

Задание для групп 134, 136 на 11.11.2021

Задание пересылать мне на электронную почту: param_e@mail.ru

Задание также можно найти на google-диске по ссылке:

<https://drive.google.com/drive/folders/1QRspDnEMXttIkj8Nju51EXOzwvnSwQwE>

Конспект лекции высылаете до 20-00 часов 11.11.2021 (конспект в тетради; высылаете фотографии конспекта со своей подписью на каждом листе). Без подписи не принимаю конспект.

Ответы на вопросы в конце лекции, если они есть (ответы в тетради после конспекта).

Если конспект выслан позже указанного времени, то баллы снижаются за каждый час просрочки на 0,5.

Представление чисел в ЭВМ

Как известно, в ЭВМ применяется двоичная система счисления. Может быть доказано, что при этом на построение ЭВМ тратится наименьшее количество базовых аппаратных элементов - «вентилей». Точнее, оптимальным основанием системы счисления по критерию «минимум аппаратных расходов» является *основание натурального логарифма* $e \approx 2,72$.

Однако по ряду очевидных причин для ЭВМ принято $P=2$. Достаточно вспомнить, что одна из первых электронных ВМ ENIAC содержала 17 468 электронных ламп, имела размеры около 6 м в высоту и 30 м в длину. Обилие применяемых вакуумных ламп, габаритные размеры машины отчасти объяснялись тем, что она работала с десятичными числами.

В ЭВМ применяются две формы представления чисел:

- естественная форма, или форма с фиксированной запятой (точкой) - ФЗ (ФТ);
- нормальная форма, или форма с плавающей запятой (точкой) - ПЗ (ПТ).

Фиксированная запятая (точка). В форме представления с *фиксированной запятой (точкой)* числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Например, пусть числа представлены в десятичной системе счисления и имеют пять разрядов в целой части числа (до запятой) и пять в дробной части (после запятой). Числа, записанные в такую разрядную сетку, имеют вид: +00721,35500; +00000.00328; -10301.20260.

Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому чаще всего неприемлема при вычислениях.

Диапазон значащих чисел N в системе счисления с основанием P при наличии m разрядов в целой части и s разрядов в дробной части числа (без учета знака числа) будет таким:

$$P^{-s} \leq N \leq P^m - P^{-s}.$$

Например, при $P=2$, $m=10$ и $s=6$ числа изменяются в диапазоне $0,015 < N < 1024$. Если в результате операции получится число, выходящее за допустимые пределы, произойдет переполнение разрядной сетки, и дальнейшие вычисления теряют смысл. В современных компьютерах естественная форма представления используется как вспомогательная и только для целых чисел.

В памяти ЭВМ числа с фиксированной точкой хранятся в трех форматах:

- а) полуслово - это обычно 16 бит, или 2 байта;
- б) слово - 32 бита, или 4 байта;
- в) двойное слово - 64 бита, или 8 байтов.

Отрицательные числа с ФТ записываются в разрядную сетку и дополнительных кодах, которые образуются прибавлением единицы к младшему разряду обратного кода. Обратный код получается за меной единиц на нули, а нулей на единицы в прямом двоичном коде.

Плавающая запятая (точка). В форме представления с *плавающей запятой (точкой)* число изображается в виде двух групп цифр: мантисса; порядок.

При этом абсолютная величина мантиссы должна быть меньше 1, а порядок должен быть

целым числом. В общем виде число в форме с плавающей запятой может быть представлено так:

$$N = \pm M * P^{\pm r},$$

где M - мантисса числа ($|M| < 1$); r - порядок числа (целое число); P - основание системы счисления.

Например, приведенные ранее числа в нормальной форме запишутся следующим образом:
 $+0,721355 \times 10^3$; $+0,328 \times 10^{-3}$; $-0,103012026 \times 10^5$.

Нормальная форма представления обеспечивает большой диапазон отображения чисел и является основной в современных компьютерах. Так, диапазон значащих чисел в системе счисления с основанием P при наличии m разрядов у мантиссы и s разрядов у порядка (без учета знаковых разрядов порядка и мантиссы) будет:

$$P^{-m} * P^{-(P^s-1)} \leq N \leq (1 - P^{-m}) * P^{(P^s-1)}$$

Например, при $P=2$, $m=22$ и $s=10$ диапазон чисел простирается примерно от 10^{-300} до 10^{300} . Для сравнения: количество секунд, которые прошли с момента образования планет Солнечной системы, составляет около 10^{18} .

Следует заметить, что все числа с плавающей запятой хранятся в машине в так называемом *нормализованном* виде.

Нормализованным называют такое число, старший разряд мантиссы которого больше нуля. У нормализованных двоичных чисел, следовательно, $0,5 < M < 1$.

Нормализованные, т. е. приведенные к правильной дроби, числа: $10,35_{10} = 0,1035_{10} \times 10^{+2}$;
 $0,00007245_8 = 0,7245_8 \times 8^{-4}$; $F5C,9B_{16} = 0,F5C9B_{16} \times 16^{+3}$.

В памяти ЭВМ числа с ПТ хранятся в двух форматах:

- слово - 32 бита, или 4 байта;
- двойное слово - 64 бита, или 8 байт.

Разрядная сетка для чисел с ПТ имеет следующую структуру:

- нулевой разряд - это знак числа (0 - «минус», 1 - «плюс»);
- с 1 по 7 разряд записывается порядок в прямом двоичном коде, пустые разряды заполняются нулями. В первом разряде указывается знак порядка (1 - «плюс» или 0 - «минус»);
- с 8 по 31 (63) указывается мантисса, слева направо без нуля целых в прямом двоичном коде и для отрицательных чисел и пустые разряды заполняются нулями.

Алгебраическое представление двоичных чисел

Знак числа обычно кодируется двоичной цифрой, при этом: код 0 означает знак + (плюс); код 1 - знак - (минус).

Для алгебраического представления чисел, т. е. для представления чисел с учетом их знака, в вычислительных машинах используются *специальные коды*: *прямой код числа*; *обратный код*; *дополнительный код*.

При этом два последних кода позволяют заменить неудобную для компьютера операцию вычитания на операцию сложения с отрицательным числом. Дополнительный код обеспечивает более быстрое выполнение операций, поэтому в компьютере чаще всего применяется именно он.

Прямой код применяется при умножении и делении, обратный - при замене вычитание сложением и дополнительный - при сложении положительных и отрицательных чисел.

Прямой код двоичного числа - это само двоичное число в нормальной форме. Значение знакового разряда для положительных чисел равно 0, а для отрицательных 1.

Например. Число $+0,1101$ прямой код будет $0,1101$; число $-0,1101$ прямой код будет $1,1101$.

Обратный код для положительных чисел совпадает с прямым кодом, а для отрицательных чисел все цифры числа заменяются на противоположные значения, т.е. выполняется *инверсия*.

Например. Число $+0,1101$ обратный код будет $0,1101$; число $-0,1101$ обратный код будет $1,0010$.

Дополнительный код для положительных чисел совпадает с прямым кодом. Для отрицательных чисел дополнительный код формируется как результат суммирования обратного кода с единицей младшего разряда.

Например. Число $+0,1101$ дополнительный код будет $0,1101$; число $-0,1101$

дополнительный код будет 1,0011.

После выполнения арифметических действий дополнительный и обратный коды отрицательных чисел преобразуются в прямой код. Обратный код преобразуется заменой всех значащих цифр, кроме знакового, на противоположные цифры.

Дополнительный код преобразуется двумя способами:

- так же, как обратный код, а затем добавляется единица к младшему разряду;
- от младшего разряда отнимается единица, а затем выполняется инверсия, т.е. заменяются все значащие цифры на противоположные.

Рассмотрим операции сложения и вычитания. Сложение производится в дополнительных кодах поразрядно. Единица переноса из знакового разряда теряется. Вычитание сводится к алгебраическому сложению кода уменьшаемого с кодом вычитаемого, взятого с обратным знаком (таблица 1). Если при сложении в ЭВМ оба слагаемых имеют одинаковый знак, то операция сложения выполняется обычным путем. Если при сложении слагаемые имеют разные знаки, то сначала необходимо выявить большее по абсолютной величине число, из него произвести вычитание меньшего по абсолютной величине числа и разности присвоить знак большего числа.

Умножение и деление двоичных чисел производится в прямом коде и сводится к двум операциям: сложению и сдвигу кодов. Знак произведения и частного определяется по тем же правилам, что и в десятичной системе счисления.

Умножение

$$\begin{array}{r}
 \times 1011 \\
 1101 \\
 \hline
 1011 \\
 + 0000 \\
 + 1011 \\
 + 1011 \\
 \hline
 10001111
 \end{array}$$

Деление

$$\begin{array}{r}
 110010 \mid 1010 \\
 - 1010 \quad \mid 101 \\
 \hline
 001010 \\
 - 1010 \\
 \hline
 0000
 \end{array}$$

Пример.

Таблица 1. Правила сложения.

Обычная запись	Сложение в дополнительном коде
$ \begin{array}{r} 0,0101 \\ 0,0110 \\ \hline 0,1011 \end{array} $	$ \begin{array}{r} 0,0101 \\ 0,0110 \\ \hline 0,1011 \end{array} $
$ \begin{array}{r} -0,1110 \\ 0,0110 \\ \hline -0,1000 \end{array} $	$ \begin{array}{r} 1,0010 \\ 0,0110 \\ \hline 1,1000 \\ \text{результат преобразуется в прямой код} \\ 1,0111 \\ \quad 1 \\ \hline 1,1000 \end{array} $
$ \begin{array}{r} 0,1110 \\ -0,0110 \\ \hline 0,1000 \end{array} $	$ \begin{array}{r} 0,1110 \\ 1,1010 \\ \hline 10,1000 \\ 1 \text{ отбрасывается} \end{array} $
$ \begin{array}{r} -0,0101 \\ -0,0110 \\ \hline -0,1011 \end{array} $	$ \begin{array}{r} 1,1011 \\ 1,1010 \\ \hline 11,0101 \\ 1 \text{ отбрасывается, преобразование в прямой код} \\ 1,1010 \\ \quad 1 \\ \hline 1,1011 \end{array} $

Типы данных

Классификация информационных единиц, обрабатываемых, на ЭВМ, включает следующие аспекты:

- *типы данных*, или совокупность соглашений о программно-аппаратурной форме представления и обработки, а также ввода, контроля и вывода элементарных данных;
- *структуры данных* - способы композиции простых данных в агрегаты и операции над ними;
- *форматы файлов* - представление информации на уровне взаимодействия операционной системы с прикладными программами.

Ранние языки программирования (ЯП), а точнее - системы программирования (СП) - Фортран, Алгол, будучи ориентированы исключительно на вычисления, не содержали развитых систем типов и структур данных.

В ЯП Алгол символьные величины и переменные вообще не предусматривались, в некоторых реализациях строки (символы в апострофах) могли встречаться только в операторах печати данных.

Позже появившиеся ЯП (СП) Кобол, PL/1, Паскаль вводят новые типы данных:

- символьные (цифры, буквы, знаки препинания и пр.);
- числовые символьные для вывода;
- числовые двоичные для вычислений;
- числовые десятичные (цифры 0÷9) для вывода и вычислений.

Разновидности числовых данных здесь соответствуют внутреннему представлению и машинным (или эмулируемым) командам обработки. Кроме того, вводятся числа двойного формата (два машинных слова), для обработки которых также необходимо наличие в процессоре (или эмуляция) команд обработки чисел двойной длины (точности).

Уместно привести пример представления числовой информации в различных перечисленных формах. Пусть задано число $135_{10} = 207_8 = 87_{16} = 100000111_2$, тогда:

- внутренняя стандартная форма представления (тип BINARY для обработки в двоичной арифметике) - сохраняется (100000111_2). Объем - 1 байт, -или 8 двоичных разрядов;
- внутренняя форма двоично-десятичного представления (тип DECIMAL, каждый разряд десятичного числа представляется двоично-десятичной, в 4 бита, комбинацией). Представление 135 есть $001\ 011\ 101_2$. Объем - 2,5 байта, 12 двоичных разрядов;
- символьное представление (тип ALPHABETIC, для вывода) - каждый разряд представляется байтом в соответствии с кодом ASCII. Представление 135 есть - $00110001\ 00110011\ 00110101_2$. Объем - 3 байта.

Некоторые системы программирования поддерживают операции над комплексными числами. Очевидно, для размещения таких чисел необходим как минимум двойной расход оперативной памяти (по одному слову для размещения действительной и мнимой частей при обычной точности и по два слова при двойной точности). Кроме того, очевидно, что процессоры обычных универсальных ЭВМ вряд ли поддерживают операции над такими числами, в связи с чем операции над ними требуют написания соответствующих подпрограмм *или эмуляции комплексной арифметики*.

Появление систем управления базами данных и систем программирования для разработки ИС приводит к появлению других типов данных:

- *дата и время*;
- *бинарные* (BLOB - Binary Large Object) и *текстовые объекты* без внутренней структуры (интерпретация возлагается на прикладные программы).

Понятие типа данных ассоциируется также с допустимыми значениями переменной и операциями над ними, например, данные типа время (ЧЧ:ММ:СС) или дата (ГГ/ММ/ДЦ) предполагают определенные диапазоны значений каждого из разрядов, а также машинные или эмулируемые операции (сложение/вычитание дат и/или моментов времени).

Двоичное кодирование мультимедиа информации.

С 80-х годов 20 столетия развивается технология обработки на компьютере графической информации. Компьютерная графика широко используется в компьютерном моделировании, компьютерной анимации, деловой графике, играх, обработке видеoinформации и т.д.

Графическая информация на экране монитора представляется в виде графического

изображения, которое формируется из точек. Разрешающая способность и количество цветов зависит от видеоадаптера и могут меняться программно.

Цветные изображения имеют различные режимы: 16 цветов, 256 цветов, 65 536 цветов (high color), 16777216 цветов (true color). Очевидно, что количество бит на точку (пиксель), например, режима true color, равно:

$$I = \log_2 65536 = 16 \text{ бит} = 2 \text{ байт}$$

Рассчитаем необходимый для режима true color объем видеопамати при разрешающей возможности 800 на 600 точек, т.е. 480000 пикселей:

$$V = 2 \text{ байта} \times 480\,000 = 960\,000 \text{ байт} = 937,5 \text{ Кбайт.}$$

Аналогично рассчитывается объем видеопамати, необходимый для хранения битовой карты изображений при других видеорежимах.

В видеопамати компьютера хранится битовый план (bit map), являющийся двоичным кодом изображения, отсюда он считывается (не реже 50 раз в секунду) и отображается на экране.

Двоичное кодирование звуковой информации.

С начала 90-х гг. персональные компьютеры получают широкие возможности для работы со звуковой информацией. Каждый компьютер, имеющий звуковую плату, может сохранять звук в виде файлов и воспроизводить его. С помощью специальных программных средств (редакторов аудиофайлов) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. В дальнейшем создаются программы распознавания речи и появляется возможность голосового управления компьютером.

При двоичном кодировании аналогового звукового сигнала непрерывный сигнал дискретизируется (оцифровывается), т. е. заменяется серией отдельных выборок. Качество двоичного кодирования зависит от двух параметров: количества распознаваемых дискретных уровней сигнала и количества выборок в секунду.

Различные звуковые карты могут обеспечить как 8-, так и 16-битные выборки. При замене непрерывного звукового сигнала его дискретным представлением в виде ступенек 8-битные карты позволяют закодировать 256 различных уровней дискретизации звукового сигнала, соответственно 16-битные 65 536 уровней.

Частота дискретизации аналогового звукового сигнала (количество выборок в секунду) также может принимать различные значения (5,5, 11, 22 и 44 кГц). Таким образом, качество звука в дискретной форме может быть очень плохим (качество радиотрансляции) при 8 битах и 5,5 кГц и весьма высоким (качество аудиоCD) при 16 битах и 44 кГц.

Можно оценить объем моноаудиофайла с длительностью звучания 1 с при среднем качестве звука (16 бит, 22 кГц). Для этого 16 бит на одну выборку необходимо умножить на 22 000 выборок в секунду, что дает в результате 43 Кбайта.

Сжатие информации.

Объем обрабатываемой и передаваемой информации быстро растет. Это связано с выполнением все более сложных прикладных процессов, появлением новых информационных служб, использованием изображений и звука. \square

Сжатие данных (data compression) процесс, обеспечивающий уменьшение объема данных. Сжатие позволяет резко уменьшить объем памяти, необходимый для хранения данных, сократить (до приемлемых размеров) время их передачи. Особенно эффективно сжатие изображений. Сжатие данных может осуществляться как программным, так и аппаратным или комбинированным методом.

Сжатие текстов связано с более компактным расположением *байтов*, кодирующих символы. Определенные результаты дает статистическое кодирование, в котором наиболее часто встречающиеся символы получают коды наименьшей длины. Здесь также используется счетчик повторений пробелов. Что же касается звука и изображений, то объем представляющей их информации зависит от выбранного шага квантования и числа разрядов аналого-дискретного преобразования. В принципе, здесь используются те же методы сжатия, что и при обработке текстов. Если сжатие текстов происходит без потери информации, то сжатие звука и изображения почти всегда приводит к ее некоторой потере. Сжатие широко используется при архивировании данных.

Сжатие изображений (images compression) процесс минимизации данных, определяющих изображение. Минимизация количества информации, предоставляющей изображение или

видеофильм прежде всего, осуществляется при выборе шага квантования и разрядности кодов. При этом, естественно, происходит определенная (допустимая) потеря информации. Затем происходит сжатие изображения, представленного дискретным сигналом.

Сжатие изображения осуществляется в несколько этапов:

–изображение делится на блоки *пикселей*, каждый из которых подвергается обработке, устраняющей избыточность;

–осуществляется кодирование с переменной длиной кодов, что исключает длинные цепочки нулей и единиц в последовательностях *битов*;

–дополнительное сжатие движущегося изображения за счет сравнения каждого изображения с предыдущим, чтобы сохранять только изменившуюся его часть.

Допускается потеря информации, которая в решении поставленной задачи считается несущественной. Например, можно при обработке изображения удалить из аналогового сигнала частоты, которые находятся вне спектра, воспринимаемого глазом человека (до 10 000 цветов, 250 оттенков серого цвета). Нередко допускается игнорирование цвета каждого второго пикселя либо группа пикселей заменяется одним со средним значением цвета. Осуществляется также групповое кодирование. Его сущность заключается в кодировании групп одинаковых пикселей (например, небо без облаков на картине).

Размер файла сжатого дискретного неподвижного изображения зависит от четырех параметров: площади изображения, квадрата разрешения, числа бит необходимых для представления пикселя, и коэффициента сжатия. Выбор коэффициентов сжатия - компромисс между пропускной способностью системы (скоростью переноса файлов) и качеством восстанавливаемого изображения. Чем выше коэффициент сжатия, тем ниже это качество. При этом следует иметь в виду, что при очень высокой разрешающей способности и большом коэффициенте сжатия можно получить изображение с низкой разрешающей способностью.

Поэтому, выбор указанных параметров обосновывается технико-экономическим анализом и алгоритмом сжатия. Что касается качества изображения, то оно зависит от конкретной поставленной задачи. Например, в системах телеконференций основной объем необходимой информации содержится в речи, тогда как качество изображения может играть вторую роль.

В зависимости от скорости сжатия изображений выполняемые процессы подразделяются на два класса. К первому относится сжатие неподвижных изображений, которое может выполняться в фоновом режиме, с любой возможной скоростью. Второй класс образует алгоритмы сжатия движущихся изображений, которые должны выполняться в реальном времени по мере получения данных.

Существует немало технологий сжатия/восстановления изображений. Наиболее популярная из них предложена *Объединенной группой экспертов в области фотографии (JPEG)* и позволяет сократить размеры графического файла в 10÷20 раз. Благодаря специальным процессорам и алгоритмам удается также сжимать видеосюжеты.

Кодирование видеoinформации.

В связи с большим объемом информации, содержащейся в видеопотоке (до 6 Мбайт/с), для записи информации в ЭВМ обычно применяют сжатое кодирование потока данных на входе с использованием алгоритмов семейства MBEG/JPEG.

Стандарт MPEG (Motion Picture Expert Group) включает несколько компонент: системного потока, описывающего структуру смешанного аудио- и видеопотока, а также MPEG-video и MPEG-audio.

В случае MPEG video сжатие достигается за счет четырех факторов.

1. Использование составляющих YUV вместо обычных RGB (красный, зеленый, синий).

Вместо элементарных цветов кодируется яркость (luminance, Y) и цветность (chrominance, U&V), причем цветность «прорежена» по вертикали и горизонтали в два раза по сравнению с яркостью (децимация). При этом достигается двукратное сжатие.

2. Дискретно-косинусное преобразование с последующим квантованием.

При этом квадраты пикселей (8x8) подвергаются двумерному дискретно-косинусному преобразованию (DCT). Это преобразование переводит пространственное представление сигнала в частотное. При этом высокие пространственные частоты передаются с меньшей точностью, чем низкие частоты.

3. Устранение временной избыточности с компенсацией движения.

Это означает, что для ликвидации избыточности, заключающейся в большой корреляции между соседними кадрами, передается разность между ними. Кадры видеопотока разбиваются на несколько типов - *Intra (I)*, которые кодируются полностью, *Predicted (P)*, для которых кодируется различие с предыдущим I- или P-кадром, и *Bidirectional (B)*, для которых в качестве *опорных (reference)* используются I- и/или P-кадры, между которыми он находится. Обычно I-кадры следуют 1 или 2 раза в секунду, и между двумя опорными кадрами лежит $2 \div 4$ B-кадра.

Типичная последовательность кадров имеет вид: *IBBPBBPBBPBBIBBP*. В общем случае вид последовательности выбирается *кодеком* (кодирующее устройство или программа) и может зависеть или нет от содержания кадров. Поскольку изображение на соседних кадрах обычно сдвинуто, применяется компенсация движения, т. е. кодируется отклонение («разность») от некоторого сдвинутого опорного изображения. Кодирование выполняется макроблоками (16 x 16 яркость, 8x8 цветность), для каждого макроблока определяется свой вектор движения.

4. Квазиоптимальное кодирование.

Коэффициенты, полученные после DCT, векторы движения и все остальное, кодируются кодами переменной длины. Это кодирование называют *квазиоптимальным*, поскольку кодовая таблица не строится заново для каждого конкретного случая, а выбрана при разработке стандарта на основе анализа типичных видеопоследовательностей.

MPEG-1 проектировался из расчета на поток 1,5 Мбит/с при 30 кадрах размером 352 x 240 в секунду, хотя он не ограничен этим и допускает существенно больший поток при произвольном размере кадра.

MPEG-2 проектировался с учетом опыта использования MPEG-1 и ориентируется на вещание, так как содержит средства для маскирования ошибок.

В случае MPEG audio исходный сигнал подвергается многоканальной фильтрации. Далее амплитуды сигналов в каждой полосе сравниваются для нахождения полос, подлежащих кодированию с учетом эффекта маскирования слабого сигнала сильным. Далее амплитуда сигнала в полосе квантуется и кодируется. При записи на Video CD скорость потока звука составляет 32 Кбайт/с.