

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií



Dokumentácia k projektu ISA

Rozšírenie SNMP agenta

Obsah

1	Úvod	2
2	Teória	2
2.1	SNMP	2
2.2	MIB a OID	2
3	Implementácia	3
3.1	MIB súbor	3
3.2	Agent	3
4	Spustenie a otestovanie	4
4.1	Nastavenie DLMOD tabulky a preklad	4
4.2	Otestovanie	5

1 Úvod

Tento dokument popisuje implementáciu vlastného MIB modulu a dynamicky načítateľného rozšírenia SNMP agenta net-snmp.

2 Teória

2.1 SNMP

SNMP je skratka pre Simple Network Monitoring Protocol. Je to protokol na prenos informácií o správe v sieťach, najmä na použitie v sieťach LAN, v závislosti od zvolenej verzie.

Jeho užitočnosť pri správe siete spočíva v tom, že umožňuje zhromažďovať informácie o zariadeniach pripojených k sieti štandardizovaným spôsobom v širokej škále typov hardvéru a softvéru.

Málokto zo správcov sietí sa vzdá protokolu SNMP, pretože takmer všetky druhy zariadení od mnohých rôznych výrobcov podporujú protokol SNMP, čo im pomáha dosiahnuť komplexné monitorovanie vďaka technológii SNMP.

Sieť má zvyčajne najmenej jeden počítač alebo server, na ktorom je spustený monitorovací softvér a predstavuje riadiaci subjekt. Sieť tiež pravdepodobne bude mať niekoľko zariadení, od prepínačov až po tlačiarne alebo čokoľvek iné, čo je potrebné monitorovať. Sú to spravované zariadenia.

Správy SNMP sa odosiľajú a prijímajú medzi takzvanými manažérmi a agentmi. V zásade možno prenos správ SNMP porovnať s typickou komunikáciou medzi klientom a serverom, ktorá ponúka technológie pull(GET request) aj push(Pushed out, trap).

2.2 MIB a OID

OID znamená Object Identifier, teda OID jedinečne identifikujú spravované objekty, ktoré sú definované v súboroch MIB.

Hierarchia objektov (OID) sa všeobecne zobrazuje ako strom s rôznymi úrovňami od koreňa až po jednotlivé listy.

```
Iso(1).org(3).dod(6).internet(1).private(3).experimental(868).  
atsign-proxy(22).<implementovany objekt>(1-4)
```

Obrázek 1: Hierarchia OID objektov v projekte

MIB znamená Management Information Base a označuje nezávislý formát pre definíciu manažérskych informácií. Inými slovami, MIB obsahujú OID presne definovaným spôsobom. V MIB získa každý objekt svoju definíciu, ktorá určuje jeho vlastnosti v rámci spravovaného zariadenia. K objektom sa pristupuje pomocou protokolu SNMP.

```

IsaOSINFO OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"Object contains information about operating system."
 ::= { IsaSNMPMIB 4 }

```

Obrázek 2: Hierarchia OID objektov v projekte

3 Implementácia

3.1 MIB súbor

Pri návrhu MIB súboru som sa držal tutoriálov na dokumentačných stránkach [SNMP](#) a inšpiroval som sa už existujúcimi MIB súbormi, ktoré sa nachádzajú v *usr/share/snmp/mibs*. Ako prvé v MIB súbore som deklaroval importy iných MIB súborov, ktoré obsahujú definície objektov využívané v mojom implementovanom MIB. Je to napríklad *DisplayString*, používaný na objekt obsahujúci login alebo *experimental*, pod ktorý sa vlastne bude náš MIB modul implementovať. Následne som definoval vlastný MIB *MODULE-ENTITY*, v ktorom sa bližšie popisuje tento MIB súbor. Ako posledné som definoval jednotlivé skalárne objekty zo zadania, tak ako bolo ukázané na obrázku [2]. Vytvorené výsledné objekty som si zobrazil pomocou príkazu *snmpwalk*.

```

[user@localhost ISA]$ snmpwalk -v 1 -c public localhost .1.3.6.1.3.22
SNMPv2-SMI::experimental.22.1.0 = STRING: "xziska03"
SNMPv2-SMI::experimental.22.2.0 = STRING: "2020-11-12T07:14:52+05:00"
SNMPv2-SMI::experimental.22.3.0 = INTEGER: 69
SNMPv2-SMI::experimental.22.4.0 = STRING: "Operating_system_is_Linux!"

```

Obrázek 3: MIB objekty

3.2 Agent

Pri implementácii agenta som využil nástroj *mib2c*, ktorá dokáže vygenerovať kostru agenta v jazyku C. Predtým ako bolo ale možné použiť tento nástroj, bolo nutné vytvorený MIB súbor **načítať**. To tak, že náš MIB je treba vložiť do už spomenutej cesty *usr/share/snmp/mibs* obsahujúcej ostatné MIB súbory. Teraz už bolo možné generovať kostru agenta. Pre *mib2c* bolo potrebné špecifikovať argument *-c CONFIGFILE*, keďže v [dokumentácii](#) je dané, že pri generovaní kódu je potrebné špecifikovať konfiguračný súbor. Konfiguračný súbor som zvolil *mib2c.scalar.conf*, pretože naše objekty sú skalárneho typu.

```
mib2c -c mib2c.scalar.conf MIBNODE
```

Obrázek 4: mib2c

Vo vygenerovanej kostre bolo potrebné vytvoriť premenné, ktoré budú obsahovať jednotlivé hodnoty, špecifikované zo zadania. Pre každý objekt bola vygenerovaná funkcia *handle*, ktorá obsluhovala požiadavky (*MODE_GET*) na objekt. V týchto jednotlivých požiadavkách som na vyznačených pozíciách, doplnil vytvorené premenné.

Pre zobrazenie aktuálneho času som využil funkciu *localtime()* knižnicu *<time.h>*. Čas je formátovaný podľa **RFC 3339**, v ktorom sa ako ústredné časové pásmo používa UTC, ostatné časové pásma sú špecifikované offsetom.

Objekt s prepisovateľným integerom, okrem *MODE_GET* obsahuje aj *MODE_SET* požiadavky. Tieto požiadavky umožňujú aj vrátenie predchádzajúcej hodnoty, preto som pre tento objekt vytvoril dve premenné. Pre objekt obsahujúci informáciu o Operačnom systéme, som využil funkciu *uname* z knižnice *<sys/utsname.h>*. Informáciu ktorú zdieľam je názov operačného systému.

4 Spustenie a otestovanie

4.1 Nastavenie DLMOD tabulky a preklad

Ako prvé je treba otvoriť terminál, v ktorom bude bežať SNMP agent. V zložke s *Makefile* preložte projekt s *make*. Následne spustíte snmp agenta, nasledovne:

```
sudo snmpd -f -L -DisaSNMPMIB,dlmod
```

Obrázek 5: Spustenie agenta

Otvorte ďalší terminál. V tomto terminály ja najskôr potreba nastaviť DLMOD tabuľku nasledovne:

```
//Create a new row in the dlmod table
snmpset localhost UCD-DLMOD-MIB::dlmodStatus.1 i create

// See that the row was created
snmptable localhost UCD-DLMOD-MIB::dlmodTable

//set the properties of the row up to point to our new object and to give it a name
snmpset localhost UCD-DLMOD-MIB::dlmodName.1 s IsaSNMPMIB UCD-DLMOD-MIB::dlmodPath.1 s /path
/to/IsaSNMPMIB.so
ptable localhost UCD-DLMOD-MIB::dlmodTable

//load the shared object into the running agent
snmpset localhost UCD-DLMOD-MIB::dlmodStatus.1 i load

snmptable localhost UCD-DLMOD-MIB::dlmodTable
```

Obrázek 6: Nastavenie DLMOD tabuľky

4.2 Otestovanie

Otestovanie požiadavok na *snmpget* je možné pomocou príkazu. Obsah všetkých objektov je možné pozrieť na skôr spomenutom obrázku ??

```
snmpget localhost ISA-SNMP-MIB::OBJECT-NAME.0
```

Obrázek 7: Požiadavok get na objekt.

Otestovanie požiadavky *snmpset* pomocou nasledujúceho príkazov. Prvý príkaz zobrazí aktuálnu hodnotu objektu, druhý ju aktualizuje na 999 a posledným skontrolujeme, že prebehla zmena.

```
snmpget localhost ISA-SNMP-MIB::IsaINT.0  
snmpset localhost ISA-SNMP-MIB::IsaINT.0 i 999  
snmpget localhost ISA-SNMP-MIB::IsaINT.0
```

Obrázek 8: Požiadavok set na objekt.