

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISA projekt - dokumentace
DNS Resolver

Obsah

1	Úvod	2
2	Použití	2
3	Implementace	2
3.1	main.c	2
3.2	error.c	2
3.3	arguments.c	3
3.4	message.c	3
3.5	sending.c	3
3.6	response.c	3
4	Testování	4
4.1	tests.c	4
4.2	tests.py	4

1 Úvod

DNS (Domain Name Service) je systém (a protokol), jehož hlavním účelem je získávání a převod doménových jmen a IP adres vzájemně mezi sebou. Tento projekt má za úkol poslat dotaz na daný DNS server a přijatá data zpracovat a vypsat uživateli v přívětivé formě. Při práci na tomto projektu jsem primárně čerpal z *RFC 1035*[2] a *RFC 3596*[1], kde je tento systém popsán.

2 Použití

Program lze spustit následujícím způsobem: `./dns [-r] [-x] [-6] -s server [-p port] adresa`

Argumenty lze psát v libovolném pořadí, zde je jejich vysvětlení:

- `-r`: je požadovaná rekurze pro získání záznamu
- `-x`: je požadován reverzní dotaz
- `-6`: je požadován dotaz typu AAAA místo výchozího A
- `-s server`: IP adresa nebo doménové jméno serveru pro zaslání dotazu
- `-p port`: číslo portu, na který se má dotaz poslat, výchozí je 53
- `adresa`: dotazovaná adresa

Program poté vypíše na stdout informace o přijaté odpovědi. Nejprve vypíše, zda je odpověď autoritativní, zda byla zjištěna rekurzivně a zda byla odpověď zkrácena. Dále vypíše všechny sekce a záznamy obdržené v odpovědi.

Pro každou sekci v odpovědi vypíše její název a počet získaných záznamů. Pro každý záznam vypíše jeho název, typ, třídu, TTL a data. Formát výstupu jsem implementoval podle ukázky v zadání projektu.

3 Implementace

Program jsem napsal v jazyce C a veškeré zdrojové soubory týkající se programu se nachází v adresáři `src/`. Dále se v adresáři `test/` nachází soubory týkající se testování tohoto programu, které popíšu později.

3.1 main.c

Tento soubor slouží jako hlavní soubor programu, obsahuje funkci `main()` a volají se v něm funkce z ostatních souborů, které postupně popíšu.

3.2 error.c

Soubor obsahuje jedinou funkci `error_exit()`, která slouží pro ukončení programu s návratovým kódem 1.

3.3 arguments.c

V tomto souboru se nachází funkce `arguments_parse()`, která slouží pro zpracování vstupních argumentů od uživatele a jejich následné uložení do struktury `Options` definované v hlavičkovém souboru. Tuto vyplněnou strukturu pak používají ostatní části programu.

Nespecifikuje-li uživatel server nebo danou adresu, program vypíše chybu a ukončí se. Dále se pak ukončí, pokud uživatel zadá kombinaci argumentů pro hledání AAAA záznamu a reverzní hledání (`-6` a `-x`).

3.4 message.c

Tento soubor obsahuje funkce pro vytvoření samotné DNS zprávy k poslání. Funkce `message_create_question()` ze zpracovaných argumentů vytvoří do struktury `Message` (definovaná v hlavičkovém souboru) dotaz na DNS server. Tato struktura samotná se skládá ze struktury `Header`, sloužící pro hlavičku DNS zprávy, řetězce `qname`, kde je uloženo jméno, na které se ptáme, a struktury `Question`, kde jsou ostatní informace o dotazu.

V tomto souboru jsou dále pomocné funkce `message_convert_to_qname()`, která slouží pro převedení uživatelem zadaného doménového jména do podoby určené pro DNS server, a funkce `message_convert_to_arpa()`, která slouží pro převedení IP adresy do požadované podoby (podporuje IPv4 i IPv6).

Poslední funkcí souboru je funkce `message_to_char()`, která převádí strukturu `Message` na pole charů.

3.5 sending.c

V tomto souboru se nachází pouze funkce `sending_packet()`, která slouží pro poslání DNS dotazu na požadovaný server a následné uložení odpovědi.

Nejdříve se připraví potřebné proměnné pro posílání packetů, poté se zpracuje zadané jméno serveru (je-li jako server zadané doménové jméno, převede se na IP adresu), packet se pomocí `sendto()` funkce pošle a pomocí funkce `recvfrom()` se přijme odpověď do proměnné `buffer`, která se pak vrátí.

Průběh celého posílání a přijímání je kontrolován a v případě chyby se program ukončí.

Obsah tohoto souboru byl inspirován druhou přednáškou předmětu ISA, *Pokročilé programování síťových aplikací TCP/IP*[3].

3.6 response.c

Hlavní funkcí tohoto souboru je `response_parse_and_print()`, která jako argument bere přijatou odpověď a postupně touto odpovědí prochází, zpracovává ji a vypisuje získaná data na standardní výstup.

Tato funkce nejdříve namapuje začátek přijaté odpovědi na strukturu `Header`, ze které pak přečte informace o počtu záznamů a z flagů vyčte, zda se jedná o autoritativní, rekursivní nebo zkrácenou odpověď. Poté prochází jednotlivé záznamy, které namapuje na strukturu `Resource` (případně `Question`), a vypisuje u nich jejich název, typ, třídu, TTL a samotné data.

Další funkce v souboru jsou `response_check_position()`, která zkontroluje aktuální pozici v odpovědi, jestli není mimo povolený rozsah, `response_print_type()`, která vypíše typ záznamu, `response_print_class()`, která vypíše třídu záznamu, `response_convert_and_print_name()`, která slouží pro převedení doménového jména zpět do čitelné formy a vypsaní, a funkce `response_print_rdata()`, která zkontroluje typ záznamu a vypíše data podle toho, o který typ se jedná.

4 Testování

V adresáři `test/` se nachází dva soubory sloužící pro základní otestování správné funkcionality programu, soubory popíšu níže. Testy lze spustit příkazem `make test`, avšak příkaz pro spustění Python testu je nejprve třeba odkomentovat v `Makefile`.

4.1 tests.c

V tomto souboru jsou základní testy pro otestování správné funkcionality funkcí zejména vracejících hodnotu, testuje se správné zpracování argumentů, správné vytvoření struktury `Message`, správné převádění doménového jména/IP adresy a správné posílání samotného dotazu.

4.2 tests.py

V tomto souboru jsou testy v jazyce Python, které slouží pro otestování aplikace jako celku - testuje se správné zpracování argumentů a správné posílání dotazů a získání (a zpracování) odpovědí pro záznamy typu `A`, `AAAA` a `PTR`.

Jelikož nevím, zda je na referenčním zdroji k dispozici Python 3, tak je spustění tohoto testu v `Makefile` zakomentováno a pro spustění je třeba buďto příkaz odkomentovat nebo ho přímo spustit pomocí: `python3 test/tests.py`.

Reference

- [1] Vladimir Ksinant, Christian Huitema, Dr. Susan Thomson, and Mohsen Souissi. DNS Extensions to Support IP Version 6. RFC 3596, October 2003.
- [2] Paul Mockapetris. Domain names - implementation and specification. RFC 1035, November 1987.
- [3] Matoušek Petr. Pokročilé programování síťových aplikací TCP/IP, 2023.
- [4] Wikipedia contributors. Domain name system — Wikipedia, the free encyclopedia, 2023.