

TCVN 10583-8:2014

ISO/IEC 9834-8:2008

Xuất bản lần 1

**CÔNG NGHỆ THÔNG TIN – LIÊN KẾT HỆ THỐNG MỞ –
THỦ TỤC ĐIỀU HÀNH CỦA CƠ QUAN ĐĂNG KÝ OSI –
PHẦN 8: TẠO VÀ ĐĂNG KÝ CÁC ĐỊNH DANH ĐƠN NHẤT
(UUIDs) VÀ SỬ DỤNG NHƯ CÁC THÀNH PHẦN ĐỊNH
DANH ĐỐI TƯỢNG ASN.1**

*Information technology – Open systems interconnection – Procedures for the operation
of OSI registration authorities –*

*Part 8: Generation and registration of universally unique identifier (UUIDs) and their use
as ASN.1 object identifier components*

HÀ NỘI - 2014

Mục lục	Trang
Lời nói đầu.....	4
Lời giới thiệu.....	5
1 Phạm vi áp dụng	7
2 Tài liệu viện dẫn.....	7
3 Thuật ngữ và định nghĩa	8
4 Các từ viết tắt	10
5 Ký pháp	11
6 Cấu trúc UUID và các phép biểu diễn	11
7 Sử dụng UUID như giá trị nguyên sơ cấp và nhãn Unicode của cung UUID.....	14
8 Sử dụng UUID để tạo URN.....	14
9 Các quy tắc so sánh và sắp xếp UUID	14
10 Tính hợp lệ	14
11 Bit biến thể.....	15
12 Sử dụng các trường UUID và thứ tự byte khi truyền.....	15
13 Thiết lập các trường của UUID theo thời gian	19
14 Thiết lập các trường UUID theo tên	19
15 Thiết lập các trường của UUID theo số ngẫu nhiên	20
16 Đăng ký UUID và cách sử dụng UUID trong cây định danh đối tượng quốc tế	21
Phụ lục A (Tham khảo) Thuật toán tạo ra UUID theo thời gian	24
Phụ lục B (Tham khảo) Đặc tính của UUID theo tên.....	26
Phụ lục C (Tham khảo) Phát sinh các số ngẫu nhiên trong một hệ thống.....	27
Phụ lục D (Tham khảo) Biểu diễn mẫu.....	29

Lời nói đầu

TCVN 10583-8:2014 hoàn toàn tương đương với ISO/IEC 9834-8:2008

TCVN 10583-8:2014 do Ban kỹ thuật tiêu chuẩn quốc gia TCVN/JTC 1 *Công nghệ thông tin* biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Lời giới thiệu

Bộ tiêu chuẩn TCVN 10583 (ISO/IEC 9834) *Công nghệ thông tin – Liên kết hệ thống mở* gồm các tiêu chuẩn sau đây:

1. TCVN 10583-1:2014 (ISO/IEC 9834-1:2012) *Công nghệ thông tin – Thủ tục điều hành của cơ quan đăng ký định danh đối tượng – Phần 1: Thủ tục chung và các cung trên cùng của cây định danh đối tượng quốc tế*
2. TCVN 10583-2:2014 (ISO/IEC 9834-2:1993) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 2: Thủ tục đăng ký cho kiểu tài liệu OSI*
3. TCVN 10583-3:2014 (ISO/IEC 9834-3:2008) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 3: Đăng ký các cung định danh đối tượng bên dưới cung mức trên cùng do ISO và ITU-T quản trị*
4. TCVN 10583-4:2014 (ISO/IEC 9834-4:1991) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 4: Sổ đăng ký của hồ sơ VTE*
5. TCVN 10583-5:2014 (ISO/IEC 9834-5:1991) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 5: Sổ đăng ký xác định đối tượng điều khiển VT*
6. TCVN 10583-6:2014 (ISO/IEC 9834-6:2005) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 6: Đăng ký quá trình ứng dụng và thực thể ứng dụng*
7. TCVN 10583-7:2014 (ISO/IEC 9834-7:2008) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 7: Việc đăng ký của các tổ chức quốc tế ISO và ITU-T*
8. TCVN 10583-8:2014 (ISO/IEC 9834-8:2008) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 8: Tạo và đăng ký các định danh đơn nhất (UUIDs) và sử dụng như các thành phần định danh đối tượng ASN.1*
9. TCVN 10583-9:2014 (ISO/IEC 9834-9:2008) *Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 9: Đăng ký các cung định danh đối tượng cho ứng dụng và dịch vụ sử dụng định danh thẻ*

Công nghệ thông tin – Liên kết hệ thống mở – Thủ tục điều hành của cơ quan đăng ký OSI – Phần 8: Tạo và đăng ký các định danh đơn nhất (UUIDs) và sử dụng như các thành phần định danh đối tượng ASN.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

1 Phạm vi áp dụng

Tiêu chuẩn này quy định định dạng và quy tắc tạo cho phép người sử dụng đưa ra các định danh 128 bit được đảm bảo đơn nhất toàn cầu hoặc đơn nhất toàn cầu với xác suất cao.

UUID được tạo ra theo Tiêu chuẩn này phù hợp với định danh tạm thời 100 nano giây tạo ra một UUID mới, hoặc phù hợp với định danh cố định.

Tiêu chuẩn này bắt nguồn từ các đặc tả ban đầu không theo tiêu chuẩn của UUID và thể hệ sau của UUID giống hệt với các đặc tả ban đầu đó.

Tiêu chuẩn này quy định các thủ tục điều hành của cơ quan đăng ký UUID trên nền tảng Web.

Tiêu chuẩn này quy định và cho phép sử dụng các UUID (được đăng ký hoặc không được đăng ký) như giá trị sơ cấp (xác định nhãn Unicode) cho các cung bên dưới cung liên kết UUID. Điều này cho phép người sử dụng tạo và sử dụng các cung mà không cần thực hiện các thủ tục đăng ký.

Tiêu chuẩn này quy định và cho phép sử dụng UUID (được đăng ký hoặc không được đăng ký) để tạo ra một URN.

2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau là rất cần thiết cho việc áp dụng tiêu chuẩn. Đối với các tài liệu viện dẫn ghi năm công bố thì áp dụng bản được nêu. Đối với các tài liệu viện dẫn không ghi năm công bố thì áp dụng phiên bản mới nhất, bao gồm cả các sửa đổi.

TCVN 8271:2010 (ISO/IEC 10646:2003) Công nghệ thông tin – Bộ ký tự mã hóa tiếng Việt.

TCVN 10583-8:2014

Recommendation ITU-T X.660 (2004)|ISO/IEC 9834-1:2008¹ *Information technology – Open systems interconnection – Procedures for the operation of OSI registration authorities: General procedures and top arcs of the international object identifier tree* (Công nghệ thông tin – Liên kết hệ thống mở - Thủ tục điều hành của cơ quan đăng ký OSI: Thủ tục chung và các cung trên cùng của cây định danh đối tượng quốc tế).

Recommendation ITU-T X.680 (2002) | ISO/IEC 8824-1:2002 *Information technology – Abstract syntax notation one (ASN.1) : Specification of basic notation* (Công nghệ thông tin – Ký pháp cú pháp trừu tượng phiên bản 1: Quy định về ký pháp cơ sở)

ISO/IEC 8802-3:2000, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area network – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications* (Công nghệ thông tin – Viễn thông và trao đổi thông tin giữa các hệ thống – Mạng cục bộ và mạng trung tâm – Các yêu cầu cụ thể - Phần 3: Đa truy cập bằng cảm sóng với phương pháp truy cập phát hiện xung đột và các quy định về tầng vật lý)

ISO/IEC 10118-3:2004, *Information technology – Security techniques – Hash functions – Part 3: Dedicated hash-functions* (Công nghệ thông tin – Kỹ thuật an toàn – Hàm băm – Phần 3: Các hàm băm dành riêng).

FIPS PUB 180-3:2008, *Federal information processing standards publication, secure hash standard (SHS)* (Công bố các tiêu chuẩn về xử lý thông tin, tiêu chuẩn hàm băm an toàn)

IETF RFC 1321(1992) , *The MD5 message – Digest algorithm* (Thông điệp MD5 – Thuật toán phân loại)

IETF RFC 2141 (1997), *URN syntax* (cú pháp URN).

3 Thuật ngữ và định nghĩa

Các định nghĩa sau đây áp dụng trong tiêu chuẩn này.

3.1 Ký pháp ASN.1

Tiêu chuẩn này sử dụng các thuật ngữ được nêu trong Recommendation ITU-T X.680 | ISO/IEC 8824-1:

- a. Giờ quốc tế (UTC);
- b. Kiểu định danh đối tượng;
- c. Kiểu định danh nguồn quốc tế hóa OID

3.2 Cơ quan đăng ký

Tiêu chuẩn này sử dụng các thuật ngữ được nêu trong TCVN 10583-1 (ISO/IEC 9834-1 | Recommendation ITU-T X.660):

¹ ISO/IEC 9834-1:2008 đã hủy và thay thế bằng ISO/IEC 9834-1:2012

- a. Cây định danh đối tượng quốc tế
- b. Giá trị IRI/URI
- c. Định danh nguồn quốc tế hóa OID
- d. Định danh đối tượng
- e. Đăng ký
- f. Cơ quan đăng ký
- g. Các thủ tục đăng ký
- h. Định danh thứ cấp
- i. Ký tự Unicode
- j. Nhãn Unicode

3.3 Thuật ngữ mạng

Tiêu chuẩn này sử dụng các thuật ngữ được nêu trong ISO/IEC 8802-3:

- Địa chỉ MAC

3.4 Định nghĩa bổ sung

3.4.1

Số ngẫu nhiên mã hóa (cryptographic-quality random-number)

Số ngẫu nhiên hoặc số giả ngẫu nhiên được tạo ra theo cơ chế đảm bảo một chuỗi đầy đủ của các giá trị lặp lại để sử dụng khi mã hóa (và sử dụng cho các công việc tương tự như vậy).

3.4.2

Cung liên kết UUID (joint UUID arc)

Là cung ở bên dưới nút cây định danh đối tượng quốc tế được định danh bởi giá trị định danh đối tượng {Joint-iso-1 tu-t (2) uuid (25)} và giá trị định danh nguồn quốc tế hóa ASSN.1 OID "/UUID".

3.4.3

Phiên bản theo tên gọi (name-based version)

UUID được tạo ra sử dụng hàm băm mật mã của vùng tên và tên trong vùng tên đó.

3.4.4

Vùng tên (name space)

Hệ thống tạo ra tên của đối tượng để đảm bảo định danh không tối nghĩa trong vùng tên đó.

CHÚ THÍCH: Các ví dụ cho vùng tên là hệ thống tên miền mạng, URNs, OIDs, tên xác định thư điều (Xem [5]), và các từ khóa dành riêng cho ngôn ngữ lập trình.

3.4.5

Phiên bản theo số ngẫu nhiên (random-number-based version)

Một UUID được tạo ra sử dụng một số ngẫu nhiên hoặc một số giả ngẫu nhiên.

3.4.6

Biến thể UUID tiêu chuẩn (standard UUID variant)

Biến thể của định dạng UUID khả thi được quy định trong tiêu chuẩn này.

CHÚ THÍCH: Xét trên phương diện lịch sử, định dạng UUID khác với định dạng biến thể của nó quy định trong tiêu chuẩn này. UUID được tạo ra theo định dạng biến thể này đều khác biệt.

3.4.7

Phiên bản theo thời gian (time-based version)

UUID duy nhất sử dụng địa chỉ MAC để xác định một hệ thống và một giá trị Clock theo thời gian UTC hiện tại.

3.4.8

Định danh đơn nhất phổ biến (UUID) (universally unique identifier (UUID))

Giá trị 128 bit được tạo ra phù hợp với Tiêu chuẩn này và phù hợp với một số đặc tính cũ và cung cấp giá trị duy nhất giữa hệ thống và giờ phụ trội. (xem 3.4.6)

4 Các từ viết tắt

Tiêu chuẩn này áp dụng các từ viết tắt sau:

ASSN.1:	Ký pháp cú pháp triu tượng
GUID:	Định danh đơn nhất toàn cầu
MAC:	Điều khiển truy cập thiết bị
MD5:	Thuật toán mã khóa thông điệp 5
OID:	Định danh đối tượng ASN.1
OID-IRI:	Định danh nguồn quốc tế hóa OID
RA:	Cơ quan đăng ký
SHA-1:	Thuật toán hàm băm bảo mật 1
URL:	Định vị nguồn đồng nhất
URN:	Tên nguồn đồng nhất
UTC:	Thời gian điều phối phổ dụng
UUID:	Định danh đơn nhất phổ dụng

5 Ký pháp

5.1 Tiêu chuẩn này quy định một dãy 8 bit để UUID sử dụng các thuật ngữ đầu tiên và cuối cùng. Dãy 8 bit đầu tiên gọi là “octet 15” và dãy 8 bit cuối cùng gọi là “octet 0”.

5.2 Các bit trong một UUID được đánh số từ “bit 127” đến “bit 0”, trong đó bit 127 là bit quan trọng nhất của octet 15 và bit 0 là bit ít quan trọng nhất của octet 0.

5.3 Khi sử dụng số và bảng trong tiêu chuẩn này octet quan trọng nhất (và cũng là bit quan trọng nhất) được hiện thị ở bên trái của trang. Điều này tương ứng với thứ tự truyền các octet, trong đó các octet phía bên trái được truyền đi đầu tiên.

5.4 Một số giá trị được sử dụng trong tiêu chuẩn này được thể hiện như giá trị của số nguyên không dấu với độ dài bit giả định (gọi là N). Các bit của giá trị nguyên không dấu N-bit được đánh số từ “bit N-1” đến “bit 0”, trong đó bit N-1 là bit quan trọng nhất và bit 0 là bit ít quan trọng nhất.

5.5 Các ký pháp được sử dụng duy nhất trong tiêu chuẩn này. Các phần biểu diễn trên bộ nhớ máy tính không được chuẩn hóa và phụ thuộc vào cấu trúc hệ thống.

6 Cấu trúc UUID và các phép biểu diễn

6.1 Cấu trúc trường UUID

6.1.1 UUID được quy định như một dãy thứ tự của 6 trường. UUID được quy định như thuật ngữ ghép các trường UUID. Các trường UUID được đặt tên là:

- a) Trường “TimeLow”
- b) Trường “TimeMid”
- c) Trường “VersionAndTimeHigh”
- d) Trường “VariantAndClockSeqHigh”
- e) Trường “ClockSeqLow”
- f) Trường “Node”

6.1.2 Các trường UUID có mức độ quan trọng theo thứ tự được liệt kê bên trên, trong đó “TimeLow” là trường quan trọng nhất (bit 31 của “TimeLow” là bit 127 của UUID, và “Node” là trường ít quan trọng nhất (bit 0 của “Node” là bit 0 của UUID).

6.1.3 Nội dung của các trường UUID được quy định trong các thuật ngữ của giá trị nguyên không dấu Version, Variant, Time, Clock Sequence và Node (mỗi trường có kích cỡ bit cố định). Việc thiết lập các giá trị này được quy định tại Điều 12 và gán giá trị cho các trường UUID được quy định tại 12.1.

CHÚ THÍCH – Một phần tên của các trường UUID (ví dụ, TimeLow, TimeMid, và TimeHigh) là thứ tự liên tiếp của các bit trong một UUID (từ bit 0 đến bit 127) được suy ra từ giá trị nguyên không dấu riêng biệt (ví dụ từ bit 0 đến bit 59 của TimeValue) không giống như thứ tự liên tiếp của các bit trong giá trị nguyên không dấu đó.

6.2 Biểu diễn nhị phân

6.2.1 Một UUID sẽ được biểu diễn theo dạng nhị phân 16 octet được hình thành bởi liên kết cố định của số nguyên không dấu mã hóa mỗi trường của nó vào trong 1 hoặc nhiều octet. Số octet sử dụng cho mỗi trường như sau:

- a) Trường “TimeLow”: 4 octet
- b) Trường “TimeMid”: 2 octet
- c) Trường “VersionAndTimeHigh”: 2 octet
- d) Trường “VariantAndClockSeqHigh”: 1 octet
- e) Trường “ClockSeqLow”: 1 octet
- f) Trường “Node”: 6 octet

CHÚ THÍCH: Thứ tự này của trường UUID được biểu diễn thường xuyên trong hệ thống máy tính và biểu diễn dưới dạng ký tự của hệ thập lục phân (Xem 6.4).

6.2.2 Bit quan trọng nhất của mã hóa số nguyên không dấu của các trường UUID sẽ là bit quan trọng nhất của octet đầu tiên (octet N, octet quan trọng nhất), và bit ít quan trọng nhất của mã hóa số nguyên không dấu là bit ít quan trọng nhất của octet cuối cùng (octet 0, bit ít quan trọng nhất).

6.2.3 Các trường UUID liên kết với nhau theo thứ tự mức độ quan trọng của chúng (xem điều 6.1.2) cùng với trường quan trọng nhất đầu tiên và trường ít quan trọng nhất cuối cùng.

6.3 Biểu diễn giá trị nguyên đơn

Một UUID được biểu diễn như giá trị nguyên đơn. Để lấy giá trị nguyên đơn của UUID thì 16 octet của biểu diễn nhị phân sẽ xem mã hóa số nguyên không dấu và bit quan trọng nhất của mã hóa số nguyên này như bit quan trọng nhất (bit 7) trong phần đầu tiên của 16 octet (octet 15), xem mã số nguyên không dấu và bit ít quan trọng nhất của mã số nguyên không dấu này như bit ít quan trọng nhất (bit 0) trong phần cuối cùng của octet 16 (octet 0).

CHÚ THÍCH: Giá trị nguyên đơn được sử dụng khi UUID hình thành giá trị nguyên ban đầu của cung liên kết UUID theo quy định tại Điều 7.

6.4 Biểu diễn thập lục phân

Đối với định dạng thập lục phân, các octet của định dạng nhị phân này được biểu diễn theo dạng chuỗi các con số, sử dụng hai số thập lục phân cho mỗi octet của định dạng nhị phân, định dạng đầu tiên là giá trị của 4 bit ô cao của octet 15, định dạng thứ 2 là giá trị của 4 bit ô thấp của octet 0 (xem 6.5). Ký tự “-” (45) (Xem TCVN 8271 (ISO/IEC 10646)) được chèn giữa biểu diễn thập lục phân của mỗi cặp trường liền kề nhau, loại trừ trường hợp bị chèn giữa trường “VariantAndClockSeqHigh” và trường “ClockSeqLow” (Xem ví dụ tại Điều 8)

6.5 Cú pháp chính tắc của biểu diễn thập lục phân

6.5.1 Định nghĩa chính tắc của cú pháp biểu diễn thập lục phân UUID quy định sử dụng ký pháp BNF mở rộng trong Recommendation ITU-TX.680/ISO/IEC 8824-1, Điều 5, ngoại trừ trường hợp có khoảng trống giữa các điều từ vựng.

6.5.2 Mục từ vựng “hexdigit” được sử dụng trong đặc tả BNF và được xác định như sau:

Tên của mục từ vựng – hexdigit

Một “hexdigit” bao gồm các ký tự sau

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 Biểu diễn thập nhị phân của UUID phải là việc tạo ra “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 Phần mềm tạo ra biểu diễn thập lục phân của UUID không sử dụng các chữ hoa.

CHÚ THÍCH: Biểu diễn thập lục phân được khuyến cáo sử dụng theo định dạng dễ đọc, hạn chế ký tự chữ thường. Tuy nhiên phần mềm xử lý biểu diễn này được yêu cầu chấp nhận cả ký tự chữ hoa và chữ thường theo quy định tại 6.5.2.

7 Sử dụng UUID như giá trị nguyên sơ cấp và nhãn Unicode của cung liên kết UUID

7.1 UUID (được đăng ký hoặc không được đăng ký) được xem như giá trị nguyên sơ cấp của cung liên kết UUID sử dụng giá trị nguyên đơn của UUID như quy định tại Điều 6.3.

CHÚ THÍCH: Điều 16 cung cấp các thông tin về cách sử dụng UUID như số nguyên đầu tiên của cung liên kết UUID.

7.2. Biểu diễn thập lục phân của UUID tại điều 6.4 được sử dụng như nhãn Unicode thập phân cho một cung.

VÍ DỤ: Sau đây là ví dụ về cách sử dụng UUID để tạo thành giá trị IRI/URI:

```
"oid:/UUID/f81d4ae-7dec-11d0-a756-00a0c91e6bf6"
```

8 Sử dụng UUID để tạo URN

Một URN (Xem IETF RFC 2141) được tạo ra bằng cách sử dụng một UUID là chuỗi "urn:uuid" theo biểu diễn thập nhị phân của UUID tại điều 6.4.

VÍ DỤ: Sau đây là ví dụ của biểu diễn chuỗi của một UUID như một URN

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

CHÚ THÍCH: Định dạng URN xen kẽ nhau (xem [6]) luôn sẵn có, nhưng không dùng cho URN do UUID tạo ra. Định dạng xen kẽ này sử dụng giá trị nguyên đơn của UUID quy định tại Điều 6.3. Ví dụ trên được biểu diễn như sau:
"urn:oid:2.25.329800735698586629295641978511506172918"

9 Các quy tắc so sánh và sắp xếp UUID

9.1 Để so sánh một cặp UUID, giá trị của các trường tương ứng (xem 6.1) của mỗi UUID được so sánh theo thứ tự mức độ quan trọng (xem 6.1.2). Hai UUID tương đương nhau khi và chỉ khi các trường tương ứng tương đương nhau.

CHÚ THÍCH 1: Thuật toán này so sánh hai UUID tương tự như việc so sánh các giá trị biểu diễn số nguyên đơn quy định tại 6.3.

CHÚ THÍCH 2: So sánh này sử dụng trường vật lý quy định tại Điều 6.1.1, không sử dụng các giá trị liệt kê tại 6.1.3 và quy định tại Điều 12 (Time, Clock, Sequence, Variant, Version và Node).

9.2 Một UUID lớn hơn một UUID khác nếu có giá trị lớn hơn cho trường quan trọng nhất của các trường khác nhau.

9.3 Khi biểu diễn thập lục phân của UUID theo thứ tự bảng chữ cái (Xem điều 6.4), một UUID lớn hơn sẽ đi cùng một UUID nhỏ hơn.

10 Tính hợp lệ

Ngoại trừ việc xác định bit biến thể có được thiết lập chính xác hay không, và giá trị Time được sử dụng trong UUID theo thời gian hay không (và do đó vẫn chưa được ấn định), không có cơ

chế xác định tính hợp lệ của UUID tại thời điểm hiện tại nên tất cả các giá trị khả thi đều có thể xảy ra.

11 Bit biến thể

11.1 Bit biến thể gồm 3 bit quan trọng nhất (bit 7, 6 và 5) của octet 7, đây là octet quan trọng nhất của trường “VariantAndClockSeqHigh”.

11.2 Tất cả UUID phù hợp với Tiêu chuẩn này có bit biến thể cùng với bit 7 của octet 7 được thiết lập cho giá trị 1 và bit 6 của octet 7 được thiết lập cho giá trị 0. Bit 5 của octet 7 là bit quan trọng nhất của ClockSequence và được thiết lập trong Điều 12.4.

CHÚ THÍCH: Bit 5 được liệt kê ở đây là bit biến thể bởi vì giá trị của nó khác với định dạng cũ. Hiểu một cách rõ ràng nó không phải là một phần của giá trị biến thể trong Tiêu chuẩn này, nó chỉ sử dụng hai bit cho giá trị biến thể.

11.3 Bảng 1, liệt kê thông tin sử dụng các giá trị khác nhau của bit biến thể

Bảng 1 – Sử dụng bit biến thể

Bit 7	Bit 6	Bit 5	Mô tả
0	-	-	Lưu trữ để cung cấp tính tương thích NCS sau này
1	0	-	Giá trị biến thể được quy định trong Tiêu chuẩn này
1	1	0	Lưu trữ để cung cấp tính tương thích sau này của tập đoàn Microsoft
1	1	1	Lưu trữ để sử dụng trong tương lai của Tiêu chuẩn này

12 Sử dụng các trường UUID và thứ tự byte khi truyền

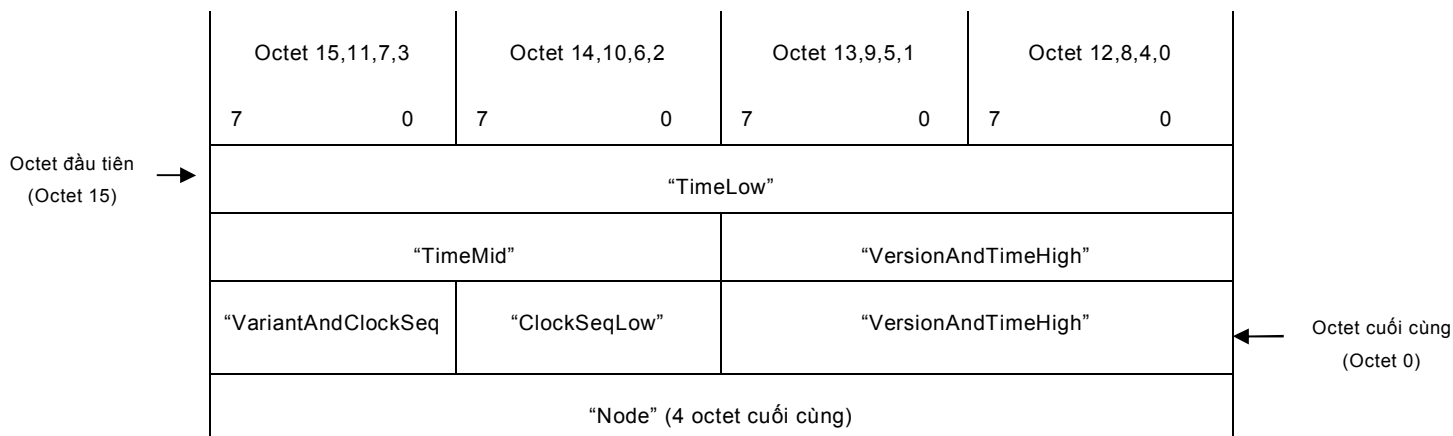
12.1 Yêu cầu chung

12.1.1 Bảng 2 cung cấp vị trí và tóm tắt cách sử dụng các trường UUID khác nhau trong biểu diễn nhị phân

Bảng 2 – Vị trí và cách sử dụng các trường UUID

Trường	Octet # trong UUID	Mô tả
“TimeLow”	15-12	Bit hàng thấp của giá trị Time (32 bits)
“TimeMid”	11-10	Bit hàng giữa của giá trị Time (16 bits)
“VariantAndTimeHigh”	9-8	Version (4 bits) kế tiếp bit hàng cao của Clock Sequence (6 bits)
“VariantAndClockSeqHigh”	7	Bit Variant (2 bits) kế tiếp bit hàng cao của Clock Sequence (6 bits)
“ClockSeqLow”	6	Bit hàng thấp của Clock Sequence (8 bits)
“Node”	5-0	Node (Xem điều 12.5) (48 bits)

12.1.2 Vị trí của trường UUID trong biểu diễn nhị phân được mô tả trong Hình 1



Hình 1 – Vị trí của trường UUID trong biểu diễn nhị phân

12.1.3 Biểu diễn nhị phân được sử dụng để truyền dẫn theo cơ chế liên lạc với 16 octets của biểu diễn nhị phân được truyền đi như một tập hợp liên tiếp của 16 bit cùng với octet đầu tiên (octet 15) đứng trước octet cuối cùng (octet 0).

GHI CHÚ 1: Thứ tự các bit trong một octet được quy định bởi cơ chế truyền thông tin.

GHI CHÚ 2: 16 octet liên tiếp theo thứ tự như quy định ở trên được sử dụng để truyền một UUID nhưng các đặc tả giao thức có thể chọn phương tiện truyền UUID luân phiên, bao gồm sự phân mảnh hoặc truyền đi các phần của UUID (ví dụ như các phần thêm vào giá trị Time).

12.2 Phiên bản

12.2.1 Ba phương thức thay thế để tạo ra UUID (theo thời gian, theo tên, theo số ngẫu nhiên) được xác định và phân biệt bằng bốn bit quan trọng nhất của trường "VersionAndTimeHigh" (Bit 4 đến 7 của octet 9 của UUID). UUID được tạo ra sử dụng các cơ chế khác nhau gọi là "các phiên bản UUID khác nhau".

CHÚ THÍCH – Không hoàn toàn đúng khi mô tả phiên bản này như "phiên bản UUID khác", tuy nhiên tên gọi này vẫn được sử dụng vì một số lý do lịch sử. Không có khái niệm "số phiên bản" truyền thống đối với định dạng UUID, phiên bản mới được xác định như một phiên bản sửa đổi trong tiêu chuẩn này. Bất cứ định dạng UUID mới nào trong tương lai sẽ được định danh bởi một giá trị khác của các bit biến thể.

12.2.2 Bảng 3 liệt kê "UUID version" hiện tại sử dụng 4 bit đầu tiên của trường "VersionAndTimeHigh" (Bit 4 đến 7 của octet 9 của UUID). Nó cũng ấn định giá trị "Version" nguyên cho liên kết của các bit.

CHÚ THÍCH - Giá trị phiên bản của 2 không được sử dụng cho mục tiêu tương thích với định nghĩa cũ của UUID. Giá trị phiên bản của 0 và 6 đến 15 được lưu trữ để sử dụng trong tương lai.

Bảng 3 – Các phiên bản UUID hiện tại

Bit 7	Bit 6	Bit 5	Bit 4	Giá trị phiên bản	Mô tả
0	0	0	1	1	Phiên bản theo thời gian trong Tiêu chuẩn này (xem điều 13)
0	0	1	0	2	Lưu trữ cho phiên bản bảo mật DCE cùng với POSIX UUID kèm theo.
0	0	1	1	3	Phiên bản theo tên quy định trong Tiêu chuẩn này cùng với MD5 (xem điều 14)
0	1	0	0	4	Phiên bản theo số ngẫu nhiên quy định trong Tiêu chuẩn này (Xem điều 15)
0	1	1	1	5	Phiên bản theo tên quy định trong Tiêu chuẩn này cùng với SHA-1 (Xem điều 14)

12.3 Thời gian

12.3.1 Thời gian phải là giá trị 60 bit

CHÚ THÍCH - Tên “Time” thích hợp cho phiên bản theo thời gian của UUID (phiên bản 1), nhưng không được sử dụng cho các nội dung của giá trị tương ứng trong phiên bản khác của UUID (phiên bản 3 và 4).

12.3.2 Đối với phiên bản theo thời gian của UUID, Time sẽ được tính bằng khoảng 100 nano giây của Giờ phối hợp quốc tế (UTC) từ nửa đêm cho đến sáng ngày 15 tháng 10 năm 1582 (ngày theo lịch Gregory có sửa đổi so với lịch Cơ đốc)

CHÚ THÍCH 1 - Trước khi thành lập văn phòng quốc tế de l'Heure (Văn phòng thời gian quốc tế), 1 phút được quy ước có 60 giây. Do đó, các giây xen kẽ xuất hiện khi cần thiết, số giây sẽ tăng (hoặc giảm) mỗi năm.

CHÚ THÍCH 2 - Hệ thống tính thời gian cầm tay có thể xảy ra sự cố khi xác định thời gian UTC, bởi vì hệ thống thường bị khóa theo giờ địa phương. Ngoại trừ trường hợp hệ thống tiếp tục sử dụng thời gian địa phương hoặc thay đổi giá trị Clock Sequence (xem điều 12.4) thì UUID do hệ thống tạo ra sẽ là duy nhất.

CHÚ THÍCH 3: Đối với các hệ thống không truy cập được vào tín hiệu thời gian tổng, một đồng hồ hệ thống sẽ ghi lại thời gian địa phương, thời gian này có thể được sử dụng cùng với thời gian khác, ngoài ra không có UUID được tạo ra khi có thay đổi về thời gian tiết giảm ban ngày, hoặc thay đổi giá trị Clock Sequence (Xem 12.4).

12.3.3 Phiên bản theo tên của UUID là giá trị 60 bit được tạo ra từ một tên chung duy nhất như quy định tại Điều 14.

CHÚ THÍCH - Ví dụ về tên chung duy nhất là OID, URN và các tên phân biệt thư điều (xem [5]).

12.3.4 Phiên bản theo số ngẫu nhiên của UUID là ngẫu nhiên hoặc giả ngẫu nhiên tạo ra giá trị 60 bit như quy định tại Điều 15.

12.4 Chuỗi định giờ

12.4.1 Đối với phiên bản theo thời gian của UUID, Clock Sequence được sử dụng để tránh trùng lặp phát sinh khi giá trị của Time được thiết lập lại hoặc giá trị Node thay đổi.

CHÚ THÍCH - Tên “Clock Sequence” thích hợp cho phiên bản theo thời gian của UUID, nhưng tên này không được sử dụng cho các nội dung của giá trị tương ứng trong phiên bản theo tên và số ngẫu nhiên của UUID.

12.4.2 Nếu giá trị Time được thiết lập lại hoặc có thể được thiết lập lại (ví dụ khi hệ thống bị ngắt) thì bộ phát UUID không thể nhận biết UUID tạo ra giá trị Time lớn hơn giá trị Time đã được thiết lập hay không. Trong trường hợp này, giá trị Clock Sequence sẽ thay đổi.

CHÚ THÍCH - Nếu giá trị trước của Clock Sequence tăng lên thì Clock Sequence sẽ được thiết lập giá trị bảo mật ngẫu nhiên hoặc giá trị giả ngẫu nhiên.

12.4.3 Tương tự như trên, nếu giá trị Node thay đổi (ví dụ, bởi vì các mạng có thể được di chuyển giữa các máy), thì giá trị Clock Sequence sẽ thay đổi.

12.4.4 Clock Sequence sẽ lấy giá trị ban đầu (có nghĩa là một lần trong chu kỳ sống của hệ thống tạo ra UUID) từ một số ngẫu nhiên không dẫn xuất từ giá trị Node.

CHÚ THÍCH - Đây là hình thức tối giản hệ thống tương quan cung cấp bảo vệ tối đa cho địa chỉ MAC, địa chỉ này có thể di chuyển hoặc hoán đổi nhanh từ hệ thống này sang hệ thống khác.

12.4.5 Đối với phiên bản theo tên của UUID, Clock Sequence là giá trị 14 bit được tạo ra từ tên như quy định tại điều 14.

12.4.6 Đối với phiên bản theo số ngẫu nhiên của UUID, Clock Sequence tạo ra giá trị 14 bit ngẫu nhiên hoặc giả ngẫu nhiên như quy định tại Điều 15.

12.5 Nút

12.5.1. Đối với phiên bản theo thời gian của UUID, giá trị Node sẽ bao gồm địa chỉ MAC (Xem ISO/IEC 8802-3) là địa chỉ host của một vài giao thức mạng.

12.5.2. Đối với hệ thống có nhiều địa chỉ MAC, có thể sử dụng bất cứ địa chỉ nào ngoại trừ địa chỉ multicast. Octet 5 của UUID (octet đầu tiên của “Node”) sẽ thiết lập octet đầu tiên của địa chỉ MAC do hệ thống tương ứng ISO/IEC 8802-3 truyền đi.

CHÚ THÍCH 1: Octet này bao gồm bit toàn cục/bit cục bộ và bit đơn nhân/bit phổ rộng. Bit đơn nhân/bit phổ rộng được đặt tại truyền thông đơn hướng để tránh xung đột với địa chỉ được tạo ra theo quy định tại điều 12.5.3.

CHÚ THÍCH 2 - Có thể lấy khóa của địa chỉ MAC từ cơ quan đăng ký địa chỉ MAC (Xem [4]).

12.5.3 Đối với hệ thống không có địa chỉ MAC, số ngẫu nhiên mật mã hoặc số giả ngẫu nhiên có thể được sử dụng (Xem phụ lục C). Bit phổ rộng được thiết lập trong địa chỉ này.

CHÚ THÍCH: Điều này đảm bảo địa chỉ tạo ra không bị xung đột với địa chỉ lấy từ các mạng như đã quy định ở điều 12.5.2.

12.5.4 Đối với UUID theo tên, giá trị Node là giá trị 48 bit được tạo ra bằng cách kết hợp hoặc phân chia từ tên duy nhất như quy định tại điều 14.

12.5.5 Đối với UUID dựa trên số ngẫu nhiên, giá trị Node sẽ tạo ra giá trị 48 bit ngẫu nhiên hoặc giả ngẫu nhiên như quy định tại điều 15.

13 Thiết lập các trường của UUID theo thời gian

Các trường của UUID theo thời gian sẽ được thiết lập như sau:

- Xác định giá trị cho Time theo UTC và Clock Sequence để sử dụng trong UUID như quy định tại điều 12.3 và 12.4.
- Mục tiêu của thuật toán này là xác định Time như số nguyên không dấu 60 bit và Clock Sequence như số nguyên không dấu 14 bit. Các bit kề nhau trong mỗi giá trị cùng với số 0 dành cho bit ít quan trọng nhất.
- Đặt trường “TimeLow” cùng với 32 bit ít quan trọng nhất (từ bit 0 đến bit 31) của Time theo thứ tự quan trọng như nhau.
- Đặt trường “TimeMid” cùng với các bit từ bit 32 đến bit 47 của Time theo thứ tự quan trọng như nhau.
- Đặt 12 bit ít quan trọng (từ bit 0 đến bit 11) của trường “VersionAndTimeHigh” cùng với các bit từ 48 đến 59 của Time theo thứ tự quan trọng như nhau.
- Đặt 4 bit quan trọng nhất (Từ bit 12 đến 15) của trường “VersionAndTimeHigh” vào số phiên bản 4 bit như quy định tại điều 12.2
- Đặt trường “ClockSeqLow” vào tám bit ít quan trọng nhất (Từ bit 0 đến 7) của trường ClockSequence theo thứ tự quan trọng như nhau.
- Đặt sáu bit ít quan trọng nhất (Từ bit 0 đến 5) của trường “VariantAndClockSeqHigh” vào 6 bit quan trọng nhất (từ bit 8 đến 13) của trường Clock Sequence theo thứ tự quan trọng như nhau.
- Đặt 2 bit quan trọng nhất (bit 6 và 7) của trường “VariantAndClockSeqHigh” vào vị trí 0 và 1 tương ứng.
- Đặt trường Node vào địa chỉ MAC 48 bit theo thứ tự quan trọng như nhau.

14 Thiết lập các trường UUID theo tên

Điều này quy định các thủ tục để tạo ra UUID theo tên. Điều 14.1 quy định các thủ tục chung cho hàm băm (Xem ISO/IEC 10118). Điều 14.2 quy định sử dụng MD5 và điều 14.3 quy định sử dụng SHA-1.

CHÚ THÍCH: Sử dụng MD5 giới hạn các trường hợp yêu cầu tương thích với các UUID hiện tại, khi SHA-1 cung cấp thuật toán băm ít tương thích hơn thì giá trị băm giống nhau được tạo ra từ các tài nguyên băm khác (xem C.4).

14.1 Các trường UUID theo tên được thiết lập như sau:

- Phân bố UUID để sử dụng “định danh vùng tên” cho tất cả UUID được tạo ra từ tên trong vùng tên đó.

CHÚ THÍCH: Điều D.9 khuyến cáo rằng UUID sử dụng cho 4 vùng tên phổ thông.

- Chuyển tên thành chuỗi chính tắc của octet (được xác định như tiêu chuẩn hoặc quy ước vùng tên của chuỗi).

- Tính giá trị băm 16 octet của định danh vùng tên liên kết với tên sử dụng hàm băm như quy định tại 14.2 hoặc 14.3. Số lượng octet trong giá trị băm từ 0 đến 15 như quy định trong IETF RFC 1321 (dành cho MD5) và quy định trong FIPS PUB 180-3 dành cho SHA-1.
- Đặt octet từ 0 đến 3 của trường “TimeLow” vào octet từ 0 đến 3 của giá trị băm.
- Đặt octet từ 0 đến 1 của trường “TimeMid” vào octet 5 và 4 của giá trị băm.
- Đặt octet 1 và 0 của trường “VersionAndTimeHigh” vào octet 7 và 6 của giá trị băm.
- Ghi đè 4 bit quan trọng nhất (từ bit 12 đến bit 15) của trường “VersionAndTimeHigh” lên số phiên bản 4 bit trong bảng 3 của điều 12.2 cho hàm băm sử dụng.
- Ghi đè 2 bit quan trọng nhất (bit 7 và 6) của trường “VariantAndClockSeqHigh” lên bit 0 và 1 tương ứng.
- Đặt trường “VariantAndClockSeqHigh” vào octet 8 của giá trị băm.
- Ghi đè hai bit quan trọng (bit 7 và 6) của trường “VariantAndClockSeqHigh” lên bit 1 và 0 tương ứng.
- Đặt trường “ClockSeqLow” vào octet 9 của giá trị băm.
- Đặt octet từ 0 đến 5 của trường “Node” vào octet từ 10 đến 15 của giá trị băm.

14.2 Điều này quy định UUID theo tên sử dụng MD5 như một hàm băm, nhưng MD5 không được sử dụng cho UUID mới (xem C.4). Đối với hàm băm MD5, “giá trị băm” được tham chiếu tại điều 14.1 là giá trị 16 octet quy định bởi IETF RFC 1321 từ octet 0 đến octet 15.

CHÚ THÍCH – Đặc tả của MD5 và số phiên bản liên quan bao gồm tính tương thích trái với quy định cũ.

14.3 Điều này quy định UUID theo tên gọi sử dụng SHA-1 như một hàm băm. Đối với hàm băm SHA-1, “giá trị băm” tham chiếu tại điều 14.1 sẽ là octet từ 0 đến 15 của giá trị 20 octet lấy từ giá trị Message Digest 160-bit quy định bởi FIPS PUB 180-3. Octet từ 16 đến 19 của giá trị 20 octet sẽ bị loại bỏ. Giá trị 20 octet được lấy từ giá trị Message Digest 160-bit của FIPS PUB 180-3 bằng cách đặt bit quan trọng nhất của giá trị 160-bit vào bit quan trọng nhất của octet đầu tiên (octet 0) của giá trị 20 octet, và đặt bit ít quan trọng nhất vào octet cuối cùng (octet 19) của giá trị 20 octet.

15 Thiết lập các trường của UUID theo số ngẫu nhiên

15.1 Các trường của UUID theo số ngẫu nhiên sẽ được thiết lập như sau:

- Đặt hai bit quan trọng nhất (bit 6 và 7) của trường “VariantAndClockSeqHigh” vào vị trí 0 và 1 tương ứng.
- Đặt bốn bit quan trọng nhất (từ bit 12 đến bit 15) của trường “VersionAndTimeHigh” vào số phiên bản 4 bit quy định tại điều 12.2.
- Đặt tất cả các bit khác của UUID vào giá trị được tạo ra ngẫu nhiên hoặc giả ngẫu nhiên.

CHÚ THÍCH: Số giả ngẫu nhiên có thể tạo ra cùng một giá trị nhiều lần. Số ngẫu nhiên dạng mật mã được đặc biệt khuyến cáo sử dụng để giảm khả năng lặp các giá trị.

15.2 Phụ lục C cung cấp hướng dẫn cách thức tạo ra các số ngẫu nhiên trong một hệ thống.

16 Đăng ký UUID và cách sử dụng UUID trong cây định danh đối tượng quốc tế

16.1 Cây định danh đối tượng quốc tế

16.1.1 Tiêu chuẩn này xác định các thủ tục hoạt động của cơ quan đăng ký UUID. Đăng ký này cho phép sử dụng các UUID như giá trị nguyên sơ cấp và cung của cây định danh đối tượng quốc tế thuộc cung liên kết UUID (Xem Điều 7).

CHÚ THÍCH: UUID có thể sử dụng để định danh các cung thuộc cung UUID mà không cần đăng ký, nhưng định danh của các cung như vậy không đảm bảo tính rõ ràng và cách sử dụng này không được chấp nhận.

16.1.2 Cây định danh đối tượng quốc tế là hình thức định danh rõ ràng theo cấu trúc cây phân cấp và cơ quan đăng ký phân cấp độc lập. Cây định danh này có nút gốc, các cung bên dưới nút gốc, các cung bên dưới các cung đó và phân cấp thấp hơn nữa. Các cung được định danh bởi giá trị nguyên bắt buộc đầu tiên (Định danh nhãn Unicode giá trị nguyên) (Xem điều 16.1.7) cung cấp định danh rõ ràng cho một cung trong nút cấp cao. Các cung có thể được cung cấp một hoặc nhiều nhãn Unicode thập phân (chuỗi ký tự Unicode), hoặc định danh thứ cấp. Một đối tượng được định danh bởi chuỗi các nhãn unicode gán cho các cung từ nút gốc đến đối tượng đó.

16.1.3 Các Nhãn Unicode là rõ ràng. Các định danh thứ cấp có thể không rõ ràng. Định danh đối tượng ASN.1 sử dụng giá trị nguyên sơ cấp rõ ràng và duy nhất cho mỗi cung khi mã hóa toàn bộ, nhưng cho phép bao gồm các định danh thứ cấp (không rõ ràng) trong ký pháp giá trị dễ đọc. Kiểu tham chiếu nguồn quốc tế hóa ASN.1 OID sử dụng nhãn Unicode (rõ ràng) trên mỗi cung trong mã hóa và trong ký hiệu giá trị dễ đọc.

16.1.4 Mã hóa giá trị định danh đối tượng ASN.1 thông thường là một mã nhị phân bị nén của chuỗi các giá trị nguyên sơ cấp. Mã hóa giá trị định danh tham chiếu quốc tế ASN.1 OID thông thường là mã hóa ký tự của nhãn Unicode.

CHÚ THÍCH: Xem mô tả đầy đủ của cây OID trong TCVN 10583-1 (ISO/IEC 9834-1 | Recommendation ITU-T X.660).

16.1.5 (HƯỚNG DẪN) UUID không đăng ký có thể được sử dụng như thành phần thể định danh đối tượng thuộc các cung giống nhau của định danh UUID được đăng ký (Xem 16.1.1). Do đó, Giá trị định danh giống nhau của các cung khác nhau sử dụng một UUID đã đăng ký và một UUID không đăng ký (hoặc hai UUID không đăng ký) cũng có thể được sử dụng, mặc dù khả năng xảy ra trường hợp này là rất nhỏ. Khả năng này sẽ tăng lên nếu UUID được tạo ra từ giá trị băm MD5 hoặc số giả ngẫu nhiên, mà không phải tạo ra từ giá trị băm SHA-1 và số ngẫu nhiên dạng mật mã. Điều này có thể gây ra xung đột giữa người sử dụng OID và khởi đầu các nguy cơ, ví dụ như lừa đảo. Cơ quan đăng ký UUID chịu trách nhiệm cho xung đột xảy ra giữa các UUID đã đăng ký, nhưng không chịu trách nhiệm cho xung đột giữa một UUID đã đăng ký và một UUID không đăng ký bởi vì cơ quan này không quản lý các UUID không đăng ký. Nếu xảy ra xung đột, ngữ nghĩa liên kết với UUID đã đăng ký sẽ được ưu tiên, ngữ nghĩa liên kết với giá trị UUID không đăng ký sẽ không được sử dụng. Do đó, đăng ký của một OID với một UUID không đảm

bảo OID này có tính duy nhất hơn UUID. Mục tiêu của đăng ký này là công khai UUID và ngữ nghĩa của chúng.

16.1.6 Biểu diễn dữ liệu trong máy tính của định danh đối tượng hoặc định danh nguồn quốc tế hóa OID chỉ ra (theo ngữ cảnh của biểu diễn dữ liệu trong máy tính) định danh của đường dẫn từ gốc đến một đối tượng trong cây OID. Do đó, nếu định danh đối tượng hoặc giá trị định danh nguồn quốc tế hóa OID được tạo thành như quy định tại Điều 7 thì một biểu diễn dữ liệu có thể bao gồm duy nhất giá trị UUID.

16.1.7 Giá trị nguyên sơ cấp của các cung là giá trị nguyên dương không giới hạn độ lớn. Nhãn Unicode trên các cung là dãy của ký tự Unicode, với điều kiện chúng không phải là các số nguyên khác với giá trị nguyên sơ cấp của cung.

16.2 Chỉ định cơ quan đăng ký

16.2.1 ITU-T|ISO/IEC được ủy nhiệm tổ chức đăng ký theo quy định trong tiêu chuẩn này. Để thực hiện việc đăng ký, theo yêu cầu và quy định của mình, ITU-T|ISO/IEC chỉ định một tổ chức thực hiện theo RA cho tiêu chuẩn này.

CHÚ THÍCH: Để đăng ký, Sử dụng đường dẫn <http://www.itu.int/ITU-T/asn1/uuid.html> (Vị trí dự án ITU-T ASSN.1)

16.2.2 Cơ quan đăng ký sẽ không chịu trách nhiệm cho bất cứ sai sót nào trong quá trình hoạt động theo các thủ tục này hoặc cho bất kỳ hoạt động nào liên quan đến trách nhiệm của cơ quan đăng ký theo Tiêu chuẩn này, ngoại trừ trường hợp nó được miễn trách nhiệm mà không bị phạt bởi nhóm nghiên cứu | tiểu ban kỹ thuật liên quan. Cơ quan đăng ký sẽ không chịu trách nhiệm cho việc sử dụng giá trị OID không đăng ký giống giá trị OID đã đăng ký, bởi vì cơ quan này không kiểm soát việc sử dụng các giá trị đó xem 16.1.5).

CHÚ THÍCH: Nếu Nhóm nghiên cứu | Tiểu ban kỹ thuật quyết định cơ quan đăng ký được miễn trách nhiệm vì nguyên nhân này hay bất kỳ nguyên nhân nào khác, thông tin đăng ký bởi RA sẽ luôn sẵn có cho bất kỳ RA được chỉ định sau này.

16.3 Lệ phí

16.3.1 Tổ chức cung cấp RA này sẽ hoạt động trên cơ sở phục hồi lệ phí. Cơ cấu lệ phí dự kiến để phục hồi lệ phí điều hành RA, đăng ký xuất bản web, hỗ trợ yêu cầu truy vấn và ngăn cản việc xuất hiện nhiều yêu cầu và các yêu cầu vô lý.

16.3.2 Lệ phí sẽ do RA quyết định, phụ thuộc vào phê duyệt của nhóm nghiên cứu | Tiểu ban kỹ thuật liên quan. Lệ phí có thể áp dụng cho:

- a) Đăng ký;
- b) Yêu cầu truy vấn;
- c) Công khai trên web;
- d) Yêu cầu cập nhật;

16.3.3 Lệ phí sẽ độc lập với quốc gia mà ứng dụng được tạo, phụ thuộc vào sự biến động tỷ giá

16.4 Thủ tục đăng ký

Điều này quy định các thủ tục tuân theo đăng ký của UUID. Các thủ tục này dự kiến để đảm bảo quy trình mở và đóng trong vận hành của RA.

16.4.1.1 Tổ chức trực tiếp trình RA đơn đăng ký UUID bằng cách điền vào mẫu đăng trên website. Nội dung của đơn được quy định tại điều 14.4.3.

16.4.1.2 Tùy thuộc vào việc hoàn thành các thủ tục đăng ký, giá trị UUID 128 bit được đăng ký cũng như chuyển giao cho (hoặc bởi) tổ chức nhận, và sau đó sẽ được xuất bản.

16.4.2 Quy trình xác nhận

Đăng ký thành công được xác nhận qua phản hồi trên web và ứng dụng web.

16.4.3 Nội dung của đơn

16.4.3.1 Điều này quy định thông tin do RA yêu cầu để kiểm soát quá trình đăng ký.

CHÚ THÍCH: Tại thời điểm xuất bản tiêu chuẩn này, thông tin có thể được gửi tới bằng email, điện thoại, đĩa cứng hoặc đăng ký trên website.

16.4.3.2 Đăng ký bao gồm các thông tin sau

- a) Quốc gia nơi tổ chức đăng ký có trụ sở chính;
- b) Tên của tổ chức cùng với thông tin đăng ký nếu công ty, tổ chức từ thiện được đăng ký, v.v...hoặc chi nhánh công ty được biết như là một tổ chức quốc tế;
- c) Tên và chức danh, địa chỉ thư tín, địa chỉ email, điện thoại, fax để liên lạc trong tổ chức đăng ký.
- d) Thông tin dạng tự do để xác minh tính chân thực của tổ chức đăng ký được xem như phương tiện kiểm tra và loại bỏ các đăng ký giả mạo.
- e) (Phần không bắt buộc) Có thể truy cập URL để cung cấp thêm thông tin về việc sử dụng UUID.

16.4.3.3 Nội dung của đơn chung cho OID được quy định bởi TCVN 10583-1 (ISO/IEC 9834-1| Recommendation ITU-T X.660), Điều 8.

16.5 Duy trì sổ đăng ký trên Web

16.5.1 RA sẽ duy trì một website để lựa chọn sổ đăng ký cho tất cả các đăng ký.

16.5.2 Thông tin về tổ chức liên quan đến việc đăng ký được cập nhật bằng lệ phí của RA nếu RA được cung cấp các chi tiết thay đổi về tên công ty hoặc các thông tin tương tự và quyền thay đổi thích hợp. Cơ chế thực hiện điều này do RA quyết định và sẽ được thông báo trên website.

Phụ lục A

(Tham khảo)

Thuật toán tạo ra UUID theo thời gian

Phụ lục này mô tả thuật toán được sử dụng để liên tục tạo ra UUID theo thời gian trong hệ thống máy tính

A.1 Thuật toán cơ bản

A.1.1 Thuật toán sau đây đơn giản, chính xác nhưng không hiệu quả:

- Lấy khóa toàn hệ thống.
- Bộ nhớ ổn định chia sẻ dữ liệu của một hệ thống lớn (ví dụ: tệp) đọc trạng thái của bộ tạo UUID: Giá trị Time, Clock Sequence và Node được sử dụng để tạo ra UUID cuối cùng.
- Lấy thời gian hiện tại khi đếm 60 bit trong khoảng 100 nano giây từ 00:00:00:00 ngày 15 tháng 10 năm 1528 vào giá trị Time.
- Lấy giá trị Node hiện tại.
- Nếu giá trị Node ở trạng thái không có sẵn (ví dụ không tồn tại hoặc bị ngắt) hoặc khác so với giá trị Node hiện tại thì sẽ tạo ra giá trị Clock Sequence ngẫu nhiên.
- Nếu giá trị Time ở trạng thái có sẵn nhưng giá trị Time lưu trữ tới muộn hơn giá trị Time hiện tại thì giá trị Clock Sequence sẽ tăng lên.
- Lưu trạng thái (giá trị Time, Clock Sequence và Node hiện tại) trở về bộ nhớ ổn định.
- Ngắt đồng hồ tổng
- Định dạng UUID từ giá trị Time, ClockSequence và Node theo từng bước trong Điều 13.

A.1.2 Nếu UUID không thường xuyên được tạo ra thì thuật toán trên đây hoàn toàn đầy đủ. Tuy nhiên, đối với các yêu cầu thể hiện cao hơn thì các vấn đề với thuật toán đơn giản bao gồm:

- Đọc trạng thái từ bộ nhớ ổn định mỗi lần là không hiệu quả;
- Độ chính xác của đồng hồ hệ thống là 100 nano giây;
- Viết trạng thái vào bộ nhớ ổn định mỗi lần là không hiệu quả;
- Chia sẻ trạng thái ngoài giới hạn quy trình có thể không hiệu quả.

A.1.3 Mỗi kết quả có thể được ghi theo kiểu mô đun bằng cách cải thiện cục bộ các chức năng đọc, viết trạng thái và đọc giờ. Chúng được ấn định địa chỉ lần lượt trong các điều nhỏ.

A.2 Bộ nhớ đọc ổn định

A.2.1 Trạng thái chỉ cần được đọc từ bộ nhớ ổn định một lần tại thời điểm khởi động nếu trạng thái đó được đưa vào bộ nhớ khả biến của hệ thống lớn (và được cập nhật bất cứ khi nào bộ nhớ ổn định được cập nhật)

A.2.2 Nếu một quá trình thực hiện không có bộ nhớ ổn định sẵn có, thì có thể khẳng định rằng các giá trị không dùng được. Quá trình thực hiện này ít được mong đợi nhất bởi vì nó làm tăng tần số tạo ra số Clock Sequence mới, dẫn đến tăng khả năng trùng lặp.

A.2.3 Nếu giá trị Node không bao giờ thay đổi (ví dụ các mạng không gắn liền với hệ thống) hoặc nếu bất kỳ thay đổi nào khởi tạo lại Clock Sequence theo giá trị ngẫu nhiên thì thay vì giữ giá trị trong bộ nhớ ổn định, giá trị Node hiện tại sẽ được trả lại.

A.3 Tính chính xác của đồng hồ hệ thống

A.3.1 Giá trị Time được tạo ra từ thời gian hệ thống có độ chính xác ít hơn độ chính xác của giá trị Time được yêu cầu.

A.3.2 Nếu UUID không được tạo ra thường xuyên thì giá trị Time đơn giản là thời gian hệ thống được phân chia thành số trong khoảng 100 nano giây cho mỗi khoảng thời gian hệ thống.

A.3.3 Nếu hệ thống bị khởi động quá do bộ sinh yêu cầu nhiều UUID trong khoảng thời gian hệ thống đơn thì dịch vụ UUID sẽ trả lại lỗi, hoặc ngắt bộ sinh UUID cho đến khi đồng hồ hệ thống bắt kịp.

A.3.4 Giá trị thời gian chính xác có thể được mô phỏng bằng cách liên tục đếm số lượng UUID được tạo ra giá trị giống nhau của thời gian hệ thống và sử dụng nó để tạo ra bit hàng thấp của giá trị Time. Khoảng đếm sẽ từ 0 đến 100 nano giây cho mỗi khoảng thời gian hệ thống.

CHÚ THÍCH: Nếu bộ xử lý thường xuyên bị tràn UUID thì các địa chỉ MAC thêm vào được phân bổ cho hệ thống sẽ cho phép phân bổ tốc độ cao bằng cách tạo ra nhiều UUID sẵn có cho mỗi giá trị Time.

A.4 Bộ nhớ ghi ổn định

Không cần thiết phải ghi trạng thái vào bộ nhớ ổn định khi UUID được tạo ra. Giá trị Time trong bộ nhớ ổn định được thiết lập định kỳ cho một giá trị lớn hơn giá trị sử dụng trong UUID. Miễn là UUID được tạo ra có giá trị Time nhỏ hơn giá trị đó, giá trị Clock Sequence và Node không bị thay đổi, khi đó chỉ bản chụp sự thay đổi của trạng thái cần được cập nhật. Ngoài ra, nếu giá trị Time của bộ nhớ ổn định trong tương lai nhỏ hơn thời gian qui chuẩn yêu cầu hệ thống khởi động lại thì sự cố này là nguyên nhân khởi tạo lại Clock Sequence.

A.5 Chia sẻ trạng thái qua các quy trình

Nếu việc truy cập vào trạng thái chia sẻ khi tạo ra UUID quá tốn kém thì bộ sinh của hệ thống lớn có thể hoạt động để phân bổ khóa của giá trị Time tại thời điểm bị gọi, và bộ tạo mỗi quy trình từ khóa đó có thể phân bổ khóa cho đến khi kết thúc.

Phụ lục B

(Tham khảo)

Đặc tính của UUID theo tên

B.1 UUID theo tên nghĩa là tạo ra một UUID từ một tên được lấy từ và duy nhất trong một vài vùng tên. Khái niệm tên và vùng tên cần được phân tích đại thể và không giới hạn tên dạng văn bản. Các cơ chế hoặc chuyển đổi để phân bố tên và đảm bảo tính duy nhất của tên, vùng tên của tên nằm ngoài phạm vi của Tiêu chuẩn này.

CHÚ THÍCH: Để tránh các vấn đề lặp lại, UUID theo tên sẽ không được tạo ra từ một OID kết thúc cùng với UUID theo tên.

B.2 Đặc tính của UUID theo tên được tạo ra theo điều 14 và vùng tên được lựa chọn phù hợp như sau:

- UUID được tạo ra tại các thời điểm khác nhau từ tên giống nhau trong vùng tên giống nhau sẽ tương đương nhau;
- UUID được tạo ra từ hai tên khác nhau trong vùng tên giống nhau sẽ có xác suất khác nhau rất cao.
- UUID được tạo ra từ tên giống nhau trong hai vùng tên khác nhau sẽ có xác suất khác nhau rất cao.
- Nếu hai UUID theo tên tương đương nhau thì hai UUID này có khả năng được tạo ra từ tên giống nhau trong vùng tên giống nhau.

Phụ lục C

(Tham khảo)

Phát sinh các số ngẫu nhiên trong một hệ thống

C.1 Nếu một hệ thống không tạo ra số ngẫu nhiên dạng mật mã thì đa số hệ thống có số lượng lớn ngang nhau các nguồn ngẫu nhiên sẵn có từ hệ thống có khả năng tạo ra số ngẫu nhiên dạng mật mã. Các nguồn này đều rõ ràng một cách hệ thống, nhưng thường bao gồm:

- Phần trăm sử dụng bộ nhớ;
- Kích thước của bộ nhớ chính tính bằng byte;
- Số lượng byte trống trong bộ nhớ chính;
- Các byte trống của trang hoặc tệp trung gian;
- Kích cỡ tính theo byte của khoảng địa chỉ ảo dành cho người sử dụng;
- Tổng số byte sẵn có của khoảng địa chỉ dành cho người sử dụng;
- Kích cỡ tính theo byte của ổ đĩa khởi động;
- Kích cỡ tính theo byte của ổ đĩa mềm;
- Kích cỡ tính theo byte của vùng đĩa trống trên ổ mềm;
- Thời gian hiện tại;
- Tổng thời gian từ hệ thống được khởi động;
- Kích cỡ riêng của tệp trong các thư điều hệ thống khác nhau;
- Chuẩn bị tệp, đọc lần cuối và sửa đổi thời gian của tệp trong các thư điều hệ thống khác nhau;
- Nhân tố đầu tiên của các nguồn hệ thống khác nhau (vùng lưu trữ heap, v...v);
- Vị trí con trỏ chuột hiện tại;
- Vị trí dấu caret (ký hiệu số mũ hoặc lũy thừa bậc);
- Số lượng hiện tại của các quy trình, các chuỗi hiện thời;
- Ô điều khiển hoặc các ID của cửa sổ màn hình và cửa sổ hiện hành;
- Giá trị con trỏ ngăn xếp của bộ gọi;
- Quy trình và định danh chuỗi của bộ gọi;
- Cấu trúc bộ xử lý khác nhau quy định việc đếm các biểu diễn (các lệnh được thực hiện, thiếu vùng nhớ đệm, hoặc thiếu TLB)

C.2 Ngoài ra, các điều như tên máy tính và tên của hệ điều hành giúp phân biệt các kết quả lấy từ hệ thống khác.

C.3 Thuật toán chính xác tạo ra giá trị Node sử dụng dữ liệu này có tính rõ ràng bởi vì dữ liệu sẵn có và các hàm chứa chúng thường rất rõ ràng. Tuy nhiên, cần tiếp cận tổng quát để tích lũy được nhiều nguồn vào bộ đệm và sử dụng ký hiệu truyền tin như SHA-1, lấy 6 octet bất kỳ từ giá trị băm và thiết lập bit đổi kiểu như mô tả trên.

C.4 Các hàm băm khác như MD5 và hàm băm quy định tại ISO/IEC 10118 có thể được sử dụng. Yêu cầu duy nhất là kết quả phù hợp ngẫu nhiên nghĩa là đầu ra từ một tập hợp các đầu vào được phân bố giống nhau và các đầu vào này cũng phân bố giống nhau, một bit đơn thay đổi trong đầu vào có thể làm một nửa các bit đầu ra thay đổi. (Sử dụng MD5 nhưng không sử dụng UUID mới bởi vì các nghiên cứu hiện nay cho thấy giá trị đầu vào không được phân bố đồng đều).

Phụ lục D

(Tham khảo)

Biểu diễn mẫu

D.1 Các tệp dự phòng

Biểu diễn này bao gồm 6 tệp: Copyrt.h, uuid.c, sysdep.h, sysdep.c và utest.c. Tệp uuid.* biểu diễn độc lập của thuật toán tạo ra UUID quy định tại Điều 13, 14 và 15 như mô tả tối ưu tại phụ lục A (ngoại trừ trạng thái đầy đủ phân chia cho các quy trình). Giả thiết rằng mã hỗ trợ số nguyên 64 bit trong đó làm cho mã này rõ ràng nhiều.

CHÚ THÍCH – Mã được kiểm tra bởi Linux (Red Hat 4.) cùng với GCC (2.7.2) và Win NT 4 cùng với VC++ 5.0.

D.2 Tệp copyrt.h

Tất cả các tệp nguồn được đánh giá để đưa ra thông báo bản quyền sau đây:

/*

***Bản quyền (c) 1990 – 1993, 1996 Tập đoàn sáng lập phần mềm mở.*

***Bản quyền (c) 1989 của công ty Hewlett-Packard, palo Alto, Ca. &*

***Tập đoàn thiết bị số Maynard, Mass.*

***Đối với các cá nhân nhận thức rằng tệp này do “AS IS” cung cấp nhưng không có
 **thông tin hoặc bảo đảm nào về việc cho phép sử dụng, sao chép, sửa đổi và phân phối
 tệp vì bất kỳ mục tiêu nào mà không trả lệ phí, trừ khi thông báo bản quyền **bên trên và
 thông báo bản quyền này sao chép mã nguồn và không sử dụng tên của **Tập đoàn sáng
 lập mã nguồn mở, công ty Hewlett-Packard hoặc tập đoàn thiết bị **số trong quảng cáo
 hoặc liên hệ công khai để phân phối phần mềm không được **cấp phép bằng văn bản rõ
 ràng trước đó. Tập đoàn sáng lập mã nguồn mở, công ty **Hewlett-Packard hoặc tập đoàn
 thiết bị số không đưa ra tuyên bố nào về tính phù hợp của phần mềm này vì bất kỳ mục
 tiêu nào.*

D.3 Tệp 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 - - tạo ra một UUID */
Int uuid_create (uuid_t * uuid);
/* uuid_create_from_name - - tạo ra một UUID sử dụng "name" từ "name
space"*/
Void uuid_create_from_name {
```

```

    uuid_t *uuid,          /* trả về UUID */
    uuid_t nsid,           /* UUID của vùng tên */
    void *name,            /* tên tạo ra UUID */
    int namelen            /* độ dài của tên */
};
/ * uuid_compare - So sánh hai "lexically" của UUID và trả giá trị
    -1      u1 đúng trước u2
    0      u1 bằng u2
    1      u1 sau u2

```

Chú thích: Thứ tự lexical không phải là thứ tự tạm thời

```

/*
Int uuid_compare (uudi_t *u1, uuid_t *u2);

```

D.4 Tập uuid.c

```

#include "copyrt.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#ifdef _WINDOW_
#include <arpa/inet.h>
#endif
#include "uuid.h"
/* Các khai báo chuyển tiếp khác nhau*/
Static int read_state (unsigned16 *clockseq, uuid_time_t *timestamp,
    uuid_node_t *node);
static void write_state (unsigned16 clockseq, uuid_tiem 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 unsigned16 true_random(void);
/* uuid_create - - tạo ra UUID */
Int uuid_create (uuid_t *uuid)
{
    uuid_time_t timestamp, last_time;
    unsigned16 clockseq;
    uuid_node_t last_node;
    int f;
    /* yêu cầu đồng hồ tổng vì vậy chúng được xếp một mình */
    LOCK;
    /* Lấy time, node identifier, saved state từ bộ nhớ không ổn định */
    get_current_time (&timestamp);
    get_ieee_node_identifier (&node);
    f = read_state (&clockseq, &last_time, &last_node);
    /* Nếu không có trạng thái NV, hoặc nếu đồng hồ bị chậm, hoặc định danh
    Node thay đổi (ví dụ như các mạng mới) làm thay đổi clockseq */
    If (! F || memcmp (&node, &last_node, sizeof node))
        Clockseq = true_random();
    Else if (timestamp < last_time)
        Clockseq++;
    /* lưu trạng thái cho lần tiếp theo*/
    Write_state (clockseq, timestamp, node);
    UNLOCK
    /* gán trường vào UUID */
    Format_uuid_v1 (uuid, clockseq, timestamp, node);
    Return 1;
}
/* format_uuid_v1 - - tạo ra UUID từ timestamp, clockseq và node
identifier */

```

```

Void format_uuid_v1(uuid_t* uuid, unsigned16 clock_seq, uuid_time_t
timestamp, uuid_node_t node)
{
    /* tạo ra phiên bản 1 UUID cùng thông tin được tập hợp thêm hằng số.*/
    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);
}
/* Kiểu dữ liệu cho trạng thái ổn định tạo ra UUID*/
typedef struct {
    uuid_time_t ts; /* timestamp được lưu trữ */
    uuid_node_t node; /* node identifier được lưu trữ */
    unsigned16 cs; /* Clock Sequence được lưu trữ */
} uuid_state;
static uuid_state st;
/* read_state -- đọc trạng thái tạo ra UUID từ bộ nhớ không ổn định*/
int read_state(unsigned16 *clockseq, uuid_time_t *timestamp, uuid_node_t
*node)
{
    static int initied = 0;
    FILE *fp;
    /* Chỉ cần đọc trạng thái một lần mỗi khi khởi động */
    if (!initied) {
        fp = fopen("state", "rb");
        if (fp == NULL)
            return 0;
        fread(&st, sizeof st, 1, fp);
        fclose(fp);
        initied = 1;
    }
    *clockseq = st.cs;
    *timestamp = st.ts;
    *node = st.node;
    return 1;
}
/* write_state -- lưu trạng thái tạo ra UUID về bộ nhớ không ổn định */
void write_state(unsigned16 clockseq, uuid_time_t timestamp, uuid_node_t
node)
{
    static int initied = 0;
    static uuid_time_t next_save;
    FILE* fp;
    if (!initied) {
        next_save = timestamp;
        initied = 1;
    }
    /* Luôn lưu trạng thái đến trạng thái chia sẻ ổn định */
    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);
    }
}

```

```

        /* schedule next save for 10 seconds from now */
        next_save = timestamp + (10 * 10 * 1000 * 1000);
    }
}

/* get-current_time -- Lấy giá trị time là 60 bit 100 nano giây từ
   UUID. Bù độ chính xác đồng hồ thực nhỏ hơn 100 nano giây */
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);
        /* Nếu đồng hồ đọc thay đổi khi tạo ra UUID cuối cùng, */
        if (time_last != time_now) {
            /* Đếm lại uuid được tạo ra cùng đồng hồ đọc */
            uuids_this_tick = 0;
            time_last = time_now;
            break;
        }
        if (uuids_this_tick < UUIDS_PER_TICK) {
            uuids_this_tick++;
            break;
        }
        /* đồng hồ chạy quá nhanh; quay vòng */
    }
    /* Bỏ sung việc đếm UUID vào bit hàng thấp của đồng hồ đọc */
    *timestamp = time_now + uuids_this_tick;
}

/* true_random - tạo ra số ngẫu nhiên mã hóa.
   ** Ví dụ này không thực hiện điều đó.** */
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 -- tạo ra UUID sử dụng "name" từ "namespace" */
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;
    /* Đặt định danh vùng tên theo thứ tự byte trên mạng mặc dù chúng đặt
    trong kiểu thứ tự byte ngược*/
    net_nsid = nsid;
    htonl(net_nsid.time_low);

```

```

    htons(net_nsid.time_mid);
    htons(net_nsid.time_hi_and_version);

    MD5lnit(&c);
    MD5Update(&c, &net_nsid, sizeof net_nsid);
    MD5Update(&c, name, namelen);
    MD5Final(hash, &c);

    /* giá trị băm trong thứ tự byte mạng tại vị trí này */
    format_uuid_v3(uuid, hash);
}
/* format_uuid_v3 -- tạo ra UUID từ số 128 bit ngẫu nhiên (giả ngẫu nhiên)
*/

void format_uuid_v3(uuid_t *uuid, unsigned char hash[16])
{
    /* Chuyển UUID theo thứ tự byte cục bộ */
    memcpy(uuid, hash, sizeof *uuid);
    ntohl(uuid->time_low);
    ntohs(uuid->time_mid);
    ntohs(uuid->time_hi_and_version);

    /* Đặt vào bit biến thể và bit phiên bản */
    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 -- So sánh hai "lexically" của UUID và trả lại giá trị */
#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 Tập sysdep.h

```

#include "copyrt.h"
/* Loại bỏ xác định sau đây nếu không chạy 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"
/* Thay đổi đối với chu kỳ của MD5. I/IETF RFC 1321 có một biểu diễn mẫu*/
#include "md5.h"
/* Thiết lập sau đây vào số 100 nano giây cho độ chính xác thực của đồng hồ
hệ thống*/
#define UUIDS_PER_TICK 1024
/* Thiết lập sau đây cho cuộc gọi để nhận và trả đồng hồ tổng*/
#define LOCK
#define UNLOCK
typedef unsigned long unsigned32;
typedef unsigned short unsigned16;
typedef unsigned char unsigned8;
typedef unsigned char byte;
/* Thiết lập này cho chương trình biên dịch sử dụng kiểu dữ liệu 64 bit */
#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 Tập sysdep.c

```

#include "copyrt.h"
#include <stdio.h>
#include <string.h>
#include "sysdep.h"
/* Lệnh call độc lập trong hệ thống nhận định danh node MAC. Biểu diễn mẫu
này tạo ra định danh ngẫu nhiên*/
void get_ieee_node_identifier(uuid_node_t *node)
{
static int initied = 0;
static uuid_node_t saved_node;
unsigned char seed[16];
FILE *fp;
if (!initied) {
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);
}
}
}
initied = 1;
}

```

```

*node = saved_node;
}
/* Lệnh call đọc lập trong hệ thống nhận thời gian hệ thống hiện tại. Trả
giá trị 100 nano giây từ giai đoạn UUID */
#ifdef _WINDOWS_
void get_system_time(uuid_time_t *uuid_time)
{
    ULARGE_INTEGER time;
    /* Win NT giữ time theo định dạng Tập TIME là 100 nano giây từ tháng
    1/1601. UUID sử dụng thời gian 100 nano giây từ 15/10/1582. Sự khác biệt ở
    đây là 17 ngày trong tháng 10 + ngày 30 (tháng 11) + ngày 31 (tháng 12)
    + 18 năm và 5 ngày nhuận */
    GetSystemTimeAsFileTime((FILETIME *)&time);
    time.QuadPart +=
        (unsigned int64) (1000*1000*10)          // giây
        * (unsigned int64) (60 * 60 * 24)        // ngày
        * (unsigned int64) (17+30+31+365*18+5);   // # của ngày
    *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;
    } r;
    DWORD tc;
    DWORD l;
    char hostname[MAX_COMPUTERNAME_LENGTH + 1];
    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);
    /* Khoảng trống giữa thời gian định dạng UUID và thời gian định dạng Unix.
    Thời gian theo UUID UTC là 15/10/1582.
    Thời gian theo Unix là 1/1/1970 */
    *uuid_time = (tp.tv_sec * 1000000) + (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 Tập 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 - In ra một UUID */
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");
}
/* Ở đây đơn giản cho bộ phát UUID */
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); /* sẽ là giá trị 0 */
    f = uuid_compare(&u, &NameSpace_DNS);
    printf("uuid_compare(u, NameSpace_DNS): %d\n", f); /* sẽ là giá trị 1 */
    f = uuid_compare(&NameSpace_DNS, &u);
    printf("uuid_compare(NameSpace_DNS, u): %d\n", f); /* sẽ là giá trị -1 */
    uuid_create_from_name(&u, NameSpace_DNS, "www.widgets.com", 15);
    printf("uuid_create_from_name(): "); puid(u);
}

```

D.8 Ví dụ của utest

```

uuid_create(): 7d444840-9dc0-11d1-b245-5ffdc74fad2
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 Một vài ID vùng tên

Điều này liệt kê ID vùng tên cho các vùng tên tiềm năng như cấu trúc đầu tiên trong ngôn ngữ C và biểu diễn chuỗi được xác định bên trên.

```

/* Chuỗi Name là tên miền đầy đủ */
uuid_t NameSpace_DNS = { /* 6ba7b810-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b810,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};
/* Name string là một URL */
uuid_t NameSpace_URL = { /* 6ba7b811-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b811,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};
/* Name string là một OID */
uuid_t NameSpace_OID = { /* 6ba7b812-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b812,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};
/* Name string là tên phân biệt thu điều (trong DER or định dạng
xuất ký tự*/
uuid_t NameSpace_X500 = { /* 6ba7b814-9dad-11d1-80b4-
00c04fd430c8 */
    0x6ba7b814,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

```

Thư mục tài liệu tham khảo

- [1] Zahn, L., Dineen, T., Leach, P. (January 1990), Network computing architecture , ISBN 0-13-611674.-4.
- [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 system 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 Address, <http://standards.ieee.org/regauth/oui/pilot-ind.html>.
- [5] ITU-T Recommendation X.500 (2008) | ISO/IEC 9594-1:2008 *Information technology – Open system Interconnection – The Directory: Overview of concept, models and services*
- [6] IETF RFC 3061 (2001), *A URN Namespace of Object identifiers*.
- [7] IETF RFC 4122 (2005), *A universally unique identifier (UUID) URN Namespace*
- .
-