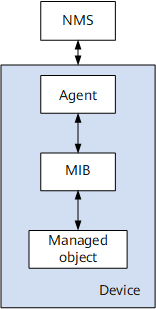
**简介**

**定义**

SNMP（Simple Network Management Protocol）是一种网络管理协议，用于管理和监控网络设备。它是一种应用层协议，基于UDP协议进行传输，通常使用UDP端口号161/162。SNMP提供了一种统一的接口，用于管理和监控网络设备，使得网络管理员可以方便地管理网络性能、故障排除和安全控制等方面。

**组成**

SNMP基本组件包括网络管理系统NMS（Network Management System）、代理进程（Agent）、被管对象（Managed Object）和管理信息库MIB（Management Information Base）。如图所示他们共同构成SNMP的管理模型，在SNMP的体系结构中都起着至关重要的作用。



**NMS**

NMS在网络中扮演管理者角色，是一个采用SNMP协议对网络设备进行管理/监视的系统，运行在NMS服务器上。

NMS可以向设备上的Agent发出请求，查询或修改一个或多个具体的参数值。

NMS可以接收设备上的Agent主动发送的Trap信息，以获知被管理设备当前的状态。

**Agent**

Agent是被管理设备中的一个代理进程，用于维护被管理设备的信息数据并响应来自NMS的请求，把管理数据汇报给发送请求的NMS。

Agent接收到NMS的请求信息后，通过MIB表完成相应指令后，并把操作结果响应给NMS。

当设备发生故障或者其它事件时，设备会通过Agent主动发送信息给NMS，向NMS报告设备当前的状态变化。

**Managed Object**

Managed Object指被管理对象。每一个设备可能包含多个被管理对象，被管理对象可以是设备中的某个硬件，也可以是在硬件、软件（如路由选择协议）上配置的参数集合。

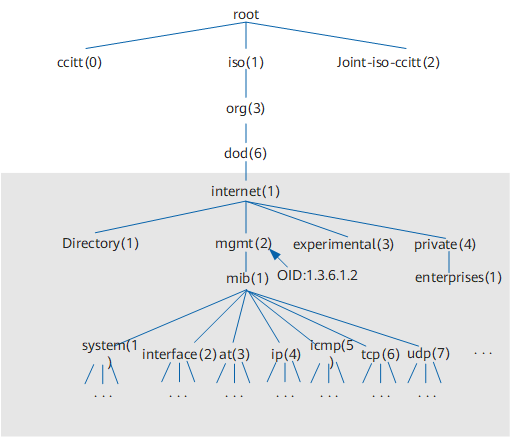
**MIB**

MIB是一个数据库，指明了被管理设备所维护的变量，是能够被Agent查询和设置的信息。MIB在数据库中定义了被管理设备的一系列属性：对象的名称、对象的状态、对象的访问权限和对象的数据类型等。通过MIB，可以完成以下功能：

Agent通过查询MIB，可以获知设备当前的状态信息。

Agent通过修改MIB，可以设置设备的状态参数。

SNMP的MIB采用树型结构，它的根在最上面，根没有名字。如下图所示的是MIB的一部分，它又称为对象命名树。每个对象标识符OID（object identifier）对应于树中的一个管理对象，该树的每个分支都有一个数字和一个名称，并且每个点都以从该树的顶部到该点的完整路径命名，如system的OID为1.3.6.1.2.1.1，interfaces的OID为1.3.6.1.2.1.2。



*OID树结构*

**版本/标准**

SNMP有三种版本：SNMPv1，SNMPv2c和SNMPv3。

SNMPv1：SNMP的第一个版本，它提供了一种监控和管理计算机网络的系统方法，它基于团体名认证，安全性较差，且返回报文的错误码也较少。它在RFC 1155和RFC 1157中定义。

SNMPv2c：第二个版本SNMPv2c引入了GetBulk和Inform操作，支持更多的标准错误码信息，支持更多的数据类型。它在RFC 1901，RFC 1905和RFC 1906中定义。RFC1910引入基于用户的安全模型，又称作SNMPv2u。

SNMPv3：鉴于SNMPv2c在安全性方面没有得到改善，IETF颁布了SNMPv3版本，提供了基于USM（User Security Module）的认证加密和基于VACM（View-based Access Control Model）的访问控制，是迄今为止最安全的版本。SNMPv3在RFC 1905，RFC 1906，RFC 2571，RFC 2572，RFC 2574和RFC 2575中定义。

**能力**

SNMP的工作原理是将协议数据单元（也称为SNMP GET请求）发送到响应SNMP的网络设备。用户通过网络监控工具可以跟踪所有通信过程，并从SNMP获取数据。

SNMP规定了几个操作类型来完成各组件之间的信息交换，如下表所示：

|  |  |  |
| --- | --- | --- |
| 操作类型 | 描述 | 备注 |
| Get | Get操作可以从Agent中提取一个或多个参数值。 | - |
| GetNext | GetNext操作可以从Agent中按照字典序提取下一个参数值。 | - |
| Set | Set操作可以设置Agent的一个或多个参数值。 | - |
| Response | Response操作可以返回一个或多个参数值。这个操作是由Agent发出的，它是GetRequest、GetNextRequest、SetRequest和GetBulkRequest四种操作的响应操作。Agent接收到来自NMS的Get/Set指令后，通过MIB完成相应的查询/修改操作，然后利用Response操作将信息回应给NMS。 | - |
| Trap | Trap信息是Agent主动向NMS发出的信息，告知管理进程设备端出现的情况。 | - |
| GetBulk | GetBulk操作实现了NMS对被管理设备的信息群查询。 | SNMPv1版本不支持GetBulk操作 |
| Inform | InformRequest也是被管理设备向NMS主动发送告警。与Trap告警不同的是，被管理设备发送Inform告警后，需要NMS回复InformResponse来进行确认。 | SNMPv1版本不支持Inform操作 |

**安全模型**

auth(community+md5+sha)+privacy(none+des+aes)

**Ireasoning MIB Browser简介**

iReasoning MIB Browser是一款由iReasoning SNMP API提供支持的强大且易于使用的工具，它是工程师管理启用SNMP的网络设备和应用程序不可或缺的工具。它允许用户加载标准的、专有的MIB，甚至一些错误的MIB。它还允许用户发出SNMP请求以检索代理的数据，或对代理进行更改。内置陷阱接收器可以根据其规则引擎接收和处理SNMP陷阱。iReasoning MIB Browser提供了直观的GUI界面，支持多种平台，包括Windows、Linux、macOS等.

使用Ireasoning MIB Browser可以直观的查看设备等SNMP agent的MIB信息，及接收trap通告。

**要求**

PPT汇报；掌握SNMP基础知识，会使用工具进行MIB操作

**使用/运维部署(netsnmp)**

**netsnmp简介**

NET-SNMP是一种开放源代码的[简单网络管理协议](https://zh.wikipedia.org/wiki/%E7%AE%80%E5%8D%95%E7%BD%91%E7%BB%9C%E7%AE%A1%E7%90%86%E5%8D%8F%E8%AE%AE)（Simple Network Management Protocol）软件。NET-SNMP支持SNMP v1, SNMP v2c 与 SNMP v3，并可以使用 IPV4 及 IPV6 。基于(BSD及BSD like)许可。

**netsnmp安装和配置**

您可以按照以下步骤在Linux系统上安装和配置Net-SNMP：

确保您拥有root权限的Linux系统，并且该系统有一个活跃的互联网连接，以便下载必要的软件包。

使用您的发行版的软件包管理器安装Net-SNMP。例如，如果您使用的是Ubuntu，您可以使用以下命令安装Net-SNMP：

sudo apt-get update

sudo apt-get install snmpd

安装完成后，您需要根据您的要求对它进行配置。SNMP守护进程的配置文件位于**/etc/snmp/snmpd.conf**。您可以使用文本编辑器打开配置文件，并根据需要进行修改。例如，您可以更改SNMP请求所使用的团体字，定义不同SNMP组的访问权限等等。

保存对配置文件的修改，并使用以下命令重启SNMP守护进程：

sudo systemctl restart snmpd

为了测试Net-SNMP的安装，您可以使用SNMP命令行工具。最常用的工具是snmpwalk

snmpwalk -v2c -c public localhost system

上述命令将使用SNMP版本2c和“public”社区字符串检索本地系统的系统描述。也可以使用上节介绍的Ireasoning MIB Browser工具进行验证

扩展测试更多的SNMP配置，包括基本的侦听/agentx配置，V1 V2C V3团体字/用户配置，视图配置，组策略配置等

**要求**

会搭建SNMP服务器，并掌握服务器的各种配置

**业务MIB开发**

**开发工具安装**

为了顺利执行mib2c，基于源码编译安装netsnmp，可以到[官网](http://www.net-snmp.org/)下载最新源码包，解压后进入源码根目录执行下述命令进行安装；注意安装位置默认为/usr/local/下，为了避免冲突，建议将apt工具安装的snmpd删除

apt remove snmpd

./configure && make && make install

**自动代码生成**

通过ls /usr/share/snmp/查看代码生成模板

创建MIB库目录，此处使用 /root/projects/mib

修改snmp的MIB搜索目录，mkdir -p ~/.snmp && echo -e "mibdirs +/root/projects/mib\nmibs +ALL" > ~/.snmp/snmp.conf

将需要编译的MIB文件放在MIB库中；此处为/root/projects/mib/[POWER-ETHERNET-MIB.txt](https://raw.githubusercontent.com/librenms/librenms/master/mibs/POWER-ETHERNET-MIB)

执行代码生成命令，对于trap需要单独生成

mib2c -c mib2c.old-api.conf {MODULE-IDENTITY} // powerEthernetMIB

mib2c -c mib2c.notify.conf {OBJECT IDENTIFIER} // pethNotifications

**测试数据**

port\_t tbl[] = {{.grpid = 1, .portid = 1, .admin\_enable = 2, .type = "type1"}, {.grpid = 1, .portid = 2, .admin\_enable = 2, .type = "type2"}};

**代码修改**

使用测试数据填充数据，并按照要求实现功能；通过main函数或net-snmp-config对自动生成代码进行初始化和调度

典型参数说明：

unsigned char \*var\_pethPsePortTable(struct variable \*vp, oid \*name, size\_t \*length, int exact, size\_t \*var\_len, WriteMethod \*\*write\_method)

vp: 当前MIB树信息，其中name/namelen不含表索引

name: 完整OID名，含表索引

length: name的长度

exact: 1表示get确定节点，0表示getn获取下一个

var\_len: 返回变量长度

write\_method: 写方法

返回值: 变量地址

int write\_pethPsePortAdminEnable(int action, u\_char \*var\_val, u\_char var\_val\_type, size\_t var\_val\_len, u\_char \*statP, oid \*name, size\_t name\_len)

action: 操作类型

var\_val: 写入值

var\_val\_type: 写入值类型

var\_val\_len: 写入值长度

statP: ??

name: 完整OID名称，含表索引

name\_len: 完整OID长度

返回值: 成功/失败

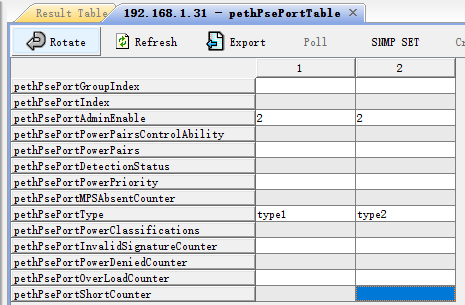
**编译参考**

net-snmp-config --compile-subagent a.out data.c pethNotifications.c powerEthernetMIB.c  && ./a.out -f -Lo -x  tcp:localhost:705

**测试要求**

能够通过MIB Browser读到测试数据，能够通过修改pethPsePortAdminEnable触发pethPsePortOnOffNotification通告，通告内容包括端口信息和状态；**注意实际开发时区分管理状态和运行状态**

**参考**





**MIB文件开发**

**SMIv1 vs. SMIv2**

SMIv1和SMIv2是用于描述和管理网络设备的不同语法。SMIv1是简单网络管理协议（SNMP）的最初版本，而SMIv2是SNMP的第二个版本，也被称为SNMPv2c。
SMIv1是基于C语言的，它提供了一种简单且易于理解的方式来描述网络设备。它使用类型/长度/值（TLV）的格式来编码消息，其中类型字段指示消息的类型，长度字段指示消息的长度，值字段包含实际的消息内容。
相比之下，SMIv2是基于ASN.1（抽象语法标记语言）的，它提供了一种更加灵活和强大的方式来描述和管理网络设备。SMIv2使用ASN.1定义管理信息结构，并使用DER（Distinguished Encoding Rules）格式对ASN.1结构进行编码。DER是一种更加紧凑和安全的编码方式，它可以提供更好的性能和安全性。
除了语法上的差异之外，SMIv2还提供了一些新的功能，例如支持更多的数据类型和更复杂的管理信息结构。此外，SMIv2还提供了一些安全性和隐私保护功能，例如对管理信息进行加密和身份验证。
总的来说，SMIv1和SMIv2是两种不同的语法，用于描述和管理网络设备。SMIv2提供了更多的功能和更好的性能和安全性。然而，由于SMIv1的简单性和易于理解性，它仍然被广泛使用。

**ASN.1语法**

ASN.1 (Abstract Syntax Notation One) 是一种用于定义数据结构的标准化语言，广泛应用于通信协议和系统之间的信息交换。使用 ASN.1，可以将复杂的数据结构定义为具有明确、唯一性的符号表示，使得不同的系统和设备之间可以正确地解析和处理这些数据。

在运维和部署过程中，ASN.1 的使用可以大大提高通信协议的可靠性和效率。例如，在路由器配置中，ASN.1 可以用于定义 IP 地址、子网掩码、默认网关等网络配置信息。通过 ASN.1，可以将这些配置信息以统一、标准化的方式进行表示，避免了不同设备之间配置的不一致性。

此外，ASN.1 还广泛应用于定义协议消息的结构。例如，在 TCP/IP 协议栈中，ASN.1 可以用于定义 HTTP 请求和响应的消息结构。通过 ASN.1，可以清晰地定义消息的各个组成部分，以及它们在二进制数据中的位置和长度。这有助于开发人员准确地解析和处理协议消息，从而提高系统的稳定性和性能。

总之，使用 ASN.1 可以提高通信协议的可靠性和效率，降低系统维护和部署的复杂性。在运维和部署过程中，了解 ASN.1 的基本概念和使用方法是非常重要的。

**MG-SOFT Visual MIB Builder安装和使用**

MG-SOFT Visual MIB Builder是一款由MG-SOFT公司提供的可视化MIB模块定义文件创建、设计、编辑和验证工具。它提供了一个易于使用的拖放用户界面，让您可以在几分钟内设计一个MIB定义文件，无论您计划创建SMIv1还是SMIv2 MIB，无论该MIB是标准的、专有的或者是错误的MIB。该工具支持多种平台，包括Windows、Linux、macOS等。

以下是安装和使用MG-SOFT Visual MIB Builder的步骤：

从网络上寻找破解版本进行安装。

打开Visual MIB Builder软件。在主界面上，单击“File”菜单，选择“New”选项，然后选择“SMIv2 MIB Module”。

在"File"菜单选择"import"导入厂商根节点"DOCKEER.my"。

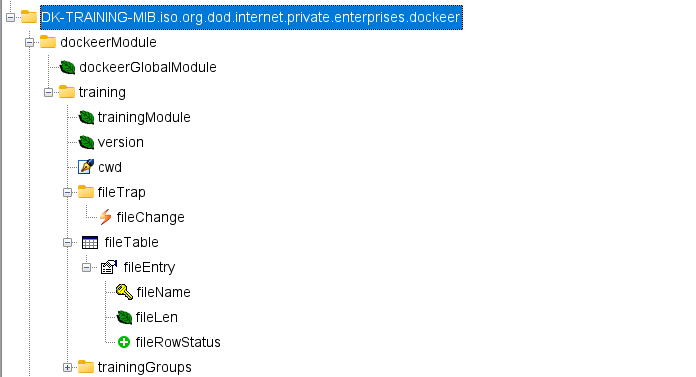
复制厂商根节点到新建的MIB树，询问复制缺失父节点和子节点时选择是。

按照下文要求，通过拖拽右侧组件完成MIB文件的创建。

点击"Tools" - "Check SMI Consistency"检查和修复语法格式错误

**要求**

最终需要将MIB文件写完整并通过代码进行实现。



|  |  |
| --- | --- |
| Syntax | INTEGER {active(1),notReady(3),createandWait(5),destroy(6)} |

**DKOS集成**

涉及代码权限/系统基础架构和通用库学习等前置条件

集成要点：

线程调度：单线程模式

agentx库转换及通用API使用

业务融合技巧

OID特点和规划

**Q&A**

消息调度机制？

**参考**

[什么是SNMP？](https://info.support.huawei.com/info-finder/encyclopedia/zh/SNMP.html)

[2.4. Extensions to the SMI in Version 2 docstore.mik.ua/orelly/networking\_2ndEd/snmp/ch02\_04.htm](https://docstore.mik.ua/orelly/networking_2ndEd/snmp/ch02_04.htm)

[Essential SNMP, 2nd Edition](https://download.itadmins.net/Linux/OReilly.Essential.SNMP.2nd.Edition.Sep.2005.pdf)

[SNMP 原理与实战详解](https://blog.csdn.net/bbwangj/article/details/80981098)

[net-snmp agent开发详解，非常简单](https://blog.csdn.net/hepeng597/article/details/8782868)

[TUT:Writing a Subagent](http://www.net-snmp.org/wiki/index.php/TUT:Writing_a_Subagent)