

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí

Programování síťové služby: Rozšíření SNMP agenta

Obsah

1	Zadanie	2
2	Architektúra agenta	2
2.1	Registrácia modulu MIB	2
2.2	Nástroj <code>mib2c</code>	2
3	Implementácia	3
3.1	Textový súbor s definíciou MIB	3
3.2	Overenie MIB definície	4
3.3	Zdrojový kód modulu agenta	4
3.3.1	<code>loginObject</code>	4
3.3.2	<code>timeObject</code>	4
3.3.3	<code>integerObject</code>	5
3.3.4	<code>infoObject</code>	5
3.4	<code>Makefile</code>	5
3.5	SNMP daemon	5
3.6	Načítanie	6
3.7	Utility <code>snmpget</code> a <code>snmpset</code>	6
4	Bibliografia	8

1 Zadanie

Implementujte vlastný MIB modul a dynamicky načítateľné rozšírenie SNMP agenta **net-snmp**. Vaša MIB sa bude registrovať pod **OID .1.3.6.1.3** (iso.org.dod.internet.experimental), s vlastným číslom **22**. MIB bude obsahovať nasledovné SNMP objekty:

- read-only string s Vaším loginom (**.1.3.6.1.3.22.1**)
- read-only string, ktorý vráti aktuálny čas naformátovaný podľa RFC 3339 (**.1.3.6.1.3.22.2**)
- read/write Int32 (**.1.3.6.1.3.22.3**)
- read-only premennú Vašej voľby, ktorá bude obsahovať nejakú informáciu o systéme (operačný systém, množstvo RAM, ...) (**.1.3.6.1.3.22.4**)

Modul agenta využívajúci túto MIB bude samostatný binárny súbor, ktorý bude možno dynamicky načítať do bežiaceho agenta (t.j. nie subagent komunikujúci cez IPC). Implementácia bude v jazyku C a je možné využiť všetky hlavičkové súbory a knižnice ktoré sú súčasťou **net-snmp**, ako aj štandardnú knižnicu jazyka C (ISO/ANSI aj POSIX). Dokumentácia bude okrem vecí popísaných v spoločnom zadaní obsahovať príklad získania informácie zo všetkých štyroch objektov pomocou utility **snmpget**. Taktiež sa očakáva zdokumentovanie objektu **.1.3.6.1.3.22.4**. Odovzdané riešenie bude obsahovať súbor .txt s MIB definíciou, .c zdrojové kódy modulu agenta a Makefile súbor ktorý projekt preloží.

2 Architektúra agenta

Počas typického používania protokolu SNMP (Simple Network Management Protocol) sa softvér, ktorý spracováva požiadavky SNMP na sieťovom uzle, nazýva agent. Agent Net-SNMP (snmpd) je zodpovedný za spracovanie prichádzajúcich požiadaviek, ktoré mu boli odovzdané z transportných a spracovateľských vrstiev knižnice Net-SNMP. Agent sa skladá z mnohých častí. Tie obsahujú opäť časti z viacerých knižníc a rôzne časti samotného agenta. Spravidla interaguje s knižnicami, sieťou, doplnkami, inými procesmi, súborovým systémom, lokálnym OS a jadrom lokálneho OS.

2.1 Registrácia modulu MIB

Moduly MIB, ktoré sú zodpovedné za implementáciu častí stromu MIB, majú možnosť zaregistrovať spätné volania pre ľubovoľnú časť stromu MIB, keď sú inicializované. Registrácie môžu funkčne pochádzať z ktoréhokoľvek z nasledujúcich zdrojov: priamo skompilovaný v kóde, dynamicky načítané zdieľané objekty, sprostredkovatelia (prostredníctvom jedného z protokolov AgentX alebo SMUX) alebo rôzne konfiguračné súbory.

2.2 Nástroj mib2c

Dizajn agenta UCD SNMP vždy formovala túžba rozšíriť jeho funkčnosť pridaním nových modulov. Jedným z prvých vývojov od základnej kódovej základne CMU bola schopnosť volať externé skripty, čo je pravdepodobne najjednoduchšia metóda rozšírenia agenta. Existujú však okolnosti, keď sa takýto prístup považuje za nevhodný - možno z dôvodu rýchlosti, prístupu k potrebným údajom, spoľahlivosti alebo elegancie. V takýchto prípadoch je zjavným riešením poskytnúť C kód, ktorý je možné skompilovať do samotného agenta na implementáciu požadovaného modulu. Mnoho z najnovších vývojov v štruktúre kódu malo tento proces uľahčiť. Jedným z najnovších prírastkov do balíka je nástroj **mib2c**. Je navrhnutý tak, aby zabral časť stromu MIB (ako je definované v súbore MIB) a vygeneroval kostru kódu potrebnú na jeho implementáciu. Potrebne sú tri súbory:

- súbor s definíciou MIB

- hlavičkový súbor C (obsahuje definície verejne viditeľných rutín a môže ich úplne vygenerovať mib2c)
- súbor implementácie C (hlavná práca pri implementácii modulu)

3 Implementácia

3.1 Textový súbor s definíciou MIB

Prvý potrebný súbor, ktorý sa má implementovať, definuje modul MIB. Na začiatku definície sa nachádzajú importy: `OBJECT-TYPE` pre jednotlivé objekty, `Integer32` pre `read/write Int32`, `MODULE-IDENTITY` pre zavedenie samotného modulu `isaProjectMIB`, `experimental`, keďže pod dané OID sa má registrovať MIB. Okrem toho aj `SnmpAdminString` používané pre `login`, `date-time` a taktiež ako informácia o operačnom systéme. Následne je definovaný modul a jednotlivé objekty:

```
ISA-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
OBJECT-TYPE, Integer32,
MODULE-IDENTITY, experimental      FROM SNMPv2-SMI
SnmpAdminString                   FROM SNMP-FRAMEWORK-MIB
MODULE-COMPLIANCE, OBJECT-GROUP    FROM SNMPv2-CONF;
```

```
isaProjectMIB MODULE-IDENTITY
LAST-UPDATED "202010290000Z"
ORGANIZATION "fit-isa"
CONTACT-INFO "Viktoria Haleckova"
DESCRIPTION "A simple MIB module as a student's project."
::= {experimental 22}
```

```
loginObject OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This is a simple object which merely houses a string. It's only purpose is
to hold a value of a single string (login of a student)."
```

```
 ::= {isaProjectMIB 1}
```

```
timeObject OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This is a simple string object. It's value indicates current date-time according
to RFC 3339."
```

```
 ::= {isaProjectMIB 2}
```

```
integerObject OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This is an object that simply supports a writable integer."
```

```

 ::= {isaProjectMIB 3}

infoObject OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This is a simple string object. It's only purpose is to hold information
about the Operating System."
 ::= {isaProjectMIB 4}

END

```

3.2 Overenie MIB definície

Po vytvorení textového súboru s definíciou MIB je tento súbor overený prostredníctvom utility `snmptranslate`:

Použitý príkaz: `snmptranslate -On -M+. -mISA-MIB -IR loginObject`

Výstup programu: `.1.3.6.1.3.22.1`

Použitý príkaz: `snmptranslate -On -M+. -mISA-MIB -IR timeObject`

Výstup programu: `.1.3.6.1.3.22.2`

Použitý príkaz: `snmptranslate -On -M+. -mISA-MIB -IR integerObject`

Výstup programu: `.1.3.6.1.3.22.3`

Použitý príkaz: `snmptranslate -On -M+. -mISA-MIB -IR infoObject`

Výstup programu: `.1.3.6.1.3.22.4`

Textový súbor je pridaný do zložky `mibs` k ostatným MIB definíciám:

Použité príkazy: `sudo cp ./ISA-MIB.txt /usr/share/snmp/mibs`
`export MIBS="+ISA-MIB"`

Nástroj `mib2c` zachytí časť stromu MIB (ako je definovaný v súbore MIB) a vygeneruje šablónu zdrojového kódu v C, ktorý je potrebný na implementáciu príslušných objektov v ňom.

Použitý príkaz: `mib2c -c mib2c.scalar.conf isaProjectMIB`

3.3 Zdrojový kód modulu agenta

V tejto časti sa pracuje so šablónou, ktorú vygeneroval nástroj `mib2c`.

3.3.1 loginObject

Pre objekt obsahujúci login študenta sa používa globálna statická premenná:

```
static const char g_loginObject[9] = "xhalec00";
```

3.3.2 timeObject

Informácia o aktuálnom čase (date-time) sa nachádza v reťazci `buf []` a je naformátovaná podľa **RFC 3336**.

```
#include <time.h>
...
time_t rawtime;
time(&rawtime);
struct tm *info = localtime(&rawtime);
char buf[21];
strftime(buf, sizeof(buf), "%FT%TZ", p);
```

3.3.3 integerObject

Pre read/write Int32 objekt sa používajú globálne statické premenné: `static int g_integerObject;`
`static int g_integerObjectB;`

3.3.4 infoObject

Tento objekt získava informáciu o operačnom systéme nasledovným spôsobom:

```
#ifdef _WIN32
static const char g_infoObject[15] = "Windows 32 bit";
#elif _WIN64
static const char g_infoObject[15] = "Windows 64 bit";
#elif __linux__
static const char g_infoObject[6] = "Linux";
#elif __unix__
static const char g_infoObject[14] = "Other unix OS";
#elif __APPLE__
static const char g_infoObject[6] = "Apple";
#elif __FREEBSD__
static const char g_infoObject[8] = "FreeBSD";
#else
static const char g_infoObject[16] = "Unidentified OS";
#endif
```

3.4 Makefile

V tomto kroku je spísaný makefile, ktorý projekt preloží a vytvorí .so súbor:

```
CC=gcc
```

```
CFLAGS= -I. 'net-snmp-config --cflags'
DLFLAGS=-fPIC -shared
```

```
isaProjectMIB.so: isaProjectMIB.c Makefile
$(CC) $(CFLAGS) $(DLFLAGS) -c -o isaProjectMIB.o isaProjectMIB.c
$(CC) $(CFLAGS) $(DLFLAGS) -o isaProjectMIB.so isaProjectMIB.o
```

3.5 SNMP daemon

Na spustenie SNMP daemon je však najprv potrebné upraviť konfiguračný súbor `/etc/snmp/snmpd.conf`:

```
#####
#
# snmpd.conf
#
# - created by the snmpconf configuration program
#
#####
# SECTION: Access Control Setup
#
# This section defined who is allowed to talk to your running snmp agent.

# rwcommunity: a SNMPv1/SNMPv2c read-write access community name
# arguments: community [default|hostname|network/bits] [oid]

rwcommunity public
dlmod isaProjectMIB /home/user/Downloads/isaProjectMIB.so
```

Následne je možné spustiť SNMP daemon pomocou príkazu: `sudo snmpd -f -L -DisaProjectMIB,dlmod`

3.6 Načítanie

Na načítanie sú použité tieto príkazy:

1. `snmpset localhost UCD-DLMOD-MIB::dlmodStatus.1 i create`
2. `snmptable localhost UCD-DLMOD-MIB::dlmodTable`
3. `snmpset localhost UCD-DLMOD-MIB::dlmodName.1 s isaProjectMIB UCD-DLMOD-MIB::dlmodPath.1 s /home/user/Downloads/isaProjectMIB.so`
4. `snmptable localhost UCD-DLMOD-MIB::dlmodTable`
5. `snmpset localhost UCD-DLMOD-MIB::dlmodStatus.1 i load`

3.7 Utility snmpget a snmpset

V poslednom kroku dochádza k samotnému testovaniu:

Použitý príkaz: `snmpget localhost ISA-MIB::loginObject.0`

Výstup programu: `ISA-MIB::loginObject.0 = STRING: xhalec00`

Použitý príkaz: `snmpget localhost ISA-MIB::timeObject.0`

Výstup programu: `ISA-MIB::timeObject.0 = STRING: 2020-10-20T12:44:40Z`

Použitý príkaz: `snmpget localhost ISA-MIB::integerObject.0`

Výstup programu: `ISA-MIB::integerObject.0 = INTEGER: 0`

Použitý príkaz: `snmpset localhost ISA-MIB::integerObject.0 i 2`

Výstup programu: `ISA-MIB::integerObject.0 = INTEGER: 2`

Použitý príkaz: `snmpget localhost ISA-MIB::integerObject.0`

Výstup programu: `ISA-MIB::integerObject.0 = INTEGER: 2`

Použitý příkaz: `snmpget localhost ISA-MIB::infoObject.0`
Výstup programu: `ISA-MIB::infoObject.0 = STRING: Linux`

4 Bibliografia

- [1] Simple Network Management Protocol. Wikipedia [online]. Wikimedia Foundation, Inc., 2020 [cit. 2020-10-19]. Dostupné z: https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol
- [2] Management information base. Wikipedia [online]. Wikimedia Foundation, Inc., 2020 [cit. 2020-10-19]. Dostupné z: https://en.wikipedia.org/wiki/Management_information_base
- [3] snmptranslate. PTC MKS [online]. 2020 [cit. 2020-10-20].
Dostupné z: <https://www.mkssoftware.com/docs/man1/snmptranslate.1.asp>
- [4] Object identifier. Wikipedia [online]. Wikimedia Foundation, Inc., 2020 [cit. 2020-10-19].
Dostupné z: https://en.wikipedia.org/wiki/Object_identifier
- [5] Agent Architecture. The Net-SNMP Wiki [online]. 2015 [cit. 2020-10-19]. Dostupné z: http://www.net-snmp.org/wiki/index.php/Agent_Architecture
- [6] Net-SNMP Tutorial – MIB Module. The Net-SNMP Wiki [online]. 2020 [cit. 2020-10-20].
Dostupné z: http://www.net-snmp.org/tutorial/tutorial-5/toolkit/mib_module/
- [7] TUT:Writing a Dynamically Loadable Object. The Net-SNMP Wiki [online]. 2010 [cit. 2020-10-20].
Dostupné z: http://www.net-snmp.org/wiki/index.php/TUT:Writing_a_Dynamically_Loadable_Object
- [8] How to write a Mib module. The Net-SNMP Wiki [online]. 2010 [cit. 2020-10-20].
Dostupné z: <http://net-snmp.sourceforge.net/tutorial/tutorial-4/agent/01-intro.html>