ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 9834-8— 2011

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПРОЦЕДУРЫ РАБОТЫ УПОЛНОМОЧЕННЫХ ПО РЕГИСТРАЦИИ ВОС

Часть 8

Создание, регистрация универсально уникальных идентификаторов (УУИд) и их использование в качестве компонентов идентификатора объекта АСН.1

ISO/IEC 9834-8:2005
Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities — Part 8: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components (IDT)

Издание официальное

Москва
Стандартинформ
2012

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Федеральным государственным унитарным предприятием «Государственный научно-исследовательский и конструкторско-технологический институт «ТЕСТ» (ФГУП «ГосНИИ «ТЕСТ») на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 7 сентября 2011 г. № 256-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 9834-8:2005 «Информационная технология. Взаимосвязь открытых систем. Процедуры работы уполномоченных по регистрации ВОС. Часть 8. Создание, регистрация универсально уникальных идентификаторов (УУИд) и их использование в качестве компонентов идентификатора объекта АСН.1» (ISO/IEC 9834-8:2005 «Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities — Part 8: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5—2004 (подраздел 3.5).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2012

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
2.1 Идентичные рекомендации | международные стандарты . . . 1
2.2 Другие нормативные ссылки . . . 2
3 Термины и определения . . . 2
3.1 Нотация АСН.1 . . . 2
3.2 Уполномоченные по регистрации . . . 2
3.3 Термины сети . . . 2
3.4 Дополнительные термины и определения . . . 2
4 Сокращения . . . 3
5 Обозначения . . . 3
6 Структура и представление УУИд . . . 3
6.1 Структура полей УУИд . . . 3
6.2 Двоичное представление . . . 4
6.3 Представление в виде единственного целого значения . . . 4
6.4 Шестнадцатиричное представление . . . 4
6.5 Формальный синтаксис шестнадцатиричного представления . . . 4
7 Использование УУИд для образования ИДО . . . 5
8 Использование УУИд для образования ЕИР . . . 5
9 Правила сравнения и упорядочения УУИд . . . 5
10 Валидация . . . 6
11 Биты варианта . . . 6
12 Использование полей УУИд и порядок передачи байтов . . . 6
12.1 Общие положения . . . 6
12.2 Версия . . . 7
12.3 Время . . . 8
12.4 Временная последовательность . . . 8
12.5 Узел . . . 8
13 Установка полей в основанном на времени УУИд . . . 9
14 Установка полей основанного на имени УУИд . . . 9
15 Установка полей основанного на случайном числе УУИд . . . 10
16 Регистрация УУИд и их использование в качестве компонентов ИДО . . . 10
16.1 Дерево ИДО АСН.1 . . . 10
16.2 Назначение уполномоченных по регистрации . . . 11
16.3 Оплата . . . 11
16.4 Процедуры регистрации . . . 12
16.4.1 Заявление на регистрацию . . . 12
16.4.2 Процесс подтверждения . . . 12

16.4.3 Содержание заявления 12
16.5 Сопровождение регистра в сети 12
Приложение А (справочное) Алгоритмы эффективного создания основанных на времени УУИд . . . 13
Приложение В (справочное) Свойства основанного на имени УУИд 15
Приложение С (справочное) Генерация случайных чисел в системе 16
Приложение D (справочное) Пример реализации 17
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации (и действующим в этом качестве межгосударственным стандартам) 27
Библиография. 28

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПРОЦЕДУРЫ РАБОТЫ УПОЛНОМОЧЕННЫХ ПО РЕГИСТРАЦИИ ВОС

Часть 8

Создание, регистрация универсально уникальных идентификаторов (УУИд) и их использование в качестве компонентов идентификатора объекта АСН.1

Information technology. Open systems interconnection. Procedures for the operation of OSI registration authorities. Part 8. Generation and registration of universally unique identifiers (UUIDs) and their use as ASN.1 object identifier components

Дата введения — 2012—06—01

1 Область применения

Настоящий стандарт устанавливает формат и правила создания, позволяющие пользователям образовывать 128-битные идентификаторы, которые являются либо гарантированно глобально уникальными, либо глобально уникальными с высокой вероятностью.

Универсально уникальные идентификаторы (УУИд), созданные в соответствии с требованиями настоящего стандарта, пригодны для временного использования с созданием УУИд каждые 100 наносекунд или в качестве постоянных идентификаторов.

Настоящий стандарт разработан на основе более ранних нестандартизированных УУИд и методов их создания и технически идентичен этим ранним спецификациям.

Настоящий стандарт устанавливает основанные на сетевых технологиях процедуры работы уполномоченных по регистрации УУИд.

Настоящий стандарт также устанавливает и допускает использование УУИд (зарегистрированных или незарегистрированных) в качестве компонентов ИДО ниже дуги {joint-iso-itu-t uuid(25)}, что позволяет пользователям создавать ИДО без каких-либо процедур регистрации.

Настоящий стандарт также устанавливает и допускает использование УУИд (зарегистрированных или незарегистрированных) для создания ЕИР.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие международные стандарты:

2.1 Идентичные рекомендации | международные стандарты

Рекомендация МСЭ-Т X.660 (2004) | ИСО/МЭК 9834-1:2004 Информационная технология. Взаимосвязь открытых систем. Процедуры работы полномочных органов регистрации ВОС. Часть 1. Общие процедуры

Рекомендация МСЭ-Т X.680 (2002) | ИСО/МЭК 8824-1:2002 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации

Издание официальное

2.2 Другие нормативные ссылки

ИСО/МЭК 8802-3:2000 Информационная технология. Телекоммуникации и информационный обмен между системами. Локальные и городские сети. Специфические требования. Часть 3: Метод многостанционного доступа с обнаружением несущей и предотвращением конфликтов (CSMA/CD) и спецификации физического уровня

ИСО/МЭК 10118-3:2004 Информационная технология. Методы безопасности. Функции хэширования. Часть 3. Специальные функции хэширования

ИСО/МЭК 10646:2003 Информационная технология. Универсальный многооктетный набор кодовых символов (UCS)

FIPS PUB 180-2:2002 Федеральный стандарт обработки информации. Стандарт безопасной хэш-функции (SHS)

IETF RFC 1321 (1992) Алгоритм обработки сообщений MD5

IETF RFC 2141 (1997) Синтаксис URN

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

3.1 Нотация ACH.1

В настоящем стандарте применены следующие термины, определенные в Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1:

a) **всемирное скоординированное время** (Coordinated Universal Time — UTC);

b) **идентификатор объекта** (ACH.1).

3.2 Уполномоченные по регистрации

В настоящем стандарте применены следующие термины, определенные в Рекомендации МСЭ-Т X.660 | ИСО/МЭК 9834-1:

a) **дерево идентификаторов объектов** (или **дерево ИДО**);

b) **регистрация**;

c) **уполномоченный по регистрации**;

d) **процедуры регистрации**.

3.3 Термины сети

В настоящем стандарте применен следующий термин, определенный в ИСО/МЭК 8802-3:

a) **адрес КДС**.

3.4 Дополнительные термины и определения

3.4.1 **случайное число криптографического качества** (cryptographicquality random-number): Случайное или псевдослучайное число, созданное способом, который гарантирует достаточно протяженный диапазон повторно сгенерированных значений для того, чтобы способ был приемлемым для использования в криптографических целях (и используется в этих целях).

3.4.2 **версия, основанная на имени** (name-based version): УУИд, созданный с использованием криптографического хеширования имени пространства имен и имени в этом пространстве имен.

3.4.3 **пространство имен** (name space): Система для создания имен объектов, которая гарантирует недвусмысленную идентификацию в пределах этого пространства имен.

Примечание — Примерами пространств имен являются система имен области сети, ЕИР, ИДО, отличающие имена справочника (см. [5]) и зарезервированные слова в языках программирования.

3.4.4 **версия, основанная на случайном числе** (random-number-based version): УУИд, созданный с использованием случайного или псевдослучайного числа.

3.4.5 **стандартный вариант УУИд** (standard UUID variant): Вариант возможного формата УУИд, установленный настоящим стандартом.

Примечание — Ранее существовали другие спецификации форматов УУИд, отличные от варианта, установленного в настоящем стандарте. УУИд, созданные в соответствии с этими вариантами форматов, являются различными.

3.4.6 **версия, основанная на времени** (time-based version): УУИд, в котором уникальность достигается использованием адреса КДС для идентификации системы, а значение времени основано на текущем времени UTC.

3.4.7 **универсально уникальный идентификатор; УУИд** (Universally Unique Identifier; UUID): 128-битовое значение, созданное в соответствии с требованиями настоящего стандарта или в соответствии с одной из ранних спецификаций и являющееся уникальным в различных системах и во времени (см. также 3.4.5).

4 Сокращения

В настоящем стандарте применены следующие сокращения:

АСН.1 — абстрактная синтаксическая нотация версии 1;

ГУИД — глобально уникальный идентификатор;

ЕИР (URN) — универсальное имя ресурса (Uniform Resource Name);

ИДО — идентификатор объекта АСН.1;

КДС — контроль доступа к среде;

ОС5 — алгоритм 5 обработки сообщений;

УР — уполномоченный по регистрации;

УПР (URL) — универсальное положение ресурса (Uniform Resource Locator);

УУИд (UUID) — универсально-уникальный идентификатор (Universally Unique Identifier);

ХАБ-1 — хеш-алгоритм безопасности 1;

IEEE — Institute of Electrical and Electronics Engineers, Inc. — Институт инженеров-электриков и электронщиков;

UTC — Coordinated Universal Time — всемирное скоординированное время.

5 Обозначения

5.1 В настоящем стандарте установлена последовательность октетов для УУИд в терминах первого и последнего октетов. Первый октет также называется «октет 15», а последний — «октет 0».

5.2 Биты внутри УУИд также пронумерованы от «бит 127» до «бит 0», где бит 127 — старший бит октета 15, а бит 0 — младший бит октета 0.

5.3 На рисунках и в таблицах, представленных в настоящем стандарте, старший октет (и старший бит) помещается слева, что соответствует порядку передачи октетов, при котором самый левый октет передается первым.

5.4 Числовые значения, использованные в настоящем стандарте, выражены как целые без знака заданной длины в битах (например, N). Биты в N-битовом целом без знака значения нумеруются от «бит N-1» до «бит 0», где бит N-1 — старший, а бит 0 — младший.

5.5 Эти обозначения используются исключительно для целей настоящей спецификации. Представление в памяти компьютера не стандартизируется и зависит от системной архитектуры.

6 Структура и представление УУИд

6.1 Структура полей УУИд

6.1.1 УУИд определен как упорядоченная последовательность из шести полей. УУИд определен в терминах сцепления этих полей УУИд. Поля УУИд называются следующим образом:

a) поле «TimeLow»;

b) поле «TimeMid»;

c) поле «VersionAndTimeHigh»;

d) поле «VariantAndClockSeqHigh»;

e) поле «ClockSeqLow»;

f) поле «Node».

6.1.2 По определению поля УУИд имеют старшинство в перечисленном выше порядке, где «TimeLow» является самым старшим полем (бит 31 поля «TimeLow» является битом 127 УУИд), а «Node» — самым младшим (бит 0 поля «Node» является битом 0 УУИд).

6.1.3 Содержимое этих полей УУИд определено в терминах целых без знака значений версии, варианта, времени, временной последовательности и узла (фиксированной длины в битах каждое). Установка этих значений представлена в разделе 12, а их отображение в указанные выше поля УУИд — в 12.1.

Примечание — В качестве частей имен некоторые из полей УУИд (например, TimeLow, TimeMid и TimeHigh) подразумевают, что последовательный порядок битов в УУИд (от бита 127 до бита 0), полученный из частичного значения целого без знака (например, от бита 59 до бита 0 значения времени Time), не является тем же, что и последовательный порядок битов в этом целом без знака значения. Такой порядок установился исторически.

6.2 Двоичное представление

6.2.1 УУИд должен быть представлен в виде двоичного значения из 16 октетов, образованного сцеплением в один или несколько октетов, закодированных целыми без знака значениями фиксированной длины каждого из его полей. Для каждого поля должно использоваться следующее число октетов:

a) поле «TimeLow» — четыре октета;
b) поле «TimeMid» — два октета;
c) поле «VersionAndTimeHigh» — два октета;
d) поле «VariantAndClockSeqHigh» — один октет;
e) поле «ClockSeqLow» — один октет;
f) поле «Node» — шесть октетов.

Примечание — Такой порядок полей УУИд является обычным представлением в компьютерных системах и шестнадцатиричном текстовом представлении (см. 6.4).

6.2.2 Старший значащий бит целого без знака каждого поля УУИд должен быть старшим битом его первого октета (октета N — самого старшего октета), а младший бит целого без знака кодирования должен быть младшим битом последнего октета (октета 0 — самым младшим битом).

6.2.3 Поля УУИд должны быть сцеплены в порядке старшинства (см. 6.1.2), где самое старшее поле является первым, а самое младшее — последним.

6.3 Представление в виде единственного целого значения

УУИд может быть представлен в виде единственного целого значения. Для получения единственного целого значения УУИд 16 октетов двоичного представления следует трактовать как целое без знака кодирование, в котором самым старшим битом целого кодирования является самый старший бит (бит 7) первого из 16 октетов (октета 15), а самым младшим битом — самый младший бит (бит 0) последнего из 16 октетов (октет 0).

Примечание — Единственное целое значение используется тогда, когда УУИд образует часть ИДО, как определено в разделе 7.

6.4 Шестнадцатиричное представление

В шестнадцатиричном формате октеты двоичного формата должны быть представлены в виде строки шестнадцатиричных цифр, с использованием двух шестнадцатиричных цифр на каждый октет двоичного формата, первая из которых является значением четырех старших битов октета 15, вторая — четырех младших битов октета 15 и так далее до последней цифры, которая является значением младших битов октета 0 (см. 6.5). Между шестнадцатиричными представлениями каждой пары смежных полей должен быть вставлен символ HYPHEN-MINUS (45) (см. ИСО/МЭК 10646), за исключением полей «VariantAndClockSeqHigh» и «ClockSeqLow» (см. пример в разделе 8).

6.5 Формальный синтаксис шестнадцатиричного представления

6.5.1 Формальная спецификация синтаксиса шестнадцатиричного представления УУИд дана с помощью расширенной нотации БНФ, определенной в Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1, раздел 5, за исключением того, что отсутствуют пробелы между лексическими единицами.

6.5.2 В спецификации БНФ используется лексическая единица «hexdigit» и определена следующим образом:

Название лексической единицы — hexdigit.

Лексическая единица «hexdigit» должна состоять из одного из следующих символов:

A B C D E F a b c d e f 0 1 2 3 4 5 6 7 8 9

6.5.3 Шестнадцатиричное представление УУИд должно быть следующей продукцией «UUID»:

```
UUID ::=
        TimeLow
        "-" TimeMid
        "-" VersionAndTimeHigh
        "-" VariantAndClockSeqHigh ClockSeqLow
        "-" Node
        TimeLow ::=
```

```
        HexOctet HexOctet HexOctet HexOctet
    TimeMid ::=
        HexOctet HexOctet
    VersionAndTimeHigh ::=
        HexOctet HexOctet
    VariantAndClockSeqHigh ::=
        HexOctet
    ClockSeqLow ::=
        HexOctet
    Node ::=
        HexOctet HexOctet HexOctet HexOctet HexOctet HexOctet
    HexOctet ::=
        hexdigit hexdigit
```

6.5.4 Программное обеспечение, создающее шестнадцатиричное представление УУИд, не должно использовать (прописные) буквы верхнего регистра.

Примечание — Рекомендуется, чтобы шестнадцатиричное представление, используемое в любых человекочитаемых форматах, было ограничено строчными буквами. Однако программное обеспечение, обрабатывающее это представление, обязательно должно принимать как строчные, так и прописные буквы, как определено в 6.5.2.

7 Использование УУИд для образования ИДО

ИДО, образованный с использованием УУИд, должен иметь вид:

{joint-iso-itu-t uuid(25) <uuid-single-integer-value>}

где <uuid-single-integer-value> — единственное целое значение УУИд, определенное в 6.3.

Примечание — Подробности использования УУИд для образования ИДО приведены в разделе 16, а в 16.1.3 приведено руководство по обеспечению уникальности ИДО, созданных таким методом.

8 Использование УУИд для образования ЕИР

ЕИР (см. IETF RFC 2141), образованное с использованием УУИд, должно быть строкой, состоящей из «urn:uuid:» с последующим шестнадцатиричным представлением УУИд, определенным в 6.4.

Пример — Примером строки, представляющей собой УУИд в качестве ЕИР, является

urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6

Примечание — Существует альтернативный формат ЕИР (см. [6]), но он не рекомендуется для ЕИР, созданных с использованием УУИд. В этом альтернативном формате используется единственное целое значение УУИд, определенное в 6.3, и приведенный выше пример выглядит как «urn:oid:2.25.329800735698586629295641978511506172918».

9 Правила сравнения и упорядочения УУИд

9.1 Для сравнения двух УУИд значения соответствующих полей (см. 6.1) каждого УУИд сравниваются в порядке старшинства (см. 6.1.2). Два УУИд эквивалентны тогда и только тогда, когда эквивалентны все соответствующие поля.

Примечание 1 — Этот алгоритм сравнения двух УУИд эквивалентен сравнению значений представлений в виде единственного целого, определенного в 6.3.

Примечание 2 — В этом сравнении используются физические поля, определенные в 6.1.1, а не значения, перечисленные в 6.1.3 и определенные в разделе 12 (время, временная последовательность, вариант, версия и узел).

9.2 УУИд считается больше другого УУИд, если имеет большее значение самого старшего поля, которым эти два УУИд различаются.

9.3 При лексикографическом упорядочении шестнадцатиричных представлений УУИд (см. 6.4) больший УУИд должен следовать за меньшим УУИд.

10 Валидация

Кроме определения того, корректно ли установлены биты варианта и не относится ли к будущему значение времени, использованное в основанном на времени УУИд (и которое, следовательно, еще не могло быть использовано), отсутствуют способы определения валидности УУИд в каком-либо реальном смысле, и могут встретиться все возможные значения УУИд.

11 Биты варианта

11.1 Биты варианта являются тремя самыми старшими битами (биты 7, 6 и 5) октета 7, который, в свою очередь, является самым старшим октетом поля «VariantAndClockSeqHigh».

11.2 Все УУИд, соответствующие требованиям настоящего стандарта, должны иметь следующие биты варианта: бит 7 октета 7 равен 1, бит 6 октета 7 равен 0. Бит 5 октета 7 является самым старшим битом временно́й последовательности и должен быть установлен в соответствии с 12.4.

Примечание — Бит 5 приведен здесь как бит варианта потому, что его значением различаются исторически сложившиеся форматы. Строго говоря, он не является частью значения варианта по настоящему стандарту, в котором для варианта используются только два бита.

11.3 Информация об использовании других значений битов варианта приведена в таблице 1.

Таблица 1 — Использование битов варианта

Бит 7	Бит 6	Бит 5	Описание
0	—	—	Зарезервировано для обеспечения обратной совместимости с компанией NCS
1	0	—	Вариант, определенный в настоящем стандарте
1	1	0	Зарезервировано для обеспечения обратной совместимости с компанией Microsoft
1	1	1	Зарезервировано для последующего использования в настоящем стандарте

12 Использование полей УУИд и порядок передачи байтов

12.1 Общие положения

12.1.1 Положения и краткое описание назначения полей УУИд в двоичном представлении приведены в таблице 2.

Таблица 2 — Положения и назначение полей УУИд

Поле	Номера октетов в УУИд	Описание
«TimeLow»	15—12	Младшие биты значения времени (32 бита)
«TimeMid»	11—10	Средние биты значения времени (16 битов)
«VersionAndTimeHigh»	9—8	Версия (4 бита) с последующими старшими битами значения времени (12 битов)
«VariantAndClockSeqHigh»	7	Биты варианта (2 бита) с последующими старшими битами временно́й последовательности (6 битов)
«ClockSeqLow»	6	Младшие биты временно́й последовательности (8 битов)
«Node»	5—0	Узел (см. 12.5) (48 битов)

12.1.2 Положение полей УУИд в двоичном представлении показано на рисунке 1.

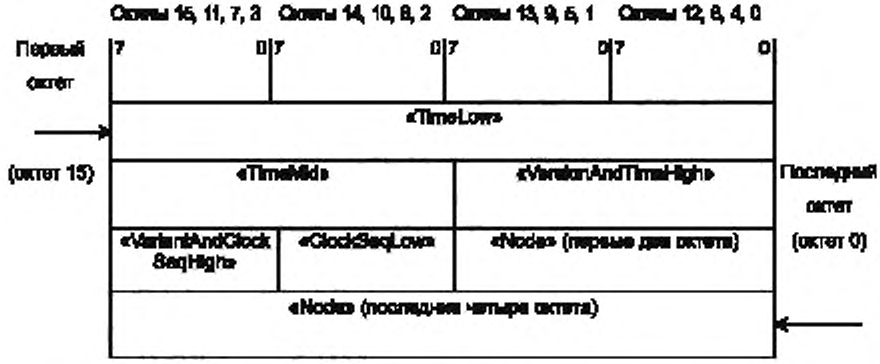

Рисунок 1 — Положение полей УУИд в двоичном представлении

12.1.3 Рекомендуется, чтобы это двоичное представление использовалось при передаче с помощью средств коммуникации, когда 16 октетов двоичного представления передаются как непрерывный набор битов с первым октетом (октет 15), предшествующим последнему (октет 0).

Примечание 1 — Порядок битов в октете определяется спецификацией метода коммуникации.

Примечание 2 — Использование 16 последовательных октетов в указанном выше порядке для передачи УУИд, но спецификации протоколов могут использовать альтернативные способы передачи УУИд, включая фрагментацию или передачу только частей УУИд (таких, как части, входящие в значение времени).

12.2 Версия

12.2.1 Идентифицировано три альтернативных способа создания УУИд (основанный на времени, имени и случайном числе) и они различаются по четырем самым старшим битам поля «VersionAndTimeHigh» (биты с 7-го по 4-й октета 9 УУИд). УУИд, созданные с использованием этих разных способов, называются «разными версиями УУИд».

Примечание — Термин «разные версии УУИд» отчасти вводит в заблуждение, но используется как исторически сложившийся. Для форматов УУИд нет традиционного понятия «номер версии» в тех случаях, когда новые версии могут быть определены при пересмотре настоящего стандарта. Новые форматы УУИд, которые могут потребоваться в будущем, будут идентифицироваться различными значениями битов варианта.

12.2.2 Определенные в настоящее время «версии УУИд», использующие первые четыре бита поля «VersionAndTimeHigh» (биты с 7-го по 4-й октета 9 УУИд), приведены в таблице 3. В таблице 3 также присвоены целые значения «версии» для каждой комбинации битов.

Примечание — Значение версии 2 не используется для совместимости с исторически сложившимися определениями УУИд. Значения версий 0 и с 6-й до 15-й зарезервированы для последующего использования.

Таблица 3 — Определенные в настоящее время версии УУИд

Бит 7	Бит 6	Бит 5	Бит 4	Значение версии	Описание
0	0	0	1	1	Основанная на времени версия, определенная в настоящем стандарте (см. раздел 13)
0	0	1	0	2	Зарезервировано для версии безопасности распределенной вычислительной среды (Distributed Computing Environment — DCE) со встроенными УУИд POSIX
0	0	1	1	3	Основанная на имени версия, определенная в настоящем стандарте (см. раздел 14)
0	1	0	0	4	Основанная на случайном числе версия, определенная в настоящем стандарте (см. раздел 15)
0	1	0	1	5	Основанная на имени версия, определенная в настоящем стандарте, с хешированием ХАБ-1 (см. раздел 14)

12.3 Время

12.3.1 Время должно быть 60-битовым значением.

Примечание — Термин «время» соответствует основанной на времени версии УУИд (версии 1), но используется также и для содержимого соответствующих значений в других версиях УУИд (версиях 3 и 4).

12.3.2 Для основанной на времени версии УУИд время должно отсчитываться в 100 наносекундных интервалах всемирного скоординированного времени (UTC), начиная от полуночи 15 октября 1582 г. (дата григорианской реформы христианского календаря).

Примечание 1 — До появления Всемирного бюро времени (Bureau International de l'Heure) каждая минута содержала ровно 60 с. Затем, по мере необходимости, вводились дополнительные секунды, увеличивающие (или потенциально уменьшающие) число секунд в году.

Примечание 2 — У переносимых систем могут быть проблемы с определением времени UTC, т. к. часто они замкнуты на локальное время своей домашней базы. При условии, что они непрерывно используют это локальное время или изменяют значение временно́й последовательности (см. 12.4), созданные ими УУИд будут оставаться уникальными.

Примечание 3 — Для систем, которые не имеют доступа к широковещательным сигналам времени, может использоваться местное время системных часов с добавлением разницы во времени при условии, что УУИд не создаются в период сезонного времени или изменения значения временно́й последовательности (см. 12.4).

12.3.3 Для основанной на имени версии УУИд время должно быть 60-битовым значением, построенным из глобально уникального имени, как определено в разделе 14.

Примечание — Примерами глобально уникальных имен являются ИДО, ЕИР и отличающие имена справочника (см. [5]).

12.3.4 Для основанной на случайном числе версии УУИд время должно быть случайным или псевдослучайным 60-битовым значением, созданным как определено в разделе 15.

12.4 Временна́я последовательность

12.4.1 Для основанной на времени версии УУИд временна́я последовательность используется с целью избежать дублирований, которые могут возникнуть, если устанавливается прошедшее значение времени или изменяется значение узла.

Примечание — Термин «временна́я последовательность» полностью соответствует основанной на времени версии УУИд, но используется также и для содержимого соответствующих значений в основанных на имени и на случайном числе версиях УУИд.

12.4.2 Если устанавливается прошедшее значение времени или оно могло быть установлено (например, пока система была выключена), то генератору УУИд не может быть известно, был ли уже сгенерирован УУИд со значением времени, большим, чем то значение времени, которое устанавливается. В такой ситуации должна быть изменена временна́я последовательность.

Примечание — Если известно предыдущее значение временно́й последовательности, то оно должно быть увеличено, иначе это значение должно быть установлено равным случайному или псевдослучайному значению криптографического качества.

12.4.3 Аналогично значение временно́й последовательности должно быть изменено, если изменилось значение узла (например, из-за того, что сетевая карта была перенесена на другую машину).

12.4.4 Временна́я последовательность должна быть первоначально (т. е. один раз за время жизни системы, создающей УУИд) проинициализирована случайным числом, полученным не на основе значения узла.

Примечание — Это делается для минимизации корреляции между системами, тем самым обеспечивая максимальную защиту от быстрого перемещения или переключения от системы к системе адресов КДС.

12.4.5 Для основанной на имени версии УУИд временна́я последовательность должна быть 14-битовым значением, построенным из имени, как определено в разделе 14.

12.4.6 Для основанной на случайном числе версии УУИд временна́я последовательность должна быть случайным или псевдослучайным 14-битовым значением, созданным так, как определено в разделе 15.

12.5 Узел

12.5.1 Для основанной на времени версии УУИд значение узла должно состоять из адреса КДС (см. ИСО/МЭК 8802-3), который обычно является адресом хоста некоторого сетевого интерфейса.

12.5.2 Для систем с кратными адресами КДС может использоваться любой доступный адрес, за исключением многоканального. Октет 5 УУИд (первый октет поля «Node») должен быть установлен равным первому октету адреса КДС, который передается системой, соответствующей ИСО/МЭК 8802-3.

Примечание 1 — Этот октет содержит биты глобальный/локальный и одноканальный/многоканальный. Требуется, чтобы бит одноканальный/многоканальный был установлен как одноканальный для того, чтобы избежать конфликта с адресами, созданными в соответствии с 12.5.3.

Примечание 2 — От уполномоченного по регистрации адресов КДС можно получить блок адресов КДС (см. [4]).

12.5.3 Для систем без адреса КДС может быть использовано случайное или псевдослучайное число криптографического качества (см. приложение С). Для таких адресов должен быть установлен многоканальный бит.

Примечание — Этот алгоритм гарантирует, что сгенерированные адреса никогда не будут конфликтовать с адресами, полученными от сетевой карты, как определено в 12.5.2.

12.5.4 Для основанного на имени УУИд значение узла должно быть 48-битовым значением, построенным путем канонизации и хеширования из глобально уникального имени, как определено в разделе 14.

12.5.5 Для основанного на случайном числе УУИд значение узла должно быть 48-битовым случайным или псевдослучайным значением, сгенерированным, как определено в разделе 15.

13 Установка полей в основанном на времени УУИд

Поля в основанном на времени УУИд должны быть установлены следующим образом:

- определяют значения для основанных на UTC времени и временно́й последовательности, которые должны быть использованы в УУИд, как определено в 12.3 и 12.4;
- для целей настоящего алгоритма рассматривают время как 60-битовое целое без знака, а временну́ю последовательность — как 14-битовое целое без знака. Биты в каждом значении считают от младшего бита, который считают нулевым;
- устанавливают поле «TimeLow» равным 32 младшим битам (битам с 31-го по 0-й) времени в том же порядке значимости;
- устанавливают поле «TimeMid» равным битам времени с 47-го по 32-й в том же порядке значимости;
- устанавливают 12 младших битов (биты с 11-го по 0-й) поля «VersionAndTimeHigh» равными битам времени с 59-го по 48-й в том же порядке значимости;
- устанавливают четыре старших бита (биты с 15-го по 12-й) поля «VersionAndTimeHigh» равными четырехбитовому номеру версии, определенному в 12.2;
- устанавливают поле «ClockSeqLow» равным восьми младшим битам (битам с 7-го по 0-й) временно́й последовательности в том же порядке значимости;
- устанавливают шесть младших битов (биты с 5-го по 0-й) поля «VariantAndClockSeqHigh» равными шести старшим битам (биты с 13-го по 8-й) временно́й последовательности в том же порядке значимости;
- устанавливают два старших бита (биты 7 и 6) поля «VariantAndClockSeqHigh» равными 1 и 0 соответственно;
- устанавливают поле узла равным 48-битовому адресу КДС в том же порядке значимости.

14 Установка полей основанного на имени УУИд

В настоящем разделе определены процедуры создания основанного на имени УУИд. В подразделе 14.1 определены общие процедуры для произвольной функции хеширования (см. ИСО/МЭК 10118-3). В подразделе 14.2 определено использование ОС5, а в 14.3 — ХАБ-1.

Примечание — Использование ОС5 ограничено случаями, требующими обратной совместимости с существующими УУИд, тогда как ХАБ-1 обеспечивает алгоритм хеширования с наименьшей вероятностью того, что одно и то же значение будет появляться при хешировании разных данных (см. раздел С.4, приложение С).

14.1 Поля основанного на имени УУИд должны быть установлены следующим образом:

- выделяют УУИд для использования в качестве «идентификатора пространства имен» для всех УУИд, создаваемых из имен этого пространства.

Примечание — В разделе D.9 приложения D рекомендованы УУИд для применения в четырех обычно используемых пространствах имен.

- преобразуют имя в каноническую последовательность октетов (как определено стандартами или соглашениями для данного пространства имен);
- вычисляют 16-октетное хеш-значение идентификатора пространства имен, сцепленного с именем, используя хеш-функцию, определенную в 14.2 или 14.3. Нумерация октетов в хеш-значении ведется от 0 до 15, как определено в IETF RFC 1321 (для ОС5) или в FIPS PUB 180-2 для ХАБ-1;
- устанавливают октеты с 3-го по 0-й поля «TimeLow» равными октетам с 3-го по 0-й хеш-значения;
- устанавливают октеты 1 и 0 поля «TimeMid» равными октетам 5 и 4 хеш-значения;
- устанавливают октеты 1 и 0 поля «VersionAndTimeHigh» равными октетам 7 и 6 хеш-значения;
- устанавливают четыре самых старших бита (биты с 15-го по 12-й) поля «VersionAndTimeHigh» равными четырехбитному номеру версии в соответствии с 12.2, таблица 3 для использованной хеш-функции;
- устанавливают поле «VariantAndClockSeqHigh» равным октету 8 хеш-значения;
- устанавливают два самых старших бита (биты 7 и 6) поля «VariantAndClockSeqHigh» равными 1 и 0 соответственно;
- устанавливают поле «ClockSeqLow» равным октету 9 хеш-значения;
- устанавливают октеты с 5-го по 0-й поля «Node» равными октетам с 15-го по 10-й хеш-значения.

14.2 В настоящем подразделе определен основанный на имени УУИд, использующий ОС5 в качестве хеш-функции, однако ОС5 не должен использоваться для вновь создаваемых УУИд (см. раздел С.4 приложения С). Для хеш-функции ОС5 указанное в 14.1 хеш-значение является 16-октетным значением, определенным в IETF RFC 1321, с октетами от 0 до 15.

Примечание — Определение УУИд на основе ОС5 с соответствующим номером версии включено в настоящий стандарт только в целях обратной совместимости с более ранними спецификациями.

14.3 В настоящем подразделе определен основанный на имени УУИд, использующий ХАБ-1 в качестве хеш-функции. Для хеш-функции ХАБ-1 хеш-значение, указанное в 14.1, должно быть октетами с 0-го по 15-й 20-октетного значения, полученного из 160-битового значения цифрового сообщения, определенного в FIPS PUB 180-2. Октеты с 16-го по 19-й этого 20-октетного значения должны быть отброшены. 20-октетное значение должно быть получено из 160-битового значения цифрового сообщения FIPS PUB 180-2 путем помещения самого старшего бита 160-битового значения в самый старший бит первого октета (октета 0) 20-октетного значения, а самого младшего бита — в последний октет (октет 19) 20-октетного значения.

15 Установка полей основанного на случайном числе УУИд

15.1 Поля основанного на случайном числе УУИд должны быть установлены следующим образом:
- устанавливают два самых старших бита (биты 7 и 6) поля «VariantAndClockSeqHigh» равными 1 и 0 соответственно;
- устанавливают четыре самых старших бита (биты с 15-го по 12-й) поля «VersionAndTimeHigh» равными четырехбитному номеру версии, определенному в 12.2;
- устанавливают остальные биты УУИд равными сгенерированному случайному (или псевдослучайному) значению.

Примечание — Псевдослучайные числа могут принимать одно и то же значение несколько раз. Настоятельно рекомендуется использовать случайные числа криптографического качества с целью уменьшить вероятность повторяющихся значений.

15.2 Руководство по генерации в системах случайных чисел приведено в приложении С.

16 Регистрация УУИд и их использование в качестве компонентов ИДО

16.1 Дерево ИДО АСН.1

Примечание — В настоящем подразделе приведена сводка основных положений Рекомендации МСЭ-Т X.660 | ИСО/МЭК 9834-1.

16.1.1 В настоящем стандарте определены процедуры работы уполномоченных по регистрации УУИд. Эта регистрация позволяет также использовать УУИд в качестве дуг дерева ИДО ниже дуги {joint-iso-itu-t uuid(25)} (см. также раздел 7).

Примечание — УУИд могут быть использованы для идентификации дуг ниже указанной дуги uuid(25) и без регистрации, но при этом нет гарантии, что идентификация таких дуг будет глобально недвусмысленной.

16.1.2 ИДО являются формой всемирно недвусмысленной идентификации, основанной на иерархической древовидной структуре и независимых иерархических уполномоченных по регистрации. Дерево ИДО имеет корень, дуги ниже этого корня, дуги каждой из этих дуг и т.д. на произвольную глубину. Дуги идентифицируются неотрицательными целыми значениями (см. 16.1.5), что обеспечивает недвусмысленную идентификацию дуг в пределах старшего узла. Дугам также могут быть присвоены имена (состоящие из одной или нескольких строчных или прописных букв, цифр и дефисов, начинающиеся со строчной буквы, не содержащие двух дефисов подряд и не заканчивающиеся дефисом), но они являются дополнительными к цифровым значениям и необязательными. Объект идентифицируется последовательностью значений для дуг (числовых или для начальных дуг также и именами) от корневого узла до объекта.

Примечание — Полное описание дерева ИДО см. в Рекомендации МСЭ-Т X.660 | ИСО/МЭК 9834-1 и Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1.

16.1.3 (Руководство) Важно отметить, что незарегистрированные УУИд могут быть использованы ниже той же дуги, что и зарегистрированные УУИд (см. 16.1.1). Таким образом, могут быть использованы идентичные значения для зарегистрированного и незарегистрированного УУИд (или для двух незарегистрированных УУИд), хотя вероятность этого очень мала. Вероятность возрастает, если УУИд генерируются из хеш-значений ОС5 или псевдослучайных чисел, а не из хеш-значений ХАБ-1 или случайных чисел криптографического качества. Это может вызвать путаницу для пользователей ИДО и спровоцировать злонамеренное использование подобных совпадений. Уполномоченный по регистрации УУИд отвечает за коллизии между зарегистрированными УУИд, но не за коллизии между зарегистрированными и незарегистрированными УУИд, т. к. он не управляет незарегистрированными УУИд. Если такая коллизия имеет место, то предпочтение отдается семантике, связанной с зарегистрированным УУИд; семантика, связанная с незарегистрированным УУИд, не должна использоваться. Таким образом, регистрация ИДО с УУИд не гарантирует того, что ИДО является более однозначным, чем его УУИд. Регистрация должна рассматриваться, главным образом, как способ опубликования УУИд и его семантики.

16.1.4 В машинном представлении ИДО можно подразумевать (в контексте данного машинного представления) идентификацию части пути от корневого узла до объекта в дереве ИДО. Таким образом, если известно, что ИДО сформирован так, как определено в разделе 7, то машинное представление может состоять из одного только значения УУИд.

16.1.5 Компоненты ИДО являются неограниченными неотрицательными целыми.

16.2 Назначение уполномоченных по регистрации

16.2.1 Организация регистрации в соответствии с требованиями настоящего стандарта входит в полномочия МСЭ-Т | ИСО/МЭК. Для этого МСЭ-Т | ИСО/МЭК назначает в соответствии со своими внутренними требованиями и правилами организацию, которая должна действовать как УР в соответствии с требованиями настоящего стандарта.

Примечание — Для регистрации используется ЕРП http://www.itu.int/ITU-T/asn1/uuid.html (сайт МСЭ-Т проекта АСН.1).

16.2.2 Уполномоченный по регистрации не несет ответственности за какие-либо недостатки в работе по настоящим процедурам или действия, относящиеся к обязанностям, определенным в настоящем стандарте, за исключением того, что УР может быть освобожден от выполнения этих обязанностей соответствующими Исследовательской группой МСЭ-Т | Подкомитетом СТК 1 ИСО/МЭК без штрафных санкций. Уполномоченный по регистрации не несет ответственности за какое-либо использование незарегистрированных значений ИДО, являющихся теми же, что и зарегистрированные значения ИДО, т. к. он не контролирует использование таких значений (см. 16.1.3).

Примечание — Если соответствующие Исследовательская группа МСЭ-Т | Подкомитет СТК 1 ИСО/МЭК решат, что уполномоченный по регистрации отстраняется от своих обязанностей по тем или иным причинам, то ожидается, что регистрационная информация, поддерживаемая УР, будет доступна вновь назначенному УР.

16.3 Оплата

16.3.1 Организация, являющаяся УР, должна осуществлять эту деятельность на основе возмещения издержек. Структура оплаты должна быть такой, чтобы покрывать расходы на работу УР, публикацию регистрационной информации в сети, обеспечение уникальности запросов и исключение необоснованных и многократных запросов.

16.3.2 Размер оплаты должен определяться УР и подлежит одобрению соответствующими Исследовательской группой МСЭ-Т | Подкомитетом СТК 1 ИСО/МЭК. Оплата может взыматься при:

a) регистрации;

b) справочном запросе;

c) публикации в сети;

d) запросе обновления.

16.3.3 Оплата не должна зависеть от страны создания заявления, не считая флуктуаций обменного курса.

16.4 Процедуры регистрации

В настоящем подразделе определены процедуры, которым нужно следовать при регистрации УУИд. Процедуры предназначены для обеспечения открытости и обязательности процесса работы УР.

16.4.1 Заявление на регистрацию

16.4.1.1 Организация направляет заявление на регистрацию УУИд непосредственно УР, заполняя форму на его веб-сайте. Содержание заявления определено в 16.4.3.

16.4.1.2 При успешном завершении процедур регистрации 128-битовое значение УУИд считается зарегистрированным как присвоенное представившей заявление организации и должно быть опубликовано.

16.4.2 Процесс подтверждения

Успешная регистрация подтверждается ответом на веб-сайте и публикацией в сети.

16.4.3 Содержание заявления

16.4.3.1 В настоящем подразделе определена информация, необходимая УР для проведения процесса регистрации.

Примечание — На момент публикации настоящего стандарта эта информация может быть предоставлена по электронной почте, телефону, в виде твердой копии или через регистрацию на веб-сайте.

16.4.3.2 Для регистрации необходима следующая информация:

a) страна, в которой находится головной офис регистрирующей организации;

b) название организации с регистрационной информацией страны, если регистрируется компания, благотворительная организация и т. д., или указанием отношения к известной международной организации;

c) имя и должность, почтовый адрес, адрес электронной почты, номер телефона и факса для контактов в регистрирующей организации;

d) информация в свободной форме о добропорядочности регистрирующей организации как средство для аудита и исключения подложных регистраций;

e) (факультативно) ЕРП, через который можно получить доступ к дополнительной информации об использовании УУИд.

16.4.3.3 Содержание заявления на ИДО определено в Рекомендации МСЭ-Т X.660 | ИСО/МЭК 9834-1, раздел 8.

16.5 Сопровождение регистра в сети

16.5.1 УР должен поддерживать веб-сайт со всеми регистрациями.

16.5.2 Информация об организациях, участвующих в регистрации, должна обновляться УР, если УР подробно приводит название компании или аналогичную информацию с соответствующей авторизацией изменений. Методы реализации должны быть определены УР и объявлены на сайте.

Приложение А
(справочное)

Алгоритмы эффективного создания основанных на времени УУИд

В настоящем приложении описан алгоритм, который может быть использован для быстрого создания основанных на времени УУИд в компьютерной системе.

А.1 Основной алгоритм

А.1.1 Следующий алгоритм является простым, корректным, но неэффективным:

- получают глобальную общесистемную блокировку,
- из общесистемной совместно используемой стабильной памяти (например, из файла) считывают состояние генератора УУИд: значения времени, временно́й последовательности и узла, использованные при создании последнего УУИд;
- получают текущее время в виде 60-битового отсчета 100-наносекундных интервалов от 00:00:00.00, 15 октября 1582 г. в качестве значения времени;
- получают текущее значение узла;
- если состояние было недействительным (например, несуществующим или поврежденным) или сохраненное значение узла отличается от текущего, то генерируют случайное значение временно́й последовательности;
- если состояние было действительным, но сохраненное значение времени больше текущего, то значение временно́й последовательности увеличивают;
- записывают состояние (текущие значения времени, временно́й последовательности и узла) обратно в стабильную память;
- освобождают глобальную блокировку;
- формируют УУИд из текущих значений времени, временно́й последовательности и узла в соответствии с алгоритмом, описанным в разделе 13.

А.1.2 Если нет необходимости часто генерировать УУИд, то приведенный выше алгоритм может оказаться вполне адекватным. Однако при высоких требованиях к производительности проявляются следующие проблемы основного алгоритма:

- чтение состояния из стабильной памяти каждый раз является неэффективным;
- разрешение системных часов может не составлять 100 наносекунд;
- запись состояния в стабильную память каждый раз является неэффективной;
- совместное использование состояния через границы процессов может быть неэффективным.

А.1.3 Каждая из этих проблем может быть решена локальными улучшениями модульным способом в тех функциях, которые читают и записывают состояние и читают время. Эти проблемы также рассмотрены в последующих разделах настоящего приложения.

А.2 Чтение стабильной памяти

А.2.1 Состояние необходимо читать из стабильной памяти только один раз во время загрузки при условии, что оно будет прочитано в переменную, общесистемную, совместно используемую память (и будет обновляться при каждом обновлении стабильной памяти).

А.2.2 Если реализация не имеет какой-либо доступной стабильной памяти, то она всегда может считать, что значения являются недействительными. Такая реализация является менее желательной, т. к. повышает частоту создания новых временны́х последовательностей, что повышает вероятность появления дубликатов.

А.2.3 Если значение узла не может никогда измениться (например, сетевая карта неотделима от системы) или любые изменения приводят к новой установке временно́й последовательности к новому случайному значению, то вместо хранения этого значения в стабильной памяти допускается возвращать текущее значение узла.

А.3 Разрешение системных часов

А.3.1 Значение времени генерируется из системного времени, разрешение которого может быть меньше, чем требуемое для времени разрешение.

А.3.2 Если нет необходимости в частой генерации УУИд, то время может быть просто системным временем, умноженным на число 100-наносекундных интервалов в интервале системного времени.

А.3.3 Если система перегружает генератор, запрашивая слишком много УУИд за один интервал системного времени, то услуга УУИд должна либо возвращать ошибку, либо приостановить генератор УУИд до захвата следующего системного времени.

А.3.4 Высокое разрешение значения времени может быть имитировано сохранением подсчета числа сгенерированных УУИд с одним и тем же значением системного времени и использованием его для построения низших битов значения времени. Подсчет должен вестись в диапазоне от нуля до числа 100-наносекундных интервалов в интервале системного времени.

Примечание — Если процессоры часто переполняют генератор УУИд, то системе могут быть выделены дополнительные адреса КДС, что позволит повысить скорость выделения УУИд, сделав для каждого значения времени потенциально доступными несколько УУИд.

А.4 Запись в стабильную память

Состояние не обязательно записывать в стабильную память при каждой генерации УУИд. Значение времени в стабильной памяти может периодически устанавливаться равным значению большему, чем любое использованное ранее в УУИд. До тех пор, пока сгенерированные УУИд имеют значение времени меньшее, чем это значение, а значения временно́й последовательности и узла остаются неизменными, необходимо обновлять только совместно используемую изменчивую копию состояния. Более того, если значение времени в стабильной памяти уходит вперед меньше, чем на типичное время перезагрузки системы, то авария системы не приведет к переустановке временно́й последовательности.

А.5 Совместное использование состояния процессами

Если слишком дорого получать доступ к совместно используемому состоянию при каждой генерации УУИд, то общесистемный генератор может быть реализован так, чтобы при каждом вызове выделял блок значений времени, а генераторы процессов могли бы выделять значения из этого блока до его исчерпания.

Приложение В
(справочное)

Свойства основанного на имени УУИд

В.1 Основанный на имени УУИд предназначен для генерации УУИд из имени, полученного из некоторого пространства имен и уникального в этом пространстве. Понятия имени и пространства имен должны толковаться широко и не ограничиваться текстовыми именами. Методы или соглашения для выделения имен и обеспечения их уникальности в пространстве имен не относятся к области применения настоящего стандарта.

Примечание — Во избежание проблем с рекурсией основанные на имени УУИд не должны генерироваться из ИДО, заканчивающихся основанным на имени УУИд.

В.2 Основанные на имени УУИд, созданные в соответствии с разделом 14 при соответствующем выборе пространства имен, будут иметь следующие свойства:

- УУИд, созданные в разное время из одного и того же имени в одном и том же пространстве имен, должны быть равны;
- УУИд, созданные из двух разных имен в одном и том же пространстве имен, должны быть различными с весьма высокой вероятностью;
- УУИд, созданные из одного и того же имени в двух разных пространствах имен, должны быть различными с весьма высокой вероятностью;
- если два основанных на имени УУИд равны, то с весьма высокой вероятностью они были сгенерированы из одного и того же имени в одном и том же пространстве имен.

Приложение С
(справочное)

Генерация случайных чисел в системе

С.1 Если система не имеет возможности генерировать случайные числа криптографического качества, то в большинстве систем обычно имеется достаточно большое число источников рандомизации, на основании которых можно генерировать такие числа. Эти источники являются специфическими для системы, но обычно включают в себя:

- процент используемой памяти;
- размер основной памяти в байтах;
- количество свободной основной памяти в байтах;
- размер файла страничного обмена или свопинга в байтах,
- свободные байты файла страничного обмена или свопинга;
- общий размер пользовательского виртуального адресного пространства в байтах;
- общее число доступных байтов пользовательского адресного пространства;
- размер загрузочного диска в байтах;
- свободное дисковое пространство на загрузочном диске в байтах;
- текущее время;
- количество времени с перезагрузки системы;
- индивидуальные размеры файлов в различных справочниках системы;
- время создания, последнего чтения и модификации файлов в различных справочниках системы;
- факторы инициализации различных ресурсов системы;
- текущее положение курсора мыши;
- текущее положение каретки;
- текущее число выполняемых процессов, связок;
- обработчика или ID окна стола и активного окна;
- значение указателя стека вызова;
- идентификатор процесса и связки вызова;
- различные специфические для архитектуры процессора счетчики выполнения (выполняемые инструкции, потери кеша, буфер TLB или его потери).

С.2 Кроме того, такие элементы, как имя компьютера и операционной системы, хотя и не являются очень случайными, но помогают дифференцировать результаты от полученных в других системах.

С.3 Точный механизм генерации значения узла является специфичным для системы, т. к. доступные данные и функции для их получения часто бывают очень специфичными для системы. Однако общий подход состоит в сборе данных из как можно большего числа источников в буфер, использовании обработки сообщений такой, например, как ХАБ-1, выборе произвольных шести октетов из хеш-значения и установке биты многоканальности так, как описано выше.

С.4 Могут быть использованы и другие хеш-функции, такие как ОС5 и хеш-функция, определенная в ИСО/МЭК 10118. Единственное требование состоит в том, чтобы результат был пригодно случайным (в том смысле, что выходы от набора однородно распределенных входов сами были однородно распределенными) и можно ожидать, что изменение одного бита на входе вызовет изменение половины битов на выходе (однако использовать ОС5 для новых УУИд не рекомендуется, т. к. последние исследования показали, что выходные значения не являются однородно распределенными).

Приложение D
(справочное)

Пример реализации

D.1 Описание файлов

Данная реализация состоит из шести файлов: copyrt.h, uuid.h, uuid.c, sysdep.h, sysdep.c и utest.c. Файлы uuid.* являются системно независимой реализацией алгоритмов генерации УУИд, описанных в разделах 13, 14 и 15, со всеми оптимизациями, описанными в приложении А (за исключением эффективного состояния, совместно используемого процессами). Код подразумевает поддержку 64-битового целого, что делает его более ясным.

Примечание — Код был протестирован в системах Linux (Red Hat 4.0) с GCC (2.7.2) и Windows NT 4.0 с VC + + 5.0.

D.2 Файл copyrt.h

Все последующие исходные файлы должны рассматриваться как имеющие следующее заявление об авторских правах:

```
/*
** Copyright (c) 1990 — 1993, 1996 Open Software Foundation, Inc.
** Copyright (c) 1989 by Hewlett-Packard Company, Palo Alto, Ca. &
** Digital Equipment Corporation, Maynard, Mass.
** Copyright (c) 1998 Microsoft.
** To anyone who acknowledges that this file is provided "AS IS"
** without any express or implied warranty: permission to use, copy,
** modify, and distribute this file for any purpose is hereby
** granted without fee, provided that the above copyright notices and
** this notice appears in all source code copies, and that none of
** the names of Open Software Foundation, Inc., Hewlett-Packard
** Company, or Digital Equipment Corporation be used in advertising
** or publicity pertaining to distribution of the software without
** specific, written prior permission. Neither Open Software
** Foundation, Inc., Hewlett-Packard Company, Microsoft, nor Digital
** Equipment Corporation makes any representations about the
** suitability
** of this software for any purpose.
*/
```

D.3 Файл uuid.h

```
#include "copyrt.h"
#undef uuid_t
typedef struct {
        unsigned32 time_low;
        unsigned16 time_mid;
        unsigned16 time_hi_and_version;
        unsigned8 clock_seq_hi_and_reserved;
        unsigned8 clock_seq_low;
        byte node[6];
} uuid_t;

/* uuid_create -- сгенерировать УУИд */
int uuid_create(uuid_t * uuid);

/* uuid_create_from_name -- создать УУИд, используя "имя"
      из "пространства имен" */
void uuid_create_from_name(
      uuid_t *uuid,     /* результирующий УУИд */
      uuid_t nsid,      /* УУИд пространства имен */
      void *name,       /* имя, из которого генерируется УУИд */
      int namelen       /* длина имени */
);
```

```
        /* uuid_compare -- Сравнить два УУИд "лексически" и вернуть
                -1      u1 лексически до u2
                0       u1 лексически равен u2
                1       u1 лексически после u2
        Лексическое упорядочение не является временным!
*/
int uuid_compare(uuid_t *u1, uuid_t *u2);
```

D.4 Файл uuid.c

```
#include "copyrt.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "sysdep.h"
#ifndef _WINDOWS_
#include <arpa/inet.h>
#endif
#include «uuid.h"

/* Различные начальные декларации */
static int read_state(unsigned16 *clockseq, uuid_time_t *timestamp,
        uuid_node_t *node);
static void write_state(unsigned16 clockseq, uuid_time_t timestamp,
        uuid_node_t node);
static void format_uuid_v1(uuid_t *uuid, unsigned16 clockseq,
        uuid_time_t timestamp, uuid_node_t node);
static void format_uuid_v3(uuid_t *uuid, unsigned char hash[16]);
static void get_current_time(uuid_time_t *timestamp);
static unsigned16 true_random(void);

/* uuid_create -- генератор УУИд */
int uuid_create(uuid_t *uuid)
{
        uuid_time_t timestamp, last_time;
        unsigned16 clockseq;
        uuid_node_t node;
        uuid_node_t last_node;
        int f;

        /* запросить общесистемную блокировку */
        LOCK;

        /* получить время, идентификатор узла, сохраненное состояние
                из неизменяемой памяти */
        get_current_time(&timestamp);
        get_ieee_node_identifier(&node);
        f = read_state(&clockseq, &last_time, &last_node);

        /* если состояние невалидно, или часы отошли назад,
                или изменился идентификатор узла (например,
                новая сетевая карта), изменить clockseq */
        if (!f || memcmp(&node, &last_node, sizeof node))
                clockseq = true_random();
        else if (timestamp < last_time)
                clockseq++;

        /* сохранить состояние для следующего раза */
        write_state(clockseq, timestamp, node);

        UNLOCK;

        /* установить поля в УУИд */
```

```
        format_uuid_v1(uuid, clockseq, timestamp, node);
        return 1;
}
/* format_uuid_v1 -- создать УУИд из timestamp, clockseq
        и node */
void format_uuid_v1(uuid_t* uuid, unsigned16 clock_seq,
        uuid_time_t timestamp, uuid_node_t node)
{
        /* Построить УУИд версии 1 с уже собранной информацией
             плюс некоторые постоянные. */
        uuid->time_low = (unsigned long)(timestamp & 0xFFFFFFFF);
        uuid->time_mid = (unsigned short)((timestamp >> 32) & 0xFFFF);
        uuid->time_hi_and_version =
        (unsigned short)((timestamp >> 48) & 0x0FFF);
        uuid->time_hi_and_version |= (1 << 12);
        uuid->clock_seq_low = clock_seq & 0xFF;
        uuid->clock_seq_hi_and_reserved = (clock_seq & 0x3F00) >> 8;
        uuid->clock_seq_hi_and_reserved |= 0x80;
        memcpy(&uuid->node, &node, sizeof uuid->node);
}

/* тип данных для устойчивого состояния генератора УУИд */
typedef struct {
        uuid_time_t ts;         /* сохраненный timestamp */
        uuid_node_t node;   /* сохраненный node */
        unsigned16 cs;          /* сохраненная временна́я
                                        последовательность */
} uuid_state;

static uuid_state st;

/* read_state -- прочитать состояние генератора УУИд из
                неизменяющейся памяти */
int read_state(unsigned16 *clockseq, uuid_time_t *timestamp,
                uuid_node_t *node)
{
        static int inited = 0;
        FILE *fp;

        /* состояние нужно читать только один раз за загрузку */
        if (!inited) {
                fp = fopen("state", "rb");
                if (fp = NULL)
                     return 0;
                fread(&st, sizeof st, 1, fp);
                fclose(fp);
                inited = 1;
        }
        *clockseq = st.cs;
        *timestamp = st.ts;
        *node = st.node;
        return 1;
}
/*write_state -- сохранить состояние генератора УУИд обратно в
        неизменяющуюся память */
void write_state(unsigned16 clockseq, uuid_time_t timestamp,
                        uuid_node_t node)
{
        static int inited = 0;
        static uuid_time_t next_save;
        FILE* fp;
```

```
        if (!inited) {
                next_save = timestamp;
                inited = 1;
        }

        /* всегда сохранять состояние в изменяющейся
                        совместно используемой памяти */
        st.cs = clockseq;
        st.ts = timestamp;
        st.node = node;
        if (timestamp >= next_save) {
                fp = fopen("state", "wb");
                fwrite(&st, sizeof st, 1, fp);
                fclose(fp);
                /*запланировать следующее сохранение через 10 секунд */
                next_save = timestamp + (10 * 10 * 1000 * 1000);
        }
}

/*get-current_time -- получить время как 60-битовые 100-наносекундные
        отсчеты с эпохи УУИд.
        Внести поправку на то, что разрешение реальных часов
        меньше чем 100 наносекунд. */
void get_current_time(uuid_time_t *timestamp)
{
        static int inited = 0;
        static uuid_time_t time_last;
        static unsigned16 uuids_this_tick;
        uuid_time_t time_now;

        if (!inited) {
                get_system_time(&time_now);
                uuids_this_tick = UUIDS_PER_TICK;
                inited = 1;
        }

        for ( ; ; ) {
                get_system_time(&time_now);

                /*если  чтение с часов изменилось с последнего
                        сгенерированного УУИд, */
                if (time_last != time_now) {
                        /*установить отсчет сгенерированных УУИд с данным чтением с часов */
                        uuids_this_tick = 0;
                        time_last = time_now;
                        break;
                }
                if (uuids_this_tick < UUIDS_PER_TICK) {
                        uuids_this_tick++;
                        break;
                }
                /*идет слишком быстро для наших часов; завернуть */
        }
        /* добавить отсчет УУИд к низшим битам, считанного с часов */
        *timestamp = time_now + uuids_this_tick;
}

/* true_random -- создать случайное число криптографического качества.
        ** В данном примере это не делается.** */
static unsigned16 true_random(void)
{
        static int inited = 0;
        uuid_time_t time_now;
```

```
        if (!inited) {
                get_system_time(&time_now);
                time_now = time_now / UUIDS_PER_TICK;
                srand((unsigned int)(((time_now >> 32) ^ time_now) & 0xffffffff));
                inited = 1;
        }
        return rand();
}

/*uuid_create_from_name -- создать УУИд, используя «имя» из
        "пространства имен" */
void uuid_create_from_name(uuid_t *uuid, uuid_t nsid, void *name,
                int namelen)
{

        MD5_CTX c;
        unsigned char hash[16];
        uuid_t net_nsid;

                /*поместить идентификатор пространства имен в сетевом порядке байтов, чтобы его хеширование
                        не зависело от используемой машины */
        net_nsid = nsid;
        htonl(net_nsid.time_low);
        htons(net_nsid.time_mid);
        htons(net_nsid.time_hi_and_version);

        MD5Init(&c);
        MD5Update(&c, &net_nsid, sizeof net_nsid);
        MD5Update(&c, name, namelen);
        MD5Final(hash, &c);

        /*здесь хеш в сетевом порядке байт */
        format_uuid_v3(uuid, hash);
}

/*format_uuid_v3 -- создать УУИд из (псевдо)случайного 128-битового числа */
void format_uuid_v3(uuid_t *uuid, unsigned char hash[16])
{
        /*преобразовать УУИд к локальному порядку байтов */
        memcpy(uuid, hash, sizeof *uuid);
        ntohl(uuid->time_low);
        ntohs(uuid->time_mid);
        ntohs(uuid->time_hi_and_version);

        /* записать uuid варианта и версии */
        uuid->time_hi_and_version &= 0x0FFF;
        uuid->time_hi_and_version |= (3 << 12);
        uuid->clock_seq_hi_and_reserved &= 0x3F;
        uuid->clock_seq_hi_and_reserved |= 0x80;
}

/* uuid_compare -- сравнить два УУИд «лексически» */
#define CHECK(f1, f2) if (f1 != f2) return f1 < f2 ? -1 : 1;
int uuid_compare(uuid_t *u1, uuid_t *u2)
{

        int i;
        CHECK(u1->time_low, u2->time_low);
        CHECK(u1->time_mid, u2->time_mid);
        CHECK(u1->time_hi_and_version, u2->time_hi_and_version);
        CHECK(u1->clock_seq_hi_and_reserved,
                        u2->clock_seq_hi_and_reserved);
```

```
        CHECK(u1->clock_seq_low, u2->clock_seq_low)
        for (i = 0; i < 6; i++) {
                if (u1->node[i] < u2->node[i])
                        return -1;
                if (u1->node[i] > u2->node[i])
                        return 1;
        }
        return 0;
}
#undef CHECK
```

D.5 Файл sysdep.h

```
#include «copyrt.h"
/* удалить следующее определение, если не используется Windows 32 */
#define WININC 0

#ifdef WININC
#include <windows.h>
#else
#include <time.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>
#endif

#include "global.h"
/* в тех случаях, где используется OC5; в IETF RFC 1321 есть
        пример реализации */
#include «md5.h"

/* установить число 100 наносекундных отсчетов фактического разрешения системных часов */
#define UUIDS_PER_TICK 1024

/* установить для получения и освобождения глобальной блокировки */
#define LOCK
#define UNLOCK

typedef unsigned long unsigned32;
typedef unsigned short unsigned16;
typedef unsigned char unsigned8;
typedef unsigned char byte;

/* установить, чтобы компилятор использовал 64-битовый тип данных */
#ifdef WININC
#define unsigned64_t unsigned __int64
#define I64(C) C
#else
#define unsigned64_t unsigned long long
#define I64(C) C##LL
#endif

typedef unsigned64_t uuid_time_t;
typedef struct {
        char nodeID[6];
} uuid_node_t;

void get_ieee_node_identifier(uuid_node_t *node);
void get_system_time(uuid_time_t *uuid_time);
void get_random_info(unsigned char seed[16]);
```

D.6 Файл sysdep.c

```
#include "copyrt.h"
#include <stdio.h>
```

```
#include <string.h>
#include "sysdep.h"

/*системно-зависимый вызов для получения идентификатора узла КДС.
        В данном примере генерируется случайный идентификатор узла. */
void get_ieee_node_identifier(uuid_node_t *node)
{
        static int inited = 0;
        static uuid_node_t saved_node;
        unsigned char seed[16];
        FILE *fp;

        if (!inited) {
                fp = fopen("nodeid", "rb");
                if (fp) {
                        fread(&saved_node, sizeof saved_node, 1, fp);
                        fclose(fp);
                }
                else {
                        get_random_info(seed);
                        seed[0] |= 0x80;
                        memcpy(&saved_node, seed, sizeof saved_node);
                        fp = fopen("nodeid", "wb");
                        if (fp) {
                                fwrite(&saved_node, sizeof saved_node, 1, fp);
                                fclose(fp);
                                }
                        }
                        inited = 1;
        }

        *node = saved_node;
}

/*системно-зависимый вызов для получения текущего системного времени. Возвращает 100-наносекундные
        отсчеты от эпохи УУИд, но разрешение может быть меньше 100 наносекунд.*/
#ifdef _WINDOWS_

void get_system_time(uuid_time_t *uuid_time)
{
        ULARGE_INTEGER time;

        /* В Windows NT время хранится в формате FILETIME, который есть число 100 наносекундных отсчетов
                от 1 января 1601 г. В УУИд используется время в 100 наносекундных отсчетах от 15 октября 1582 г.
                Разница составляет 17 дней в октябре + 30 (ноябрь) + 31 (декабрь) + 18 лет + 5 дней в високосных годах. */
        GetSystemTimeAsFileTime((FILETIME *)&time);
        time.QuadPart +=
                (unsigned __int64) (1000*1000*10) // секунды
                * (unsigned __int64) (60 * 60 * 24) // дни
                * (unsigned __int64) (17+30+31+365*18+5); // число дней
        *uuid_time = time.QuadPart;
}

void get_random_info(unsigned char seed[16])
{
        MD5_CTX c;
        struct {
                MEMORYSTATUS m;
                SYSTEM_INFO s;
                FILETIME t;
                LARGE_INTEGER pc;
                DWORD tc;
```

```
            DWORD l;
            char hostname[MAX_COMPUTERNAME_LENGTH + 1];
        } r;

        MD5Init(&c);
        GlobalMemoryStatus(&r.m);
        GetSystemInfo(&r.s);
        GetSystemTimeAsFileTime(&r.t);
        QueryPerformanceCounter(&r.pc);
        r.tc = GetTickCount();
        r.l = MAX_COMPUTERNAME_LENGTH + 1;
        GetComputerName(r.hostname, &r.l);
        MD5Update(&c, &r, sizeof r);
        MD5Final(seed, &c);
}

#else

void get_system_time(uuid_time_t *uuid_time)
{
        struct timeval tp;

        gettimeofday(&tp, (struct timezone *)0);
        /* Сдвиг между форматами времени УУИд и Unix.
                Для УУИд базовым UTC является 15 октября 1582 г.
                Для Unix базовым временем является 1 января 1970 г. */
        *uuid_time = (tp.tv_sec * 10000000) + (tp.tv_usec * 10) + I64(0x01B21DD213814000);
}

void get_random_info(unsigned char seed[16])
{
        MD5_CTX c;
        struct {
                struct timeval t;
                char hostname[257];
        } r;

        MD5Init(&c);
        gettimeofday(&r.t, (struct timezone *)0);
        gethostname(r.hostname, 256);
        MD5Update(&c, &r, sizeof r);
        MD5Final(seed, &c);
}

#endif
```

D.7 Файл utest.c

```
#include "copyrt.h"
#include "sysdep.h"
#include <stdio.h>
#include "uuid.h"

uuid_t NameSpace_DNS = { /* 6ba7b810-9dad-11d1-80b4-00c04fd430c8 */
        0x6ba7b810,
        0x9dad,
        0x11d1,
        0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/* puid -- печать УУИд */
void puid(uuid_t u)
{
```

```
        int i;
        printf("%8.8x-%4.4x-%4.4x-%2.2x%2.2x-", u.time_low, u.time_mid,
        u.time_hi_and_version, u.clock_seq_hi_and_reserved,
        u.clock_seq_low);
        for (i = 0; i < 6; i++)
        printf("%2.2x", u.node[i]);
        printf("\n");
}

/* простой драйвер для генератора УУИд */
int main(int argc, char **argv)
{
        uuid_t u;
        int f;

        uuid_create(&u);
        printf("uuid_create(): "); puid(u);

        f = uuid_compare(&u, &u);
        printf("uuid_compare(u,u): %d\n", f); /* должен быть 0*/
        f = uuid_compare(&u, &NameSpace_DNS);
        printf("uuid_compare(u, NameSpace_DNS): %d\n", f);
                                                /* должна быть 1 */
        f = uuid_compare(&NameSpace_DNS, &u);
        printf("uuid_compare(NameSpace_DNS, u): %d\n", f);
                                                /* должна быть -1 */
        uuid_create_from_name(&u, NameSpace_DNS, "www.widgets.com",
                                        15);
        printf("uuid_create_from_name(): "); puid(u);
}
```

D.8 Пример выхода utest

```
uuid_create(): 7d444840-9dc0-11d1-b245-5ffdce74fad2
uuid_compare(u,u): 0
uuid_compare(u, NameSpace_DNS): 1
uuid_compare(NameSpace_DNS, u): -1
uuid_create_from_name(): e902893a-9d22-3c7e-a7b8-d6e313b71d9f
```

D.9 Некоторые идентификаторы пространств имен

В настоящем разделе перечислены идентификаторы некоторых потенциально интересных пространств имен в виде структур языка С соответствующих строковых представлений.

```
/*Строка имени является полностью уточненным доменным именем*/
uuid_t NameSpace_DNS = {/*6ba7b810-9dad-11d1-80b4-00c04fd430c8*/
0x6ba7b810,
0x9dad,
0x11d1,
0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/*Строка имени является ЕПР*/
uuid_t NameSpace_URL = {/*6ba7b811-9dad-11d1-80b4-00c04fd430c8*/
0x6ba7b811,
0x9dad,
0x11d1,
0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/*Строка имени является ИДО */
uuid_t NameSpace_OID = {/*6ba7b812-9dad-11d1-80b4-00c04fd430c8*/
0x6ba7b812,
0x9dad,
0x11d1,
```

```
0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/*Строка имени является отличающим именем справочника
      (в формате DER или текстовом)*/
uuid_t NameSpace_X500 = {/*6ba7b814-9dad-11d1-80b4-00c04fd430c8*/
0x6ba7b814,
0x9dad,
0x11d1,
0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};
```

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации (и действующим в этом качестве межгосударственным стандартам)

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8802-3:2000	MOD	ГОСТ 34.913.3—91 (ИСО 8802-3—89) «Информационная технология. Локальные вычислительные сети. Метод случайного доступа к шине и спецификация физического уровня»
ИСО/МЭК 8824-1:2002	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО/МЭК 9834-1:2004	IDT	ГОСТ Р ИСО/МЭК 9834-1—2009 «Информационная технология. Взаимосвязь открытых систем. Процедуры действий уполномоченных по регистрации ВОС. Часть 1. Общие процедуры и верхние дуги дерева идентификатора объекта АСН.1»
ИСО/МЭК 10118-3:2004	—	*
ИСО/МЭК 10646:2003	—	*
FIPS PUB 180-2:2002	—	*
IETF RFC 1321 (1992)	—	*
IETF RFC 2141 (1997)	—	*

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использованы следующие условные обозначения степени соответствия стандартов:

- IDT — идентичные стандарты;
- MOD — модифицированные стандарты.

Библиография

[1] ZAHN (L.), DINEEN (T.), LEACH (P.): Network Computing Architecture, ISBN 0-13-611674-4, January 1990
[2] Open Group CAE: DCE. Remote Procedure Call, Specification C309, ISBN 1-85912-041-5, August 1994
[3] ISO/IEC 11578:1996 Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)
[4] IEEE, Request Form for an Individual Address Block (also known as an Ethernet Address Block) of 4,096 MAC Addresses, gost.e-ecolog.ru
[5] ITU-T Recommendation X.500 (2001)/ISO/IEC 9594-1:2001 Information technology — Open Systems Interconnection — The Directory: Overview of concepts, models and services
[6] IETF RFC 3061 (2001), A URN Namespace of Object Identifiers
[7] Internet-Draft draft-mealling-uuid-urn-00, A UUID URN Namespace, M. Mealling, P. Leach, R. Salz, October 2002

УДК 681.3:691.39:006.354 ОКС 35.100.01 П85

Ключевые слова: обработка данных, информационный обмен, сетевое взаимодействие, взаимосвязь открытых систем, регистрация

Редактор *В.Н. Копысов*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Д. Дульнева*
Компьютерная верстка *И.А. Налейкиной*

Сдано в набор 20.01.2012. Подписано в печать 14.02.2012. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 3,72. Уч.-изд. л. 3,20. Тираж 99 экз. Зак. 154.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru »@gostinfo.ru
Набрано во ФГУП «СТАНДАРТИНФОРМ» на ПЭВМ.
Отпечатано в филиале ФГУП «СТАНДАРТИНФОРМ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.

gost.e-ecolog.ru