

1.添加并开启 SNMP 代理服务

1) windows 问题

①无法找到 snmputil 命令

'snmputil' 不是内部或外部命令，也不是可运行的程序或批处理文件。

解决方案一：

1、环境变量里添加 snmputil 的位置， 如图：

snmputil E:\snmputil\snmputil\Snmputil.exe

2、在 snmputil 目录下， 再次在 cmd 命令行输入：

snmputil get localhost public .1.3.6.1.2.1.1.5.0

```
E:\snmputil\snmputil>snmputil get localhost public .1.3.6.1.2.1.1.5.0
Variable = system.sysName.0
Value    = String LAPTOP-4VGPEC2M
```

解决方法二：

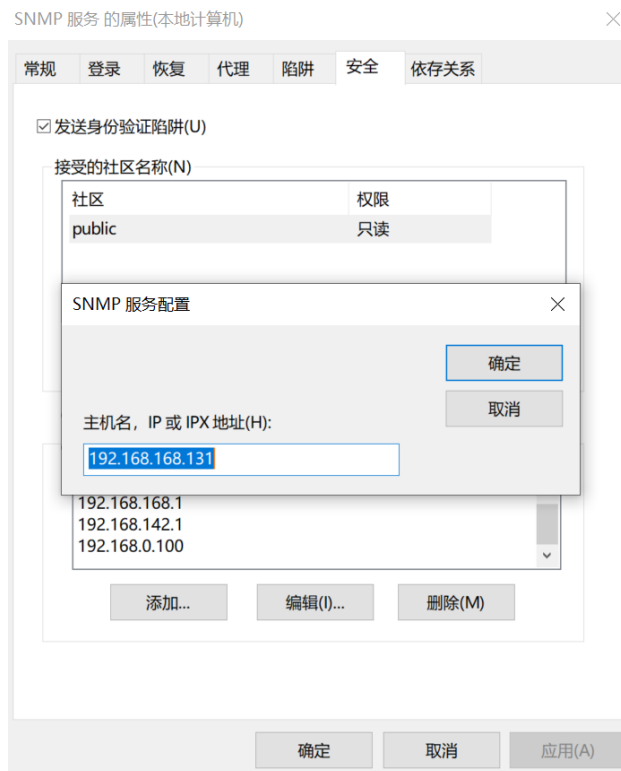
不用添加环境变量，

cmd 不可行，使用 powershell 在 snmputil 目录下输入：

./snmputil get localhost public .1.3.6.1.2.1.1.5.0

```
PS E:\snmputil\snmputil> ./snmputil get localhost public .1.3.6.1.2.1.1.5.0
Variable = system.sysName.0
Value    = String LAPTOP-4VGPEC2M
```

②注意 ip 要修改成管理端的 ip， 我的 linux 虚拟机 ip 如图： 192.168.168.131



2) linux 问题

①没有安装 gcc, 输入:

```
sudo apt-get install gcc
```

本次虚拟机使用的是 ubuntu, 如果是其他 linux 内核如 centOS、openEuler 等使用对应语句下载 gcc

②没有安装 libperl-dev, 输入:

```
sudo apt-get install libperl-dev
```

如果找不到这个包, 报错 (unable to locate package), 就更新一下源, 可输入:

```
sudo apt-get update
```

如果还不能找到, 输入:

```
sudo apt-get upgrade
```

③注意版本问题

实验的参考资料有 5.7.3 和 5.9.1 的版本, 注意要用 5.9.1 版本的参考资料, 本人因使用 5.7.3 版本的 net-snmp 产生许多问题。

④无法找到系统名字

```
test@ubuntu:/etc/snmp$ snmpget -v 2c -c public localhost sysName.0
sysName.0: Unknown Object Identifier (Sub-id not found: (top) -> sysName)
test@ubuntu:/etc/snmp$ snmpget --version
NET-SNMP version: 5.7.3
```

解决方案:

这个是因为你缺少snmp所需的mib库, 如下安装使用安装最新的mibs库

```
# sudo apt-get install snmp-mibs-downloader
```

然后再测试, 你还可能会出错, 那是因为snmp配置的mibs不是想要的, 你只需要如下操作

```
# sudo > /etc/snmp/snmp.conf
把snmp文件清空, 活着注释mibs那行也行!
```

2.编制控制台程序, 接受用户输入的 OID 字符串, 返回在 SNMP 代理中的对应值; (5 分)

1) 查询 oid 失败

虽然电脑的硬件设备的 oid 都是一样的, 但是要注意 linux 和 windows 的区别, 多线程的区别。

如查询 CPU 占用率的 oid, 我的 windows 下是.1.3.6.1.2.1.25.3.3.1.2, 但也能查到资料说是.1.3.6.1.4.1.2011.5.25.31.1.1.1.5, 要辨别清楚自己的系统是哪种

多线程的例子, 使用虚拟机时, 本机和虚拟机都会占用 cpu, 并且产生多线程, 这个和每台设备的处理器有关。

2) snmpget 和 snmpwalk 的区别

```
test@ubuntu:/usr/local/snmp/etc$ snmpget -v 2c -c public localhost .1.3.6.1.2.1.1
SNMPv2-MIB::system = No Such Object available on this agent at this OID
```

oid 值的管理是树形结构, snmpget 需要具体到某一个设备的 oid, 但是 snmpwalk 遍历当前 oid 下的所有子节点。所以当用 snmpget 查询一个父节点就会报如图的错误。

3.开发 GUI 界面程序,使用户可通过该程序观察主机 CPU、内存、硬盘空间、流量值;(5 分)

4.开发阈值告警功能, 用户通过界面可设置性能阈值 (如 CPU), 当超过阈值时自动报警。(3 分)

第三问可以调用第二问的查询函数, 通过查找不同硬件设备的 oid 值, 来达到目的。

1) CPU

CPU 的占用率并不能直接查找出来, 这个 oid 是 CPU 的负载情况。我的电脑有 8 核, 所以 CPU 占用率 = 平均负载 / 8

CPU: .1.3.6.1.2.1.25.3.3.1.2

2) 内存

内存同样也不能直接查找, 但可以查找每个进程使用的内存, 和总的内存数。那么内存占用率 = 所有进程占据内存之和 / 总内存

每个进程占据的内存: .1.3.6.1.2.1.25.5.1.1.2

内存总的大小: .1.3.6.1.2.1.25.2.2.0

3) 磁盘

磁盘也只能另辟蹊径, 我选择查找簇的数目和簇的大小来计算磁盘的总容量。那么总容量 = 簇的数目 * 簇的大小, 磁盘占用率 = 磁盘占用的簇 / 总容量

同时, 我发现了另一个有用的 oid, 可以描述不同的磁盘。除开我电脑已知的 C~F 盘, 还有另外两个盘, 如图。virtual 对应的是虚拟内存, physical 对应的物理内存。通过查询资料, 了解到为了让进程能有更大的内存, 选择使用虚拟内存对物理内存进行扩展, 采用页表交换的方式在有限的物理内存运行更大的进程, 再深入研究就涉及到操作系统的相关知识了。

```
请输入oid: .1.3.6.1.2.1.25.2.3.1.3
.1.3.6.1.2.1.25.2.3.1.3.1 "C:\\ Label:Windows-SSD Serial Number f0b9a3b2"
.1.3.6.1.2.1.25.2.3.1.3.2 "D:\\ Label:software Serial Number a8571146"
.1.3.6.1.2.1.25.2.3.1.3.3 "E:\\ Label:code Serial Number 143e62bb"
.1.3.6.1.2.1.25.2.3.1.3.4 "F:\\ Label:data Serial Number 60540a22"
.1.3.6.1.2.1.25.2.3.1.3.5 "Virtual Memory"
.1.3.6.1.2.1.25.2.3.1.3.6 "Physical Memory"
```

磁盘描述: .1.3.6.1.2.1.25.2.3.1.3

磁盘中簇的数目:.1.3.6.1.2.1.25.2.3.1.5

磁盘使用簇的数目: .1.3.6.1.2.1.25.2.3.1.6

4) 流量

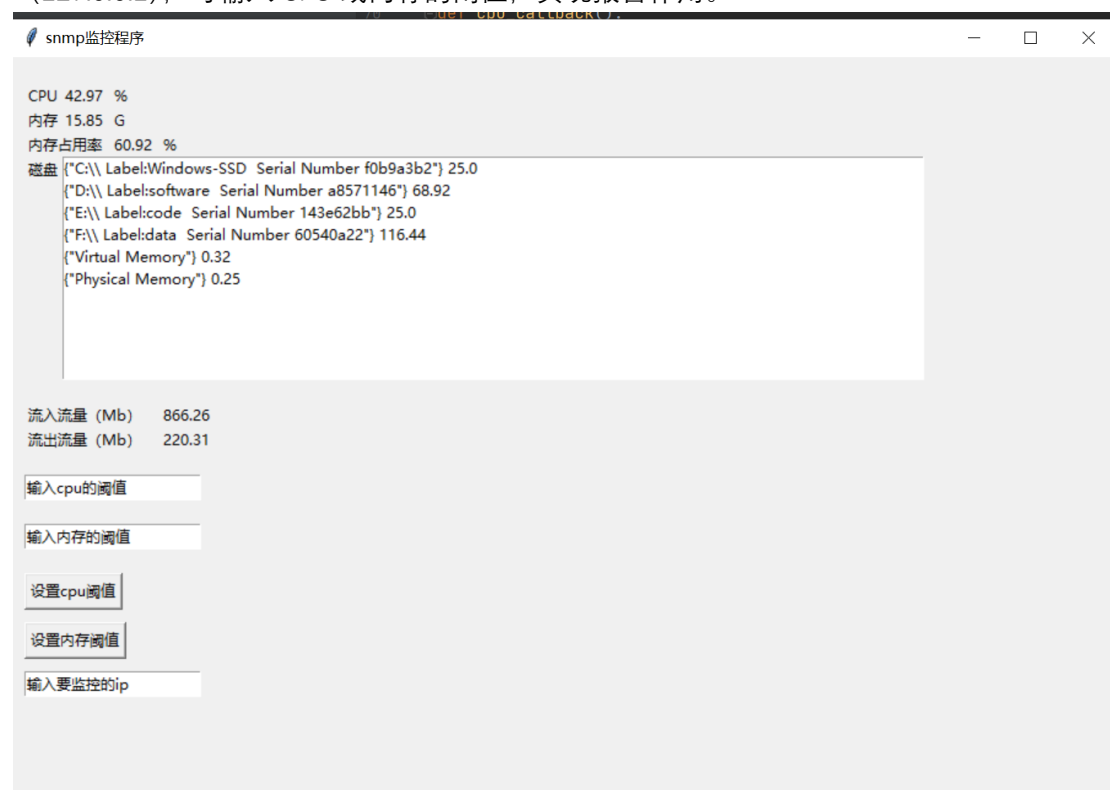
流量比较简单，直接查找流入与流出的字节数就可以。

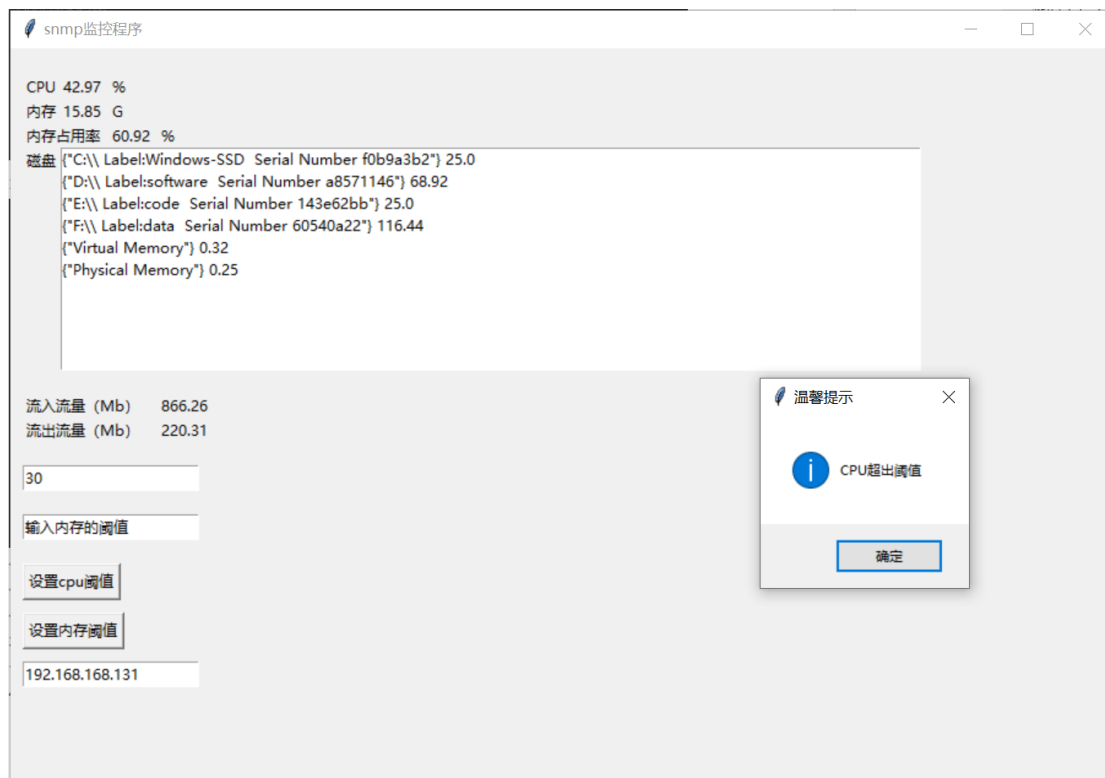
收到的流量: .1.3.6.1.2.1.2.2.1.10

发送的流量: .1.3.6.1.2.1.2.2.1.16

5) 程序结果

第三问和第四问都是同一个gui界面,所以一起展示。gui界面使用的是python的tkinter库,实现了观察目标主机的CPU、内存、磁盘、流量的功能。目标主机的ip默认值为localhost(127.0.0.1),可输入CPU或内存的阈值,实现报警作用。





5.实现 SNMP Trap、Set 功能并测试。(3 分)

1) set 命令实现

snmpset 命令将指定的值写入对象 ID。此外，除了要写入对象 ID 的值外，还必须在 snmpset 命令中指定对象 ID 的数据类型，因为 SNMP 对象支持多种数据类型。

使用 snmpset 命令更改 MIB 对象的值。

使用 snmpget 命令检验 MIB 对象更改为要求的值。

```
test@ubuntu:/etc/snmp$ snmpget -v 2c -c public 192.168.168.131 sysName.0
SNMPv2-MIB::sysName.0 = STRING: ubuntu
test@ubuntu:/etc/snmp$ snmpset -v 2c -c public 192.168.168.131 sysName.0 s xrj
SNMPv2-MIB::sysName.0 = STRING: xrj
test@ubuntu:/etc/snmp$
```

2) snmptrap 命令实现

当被管理设备出现问题时，被管理设备主动向 nms 发送消息报告问题。

管理站：

①配置 snmptrapd.conf 文件，与管理端的 snmpd.conf 文件在同一路径下。

```
authcommunity execute,log,net public
traphandle .1.3.6.1.2.1.1.5.0 /usr/local/snmp/etc/test_trap.pl
```

②编写对应的脚本语言处理不同 oid 返回的 trap 包，为了方便，我只设置了一个 oid

(1.3.6.1.2.1.1.5.0) 的处理程序。功能很简单，就是输出接收到的 trap 包。

③将 snmptrapd.conf 文件设置为默认配置，打开 162 端口，监听并接收 trap 输入：

```
snmptrapd -C -c /usr/local/snmp/etc/snmptrapd.conf -Lo -d -f
```

```
test@ubuntu:/etc/snmp$ sudo snmptrapd -d -f -Lo -C -c /usr/local/snmp/etc/snmptrapd.conf
```

上面命令中的选项表示：

-C：表示不使用 net-snmp 默认路径下的配置文件 snmptrapd.conf；

-c：指定 snmptrapd.conf 文件；

-d：显示收到和发送的数据报，通过这个选项可以看到数据报文；

-f：默认情况下，snmptrapd 是在后台中运行的，加上这个选项，表示在前台运行；

-L：指定日志记录在哪里，后面的 o 表示直接输出到屏幕上，如果是跟着 f 表示日志记录到指定的文件中

④效果如图，日志消息被直接输出到命令行

```
Sending 72 bytes to Local IPC: /var/agentx/master
0000: 01 03 08 00 0C 00 00 00 00 00 00 00 01 2F 4B 04 ...../K.
0016: 34 00 00 00 09 00 00 00 73 6E 6D 70 74 72 61 70 4.....snmptrap
0032: 64 00 00 00 FF 7F 00 00 07 06 00 00 03 00 00 00 d.....
0048: 0F 00 00 00 01 00 00 00 02 00 00 00 02 00 00 00 .....
0064: 01 00 00 00 0C 00 00 00 .....

Received 64 byte packet from Local IPC: abstract
0000: 01 12 00 00 0C 00 00 00 00 00 00 00 01 2F 4B 04 ...../K.
0016: 2C 00 00 00 6E F6 05 00 07 01 00 00 05 00 00 00 ,...n.....
0032: 07 06 00 00 03 00 00 00 0F 00 00 00 01 00 00 00 .....
0048: 02 00 00 00 02 00 00 00 01 00 00 00 0C 00 00 00 .....

```

代理：

①输入以下代码：

```
snmptrap -v 2c -c public localhost "user" .1.3.6.1.2.1.1 SNMPv2-MIB::sysContact.0 s 'test'
```

意思是向管理站的 user（可以为空“”）发送了一个内容为 test 的 trap 包。

```
test@ubuntu:/etc/snmp$ sudo snmptrap -v 2c -c public localhost "user" .1.3.6.1.2.1.1.5.0 SNMPv2-MIB::sysContact.0 s 'test'
```

3) 问题

①snmpset 无法修改对应值

```
test@ubuntu:/etc/snmp$ snmpset -v 2c -c public localhost sysName.0 i ubuntu
sysName.0: Bad variable type (Type of attribute is OCTET STRING, not INTEGER)
```

snmpset 命令中需要指定对象 ID 的数据类型，因为 SNMP 对象支持多种数据类型。i 对应的是 interger 整型，s 对应的是 string 字符串类型。我想修改系统名称，需要的数据类型是 string，所以会报错

②snmpset 命令无法写入

```
test@ubuntu:/etc/snmp$ snmpset -v 2c -c public localhost sysName.0 s injin
Error in packet.
Reason: noAccess
Failed object: SNMPv2-MIB::sysName.0
```

如果尝试使用公用团体 public 而非专用团体 private 来执行该 snmpset 命令，将不起

作用。这是因为 public 没有写入权限，该命令返回的原因不会对此进行解释，因为它只会声明对象不可写入。不同版本的 snmp 返回的原因不同，有的版本会返回 nowritable，具体查看官网文档[1]。

解决方案：

修改 snmpd.conf 文件，加入：

rwcommunity public

原有指令是 rocommunity public，我的理解是 public 团体是 read only 权限。改成 rw 意味着 read and write，所以 public 也能被写入了。

注意：根据教程，snmpd.conf 文件的路径在 /usr/local/snmp/etc/snmpd.conf 下，但是这个文件不是配置本地主机的 snmp 文件，正确的路径是 /etc/snmp/snmpd.conf 文件。只有在这个文件中添加上述代码，才能让本地主机能够被修改。

另一种解决方案（针对 windows）是重新创建一个 private 团体，默认是可写入的。显然方案二更方便。



③文件路径的问题

本次实验有两个 snmpd.conf 文件。教程里的 snmpd.conf 文件是用于 NMS 端的文件，本地主机路径 /etc/snmp/snmpd.conf 文件是当本机作为代理的配置文件。当要修改内容时，需要考虑清楚这个需求是作为代理还是作为管理站实现的。

如果需要让其他机器远程访问自己，就需要修改 /etc/snmp/snmpd.conf 文件配置；如果需要作为管理站接收其他主机发送过来的 trap 包，就需要修改 /usr/local/snmp/etc 下的 snmpd.conf 文件。

6.控制台动态显示主机 CPU、内存利用曲线（2 分）

这一步的想法是设置一个时钟函数，while 循环每秒执行一次查询函数，这样就可以得到 CPU 等硬件设备的实时状况，然后使用 python 库的 matplotlib 库对得到的数据进行实时绘图。但是两分懒得画了，摆烂。

[1] [TUT:snmpset - Net-SNMP Wiki](#)

[2] [Snmp 学习总结\(六\)——linux 下安装和配置 SNMP - 孤傲苍狼 - 博客园 \(cnblogs.com\)](#)