



Dokumentace

DNS Resolver

ISA - Síťové aplikace a správa sítí

Obsah

1	Úvod	1
2	Návrh a implementace	1
2.1	main()	1
2.2	ngethostbyname()	1
2.3	recieved_info()	1
2.4	Ostatní pomocné funkce	2
2.4.1	change_to_dns()	2
2.4.2	get_dns_servers()	2
2.4.3	reverse_IPv4()	2
2.4.4	reverse_IPv6()	2
2.4.5	read_name()	2
3	Návod k použití	3
4	Testování	3
5	Závěr	4
6	Zdroje	4

1 Úvod

Cílem projektu bylo napsat program v jazyce C vytvořit DNS resolver, neboli program, který bude umět zasílat dotazy na DNS servery a v čitelné podobě vypisovat přijaté odpovědi na standardní výstup.

2 Návrh a implementace

Aplikace je v jednom souboru s názvem `dns.c`, který se pomocí Makefile přeloží do `dns`. Z příkazového řádku vezme parametry zadané uživatelem, které projde, zjistí zda jsou v pořádku a pokud jsou, tak podle již zmíněných parametrů odešle DNS dotaz, získá odpovědi a ty poté vypíše na standardní výstup.

2.1 main()

V `main()` funkci se inicializují všechny proměnné, kterou budou potřeba, jako například všechny filtry na jejich počáteční hodnotu, pro typ to bude A a pro port to bude 53, který se případně podle vstupních argumentů přepíše.

Také se tu prochází všechny argumenty, nastaví jejich příslušnou hodnotu do filtrů. Uživatel zadá argumenty v libovolném pořadí ve formátu:

```
dns [-r] [-x] [-6] -s server [-p port] adresa
```

Po zkontrolování všech argumentů, případného ukončení s chybou, jde program do funkce `get_dns_servers()`.

2.2 ngethostbyname()

Funkce je zodpovědná za zasílání DNS dotazů na specifikované servery a zpracování odpovědí. Začíná inicializací a nastavením struktur pro přenos DNS zprávy. Na základě předaných argumentů formuluje dotaz a vytváří odpovídající síťové sokety pro odeslání dotazu. Pokud je k dispozici specifikace serveru, použije ji k navázání soketu, jinak použije výchozí port 53. Následně odešle dotaz na DNS server a čeká na odpověď.

Funkce obsahuje podmínky pro rekurzivní či reverzní dotazy a provádí konverze mezi IPv4 a IPv6 formáty v závislosti na argumentech. Po odeslání dotazu zpracovává příchozí data, čeká na odpověď a uvolňuje alokované prostředky. Výsledky jsou následně zpracovány v pomocných funkcích a výsledné informace jsou poskytnuty v rámci dalšího zpracování programem.

2.3 recieved_info()

V tomto odstavci se budeme zabývat funkcí na zpracování odpovědí na DNS dotazy v rámci programu. Po přijetí odpovědi na zasláný dotaz tato funkce dekoduje a vypisuje informace.

Nejprve funkce zjišťuje základní charakteristiky odpovědi - zda je autoritativní, zda se jedná o rekurzivní dotaz a zda je odpověď zkrácená. Poté identifikuje typ dotazu a jeho příslušnou sekci. Tímto způsobem je výstup strukturován a prezentován na výstupu.

Dále provádí postupné čtení a interpretaci jednotlivých částí odpovědi DNS serveru. Zpracovává sekci s odpověďmi, autoritativními servery a dodatečnými informacemi. Podrobně analyzuje každý záznam v odpovědi - extrahuje jména, typy záznamů, a v případě IP adres je konvertuje do lidsky čitelné podoby (IPv4 a IPv6) pro následný výstup.

Tato funkce je klíčovým prvkem pro interpretaci a vizualizaci odpovědí DNS serverů v rámci programu, umožňuje uživateli snadno porozumět získaným informacím z DNS dotazů a odpovědí.

2.4 Ostatní pomocné funkce

2.4.1 `change_to_dns()`

Převádí běžný internetový doménový název (např. “www.google.com”) na specifický formát používaný v DNS záznamech.

K převodu dochází tak, že každá část názvu domény (oddělená tečkou) je nahrazena určitým označením délky následující části a samotnou částí názvu. Například “www” je nahrazeno “3www”, “google” je nahrazeno “6google” a “com” se stane “3com”. Tímto způsobem je vytvořen nový řetězec obsahující název domény ve formátu používaném v DNS záznamech, kde délky jednotlivých částí jsou zakódovány před samotnými částmi doménového názvu.

2.4.2 `get_dns_servers()`

Tato funkce pouze nastaví výchozí hodnoty serverů na “208.67.222.222” a “208.67.220.220”.

2.4.3 `reverse_IPv4()`

Slouží k obrácení formátu IPv4 adresy. Přijímá jako vstup řetězec obsahující IPv4 adresu v podobě “93.184.216.34”. Nejprve kontroluje délku vstupního řetězce a pokud je poslední znak tečka, odstraní ji. Následně se pokusí převést tento řetězec na formu ‘struct in_addr’ pomocí funkce ‘inet_pton’, která konvertuje textový řetězec IPv4 adresy do binární podoby. V případě chyby konverze vypíše chybovou hlášku a ukončí program.

Poté, co získá binární podobu adresy, provede reverzní operaci. Z adresy v binárním formátu získá jednotlivé bajty a vytvoří z nich nový textový řetězec ve formátu in-addr.arpa. Tento nový řetězec reprezentuje reverzní verzi původní IPv4 adresy. Nakonec tuto reverzní adresu vrátí jako výstup funkce.

2.4.4 `reverse_IPv6()`

Funkce navržena pro reverzní transformaci IPv6 adresy do formátu in-addr.arpa. Přijímá jako vstup řetězec, který reprezentuje IPv6 adresu. Nejprve zjišťuje délku vstupního řetězce a poté iteruje přes jednotlivé znaky. Během tohoto procesu se snaží doplnit chybějící nuly pro dosažení přesné délky IPv6 adresy.

Kromě toho vypočítává celkový počet znaků, které by měly být v reverzní formě, a doplňuje chybějící nuly na odpovídající pozice. Pokud jsou znaky odděleny dvojicí dvojteček “::”, funkce určuje počet nul, které je třeba vložit mezi tyto znaky. Následně vytváří textový řetězec obsahující reverzní podobu IPv6 adresy v podobě in-addr.arpa.

Na závěr alokuje paměť pro výstupní řetězec, kopíruje do něj výslednou reverzní adresu a tento výsledek vrátí jako výstup funkce. Funkce tedy slouží k vytvoření reverzní podoby zadané IPv6 adresy v souladu s formátem in-addr.arpa.

2.4.5 `read_name()`

Slouží k čtení doménového jména ze zprávy DNS. Tato funkce pracuje s formátem, který reprezentuje doménové jméno v kompaktním formátu, například “3www6google3com”. Čte jednotlivé znaky a provádí dekompresi kompaktního formátu, abychom získali plné doménové jméno.

Nejprve inicializuje proměnné a alokuje paměť pro výsledné doménové jméno. Poté provádí dekompresi - pokud narazí na označení skokového offsetu (značeného prvními dvěma bity jako “11” v binární reprezentaci), tak se posune na příslušnou pozici v bufferu, kde se nachází následující část doménového jména.

Funkce také počítá počet znaků v doménovém jméně a při dekompresi správně přičítá hodnotu k to-muto číslu. Nakonec provede transformaci z kompaktního formátu (například “3www6google3com0”) na plné doménové jméno ve standardním tvaru (například “www.google.com”), kde odděluje jednotlivé části jména tečkou.

Výsledné plné doménové jméno je pak vráceno jako výstup funkce. Tato funkce je klíčová pro správné zpracování kompaktních doménových jmen ve zprávách DNS.

3 Návod k použití

Po překladu programu (příkazem `make`), je spustitelný následujícím způsobem

```
./dns [-r] [-x] [-6] -s server [-p port] adresa
```

- `dns` již přeložený program.
- `-r` požadujeme rekurzi.
- `-x` požadujeme reverzní dotaz.
- `-6` požadujeme dotaz typu AAAA místo A.
- `-s [server]` dotazovaný server.
- `-p port` číslo portu, na který se má poslat dotaz (jinak 53).
- `[adresa]` dotazovaná adresa.

Povinné argumenty jsou všechny, které nejsou v hranatých závorkách. Pokud nějaký argument bude zadán víckrát, program skončí s chybou.

4 Testování

Testování probíhalo přes příkaz `dig`, do kterého jsem zadal požadovaný příkaz a ten jsem poté kontroloval se svým výstupem.

Testy, spustitelné příkazem `make test`, napsané v jazyce `python`, zahrnují základní funkčnost projektu. Od základních dotazů po ty složitější. Zapomenout nešlo ani na špatné argumenty.

Provedené příkazy na testování:

1. `./dns -6 -x -s kazi.fit.vutbr.cz 2001:4860:4860::8888`
2. `./dns -6 -x -s kazi.fit.vutbr.cz 2001:67c:1220:809::93e5:917`
3. `./dns -r -6 -x -s dns.google 2a00:1450:400d:806::2004`
4. `./dns -r -x -s dns.google 15.197.204.56`
5. `./dns -r -x -s dns.google 93.184.216.34`
6. `./dns -r -s kazi.fit.vutbr.cz www.fit.vut.cz`
7. `./dns -r -x -s 2001:67c:1220:808::93e5:80c 2a00:1450:400d:80c::200e`
8. `./dns -r -s kazi.fit.vutbr.cz www.github.com`
9. `./dns -r -s 30 kazi.fit.vutbr.cz www.github.com`
10. `./dns -r kazi.fit.vutbr.cz www.github.com`

5 Závěr

Program se mi podařilo úspěšně naimplementovat, bez rozšíření, s veškerou požadovanou funkcí. Při testování jsem se nesetkal s žádnými chybami. Projekt byl zajímavý, naučil jsem se něco o DNS komunikaci a rozšířil jsem si tak znalosti, které mi byly předány na přednáškách a cvičích. Časovou náročností se řadil doprostřed.

6 Zdroje

BinaryTides. "DNS Query Code in C with Linux Sockets." BinaryTides, [20. listopadu 2023]. URL:

<https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/>

Stack Overflow. "Convert IP for Reverse IP Lookup." [20. listopadu 2023]. URL:

<https://stackoverflow.com/questions/16373248/convert-ip-for-reverse-ip-lookup>

The Open Group. "The Open Group Base Specifications Issue 7, 2018 edition: getaddrinfo()." [20. listopadu 2023]. URL:

<https://pubs.opengroup.org/onlinepubs/009619199/getad.htm>

Stack Overflow. "How to Reverse IPv6 Address for DNS Lookup - C." [20. listopadu 2023]. URL:

<https://stackoverflow.com/questions/58826162/how-to-reverse-ipv6-address-for-dns-lookup-c>