

## Chương 3

# Cơ sở thông tin quản lý (MIB)

---

- Cấu trúc của MIB version 1 (SMIv1)
- Cấu trúc của MIB version 2 (SMIv2)
- MIB-2 (RFC1213), mib chuẩn dành cho các thiết bị TCP/IP.
- Host-resources-mib (RFC2790), mib chuẩn dành cho host.
- Thực hiện các phương thức SNMP bằng một chương trình MIB Browser trên Windows
- Thực hiện các phương thức SNMP bằng net-snmp-utils trên Linux

Chương 2 đã trình bày cách sử dụng một số phần mềm giám sát SNMP phổ biến như PRTG hay Solarwinds. Đây là các công cụ giám sát tự động, chúng định kỳ thực hiện “quét” lấy các thông tin một lượt và hiển thị lên dạng biểu đồ thống kê theo thời gian. Mỗi lần quét chúng thực hiện một loạt các phương thức SNMP GetRequest/GetNextRequest.

Trong chương này chúng ta sẽ khảo sát cấu trúc của mib, một số mib phổ biến và cách dùng các tiện ích thực thi các phương thức SNMP để lấy các thông tin mà ta muốn. Tìm hiểu chương này sẽ giúp bạn :

- + Hiểu được cấu trúc của mib và đặc tả các mib phổ biến như mib-2 hay host-resources-mib.
- + Hiểu được cách các phần mềm giám sát SNMP lấy thông tin như thế nào.
- + Sử dụng được các công cụ SNMP Browser trên Windows & Linux.
- + Có nền tảng hỗ trợ cho việc lập trình phần mềm SNMP.

## 1. SMIv1

MIB là một cấu trúc dữ liệu định nghĩa các đối tượng được quản lý, được thiết kế để quản lý các thiết bị không chỉ riêng TCP/IP. RFC1155<sup>1</sup> mô tả cấu trúc của mib file, cấu trúc này gọi là SMI (Structure of Management Information). Sau này người ta mở rộng thêm cấu trúc của mib thành SMI version 2, và phiên bản trong RFC1155 được gọi là SMIv1.

Trước khi đi vào tìm hiểu cấu trúc của mib, chúng ta phải đi sơ lược qua một chuẩn gọi là ASN.1 :

- + ASN.1 (Abstract Syntax Notation One) là chuẩn mô tả các luật mã hóa dữ liệu (encoding rules) cho các hệ thống truyền thông số. Một trong 3 hệ thống luật mã hóa trong ASN.1 là BER (Basic Encoding Rules). BER được SNMP dùng làm phương pháp mã hóa dữ liệu. Vì vậy trong các RFC liên quan đến SNMP ta hay bắt gặp dòng ghi chú “use of the basic encoding rules of ASN.1”.
- + BER mô tả nhiều kiểu dữ liệu như : BOOLEAN, INTEGER, ENUMERATED, OCTET STRING, CHOICE, OBJECT IDENTIFIER, NULL, SEQUENCE, ....
- + Chúng ta sẽ dành hẳn một chương để nói về các luật mã hóa của “BER of ASN.1” và cách đọc bản tin SNMP từ việc phân tách các byte dựa vào luật BER.

Quay lại RFC1155, mỗi đối tượng bao gồm 3 phần : Name, Syntax và Encoding.

### Name

Name là định danh của object, có kiểu OBJECT IDENTIFIER. OBJECT IDENTIFIER là một chuỗi thứ tự các số nguyên biểu diễn các nút (node) của một cây từ gốc đến ngọn.

Gốc (root node) trong mib không có tên. Dưới root là 3 node con :

- + ccitt(0) : do CCITT quản lý (Consultative Committee for International Telephone and Telegraph).
- + iso(1) : do tổ chức ISO quản lý (International Organization for Standardization).
- + joint-iso-ccitt(2) : do cả ISO và CCITT quản lý.

Dưới node iso(1), tổ chức ISO thiết kế 1 node dành cho các tổ chức khác là org(3).

Dưới org(3) có nhiều node con, một node được dành riêng cho US Department of Defense, dod(6).

Bộ Quốc phòng Mỹ được coi là nơi sáng lập ra mạng Internet, dưới dod(6) chỉ có 1 node dành cho cộng đồng internet ngày nay, là node internet(1).

Tất cả mọi thứ thuộc về cộng đồng Internet đều nằm dưới .iso.org.dod.internet, mọi object của các thiết bị TCP/IP đều bắt đầu với prefix .1.3.6.1 (dấu chấm đầu tiên biểu diễn rằng .iso là cây con của root, và root thì không có tên)

RFC1155 định nghĩa các cây con như sau :

internet	OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
directory	OBJECT IDENTIFIER ::= { internet 1 }
mgmt	OBJECT IDENTIFIER ::= { internet 2 }
experimental	OBJECT IDENTIFIER ::= { internet 3 }
private	OBJECT IDENTIFIER ::= { internet 4 }
enterprises	OBJECT IDENTIFIER ::= { private 1 }

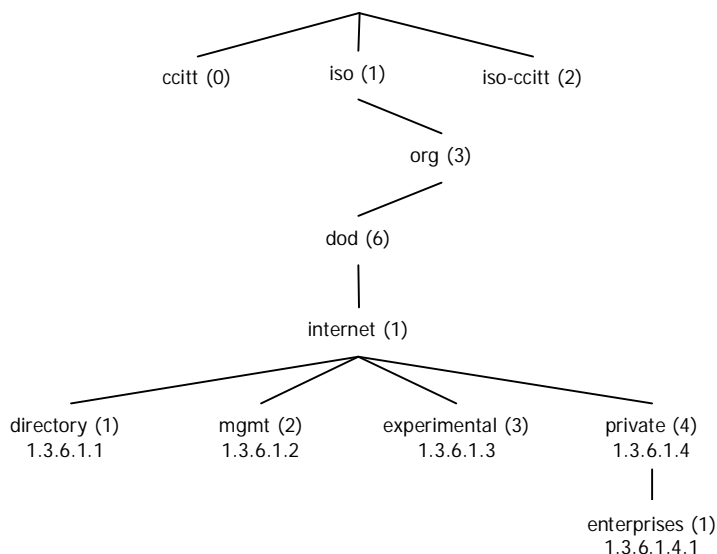
+ directory : dành riêng cho tương lai nếu dịch vụ OSI Directory được sử dụng trên internet.

<sup>1</sup> RFC1155 – Structure and Identification of Management Information for TCP/IP-based Internets

+ mgmt (management) : tất cả các mib chuẩn chính thức của internet đều nằm dưới mgmt. Mỗi khi một RFC mới về mib ra đời thì tổ chức IANA (Internet Assigned Numbers Authority) sẽ cấp cho mib đó một object-identifier nằm dưới mgmt.

+ experimental : dùng cho các object đang trong quá trình thử nghiệm, được IANA cấp phát.

+ private : dùng cho các object do người dùng tự định nghĩa, tuy nhiên các chỉ số cũng do IANA cấp. Tất cả các đơn vị cung cấp hệ thống mạng có thể đăng ký object-identifier cho sản phẩm của họ, chúng được cấp phát dưới node private.enterprises.



Hình : SMIV1 (RFC1155)

## Syntax

Syntax mô tả kiểu của object là gì. Syntax được lấy từ chuẩn ASN.1 nhưng không phải tất cả các kiểu đều được hỗ trợ. SMIV1 chỉ hỗ trợ 5 kiểu nguyên thủy (primitive types) lấy từ ASN.1 và 6 kiểu định nghĩa thêm (defined types).

Primitive types : INTEGER, OCTET-STRING, OBJECT-IDENTIFIER, NULL, SEQUENCE.

Defined types :

- + NetworkAddress : kiểu địa chỉ internet (ip).
- + IPAddress : kiểu địa chỉ internet 32-bit (ipv4), gồm 4 octet liên tục.
- + Counter : kiểu số nguyên không âm 32-bit và tăng đều, khi số này tăng đến giới hạn thì phải quay lại từ 0. Giá trị tối đa là  $2^{32}-1$  (4294967295).
- + Gauge : kiểu số nguyên không âm 32-bit, có thể tăng hoặc giảm nhưng không tăng quá giá trị tối đa  $2^{32}-1$ .
- + TimeTicks : kiểu số nguyên không âm, chỉ khoảng thời gian trôi qua kể từ một thời điểm nào đó, tính bằng phần trăm giây. VD từ khi hệ thống khởi động đến hiện tại là 1000 giây thì giá trị sysUpTime=100000.
- + Opaque : kiểu này cho phép truyền một giá trị có kiểu tùy ý nhưng được đóng lại thành từng OCTET-STRING theo quy cách của ASN.1.

## Encoding

Cơ chế Encoding như đã nói, là chuẩn BER trong ASN.1 – sẽ được trình bày trong chương 5.

## Cấu trúc kiểu OBJECT-TYPE

RFC1155 quy định cấu trúc của một record "định nghĩa đối tượng quản lý" (a managed object definition), kiểu dữ liệu này gọi là OBJECT-TYPE, các tài liệu mib khác khi viết định nghĩa cho một managed object nào đó thì phải theo quy định của SMI. Một "Managed Object Definition" có kiểu OBJECT-TYPE bao gồm các trường :

- + SYNTAX : kiểu của object, là một trong các primitive types hoặc defined types ở trên.
- + ACCESS : mức truy nhập của object, mang một trong các giá trị read-only, read-write, write-only, not-accessible.
- + STATUS : mang một trong các giá trị mandatory (bắt buộc phải hỗ trợ), optional (có thể hỗ trợ hoặc không), obsolete (đã bị thay thế). Một agent nếu hỗ trợ một chuẩn mib nào đó thì bắt buộc phải hỗ trợ tất cả các object có status=mandatory, còn status=optional thì có thể hỗ trợ hoặc không.
- + DESCRIPTION : dòng giải thích cho ý nghĩa của object.

Ví dụ định nghĩa cho object sysUpTime trong RFC1213 :

```
sysUpTime OBJECT-TYPE
    SYNTAX      TimeTicks
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The time (in hundredths of a second) since the
         network management portion of the system was last
         re-initialized."
    ::= { system 3 }
```

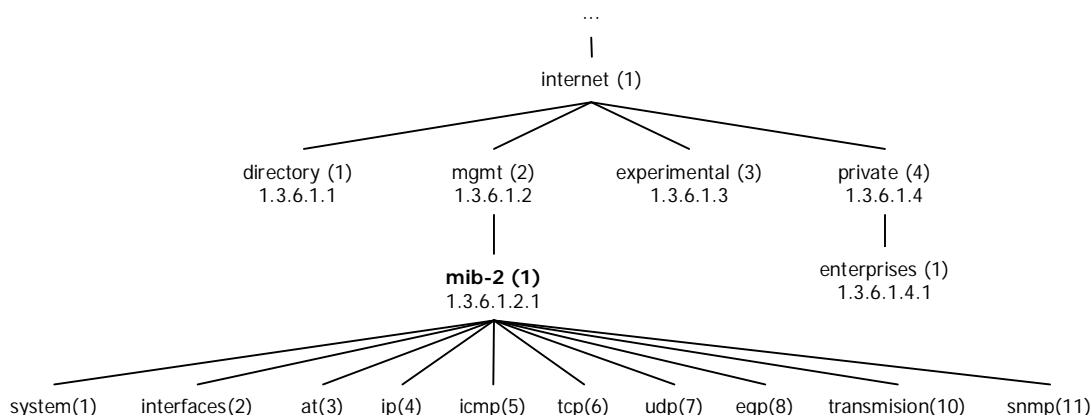
## 2. Mib-2 (RFC1213)

RFC1155 mô tả cách trình bày một mib file như thế nào chứ không định nghĩa các object. RFC1213 là một chuẩn định nghĩa nhánh mib nằm dưới iso.org.dod.internet.mgmt.mib-2 (tất nhiên phải theo cấu trúc mà RFC1155 quy định). Chúng ta sẽ khảo sát một phần RFC1213 để hiểu ý nghĩa của một số object trước khi dùng công cụ để đọc chúng.

RFC1156 là đặc tả mib chuẩn cho các thiết bị TCP/IP, được coi là Internet-Standard Mib (mib version 1). RFC1213 là đặc tả mib chuẩn version 2, thường gọi là mib-2. Chú ý phân biệt mib-1 và mib-2 là các chuẩn đặc tả định nghĩa của các object, còn SMIV1 và SMIV2 là đặc tả cấu trúc của mib file. Mib-1 và mib-2 sử dụng cấu trúc của SMIV1.

Mib-2 là một trong những mib được hỗ trợ rộng rãi nhất. Nếu một thiết bị được tuyên bố là có hỗ trợ SNMP thì hãng sản xuất phải chỉ ra nó hỗ trợ các RFC nào, và thường là RFC1213. Nhiều bạn chỉ biết thiết bị của mình "có hỗ trợ SNMP" nhưng không rõ hỗ trợ các RFC nào, và dùng phần mềm giám sát SNMP hỗ trợ RFC1213 để giám sát thiết bị nhưng không thu được kết quả. Lý do là phần mềm thì hỗ trợ RFC1213 nhưng thiết bị thì không.

Vị trí của MIB-2 trong mib như sau :



Các kiểu dữ liệu mới được định nghĩa trong mib-2 gồm :

- + Display String : kế thừa từ kiểu OCTET STRING nhưng chỉ bao gồm các ký tự in được (printable characters) và dài không quá 255 ký tự.
- + Physical Address : giống kiểu OCTET STRING, được dùng để biểu diễn địa chỉ vật lý của thiết bị.

Trong mib-2 có 10 group, tài liệu này chỉ trình bày định nghĩa các object trong group *system* và *interfaces* do chúng ta sẽ thực hành dùng các công cụ đọc mib để đọc các object này. Để có được mô tả đầy đủ các bạn hãy tham khảo RFC1213.

Name		Syntax	Description
mib-2(1)			Internet-standard mib version 2 (RFC1213) OID : .1.3.6.1.2.1
system(1)			
	sysDescr(1)	DisplayString	Dòng văn bản mô tả node hiện đang hỗ trợ mib này, có thể bao gồm tên, version, kiểu phần cứng, hệ điều hành, ...
	sysObjectID(2)	Object identifier	Định danh đã được đăng ký của hàng sản xuất hệ thống. Giá trị này phải khó nhầm lẫn và miêu tả được đây là loại thiết bị gì
	sysUpTime(3)	TimeTicks	Thời gian tính từ khi module quản trị mạng của hệ thống khởi động lại (kiểu TimeTicks tính bằng phần trăm giây)
	sysContact(4)	DisplayString	Dòng văn bản chỉ định người cần liên lạc nếu có các vấn đề đối với hệ thống
	sysName(5)	DisplayString	Tên được gán cho node để quản lý
	sysLocation(6)	DisplayString	Vị trí vật lý đặt node
	sysServices(7)	Integer	Chỉ ra node có thể hoạt động ở các layer nào của OSI. Giá trị của nó là tổng tất cả các $2^{(Layer-1)}$ với Layer là số lớp OSI. VD một router hoạt động ở lớp 3 thì giá trị này sẽ là $2^{(3-1)}=4$
interfaces(2)			
	ifNumber(1)	Integer	Tổng số giao tiếp mạng hiện có trong hệ thống
	ifTable(2)	Sequence	Danh sách các thông tin của từng interface
	ifEntry(1)	ifEntry	Một entry chứa các object mang thông tin của một interace trong danh sách
	ifIndex(1)	Integer	Giá trị duy nhất của mỗi interface, giá trị này chạy từ 1 đến ifNumber, và không thay đổi ít nhất cho đến khi hệ thống khởi động lại
	ifDescr(2)	DisplayString	Dòng text mang thông tin của một interface
	ifType(3)	Integer	Kiểu interface, dựa vào giao thức lớp physical/link của interface. VD : ethernetCsmacd(6), fddi(15), e1(19), atm(37), sonet(39), v35(45), ...
	ifMtu(4)	Integer	Kích thước của datagram lớn nhất có thể truyền/nhận trên interface
	ifSpeed(5)	Gauge	Bảng thông hiện tại của interface, tính bằng bit per second
	ifPhysAddress(6)	PhysicalAddress	Địa chỉ vật lý của interface
	ifAdminStatus(7)	Integer	Trạng thái mong muốn của interface
	ifOperStatus(8)	Integer	Trạng thái hoạt động thực tế của interface
	ifLastChange(9)	TimeTicks	Giá trị của sysUpTime tại thời điểm interface đi vào trạng thái hoạt động như hiện tại
	ifInOctets(10)	Counter	Tổng số octet đã nhận trên interface

				ifInUcastPkts(11)	Counter	Số gói unicast được đưa đến giao thức lớp cao hơn
				ifInNUcastPkts(12)	Counter	Số gói nonunicast được đưa đến giao thức lớp cao hơn (broadcast, multicast)
				ifInDiscards(13)	Counter	Số gói tin nhận được bị hủy (kể cả các gói không bị lỗi) để ngăn không cho chúng đến tầng xử lý cao hơn, vd khi tràn bộ đệm nhận.
				ifInErrors(14)	Counter	Số gói tin nhận được có chứa lỗi
				ifInUnknownProtos(15)	Counter	Số gói tin nhận được từ interface nhưng bị discard vì nó thuộc giao thức không được hỗ trợ
				ifOutOctets(16)	Counter	Tổng số octet đã truyền ra interface
				ifOutUcastPkts(17)	Counter	Tổng số gói tin unicast mà tầng giao thức cao hơn yêu cầu truyền ra (kể cả các gói sẽ bị discard)
				ifOutNUcastPkts(18)	Counter	Tổng số gói tin non-unicast mà tầng giao thức cao hơn yêu cầu truyền ra (kể cả các gói sẽ bị discard)
				ifOutDiscards(19)	Counter	Số gói tin cần truyền ra bị hủy (kể cả các gói không bị lỗi) để ngăn không cho chúng đến tầng xử lý cao hơn, vd khi tràn bộ đệm phát
				ifOutErrors(20)	Counter	Số gói tin không thể truyền ra do có lỗi
				ifOutQLen(21)	Gauge	Độ dài của hàng đợi bản tin truyền đi
				ifSpecific(22)	Object identifier	Tham chiếu đến định nghĩa mib dành riêng cho loại media của interface. VD nếu interface thuộc ethernet thì giá trị này chỉ ra tài liệu mô tả các object của riêng ethernet. Nếu node không cung cấp được thông tin này thì giá trị của ifSpecific phải là .0.0

Cấu trúc của mib là dạng cây, để xác định object identifier của một object bạn phải đi từ gốc đến object đó. Ví dụ : bandwidth của interface thứ 3 trên thiết bị thì có OID là .1.3.6.1.2.1.2.2.1.5 (.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifSpeed.3).

Chú ý : mặc dù mib-2 đã quy định index của từng interface phải liên tục và chạy từ 1 đến ifNumber, nhưng trong thực tế nhiều thiết bị không đặt index liên tục mà đặt theo cách riêng để dễ quản lý. Do đó đối với C2950 thì interface thứ 3 có index là 3, nhưng đối với thiết bị khác thì interface thứ 3 có thể có index khác 3, thậm chí là số rất lớn. Chẳng hạn một switch có nhiều card, mỗi card có 12 port thì port1-card1 có index là 101, port12-card1 có index là 112, port1-card2 có index là 201.

### 3. Sử dụng iReasoning MIB Browser trên Windows

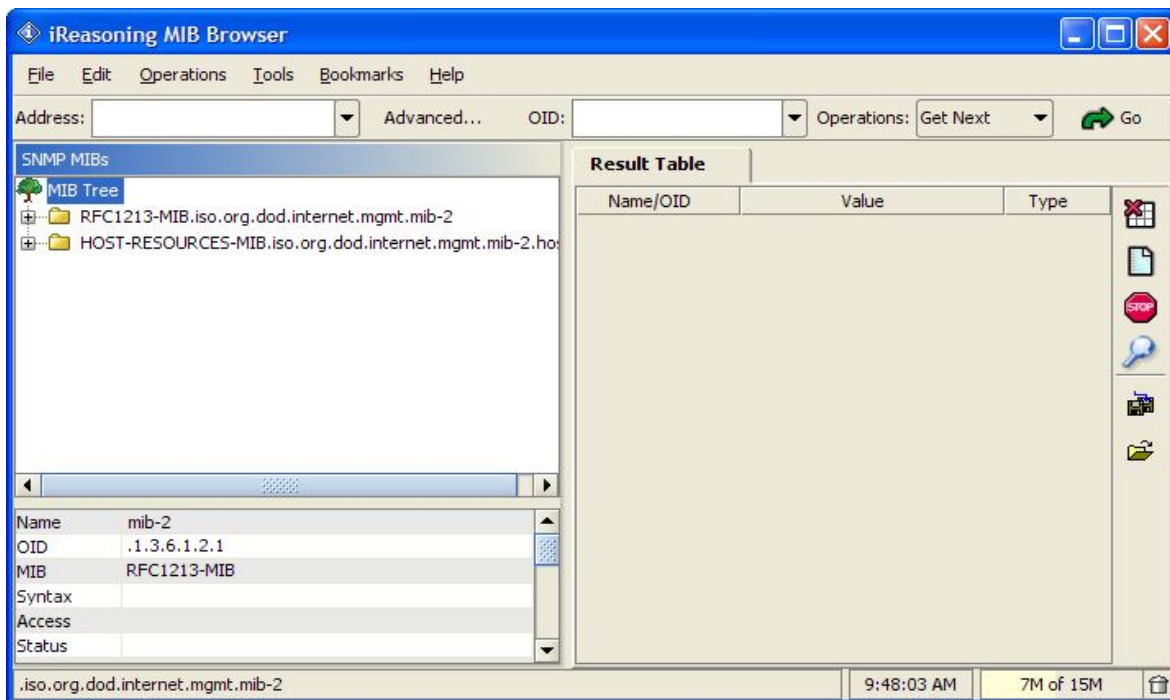
#### iReasoning MIB Browser

Để lấy thông tin của một object trong mib ta cần có một phần mềm có thể thực hiện các phương thức SNMP (Get, GetNext, Set). Các phần mềm này gọi chung là MIB Browser, hiện có rất nhiều và hầu hết đều có phiên bản miễn phí.

Một trong những công cụ mib browser để sử dụng là iReasoning MIB Browser. Phần mềm này được viết trên Java nên có thể chạy trên Windows lẫn Linux, nó cung cấp cả giao diện đồ họa và dòng lệnh. Phiên bản hiện tại là Personal Edition 6.33, download tại <http://www.ireasoning.com/download.shtml>

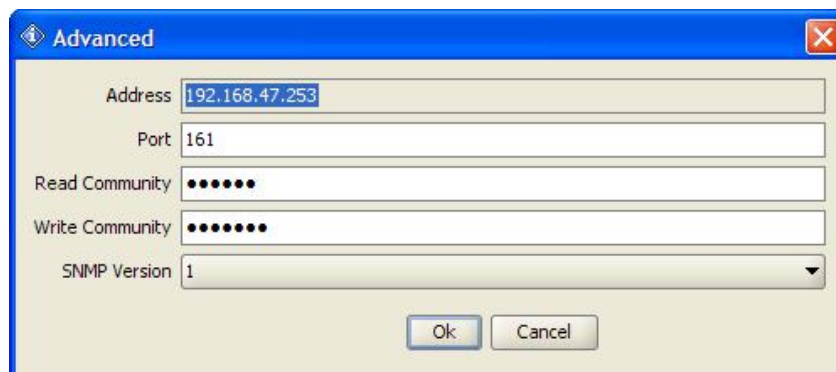
Trong phần này chúng ta sẽ dùng iReasoning MIB Browser để lấy thông tin của một switch Cisco Catalyst 2950. Thông tin được lấy thuộc đặc tả mib-2.


Sau khi cài đặt và khởi động, giao diện của phần mềm như sau



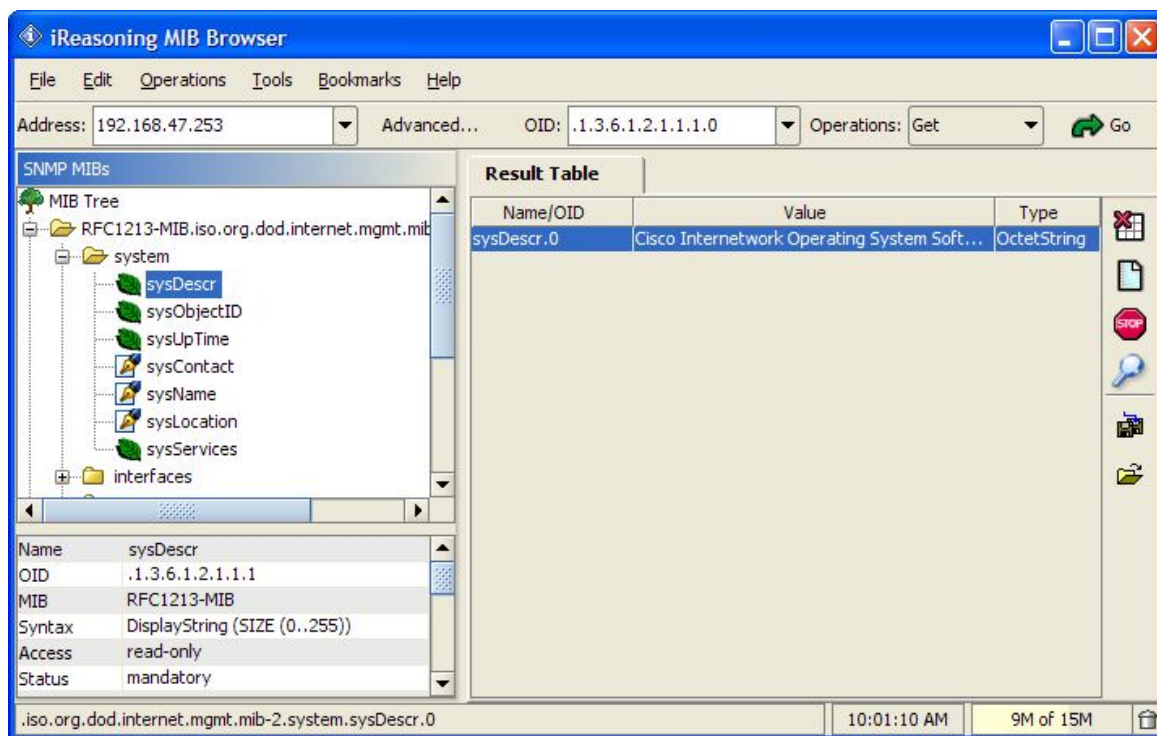
### Lấy thông tin bằng Get/GetNextRequest dùng MIB Browser

Đầu tiên bạn khai báo đầy đủ tính năng SNMP agent trên C2950 như chương 2 đã trình bày. Kế tiếp trên iReasoning MIB browser bạn khai báo như sau : nhập IP của switch vào ô [Address]; nhấn nút [Advanced], nhập các thông tin Read/Write Community (mặc định là public/private), chọn version SNMP là 1 (các version khác chúng ta sẽ thực hiện sau), nhấn [OK].



Sau đó, mở cây RFC1213.system, chọn object sysDescr; bên phải màn hình chọn Operation là GET và nhấn nút  Go, MIB Browser sẽ gửi bản tin GetRequest có OID là 1.3.6.1.2.1.1.0 (sysDescr.0) đến thiết bị, thiết bị gửi lại bản tin GetResponse và phần mềm hiển thị thông tin trả lời lên màn hình.

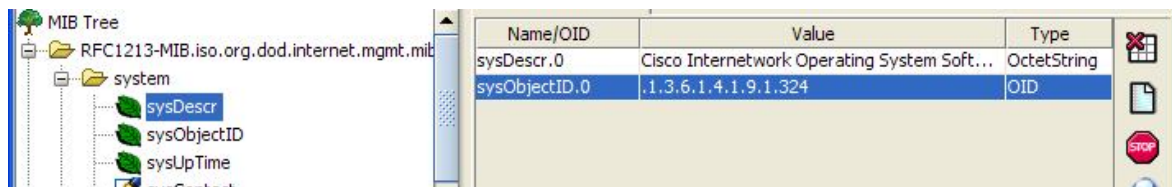




Khi bạn click vào object sysDescr bạn sẽ thấy các thông tin định nghĩa về object này hiển thị ở panel phía dưới như Name, OID, MIB, Syntax, Access, Status. Phần mềm MIB Browser biết được các thông tin này chính là nhờ vào RFC1213, mib file này được lưu ở đường dẫn cài đặt chương trình ..\ireasoning\mibbrowser\mibs\rfc1213.

Kết quả trả lời cho sysDescr là "Cisco Internetwork Operating System Software ...", kiểu là OctetString.

Tiếp theo bạn chọn Operation là GetNext, phương thức GetNext lấy về thông tin của object kế tiếp. Lúc này phần mềm sẽ gửi bản tin GetNextRequest có OID vẫn là .1.3.6.1.2.1.1.0 (sysDescr.0) nhưng thiết bị sẽ gửi bản tin trả lời GetResponse có OID là .1.3.6.1.2.1.1.2.0 (sysObjectID.0), vì sysObjectID là object kế tiếp sysDescr trong mib-2.

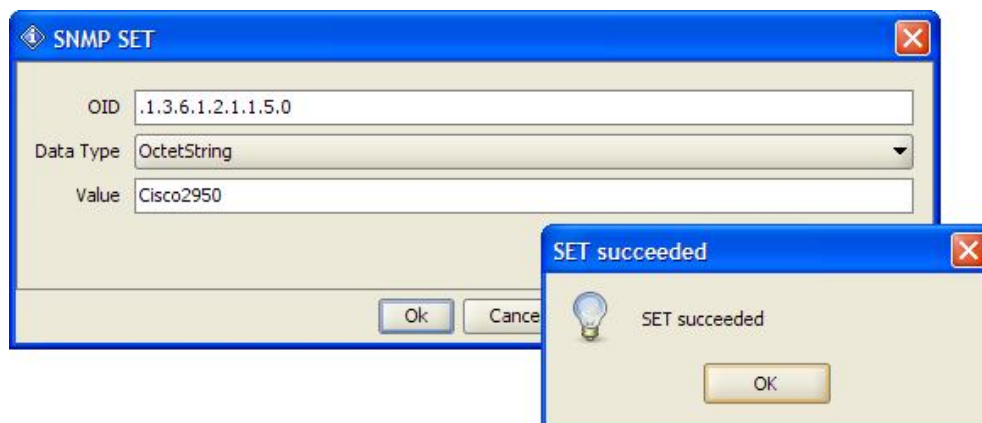


Giá trị trả về lần này là giá trị của sysObjectID, là .1.3.6.1.4.1.9.1.324, có kiểu Object Identifier. Giá trị này có nghĩa là .iso.org.dod.internet.private.enterprises.cisco.1.324, trong đó prefix đầu là enterprise number của Cisco (9) còn 1.324 là OID của thiết bị switch C2950 do Cisco tự định nghĩa. Bạn có nhớ trong chương 2, phần mềm PRTG khi auto-discover con switch này đã phát hiện ra nó là [Cisco ISO Cisco Switch] hay không? PRTG đã dựa vào sysObjectID để nhận dạng ra Cisco Switch, dĩ nhiên là nó phải có CSDL nhận dạng đầy đủ.

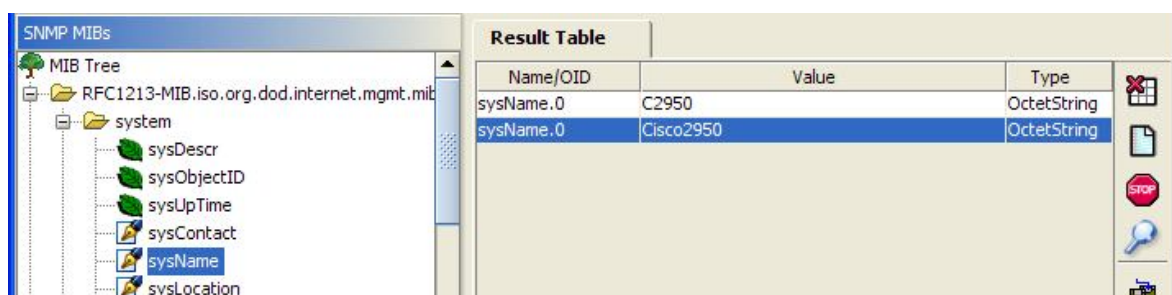
### Thay đổi thông tin bằng SetRequest dùng MIB Browser

Trong panel bên trái, những object có hình 🌿 là read-only, hình 🖋️ là read-write. Tiếp theo ta sẽ thực hiện phương thức Set để thay đổi thông tin thiết bị qua SNMP. Bạn click chọn object sysName và thực hiện Get để lấy thông tin trước khi thay đổi. Sau đó bạn chọn phương thức Set và nhấn nút [Go]. Trong hộp thoại [SNMP SET] bạn nhập giá trị cần thay đổi vào [Value] và nhấn [OK], nếu write community là đúng thì thao tác Set sẽ được thực hiện thành công.





Thực hiện lại phương thức Get, bạn sẽ thấy sysName của thiết bị đã mang giá trị mới.

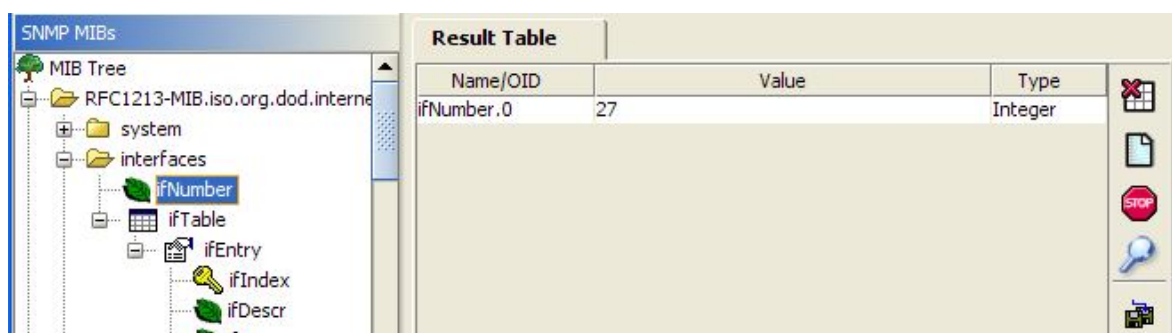


Login vào thiết bị bạn sẽ thấy hostname của switch đã thay đổi.

### Giám sát lưu lượng interface bằng MIB Browser

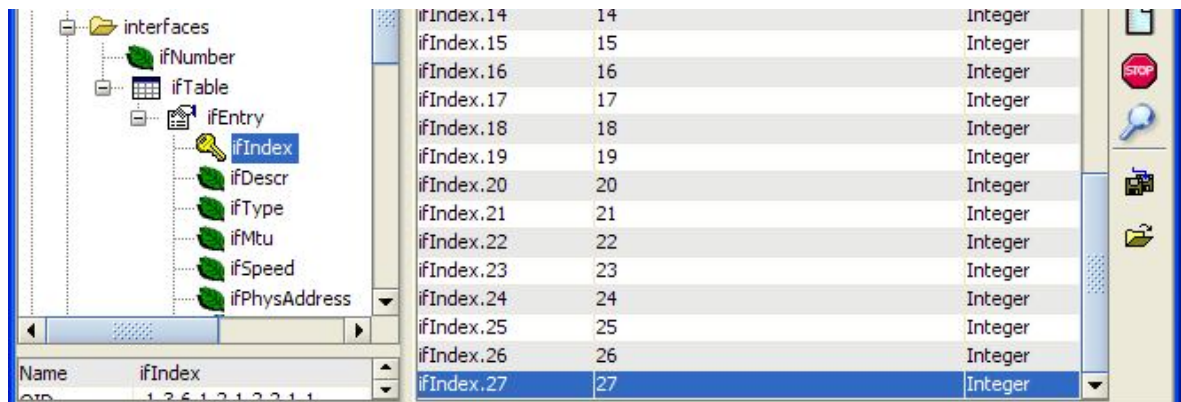
Bây giờ chúng ta sẽ thực hiện các thủ tục phức tạp hơn để giám sát lưu lượng interface trên switch, thông tin cần lấy bao gồm tên interface, description, speed, current status, lưu lượng đã truyền/nhận. Mục này chỉ nhằm giúp người đọc hiểu các phần mềm giám sát lưu lượng hoạt động theo trình tự như thế nào, trong thực tế không ai dùng MIB Browser để giám sát lưu lượng thủ công cả.

Đầu tiên bạn phải lấy số lượng interface có trên C2950, chọn node interfaces.ifNumber, thực hiện GET. C2950 của tác giả cho biết có 27 interface, switch của bạn có thể có số lượng khác.

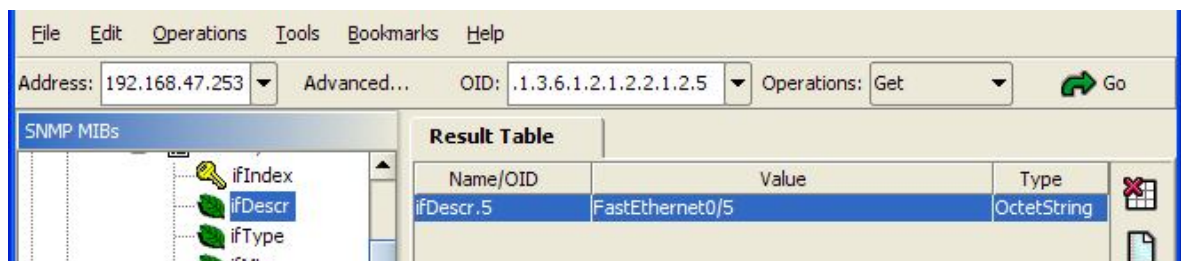


Tiếp theo bạn lấy index của từng interface, vẫn click vào node ifNumber, chọn GetNext, bạn sẽ lấy được object kế tiếp của ifNumber, tức là index của interface đầu tiên, vì ifIndex là object accessible kế tiếp ifNumber (ifTable và ifEntry là not-accessible). Trong hình dưới bạn sẽ thấy interface đầu tiên có index là 1 (.1.3.6.1.2.1.2.2.1.ifIndex.1). Có thể bạn sẽ hỏi tại sao không sử dụng GetRequest trực tiếp trên node ifIndex.1 mà phải sử dụng GetNextRequest trên node nằm trước đó? Như phía trên đã lưu ý, nhiều thiết bị không dùng ifIndex trong khoảng từ 1 đến ifNumber, và index của interface đầu tiên chưa hẳn là 1, nên nếu dùng GetRequest ifIndex.1 trên nhiều thiết bị khác có thể sẽ gặp lỗi "agent not responding".

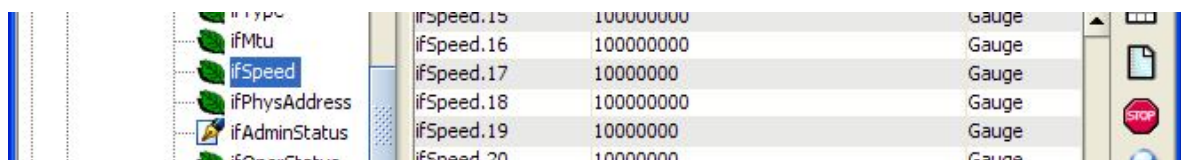
Lần lượt thực hiện GetNext liên tục đến khi nào đủ số lượng index của 27 interface. Cuối cùng bạn sẽ được index của các interface được đánh số từ 1 đến 27.



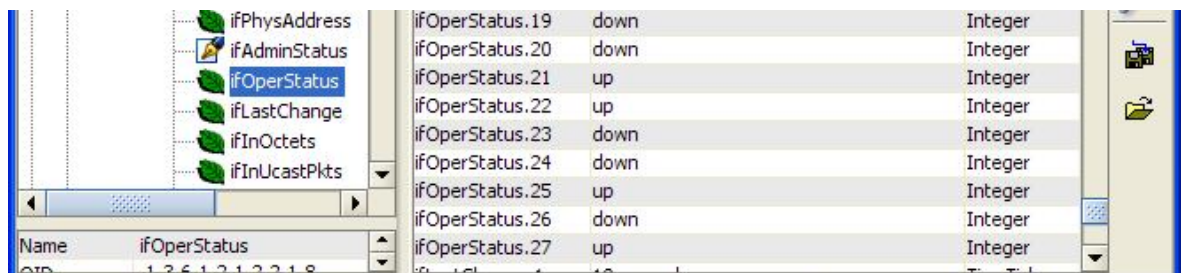
Nếu tiếp tục GetNext thì bạn sẽ lấy được các dòng mô tả của 27 interface (ifDescr). Lúc này bạn có thể lấy thông tin của interface bất kỳ vì bạn đã biết index của chúng. Ví dụ để lấy description của interface thứ 5, bạn hãy nhập vào hộp "OID" giá trị ".1.3.6.1.2.1.2.2.1.2.5" và thực hiện Get.



Tiếp theo bạn phải lấy bandwidth của các interface (ifSpeed), một số interface có speed là 10Mbps, một số khác là 100Mbps.

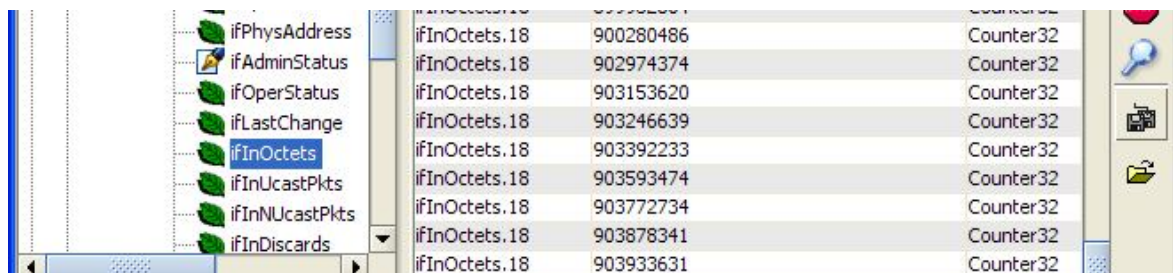


Tiếp theo nữa bạn phải lấy trạng thái hoạt động của từng interface để biết interface nào đang up hay down. Theo RFC1213, ifOperStatus có kiểu là INTEGER, có thể mang các giá trị : up(1), down(2), testing(3), unknown(4), dormant(5), notPresent(6), lowerLayerDown(7).



Để theo dõi lưu lượng, bạn phải thực hiện lấy liên tục tổng số octet đã nhận (ifInOctets) và tổng số octet đã truyền (ifOutOctets), tức là thực hiện phương pháp Poll. Tốc độ hiện tại được tính bằng lấy số octet tăng thêm giữa 2 lần poll chia cho thời gian giữa 2 lần poll.

Hình sau minh họa kết quả lấy ifInOctets của interface số 18.



Object	Value	Type
ifPhysAddress	900280486	Counter32
ifAdminStatus	902974374	Counter32
ifOperStatus	903153620	Counter32
ifLastChange	903246639	Counter32
ifInOctets	903392233	Counter32
ifInUcastPkts	903593474	Counter32
ifInNUcastPkts	903772734	Counter32
ifInDiscards	903878341	Counter32
ifInOctets.18	903933631	Counter32

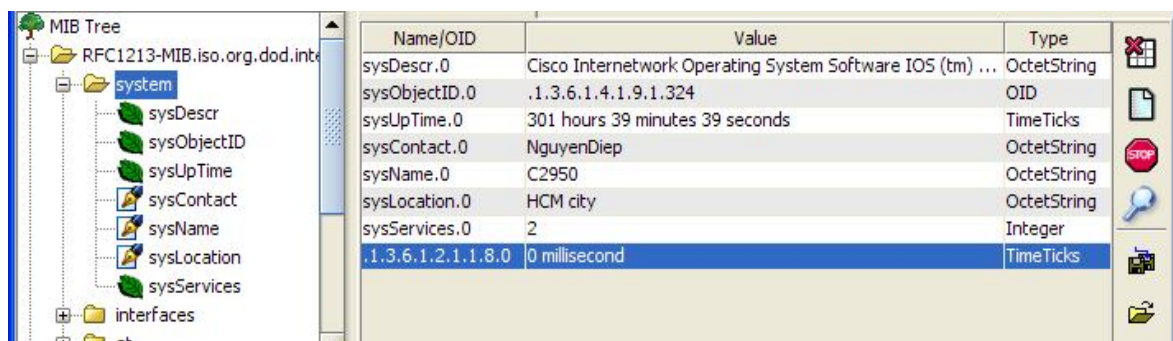
Như vậy đến đây bạn đã biết chương trình giám sát lưu lượng bằng SNMP hoạt động như thế nào để lấy được lưu lượng của các interface.

### Các operation khác của iReasoning MIB Browser

Ngoài các operation tương ứng các phương thức của SNMPv1, iReasoning MIB Browser còn có một số operation khác.

**GetBulk** : thực hiện phương thức GetBulkRequest. GetBulk là phương thức có từ SNMPv2, nó có thể lấy thông tin của nhiều object cùng lúc chỉ bằng 1 bản tin; ví dụ thay vì bạn gửi 10 GetRequest thì bạn chỉ cần gửi 1 GetBulk. Chúng ta sẽ thực hiện GetBulk sau khi tìm hiểu về SNMPv2.

**GetSubTree** : thực hiện lấy tất cả các object nằm dưới một node. Ví dụ bạn đang ở node mib-2.system, khi dùng GetSubTree thì phần mềm sẽ lấy tất cả các OID nằm dưới một cấp. Lưu ý GetSubTree không phải là một phương thức chuẩn của SNMP mà là tính năng của riêng phần mềm, bản chất của việc thực hiện GetSubTree là thực hiện các GetNextRequest liên tục đến khi nào nhận được object cùng cấp với object khởi đầu.



Name/OID	Value	Type
sysDescr.0	Cisco Internetwork Operating System Software IOS (tm) ...	OctetString
sysObjectID.0	.1.3.6.1.4.1.9.1.324	OID
sysUpTime.0	301 hours 39 minutes 39 seconds	TimeTicks
sysContact.0	NguyenDiep	OctetString
sysName.0	C2950	OctetString
sysLocation.0	HCM city	OctetString
sysServices.0	2	Integer
.1.3.6.1.2.1.1.8.0	0 millisecond	TimeTicks

**Walk** : thực hiện GetNextRequest liên tục đến khi hết bảng mib. Chỉ cần một lần Walk bạn có thể lấy toàn bộ các object nhưng sẽ tốn nhiều thời gian. Walk cũng không phải là phương thức chuẩn của SNMP.

## 4. SMIV2

SMIV2 (Structure of Management Information version 2) được trình bày trong RFC2578, bao gồm nhiều thay đổi trong cấu trúc mib file. Phần này trình bày những thay đổi chủ yếu nhất.

### Các kiểu dữ liệu mới hoặc thay đổi so với SMIV1

- + INTEGER32 : số nguyên nằm trong khoảng  $-2^{31}$  và  $2^{31}-1$  (-2147483648 to 2147483647 decimal).
- + OCTET STRING : kiểu chuỗi ký tự, độ dài tối đa 65535.
- + OBJECT IDENTIFIER : định danh của object, không quá 128 phần tử (sub-identifier), mỗi phần tử là số nguyên không quá  $2^{32}-1$ .
- + COUNTER32 : kiểu số nguyên không âm tăng dần, tối đa là  $2^{32}-1$ , khi vượt giá trị tối đa thì quay lại từ 0. Counter32 không bắt buộc giá trị bắt đầu phải là 0.
- + GAUGE32 : kiểu số nguyên không âm tăng hoặc giảm, giới hạn trong khoảng  $0 \sim 2^{32}-1$ , nó không thể vượt ra giới hạn này.
- + COUNTER64 : kiểu số nguyên không âm tăng dần, tối đa là  $2^{64}-1$  (18446744073709551615).

+ UNSIGNED32 : kiểu số nguyên từ 0 ~  $2^{32}-1$ .

### Kiểu dữ liệu OBJECT-TYPE

Trong SMIV1 kiểu OBJECT-TYPE bao gồm : SYNTAX, ACCESS, STATUS, DESCRIPTION. Trong SMIV2 kiểu OBJECT-TYPE bao gồm các trường : SYNTAX, UNITS, MAX-ACCESS, STATUS, DESCRIPTION, REFERENCE, INDEX, AUGMENTS, DEFVAL.

+ SYNTAX : kiểu dữ liệu của object, là một kiểu theo chuẩn ASN.1 hoặc các kiểu định nghĩa riêng của SMIV2.

+ UNITS : là dòng text mô tả một unit nào đó gắn liền với object, trường này không bắt buộc phải có.

+ MAX\_ACCESS : có 5 quyền truy xuất object có ưu tiên từ thấp đến cao là "not-accessible", "accessible-for-notify", "read-only", "read-write", "read-create"; MAX\_ACCESS quy định quyền cao nhất tác động đến object, quyền cao hơn bao gồm các quyền thấp hơn. VD object có MAX\_ACCESS là "read-write" thì có thể được đọc/ghi nhưng không thể tạo.

+ STATUS : trạng thái của object, mang một trong các giá trị "current" (định nghĩa của object đang có hiệu lực và đang được sử dụng), "obsolete" (định nghĩa này đã cũ và có thể bỏ đi), "deprecated" (định nghĩa này đã cũ và các chuẩn tiếp theo có thể định nghĩa lại).

+ DESCRIPTION : dòng text mô tả thông tin ý nghĩa của object.

+ REFERENCE : là dòng text mô tả đến các tài liệu khác có liên quan đến object này, reference không bắt buộc phải có.

+ INDEX : chỉ ra trường index của object hiện tại. VD ifDescr có INDEX = ifIndex.

+ AUGMENTS : tương tự như INDEX và có thể dùng thay thế INDEX, nhưng chỉ một trong 2 trường INDEX hoặc AUGMENTS tồn tại, không thể tồn tại cùng lúc cả 2.

+ DEFVAL : giá trị mặc định (default value) của object khi nó được tạo ra.

### Kiểu dữ liệu NOTIFICATION-TYPE

Kiểu NOTIFICATION-TYPE được dùng để mô tả những thông tin quản lý mạng được truyền không theo yêu cầu (ví dụ bản tin TrapPDU hoặc InformRequestPDU của SNMPv2, chúng được tự động gửi đi khi có sự kiện xảy ra mà không cần phải có request từ thiết bị khác).

Các notification phải được định nghĩa trong mib, cấu trúc của chúng bao gồm các mệnh đề sau :

+ OBJECT : danh sách có thứ tự các object có liên quan đến notification, vd bản tin notification cho 4 interface của thiết bị thì OBJECT phải chứa ifIndex của 4 interface đó.

+ STATUS : mang một trong 3 giá trị "current", "obsolete" hoặc "deprecated".

+ DESCRIPTION : dòng text mô tả ý nghĩa của notification.

+ REFERENCE : mô tả các tài liệu có liên quan đến định nghĩa của notification, REFERENCE không bắt buộc phải có.

Ví dụ : khi cấu hình của thiết bị thay đổi thì nó sẽ gửi bản tin entConfigChange có kiểu NOTIFICATION-TYPE, được định nghĩa như sau :

```
entConfigChange NOTIFICATION-TYPE
  STATUS current
  DESCRIPTION
    "An entConfigChange trap is sent when the value of entLastChangeTime
    changes. It can be utilized by an NMS to trigger logical/physical entity
    table maintenance polls. An agent must not generate more than one
    entConfigChange 'trap-event' in a five second period, where a 'trap-
    event' is the transmission of a single trap PDU to a list of trap
    destinations. If additional configuration changes occur within the five
    second 'throttling' period, then these trap-events should be suppressed
    by the agent. An NMS should periodically check the value of
    entLastChangeTime to detect any missed entConfigChange trap-events, e.g.
    due to throttling or transmission loss."
  ::= { entityMIBTrapPrefix 1 }
```



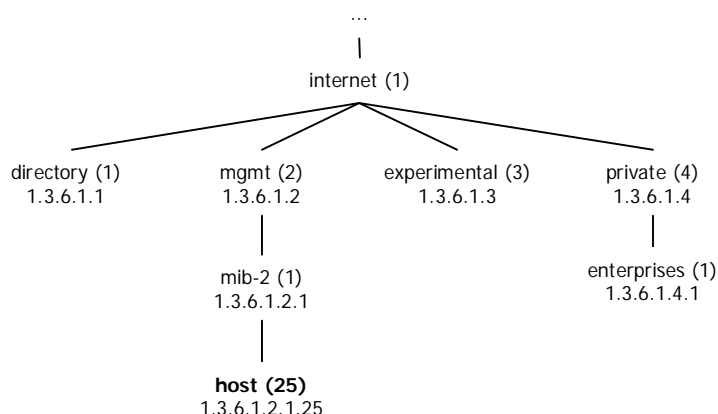
## 5. Host-Resources-Mib (RFC2790)

RFC2790 là mib dùng cho host, nó cung cấp định nghĩa nhiều object như thông tin hệ thống, lưu trữ, device, software, performance. Dịch vụ SNMP agent trên Windows và Linux đều hỗ trợ RFC2790.

Vị trí của Host-mib trong mib như sau :

host      OBJECT IDENTIFIER ::= { mib-2 25 }

Tức là .iso.org.dod.internet.mgmt.mib-2.host hay .1.3.6.1.2.1.25.



Các kiểu dữ liệu mới được định nghĩa trong host-mib gồm :

- + Kbytes : kiểu INTEGER32, thể hiện kích thước của thiết bị lưu trữ, đơn vị tính là 1024 Bytes.
- + ProductID : xác định nhà sản xuất, model, phiên bản của phần cứng hay phần mềm.
- + AutonomousType : kiểu giá trị định danh có thể mở rộng độc lập, ví dụ nó có thể chỉ ra một cây mib con nào đó được định nghĩa bởi một tài liệu khác.
- + DateAndTime : kiểu ngày và giờ, định dạng như sau :  
 "year-mon-day,hour:min:sec.centisec,±HourFromUTC:MinFromUTC". Ví dụ "15/01/2010 1:30:15 PM, GMT+7" được biểu diễn là "2010-01-15,13:30:15.0,+7:0"

Mục này trích định nghĩa một số object quan trọng trong công tác giám sát server như tình trạng hệ thống, tình trạng lưu trữ (ổ cứng), tỷ lệ sử dụng cpu & bộ nhớ của từng process, danh sách các phần mềm cài đặt; do chúng ta sẽ thực hành dùng các công cụ đọc mib để đọc các object này. Để có được mô tả đầy đủ các bạn hãy tham khảo RFC2790.

Name	Syntax	Description
host-mib(25)		Host-Resources-Mib (RFC2790) OID : .1.3.6.1.2.1.25
hrSystem(1)		
hrSystemUptime(1)	TimeTicks	Thời gian trôi qua kể từ khi host khởi động, tính bằng phần trăm giây
hrSystemDate(2)	DateAndTime	Ngày giờ local của host
hrSystemInitialLoadDevice(3)	Integer32	Index của thiết bị mà host được cấu hình để load hệ điều hành (danh sách thiết bị và index nằm trong hrDeviceEntry)
hrSystemInitialLoadParameters(4)	DisplayString	Chuỗi tham số cung cấp cho thiết bị chứa HĐH khi khởi động
hrSystemNumUsers(5)	Gauge32	Số lượng session của các user đang log vào host
hrSystemProcesses(6)	Gauge32	Số lượng process đang chạy trên host

		hrSystemMaxProcesses(7)	Integer32	Số lượng process tối đa mà hệ thống có thể hỗ trợ, nếu hệ thống không giới hạn số process thì object này = 0
		hrStorage(2)		
		hrStorageTypes(1)		(chưa được định nghĩa)
		hrMemorySize(2)	KBytes	Dung lượng bộ nhớ vật lý chính của host, thường là RAM
		hrStorageTable(3)	Sequence	Bảng chứa các vùng lưu trữ logic trên host
		hrStorageEntry(1)	hrStorageEntry	
		hrStorageIndex(1)	Integer32	Giá trị duy nhất của mỗi thiết bị lưu trữ vật lý trên host
		hrStorageType(2)	AutonomousType	Kiểu thiết bị lưu trữ
		hrStorageDescr(3)	DisplayString	Mô tả thiết bị lưu trữ
		hrStorageAllocationUnits(4)	Integer32	Kích thước tính bằng byte của một đơn vị lưu trữ dữ liệu, vd trên windows ntfs mặc định mỗi block là 4KB
		hrStorageSize(5)	Integer32	Kích thước của thiết bị lưu trữ, đơn vị tính là hrStorageAllocationUnits
		hrStorageUsed(6)	Integer32	Dung lượng đã sử dụng, tính bằng hrStorageAllocationUnits
		hrStorageAllocationFailures(7)	Counter32	Số lần các yêu cầu lưu trữ bị từ chối do không còn dung lượng trống
		hrDevice(3)		
		hrProcessorTable(3)		Bảng chứa thông tin về bộ xử lý của host
		hrProcessorEntry(1)		
		hrProcessorFrwID(1)	ProductID	Mã sản phẩm của bộ xử lý
		hrProcessorLoad(2)	Integer	Tỷ lệ phần trăm thời gian rỗi của cpu, tính trung bình trong 1 phút vừa qua
		hrSWRunPerf(5)		
		hrSWRunPerfTable(1)		Bảng chứa các kết quả đo hiệu suất chiếm dụng cpu & ram của từng process
		hrSWRunPerfEntry(1)		
		hrSWRunPerfCPU(1)	Integer32	Thời gian chiếm dụng cpu của process, tính bằng centi-second
		hrSWRunPerfMem(2)	KBytes	Lượng bộ nhớ được cấp phát cho process
		hrSWInstalled(6)		
		hrSWInstalledLastChange(1)	TimeTicks	Giá trị của sysUpTime từ khi có một phần mềm nào đó được cài, thay đổi hay gỡ bỏ
		hrSWInstalledLastUpdateTime(2)	TimeTicks	Giá trị của sysUpTime từ khi hrSWInstalledTable được cập nhật hoàn toàn lần gần nhất
		hrSWInstalledTable(3)	Sequence	Danh sách các phần mềm được cài trên host
		hrSWInstalledEntry(1)		
		hrSWInstalledIndex(1)	Integer32	Giá trị duy nhất của từng phần mềm trên host
		hrSWInstalledName(2)	DisplayString	Tên của từng phần mềm trên host, có thể bao gồm cả tên nhà sản xuất, version.
		hrSWInstalledID(3)	ProductID	Mã sản phẩm của phần mềm



				hrSWInstalledType(4)	Integer	Kiểu phần mềm, gồm : unknown(1), operatingSystem(2), deviceDriver(3), application(4)
				hrSWInstalledDate(5)	DateAndTime	Thời điểm thay đổi gần nhất trong cây thư mục của phần mềm. Nếu host không biết thông tin này thì object này phải mang giá trị 0000-01-01,0:0:0,+0:0

## 6. Sử dụng net-snmp-utils trên Linux

Phần trên bạn đã sử dụng một công cụ mib browser trên Windows, phần này chúng ta sẽ thực hành cách lấy thủ công các thông tin của máy chủ, bằng cách dùng các lệnh command-line trong bộ công cụ net-snmp-utils để làm việc với host-resources-mib. Hệ điều hành được chọn minh họa vẫn là CentOS 5, tuy nhiên snmp agent trên CentOS 5 lại không hỗ trợ đầy đủ mib này nên chúng ta sẽ dùng các lệnh trên CentOS để giám sát một server Windows.

Để dùng được các lệnh snmp dạng command line trên CentOS, bạn phải cài đặt package net-snmp-utils. Chương 2 đã có phần hướng dẫn cách cài và khởi động package net-snmp-utils trên CentOS và cách cấu hình dịch vụ SNMP trên Windows.

Các ví dụ trong phần này sẽ thực hiện với snmp version = 1, read-community = public, write-community = private, host = 192.168.1.100 (IP của server Windows cần giám sát).

### GetRequest

Cú pháp :

```
% snmpget -v version -c read-community host object-id
```

Trong đó :

- + *version* : phiên bản snmp, bao gồm : 1, 2c, 3.
- + *read-community* : chuỗi community dùng để lấy thông tin.
- + *host* : domainname hoặc IP của thiết bị cần quản lý.
- + *object-id* : định danh của object cần lấy.

**Ví dụ :** lấy thời gian trôi qua kể từ lúc khởi động máy chủ.

+ OID cần lấy là host-resources-mib.hrSystemUptime.0 (.1.3.6.1.2.1.25.1.1.0)

```
# snmpget -v 1 -c public 192.168.1.100 .1.3.6.1.2.1.25.1.1.0
HOST-RESOURCES-MIB::hrSystemUptime.0 = Timeticks: (16363328) 1 day, 21:27:13.28
```

+ Giá trị lấy được là 16363328, căn cứ vào định nghĩa của host-resources-mib thì hrSysUptime được tính bằng phần trăm giây, do đó CentOS hiển thị luôn giá trị đã được chuyển đổi thành giờ:phút:giây.miligiây.

Trong các câu lệnh snmp, ngoài việc dùng object-id dạng số thì ta cũng có thể nhập dạng chữ, vd :

```
# snmpget -v 1 -c public 192.168.1.100 HOST-RESOURCES-MIB::hrSystemUptime.0
HOST-RESOURCES-MIB::hrSystemUptime.0 = Timeticks: (16363328) 1 day, 21:27:13.28
```

### GetNextRequest

Cú pháp :

```
% snmpgetnext -v version -c read-community host object-id
```

**Ví dụ :** lấy tỷ lệ chiếm dụng CPU.

+ OID cần lấy là host-resources-mib.hrDevice.hrProcessorTable.hrProcessorEntry.hrProcessorLoad (.1.3.6.1.2.1.25.3.3.1.2). Một host có thể có nhiều processor nên thông tin của mỗi processor là một entry trong bảng hrProcessorTable, nếu muốn lấy thông tin của processor x thì OID cần lấy phải có dạng

hrProcessorLoad.x (.1.3.6.1.2.1.25.3.3.1.2.x) với x là index của processor đó. Các OID hrProcessorLoad.x nằm sau hrProcessorLoad, tuy nhiên ta chưa biết số x này bắt đầu từ mấy nên ta phải thực hiện GetNext trên OID hrProcessorLoad để lấy được tài của cpu thứ nhất. Sau đó thực hiện GetNext trên OID của cpu thứ nhất thì ta sẽ lấy được tài của cpu thứ hai.

```
# snmpgetnext -v 1 -c public 192.168.1.100 .1.3.6.1.2.1.25.3.3.1.2
HOST-RESOURCES-MIB::hrProcessorLoad.3 = INTEGER: 3
# snmpgetnext -v 1 -c public 192.168.1.100 .1.3.6.1.2.1.25.3.3.1.2.3
HOST-RESOURCES-MIB::hrProcessorLoad.4 = INTEGER: 3
```

+ Ta thấy sau khi GetNext OID hrProcessorLoad thì ta được OID hrProcessorLoad.3 với giá trị kiểu INTEGER là 3. Nghĩa là cpu thứ nhất có id là 3 trong agent của host đang được giám sát, và tỷ lệ chiếm dụng cpu trung bình trong 1 phút vừa qua là 3%. CPU thứ hai có id là 4 và tỷ lệ chiếm dụng cũng là 3%.

+ Bằng cách lấy giá trị hrProcessorLoad đều đặn, phần mềm giám sát SNMP có thể vẽ được biểu đồ hoạt động của cpu trên host được giám sát. Lưu ý hrProcessorLoad được agent cập nhật mỗi phút một lần.

### SetRequest

Cú pháp :

```
% snmpset -v version -c write-community host object-id type value
```

Trong đó :

- + *version* : phiên bản snmp, bao gồm : 1, 2c, 3.
- + *write-community* : chuỗi community dùng để thiết lập thông tin.
- + *host* : domainname hoặc IP của thiết bị cần quản lý.
- + *object-id* : định danh của object cần set.
- + *type* : kiểu của dữ liệu sẽ được set, là một trong các giá trị i, u, t, a, o, s, x, d, b, U, I, F, D  
(i: Integer, u: Unsigned Integer; t: TimeTicks; a: IPAddress; o: Object Identifier; s: String;  
x: Hex String; d: Decimal String; b: Bits; U: Unsigned Int64; I: Signed Int64; F: Float; D:  
Double)
- + *value* : giá trị được set.

**Ví dụ :** Đặt lại tên máy tính.

+ Tên của host không thuộc host-resources-mib mà thuộc về mib-2, OID là mib-2.system.sysName. Do host chỉ có một tên nên tên của host là sysName.0 (.1.3.6.1.2.1.1.5.0). sysName có kiểu là OctetString nên tham số *type* trong lệnh snmpset sẽ là "s". Ta thực hiện lấy tên trước và sau khi set để thấy sự thay đổi.

```
# snmpget -v 1 -c public 192.168.1.100 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: NguyenDiepPC
#
# snmpset -v 1 -c private 192.168.1.100 .1.3.6.1.2.1.1.5.0 s NewComputer
SNMPv2-MIB::sysName.0 = STRING: NewComputer
#
# snmpget -v 1 -c public 192.168.1.100 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: NewComputer
```

### Trap

Câu lệnh gửi trap so với các lệnh khác thì phức tạp hơn do có nhiều tham số. Trong chương 1 có trình bày cấu trúc của bản tin trap SNMPv1, bao gồm các trường enterpriseId, agent, genericTrap, specificTrap, timeStamp và variableBindings; lệnh gửi trap cũng phải chỉ ra đầy đủ các trường này.

Cú pháp :

```
% snmptrap -v version -c trap-community host enterprise agent generic specific
timestamp object-id type value
```

Trong đó :

- + *version* : phiên bản snmp, bao gồm : 1, 2c, 3.

- + *trap-community* : chuỗi community dùng để gửi trap.
- + *host* : domainname hoặc IP của máy tính nhận trap.
- + *enterprise* : id của doanh nghiệp, dùng để nhận dạng thiết bị gửi trap là của hãng nào, model gì.
- + *agent* : domainname hoặc IP của thiết bị cần quản lý, tham số agent này có thể không cần chỉ ra trong dòng lệnh để hệ thống tự động chèn IP vào.
- + *generic* : loại trap generic (từ 0 đến 6), nếu là trap riêng thì generic = 6 (enterprise-specific trap).
- + *specific* : loại trap specific.
- + *timestamp* : thời điểm xảy ra sự kiện làm cho hệ thống gửi trap, thông thường hệ thống gửi trap ngay khi sự kiện xảy ra nên timestamp bằng với thời điểm gửi trap; tham số này cũng có thể không cần chỉ ra.
- + *object-id* : định danh của object có liên quan đến sự kiện.
- + *type* : kiểu giá trị của object.
- + *value* : giá trị hiện tại của object.

**Ví dụ :** thực hiện gửi trap thông báo rằng tên của hệ thống đã bị thay đổi.

+ Bạn hãy dùng phần mềm Trap Viewer trong bộ phần mềm Solarwinds để xem trap. Ta sẽ giả lập gửi một trap từ một switch Cisco Catalyst 2950 thông báo rằng hostname của nó đã bị thay đổi thành "Noname", mặc dù nơi gửi trap là máy tính chạy CentOS.

+ Cisco có enterprise number là 9, và dòng sản phẩm C2950 có id là .1.324 nên *enterprise* = .iso.org.dod.internet.private.enterprise.Cisco.1.324. Trap này là loại tự định nghĩa nên *generic* = 6, và ta cho *specific* là bất kỳ. *object-id* là sysName.0 vì sự kiện này liên quan đến tên của hệ thống. Còn *value* chính là tên mới của hệ thống, kiểu của sysName là OctetString nên *type* = s.

```
# snmptrap -v 1 -c public 192.168.1.100 .1.3.6.1.4.1.9.1.324 "" 6 10 "" sysName.0 s "Noname"
```

+ Quan sát trên Trap Viewer bạn sẽ thấy một trap hiện ra.



## Walk

Walk không phải là một phương thức trong SNMP (không có bản tin walk) mà là tiện ích của net-snmp-utils. Walk cho phép lấy tất cả object là con cháu nằm dưới một object nào đó nhưng không lấy hết bảng mib. Trong khi đó công cụ iReasoning Mib Browser khi thực hiện Walk thì nó sẽ lấy tất cả object từ đó về sau cho đến hết bảng mib.

Cú pháp :

```
% snmpwalk -v version -c read-community host root-object-id
```

Trong đó :

- + *version* : phiên bản snmp, bao gồm : 1, 2c, 3.
- + *read-community* : chuỗi community dùng để lấy thông tin.
- + *host* : domainname hoặc IP của thiết bị cần quản lý.
- + *root-object-id* : định danh của object được lấy làm gốc.

**Ví dụ :** Lấy tất cả thông tin nằm dưới mib-2.system.

```
# snmpwalk -v 1 -c public 192.168.1.100 system
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 107 Stepping 2
AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Multiprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.2
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (2366393) 6:34:23.93
SNMPv2-MIB::sysContact.0 = STRING: NguyenDiep
SNMPv2-MIB::sysName.0 = STRING: NewComputer
SNMPv2-MIB::sysLocation.0 = STRING: HCM
SNMPv2-MIB::sysServices.0 = INTEGER: 79
```

### Tóm tắt

- + MIB là một cấu trúc dữ liệu được thiết kế để quản lý các thiết bị, là một tập hợp định danh đối tượng được quản lý.
- + SMI là chuẩn quy định các cấu trúc dữ liệu và thuộc tính của từng đối tượng được quản lý.
- + Theo SMIV1, một đối tượng được quản lý có kiểu OBJECT-TYPE bao gồm các thuộc tính : SYNTAX (kiểu dữ liệu), ACCESS (quyền truy xuất), STATUS (bắt buộc, tùy chọn), DESCRIPTION (mô tả ý nghĩa). Trong SMIV2 thì kiểu OBJECT-TYPE bao gồm các thuộc tính : SYNTAX, MAX-ACCESS, UNITS, STATUS, DESCRIPTION, REFERENCE, INDEX, AUGMENT, DEFVAL.
- + mib-2 là một chuẩn mib của các thiết bị chạy trên nền TCP/IP.
- + host-resources-mib là chuẩn mib dùng cho host.
- + Để thực hiện các phương thức SNMP bạn có thể dùng công cụ iReasoning Mib Browser trên Windows hoặc net-snmp-utils trên Linux.