеет низкий уровень;
- 1 — наличие сигнала или сигнал имеет высокий уровень.

Эти состояния легко различать. **Недостаток двоичного кодирования — длинные коды.** Но в технике легче иметь дело с большим числом простых элементов, чем с небольшим количеством сложных.

Вам и в быту ежедневно приходится сталкиваться с устройством, которое может находиться только в двух устойчивых состояниях: включено/выключено. Конечно же, это хорошо знакомый всем выключатель. А вот придумать выключатель, который мог бы устойчиво и быстро переключаться в любое из 10 состояний, оказалось невозможным. **В результате после ряда неудачных попыток разработчики пришли к выводу о невозможности построения компьютера на основе десятичной системы счисления.** И в основу представления чисел в компьютере была положена именно двоичная система счисления.

В настоящее время существуют разные способы двоичного кодирования и декодирования информации в компьютере. В первую очередь это зависит от вида информации, а именно, что должно кодироваться: текст, числа, графические изображения или звук. Кроме того, при кодировании чисел важную роль играет то, как они будут использоваться: в тексте, в расчетах или в процессе ввода-вывода. Накладываются также и особенности технической реализации.

Кодирование чисел

Система счисления — совокупность приемов и правил записи чисел с помощью определенного набора символов.

Для записи чисел могут использоваться не только цифры, но и буквы (например, запись римских цифр — XXI). Одно и то же число может быть по-разному представлено в различных системах счисления.

Кодирование символьной информации

Нажатие алфавитно-цифровой клавиши на клавиатуре приводит к тому, что в компьютер посылается сигнал в виде двоичного числа, представляющего собой одно из значений кодовой таблицы. Кодовая таблица - это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII (American Standart Code for Informational Interchange - американский стандартный код информационного обмена).

A	А	•—	L	Л	•—••	C	Ц	—•—•
B	Б	—•••	M	М	——		Ч	——— •
W	В	•——	N	Н	—•		Ш	—— ——
G	Г	——•	O	О	———	Q	Щ	——•—
D	Д	—••	P	П	•——•		Ъ	•——•—•
E	Е	•	R	Р	•—•		Ы	—•——
V	Ж	•••—	S	С	•••	X	Ь	—••—
Z	З	——••	T	Т	—		Э	••—••
I	И	••	U	У	••—		Ю	••——
J	Й	•———	F	Ф	••—•		Я	•—•—
K	К	—•—	H	Х	••••			

Кодовая таблица азбуки Морзе • • • — — — • • • *SOS*

Для хранения двоичного кода одного символа выделен 1 байт = 8 бит. Учитывая, что каждый бит принимает значение 1 или 0, количество возможных сочетаний единиц и нулей равно $2^8 = 256$.

Значит, с помощью 1 байта можно получить 256 разных двоичных кодовых комбинаций и отобразить с их помощью 256 различных символов. Эти коды и составляют таблицу ASCII.

Пример, при нажатии клавиши с буквой **S в память компьютера записывается код 01010011**. При выводе буквы S на экран компьютер выполняет декодирование — на основании этого двоичного кода строится изображение символа.

SUN (СОЛНЦЕ) - 01010011 010101101 01001110

Стандарт ASCII кодирует первые 128 символов от 0 до 127: цифры, буквы латинского алфавита, управляющие символы. Первые 32 символа являются управляющими и предназначены в основном для передачи команд управления. Их назначение может варьироваться в зависимости от программных и аппаратных средств. Вторая половина кодовой таблицы (от 128 до 255) американским стандартом не определена и предназначена для символов национальных алфавитов, псевдографических и некоторых

математических символов. В разных странах могут использоваться различные варианты второй половины кодовой таблицы.

Обратите внимание! Цифры кодируются по стандарту ASCII записываются в двух случаях - при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то осуществляется их преобразование в другой двоичный код.

Для сравнения рассмотрим число 45 для двух вариантов кодирования.

При использовании в тексте это число потребует для своего представления 2 байта, поскольку каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII . ***В двоичной системе - 00110100 00110101.***

При использовании в вычислениях код этого числа будет получен по специальным правилам перевода и представлен в виде 8-разрядного двоичного числа 00101101, на что потребуется 1 байт.

4.2.2 Взаимно однозначное кодирование

Рассмотрим необходимые и достаточные условия взаимной однозначности кодирования. Теорема 1. Алфавитное префиксное кодирование является взаимно однозначным. Доказательство этого факта нетрудно провести от противного. Пусть некоторое кодовое слово допускает, по крайней мере, две различные расшифровки. Тогда элементарные коды, начиная с которых появляется различие, не будут обладать свойством префикса. Префиксность кода не является необходимым условием взаимной однозначности кодирования.

Пример 1. Алфавитное кодирование по схеме

$$a_1 \rightarrow 0$$

$$a_2 \rightarrow 01$$

не является префиксным, хотя свойством взаимной однозначности оно обладает. Достаточно в двоичном кодовом слове заменить все фрагменты 01 на буквы а2, а возможные оставшиеся нули - на буквы а1.

Необходимое условие взаимной однозначности кодирования установлено Б. Макмилланом [10]. Теорема 2. Если алфавитное кодирование является взаимно однозначным, то спектр кода удовлетворяет неравенству

$$\sum_{i=1}^{r} 2^{-l_i} \leq 1$$

Распознавание взаимной однозначности кода, не являющегося префиксным, может потребовать проверки счетного множества слов. Поэтому

важно использовать некоторую процедуру (алгоритм), которая бы упорядочивала эти проверки

4.2.3 Схемы кодирования.

Схема кодирования и декодирования

Принципиальная схема цифровой системы связи изображена на рисунке 5

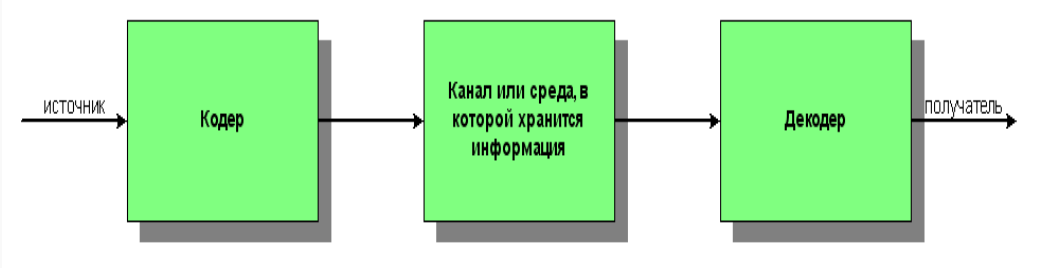


Рисунок 5.

Эта же самая модель описывает и систему хранения информации, если среду, в которой хранится информация, рассматривать как канал. Типичным каналом для передачи информации является телефонный канал. Для хранения информации обычно используется магнитофон, имеющий записывающую и считывающую головки. Источником информации является, как правило, сообщение, состоящее из двоичных или десятичных цифр, или же текст, записанный с помощью некоторого алфавита. Кодер преобразует эти сообщения в сигналы, которые могут быть переданы по каналу. Типичными сигналами являются электрические сигналы с некоторыми ограничениями по мощности, полосе частот и продолжительности. Эти сигналы поступают в канал и искажаются шумом. Затем искаженный сигнал поступает в декодер, который восстанавливает посланное сообщение и направляет его получателю. Основная задача, возникающая в технике связи, состоит в том, чтобы построить кодер и декодер, хотя она может включать в себя также задачу улучшения самого канала. Заметим, что кодер включает в себя устройство, которое обычно называют модулятором, а в декодер входит устройство, которое обычно называют детектором.

Использование кодов, исправляющих ошибки, является попыткой разрешить одновременно обе проблемы.

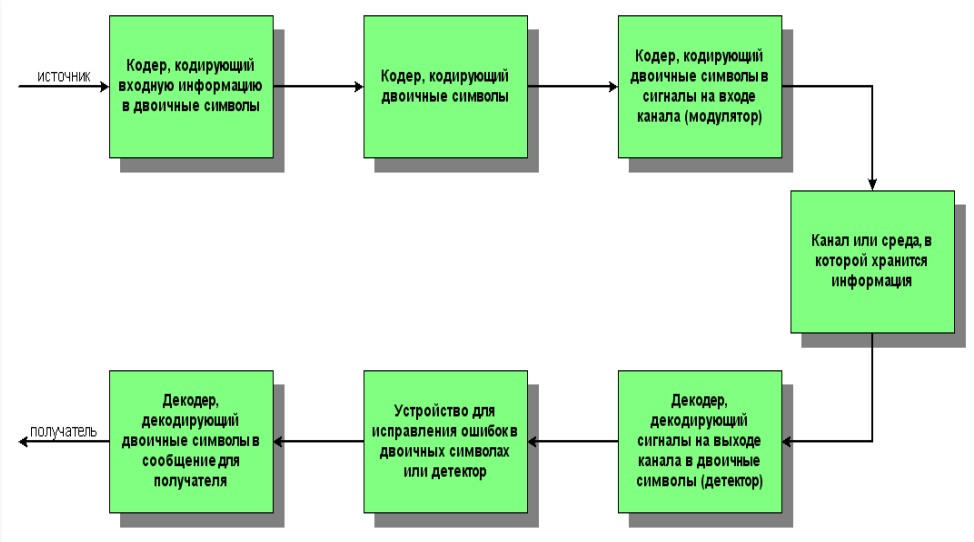


Рисунок 6.

Преобразователи в типичной системе связи, представленной на рисунке 2, просто переводят символы одного алфавита в символы другого алфавита. Обычно оба алфавита невелики — типичным преобразованием является переход от десятичного алфавита к двоичному. На вход модулятора подается один символ, который превращается на его выходе в сигнал соответствующей формы, т.е. в импульс, всегда сопоставляемый именно этому символу. То, что в каждый момент времени модулятором обрабатывается единственный символ, является ограничением, которое снижает пропускную способность канала. Демодулятор осуществляет противоположную операцию: он пытается сопоставить принятому на входе и искаженному шумом сигналу некоторый символ канала. Независимое демодулирование отдельных импульсов приводит к дальнейшему уменьшению пропускной способности канала. Кодер и декодер реализуют используемый код, исправляющий ошибки. Даже если разделение приемника на демодулятор и декодер, а передатчика на модулятор и кодер приводит к уменьшению предельных возможностей данной системы передачи, тем не менее все же можно добиться произвольно малой вероятности ошибочного декодирования, хотя и при несколько меньшей пропускной способности канала. Кроме того, что даже важнее с практической точки зрения, использование кодирования дает возможность спроектировать и построить высокоэффективные системы передачи и хранения информации.

4.3 Лекция 8. Теорема Маркова о взаимной однозначности алфавитного кодирования. Неравенство Макмиллана. Префиксные и суффиксные коды. Стандартный алфавит, в том числе двухбуквенный. Американский стандартный код ASCII, Unicode.

При построении схем алфавитного кодирования возникают две основные задачи: 1) Проблема распознавания взаимной однозначности

алфавитного кодирования; 2) Задача построения схемы кодирования, обеспечивающего наибольшее сжатие.

1.2. Проблема распознавания взаимной однозначности алфавитного кодирования

Пример 1.1. Рассмотрим схему

$$f_1 : \begin{cases} b_1 \to 0 \\ b_2 \to 01. \end{cases}$$

f_1 задает взаимно-однозначное кодирование. Достаточно заметить, что перед каждым вхождением символа 1 стоит символ 0, поэтому каждое вхождение 1 вместе с предшествующим нулем кодирует букву b_2. Символ 0, за которым не следует 1, кодирует букву b_1. Например, последовательность $\beta = 001010001$ имеет единственную расшифровку $\alpha = b_1 b_2 b_2 b_1 b_1 b_2$.

Пример 1.2. Рассмотрим схему

$$f_2 : \begin{cases} b_1 \to 0 \\ b_2 \to 01 \\ b_3 \to 001. \end{cases}$$

Кодирование, задаваемое схемой f_2, не является взаимно-однозначным. Последовательность $\beta = 001$ допускает две расшифровки $\alpha_1 = b_1 b_2$ и $\alpha_2 = b_3$.

Для непосредственной проверки взаимной однозначности необходимо в общем случае проверить бесконечное множество пар слов.

Определение 1.1. *Пусть слово α имеет вид $\alpha_1 \alpha_2$. Тогда α_1 называется префиксом слова α, а α_2 - суффиксом α. Если $0<|\alpha_1|<|\alpha|$, то α_1 называется собственным префиксом α, если $0<|\alpha_2|<|\alpha|$, то α_2|– собственный суффикс α.*

Определение 1.2. *Схема алфавитного кодирования f обладает свойством префикса, если для любых i и j $(1 \le i, j \le m, i \ne j)$ элементарный код v_i не является префиксом элементарного кода v_j.*

Определение 1.3. *Алфавитное кодирование, схема которого обладает свойством префикса, называется префиксным.*

Теорема 1.1. *Если схема обладает свойством префикса, то алфавитное кодирование является взаимно-однозначным.*

Таким образом, свойство префикса является достаточным условием взаимной однозначности.

Теорема 1.2. *Если алфавитное кодирование со схемой f обладает свойством взаимной однозначности, то длины элементарных кодов* $l_i = |v_i| \, (i = 1, \ldots, m)$ *удовлетворяют неравенству Мак-Миллана:*

$$\sum_{i=1}^{m} 2^{-l_i} \le 1.$$

Неравенство Мак-Миллана является необходимым условием взаимной однозначности кода со схемой *f*, но не достаточным.

Рассмотрим схему f_2, приведенную ранее:

$$f_2 : \begin{cases} b_1 \to 0 \\ b_2 \to 01 \\ b_3 \to 001. \end{cases}$$

Для нее $l_1 = 1$, $l_2 = 2$, и $l_3 = 3$, и неравенство Мак-Миллана выполняется: $2^{-1} + 2^{-2} + 2^{-3} = 7/8 < 1$. Однако задаваемое схемой f_1 кодирование не является взаимно-однозначным.

Теорема 1.3. *Если набор натуральных чисел* $l_1, l_2, \ldots l_m$ *удовлетворяет неравенству Мак-Миллана, то существует префиксное кодирование, удовлетворяющее равенствам:*

$$|v_1| = l_1, |v_2| = l_2, \ldots, |v_m| = l_m.$$

Следствие. *Если существует взаимно-однозначное алфавитное кодирование с заданными длинами элементарных кодов, то существует также префиксное кодирование с теми же длинами элементарных кодов.*

Неравенство Мак-Миллана в силу справедливости теоремы 1.3 можно рассматривать и как достаточное условие взаимной однозначности, в том смысле, что если неравенство для некоторой схемы кодирования выполняется, можно заменить эту схему схемой со свойством префикса с теми же длинами элементарных кодов.

Пусть задан код $V = (v_1, v_2, \ldots v_m)$. Элементарные коды определяют бинарное кодовое дерево. Из каждой вершины дерева выходит не более двух ребер в вершины следующего яруса, левое из которых помечается символом 0, а правое – символом 1. Элементарным

кодам соответствуют вершины дерева, определяемые путем, идущим от корня. Если код префиксный, элементарные коды расположены в листьях дерева.

Дерево называется насыщенным, если из каждой вершины, не являющейся листом, в следующий ярус выходит ровно два ребра.

На рис. 1.1 изображено дерево для схемы префиксного кодирования f_3.

$$f_3 : \begin{cases} b_1 \to 00 \\ b_2 \to 111 \\ b_3 \to 100 \\ b_4 \to 01 \\ b_5 \to 101 \\ b_6 \to 1100 \\ b_7 \to 1101. \end{cases}$$

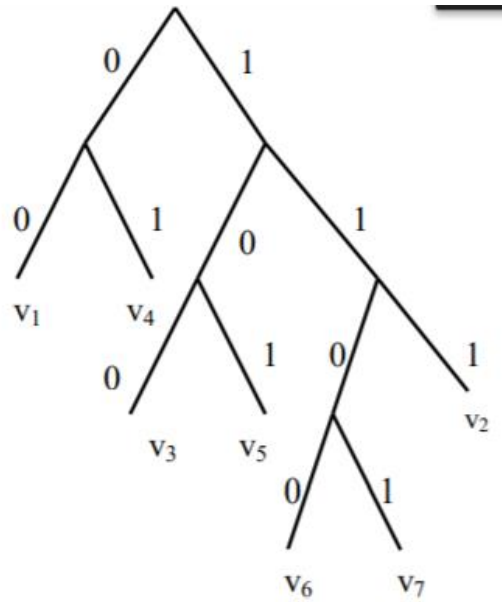


Рис.1.1. Кодовое дерево для схемы кодирования f_3.

Проблема распознавания взаимной однозначности алфавитного кодирования решена Ал. А. Марковым (1963 г.). Алгоритм распознавания состоит в следующем.

Для кода $V = (v_1, v_2, \ldots, v_m)$ пусть S_1 - множество слов, обладающих следующим свойством. Слово β является собственным префиксом некоторого элементарного кода v_i и одновременно собственным суффиксом некоторого v_j. Положим $S = S_1 \cup \{\lambda\}$ (λ - пустое слово).

Сопоставим коду V ориентированный граф G, вершинами которого являются элементы множества S. Вершины α и β соединяем ориентированным ребром (α, β), если существует элементарный код v_j и последовательность элементарных кодов $P = v_{i_1} v_{i_2} \ldots v_{i_k}$, такие, что $v_j = \alpha v_{i_1} v_{i_2} \ldots v_{i_k} \beta$. При этом P может быть пустой, если α и β оба непустые.

Ребру (α, β) припишем последовательность $v_{i_1} v_{i_2} \ldots v_{i_k}$. Ребро ($\lambda$, λ) присутствует в графе тогда и только тогда, когда существует v_j и последовательность $P = v_{i_1} v_{i_2} \ldots v_{i_k}$ ($k \geq 2$), такие, что $v_j = v_{i_1} v_{i_2} \ldots v_{i_k}$.

Теорема 1.4. *Алфавитный код V является взаимно-однозначным тогда и только тогда, когда в графе G отсутствуют ориентированные циклы, проходящие через вершину λ.*

Пример 1.3. Пусть $B = (b_1, b_2, b_3)$, $V = \{1, 010, 101\}$. Построим множества S_1 и S: $S_1 = \{0, 1, 01, 10\}$, $S = \{0, 1, 01, 10, \lambda\}$. Для этого выпишем все нетривиальные разложения элементарных кодов v_i.

Для v_1 нет нетривиальных разложений.

$v_2 = 010 = 0\ v_1\ 0 = 01\ \lambda\ 0 = 0\ \lambda\ 10$

$v_3 = 101 = \lambda\ v_1\ 01 = 10\ v_1\ \lambda = 1\ \lambda\ 01 = 10\ \lambda\ 1.$

Соответствующий коду граф изображен на рис.1.2.

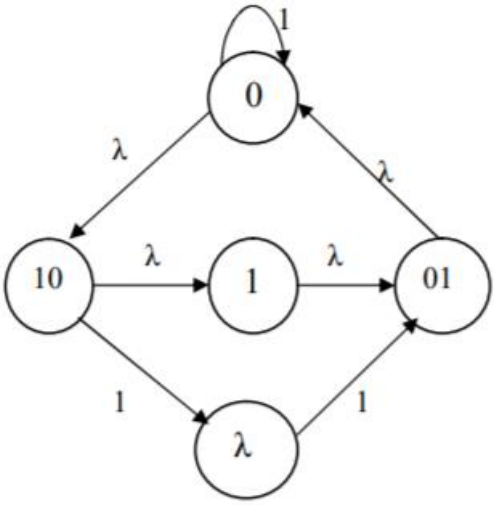


Рис.1.2. Граф для примера 1.3

Граф содержит ориентированный цикл, проходящий через вершину λ, следовательно, код V не является взаимно-однозначным.

По графу нетрудно построить двоичную последовательность, допускающую две расшифровки. Для этого достаточно, начиная с вершины λ, приписать друг к другу двоичные последовательности, соответствующие вершинам и ребрам графа, вдоль найденного цикла.

Слово γ=1010101, соответствующее циклу, допускает две расшифровки: $b_1b_2b_3$ и $b_3b_2b_1$.

1.3. Алгоритм построения префиксного кода по набору длин элементарных кодов

Пусть задан набор чисел $l_1, l_2, \ldots l_m$, удовлетворяющих неравенству Мак-Миллана: $\sum_{i=1}^{m} 2^{-l_i} \le 1$. В силу теоремы 1.3 существует префиксный код с набором длин $(l_1, l_2, \ldots l_m)$ элементарных кодов. Приведем алгоритм К. Шеннона построения префиксного кода по набору длин.

Будем полагать $l_1 \le l_2 \le \ldots \le l_m$. Построим последовательность чисел $q_1, q_2, \ldots, q_m$ по следующим правилам:

$q_1 = 0,$

$q_{i+1} = q_i + 2^{-l_i} (i = 1, 2, \ldots, m-1).$

Очевидно, $0 \le q_i < 1$ и q_i имеет единственное представление в виде двоичной дроби с l_i знаками после запятой:

$q_i = \sum_{j=1}^{l_i} c_j^{(i)} \cdot 2^{-j}$, где $c_j^{(i)}$=0 или $c_j^{(i)}$=1.

Рассмотрим код $V = (v_1, v_2, \ldots, v_m)$, где $v_i = c_1^{(i)} c_2^{(i)} \ldots c_{l_i}^{(i)}$.

Так как наборы длин упорядочены по неубыванию, при $h > i$ выполняются неравенства $l_h \ge l_i$ и $q_h \ge q_i + 2^{-l_i}$. Поэтому элементарный код v_h отличается от элементарного кода v_i в l_i первых разрядах. Следовательно, построенный код является префиксным.

Построим схему f алфавитного кодирования, выбирая в качестве элементарного кода v_i последовательность из 0 и 1 длины l_i, образующую дробную часть числа q_i:

$$f:\begin{cases} v_1 = 00 \\ v_2 = 010 \\ v_3 = 011 \\ v_4 = 100 \\ v_5 = 1010 \\ v_6 = 1011 \\ v_7 = 1100. \end{cases}$$

Нетрудно убедиться в том, что построенный код является префиксным.

Пример 1.5. Рассмотрим набор чисел L=(2,3,3,3,4,4,4). Так как

$2^{-2}+2^{-3}+2^{-3}+2^{-3}+2^{-4}+2^{-4}+2^{-4}=\frac{13}{16}<1$, неравенство Мак-Миллана выполняется.

Построим последовательность чисел $q_1, q_2, q_3, q_4, q_5, q_6, q_7$, записывая их в двоичной системе счисления.

$$q_1 = 0{,}00$$
$$q_2 = 0 + 2^{-2} = 0{,}010$$
$$q_3 = 0{,}01 + 2^{-3} = 0{,}01 + 0{,}001 = 0{,}011$$
$$q_4 = 0{,}011 + 2^{-3} = 0{,}011 + 0{,}001 = 0{,}100$$
$$q_5 = 0{,}1 + 2^{-3} = 0{,}1 + 0{,}001 = 0{,}1010$$
$$q_6 = 0{,}101 + 2^{-4} = 0{,}101 + 0{,}0001 = 0{,}1011$$
$$q_7 = 0{,}1011 + 2^{-4} = 0{,}1011 + 0{,}0001 = 0{,}1100.$$

4.3.1 Префиксные и постфиксные выражения

Когда вы записываете арифметическое выражение вроде B * C, то его форма предоставляет вам достаточно информации для корректной интерпретации. В данном случае мы знаем, что переменная B умножается на переменную C, поскольку оператор умножения * находится в выражении между ними. Такой тип записи называется **инфиксной**, поскольку оператор расположен *между* (*in between*) двух операндов, с которыми он работает.

Рассмотрим другой инфиксный пример: A + B * C. Операторы + и * по-прежнему располагаются между операндами, но тут уже есть проблема. С какими именно операндами они будут работать? + работает с A и B или * принимает B и C? Выражение выглядит неоднозначно.

Фактически, вы можете читать и писать выражения такого типа долгое время, и они не будут вызывать у вас вопросов. Причина в том, что вы кое-что

знаете о + и *. Каждый оператор имеет свой **приоритет**. Операторы с высоким приоритетом используются прежде операторов с низким. Единственной вещью, которая может изменить порядок приоритетов, являются скобки. Для арифметических операций умножение и деление стоят выше сложения и вычитания. Если появляются два оператора одинакового приоритета, то используются порядок слева направо, или их ассоциативность.

Давайте интерпретируем вызвавшее затруднение выражение A + B * C, используя приоритет операторов. B и C перемножаются первыми, затем к результату добавляется A. (A + B) * C заставит выполнить сложение A и B перед умножением. В выражении A + B + C по очерёдности (через ассоциативность) первым будет вычисляться самый левый +.

Хотя это очевидно для вас, помните: компьютер нуждается в точном знании того, как и в какой последовательности вычисляются операторы. Одним из способов записи выражения, гарантирующим, что не возникнет путаницы по отношению к порядку операций, является создание того, что называется выражением **с полностью расставленными скобками**. Такой тип выражения использует пару скобок для каждого оператора. Скобки диктуют порядок операций, так что здесь не возникает многозначности. Так же отпадает необходимость помнить правила расстановки приоритетов.

Выражение A + B * C + D может быть переписано как ((A + (B * C)) + D) с целью показать, что умножение происходит в первую очередь, а затем следует крайнее левое сложение. A + B + C + D перепишется в (((A + B) + C) + D), поскольку операции сложения ассоциируются слева направо.

Существует ещё два очень важных формата выражений, которые на первый взгляд могут показаться вам неочевидными. Рассмотрим инфиксную запись A + B. Что произойдёт, если мы поместим оператор перед двумя операндами? Результирующее выражение будет + A B. Также мы можем переместить оператор в конец, получив A B +. Всё это выглядит несколько странным.

Эти изменения позиции оператора по отношению к операндам создают два новых формата - **префиксный** и **постфиксный**. Префиксная запись выражения требует, чтобы все операторы предшествовали двум операндам, с которыми они работают. Постфиксная, в свою очередь, требует, чтобы операторы шли после соответствующих операндов. Несколько дополнительных примеров помогут прояснить этот момент (см. *таблицу 2*).

A + B * C в префиксной нотации можно переписать как + A * B C. Оператор умножения ставится непосредственно перед операндами B и C, указывая на приоритет * над +. Затем следует оператор сложения перед A и результатом умножения.

В постфиксной записи выражение выглядит как A B C * +. Порядок операций вновь сохраняется, поскольку * находится непосредственно после B и C, обозначая, что он имеет приоритет выше следующего +. Хотя операторы перемещаются и теперь находятся до или после соответствующих операндов,

порядок последних по отношению друг к другу остаётся в точности таким, как был.

Таблица 2: Примеры инфиксной, префиксной и постфиксной записи

Инфиксная запись	Префиксная запись	Постфиксная запись
A + B	+ A B	A B +
A + B * C	+ A * B C	A B C * +

А сейчас рассмотрим инфиксное выражение (A + B) * C. Напомним, что в этом случае запись требует наличия скобок для указания выполнить сложение перед умножением. Однако, когда A + B записывается в префиксной форме, то оператор сложения просто помещается перед операндами: + A B. Результат этой операции является первым операндом для умножения. Оператор умножения перемещается в начало всего выражения, давая нам * + A B C. Аналогично, в постфиксной записи A B + явно указывается, что первым происходит сложение. Умножение может быть выполнено для получившегося результата и оставшегося операнда C. Соответствующим постфиксным выражением будет A B + C *.

Рассмотрим эти три выражения ещё раз (см. *таблицу 3*). Происходит что-то очень важное. Куда ушли скобки? Почему они не нужны нам в префиксной и постфиксной записи? Ответ в том, что операторы больше не являются неоднозначными по отношению к своим операндам. Только инфиксная запись требует дополнительных символов. Порядок операций внутри префиксного и постфиксного выражений полностью определён позицией операторов и ничем иным. Во многом именно это делает инфиксную запись наименее желательной нотацией для использования.

Таблица 3: Выражение со скобками

Инфиксное выражение	Префиксное выражение	Постфиксное выражение
(A + B) * C	* + A B C	A B + C *

Таблица 4 демонстрирует некоторые дополнительные примеры инфиксных выражений и эквивалентных им префиксных и постфиксных записей. Убедитесь, что вы понимаете, почему они эквивалентны с точки зрения порядка выполнения операций.

Таблица 4: Дополнительные примеры инфиксной, префиксной и постфиксной записи

Инфиксное выражение	Префиксное выражение	Постфиксное выражение
A + B * C + D	+ + A * B C D	A B C * + D +
(A + B) * (C + D)	* + A B + C D	A B + C D + *

Таблица 4: Дополнительные примеры инфиксной, префиксной и постфиксной записи

Инфиксное выражение	Префиксное выражение	Постфиксное выражение
A * B + C * D	+ * A B * C D	A B * C D * +
A + B + C + D	+ + + A B C D	A B + C + D +

Преобразование инфиксного выражения в префиксное и постфиксное

До сих пор мы использовали специальные методы для преобразования между инфиксными выражениями и эквивалентными им префиксной и постфикской записями. Как вы можете ожидать, существуют алгоритмические способы выполнения таких преобразований, позволяющие корректно трансформировать любое выражение любой сложности.

Первой из рассматриваемых нами техник будет использование идеи полной расстановки скобок в выражении, рассмотренной нами ранее. Напомним, что A + B * C можно записать как (A + (B * C)), чтобы явно показать приоритет умножения перед сложением. Однако, при более близком рассмотрении вы увидите, что каждая пара скобок также отмечает начало и конец пары операндов с соответствующим оператором по середине.

Взгляните на правую скобку в подвыражении (B * C) выше. Если мы передвинем символ умножения с его позиции и удалим соответствующую левую скобку, получив B C *, то произойдёт конвертирование подвыражение в постфиксную нотацию. Если оператор сложения тоже передвинуть к соответствующей правой скобке и удалить связанную с ним левую скобку, то результатом станет полностью постфиксное выражение (см. *рисунок 6*).

Рисунок 6: Перемещение операторов вправо для постфиксной записи

Если мы сделаем тоже самое, но вместо передвижения символа на позицию к правой скобке, сдвинем его к левой, то получим префиксную нотацию (см. *рисунок 7*). Позиция пары скобок на самом деле является ключом к окончательной позиции заключённого между ними оператора.

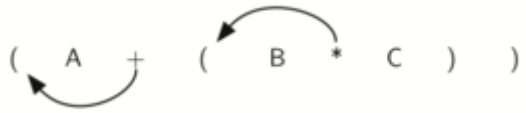


Рисунок 7: Перемещение операторов влево для префиксной записи.

Таким образом, при преобразовании выражения (неважно, насколько сложного) в префиксную или постфиксную запись для установления порядка выполнения операций используется полная расстановка скобок. Затем находящийся внутри них оператор передвигается на крайнюю левую или крайнюю правую позицию - в зависимости от того, префиксную или постфиксную запись вы хотите получить.

Вот более сложное выражение: (A + B) * C - (D - E) * (F + G). *Рисунок 8* демонстрирует его преобразование в постфиксный и префиксный виды.

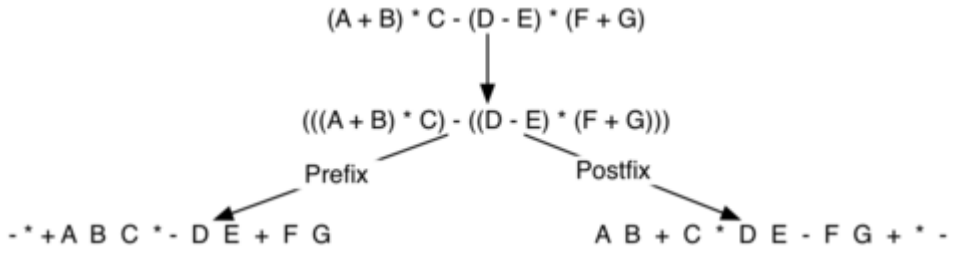


Рисунок 8: Преобразование сложного выражения к префиксной и постфиксной записи.

Обобщённое преобразование из инфиксного в постфиксный вид

Нам необходимо разработать алгоритм преобразования любого инфиксного выражения в постфиксное. Для этого посмотрим ближе на сам процесс конвертирования.

Рассмотрим ещё раз выражение A + B * C. Как было показано выше, его постфиксным эквивалентом является A B C * +. Мы уже отмечали, что операнды A, B и C остаются на своих местах, а местоположение меняют только операторы. Ещё раз взглянем на операторы в инфиксном выражении. Первым при проходе слева направо нам попадётся +. Однако, в постфиксном выражении + находится в конце, так как следующий оператор, *, имеет приоритет над сложением. Порядок операторов в первоначальном выражении обратен результирующему постфиксному выражению.

В процессе обработки выражения операторы должны где-то храниться, пока не найден их соответствующий правый операнд. Также порядок этих сохраняемых операторов может быть обратным (из-за их приоритета), как в данном примере со сложением и умножением. Поскольку оператор сложения, появляющийся перед оператором умножения, имеет более низкий приоритет, то он должен появиться после использования последнего. Из-за такого обратного порядка имеет смысл рассмотреть использование стека для хранения операторов до тех пор, пока они не понадобятся.

Что насчёт (A + B) * C? Напомним его постфиксный эквивалент: A B + C *. Повторимся, что обрабатывая это инфиксное выражение слева направо, первым мы встретим +. В этом случае, когда мы увидим *, + уже будет помещён в результирующее выражение, поскольку имеет преимущество над * в силу использования скобок. Теперь можно приступить к рассмотрению работы алгоритма преобразования. Когда мы видим левую скобку, то сохраняем её как знак, что должен будет появиться другой оператор с высоким приоритетом. Он будет ожидать, пока не появится соответствующая правая скобка, чтобы отметить его местоположение (вспомните технику полной расстановки скобок). После появления правой скобки оператор выталкивается из стека.

Поскольку мы сканируем инфиксное выражение слева направо, то для хранения операторов будем использовать стек. Это предоставит нам обратный порядок, который был отмечен в первом примере. На вершине стека всегда будет последний сохранённый оператор. Когда бы мы не прочитали новый оператор, мы должны сравнить его по приоритету с операторами в стеке (если таковые имеются).

Предположим, что инфиксное выражение есть строка токенов, разделённых пробелами. Токенами операторов являются *, /, + и - вместе с правой и левой скобками, (и). Токены операндов - это однобуквенные идентификаторы A, B, C и так далее. Следующая последовательность шагов даст строку токенов в постфиксном порядке.

Создать пустой стек с названием opstack для хранения операторов. Создать пустой список для вывода.

Преобразовать инфиксную строку в список, используя строковый метод split.

Сканировать список токенов слева направо.

Если токен является операндом, то добавить его в конец выходного списка.

Если токен является левой скобкой, положить его в opstack.

Если токен является правой скобкой, то выталкивать элементы из opstack пока не будет найдена соответствующая левая скобка. Каждый оператор добавлять в конец выходного списка.

Если токен является оператором *, /, + или -, поместить его в opstack. Однако, перед этим вытолкнуть любой из операторов, уже находящихся в opstack, если он имеет больший или равный приоритет, и добавить его в результирующий список.

#. Когда входное выражение будет полностью обработано, проверить opstack. Любые операторы, всё ещё находящиеся в нём, следует вытолкнуть и добавить в конец итогового списка.

Рисунок 9 демонстрирует алгоритм преобразования, работающий над выражением A * B + C * D. Заметьте, что первый оператор * удаляется до того, как мы встречаем оператор +. Также + остаётся в стеке, когда появляется второй *, поскольку умножение имеет приоритет перед сложением. В конце инфиксного выражения из стека дважды происходит выталкивание, удаляя оба оператора и помещая + как последний элемент в результирующее постфиксное выражение.

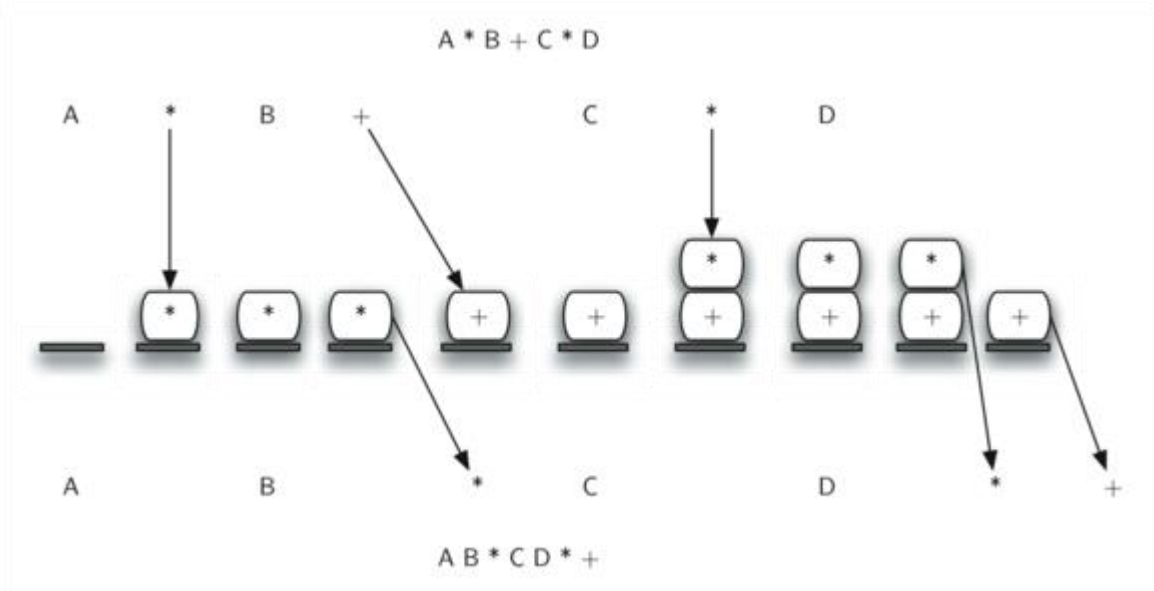


Рисунок 9: Преобразование A * B + C * D в постфиксную запись

Чтобы закодировать алгоритм на Python, мы будем использовать словарь под именем prec для хранения значений приоритета операторов. Он связывает каждый оператор с целым числом, которые можно сравнивать с числами других операторов, как уровень приоритетности (для этого мы произвольно выбрали целые числа 3, 2 и 1). Левая скобка получит самое низкое значение. Таким образом, любой сравниваемый с ней оператор будет иметь приоритет выше и располагаться над ней. Строка 15 определяет, что операнды могут быть любыми символами в верхнем регистре или цифрами. Полная функция преобразования показана в ActiveCode 8.

4.3.3 Американский стандартный код ASCII, Unicode

Кодовая страница (англ. *code page*) — таблица, сопоставляющая каждому значению байта некоторый символ (или его отсутствие). Обычно код символа имеет размер 8 бит, так что кодовая страница может содержать максимум 256 символов, из чего вытекает резкая недостаточность всякой 8-битной кодовой страницы для представления многоязычных текстов. К тому же часть символов используется как управляющие, из-за чего число печатных символов редко превышает 223.

В интерфейсе современных версий операционной системы Microsoft Windows имеется два способа представления текста:

в форме традиционных 8-битных кодовых страниц

в виде UTF-16.

В любом случае все символы кодируются одинаковым числом бит.

В других операционных системах может использоваться неравномерный код UTF-8.

Таблица ASCII

ASCII (англ. *American Standard Code for Information Interchange*) — американский стандартный код для обмена информацией.

ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов. Изначально разработанная как 7-битная, с широким распространением 8-битного байта ASCII стала восприниматься как половина 8-битной. В компьютерах обычно используют расширения ASCII с задействованной второй половиной байта.

Вариант ASCII без национальных символов называется US-ASCII, или «International Reference Version».

Таблицу соответствия кодов и символов можно посмотреть на официальном сайте.

Для кодирования текстов на русском языке (т.е. букв кириллицы) наиболее широко применяются следующие кодовые страницы:

Windows-1251, она же Microsoft code page 1251 (CP1251) — в системах Windows;

Семейство кодовых страниц KOI-8;

Альтернативная кодировка, она же IBM code page 866 — в системах DOS, а также в текстовых окнах Microsoft Windows;

MacCyrillic — на компьютерах Macintosh.

Использование различных кодовых страниц создаёт много неудобств, как для пользователей, так и для программистов. При попытке прочесть текстовый файл при помощи кодовой страницы, несовместимой с той, в которой он был создан, возникают крокозябры. В последние годы получил широкое распространение Unicode как альтернатива традиционным кодовым страницам.

Unicode

Юникод (англ. *Unicode*) — стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков. Стандарт предложен в 1991 году некоммерческой организацией «Консорциум Юникода» (англ. Unicode Consortium, Unicode Inc.). Применение этого стандарта позволяет закодировать очень большое число символов из разных письменностей: в документах Unicode могут соседствовать китайские иероглифы, математические символы, буквы греческого алфавита, латиницы и кириллицы, при этом становится ненужным переключение кодовых страниц.

UTF-16 (англ. *Unicode Transformation Format*) в информатике — один из способов кодирования символов из Unicode в виде последовательности 16-битных слов. В кодировке UTF-16 под любой символ уходит ровно 2 байта.

Есть кодировки и с переменным количеством бит на символ, например, кодировка UTF-8.

РЕКОМЕНДУЕМЫЕ Разница между IPV4 и IPV6 (с таблицей)

Американский стандартный код для обмена информацией или ASCII кодирует 128 символов преимущественно английского языка, которые используются в современных компьютерах и программировании.

ASCII использует 7 бит данных для кодирования любого символа и, следовательно, занимает меньше места. ASCII широко использовался для кодирования символов во всемирной паутине и до сих пор используется в современных компьютерных программах, таких как HTML.

ASCII кодирует любой текст путем преобразования текста в числа, потому что набор чисел легче хранить в памяти компьютера, чем алфавиты как язык. В широком смысле сам этот процесс называется кодированием.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Основные различия между Unicode и ASCII

Unicode - это стандарт ИТ, который кодирует, представляет и обрабатывает текст на компьютерах, тогда как ASCII - это стандарт, кодирующий текст (преимущественно английский) для электронных коммуникаций.

Юникод - это аббревиатура универсального набора символов, тогда как ASCII означает Американский стандартный код для обмена информацией.

Оба термина отличаются друг от друга в контексте функции. Unicode - это стандарт кодирования, который кодирует большое количество символов, таких как тексты и алфавиты из других языков (даже двунаправленные тексты), символы, исторические сценарии, тогда как ASCII кодирует алфавиты английского языка, буквы верхнего и нижнего регистра, символы и т. Д. .

Unicode использовал 8-битный, 16-битный или 32-битный для кодирования большого количества символов, тогда как ASCII использует 7-

битный для кодирования любого символа, потому что он состоит всего из 128 символов.

Больше места занимает Unicode, потому что он является расширенным набором ASCII, тогда как ASCII требует меньше места.

Юникод или универсальный набор символов - это стандарт кодирования, который кодирует, представляет и обрабатывает тексты для телекоммуникационных услуг и другого оборудования, тогда как ASCII или Американский стандартный код для обмена информацией - это стандартный код, который используется для кодирования в электронной связи.

Unicode охватывает кодирование текстов на разных языках (даже с двунаправленными сценариями, такими как иврит и арабский), символов, математических и исторических сценариев и т. Д., Тогда как ASCII охватывает кодирование символов английского языка, которое включает заглавную букву (AZ) , строчные буквы (az), цифры (0–9) и символы, например знаки препинания.

Unicode использует три типа кодирования, а именно 8-битную, 16-битную и 32-битную, тогда как ASCII работает, используя 7-битную кодировку для представления любого символа. Следовательно, Unicode также является расширенным набором ASCII и занимает больше места, чем он.

5 Лекция № 9. Представление числовой информации в цифровых устройствах

Цель лекции: изучить понятия и типы системы счисления, изучить как происходит преобразование чисел.

Содержание лекции: Понятие системы счисления. Двоичная система счисления. Преобразование чисел из одной системы счисления в другую.

5.1Понятие системы счисления

Система счисления -это совокупность правил и приемов записи чисел с помощью набора цифровых знаков. Количество цифр, необходимых для записи числа в системе, называют основанием системы счисления. Основание системы записывается в справа числа в нижнем индексе: 5_{10}; 1110110_2; $AF178_{16}$ и т. д.

Различают два типа систем счисления:

– позиционные, когда значение каждой цифры числа определяется ее позицией в записи числа;

– непозиционные, когда значение цифры в числе не зависит от ее места в записи числа.

Примером непозиционной системы счисления является римская: числа IX, IV, XV и т.д. Примером позиционной системы счисления является десятичная система, используемая повседневно.

Любое целое число в позиционной системе можно записать в форме многочлена:

$$X_S = \{A_n A_{n-1} A_{n-2} ... A_2 A_1\}_S = A_n \cdot S^{n-1} + A_{n-1} \cdot S^{n-2} + A_{n-2} \cdot S^{n-3} + ... + A_2 \cdot S^1 + A_1 \cdot S^0$$

где S - основание системы счисления;

A_n - цифры числа, записанного в данной системе счисления;

n - количество разрядов числа.

Пример. Число 6293_{10} запишется в форме многочлена следующим образом:

$$6293_{10} = 6 \cdot 10^3 + 2 \cdot 10^2 + 9 \cdot 10^1 + 3 \cdot 10^0$$

Позиционная система счисления подразделяется на:

– Однородные
– Смешанные

Однородная система — для всех разрядов (позиций) числа набор допустимых символов (цифр) одинаков. В качестве примера возьмем упоминавшуюся ранее 10-ю систему. При записи числа в однородной 10-й системе вы можете использовать в каждом разряде исключительно одну цифру от 0 до 9, таким образом, допускается число 450 (1-й разряд — 0, 2-й — 5, 3-й — 4), а 4F5 — нет, поскольку символ F не входит в набор цифр от 0 до 9.

Смешанная система — в каждом разряде (позиции) числа набор допустимых символов (цифр) может отличаться от наборов других разрядов. Яркий пример — система измерения времени. В разряде секунд и минут возможно 60 различных символов (от «00» до «59»), в разряде часов – 24 разных символа (от «00» до «23»), в разряде суток – 365 и т. д.

5.2 Двоичная система счисления

Эта система, в основном, используется в вычислительной технике. Почему не стали использовать привычную нам 10-ю? Первую вычислительную машину создал Блез Паскаль, использовавший в ней десятичную систему, которая оказалась неудобной в современных электронных машинах, поскольку требовалось производство устройств, способных работать в 10 состояниях, что увеличивало их цену и итоговые размеры машины. Этих недостатков лишены элементы, работающие в 2-ой системе. Тем не менее, рассматриваемая система была создана за долго до изобретения вычислительных машин и уходит “корнями” в цивилизацию Инков, где использовались кипу — сложные верёвочные сплетения и узелки.

Двоичная позиционная система счисления имеет основание 2 и использует для записи числа 2 символа (цифры): 0 и 1. В каждом разряде допустима только одна цифра — либо 0, либо 1.

Примером может служить число 101. Оно аналогично числу 5 в десятичной системе счисления. Для того, чтобы перевести из 2-й в 10-ю необходимо умножить каждую цифру двоичного числа на основание “2”, возведенное в степень, равную разряду. Таким образом, число $101_2 = 1*2^2 + 0*2^1 + 1*2^0 = 4+0+1 = 5_{10}$.

Хорошо, для машин 2-я система счисления удобнее, но мы ведь часто видим, используем на компьютере числа в 10-й системе. Как же тогда машина определяет какую цифру вводит пользователь? Как переводит число из одной системы в другую, ведь в её распоряжении всего 2 символа — 0 и 1?

Чтобы компьютер мог работать с двоичными числами (кодами), необходимо чтобы они где-то хранились. Для хранения каждой отдельной цифры применяется триггер, представляющий собой электронную схему. Он может находится в 2-х состояниях, одно из которых соответствует нулю, другое — единице. Для запоминания отдельного числа используется регистр — группа триггеров, число которых соответствует количеству разрядов в двоичном числе. А совокупность регистров — это оперативная память. Число, содержащееся в регистре — машинное слово. Арифметические и логические операции со словами осуществляет арифметико-логическое устройство (АЛУ). Для упрощения доступа к регистрам их нумеруют. Номер называется адресом регистра. Например, если необходимо сложить 2 числа — достаточно указать номера ячеек (регистров), в которых они находятся, а не сами числа. Адреса записываются в 8- и 16-ричной системах (о них будет рассказано ниже), поскольку переход от них к двоичной системе и обратно осуществляется достаточно просто. Для перевода из 2-й в 8-ю число необходимо разбить на группы по 3 разряда справа налево, а для перехода к 16-ой — по 4. Если в крайней левой группе цифр не достает разрядов, то они заполняются слева нулями, которые называются ведущими. В качестве примера возьмем число 101100_2. В восьмеричной — это $101\ 100 = 54_8$, а в шестнадцатеричной — $0010\ 1100 = 2C_{16}$. Отлично, но почему на экране мы видим десятичные числа и буквы? При нажатии на клавишу в компьютер передаётся определённая последовательность электрических импульсов, причём каждому символу соответствует своя последовательность электрических импульсов (нулей и единиц). Программа драйвер клавиатуры и экрана обращается к кодовой таблице символов (например, Unicode, позволяющая закодировать 65536 символов), определяет какому символу соответствует полученный код и отображает его на экране. Таким образом, тексты и числа хранятся в памяти компьютера в двоичном коде, а программным способом преобразуются в изображения на экране.

5.3 Преобразование чисел из одной системы счисления в другую

5.3.1 Преобразование в десятичную систему счисления.

Имеется число $a_1a_2a_3$ в системе счисления с основанием b. Для перевода в 10-ю систему необходимо каждый разряд числа умножить на b^n, где n — номер разряда. Таким образом, $(a_1a_2a_3)_b = (a_1*b^2 + a_2*b^1 + a_3*b^0)_{10}$.

Пример: $101_2 = 1*2^2 + 0*2^1 + 1*2^0 = 4+0+1 = 5_{10}$

5.3.2 Преобразование из десятичной системы счисления в другие.

Целая часть:

1. Последовательно делим целую часть десятичного числа на основание системы, в которую переводим, пока десятичное число не станет равно нулю.

2. Полученные при делении остатки являются цифрами искомого числа. Число в новой системе записывают, начиная с последнего остатка.

Дробная часть:

1. Дробную часть десятичного числа умножаем на основание системы, в которую требуется перевести. Отделяем целую часть. Продолжаем умножать дробную часть на основание новой системы, пока она не станет равной 0.

2. Число в новой системе составляют целые части результатов умножения в порядке, соответствующем их получению.

Пример: переведем 15_{10} в восьмеричную:

15\8 = 1, остаток 7

1\8 = 0, остаток 1

Записав все остатки снизу вверх, получаем итоговое число 17. Следовательно, $15_{10} = 17_8$.

5.3.3 Преобразование из двоичной в восьмеричную и шестнадцатеричную системы.

Для перевода в восьмеричную — разбиваем двоичное число на группы по 3 цифры справа налево, а недостающие крайние разряды заполняем ведущими нулями. Далее преобразуем каждую группу, умножая последовательно разряды на 2^n, где n — номер разряда.

В качестве примера возьмем число 1001_2: 1001_2 = 001 001 = $(0*2^2 + 0*2^1 + 1*2^0)\ (0*2^2 + 0*2^1 + 1*2^0) = (0+0+1)\ (0+0+1) = 11_8$

Для перевода в шестнадцатеричную — разбиваем двоичное число на группы по 4 цифры справа налево, затем — аналогично преобразованию из 2-й в 8-ю.

5.3.4 Преобразование из восьмеричной и шестнадцатеричной систем в двоичную.

Перевод из восьмеричной в двоичную — преобразуем каждый разряд восьмеричного числа в двоичное 3-х разрядное число делением на 2 (более подробно о делении см. выше пункт “Преобразование из десятичной системы

счисления в другие”), недостающие крайние разряды заполним ведущими нулями.

Для примера рассмотрим число 45_8: 45 = (100) (101) = 100101_2

Перевод из 16-ой в 2-ю — преобразуем каждый разряд шестнадцатеричного числа в двоичное 4-х разрядное число делением на 2, недостающие крайние разряды заполняем ведущими нулями.

Преобразование дробной части любой системы счисления в десятичную

Преобразование осуществляется также, как и для целых частей, за исключением того, что цифры числа умножаются на основание в степени “-n”, где n начинается от 1.

Пример: $101{,}011_2 = (1*2^2 + 0*2^1 + 1*2^0), (0*2^{-1} + 1*2^{-2} + 1*2^{-3}) = (5), (0 + 0{,}25 + 0{,}125) = 5{,}375_{10}$

5.3.5 Преобразование дробной части двоичной системы в 8- и 16-ую.

Перевод дробной части осуществляется также, как и для целых частей числа, за тем лишь исключением, что разбивка на группы по 3 и 4 цифры идёт вправо от десятичной запятой, недостающие разряды дополняются нулями справа.

Пример: $1001{,}01_2$ = 001 001, 010 = $(0*2^2 + 0*2^1 + 1*2^0)$ $(0*2^2 + 0*2^1 + 1*2^0)$, $(0*2^2 + 1*2^1 + 0*2^0)$ = (0+0+1) (0+0+1), (0+2+0) = $11{,}2_8$

5.3.6 Преобразование дробной части десятичной системы в любую другую.

Для перевода дробной части числа в другие системы счисления нужно обратить целую часть в ноль и начать умножение получившегося числа на основание системы, в которую нужно перевести. Если в результате умножения будут снова появляться целые части, их нужно повторно обращать в ноль, предварительно запомнив (записав) значение получившейся целой части. Операция заканчивается, когда дробная часть полностью обратится в нуль.

Для примера переведем $10{,}625_{10}$ в двоичную систему:
0,625*2=1,25
0,250*2=0,5
0,5*2=1,0

Записав все остатки сверху вниз, получаем $10{,}625_{10}$ = (1010), (101) = $1010{,}101_2$.

5.4 Формы представления чисел в цифровых устройствах

Числа в цифровых устройствах и микропроцессорах могут представляться в форме целых чисел, чисел с фиксированной запятой (точкой) и чисел с плавающей запятой (точкой).

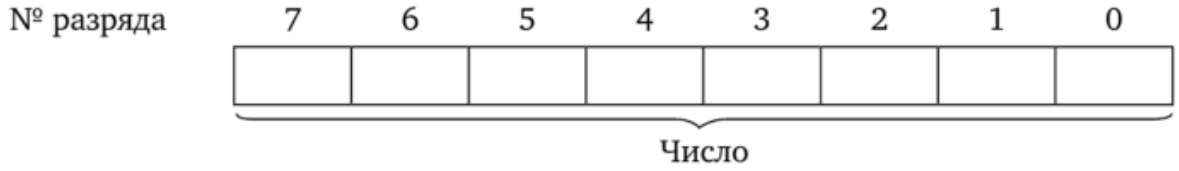
Представление целых чисел.

Целые числа могут быть как положительными, так и отрицательными. Для представления целых отрицательных чисел один из разрядов (старший) отводится под знак числа, остальные разряды отводятся под его модуль.

Обычно применяется следующий способ кодирования знака числа: «+2» обозначают цифрой 0 в знаковом разряде, «-» — цифрой 1 в знаковом разряде. Модуль числа занимает в разрядной сетке ее младшие разряды, свободные старшие разряды заполняются нулями.

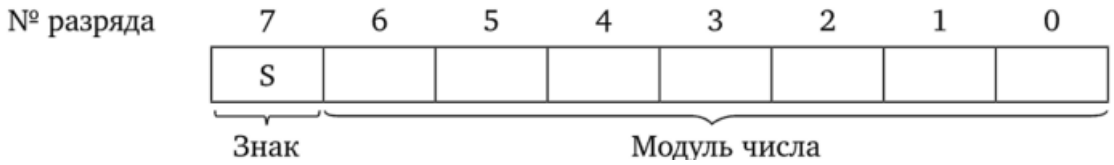
Форматы представления двоичных целых чисел.

1. 8-битные данные представляются как 8-разрядное целое двоичное число без знака:



Диапазон представления чисел: от 0 до 2п - 1= 28 - 1 = 256 - 1 = 255, т.е. от 0 до 255, где п = 8 разрядность сетки.

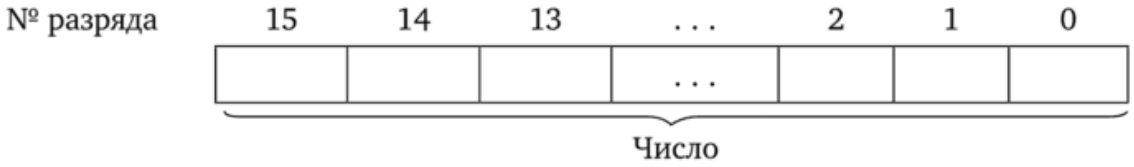
2. 8-битные данные представляются как 7-разрядное целое двоичное число со знаком:



S — знаковый бит (разряд), отрицательному числу соответствует 1, а положительному — 0. Отрицательные числа представляются в дополнительном коде (получение дополнительного кода будет рассмотрено ниже).

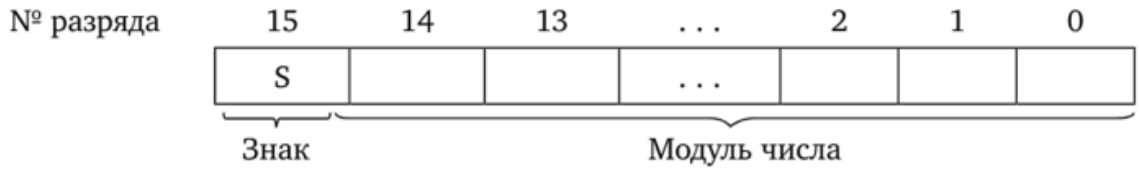
Диапазон представления чисел: от -2п_1до +2п~1 - 1 = от -27до +27 - 1, т.е. от -128 до +127.

3. 16-битные данные представляются как 16-разрядное целое двоичное число без знака:



Диапазон представления чисел: от 0 до 2п - 1 = 216 - 1 = 65 536 - 1 = = 65535, т.е. от 0 до 65 535.

4. 16-битные данные представляются как 15-разрядное целое двоичное число со знаком:



Диапазон представления чисел: от -2п_1до +2n_1 - 1, от -215 до +215 - 1, т.е. от -32 768 до +32 767.

Рассмотренную форму представления целых чисел можно определить как форму представления с фиксированной запятой, причем запятая фиксирована после младшего разряда.

В микропроцессорах, оперирующих 16-битными данными, целые числа представляются в 16-разрядном формате машинного слова или в 32-разрядном формате двойного слова.

6 Лекция № 10. Нормализованное представление чисел. Диапазон и точность представления чисел. Кодирование отрицательных чисел. Прямой, дополнительный и обратный коды

В компьютерной технике вещественными называются числа, имеющие дробную часть.

Дробные числа могут содержать большой набор цифр. Например: 0.0000345 или 10900000 (т.е очень большие или очень маленькие числа). Для удобства вещественные числа приводят к виду так называемого нормализованного представления числа. Заключается такое представление в том, что число записывается в виде произведения на основание системы счисления, возведенное в ту или иную степень. Например, предыдущие два числа в нормализованном виде будут выглядеть так: 0.345 * 10-4 и 0.109 * 108. Здесь числа 0.345 и 0.109 – **мантиссы** вещественных чисел, 10 – **основание системы счисления**, а -4 и 8 – **порядки**. При этом запятая (точка), разделяющая дробную и целую части ставится перед первой значащей цифрой (отличной от 0).

Нормализованная форма числа является наиболее удобной для представления дробных чисел в компьютере.

Понятно, что нормализированное представление используется не только для десятичной системы счисления. Вот примеры нормализованных записей дробных чисел в двоичной системе счисления:

101.11 = 0.10111 * 211
0.001 = 0.1 * 2-10

Здесь степени 11 и 10 – это двоичная форма десятичных чисел 3 и 2.

Нормализованная форма представления числа – это одна из форм множества вариантов экспоненциальной формы записи числа.

Пусть слово состоит из 2 байт, два слова – это 4 байта или 32 бита.

Нормализированное число одинарной точности, представленное в формате с плавающей точкой, записывается в память следующим образом: знак числа – в бите 15 первого слова (0 – для положительных и 1 – для отрицательных чисел); порядок размещается в битах 7-14 первого слова, а мантисса занимает остальные 23 бита в двух словах (с 0 по 6 бит первого слова и все биты второго слова). Нормализированное число двойной точности

записывается в четыре слова памяти и отличается от представления чисел с одинарной точностью только тем, что продолжение мантиссы размещается в следующих за первым словом трех последовательных словах памяти, а всего под мантиссу в этом случае отводится 55 бит.

Порядок числа, представленного в формате с плавающей точкой, изменяется в диапазоне от -128 до +127 и запоминается увеличенным на 128. Такой способ представления порядка называется *смещенным*.

Следует иметь в виду, что, хотя для мантиссы отведено 23 разряда для чисел одинарной точности и 55 разрядов – для чисел двойной точности, в операциях участвует 24 и 56 разрядов соответственно, т.к. старший разряд мантиссы нормализированного числа не хранится, т.е. имеет место так называемый скрытый разряд. Однако при аппаратном выполнении операций этот разряд автоматически восстанавливается и учитывается. Порядок числа также учитывает скрытый старший разряд мантиссы.

Нормализованная мантисса в двоичной системе счисления всегда представляется десятичным числом m, лежащим в диапазоне 0,5 <= m < 1.

Пример представления числа в формате с плавающей точкой:

0.110 = 0.000(1100)2 = 0.(1100)2*2-3 -310 = (-3 + 128)10 = 011111012.

Если мантисса представлена бесконечной периодической дробью, то последний учитываемый разряд мантиссы округляется.

-49.510 = -110001.1002 = -0.11000112*26 610 = (6 + 128)10 = 100001102.

При выполнении арифметических операций над числами, представленными в формате с плавающей точкой, надо отдельно выполнять их для порядков и мантисс. При алгебраическом сложении чисел надо сначала уравнять порядки слагаемых. При умножении порядки надо складывать, а мантиссы — перемножать. При делении из порядка делимого вычитают порядок делителя, а над мантиссами совершают обычную операцию деления. После выполнения операций, если это необходимо, проводят нормализацию результата, что влечет изменение порядков, т.к. каждый сдвиг на один разряд влево соответствует уменьшению порядка на единицу, а сдвиг вправо увеличению на единицу. Введение термина «плавающая точка» как раз и объясняется тем, что двоичный порядок, определяющий фактическое положение точки в изображении числа, корректируется после выполнения каждой арифметической операции, т.е. точка в изображении числа «плавает» (изменяется ее положение) по мере изменения данной величины. А в изображении чисел, представленных в формате с фиксированной точкой, она жестко зафиксирована в определенном месте.

Арифметические операции с числами, представленными в формате с плавающей точкой, намного сложнее таких же операций для чисел, представленных в формате с фиксированной точкой. Но зато плавающая точка позволяет производить операции масштабирования автоматически в самой

машине и избавляет от накопления абсолютной погрешности при вычислениях (хотя не избавляет от накопления относительной погрешности).

Числа с фиксированной запятой.

Форму числа с фиксированной запятой (ЧФЗ) применяют для представления целых чисел (запятая фиксируется после младшего разряда). Нумерация разрядов в слове выбирается справа налево (рисунок 3.1).

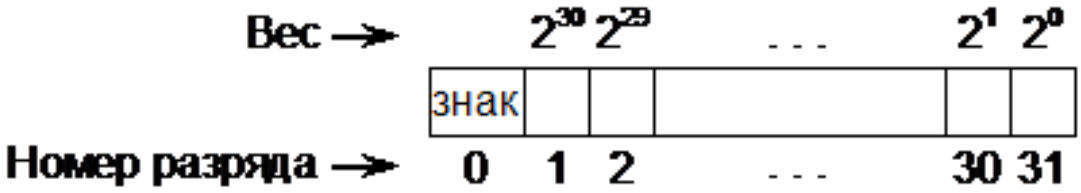


Рисунок 3.1 – Нумерация разрядов

Целые числа могут быть представлены как в формате слова (32 разряда), так и в формате полуслова (16 разрядов). В современных ЭВМ, как правило, слова содержат целое число байт, кратное степени двойки (1,2,4… байта).

Рассмотрим диапазон представления чисел с фиксированной запятой (только целых чисел). Если в разрядной сетке N разрядов, то под модуль числа отводится N-1 разряд (число со знаком). Самое большое по модулю число, записанное в такой сетке, имеет вид:

Номер разряда

$$|X|_{max} = \overset{N-2}{1}\,1\,1\,1\ldots\overset{0}{1}$$

$$+\qquad\qquad 1$$

$$\underset{\text{разряд}}{N-1} \rightarrow 1\,0\,0\,0\,0\ldots 0 = 2^{N-1}; \rightarrow |X|_{max} + 1 = 2^{N-1}$$

Студентам задание: привести свой пример

Следовательно, $|X|_{max} = 2^{N-1} - 1$

ПРИМЕР №1. Студентам задание: привести свой пример Дано: x=0,110001; y= -0,001001, сложить в обратном модифицированном коде. 1. y= -0,001001=y $_{обр}$= =1,110110

$[x]_{обр}=0,110001$ $\quad$ $[x]^{м}_{обр}=00,110001$ +

$[y]_{обр}=1,110110$ $\quad$ $[y]^{м}_{обр}=11,110110$

$100,100111$

$00,101000$

Дано: x=0,101001; y= -0,001101, сложить в дополнительном модифицированном коде.

y= -0,001101

прибавляем 0 к x и 1 к У и получаем то, что находится справа при сложении

$[x]_{доп}=0,101001$ $\quad$ $[x]^{м}_{доп}=00,101001$ +

$[y]_{доп}=1,110011$ $\quad$ $[y]^{м}_{доп}=11,110011$

$100,011100$

$00,011100$

В таблице 3.1 приведены диапазоны представления десятичных чисел 8, 16- и 32- разрядными двоичными числами.

таблица 3.1

N	8	16	32
X_{max}	127	32767	2147483647
X_{min}	-128	-32768	- 2147483648

Задание студентам-Описать, как получаются минимальные и махсимальные значения

1.1.1 Числа с плавающей запятой.ЧПЗ=числа, ЧФЗ. Числа с фиксированной запятой

Представление чисел в виде ЧПЗ позволяет избавиться от операции масштабирования при вычислениях, поскольку диапазон представляемых чисел существенно расширяется по сравнению с ЧФЗ. Однако в большинстве ЭВМ общего назначения, для целых чисел сохраняется возможность представления в виде ЧФЗ. Операции с ЧФЗ всегда выполняются за меньшее время, чем операции с ЧПЗ. В частности, к операциям с целыми числами сводятся операции над кодами адресов (операции индексной арифметики).

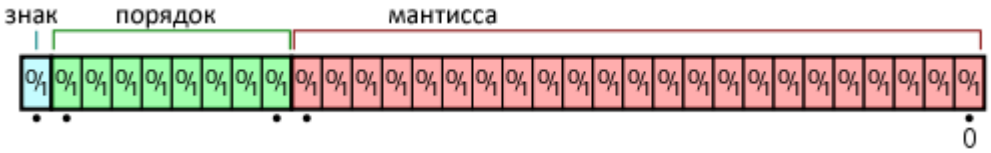
Число с плавающей запятой (или **число с плавающей точкой**) — экспоненциальная форма представления вещественных (действительных) чисел, в которой число хранится в виде мантиссы и порядка (показателя степени). При этом число с плавающей запятой имеет фиксированную относительную точность и изменяющуюся абсолютную. Используемое наиболее часто представление утверждено в стандарте IEEE 754. Реализация

математических операций с числами с плавающей запятой в вычислительных системах может быть как **аппаратная, так и программная.**

Основы

Множество целых чисел бесконечно, но мы всегда можем подобрать такое число бит, чтобы представить любое целое число, возникающее при решении конкретной задачи. Множество действительных чисел не только бесконечно, но еще и непрерывно, поэтому, сколько бы мы не взяли бит, мы неизбежно столкнемся с числами, которые не имеют точного представления. Числа с плавающей запятой — один из возможных способов представления действительных чисел, который является компромиссом между точностью и диапазоном принимаемых значений.

Число с плавающей запятой состоит из набора отдельных разрядов, условно разделенных на знак, порядок и мантиссу. Порядок и мантисса — целые числа, которые вместе со знаком дают представление числа с плавающей запятой в следующем виде:



Математически это записывается так: $\mathbf{(-1)^s \times M \times B^E}$, где s — знак, B-основание, E — порядок, а M — мантисса.

Основание определяет систему счисления разрядов. Математически доказано, что числа с плавающей запятой с базой B=2 (двоичное представление) наиболее устойчивы к ошибкам округления, поэтому на практике встречаются только базы 2 и, реже, 10. Для дальнейшего изложения будем всегда полагать B=2, и формула числа с плавающей запятой будет иметь вид:

$$\mathbf{(-1)^s \times M \times 2^E}$$

$$\mathbf{(-1)^s \times M \times B^E}$$ **(Сравним)**

Что такое мантисса и порядок? ***Мантисса*** **– это целое число фиксированной длины, которое представляет старшие разряды действительного числа**. Допустим наша мантисса состоит из трех бит (|M|=3). Возьмем, например, число «5», которое в двоичной системе будет равно $\mathbf{101_2}$**.** Старший бит соответствует $\mathbf{2^2=4}$, средний (который у нас равен нулю) $\mathbf{2^1=2}$, а младший $\mathbf{2^0=1}$. ***Порядок*** **– это степень базы (двойки) старшего разряда. В нашем случае E=2**. Такие числа удобно записывать в так называемом **стандартном виде, например «1.01e+2».** Сразу видно, что мантисса состоит из трех знаков, а порядок равен двум.

Допустим мы хотим получить дробное число, используя те же 3 бита мантиссы. Мы можем это сделать, если возьмем, скажем, E=1. Тогда наше число будет равно

1,01e+1 = $1\times2^{1}+0\times2^{0}+1\times2^{-1}=2+0,5=2,5$

Целые числа в ЭВМ могут быть представлены в виде:

прямого кода. Прямой код двоичного числа совпадает по изображению с записью самого числа.

обратного кода. Обратный код для положительного числа совпадает с прямым кодом. Для отрицательного числа все цифры числа заменяются на противоположные (1 на 0, 0 на 1), а в знаковый разряд заносится единица.

дополнительного кода. Дополнительный код положительного числа совпадает с прямым кодом. Для отрицательного числа дополнительный код образуется путем получения обратного кода и добавлением к младшему разряду единицы.

Прямой код числа кодирует только знаковую информацию и используется для хранения положительных и отрицательных чисел в ЭВМ. Прямой код двоичного числа совпадает по изображению с записью самого числа, но в знаковом разряде ставится 0, если число положительное и, 1 если число отрицательное.
Обратный и дополнительный коды используются для выполнения всех арифметических операций через операцию сложения. Следует помнить, что положительные числа в обратном и дополнительном коде совпадают с прямым кодом.
1) Прямой код числа (кодируется только знаковая информация), “+”=0; ”-”=1. Для прямого кода возможны два представления нуля, машинный положительный ноль, т.е. +0,110=0,110, машинный отрицательный ноль, т.е. -0,111=1,111.

ПРИМЕР ПЕРЕВОДА

x1=10101-[x1]пр=010101
x2=-11101-[x2]пр=111101
x3=0,101-[x3]пр=0,101
x4=-0,111-[x4]пр=1,111

2) Обратный код числа, используется для выполнения арифметических операций вычитания, умножения, деления, через сложение. Обратный код положительного числа совпадает с его прямым кодом, обратный код отрицательного числа формируется по правилам: в знаковом разряде записывается “1”; цифровые значения меняются на противоположные.

ПРИМЕР ПЕРЕВОДА

x1=10101-[x1]обр=010101
x2=-11101-[x2]обр=100010

x3=0,101-[x3]обр=0,101
x4=-0,111-[x4]обр=1,000

3) *Дополнительный код числа*, имеет такое же назначение, как и обратный код числа. Формируется по следующим правилам: положительные числа в дополнительном коде выглядят также как и в обратном и в прямом коде, т.е. не изменяются. Отрицательные числа кодируются следующим образом: к обратному коду отрицательного числа (к младшему разряду) добавляется 1, по правилу двоичной арифметики.

ПРИМЕР ПЕРЕВОДА

x1=10101-[x1]доп=010101
x2=-11101-[x2]обр=100010+1-[x2]доп=100011
x3=0,101-[x3]доп=0,101
x4=-0,111-[x4]обр=1,000+1-[x4]доп=1,001

Для выявления ошибок при выполнении арифметических операций используются также модифицированные коды: модифицированный прямой; модифицированный обратный; модифицированный дополнительный, для которых под код знака числа отводится два разряда, т.е. “+”=00; ”-”=11. Если в результате выполнения операции в знаковом разряде появляется комбинация 10 или 01 то для машины это признак ошибки, если 00 или 11 то результат верный.

Как определить, положительное или отрицательное число? Знак числа определяет старший бит: 0 - положительное число, 1 - отрицательное число. Например, для числа 1,001 сразу можно определить, что оно отрицательное (меньше нуля).

ПРИМЕР. Представить в дополнительном коде десятичные числа: *-4*.
Решение. Представим число в двоичном коде.
4 = 00001002
Инвертируем все разряды числа, а в знаковый разряд заносим 1.
Двоичное число 0000100 имеет обратный код 1,1111011
Добавляем к младшему разряду 1.
В 0-ом разряде возникло переполнение (1 + 1 = 10). Поэтому записываем 0, а 1 переносим на 1-й разряд.

7	6	5	4	3	2	1	0
						1	
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	1
							0

В 1-ом разряде возникло переполнение (1 + 1 = 10). Поэтому записываем 0, а 1 переносим на 2-й разряд.

7	6	5	4	3	2	1	0
					1	1	
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	1
						0	0

В итоге получаем:

7	6	5	4	3	2	1	0
					1	1	
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	1
1	1	1	1	1	1	0	0

Число -4 представляется в двоичном дополнительном коде как 11111100

7 Лекция 11 Эффективное кодирование

Цель лекции: изучить основы кодирования данных, виды кодирования данных.

Содержание лекции: Информационная избыточность. Систематический код, кодовое расстояние, корректирующая способность кода. Теорема Шеннона для каналов без шума. Код Шеннона - Фано. Посылочные корректирующие коды, обнаруживающие и исправляющие ошибки. Код с проверкой на четность. Коды Хэмминга. Циклический код. Линейно-групповые коды.

6.1 Информационная избыточность

Избыточность — термин из теории информации, означающий превышение количества информации, используемой для передачи или хранения сообщения, над его информационной энтропией. Для уменьшения избыточности применяется сжатие данных без потерь, в то же время контрольная сумма применяется для внесения дополнительной избыточности в поток, что позволяет производить исправление ошибок при передаче информации по каналам, вносящим искажения (спутниковая трансляция, беспроводная передача и т. д.).

Мера избыточности между двумя случайными величинами - взаимная информация.

При телевизионной передаче различают несколько видов избыточности изображений, на сокращении (сжатие) которых основываются методы и устройства кодирования изображений:

- **пространственная** — присутствует, если значения элементов изображения (пикселей) коррелированы в пределах некоторой части изображения (видеокадра), т.е. можно передать значение одного пикселя, а значения других предсказать;

- **временная** — корреляция между одноименными пикселями наблюдается в последовательности нескольких видеокадров, т.е. техника предсказания и передачи движения может быть распространена на несколько кадров;

Иногда оба эти вида избыточности называют **статистической избыточностью.** Использование корреляционных связей позволяет не передавать многократно одни и те же данные и сокращать за счет этого объем сообщения. При этом устранение статистической избыточности в ТВ сигнале может происходить без потери информации, и исходные данные будут полностью восстановлены.

- **психовизуальная** (физиологическая) — характеризуется тем, что глаз не способен различать очень мелкие детали и высокочастотные цветовые переходы. Использовать физиологическую избыточность - значит, не передавать ту информацию, которая не будет воспринята нашим зрением. Как следствие, устранение физиологической избыточности связано с частичной потерей информации, но той информации, которая не воспринимается зрением, и ее потеря не повлияет на визуальное качество изображения.

Далее рассмотрим наиболее важные особенности восприятия изображений.

6.2 Систематический код

Систематический код – это групповой *n*-значный код в котором *k* символов является информационными, а *n-k* являются проверочными, избыточными. Они занимают одни и те же позиции. Задание СК производится с помощью производящей матрицы А и проверочной матрицы Н.

Производящая матрица позволяет получить все комбинации кода путем суммирования по модулю 2.

$$
\begin{array}{c}
 \\ k \\ \\ \\ \\
\end{array}
\left\{
\begin{array}{ccccc|cccc}
\multicolumn{5}{c|}{\overbrace{\qquad\qquad\qquad\qquad}^{k}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{n-k}} \\
1 & 0 & 0 & \dots & 0 & a_{11} & a_{12} & \dots & a_{1,n-k} \\
0 & 1 & 0 & \dots & 0 & a_{21} & a_{22} & \dots & a_{2,n-k} \\
0 & 0 & 1 & \dots & 0 & a_{31} & a_{32} & \dots & a_{3,n-k} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & \dots & 1 & a_{k1} & a_{k2} & \dots & a_{k,n-k}
\end{array}
\right\}
$$

(n-k) разрядные кодовые комбинации, в которых количество единиц не менее чем d_{min}-1, а сумма любых двух строк по модулю 2 должна содержать не менее чем d_{min}-2«единиц». Желательно при этом, чтобы единицы чередовались с нулями, чтобы серии единиц были короткие. 2^k – число кодовых комбинации.

Структура производящей матрицы позволяет строить систематический код, у которого первые *k*-позиции будут информационными, а оставшиеся (*n-k)* проверочные, при этом общее число разрешенных кодовых комбинаций равно 2^k и все различные кодовые комбинации могут быть получены в результате линейной операции суммы по |2| любых двух, трех и более строк производящей матрицы.

Построение проверочной матрицы:

Берем единичную матрицу размерности *n-k* и слева к ней приписывают транспонированную дополнительную матрицу D, которую использовали для построения производящей матрицы А.

$$D = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,n-k} \\ a_{21} & a_{22} & \dots & a_{2,n-k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{k,n-k} \end{bmatrix}, \quad D^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{k1} \\ a_{12} & a_{22} & \dots & a_{k2} \\ \dots & \dots & \dots & \dots \\ a_{1,n-k} & a_{2,n-k} & \dots & a_{k,n-k} \end{bmatrix}$$

$$H = \left| \underbrace{\begin{bmatrix} a_{11} & a_{21} & \dots & a_{k1} \\ a_{12} & a_{22} & \dots & a_{k2} \\ \dots & \dots & \dots & \dots \\ a_{1,n-k} & a_{2,n-k} & \dots & a_{k,n-k} \end{bmatrix}}_{k} \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}}_{(n-k)} \right|_{(n-k)*n}$$

Таким образом, проверочная матрица *H* позволяет построить алгоритм кодирования и декодирования систематического кода, исходя из того, что единица в каждой строке соответствует тем разрядам, сумма которых по модулю 2 должна быть равна нулю.

Из проверочных уравнений выделяют проверочные разряды, количество которых равно *n-k*. Проверочные разряды встречаются в каждом из проверочных уравнений только по одному разу.

6.2.1 Процесс приема-передачи с корректировкой ошибок.

Формируется информационная часть кодовой комбинации. На ее основе строится проверочная часть с помощью проверочных уравнений.

На приемной стороне полученные разряды подставляются в проверочные уравнения. Если кодовая комбинация передана без ошибки, то все проверочные уравнения примут нулевые значения.

Если какие либо уравнения примут ненулевое значение, тогда формируют опознаватель для обнаружения и исправления ошибок.

$$\overbrace{Y_1 \ldots Y_i \ldots Y_{n-k}}^{R}$$

Если вектор-опознаватель не равен нулю, тогда производится обнаружение ошибочных разрядов следующим образом:

доказано, что вектор-опознаватель совпадает со значением «правой части» производящей матрицы, если кратность ошибки равна 1.

Если кратность ошибки больше 1, тогда ошибочные разряды могут быть обнаружены следующим образом: формируется сумма по модулю 2 из «правых частей» строк проверочной матрицы. Количество различных строк, которые включаются в сумму по модулю 2 определяют кратность ошибки.

6.3 Кодовое расстояние

Кодовым расстоянием *d* называется минимальное количество разрядов в которых одна кодовая комбинация отличается от другой кодовой комбинации.

Для конкретного кода кодовым расстоянием данного кода называется минимальное число элементов, которыми одна кодовая комбинация данного кода отличается от другой кодовой комбинации того же кода.

Иногда кодовое расстояние называется ***Хемминговым расстоянием*** (по имени Ричарда Хемминга, основателя помехоустойчивых кодов) .

У простого двоичного кода d=1. Каким должно быть кодовое расстояние у кодов, обнаруживающих ошибки, и у кодов, исправляющих ошибки?

В общем случае для обеспечения возможности исправления всех ошибок кратности до t включительно при декодировании по методу максимального правдоподобия каждая из ошибок должна приводить к запрещенной кодовой комбинации, входящее в группу таких комбинаций , соответствующих исходной кодовой комбинации.

Пусть имеется n-разрядный двоичный код. Выберем 2 кодовые комбинации A_i и A_j, которые будем считать разрешенными (рисунок 7.1). Каждой разрешенной кодовой комбинации соответствует свое подмножество запрещенных кодовых комбинаций с одиночными ошибками. Число их равно C_n^1, а кодовое расстояние относительно исходной кодовой комбинации d=1. Графически это представляется окружностью с радиусом d=1.

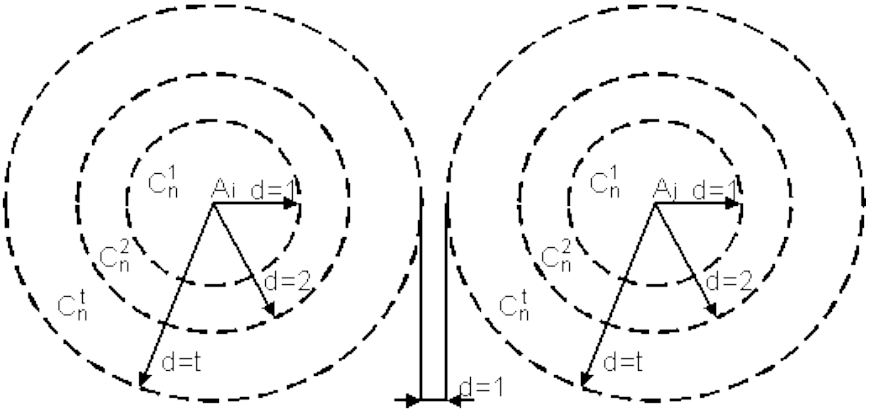


Рисунок 6.1 – Определение кодового расстояния

Аналогично, подмножество запрещенных кодовых комбинаций с двойной ошибкой имеет кодовое расстояние относительно исходной d=2, а число их равно C_n^2. И так далее до ошибок кратности t включительно.

Для того чтобы при приеме восстановить именно A_i кодовую комбинацию необходимо, чтобы набор ее запрещенных кодовых комбинаций не пересекался с набором запрещенных кодовых комбинаций A_j.

Для исправления ошибок кратности t кодовое расстояние должно удовлетворять условию: $d \geq 2t+1$.

Для исправления ошибки кратности s кодовое расстояние должно быть: $d \geq s+1$.

Если код должен исправлять ошибки кратности t и обнаруживать ошибки кратности s, то кодовое расстояние должно быть не менее $d \geq s+1+t; s \geq t.$

Помехоустойчивые коды делятся на два больших класса:

1. Блоковые коды;
2. Непрерывные коды.

Для **блоковых кодов** к каждой исходной комбинации добавляется блок избыточных символов и получается новая комбинация (рисунок 6.2).

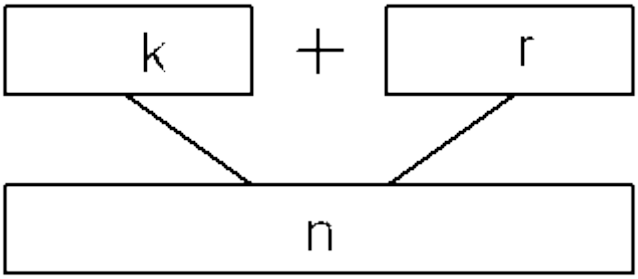


Рисунок 6.2 – Построение блокового кода

Различают разделимые и неразделимые блоковые коды. В разделимых блоковых кодах k символов являются информативными, а r - проверочными.

Такие коды обозначают как (n, k) – коды.

6.3 Теорема Шеннона для каналов без помех

Предельные возможности статистического кодирования раскрывается в ***теореме Шеннона*** для канала без шума, которая является одним из основных положений теории передачи информации. Эта теорема может быть сформулирована следующим образом (обозначения те же, что и в пункте 2.1).

Пусть источник сообщений имеет производительность $H®(U) = v_C \cdot H(U)$, а канал имеет пропускную способность $C = v_K \cdot \log M$.. Тогда можно закодировать сообщения на выходе источника таким образом, чтобы получить среднее число кодовых символов, приходящихся на элемент сообщения:

$$\eta = \frac{v_K}{v_C} = (H(U)/\log M) + \varepsilon \qquad (2.2),$$

где ε - сколь угодно мало (***прямая теорема***).

Получить меньшее значение η невозможно ***(обратная теорема)***.

Обратная часть теоремы, утверждающая, что невозможно получить значение

$$\eta = v_K/v_C < H(U)/\log M \qquad (2.3),$$

может быть доказана, если учесть, что неравенство (2.3) эквивалентно неравенству:

$$v_C \cdot H(U) > v_K \cdot \log M,$$

$$H®(U) > C.$$

Последнее неравенство не может быть выполнено т.к. рассматриваемое кодирование должно быть обратимым преобразованием (т.е. без потерь информации).

Энтропия в секунду на входе канала или производительность кодера не может превышать пропускную способность канала (см. пункт 1.6).

Докажем прямую теорему двумя различными способами. При этом источник сообщений будем полагать **источником без памяти**, имея в виду, что к нему со сколь угодно высокой степенью точности может быть сведен любой источник, если блоки элементов сообщения достаточно большой длинны K_1 рассматривать, как элементарные сообщения U_i с объемом алфавита N_u.

Первый способ доказательства состоит в рассмотрении множества сообщений из K символов создаваемых источником. Каждую из таких последовательностей содержащую K элементарных сообщений U_i выбираемых из алфавита объемом N_u будем считать сообщением:

$\propto = (U_{k-1}U_{k-2} \dots U_1 U_0)$.

Вероятность этих сообщений для источников без памяти:

$P(\propto) = P(U_{k-1}) \cdot P(U_{k-2}) \cdot \dots \cdot P(U_0)$.

При этом количество L отличных друг от друга сообщений α будет равно:

$L = N_u^k$.

При достаточно большой длине сообщения K все множество L возможных сообщений может быть разбито на 2 подмножества.

Одно из них (назовем его ***высоковероятным*** или ***типичным***) содержит K_1 наиболее вероятных сообщений, сумма вероятностей которых $1-\delta$ близка к единице,

второе содержит остальные $K_2 = L - K_1$ сообщений (***маловероятных*** или ***нетипичных***), суммарная вероятность которых δ близка к нулю (см. задача 1.10 пункт 2).

Увеличивая K можно сделать δ сколь угодно малым.

Сказанное следует из закона теории вероятностей, согласно которому при достаточно большом числе испытаний количество каждого из возможных исходов близко к своему математическому ожиданию (закон больших чисел).

В данном случае числом испытаний является число K элементов сообщений, исходом являются значения элементов u_i и математическое ожидание количества исходов U_i равно $K\,P(U_i)$ (см. задачу 1.10 пункт 1). Такие сообщения и составляют ***высоковероятное*** подмножество.

Вероятность того, что сообщения содержат другие числа элементов при $K\,P$ ничтожно мала.

Поскольку при отсутствии памяти у источника вероятность того или иного сочетания различных элементов сообщения зависит только от их количества, то все сообщения, принадлежащие к **высоковероятному** подмножеству, т.е. содержащие примерно одно и тоже количество разных элементов и отличающихся только расположением этих элементов в последовательности, имеют одну и туже вероятность, близкую к:

$P(0)^{K-P(0)} \cdot P(1)^{K-P(1)} \cdot \dots \cdot P(N_u - 1)^{K-P(N_u-1)} = P$.

Прологарифмируем полученное равенство:

$$\log P = \sum_{U=0}^{U_n-1} \log P(U)^{K\cdot P(U)} = \sum_{U=0}^{U_n-1} K \cdot P(U) \cdot \log P(U) = -K \underbrace{\sum_{U=0}^{U_n-1} P(U) \cdot \log(1/P(U))}_{H(U)} =$$

$$= -K \cdot H(U)$$

В силу равновероятности число сообщений в высоковероятном подмножестве:

$K_1 > 1/P$.

Учитывая полученное выражение для логарифма P, K_1 можно выразить через энтропию источника:

$$P = 2^{-K\cdot H(U)},\ K_1 = 1/P = 2^{K\cdot H(U)} \qquad (2.4)$$

Поскольку $L = N_u^K = 2^{K\cdot\log(N_u)}$:

$$\frac{K_1}{L} = 2^{K\cdot[H(U)-\log N_u]} = 2^{\underbrace{-\frac{K}{\log N_u}\left[1-\frac{H(U)}{\log N_u}\right]}_{\alpha}} = 2^{-\frac{K\mu}{\log N_u}} \qquad (2.5),$$

где μ - избыточность источника.

Анализ выражения (2.5) показывает, что при любой не нулевой избыточности источника μ число сообщений высоковероятного подмножества составляет сколь угодно малую часть от всех возможных сообщений, если длина сообщения K достаточно велика (сколь угодно малую вероятность имеет большая часть достаточно длинных сообщений). Это большинство и позволяет добиться близкого к минимуму числа η кодовых символов на элемент сообщения.

Применен следующий способ кодирования. **Высоковероятной** группе сообщений поставим в однозначное соответствие различные короткие (их мало) кодовые комбинации равномерного кода длинны n_1, оставив одну для использования в качестве разделительного сигнала. Для остальных $K_2 = L - K_1 > L$ сообщений, составляющих мало вероятное подмножество используем более длинные кодовые комбинации, содержащие n_2 символов, начинающихся с упомянутой выше разделительной комбинации длинны n_1 (для того, чтобы при декодировании можно разграничить принимаемые сообщения).

Длины кодов n_1 и n_2 определим из условия:

$$S \le m^n \qquad (2.6),$$

где S - число различных кодовых комбинаций равномерного кода,

m - объем алфавита кодовых символов,

n - число символов (длина кодовой комбинации).

Полагая в (2.6) $S = K_1 + 1 + \lambda,\ n = n_1$,

где λ - число, дополняющее $K_1 + 1$ до m^{n_1}, можно записать:

$$n_1 = \frac{\log(K_1 + 1 + \lambda)}{\log m} = \frac{\log K_1(1 + \frac{1+\lambda}{K_1})}{\log m} = \frac{\log K_1}{\log m} + \theta = \frac{K\cdot H(U)}{\log m} + \theta,$$

где:

$$\theta = \frac{\log(1 + \frac{1+\lambda}{K_1})}{\log m}.$$

Увеличивая число K можно сколь угодно уменьшить число θ по сравнению с тем числом, с которым оно суммируется (при KP и KP_1).

Рассуждая аналогично получим выражение для n_2 с учетом разделительной комбинации:

$$n_2 = n_1 + \frac{\log(L+\psi)}{\log m} = \frac{K \cdot H(U)}{\log m} + \theta + \frac{\log L(1+\frac{\psi}{L})}{\log m} = \frac{K \cdot H(U)}{\log m} + \theta + \frac{K \cdot \log N_u}{\log m} + \gamma$$

$L = N_u^k$,

(про ψ и γ можно сказать тоже самое, что про λ и θ).

Поскольку средняя длина кодовой комбинации при этом равна $\bar{n} = (1-\delta) \cdot n_1 + \delta \cdot n_2$, то среднее число кодовых символов приходящихся на один элемент сообщения равно:

$$\eta = \frac{\bar{n}}{K} = \frac{H(U)}{\log m} + \xi,$$

где:

$$\xi = \frac{\theta}{K} + \delta \cdot \frac{\gamma}{K} + \delta \cdot \frac{\log N_u}{\log m}.$$

ξ становится сколь угодно малым при больших K.

Это и доказывает прямую теорему.

В приведенном доказательстве мы полагали, что множество маловероятных последовательностей кодируется длинными кодовыми комбинациями (длину $n_2 + n_1$), что и делает возможным их однозначное ***декодирование***.

Однако, в принципе, всем сообщениям этой группы можно поставить в соответствие одну и туже разделительную комбинацию длиной n_1, и, определив по ней принадлежность принятого сообщения к маловероятной группе отбрасывать их как ошибочные.

Поскольку с ростом K вероятность нетипичных сообщений становится сколь угодно малой, то сколь угодно малой будет вероятность ошибки. Данное замечание не меняет существа приведенного доказательства, но позволяет назвать рассмотренный в нем вариант кодирования равномерным кодированием.

Код Шеннона - Фано

Алгоритм Шеннона — Фано — один из первых алгоритмов сжатия, который впервые сформулировали американские учёные Шеннон и Роберт Фано. Данный метод сжатия имеет большое сходство с алгоритмом Хаффмана, который появился на несколько лет позже и является логическим продолжением алгоритма Шеннона. Алгоритм использует коды переменной длины: часто встречающийся символ кодируется кодом меньшей длины, редко встречающийся — кодом большей длины. Коды Шеннона — Фано — префиксные, то есть никакое кодовое слово не является префиксом любого другого. Это свойство позволяет однозначно декодировать любую последовательность кодовых слов.

Кодирование Шеннона — Фано (англ. Shannon–Fano coding) — алгоритм префиксного неоднородного кодирования. Относится

к вероятностным методам сжатия (точнее, методам контекстного моделирования нулевого порядка). Подобно алгоритму Хаффмана, алгоритм Шеннона — Фано использует избыточность сообщения, заключённую в неоднородном распределении частот символов его (первичного) алфавита, то есть заменяет коды более частых символов короткими двоичными последовательностями, а коды более редких символов — более длинными двоичными последовательностями.

Алгоритм был независимо друг от друга разработан Шенноном (публикация «Математическая теория связи», 1948 год) и, позже, Фано (опубликовано как технический отчёт).

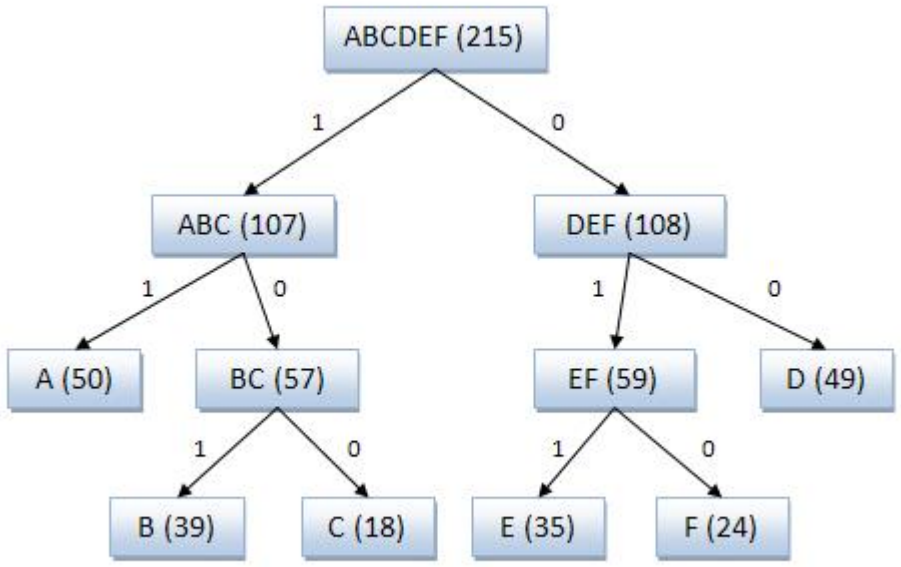
Код Шеннона — Фано строится с помощью дерева. Построение этого дерева начинается от корня. Всё множество кодируемых элементов соответствует корню дерева (вершине первого уровня). Оно разбивается на два подмножества с примерно одинаковыми суммарными вероятностями. Эти подмножества соответствуют двум вершинам второго уровня, которые соединяются с корнем. Далее каждое из этих подмножеств разбивается на два подмножества с примерно одинаковыми суммарными вероятностями. Им соответствуют вершины третьего уровня. Если подмножество содержит единственный элемент, то ему соответствует концевая вершина кодового дерева; такое подмножество разбиению не подлежит. Подобным образом поступаем до тех пор, пока не получим все концевые вершины. Ветви кодового дерева размечаем символами 1 и 0, как в случае кода Хаффмана.

При построении кода Шеннона — Фано разбиение множества элементов может быть произведено, вообще говоря, несколькими способами. Выбор разбиения на уровне n может ухудшить варианты разбиения на следующем уровне (n + 1) и привести к неоптимальности кода в целом. Другими словами, оптимальное поведение на каждом шаге пути ещё не гарантирует оптимальности всей совокупности действий. Поэтому код Шеннона — Фано не является оптимальным в общем смысле, хотя и дает оптимальные результаты при некоторых распределениях вероятностей. Для одного и того же распределения вероятностей можно построить, вообще говоря, несколько кодов Шеннона — Фано, и все они могут дать различные результаты. Если построить все возможные коды Шеннона — Фано для данного распределения вероятностей, то среди них будут находиться и все коды Хаффмана, то есть оптимальные коды.

Пример кодового дерева

Исходные символы:

- A (частота встречаемости 50)
- B (частота встречаемости 39)
- C (частота встречаемости 18)
- D (частота встречаемости 49)
- E (частота встречаемости 35)
- F (частота встречаемости 24)



Кодовое дерево

Полученный код: A — 11, B — 101, C — 100, D — 00, E — 011, F — 010.

Лекция 12 Посылочные корректирующие коды, обнаруживающие и исправляющие ошибки

Коды обнаружения ошибок

Методы помехозащищенного кодирования направленные только на обнаружение ошибки называют – коды обнаружения ошибок. Обнаружение ошибки позволяет, в некоторых ситуациях, перезапросить данные с источника и исправить ошибку. Ну или не использовать ошибочные данные. Поскольку коррекция ошибок в этих методах не требуется, то размер дополнительно передаваемой информации минимален.

Простейшим кодом обнаружения ошибок является «контроль четности». В этом случае, все сообщение разбивается на блоки бит равного размера, например, по 8-мь бит. И вычисляется сумма значений бит каждого из этих блоков. К блоку добавляется еще один бит: если сумма нечетная, то бит 1, иначе — бит 0. Т.е. получается блок длиной а один бит больше исходного — это «бит четности». При получении данных вычисление «бита четности» повторяется и результат сравнивается с переданным битом четности. Если вычисленное и переданное значение «бита четности» совпадают, то значит ошибки не было, иначе — ошибка передачи.

Легко сообразить, что искажение двух, четырех и любого четного количества бит в блоке не будет обнаружено.

Однако, в силу своей простоты, метод широко применяется в информационных каналах, где вероятность ошибки мала. Данный метод позволяет легко регулировать избыточность данных, выбирая размер блока.

Чем больше исходный блок — тем меньше относительный размер дополнительной информации и тем выше вероятность пропустить ошибку.

Другим примером кодов обнаружения ошибок является «циклическая сумма» (Cyclic Redundancy Checksum или CRC). Этот метод является неблочным, но может быть использован и в блочном варианте. Принцип обнаружения ошибки схож с кодом «контроля четности». Используя особый алгоритм суммируются значения всех битов данных или порции данных, в результате суммирования получается некое число (не один бит, а несколько), которое передается вместе с данными или порцией данных. Принимающая сторона вычисляет CRC повторно и сравнивает с переданным.

«Контроль четности» — частный случай CRC. Алгоритм CRC вы сможете изучить на практических занятиях.

Корректирующие коды — коды, служащие для обнаружения и исправления ошибок, возникающих при передаче информации под влиянием помех.

Для этого при передаче к исходной информации добавляют дополнительную информацию. Т.е. из исходного блока длиной n, делают блок длиной m, где $m > n$.

Приемник повторно вычисляет эту дополнительную информацию и сравнивает ее с переданной. На основании различий делается вывод о наличии или отсутствии ошибок. При обнаружении ошибок выполняется корректирующее действие.

Одним из самых простых, но неэффективных, способов коррекции является тройное мажорирование. Тройное мажорирование состоит в том, что данные (каждый бит или каждый блок битов) передаются три раза, а на приёмной стороне производится простое голосование по трём принятым значениям бита. Например, если требуется передать «1», то в канал связи поступит «111». В результате искажений может быть принято, например, «101», а после голосования получится «1» (так как в строке «101» единиц больше чем нулей). Однако искажение двух битов (например, принято «100», вместо «111») никогда не будет обнаружено и исправлено.

Надежность тройного мажорирования оставляет желать лучшего, ибо при некоторых типах информационных линий и некоторых типах ошибок обнаружить ошибку невозможно. Например, линия из 8-и проводов передает по 8-мь бит за такт (по одному через каждый провод). Если один из проводов оборван — эта ошибка никогда не будет обнаружена тройным мажорированием.

Другой существенный недостаток этого метода — перегрузка канала связи — передается в три раза больше бит, чем необходимо.

Код с проверкой на четность

Такой код образуется путем добавления к передаваемой комбинации, состоящей из m информационных символов, одного контрольного символа k

(0 или 1) так, чтобы общее число единиц в передаваемой комбинации было четным.

m	k	n
11011	0	110110
11100	1	111001

Если получено слово с нечётным количеством единичек, то при передаче произошла ошибка.

Например, для кода с m = 5 и вероятностью ошибки p = 10-2 коэффициент обнаружения составит Kобн = 0,9. То есть обнаруживаем 90 % ошибок, при этом избыточность будет составлять L = 0,17 (17 %).

Код с постоянным весом. Этот код содержит постоянное число единиц и нулей. Для примера приведены коды с двумя единицами из пяти и тремя единицами из семи.

N = C52 = 10	N = C73 = 35
11000	0000111
10010	1001001
00101	1010100

Этот код позволяет обнаруживать любые одиночные ошибки и часть многократных ошибок. Не обнаруживаются этим кодом только ошибки смещения, когда одновременно одна единица переходит в ноль и один ноль переходит в единицу, два ноля и две единицы меняются на обратные символы и т.д.

Для кода C73 при p = 10-2 коэффициент обнаружения составит Kобн = 0,985, избыточность составляет L = 27%.

Корреляционный код (Код с удвоением). Элементы данного кода заменяются двумя символами, единица «1» преобразуется в 10, а ноль «0» в 01.

Вместо комбинации 1010011 передается 10011001011010. Ошибка обнаруживается в том случае, если в парных элементах будут одинаковые символы 00 или 11 (вместо 01 и 10).

Например, при m = 5, n = 10 и вероятности ошибки p = 10-2, Kобн = 0,995. Но при этом избыточность будет составлять 50%.

Инверсный код. К исходной комбинации добавляется такая же комбинация по длине, т. е. число символов удваивается. Если в исходной комбинации четное число единиц, то добавляемая комбинация повторяет исходную комбинацию, если нечетное, то добавляемая комбинация является инверсной по отношению к исходной.

m	k	n
11011	11011	1101111011
11100	00011	1110000011

Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в первой основной группе символов. Если число единиц четное, то контрольные символы принимаются без изменения, если нечетное, то контрольные символы инвертируются. На втором этапе контрольные символы суммируются с информационными символами по модулю два. Нулевая сумма говорит об отсутствии ошибок. При ненулевой сумме, принятая комбинация бракуется. Покажем суммирование для принятых комбинаций без ошибок (1, 3) и с ошибками (2, 4):

Данный код обнаруживает практически любые ошибки, кроме редких ошибок смещения, которые одновременно происходят как среди информационных символов, так и среди соответствующих контрольных.

При m = 5, n = 10 и p = 10-2, Кобн = 1 · 10-5.

8 Лекция №13. Передача информации

Цель лекции: изучить основы передачи информации, способы и пути ее передачи**.**

Содержание лекции: Каналы передачи информации. Линии связи, классификация, характеристики. Передача информации по каналу без помех. Передача информации по каналу с помехами. Скорость передачи информации, пропускная способность. Бинарный дискретный канал. Помехоустойчивое кодирование.

Каналы передачи информации

Каналы передачи информации Обычно под этим термином понимаются комплексы технических средств, задействованных в передаче данных от источника к приемнику, а также среда между ними. Структура такого канала, использующая типовые средства передачи информации, представлена следующей последовательностью преобразований:

ИИ – ПС – (КИ) – КК – М – ЛПИ – ДМ – ДК – ДИ - ПС

Здесь:

• ИИ – источник информации: человек либо иное живое существо, книга, документ, изображение на неэлектронном носителе (холст, бумага) и т.д.

• ПС – преобразователь информсообщения в информсигнал, выполняющий первую стадию передачи данных. В качестве ПС могут выступать микрофоны, теле- и видеокамеры, сканеры, факсы, клавиатуры ПК и т. д.

• КИ – кодер информации в информсигнале для сокращения объема (сжатия) информации с целью повысить скорость ее передачи или сократить полосу частот, требуемую для передачи. Данное звено необязательно, что показано скобками.

• КК – канальный кодер для повышения помехозащищённости информсигнала.

• М – сигнальный модулятор для изменения характеристик промежуточных сигналов-носителей в зависимости от величины информсигнала. Типичный пример – амплитудная модуляция сигнала-носителя высокой несущей частоты в зависимости от величины низкочастотного информсигнала.

• ЛПИ – линия передачи информации, представляющая совокупность физической среды (например, электромагнитное поле) и технических средств для изменения ее состояния с целью передачи сигнала-носителя к приемнику.

• ДМ – демодулятор для отделения информсигнала от сигнала-носителя. Присутствует только при наличии М.

• ДК – канальный декодер для выявления и/или исправления ошибок в информсигнале, возникших на ЛПИ. Присутствует только при наличии КК.

• ДИ – декодер информации. Присутствует только при наличии КИ.

• ПИ – приемник информации (компьютер, принтер, дисплей и т. д.). Если передача информации двусторонняя (канал дуплексный), то по обе стороны ЛПИ имеются блоки-модемы (МОдулятор-ДЕМодулятор), объединяющие в себе звенья М и ДМ, а также блоки-кодеки (КОдер-ДЕКодер), объединяющие кодеры (КИ и КК) и декодеры (ДИ и ДК).

Примером канала может служить почта. Информация, закодированная в виде текста, помещается в конверт, поступает в почтовый ящик, извлекается оттуда и перевозится в почтовое отделение, где сортируется (вручную или машиной). Далее информация перемещается с помощью поезда (самолета, теплохода и т.п.) в почтовое отделение пункта назначения, сортируется и доставляется адресату. Таким образом, почтовый канал включает в себя: конверт (предмет), транспорт и сортировочные машины (устройства), почтовых работников (объекты). Информация, помещенная в этот канал, остается неизменной.

Линии связи, их классификация, характеристики

Линия связи - это совокупность устройств, образующих направляющую систему для передачи сигналов.

Линии связи можно классифицировать по используемой среде распространения сигнала:

1. Радиолинии (среда распространения сигнала - открытое пространство).

Радиолинии подразделяются на :

- радиолинии (РЛ), в которых используют следующие диапазоны радиоволн: длинные волны (ДВ), средние волны (СВ), короткие волны (КВ), ультрокороткие волны (УКВ)

- радиорелейные линии (РРЛ) - используют УКВ диапазон;

- спутниковые линии (СЛ) - используют УКВ диапазон.

2. Проводные линии – воздушные и кабельные линии связи (среда распространения сигнала металлический провод).

3. Волоконно-оптические линии (среда распространения сигнала – кварцевый волоконный световод).

Кабельные линии связи по назначению различают: магистральные и местной связи.

В зависимости от способа прокладки они подразделяются на подземные, подводные, воздушные; от конструкции: симметричный (рис. 3.16, а), коаксиальный (рис. 3.16, б).

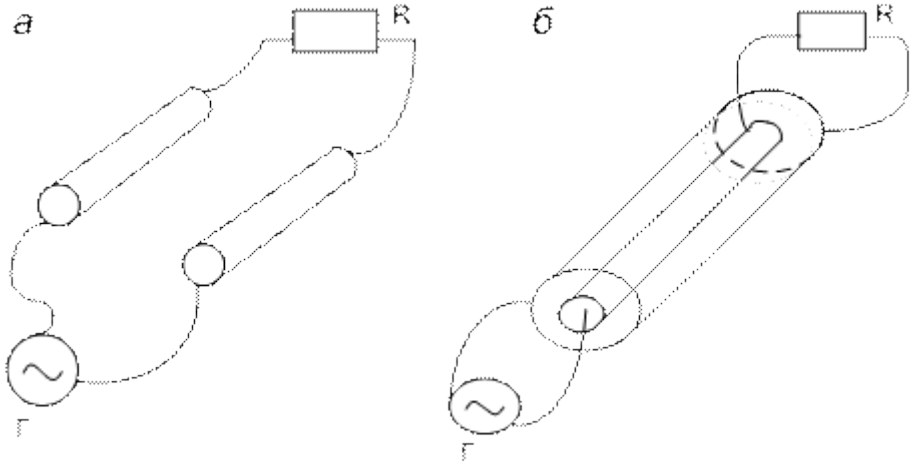


Рис. 3.16. Конструкция кабеля: а – симметричный; б – коаксиальный

В симметричных кабелях двухпроводная цепь состоит из двух одинаковых жил.

В коаксиальных кабелях такую цепь образуют полая цилиндрическая гибкая трубка из медной ленты и сплошной цилиндрический проводник, расположенный в центре этой трубки, из медной или биметаллической проволоки.

Кабель состоит из проводников и изоляции.

Проводники (жилы) изготовляются из медной проволоки для местных сетей

d = 0,4;0,5;0,6;0,7 мм, для магистральных сетей d = 0,8¸1,2 мм.

Токопроводящие жилы кабелей должны обладать хорошей электропроводимостью, гибкостью и достаточной механической прочностью.

Изоляцию жил кабеля различают:

трубчатую (накладывается в виде спиральной трубки рис. 3.17, а);

кордельную (жила обвивается спиралью из корделя (жгута) рис. 3.17, б);

сплошную (рис. 3.17, в).

Кабели многоканальной связи с полиэтиленовой и полистирольной изоляцией жил отличаются от кабелей с кордельно-бумажной изоляцией, т.к. двухпроводные цепи этих кабелей имеют меньшую электрическую емкость и меньшие потери в диэлектрике, не зависящие от частоты тока, передаваемого по этим цепям. Эти преимущества делают выгодным применение таких кабелей при передаче по кабельным цепям токов высокой частоты в полосе до 252 кГц и более.

Отдельные изолированные кабельные жилы скручивают в повивы. Различают простую и сложную скрутку жил. В простой кабельной скрутке, применяемой в сигнально-блокировочных кабелях, повивы кабеля состоят из изолированных жил, предварительно скрученных в группы. Существует несколько способов свивания (скручивания) жил кабеля в группы, самым распространенным из которых являются парная скрутка (рис. 3.18, а) и четверочная (звездная) скрутка (рис. 3.18, б).

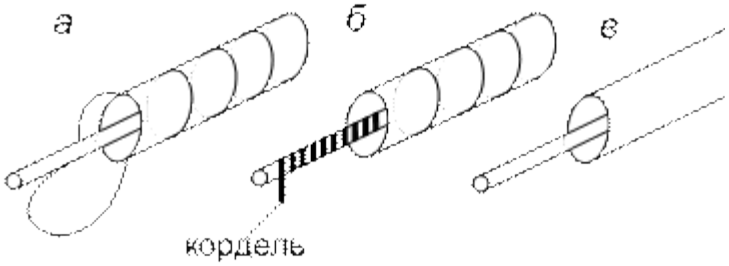
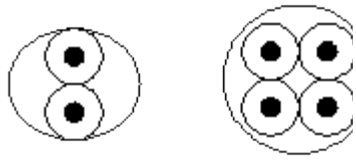


Рис. 3.17. Тип изоляции жил: а – трубчатая; б – кордельная; в – сплошная



а б

Рис. 3.18. Способы скрутки жил кабеля: а – парная; б – четверочная (звездная)

Общую скрутку кабеля заключают в защитную герметическую оболочку, предохраняющую кабель от проникновения в него влаги и защищающую скрутку кабеля от механических воздействий, при транспортировке, прокладке и эксплуатации. В качестве защитных оболочек применяют оболочки из алюминия, свинца, гофрированной стали, а также оболочки из пластмассы и металлопластмассы. По сравнению с пластмассовыми оболочками оболочки из свинца, и особенно из алюминия, защищают кабельные цепи от внешних электромагнитных влияний, т.е. являются экранами. Общий вид кабеля приведен на рис.

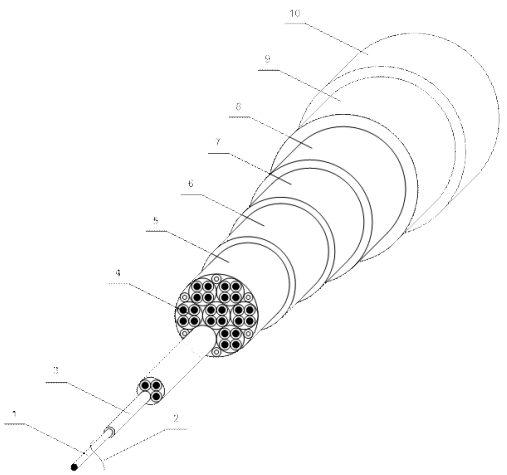


Рисунок 3.19. Конструкция магистрального кабеля: 1 – медная жила; 2 – кордель; 3 – изоляция; 4 – четверка; 5 – сигнальная жила; 6 – поясная изоляция; 7 – алюминиевая оболочка; 8 – полихлорвиниловая лента и подушка под броней; 9 – броня из двух стальных лент; 10 – наружный покров

Типы кабелей:

1) магистральной связи МКАПАБ (в маркировке кабеля буквы обозначают МК - магистральный кабель, П - кордельно-трубчатая полиэтиленовая изоляция жил, А - с алюминиевой оболочкой, Б - бронированный двумя стальными лентами.

МКАБАБ (Б - кордельно-бумажная изоляция жил, остальные то же что в МКАПАБ).

МКПАБ-7Х4Х105+5Х2Х0,7+1Х0,7

Четверки, диаметр | Пары, диаметр | Контрольная ж диаметр

Кабели марки МКАПАБ изготавливают емкостью 4; 7 и 14 четверок; в него так же входят сигнальные пары и контрольная жила. Сигнальные пары предназначены для передачи сигналов устройств железнодорожной автоматики. Контрольная жила имеет не сплошную, а с прерывистую (прореженную) изоляцию. При нарушении герметичности кабеля и проникновении в него влаги, сигнал о повреждении поступает быстрей, что облегчает поиск неисправности.

2) местной связи ТГ, ТБ, ТБГ, ТК, ТПП

Т - телефонный, Г - в свинцовой оболочке, Б - в свинцовой оболочке, бронированный двумя стальными лентами с наружной обмоткой из кабельной пряжи, БГ - свинцовой оболочке, бронированный двумя стальными лентами с наружной обмоткой из вязкого компаунда или лака, К - бронированный круглыми стальными оцинкованными проволоками с

наружным слоем из кабельной пряжи, ПП - с полиэтиленовой изоляцией в полиэтиленовой оболочке.

Кабели марки ТГ и ТПП прокладывают в телефонной канализации и в шахтах, ТБ - в земле, ТБГ - внутри помещений в каналах и тунелях, ТК - в земле и под водой.

Оптическим кабелем называется кабельное изделие, содержащее ряд оптических волокон, заключенных в общую оболочку, поверх которой в зависимости от условий эксплуатации может быть наложен защитный покров.

Основным элементом оптического кабеля является оптический волновод – круглый стержень из оптически прозрачного диэлектрика, структура которого обеспечивает распространение вдоль него световых сигналов. Оптические кабели по своему назначению могут быть разделены на четыре группы:

– междугородние;
– городские;
– объектовые;
– подводные.

В отдельную группу выделяют монтажные оптические кабели.

Технические особенности:

1. Высокая помехоустойчивость.
2. Отсутствие взаимных влияний в соседних волокнах.
3. безопасность работы волоконно-оптической связи в детонирующих, воспламеняющихся или электрически опасных средах и условиях.
4. Большая ширина спектра оптического кабеля, повышенная емкость канала, большая длина регенерационных участков.
5. Меньшие габариты и вес, чем у медного кабеля.
6. Защита от несанкционированного доступа.

Оптические волокна можно разделить на следующие типы: кварцевые, кварц-полимерные и полимерные.

Кварцевые оптические волокна изготавливаются из высокочистого кварцевого стекла (сердечник и светоотражающая оболочка) и применяются для систем дальней, внутри- и межобъектовой связи.

Кварц-полимерные оптические волокна изготавливаются с кварцевым сердечником и полимерной светоотражающей оболочкой и предназначены для систем внутри- и межобъектовой связи.

Полимерные оптические волокна изготавливаются из полимерных материалов, имеющих высокие оптические свойства, и используются для некоторых систем внутриобъектовой связи, подсветки, декоративного оформления и в медицине.

Конструкция волоконно-оптического кабеля приведена на рис. 3.21.

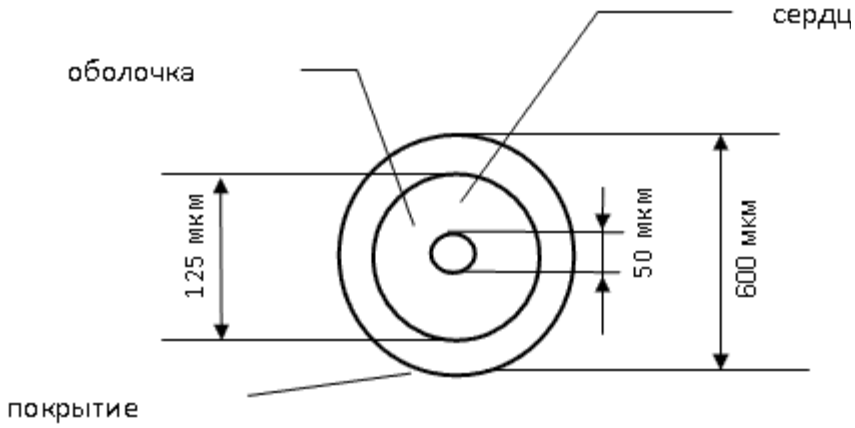


Рис. 3.21. Конструкция оптического кабеля

Для связи по световодам используются видимые лучи (0,4…0,75 мкм) и ближний диапазон инфракрасных лучей (0,85; 1,3; 1,55 … 6 мкм). При этом возможна передача большого числа различных типов волн-мод (m). Исходя из двойственной природы света (лучевой и волновой) различным типам волн – модам соответствует различное число лучей. Одномодовой передаче соответствует один луч (рис. 3.22, *а*), а многомодовой, например, три (рис. 3.22, *б*).

Одномодовый режим возможен при l » d, l < d, где d – диаметр световода.

Достоинства одномодовых систем:

- малая дисперсия (искажение сигналов);
- большая пропускная способность;
- большая дальность передачи;
- нет модовых искажений (в многомодовых – различные лучи (моды) идут в световоде под различными углами, проходит различный путь и к концу приходят в различные отрезки времени).

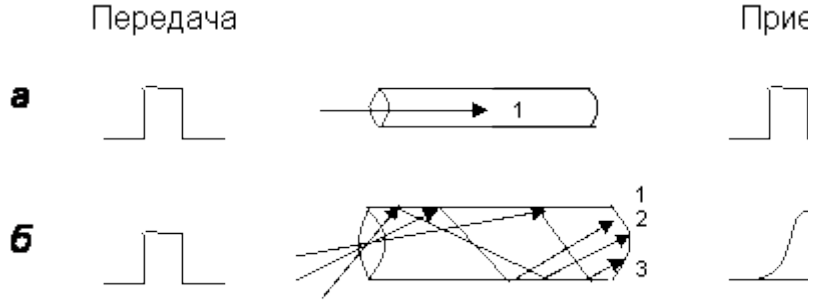


Рис. 3.22. Режимы распространения оптического излучения по световоду: *а* – одномодовый; *б* – многомодовый

Особенностью работы систем передачи, работающих по волоконно-оптическим линиям, является дополнительное преобразование электрического сигнала в оптический на передающем конце (с помощью полупроводниковых лазеров и светодиодов) и обратное преобразование на приеме (с помощью фотодиодов). Оптические системы передачи являются цифровыми.

Лекция 14 Передача информации по каналу с помехами

Действие помех в канале связи может привести к потере части передаваемой информации. В связи с этим возникает вопрос о пропускной способности канала при наличии помех. Под пропускной способностью в этом случае понимается та максимальная скорость передачи информации при заданном уровне помех, при которой вероятность появления ошибки сколь угодно мала.

Пропускная способность канала определяется из выражения:

$C = \frac{n}{T}\left[H(X) - H_Y(X)\right]$ дв. ед/сек,

где $H(X)$ – энтропия источника информации;

$H_Y(X)$ – энтропия неопределенности переданного сигнала X относительно принятого Y.

Из этой формулы следует, что пропускная способность уменьшается с увеличением $H_Y(X)$. Это означает, что для уменьшения действия помех необходимо вводить помехоустойчивое кодирование. Поскольку обычные шумовые помехи оказывают наибольший эффект на уровень (амплитуду) сигнала, все способы модуляции, в которых уровень не используется в качестве переносчика информации, являются помехоустойчивыми. В случае амплитудно–импульсной модуляции (АИМ) по мере повышения уровня помехи все большее число импульсов полезного сигнала теряется на фоне шума и, следовательно, помехоустойчивость канала падает.

Передача информации в канале с кодоимпульсной модуляции (КИМ) осуществляется за счет определенного расположения импульсов на временных позициях кода. В идеальном случае параметры самих импульсов не несут информации и, следовательно, их искажение не вносит ошибок в передачу. Ошибки возникают лишь тогда, когда уровень помех настолько велик, что возможно образование на свободных позициях кода ложных импульсов и подавление полезных сигналов.

Пропускная способность канала при наличии помех будет определятся из выражения:

$$C = F_m \log\left(1 + \frac{P_c}{P_n}\right),$$

где P_c – мощность полезного сигнала;

P_n – мощность помехи;

F_m – наивысшая частота в спектре сигнала.

Используем формулу для равномерно квантованного сигнала и равновероятного появления каждого уровня. В этом случае $P_c = \frac{X_m^2}{3}$, где X_m – максимальное значение уровня сигнала.

Дисперсия помехи:

$$P_n = \frac{\delta}{12},$$

где δ – шаг квантования.

Подставляя эти значения в (16) получим:

$$C = F_m \cdot \log(1 + m^2) = 2F_m \cdot \log m,$$

где m – число уровней квантования; $m^2 = \frac{4X_m^2}{\delta^2}$

Мощность помехи в полосе частот полезного сигнала определяется из выражения:

$$P_n = S_0 \cdot F_m,$$

где S_0 – спектральная плотность шума.

Определим скорость передачи информации по каналу связи при амплитудно–импульсной модуляции (АИМ). Шаг дискретизации по теореме Котельникова будет определятся:

$$\Delta t = \frac{1}{2F_m}$$

Обычно интервал между импульсами кодовой комбинации равен длительности импульса. Это означает, что ширина спектра импульса $F_И = 4F_m$.

Так как все уровни сигнала равновероятны, то дисперсия сигнала (средняя мощность) будет равна:

$$P_c = \frac{X_m^2}{3}$$

Или

$$P_c = \frac{(m \cdot \delta)^2}{12}$$

где m – количество уровней квантования;

δ – шаг квантования.

Для безошибочной передачи информации необходимо выбрать такой шаг квантования, при котором влияние шума было бы мало.

Вероятность ошибки в случае АИМ есть вероятность того, что уровень шума превысит половину шага квантования. Эта вероятность для нормального шума равна:

$$P_0 = \int_{\delta}^{\infty} P_m(X)dX,$$

где

$$P_m(X) = \frac{12}{\sqrt{2\pi P_n}} \cdot e^{-\frac{X^2}{2P_n}}$$

В связи с этим мощность помехи в полосе частот импульсов АИМ будет определятся:

$$P_n = 4S_0 \cdot F_m$$

Из соотношений (20), (21) и (23) получим:

$$m^2 = \frac{3P_c}{F_m \cdot K^2 \cdot S_0},$$

где K выбирается из табл. 1, а $P_0 = \frac{\Delta t}{T} = \frac{1}{2F_m \cdot T}$.

Таблица 1

P_0	$5 \cdot 1$	6.6	$7 \cdot 1$	$6 \cdot 1$	$4 \cdot 1$
K	7	8	9	10	11
K_1	14	16	18	20	22

Скорость передачи по каналу связи будет равна:

$$V_{АИМ} = \frac{n \log m}{T} = F_m \log m^2$$

Избыточность кодов АИМ по отношению к идеальному кодированию может быть вычислена по формуле:

$$R = 1 - \frac{V_{АИМ}}{C}$$

Определим скорость передачи информации по каналу при кодо–импульсной модуляции (КИМ). Для передачи одного значения сигнала $X(t)$ при использовании КИМ требуется передавать кодовую группу из n импульсов двух уровней (0 и 1). При этом общая длительность кодовых импульсов (с учетом интервалов между ними) должна быть равна длительности импульса при АИМ.

Следовательно, ширина спектра импульса КИМ будет:

$$F_{И2} = (n-1) * F_{И_1} = 4(n-1)F_m$$

Будем полагать, что прием импульсов КИМ производится приемником с пороговым устройством, всякий раз, когда уровень импульса U превышает порог h на малую величину ε, то есть $U = h$ (рисунок 1).

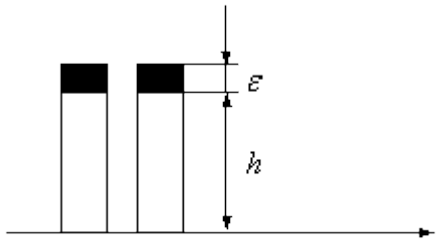


Рис. 1

В таблице 1 приведены значения K_1 в зависимости от вероятности ошибки

$$K_1 = \frac{h}{\sqrt{P_n}}$$

Мощность шума будет равна:

$$P_ш = S_0 \cdot F_{ИЗ} = 4S_0 \cdot (n-1) \cdot F_m$$

А средняя мощность полезного сигнала:

$$P_c = \frac{P_п}{2} = \frac{U^2}{2} = \frac{h^2}{2},$$

т.к. $P_п = U^2 = h^2$

На основании выражений (28), (29), (30) получим:

$$n = \frac{P_c}{2S_0 \cdot F_m \cdot K_1^2} + 1$$

где n округляется до большего целого числа.

Скорость передачи по каналу с КИМ:

$$V_{КИМ} = \frac{n}{\Delta t} = 2F_m \cdot n$$

Избыточность кода КИМ по отношению к идеальному кодированию определяется из выражения:

$$R_{КИМ} = 1 - \frac{V_{КИМ}}{C}$$

Скорость передачи информации, пропускная способность

Обмен информацией производится по каналам передачи информации.

Каналы передачи информации могут использовать различные физические принципы. Так, при непосредственном общении людей информация передаётся с помощью звуковых волн, а при разговоре по телефону — с помощью электрических сигналов, которые распространяются по линиям связи.

Канал связи — технические средства, позволяющие осуществлять передачу данных на расстоянии.

Компьютеры могут обмениваться информацией с использованием каналов связи различной физической природы: кабельных, оптоволоконных, радиоканалов и др.

Скорость передачи информации (скорость информационного потока) — количество информации, передаваемое за единицу времени.

Общая схема передачи информации включает в себя отправителя информации, канал передачи информации и получателя информации.

Основной характеристикой каналов передачи информации является их пропускная способность.

Пропускная способность канала — максимальная скорость передачи информации по каналу связи в единицу времени.

Пропускная способность канала равна количеству информации, которое может передаваться по нему в единицу времени.

Объем переданной информации V вычисляется по формуле:

$V=q\cdot t$

где q — пропускная способность канала (в битах в секунду или подобных единицах), а t— время передачи.

Обычно пропускная способность измеряется в битах в секунду (**бит/с**) и кратных единицах **Кбит/с** и **Мбит/с**.

Однако иногда в качестве единицы используется байт в секунду (**байт/с**) и кратные ему единицы **Кбайт/с** и **Мбайт/с**.

Соотношения между единицами пропускной способности канала передачи информации такие же, как между единицами измерения количества информации:

1 байт/с = 23 бит/с = 8 бит/с;1 Кбит/с = 210 бит/с = 1024 бит/с;1 Мбит/с = 210 Кбит/с = 1024 Кбит/с;1 Гбит/с = 210 Мбит/с = 1024 Мбит/с.

Помехоустойчивое кодирование

Помехоустойчивое кодирование (англ. Error Correcting Coding, ECC) — процесс преобразования информации, предоставляющий возможность обнаружить и исправить ошибки, возникающие при передаче информации по каналам передачи данных.

Процесс помехоустойчивого кодирования заключается во введении избыточности, т. е. для передачи информации используется код, у которого используются не все возможные комбинации, а только некоторые из них.

Такие коды называют избыточными или корректирующими. Соответственно, процесс введения избыточности (преобразование информационных символов в кодовое слово) называется кодированием, а обратный процесс восстановления информации из кодового слова, возможно содержащего ошибки — декодированием.

Еще одним важным свойством помехоустойчивого кодирования является эффект усреднения шума, который достигается за счет того, что избыточные символы зависят от нескольких информационных символов. В рамках цифровой системы передачи данных задачи кодирования и декодирования возложены на кодер и декодер соответственно.

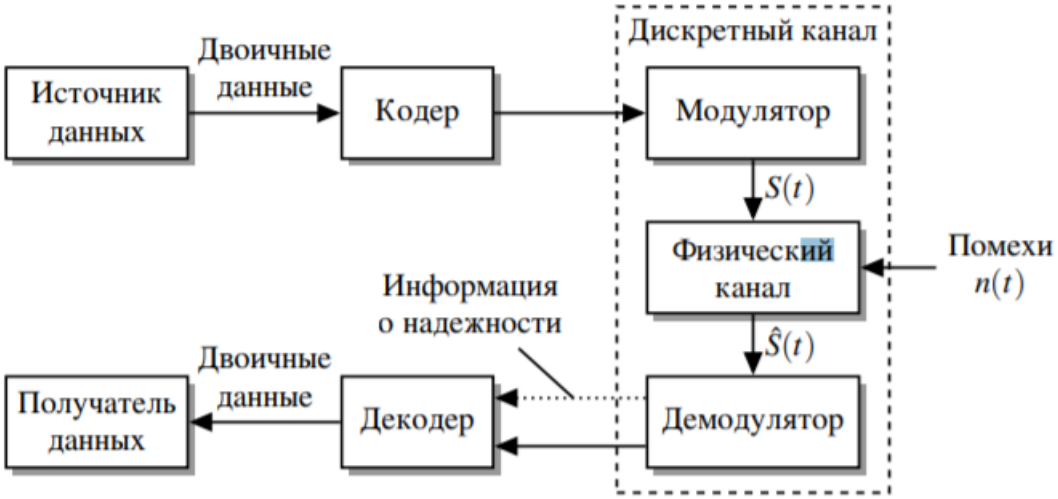


Рис. 1.1. Структура цифровой системы передачи данных

Структура цифровой системы передачи данных показана на рис. 1.1

На рис. 1.1 также показано, что часто декодеру доступна информация, указывающая на надежность решений, принимаемых о различных символах кодового слова. Такая информация часто может быть использована для упрощения процесса декодирования, либо для улучшения его характеристик [1]. В целом, способность помехоустойчивых кодов определять и исправлять ошибки — их корректирующие свойства — зависят от правил построения этих кодов и параметров кода (числа разрядов, избыточности и др.), а также от используемых алгоритмов декодирования.

Основными параметрами, характеризующими корректирующие свойства кодов являются:

1. Избыточность кода.
2. Кодовое расстояние.
3. Кратность гарантированно обнаруживаемых ошибок.
4. Кратность гарантированно исправляемых ошибок.

9 Лекция №. 15 Кодирование и шифрование информации

Цель лекции: изучить основные принципы кодирования и шифрования информации, их виды и преобразования.

Содержание лекции: Понятие о криптографической защите информации. Классификация криптографических методов защиты информации. Смысловое и символьное кодирование. Сжатие информации.

Понятие о криптографической защите информации

Криптографическая защита информации – это механизм защиты посредством шифрования данных для обеспечения информационной безопасности общества.

Криптографические методы защиты информации активно используются в современной жизни для хранения, обработки и передачи информации по сетям связи и на различных носителях.

Ключевой целью криптографической защиты информации является обеспечение конфиденциальности и защиты информационных данных компьютерных сетей в процессе передачи ее по сети между пользователями системы.

Защита конфиденциальной информации, которая основана на криптографической защите, зашифровывает информационные данные посредством обратимых преобразований, каждое из которых описывается ключом и порядком, что определяет очередность их применения.

Классификация криптографических методов защиты информации

Процесс ***шифрования*** заключается в проведении обратимых математических, логических, комбинаторных и других преобразований исходной информации, в результате которых зашифрованная информация представляет собой хаотический набор букв, цифр, других символов и двоичных кодов.

Для шифрования информации используются **алгоритм преобразования** и **ключ**. Как правило, алгоритм для определенного метода шифрования является неизменным. Исходными данными для алгоритма шифрования служит информация, подлежащая зашифрованию, и ключ шифрования. Ключ содержит управляющую информацию, которая определяет выбор преобразования на определенных шагах алгоритма и величины операндов, используемых при реализации алгоритма шифрования. Операнд – это константа, переменная, функция, выражение и другой объект языка программирования, над которым производятся операции.

В отличие от других методов криптографического преобразования информации, методы ***стеганографии*** позволяют скрыть не только смысл хранящейся или передаваемой информации, но и сам факт хранения или передачи закрытой информации. В основе всех методов стеганографии лежит маскирование закрытой информации среди открытых файлов, т.е. скрываются секретные данные, при этом создаются реалистичные данные, которые невозможно отличить от настоящих. Обработка мультимедийных файлов в

информационных системах открыла практически неограниченные возможности перед стеганографией.

Графическая и звуковая информация представляются в числовом виде. Так, в графических объектах наименьший элемент изображения может кодироваться одним байтом. В младшие разряды определенных байтов изображения в соответствии с алгоритмом криптографического преобразования помещаются биты скрытого файла. Если правильно подобрать алгоритм преобразования и изображение, на фоне которого помещается скрытый файл, то человеческому глазу практически невозможно отличить полученное изображение от исходного. С помощью средств стеганографии могут маскироваться текст, изображение, речь, цифровая подпись, зашифрованное сообщение.

Скрытый файл также может быть зашифрован. Если кто-то случайно обнаружит скрытый файл, то зашифрованная информация будет воспринята как сбой в работе системы. Комплексное использование стеганографии и шифрования многократно повышает сложность решения задачи обнаружения и раскрытия конфиденциальной информации.

Содержанием процесса ***кодирование*** информации является замена исходного смысла сообщения (слов, предложений) кодами. В качестве кодов могут использоваться сочетания букв, цифр, знаков. При кодировании и обратном преобразовании используются специальные таблицы или словари. В информационных сетях кодирование исходного сообщения (или сигнала) программно-аппаратными средствами применяется для повышения достоверности передаваемой информации.

Часто кодирование и шифрование ошибочно принимают за одно и тоже, забыв о том, что для восстановления закодированного сообщения, достаточно знать правило замены, в то время как для расшифровки сообщения помимо знания правил шифрования, требуется ключ к шифру.

Сжатие информации может быть отнесено к методам криптографического преобразования информации с определенными оговорками. Целью сжатия является сокращение объема информации. В то же время сжатая информация не может быть прочитана или использована без обратного преобразования. Учитывая доступность средств сжатия и обратного преобразования, эти методы нельзя рассматривать как надежные средства криптографического преобразования информации. Даже если держать в секрете алгоритмы, то они могут быть сравнительно легко раскрыты статистическими методами обработки. Поэтому сжатые файлы конфиденциальной информации подвергаются последующему шифрованию. Для сокращения времени передачи данных целесообразно совмещать процесс сжатия и шифрования информации.

Основным видом криптографического преобразования информации в компьютерных сетях является ***шифрование***. Под шифрованием понимается процесс преобразования открытой информации в зашифрованную информацию (шифртекст) или процесс обратного преобразования

зашифрованной информации в открытую. Процесс преобразования открытой информации в закрытую получил название зашифрование, а процесс преобразования закрытой информации в открытую – расшифрование.

За многовековую историю использования шифрования информации человечеством изобретено множество методов шифрования или шифров. ***Методом шифрования (шифром)*** называется совокупность обратимых преобразований открытой информации в закрытую информацию в соответствии с алгоритмом шифрования. Большинство методов шифрования не выдержали проверку временем, а некоторые используются и до сих пор. Появление компьютеров и компьютерных сетей инициировало процесс разработки новых шифров, учитывающих возможности использования компьютерной техники как для зашифрования/расшифрования информации, так и для атак на шифр. ***Атака на шифр (криптоанализ, криптоатака)*** – это процесс расшифрования закрытой информации без знания ключа и, возможно, при отсутствии сведений об алгоритме шифрования.

Современные методы шифрования должны отвечать следующим требованиям:

- стойкость шифра противостоять криптоанализу (криптостойкость) должна быть такой, чтобы вскрытие его могло быть осуществлено только путем решения задачи полного перебора ключей;
- криптостойкость обеспечивается не секретностью алгоритма шифрования, а секретностью ключа;
- шифртекст не должен существенно превосходить по объему исходную информацию;
- ошибки, возникающие при шифровании, не должны приводить к искажениям и потерям информации;
- время шифрования не должно быть большим;
- стоимость шифрования должна быть согласована со стоимостью закрываемой информации.

Криптостойкость шифра является его основным показателем эффективности. Она измеряется временем или стоимостью средств, необходимых криптоаналитику для получения исходной информации по шифртексту, при условии, что ему неизвестен ключ.

Сохранить в секрете широко используемый алгоритм шифрования практически невозможно. Поэтому алгоритм не должен иметь скрытых слабых мест, которыми могли бы воспользоваться криптоаналитики. Если это условие выполняется, то криптостойкость шифра определяется длиной ключа, так как единственный путь вскрытия зашифрованной информации – перебор комбинаций ключа и выполнение алгоритма расшифрования. Таким образом, время и средства, затрачиваемые на криптоанализ, зависят от длины ключа и сложности алгоритма шифрования.

Преобразование шифрования может быть **симметричным** и **асимметричным** относительно преобразования расшифрования. Это важное свойство определяет два класса криптосистем:

- симметричные (одноключевые) криптосистемы;
- асимметричные (двухключевые) криптосистемы (с открытым ключом).

Симметричное шифрование

Симметричное шифрование, которое часто называют шифрованием с помощью секретных ключей, в основном используется для обеспечения конфиденциальности данных. Для того чтобы обеспечить конфиденциальность данных, пользователи должны совместно выбрать единый математический алгоритм, который будет использоваться для шифрования и расшифровки данных. Кроме того, им нужно выбрать общий (секретный) ключ, который будет использоваться с принятым ими алгоритмом шифрования/дешифрования, т.е. один и тот же ключ используется и для зашифрования, и для расшифрования (слово "симметричный" означает одинаковый для обеих сторон).

Сегодня широко используются такие алгоритмы шифрования, как Data Encryption Standard (DES), 3DES (или "тройной DES") и International Data Encryption Algorithm (IDEA). Эти алгоритмы шифруют сообщения блоками по 64 бита. Если объем сообщения превышает 64 бита (как это обычно и бывает), необходимо разбить его на блоки по 64 бита в каждом, а затем каким-то образом свести их воедино. Такое объединение, как правило, происходит одним из следующих четырех методов:

- электронной кодовой книги (Electronic Code Book, ECB);
- цепочки зашифрованных блоков (Cipher Block Changing, CBC);
- x-битовой зашифрованной обратной связи (Cipher FeedBack, CFB-x);
- выходной обратной связи (Output FeedBack, OFB).

Triple DES (3DES) – симметричный блочный шифр, созданный на основе алгоритма DES, с целью устранения главного недостатка последнего – малой длины ключа (56 бит), который может быть взломан методом полного перебора ключа. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше. Время, требуемое для криптоанализа 3DES, может быть намного больше, чем время, нужное для вскрытия DES.

Алгоритм ***AES*** (Advanced Encryption Standard), также известный как Rijndael – симметричный алгоритм блочного шифрования – шифрует сообщения блоками по 128 бит, использует ключ 128/192/256 бит.

Шифрование с помощью секретного ключа часто используется для поддержки конфиденциальности данных и очень эффективно реализуется с помощью неизменяемых "вшитых" программ (firmware). Этот метод можно использовать для аутентификации и поддержания целостности данных.

Смысловое и символьное кодирование.

Символьный способ кодирования текста - это способ кодирования текста с помощью того же алфавита, что и исходный текст.

Приведем несколько исторических примеров.

Наиболее известен **шифр Цезаря**. Применительно к современному русскому языку он состоял в следующем. Выписывался алфавит: А, Б, В, Г, Д, Е,...,; затем под ним выписывался тот же алфавит, но со сдвигом на 3 буквы влево

При зашифровке буква А заменялась буквой Г, Б заменялась на Д, Б-Е и так далее. Так, например, слово **"РИМ"** превращалось в слово **"УЛП"**. Получатель сообщения "УЛП" искал эти буквы в нижней строке и по буквам над ними восстанавливал исходное слово "РИМ". Ключом в шифре Цезаря является величина сдвига 3-й нижней строки алфавита. Преемник Юлия Цезаря - **Цезарь Август** - использовал тот же шифр, но с ключом - сдвиг 4. Слово **"РИМ"** он в этом случае зашифровал бы в буквосочетание **"ФМР"**.

Система Цезаря не является старейшей. Возможно, что наиболее древней из

известных является система греческого историка **Полибия**, умершего за 30 лет до рождения Цезаря. Организация «Земля и воля» применяла **«шифр Полибия»**, который нередко в последствии называли «тюремным шифром». Этот шифр был удобен тем, что им можно было легко перестукиваться через стенки тюремных камер, в нем буквы русского алфавита последовательно вписывались в прямоугольник размера 5 на 6, и буква заменялась на ее координаты в таблице. Например, буква Б стояла в первой строке на втором месте, тогда она передавалась при перестукивании следующим образом: удар – длинная пауза – два коротких удара. таким шифром активно пользовался декабрист М. А. Бестужев, находившийся в 1826 году в Алексеевском равелине Петропавловской крепости .

Каждая буква может быть представлена парой букв, указывающих строку и столбец, в которых расположена данная буква.

Активно революционерами в XIX веке использовались и **маршрутные тексты**

Пусть имеется следующий секретный ключ 2.5.6.3.1.4.

Открытый текст, например: «Развитие криптографии идет своим чередом» вписывается в квадрат размером 6 на 6.

	1	2	3	4	5	6
1	р	а	з	в	и	Т
2	и	е	к	р	и	п
3	т	о	г	р	а	ф
4	и	и	и	д	е	т
5	с	в	о	и	м	ч

6	е	р	е	д	о	м

В соответствии с ключом производится перестановка столбцов: первый столбец встает на 5-ое место, второй-на первое. Получим

	1	2	3	4	5	6
1	а	и	т	з	р	в
2	е	и	п	к	и	р
3	о	а	ф	г	т	р
4	и	е	т	и	и	д
5	в	м	ч	о	с	и
6	р	о	м	е	е	д

Затем по этому же закону переставляются строки: первая-на 5 место, вторая-на 1-е и т. д.

	1	2	3	4	5	6
1	е	и	п	к	и	р
2	в	м	ч	о	с	и
3	р	о	м	е	е	д
4	о	а	ф	г	т	р
5	а	и	т	з	р	в
6	и	е	т	и	и	д

Шифрованный текст выписывается построчно:
еипкирвмчосиромеедоафгтраитзрвиетиид

Для примера шифра перестановки выберем целое положительное число, скажем 5; расположим числа от 1 до 5 в двустрочной записи

1	2	3	4	5
3	2	5	1	4

Зашифруем фразу "шифрование". Выпишем эту, разбив ее на пятизначные группы: шифро вание.

Буквы каждой группы переставим в соответствии с указанной двухстрочной записью по следующему правилу: первая буква встает на третье место, вторая - на второе, третья на пятое, четвертая на первое и пятая на четвертое. Полученный текст выписывается без пропусков:

ришориавен

Список литературы

Основная литература

22. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. - Москва: Наука, 2015. - 552 с.

23. Бабаш, А. В. Информационная безопасность (+ CD-ROM) / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. - М.: КноРус, 2013. - 136 с.

24. Васильков, А. В. Безопасность и управление доступом в информационных системах / А.В. Васильков, И.А. Васильков. - М.: Форум, 2015. - 368 с.

25. Гафнер, В. В. Информационная безопасность / В.В. Гафнер. - М.: Феникс, 2014. - 336 с.

26. Информационная безопасность открытых систем. В 2 томах. Том 1. Угрозы, уязвимости, атаки и подходы к защите / С.В. Запечников и др. - Москва: Машиностроение, 2016. - 536 с.

27. Информационная безопасность открытых систем. В 2 томах. Том 2. Средства защиты в сетях / С.В. Запечников и др. - Москва: СПб. [и др.] : Питер, 2014. - 560 с.

28. Мельников, В. П. Информационная безопасность / В.П. Мельников, С.А. Клейменов, А.М. Петраков. - М.: Academia, 2017. - 336 с.

29. Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников. - М.: Академия (Academia), 2016. - 282 с.

30. Партыка, Т. Л. Информационная безопасность / Т.Л. Партыка, И.И. Попов. - М.: Форум, Инфра-М, 2016. - 368 с.

31. Степанов, Е.А. Информационная безопасность и защита информации. Учебное пособие / Е.А. Степанов, И.К. Корнеев. - М.: ИНФРА-М, 2017. - 304 с.

32. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей / В.Ф. Шаньгин. - М.: Форум, Инфра-М, 2017. - 416 с.

33. Ярочкин, В. Безопасность информационных систем / В. Ярочкин. - М.: Ось-89, 2016. - 320 с.

Интернет Ресурсы

34. Информационная безопасность - http://protect.htmlweb.ru

35. Информационная безопасность - http://wikipedia.org

36. Информационная безопасность и защита информации. Учебное пособие – М.: 2004 – 82 с. http://bezopasnik.org›article/book/23.pdf

37. https://moodle.tsu.ru/course/view.phpid=1557

38. http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000534758

39. http://www.intuit.ru/studies/courses/2259/155/info

40. http://www.intuit.rU/studies/courses/l0/10/info

Дополнительная

41. Девянин, П.Н. Анализ безопасности управления доступом и информационными потоками в компьютерных системах / П.Н. Девянин. - М.: Радио и связь, 2013. - 176 с.

42. Айтхожаева Е.Ж., Шерстобитова Т.М. Информатика: учебное пособие.- Алматы: КазНТУ, 2012.