Реферат: Методы сжатия цифровой информации. Метод Лавинского

Болохово 2014. Гр 313 Лифанова А.Э.

**Содержание**

Введение

1. Постановка задачи

2. Обзор существующих методов решения задачи

2.1 Сжатие и кодирование информации в информационно вычислительных комплексах (ИВК)

2.2 Сжатие с восстановлением

2.3 Методы сжатия цифровой информации с повторяющимися фрагментами

3. Выбор и обоснование решения задачи

4. Теоретическое обоснование метода Лавинского

5. Программное обеспечение информационный выбор метода

Заключение

Библиографический список

Приложение А

Приложение Б

**Введение**

В наши дни все большее распространение получает обработка и хранение информации при помощи ЭВМ. При этом одной из важнейших задач является сохранение ее целостности, т.е. защита от потери данных, как при их передаче, так и в некоторых случаях при хранении.

Метод Лавинского относится к простейшим методам сжатия информации (числовых массивов) и он осуществляет сжатие путем уменьшения разрядности числа (исходного). Метод тем лучше функционирует, чем больше массив и разность между числами в нем составляет малую величину.

**1. Постановка задачи**

Составить программу сжатия по методу Лавинского, показать её возможности на выбранном Вами примере.

Программный продукт предусматривает сжатие массива, прочитанного из файла, по методу Лавинского,т.е. уменьшения разрядности чисел содержащихся в исходном файле. Это достигается путем преобразования символов файла в биты и запись их в новый файл.

Де архивация строится на основе того, что в новый (сжатый файл) перед каждым символом записывается номер границы к которой это число относится, а размер для каждой границы есть константа умноженная на номер границы.

**2. Обзор существующих методов решения задачи**

**2.1 Сжатие и кодирование информации в информационно вычислительных комплексах (ИВК)**

ИВК – это набор, состоящий из одного или нескольких ЭВМ, снабженных устройствами хранения, ввода вывода и передачи информации. ИВК имеет отдельные устройства, разнесенные между собой.

Информационная сеть представляет собой набор ИВК, соединенных между собой каналами передачи информации (каналы могут быть любой протяженности). Для того, чтобы сеть могла функционировать, она снабжается набором протоколов и интерфейсов.

Протокол – некоторое множество информационных функций и алгоритмов обработки информации, которые приняты в той или иной сети.

Интерфейс – некоторое соединение или канал между отдельными функциональными частями сети. Интерфейсы бывают физические и программные.

Физический интерфейс –набор шин, для передачи сигналов, и электрических устройств для управления прохождения сигналов по этим шинам. Большинство физических интерфейсов — стыки.

Программный интерфейс –часть программного обеспечения сети, отвечающая за передачу информации от узла к узлу.

Сети могут быть гомогенными гетерогенными (однородными и разнородными).

В гомогенных сетях используются однотипные ЭВМ и однотипные программное обеспечение.

В гетерогенных сетях протоколы, кроме всего прочего, согласуют разнородное программное обеспечение, а интерфейсы согласуют физические пороги сигналов.

Все универсальные сети являются гомогенными. Сеть чаще всего является открытой системой.

Открытой называется такая система, которая может взаимодействовать с другими системами. Для того, чтобы открытая система нормально функционировала она должна обеспечивать семь уровней этого функционирования:

прикладной

представительский

сеансовый

транспортный

сетевой

канальный

физический

Прикладной уровень функционирования предполагает унификацию и

/> <td/> />
стандартизацию управляющих сигналов, форматы информационных кадров, методов кодирования и защиты от ошибки основные служебные посылки.

Ф1 и Ф2 – флаги;

А – адрес;

З О – защита от ошибок.

Представительский уровень унифицирует форму представления информации, то есть тип сигналов, вид кодов, способы защиты от ошибок и правила семиотики (науки о знаках) для выбранной знаковой системы.

Сеансовый уровень унифицирует длительность сеансов связи между узлами сети, служебную информацию для вызова или организации таких сеансов, способ стыковки между функциональными частями при сеансе связи.

Транспортный уровень унифицирует собственно передачу информации, то есть ее скорость или время, способ передачи (параллельно, последовательно или смешанно) информации и виды модемов и аппаратуры передачи данных.

Сетевой уровень унифицирует (стандартизует) прохождение сигналов по очередям, вид этих очередей, способ обслуживания, разновидности персональной защиты и доступа(ключи, пароли, шифры).

Канальный уровень унифицирует прохождение сигналов по каналу с помощью унификации инициализации, синхронизации и аппаратуры защиты от ошибок.

Физический уровень проводит унификацию физических сигналов по уровню (амплитуде), частоте, фазе ипо виду модуляции сигналов.

В открытых сетях, в виду огромных объемов проходящей информации, производится сжатие информации. Существует сжатие без восстановления и с восстановлением. Сжатие без восстановления предполагает, что передается алфавитно-цифровая информация, которая тем или иным способом уменьшается в объеме и на приемной стороне принимается сжатый объем. А при сжатии с восстановлением приемник получает исходный текст, при условии, что передавался сжатый текст. В общем случае, сжатие (компрессия) данных представляет собой процесс выделения из исходного информационного массива его информативной части путем отбрасывания некоторых символов, несущих минимальное число этой информации. Сжатие производится до тех пор, пока информативность сохраняется. Простейший способ сжатия без восстановления для текстовой информации на естественном языке, предполагает наличие словаря запретов, который поддерживается сетью и доводится до всех абонентов. В него входят одно, двух и трехбуквенные слова с минимальной информативностью, которые из текста исключаются. Из текста, начиная от концаслова к началу, убирают все гласные и часть согласных до наличия еще в слове необходимого смысла. Первые три согласные несут 84% информации.

Если информация представлена в цифровом виде, то в этом случае задают длину блока, до которого необходимо ее сжать.

Весь текст бьется на блоки заданной длины или меньшей и затем производится либо сложение их по модулю два, либо двоичное сложение и передается их сумма.

Если информация нецифровая, а текстовая, то можно использовать тот же метод, если каждую букву закодировать некоторым кодом равномерной длины.

/> <td/> />
В том случае, если длина блока, кодирующего букву меньше, чем требуемая длина блока для передачи, сложение производится при ступенчатом сдвиге одной буквы относительно другой на одну позицию влево, начиная от первой буквы слова к последней.

**2.2 Сжатие с восстановлением**

Методы сжатия с восстановлением должны обеспечить переход к исходному сообщению при заданном КСЖ.

n1 – число символов в исходном сообщении

n2 – число символов в сжатом сообщении

Простейшим способом такого сжатия является способ хранения атрибутов в виде битовой матрицы.

Передаются только единицы, которые оговариваются либо частотой, либо временем и т. д.

**2.3 Методы сжатия цифровой информации с повторяющимися фрагментами**

Предполагается, что информация записывается в файлы. Первая часть применима для тех информационных массивов, в которых повторяющиеся фрагменты стоят в начале строки. В этом случае используется символ пропуска r, весь массив передается одной строкой.

123456r7r41r2

Восстановление начинается от конца к началу, при известном количестве символов в строке. Запись каждой строки производится до символа пропуска.

/>

Далее сверху вниз записываем символы предыдущей строки.

Второй способ используется для тех массивов, в которых повторы не только в начале строки: используется символ r и символ конца строки k.

/>

Если массив строк одинаковой длины содержит несколько повторяющихся фрагментов в различных местах строки, то в этом случае вводятся символы, обозначающее количество пропусков и можно не использовать символ r конца строки.

Восстановление начинают с первой исходной строки, где количество пропусков определяется предыдущей строкой.

Если информация анкетного типа, записанная в алфавитно-цифровой форме, то можно использовать вместо части повторяющегося текста два значка (символа). Один из символ повтора, а второй –сколько букв пропущено при повторе.

**2.4 Зонное сжатие**

Метод зонного сжатия используют для текстовых массивов с учетом символов естественного алфавита и знаков препинания.

28 = 256

предложено использовать четыре бита (полубайт) для записи каждой буквы, тем самым создавая некоторый алфавит из шестнадцати букв, где каждая ячейка дает нам m = 24 = 16

162 = 256

Мы наши шестнадцать букв делим на некоторое количество зон:

0000 4 0100 8 1000 1 0001 5 0101 9 1001 2 0010 6 0110 A 1100 3 0011 7 0111 B 1011 C 1100 D 1101 E 1110 F 1110

Для русского алфавита достаточно 13 букв, распределенных по трем зонам.

0…С – имена букв

D…F – имена зон

С учетом вероятности появления букв друг с другом в тексте, таблица для русского алфавита выглядит следующим образом:

D E F Space З Ц 1 О У Ж 2 Е Д Х 3 А Я Ч 4 Р Ь Э 5 П Ф Ю 6 Т Ы , 7 Н Щ . 8 В Ш ; 9 И Б : A С Г ! B М К ? C Л Й -

Сжатие определяется нахождением букв в одной или в соседних зонах. И вероятность нахождения буквы где-либо по отношению к другой букве:

а – вероятность нахождения буквы в зоне D

/> <td/> />
б – вероятность нахождения буквы в зоне E

в – вероятность нахождения буквы в зоне F

Каждая буква записывается двумя символами:

номер зоны

номер буквы

если рядом стоящие буквы попадают в одну зону, то номер зоны пишется только один раз, а буквы чередуем.

В некоторых случаях можно использовать сочетание из двух и более букв в таблице, разделенные по зонам, принцип разделения можно оставить тем же.

Любой из перечисленных методов сжатия используется тогда, когда он дает максимальный эффект, который определяется величиной коэффициента сжатия или объемом сэкономленной памяти.

**3. Выбор и обоснование решения задачи**

**4. Теоретическое обоснование метода Лавинского**

Данный метод предполагает сжатие последовательности чисел путём разбиения последовательности на равномерные интервалы и отыскание числа не в его натуральном виде путем перебора всех подряд, а с помощью порядкового номера отсчета от ближайшей границы.

Сам метод состоит в следующем. Пусть имеем некоторое количество чисел М и максимальное по длине число L. Очевидно, что для хранения в натуральном виде этих чисел необходимо следующее количество ячеек

Q = M \* log 2 L (1)

Левински предложил множество М разбить на N интервалов. Интервалы между собой равные и тогда очевидно, что для хранения самих чисел необходим следующий объем памяти

Q’ = M \* log 2 (L / N) (2)

И нам необходимо хранить информацию об этих самых границах

Q” = (M – 1) \* log 2 (N — 1) (3)

В целом объем памяти необходимый для самих чисел и памяти будет следующим

Q = Q’ + Q” (4)

Взяв производную из (4),для нахождения оптимального разбиения последовательности и приравняв ее нулю, получим

N ОПТ = М / ln M (5)

Формула (5) определяет оптимальное количество интервалов. Для определения величины самого интервала мы все количество возможных вариантов относим к количеству самого интервала, то есть для определения количества символов в интервале используем следующую формулу

C = 2 [log2 N] / N (6)

Для нахождения самого интервала (номер границы) используем следующее отношение

К = Х / С (7),

где Х – искомое число в натуральном виде.

После этого само число записывается, как порядковый отсчет этой границы и определяет экономию памятикак

D Q = QИСХ – QОПТ (8)

Пусть интервал 128, Х =200, тогда К = 1.56. К лежит 2> K> 1, значит код числа составит 200 – 127= 73.

/>**5 Программная реализация**

Для разработки программы был выбран язык программирования высокого уровня Delphi 5.0 (Object Pascal).

Он весьма полно выражает идеи структурного программирования. Это проявляется в том, что Delphi может успешно использоваться для записи программ на разных уровнях ее детализации, не прибегая к помощи блок-схем или специального языка проектирования программ. Средства языка Delphi позволяют осуществлять достаточный контроль правильности использования данных различных типов и программных объектов как на этапе трансляции, так и на этапе ее выполнения.

Delphi позволяет без особых трудностей реализовать удобный пользовательский интерфейс, не прибегая к написанию низкоуровневого кода.

В программе есть так же возможность считать данные для кодирования из файла.

**Библиографический список**

Конспект лекций по дисциплине «Кодирование и защита информации”.

Березюк Н. Т., Андрющенко А. Г.,Мощинский С. С. и др. Кодирование информации – Харьков: Выща школа, 1978. – 252с.

Кузьмин И. В., Кедрус В. А. Основы теории информации и кодирования. – Киев: Выща школа, 1977. – 280 с.

Цымбал В. П. Теория информации и кодирование. Киев, ”Вища школа”, 1997, 288 с.