



Počítačové komunikace a sítě
Dokumentace k 2. projektu
Varianta 3: DNS Lookup nástroj

Obsah

1	Popis zadání	2
2	Uvedení do problematiky	2
2.1	Domain Name System	2
2.2	Sestavení dotazu	2
2.3	Zpracování odpovědi	3
2.4	Způsob dotazování	3
3	Implementace	4
3.1	Popis implementace	4
3.2	Návratové kódy	4
4	Použití	4
4.1	Překlad	4
4.2	Spuštění	4
4.3	Příklady spuštění	5
5	Další informace	6
	Zdroje	7

1 Popis zadání

Úkolem projektu bylo nastudovat si details protokolu DNS, systému DNS obecně a následně naprogramovat v jazyce C/C++ nástroj, který se za pomoci síťové knihovny BSD sockets dotazuje systému DNS a realizuje překlad doménových jmen a IP adres.

2 Uvedení do problematiky

2.1 Domain Name System

DNS (Domain Name System) je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si vyměňují informace. Jeho hlavním úkolem jsou vzájemné převody doménových jmen a IP adres sítě. Slouží de facto jako distribuovaná databáze síťových informací.

2.2 Sestavení dotazu

Naše aplikace naváže přes BSD socket komunikaci s DNS serverem, který ji byl předán uživatelem jako jeden z parametrů. Aplikace sestaví zprávu (message), která musí přesně odpovídat následujícímu formátu.

Header
Question
Answer
Authority
Additional

Od serveru budeme chtít zjistit IP adresu doménového jména, který uživatel zadal jako další parametr při spuštění programu. Na server tedy budeme chtít odeslat dotaz (query). Asi není překvapením, že při sestavování dotazu, zůstanou sekce Answer, Authority a Additional prázdné.

Jako první nastavíme Header zprávy, ten má pevně danou velikost (12B) a má následující strukturu.

ID
QR Opcode AA TC RD RA Z RCODE
QDCOUNT
ANCOUNT
NSCOUNT
ARCOUNT

ID je identifikátor dotazu (16 bitové číslo), odpověď na něho bude mít stejný. Flag **QR** specifikuje jestli se jedná o dotaz nebo odpověď (response). **RCODE** je kód odpovědi, má hodnotu 0 pokud nenastala žádná chyba. **QDCOUNT** udává počet dotazů v sekci Question. **ANCOUNT**, **NSCOUNT** a **ARCOUNT** udávají počet zdrojových záznamů (Resource Record) v sekcích Answer, Authority a Additional. Při dotazu mají hodnotu 0.

Dále je potřeba nastavit sekci Question, ta má následující formát.

QNAME
QTYPE
QCLASS

QNAME je proměnné délky (až 255 bytů) a obsahuje doménové jméno na které se dotazujeme. **QTYPE** značí typ požadovaného záznamu (je zadán uživatelem jako parametr, např. **A** – IPv4) a **QCLASS** třídu požadovaného záznamu (1 – Internet). To je vše co ve zprávě musíme nastavit, následně dotaz odešleme na server.

2.3 Zpracování odpovědi

Aplikace poté čeká na odpověď serveru. Dobu čekání specifikuje uživatel přes parametr, nebo je použita výchozí hodnota. Server vrátí zprávu a vyplní zbylé tři sekce (Resource records), ty mají následující formát. Počet záznamů a další informace server specifikuje v Headeru.

RRNAME
RRTYPE
RRCLASS
TTL
RDLENGTH
RDATA

RRNAME je proměnné délky a jedná se o doménové jméno na které se aplikace ptala. **RRTYPE** specifikuje typ dat v **RDATA**. **RRCLASS** specifikuje třídu dat v **RDATA**. **TTL** (time to live) udává jak dlouho může být záznam uložen v cache. Po vypršení by měl zaniknout, protože se mezitím mohl změnit. **RDLENGTH** značí velikost dat v **RDATA**. A konečně **RDATA** jsou informace o které jsme server žádali. Například požadovali jsme záznam typu **A**, třídy **IN**, bude zde uložena IPv4 adresa doménového jména.

2.4 Způsob dotazování

Dotazování může probíhat dvěma způsoby – rekurzivně, nebo iterativně.

V případě rekurzivního dotazování aplikace pošle DNS serveru pouze jeden dotaz. Server jej zpracuje a pokud nezná odpověď, sám zasílá dotazy dalším serverům. V případě zjištění odpovědi na původní dotaz ji odešle klientovi.

U iterativního dotazování aplikace pošle DNS serveru dotaz a ten vrátí buďto odpověď nebo adresu dalšího serveru, který by odpověď mohl znát. Aplikace se poté ptá dalších serverů až dokud nedostane odpověď nebo má komu zasílat dotazy.

3 Implementace

3.1 Popis implementace

Aplikace je napsaná v jazyce C/C++ podle nejnovějšího standardu C++17. Síťová komunikace využívá UDP protokol a je realizována pomocí BSD socketů na portu 53, na kterém DNS servery pracují. Parametry příkazové řádky jsou ošetřeny pomocí funkce `getopt()` a je povoleno jejich libovolné pořadí. Timeout je řešen nastavením socketu na daný čas čekání na odpověď pomocí funkce `setsockopt()`. Odeslání zprávy na server je řešeno funkcí `sendto()` a přijmutí odpovědi funkcí `recvfrom()`. Odpověď se poté rozkládá a v případě úspěchu je vypsan záznam na `STDOUT` v následujícím tvaru:

```
<name> IN <type> <answer>\n
```

Nastane-li chyba v kterékoliv části programu, je zavolána funkce `error()`, která vypíše příslušnou chybovou zprávu na `STDERR` a ukončí program s odpovídajícím chybovým kódem.

Implementace je blíže popsána v hlavičkových souborech.

3.2 Návrátové kódy

- 0 – úspěch, uspokojující odpověď serveru
- 1 – neúspěch, neexistující záznam nebo vypršení timeoutu
- 2 – chybné zadání parametrů aplikace
- 3 – interní chyba (např. chyba alokace paměti)

4 Použití

4.1 Překlad

Překlad programu probíhá pomocí UNIXové utility `make` podle přiloženého `Makefile`.

```
$ make – pro překlad programu
```

```
$ make clean – pro smazání všech objektových souborů
```

```
$ make clean-all – pro smazání všech objektových a binárních souborů
```

```
$ make pack – pro zabalení všech zdrojových a hlavičkových souborů, Makefile a dokumentace
```

4.2 Spuštění

```
$ ./ipk-lookup [-h]
```

```
$ ./ipk-lookup -s server [-T timeout] [-t type] [-i] name
```

- `-h` (`help`) – volitelný parametr, při jeho zadání se vypíše nápověda
- `-s server` – povinný parametr, IPv4 adresa DNS serveru, kterého se budeme dotazovat

- **-T timeout** – volitelný parametr, doba čekání (v sekundách) na odpověď serveru, výchozí hodnota je 5 sekund
- **-t type** – volitelný parametr, typ dotazovaného záznamu (A (výchozí), AAAA, NS, PTR, CNAME)
- **-i (iterative)** – volitelný parametr, iterativní způsob rezoluce (výchozí rekurzivní)
- **name** – povinný parametr, překládané doménové jméno, v případě záznamu typu PTR se jedná o IPv4 nebo IPv6 adresu

Všechny parametry mohou být zadávány v libovolném pořadí a je možná jejich libovolná kombinace. Pouze parametr pro výpis nápovědy (-h) musí být zadán samostatně.

4.3 Příklady spuštění

```
$ ./ipk-lookup -s 8.8.4.4 www.fit.vutbr.cz
```

```
-----  
www.fit.vutbr.cz. IN A 147.229.9.23
```

```
$ ./ipk-lookup -s 8.8.8.8 -t AAAA www.sport.cz
```

```
-----  
www.sport.cz. IN AAAA 2a02:598:2::63
```

```
$ ./ipk-lookup -s 8.8.4.4 -t AAAA www.vladsminds.blogspot.com
```

```
-----  
www.vladsminds.blogspot.com. IN CNAME blogspot.l.googleusercontent.com.  
blogspot.l.googleusercontent.com. IN AAAA 2a00:1450:4014:80c::2001
```

```
$ ./ipk-lookup -s 8.8.4.4 -t CNAME -T 3 www.kn.vutbr.cz
```

```
-----  
www.kn.vutbr.cz. IN CNAME web.kn.vutbr.cz.
```

```
$ ./ipk-lookup -s 8.8.8.8 -t PTR -T 4 147.229.13.249
```

```
-----  
249.13.229.147.in-addr.arpa. IN PTR pc-cznic1.fit.vutbr.cz.
```

```
$ ./ipk-lookup -s 8.8.8.8 -t PTR 2001:67c:1220:8b0::93e5:b013
```

```
-----  
3.1.0.b.5.e.3.9.0.0.0.0.0.0.0.0.b.8.0.0.2.2.1.c.7.6.0.1.0.0.2.ip6.arpa. IN PTR merlin6.fit.vutbr.cz.
```

5 Další informace

Program se skládá z následujících zdrojových souborů:

`main.cpp` – `main`, volá zpracování parametrů, poté předá řízení programu modulu `lookup`

`lookup.cpp`, `lookup.h` – hlavní modul, komunikuje s DNS servery, sestavuje a zpracovává zprávy

`config.cpp`, `config.h` – zpracovává parametry programu

`error.cpp`, `error.h` – vypisuje chybové hlášení a ukončuje program v případě chyby

`Makefile`

Celkem se jedná o 797 řádků zdrojového kódu. Velikost výsledného binárního souboru `ipk-lookup` je 29.9 kB.

Zdroje

- [1] Simon Lewis, DNS messages [online], 2013-10-31, <https://justanapplication.wordpress.com/category/dns/dns-messages/dns-message-format/dns-message-header-format/>, [cit. 2018-04-09]
- [2] Silver Moon, DNS Query Code in C with linux sockets [online], 2011-10-21, <https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/>, [cit. 2018-04-09]
- [3] Network Working Group, RFC 1035 - Domain names - implementation and specification [online], 1987-11, <http://www.faqs.org/rfcs/rfc1035.html/>, [cit. 2018-04-09]
- [4] Wikipedia contributors, List of DNS record types [online], 2018-04-06, https://en.wikipedia.org/wiki/List_of_DNS_record_types/, [cit. 2018-04-09]
- [5] Wikipedia contributors, Reverse DNS lookup [online], 2018-03-02, https://en.wikipedia.org/wiki/Reverse_DNS_lookup/, [cit. 2018-04-09]
- [6] Wikipedia contributors, Domain Name System [online], 2018-04-07, https://en.wikipedia.org/wiki/Domain_Name_System/, [cit. 2018-04-09]