



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2016114982, 19.04.2016

(24) Дата начала отсчета срока действия патента:  
19.04.2016Дата регистрации:  
04.04.2017

Приоритет(ы):

(22) Дата подачи заявки: 19.04.2016

(45) Опубликовано: 04.04.2017 Бюл. № 10

Адрес для переписки:

127287, Москва, Старый Петровско-Разумовский  
пр-д, 1/23, стр. 1, Открытое акционерное  
общество "Информационные технологии и  
коммуникационные системы"

(72) Автор(ы):

Лапин Виталий Георгиевич (RU)

(73) Патентообладатель(и):

Открытое акционерное общество  
"Информационные технологии и  
коммуникационные системы" (RU)(56) Список документов, цитированных в отчете  
о поиске: RU 2543961 C2, 10.03.2015. US  
5793716 A, 11.08.1998. US 6044219 A,  
28.03.2000. US 2008301212 A1, 04.12.2008.

(54) Способ кодирования и вычисления даты с использованием упрощенного формата в цифровых устройствах

(57) Реферат:

Изобретение относится к вычислительной технике и может быть использовано для кодирования и преобразования даты в цифровых устройствах. Техническим результатом является увеличение диапазона возможных значений даты. Способ содержит этапы, на которых выделяют для хранения данных о дате, включающей год, месяц и день, целое число длиной K бит; устанавливают для отсчета фиксированную дату как первое января определенного года Y0; вводят

значения текущего года Y, месяца M, дня D; вычисляют целое число N для хранения данных о дате по формуле  $N=D+M*31+(Y-Y0)*372-32$ ; сохраняют число N в двоичном формате; вычисляют при необходимости дату по формулам  $D=((N \bmod 372) \bmod 31)+1$ ,  $M=((N \bmod 372) \div 31)+1$ ,  $Y=Y0+(N \div 372)$ , где div - операция целочисленного деления (деления с отбрасыванием дробной части); mod - операция взятия остатка от целочисленного деления.



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**(21)(22) Application: **2016114982, 19.04.2016**(24) Effective date for property rights:  
**19.04.2016**Registration date:  
**04.04.2017**

Priority:

(22) Date of filing: **19.04.2016**(45) Date of publication: **04.04.2017** Bull. № 10

Mail address:

**127287, Moskva, Staryj Petrovsko-Razumovskij pr-  
d, 1/23, str. 1, Otkrytoe aktsionernoe obshchestvo  
"Informatsionnye tekhnologii i kommunikatsionnye  
sistemy"**

(72) Inventor(s):

**Lapin Vitalij Georgievich (RU)**

(73) Proprietor(s):

**Otkrytoe aktsionernoe obshchestvo  
"Informatsionnye tekhnologii i  
kommunikatsionnye sistemy" (RU)**

(54) **METHOD FOR ENCODING AND COMPUTING DATE USING SIMPLIFIED FORMAT IN DIGITAL DEVICES**

(57) Abstract:

FIELD: physics.

SUBSTANCE: method comprises the stages, at which the K bit long integer is allocated for storing the date data, including year, month and day; a fixed date is set as the first of January of the certain year Y0; the values of the current year Y, month M, day D are introduced; the integer N is calculated for storing the date data by the formula  $N=D+M*31+(Y-Y0)*372-32$ ;

N number is stored in the binary format; the date is calculated, if required, by the formula  $D=((N \text{ mod } 372) \text{ mod } 31)+1$ ,  $M=((N \text{ mod } 372) \text{ div } 31)+1$ ,  $Y=Y0+(N \text{ div } 372)$ , where div - the integer division operation (division with discarding the fractional part); mod - the remainder taking operation of the integer division.

EFFECT: increasing the possible range of the date values.

Область техники, к которой относится изобретение

Предлагаемое изобретение относится к вычислительной технике и, в частности, к способам кодирования и преобразования даты для хранения и использования в цифровых устройствах: компьютерах, терминалах и пр.

5 Уровень техники

В современных цифровых устройствах для хранения и использования даты и времени применяются различные внутренние форматы данных. Для преобразования из внутреннего формата во внешний и наоборот требуется проведение ряда вычислений и операций кодирования/декодирования.

10 Так, известен способ кодирования и преобразования даты в программном обеспечении (ПО) SQL Server 2005 и SQL Server 2012 компании Microsoft [1, 2]. Для хранения даты и времени отводится два 4-байтовых целых числа (всего используется 8 байт). Первые 4 байта используются для хранения собственно даты и содержат количество дней до или после даты отсчета, в качестве которой выбрано 1 января 1900  
15 года (для типа данных `datetime`; вторые 4 байта содержат текущее значение времени, представленного в виде трехсотых долей секунды, прошедших после полуночи).  
Используется также хранение даты и времени с меньшей точностью. В этом варианте даты хранятся в виде 2-байтового целого числа, представляющего количество дней, прошедших с 1 января 1900 года (для типа данных `smalldatetime`; в двух следующих  
20 байтах хранится количество минут, прошедших после полуночи).

Таким образом, дата может храниться в виде целого числа и минимально занимать 2 байта (16 бит).

Для прямого преобразования даты по григорианскому календарю в содержимое ячейки памяти может использоваться следующий алгоритм [3]:

25  $a=(M-14) \text{ div } 12,$   
 $y=Y+4800-a,$   
 $m=M+12 \times a-3,$   
 $J=D+(153 \times m+2) \text{ div } 5+365 \times y+y \text{ div } 4-y \text{ div } 100+y \text{ div } 400-32075,$   
где  $a, y, m$  - промежуточные переменные целого типа;  
30  $M$  - месяц года, от 1 (январь) до 12 (декабрь);  
 $D$  - день месяца, от 1 до 31;  
 $Y$  - год григорианского календаря.

После этого для получения  $N$  из полученного значения  $J$  (номер юлианского дня для исходной даты) вычитают константу  $J_0$ , которая зависит от стартовой даты и является  
35 номером юлианского дня для стартовой даты. Однако для обратного преобразования целого числа  $N$  в дату используется еще большее количество преобразований и вычислений, сначала получают номер юлианского дня  $J$ , прибавляя константу  $J_0$ , далее вычисляют

40  $j=J+32044,$   
 $g=j \text{ div } 146097,$   
 $dg=j \text{ mod } 146097,$   
 $c=(dg \text{ div } 36524+1) \times 3 \text{ div } 4,$   
 $dc=dg-c \times 36524,$   
 $b=dc \text{ div } 1461,$   
45  $db=dc \text{ mod } 1461,$   
 $a=(db \text{ div } 365+1) \times 3 \text{ div } 4,$   
 $da=db - a \times 365,$   
 $y=g \times 400+c \times 100+b \times 4+a,$

$$m=(da \times 5 + 308) \text{ div } 153 - 2,$$

$$d=da - (m+4) \times 153 \text{ div } 5 + 122,$$

$$Y=y - 4800 + (m+2) \text{ div } 12,$$

$$M=(m+2) \text{ mod } 12 + 1,$$

5  $D=d+1,$

где b, c, d, j, g, da, db, dc, dg - промежуточные переменные целого типа.

Недостатками известного способа являются необходимость проводить ряд сложных вычислений и большой занимаемый объем хранимых данных.

10 В ряде цифровых устройств возникает необходимость использования только даты, включающей год, месяц и день, что позволяет снизить объем хранимых данных и упростить вычисления.

Для таких устройств может быть использован известный способ кодирования даты [4], заключающийся в том, что

15 - выделяют для хранения данных о дате, включающей год, месяц и день, целое число длиной K бит;

- устанавливают для отсчета фиксированную дату как первое января определенного года Y0;

- вводят значения текущего года Y, месяца M, дня D;

20 - вычисляют целое число N для хранения данных о дате по формуле

$$N=D+(M-1) \times 32+(Y-Y_0) \times 384;$$

- сохраняют число N в двоичном формате;

- вычисляют при необходимости дату по формулам

$$D=N \text{ mod } 32,$$

$$M=(N \text{ div } 32) \text{ div } 12 + 1,$$

25  $Y=Y_0+(N \text{ div } 384),$

где div - операция целочисленного деления (деления с отбрасыванием дробной части);

mod - операция взятия остатка от целочисленного деления.

Описанный способ принят за прототип.

Недостатком известного способа является ограничение диапазона хранимых дат.

30 Несмотря на то что способ выигрывает в простоте вычисления и компактности данных, часть возможных значений не используется.

Для сравнения рассмотрим возможный диапазон хранимых дат. В качестве первоначальной даты Y0 возьмем 01.01.1900 (1 января 1900 г. ). Размер даты 2 байта (16 бит).

35 Если использовать способ компании Microsoft, когда дата вычисляется в днях от первоначальной даты, получаем конечную дату 06.06.2079. Если учитывать только полностью входящие годы, то 31.12.2078. Таким образом, данный способ обеспечивает максимально возможный диапазон в 179 лет.

40 Если использовать способ, принятый за прототип, то получаем максимальную дату 30.08.2070. Если учитывать только полностью входящие годы, то получаем 31.12.2069. В результате, обеспечивается диапазон в 170 лет. Данный способ содержит меньше вычислений, но приходится жертвовать частью значений, поэтому диапазон становится короче на 9 лет.

Раскрытие изобретения

45 Техническим результатом является увеличение диапазона возможных значений даты.

Для этого предлагается способ, заключающийся в том, что

- выделяют для хранения данных о дате, включающей год, месяц и день, целое число длиной K бит;

- устанавливают для отсчета фиксированную дату как первое января определенного года  $Y_0$ ;

- вводят значения текущего года  $Y$ , месяца  $M$ , дня  $D$ ;

- вычисляют целое число  $N$  для хранения данных о дате по формуле

$$N=D+M*31+(Y-Y_0)*372-32;$$

- сохраняют число  $N$  в двоичном формате;

- вычисляют при необходимости дату по формулам

$$D=((N \bmod 372) \bmod 31)+1,$$

$$M=((N \bmod 372) \operatorname{div} 31)+1,$$

$$Y=Y_0+(N \operatorname{div} 372),$$

где  $\operatorname{div}$  - операция целочисленного деления (деления с отбрасыванием дробной части);

$\bmod$  - операция взятия остатка от целочисленного деления.

В предложенном способе вычисляют целое число  $N$  для хранения данных о дате по формуле

$$N=D-1+(M-1)*31+(Y-Y_0)*372.$$

Значение  $N$  представляет собой количество дней от первоначальной даты.

При вычислении  $N$  по приведенной формуле в качестве количества дней в любом

месяце берется максимально возможное значение 31. Таким образом, каждая дата

внутри года будет иметь свой постоянный порядковый номер от 1 (1 января) до  $31*12=$

372 (31 декабря) независимо от того, был ли год високосный или нет. Смещение же

между одним и тем же днем текущего и последующего месяца будет равно 31, а между

датой текущего года и такой же датой последующего года будет равно  $31 * 12=372$ .

Лишние значения дней 30 февраля, 31 февраля или же 29 февраля для года, не

являющегося високосным, не используются.

Сокращая запись для  $N$ , получаем окончательно

$$N=D+M*31+(Y-Y_0)*372-32.$$

Количество байт для хранения числа  $N$  остается таким же, как в прототипе (2 байта = 16 бит). Соответственно, максимально  $N$  содержит  $2^{16} = 65536$  значений в диапазоне от 0 до 65535.

Если  $Y_0=01.01.1900$ , то получаем

- при  $N=0$  дату 01.01.1900,

- при  $N=1$  - 02.01.1900,

- при  $N=31$  - 01.02.1900,

- при  $N=372$  - 01.01.1901 и т.д.

Используя данный метод для первоначальной даты  $Y_0=01.01.1900$ , получаем

максимальную дату 02.03.2076. Если брать только полностью входящие годы -

31.12.2075. В результате, обеспечивается диапазон в 176 лет, что на 6 лет больше, чем

у известного способа, принятого за прототип, при сравнимой скорости вычислений.

Для рассмотренного варианта, когда за первоначальную дату берется  $Y_0=01.01.1900$ ,

максимальная дата по способу от Microsoft получается 06.06.2079 (покрывается диапазон

в 65536 дней, обозначим как  $i$ ), по способу прототипа - 30.08.2070 (62334 дня, обозначим

как  $p$ ), по предложенному способу - 02.03.2076 (64345 дня, обозначим как  $v$ ). Таким

образом, способ прототипа обеспечивает  $p/i=62334/65536 \approx 95.11\%$ , а предложенный

способ  $v/i=64345/65536 \approx 98.18\%$ , от максимально возможного. При этом, если способ

прототипа теряет около 5% значений от максимально возможного, то предложенный

способ - только около 2%.

Таким образом, предложенный способ позволяет увеличить диапазон значений даты при сравнимой простоте и скорости вычислений, а также компактности хранения

данных.

Осуществление изобретения

5 Реализация предложенного способа может быть осуществлена в вычислительной системе, способной производить операции целочисленного деления, т.е. практически в любой, включая системы, работающие, например, под управлением операционной системы MS Windows.

Для реализации способа достаточно реализовать выполнение предложенных операций на каком-либо языке программирования.

Например, на языке C# реализация может выглядеть следующим образом:

10

15

20

25

30

35

40

45

```
// В качестве стартовой даты берем 1 января 1900 г.  
const int StartYear = 1900;  
// Внешнее представление даты - отдельно год, месяц и день.  
5 public struct Date  
  {  
    public Date(int year, int month, int day) : this()  
10  {  
        Year = year;  
        Month = month;  
15        Day = day;  
    }  
    public int Year { get; set; }  
    public int Month { get; set; }  
20    public int Day { get; set; }  
  }  
  // Получаем внутреннее представление по внешнему  
25 private static UInt16 PackDate(Date date)  
  {  
    return (UInt16)(date.Day + date.Month * 31 + (date.Year - StartYear) * 372 - 32);  
30  }  
  // Получаем внешнее представление по внутреннему вариант 1  
  private static Date UnpackDate_1(UInt16 value)  
  {  
35    int y = StartYear + value / 372;  
    int m = (value % 372) / 31 + 1;  
    int d = (value % 372) % 31 + 1;  
40    return new Date(y, m, d);  
  
45
```

```

    }
    // Получаем внешнее представление по внутреннему вариант 2
private static Date UnpackDate_2(UInt16 value)
5
    {
    int y = StartYear + value / 372;
    int tmp1 = value % 372;
10
    int m = tmp1 / 31 + 1;
    int d = tmp1 % 31 + 1;
    return new Date(y, m, d);
15
    }

```

Реализовать действия предложенного способа в составе программы или функции может специалист в области программирования (программист). Наибольший эффект применение предложенного способа может дать в системах, обрабатывающих данные о датах и имеющих сильные ограничения на объем памяти и аппаратные ресурсы, а также передающие информацию о датах в условиях ограничения на объем передаваемой информации.

Примерами таких систем могут служить системы, хранящие и/или передающие различную информацию, в том числе и даты (например, даты рождения человека, дата отправки груза, дата отправления поезда и т.п.) в виде штрих-кода. В этом случае каждый сэкономленный бит имеет значение.

Эффективно также применение способа в системах и базах данных, в которых имеется большой объем данных с датами, в этом случае может достигаться значительная экономия объема хранения.

#### Источники информации

- 30 1. Дата и время (Transact-SQL), статья по адресу [https://msdn.microsoft.com/ru-ru/library/ms187819\(%D1%83=sq1.90\).aspx](https://msdn.microsoft.com/ru-ru/library/ms187819(%D1%83=sq1.90).aspx)
2. Datetime (Transact-SQL), статья по адресу [https://msdn.microsoft.com/ru-ru/library/ms187819\(%D1%83=sq1.110\).aspx](https://msdn.microsoft.com/ru-ru/library/ms187819(%D1%83=sq1.110).aspx)
3. Julian day, статья по адресу [http://en.wikipedia.org/wiki/Julian\\_day](http://en.wikipedia.org/wiki/Julian_day)
- 35 4. Патент РФ №2543961, приоритет от 16.01.2013 г.

#### (57) Формула изобретения

Способ кодирования и вычисления даты с использованием упрощенного формата в цифровых устройствах, заключающийся в том, что

40 выделяют для хранения данных о дате, включающей год, месяц и день, целое число длиной K бит;

устанавливают для отсчета фиксированную дату как первое января определенного года Y0;

45 вводят значения текущего года Y, месяца M, дня D;

вычисляют целое число N для хранения данных о дате по формуле  $N = D + M * 31 + (Y - Y0) * 372 - 32$ ;

сохраняют число N в двоичном формате;



вычисляют при необходимости дату по формулам

$$D = ((N \bmod 372) \bmod 31) + 1,$$

$$M = ((N \bmod 372) \div 31) + 1,$$

$$Y = Y_0 + (N \div 372),$$

5 где  $\div$  - операция целочисленного деления (деления с отбрасыванием дробной части);  
 $\bmod$  - операция взятия остатка от целочисленного деления.

10

15

20

25

30

35

40

45