基于SNMP的MIB库转化成C文件的实现

张登银[1],陈 玮[1],任勋益[1,2],马晓东[2]
(1.南京邮电大学 计算机学院,江苏 南京 210003;
2.江苏常熟市亿通高技术股份有限公司,江苏 常熟 215500)

摘 要:网络管理日益重要,SNMP以其简单、灵活的特点而得到广泛应用。MIB库是网络管理中的资源以及属性的集合,是网管的核心,要访问MIB库,必须将其转换成C语言。MIB库转换成C文件,先要按照一定的语法格式编写MIB文件,将新编写的MIB文件放入MIB库中,然后根据是转换简单变量还是表变量使用NET-SNMP源码包自带的模板转换。经实验证明,这种方法能够简单快速地将MIB文件转换成C文件,并且转换后的C文件代码格式统一,经过少量的修改,就可以重新编译到SNMP源码包中使用。

关键词:简单网络管理协议;管理信息库;对象命名树
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2011)09-0006-04

Realization of MIB Database into C Based on SNMP

ZHANG Deng-yin[1], CHEN Wei[1], REN Xun-yi[1,2], MA Xiao-dong[2]
(1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;
2. Jiangsu Yitong High Tech Stock Limited Company, Changshu 215500, China)

Abstract: The growing importance of network management, SNMP is widely used for its simple, flexible features. MIB repository is a collection of resources and attributes in network management, is the core of network , to access the MIB database, it must be converted into C language. MIB database into C files, first according to certain syntax prepared MIB files, put the new MIB file into the MIB database, and then according to a simple variable or a table variable use the NET-SNMP source package's template conversion converted the MIB file to C file . The experiment proved that this method can quickly and easily convert the MIB file to C file, and converted the C code file format is unified, after a few changes you can recompile the source code to the SNMP package and use it.

Key words: SNMP; MIB database; object naming tree

0 引 言

随着计算机网络的迅速发展,网络管理变得越来越重要。SNMP以其简单、灵活的特点而得到广泛应用。MIB2C是一种基于NET-SNMP协议的把SNMP的业务逻辑转化为实际的C语言代码的工具。而把SNMP的业务逻辑转化为实际的C语言代码是一个比较繁琐的过程,文中从MIB库入手,研究如何把MIB文件转换成C语言文件的过程。

1 MIB库介绍

SNMP自从1988年发布以来,已成为TCP/IP网络中非常流行和实用的网络管理工具。一般来说SNMP网络管理体系结构包含:管理站、管理代理、管理信息库(MIB)和网络管理协议[1]。

管理信息库(MIB)是资源对象的集合,SNMP以MIB为基础来描述被监管资源。形成MIB库的资源对象表示被管资源的某方面属性。MIB变量记录了相连网络的状态、通信量统计数据、发生差错的次数以及内部数据结构的当前内容等相关信息。网络管理者可以通过对MIB库的存取访问,来实现性能管理、故障管理、安全管理、配置管理和计费管理这些功能[2]。管理信息库(MIB)是一种树型结构的数据库,被监管的对象都处于叶子节点上。每个被监管对象都由一个唯一的对象标识符(OID)来识别。

图1是管理信息库(MIB)的一部分,称为对象命名树(object naming tree)。

对象命名树的子树:root—iso(1)---org(3),在其下面有一个美国国防部的子树dod(6),所有通过dod的协议(例如TCP/IP)来通信的设备,能够从协议那里获得的信息都位于该子树下的internet节点[2],它的完

收稿日期:2011-01-19;**修回日期**:2011-04-26
基金项目:江苏省重大科技支撑计划(BE2009063);国家自然科学基金(61073188);国家博士后基金(20100471355)
作者简介:张登银(1964-),男,博士,CCF会员,研究方向为现代通信网络与信息处理技术。

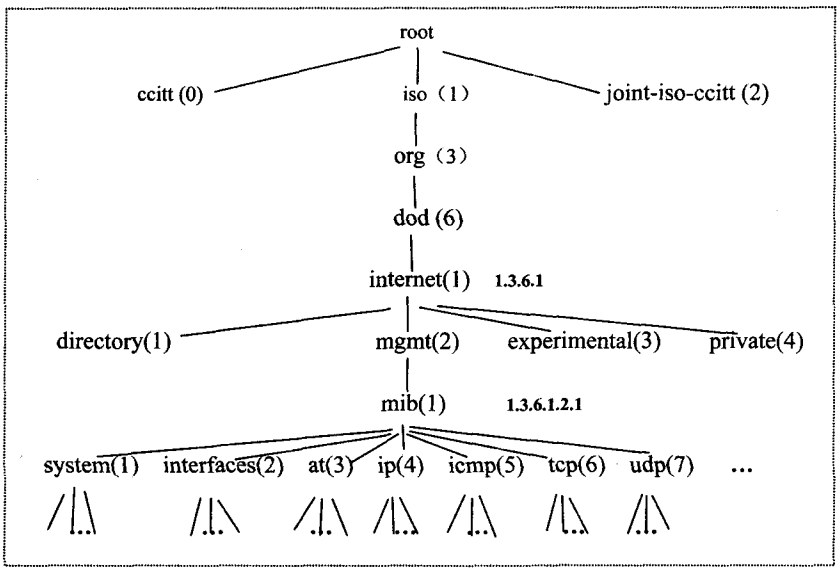

图1　管理信息库的对象命名树

整的OID是1.3.6.1,针对企业使用的OID可以申请使用挂载到internet节点下面的private节点上。

管理站通过获取MIB对象的值来实现监视功能。对象信息的存储结构由MIB定义的简单变量和表来构造,它一般包含对象标识符(OID)、数据类型、读写规则、功能描述、状态。MIB定义了四种基本数据类型:INTEGER,OCTET STRING,OBJECT IDENTIFIER和NULL。在这四种基本类型上扩展了新的类型,如:NetworkAddress、IpAddress、Counter、TimeTicks等[3]。

2　MIB2C工具

MIB2C是一个将MIB文件转换成C代码的工具,使用它转换成C可以简化写代码的过程和节约写重复代码的时间。MIB模块一般都由简单变量和表组

表1　mib2c所使用的配置文件

配置文件名	用途	示例文件
mib2c. scalar. conf	scalar	
mib2c. int_watch. conf	int scalar	
mib2c. mfd. conf	Table(官方推荐),需要修改的地方最少,但是最不灵活	if - mib/ifTable/ifTable *. c
mib2c. iterate. conf mib2c. iterate _ access. conf	Table,要获取的数据在Agent外部	mibII/vacm_context. c examples/netSnmpHostsTable *
mib2c. create - dataset. conf mib2c. table_data. conf mib2c. container. conf	Table变量,要获取的数据在Agent内部	examples/data_set. c
mib2c. array-user. conf	Table变量,要获取的数据在Agent内部,并对表排序	
mib2c. column _ defines. conf mib2c. column _ enums. conf	产生头文件	
mib2c. notify. conf	TRAP	
mib2c. old-api. conf	OLD API(4. X)	

成[4]。因此NET-SNMP把SMI中的对象分为两大类:Scalar和Table。Scalar就包含常用的整型、字符串、时间等数据类型。Table是Scalar的一种集合,有一个和多个列组成,类似于数据库中的表。它必须具有索引项,用来按一定顺序检索表项[5]。

MIB2C工具通过模板把MIB文件解析成.c和.h文件。针对MIB文件中不同的数据类型它有多种转换模板,如表1所示,这些模板都保存在安装目录下的/share/snmp目录里面。但MIB2C目前不支持同时解析Scalar和Table对象,对于具有这两种对象的MIB模块,需要分别生成代码文件,然后再合并成整体。一般情况下,使用mib2c. scalar. conf模板转化MIB中简单变量,用mib2c. iterate. conf模板转化MIB中的表[6]。

3　编写MIB文件

MIB使用标准抽象句法表示法ASN. 1(Abstract Syntax Notation One)语法规则定义。ASN. 1是一个构造复杂数据结构和信息结构的工具,用于定义数据的表示、结构和类型标识[7]。

MIB中的简单变量用ASN. 1描述如下:

objectname OBJECT-TYPE
SYNTAX syntax
MAX ACCESS access
STATUS status
DESCRIPTION "description"
::= { parent　number }

ASN. 1句法中关于定义一个简单变量的规则如下:

(1)objectname是对象的名字,该名字在MIB库中必须是唯一的。

(2)syntax为被管对象的类型,通常以大写字母开头,常用的类型有Counter、OCTET STRING、TimeTicks、IpAddress、INTEGER。

(3)access是被管对象的访问方式,通常为:read-only、read-write、no-accessible和read-create[8]。

(4)status表示被管对象的状态,可取的值为:mandatory(对象状态为必备)、optional(对象状态为可选)、current(对象状态为当前可用)、obsolete(对象状态为不再使用)。

(5)description为被管对象的文本描述。可增强

MIB 文件的可读性,用双引号把其文本说明括起来,也可为空,即“”。

(6)parent 表示此叶子对象的父树枝。

(7)number 表示此叶子对象在父树枝下是第几个对象[9]。

SNMP 中表是一个特殊类型的声明,表内声明的对象称为列对象,声明如下:

```
tablename OBJECT-TYPE
SYNTAX SEQUENCE OF tabletype
ACCESS not-accessible
STATUS mandatory
DESCRIPTION “description”
::= { (parent) (number) }

entryname OBJECT-TYPE
SYNTAX tabletype
ACCESS not-accessible
STATUS mandatory
DESCRIPTION description
::= { (tablename) 1 }
(tabletype) ::= SEQUENCE {
(column) (columntype) }
```

ASN.1 句法中关于定义一个表的规则如下:

(1) 表名(tablename)中,必须要有一个“Table”关键字,例如“xxx Table”。

(2)表目(entryname)中,必须有一个“Entry”关键字,例如“xxx Entry”。

(3)(column)是表的列对象,(columntype)是此列对象的类型,可以定义多个列对象和与之对应的列对象类型。

(4)tablename 和 entryname 中的 ACCESS 必须为 not-accessible[10]。

4 MIB 库转化成 C 方法

MIB 库模块定义后,需要在代理程序中增加与所定义的 MIB 库模块相对应的源代码;用 MIB2C 工具把 MIB 库模块文件转换成 C 语言。

Step1 把设计好的 MIB 库文件(XXX.txt)拷贝到 net-snmp 源代码的 mibs 目录中。

MIB2C 工具是使用 perl 脚本编写的,有些版本的 net-snmp 默认并没有安装 perl 模块,所以在使用 MIB2C 工具之前要手动安装 perl,它的源码在 net-snmp 源码包的 perl 目录下。

```
cd perl/
perlMakefile.PL [-NET-SNMP-CONFIG="sh../../net-snmp-config"] [-NET-SNMP-IN-SOURCE=true]
make
make test
make install
```

把 MIB 文件放入/usr/local/share/snmp/mibs 中,这里直接使用国家广电总局定义的 MIB 文件:NSCRTV-EPONEOC-ALARMS-MIB.txt 和 NSCRTV-ROOT.txt 文件。

Step2 遍历 MIB 树。

在使用 mib2c 时,先修改 mib2c 的配置文件:snmp.conf (/usr/local/share/snmp)。在配置文件中要加入:mibs +NSCRTV-ROOT 和 mibs +NSCRTV-EPONEOC-ALARMS-MIB,在配置好 snmp.conf 文件后,就可以使用 snmptranslate 工具遍历自定义的 MIB 文件,这样的好处是可以很清晰地了解 MIB 库的结构。同时,能用该工具遍历的 MIB 节点,都可以直接使用 MIB2C 工具转换成 C 代码,当不能把 MIB 文件转换成 C 文件时,可以先通过 snmptranslate 工具遍历自定义的 MIB 文件,如果遍历失败,可能是由于 snmp.conf 的配置或是 MIB 文件本身存在问题[11]。

```
root@yt-desktop:~# snmptranslate -IR -Tp alarmsIdent
+--alarmsIdent(2)
|
+-- -R-- INTEGER    alarmLogNumberOfEntries(1)
+-- -R-- INTEGER    alarmLogLastIndex(2)
|
   +--alarmLogTable(3)
      |
      +--alarmLogEntry(1)
         |  Index: alarmLogIndex
         |
         +-- -R-- INTEGER    alarmLogIndex(1)
         |          Range: 1..32767
         +-- -R-- String     alarmLogInformation(2)
                    Size: 17..255
```

这样得到了一棵 MIB 树,从树中可以清楚地看到每个 MIB 树的对象,值得注意的是 snmptranslate -IR -Tp xxx 的后缀名是 MIB 子树的根节点名,而不是 MIB 文件的名称。

Step3 export MIBS=ALL//export 是设置 mib2c 后对象名的查找范围。上面设置表示在 mibs 目录下查找所有文件。

Step4 mib2c modulename// modulename 是要转换的对象名,既不是 mib 文件名,也不是节点名;对于简单变量推荐使用 mib2c.scalar.conf 模板转换,对于表推荐使用 mib2c.iterate.conf 模板转换,当然用户也可以根据自己的实际情况选用其他的模板转换。

mib2c会在当前目录下生成modulename.h和modulename.c,这两个文件是根据所设计的MIB库模块转换而成的[12]。

export　MIBS=ALL

选择MIB树中的一个表alarmLogTable:

root@yt-desktop:~# mib2c -c mib2c.iterate.conf alarmLogTable

writing to alarmLogTable.h

writing to alarmLogTable.c

running indent on alarmLogTable.c

running indent on alarmLogTable.h

这样就在当前目录下产生一个表的alarmLogTable.c alarmLogTable.h文件;

选择MIB数中一个简单变量:

root@yt-desktop:~# mib2c -c mib2c.scalar.conf alarmLogNumberOfEntries

writing to alarmLogNumberOfEntries.h

writing to alarmLogNumberOfEntries.c

running indent on alarmLogNumberOfEntries.h

running indent on alarmLogNumberOfEntries.c

这样就在当前目录下产生一个简单变量的alarmLogNumberOfEntries.c和alarmLogNumberOfEntries.h文件。

5　结束语

MIB2C在SNMP中处于一个非常重要的地位,文中介绍了MIB库,MIB工具和怎样编写MIB文件。通过实际例子介绍了简单快速地将MIB文件转换成实际的C文件的方法。但是通过模版转换的C文件,只是个半成品,需要用户针对实际项目对代码进行添加、修改。而且转换后的C文件,只是针对MIB树上一个叶子节点,实际应用中,为了减少代码冗余,还需要把多个C文件合并成一个文件[13]。

参考文献:

[1] 武孟军.精通SNMP[M].北京:人民邮电出版社,2010.

[2] 蔡国森.基于SNMP的MIB库访问实现[J].计算机与信息技术,2005(3):12-15.

[3] 刘一兰.基于SNMP MIB编译器的实现及其生成器技术的研究[D].武汉:华中师范大学,2004.

[4] 赵小蓉,吕　斌,宋　凯.基于SNMP网络管理协议的MIB库分析与存取实现[J].电脑开发与应用,2001(7):56-59.

[5] 郭方方,王克冬,耿　科,等.应用ASN.1对SNMP中SMI进行描述的方法[J].应用科技,2001(8):34-38.

[6] 蔡　丽,张大方,谢高岗,等.基于SNMP网络管理的设计与实现[J].计算机应用,2003(5):30-35.

[7] 崔晓乾.基于SNMP的网络系统的设计和实现[D].成都:电子科技大学,2005.

[8] 岑贤道,安常青.网络管理协议及应用开发[M].北京:清华大学出版社,1998.

[9] 罗雅过.基于SNMP的MIB库访问实现研究[J].西安文理学院学报:自然科学版,2010(10):33-36.

[10] 蓝　波,杨　琴.基于SNMP动态扩展MIB的设计与实现[J].计算机工程,2004(4):192-194.

[11] Xions C. SMP/SNMP Version 2 : the Evolution of SNMP [M].[s.l.]:[s.n.],1992.

[12] Mauro D, Schmidt K. Essential SNMP[M].[s.l.]: Oilly&Associates,2005.

[13] RFC1157. A Simple Network Management Protocol[S].[s.l.]:[s.n.],2005.

(上接第5页)

[4] 江　涌,谷建华,杜鹏雷,等.无线传感器网络测试平台研究[J].计算机技术与发展,2010,20(9):188-192.

[5] 陈积明,林瑞仲,孙优贤.无线传感器网络仿真平台WSNSim的设计[J].传感技术学报,2006,19(2):457-462.

[6] 梁韦华,于海斌,臧传治,等.基于多智能体的无线传感器网络仿真平台的研究[J].系统仿真学报,2006,18(4):913-917.

[7] Newell A, Akkaya K. Distributed collaborative camera actuation for redundant data elimination in wireless multimedia sensor networks[J]. Ad Hoc Networks,2011,9(4):514-527.

[8] Yaghmaee M H, Adjeroh D A. Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks [J]. Computer Networks, 2009, 53 (11):1798-1811.

[9] 鲁　琴,杜列波,左　震.无线多媒体传感网节点能耗问题评述[J].传感器与微系统,2008,27(12):1-3.

[10] 孙　岩,马华东.无线多媒体传感器网络QoS保障问题[J].电子学报,2008,36(7):1412-1420.

[11] 李晓维,徐勇军,任丰原.无线传感器网络技术[M].北京:北京理工大学出版社,2007:129-156.

[12] Chen Feng, Dressler F. IEEE 802.15.4 Model for OMNeT++/INET Framework [EB/OL]. 2008-10-30[2011-01-20]. http://www7.informatik.uni-erlangen.de/~fengchen/omnet/802154/index.shtml.

[13] 金晓磊,闫红漫,翁之浩,等.基于虚拟数据库的信息系统集成研究[J].计算机技术与发展,2009,19(6):87-90.

[14] Mendes L D P, Rodrigues J J P C. A survey on cross-layer solutions for wireless sensor networks[J]. Journal of Network and Computer Applications,2011,34(2):523-534.