VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Projekt ISA
-
DNS Resolver

Obsah

1. Popis zadání:

Cílem projektu bylo vytvořit program dns, který bude umět zasílat dotazy na DNS servery a následně vypisovat přijaté odpovědi.

2. Spuštění

V souboru se nachází *Makefile*, který umožní sestavit program pomocí příkazu *make*

Samotný dns program se spouští s použitím přepínačů v podobě:

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

- *-r* Volitelný přepínač, který určuje, že je vyžadovaná rekurze.
- *-x* Volitelný přepínač, který určuje, že se má zaslat reverzní dotaz
- *-6* Volitelný přepínač, který určuje zaslání dotazu typu AAAA. (výchozí A)
- *-s server* IP adresa nebo doménové jméno serveru, kam se má zaslat dotaz.
- *-p port* Volitelný přepínač, který určuje, číslo portu na který se má poslat dotaz. (výchozí 53)
- *adresa* dotazovaná adresa
- *-h* Vypíše nápovědu.

3. Implementace

3.1 Zpracování příkazové řádky

Na začátku se pomocí funkce *void getArgs(int argc, char *argv[], Params *params)* zpracují zadané přepínače z příkazové řádky a zapíše tyto informace do struktury *Params*

3.2 Popis implementace

Program je implementován pomocí jazyka C++ v souboru dns.cpp. Mezi hlavní funkce programu patří funkce *query(Params *params)*. Tato funkce vytváří UDP socket, sestavuje DNS dotaz, odesílá jej na DNS server, přijímá odpověď a následně zpracovává tuto odpověď. V kódu jsou použity struktury pro reprezentaci DNS zprávy a odpovědí, a to jak pro IPv4, tak i pro IPv6 adresy. Funkce také využívá další pomocné funkce, jako je DnsFormat() pro konverzi adresy do formátu DNS názvu a ReadName() pro čtení DNS názvu ze zprávy. Jejich popis je uveden níže.

Po vytvoření socketu se získá adresa serveru, na který se bude dotaz posílat. Podle zadaných parametrů z příkazové rádky se nastaví jednotlivé proměnné ve struktuře DNS_HEADER, která reprezentuje hlavičku DNS zprávy. Tato struktura obsahuje informace o celkovém formátu zprávy, identifikátoru, příznacích a dalších důležitých údajích.

3.3 Změna formátu adresy pro reverzní dotaz

Pokud byl zadán přepínač *-x* pro reverzní dotaz, změní se formát adresy pomocí funkce *char** reversedFormat(char* host) pro adresu typu IPv4, nebo funkce *void expandIPv6(const char** compressedIPv6, char* expandedIPv6, size_t expandedIPv6Size) pro typ IPv6.

- Funkce reversedFormat je zodpovědná za převod IPv4 adresy do reverzní podoby, která je používána v DNS pro PTR záznamy. Při PTR dotazu je potřeba převést běžnou IPv4 adresu na formu vhodnou pro reverzní dotaz, kde části adresy jsou zpětně pořadovány a přidána je koncovka "in-addr.arpa".

- Funkce expandIPv6 slouží k rozšíření zkrácené IPv6 adresy na plnou délku v čitelnějším tvaru, převede ji na formu pro reverzní dotaz a přidá koncovku "ip6.arpa".

3.4 Změna formátu adresy na DNS formát

Změna na formát DNS pomocí funkce *void DnsFormat (unsigned char** dns,char* host).

DNS názvy jsou ve zprávě reprezentovány speciálním formátem, kde jednotlivé části názvu jsou odděleny tečkou. Navíc délka každé části je zakódována na začátku, což umožňuje efektivnější reprezentaci názvů.

- dns: Ukazatel na buffer, kam bude zapsán výsledný DNS název.
- host: Řetězec obsahující adresu.

3.5 Odeslání dotazu

Pomocí funkce *sendto()* se odešle zpráva na zadaný server. Následně se příjme odpověď pomocí funkce *recvfrom*. Obě funkce jsou z knihovny **sys/socket.h**. Předtím, než se začne zpracovávat přijatá zprává, se na standardní výstup vypíše informace o získané odpovědi a odeslaném dotazu pomocí funkcí printInfo() a printQuestions(). Následuje zpracování přijaté zprávy pomocí funkce *unsigned char** ReadName (unsigned char* reader, unsigned char* buffer, int* count)

3.6 Zpracování odpovědi

*unsigned char** ReadName (unsigned char* reader, unsigned char* buffer, int* count)
Funkce ReadName je zodpovědná za čtení DNS názvu ze zprávy. Tato funkce je základní logikou při zpracování DNS odpovědí, protože DNS názvy jsou ve zprávě reprezentovány speciálním formátem.

- reader: Ukazatel na aktuální pozici v zprávě.
- buffer: Ukazatel na začátek bufferu s kompletní zprávou.
- count: Ukazatel na proměnnou, do které se uloží počet kroků při čtení.

3.7 Výstup programu

Po získání a zpracování přijaté zprávy se na standardní výstup vypíší informace o názvu, typu, třídě, TTL a samotná data zprávy pomocí funkce void printAnswers(RES_RECORD answers[],Params *params, int i,sockaddr_in a,in_addr,in6_addr)

Jednotlivé funkce a jejich detailnější popis je uveden v komentářích v kódu programu.

3.8 Nedostatky

Mezi hlavní nedostatky implementace patří nefunkční konverze IPv6 adresy pomocí funkce expandIPv6. Program nefunguje pokud byla za parametrem *-s* zadána IPv6 adresa.

3 Výstup programu

```
adamnieslanik@Adams-MacBook-Air ISA % ./dns -s kazi.fit.vutbr.cz www.fit.vut.cz
Authoritative: Yes,Recursion: Yes,Truncated: No
Question section (1)
 www.fit.vut.cz., A, IN
Answer section (1)
 www.fit.vut.cz., A, IN, 14400,147.229.9.26
Authority section (4)
 fit.vut.cz., NS, IN, 14400,gate.feec.vutbr.cz.
 fit.vut.cz., NS, IN, 14400,guta.fit.vutbr.cz.
 fit.vut.cz., NS, IN, 14400,kazi.fit.vutbr.cz.
 fit.vut.cz., NS, IN, 14400,rhino.cis.vutbr.cz.
Additional section (0)
```

Ob. 1 Výstup programu

4 Testování

Testy je možné spustit příkazem *make test* , který spustí python script s přiloženými testy. Jednotlivé testy porovnávají výstup programu *dns.cpp* a programu *dig*.

```
[adamnieslanik@Adams-MacBook-Air ISA % make
g++ -Wall -Werror -pedantic -std=c++11 -lpcap -o dns dns.cpp
[adamnieslanik@Adams-MacBook-Air ISA % make test
python3 test.py
-----------------------------------------------
ADDRESS TESTS:
Test 1
Expected : www.fit.vut.cz.
Got :      www.fit.vut.cz.
PASSED
------------------------
Test 2
Expected : 147.229.9.26
Got :      147.229.9.26
PASSED
------------------------
Test 3
Expected : 140.82.121.3
Got :      140.82.121.3
PASSED
------------------------
Test 4
Expected : 142.251.37.110
Got :      142.251.37.110
PASSED
------------------------
Test 5
Expected : prg03s13-in-f4.1e100.net.
Got :      prg03s13-in-f4.1e100.net.
PASSED
------------------------
-----------------------------------------------
QTYPE TESTS:
Test 1
Expected : PTR
Got :      PTR
PASSED
------------------------
Test 2
Expected : A
Got :      A
PASSED
------------------------
Test 3
Expected : A
Got :      A
PASSED
------------------------
Test 4
Expected : A
Got :      A
PASSED
------------------------
Test 5
Expected : PTR
Got :      PTR
PASSED
------------------------
-----------------------------------------------
Passed:  10
Failed: 0
```

Ob. 2 Výstup testů

5 Použité zdroje

[1] DNS Query Code in C with Linux sockets

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

[2] IPv4 / IPv6 reverse DNS

URL = https://www.whatsmydns.net/reverse-dns-generator

[3] INET_PTON()

URL = https://man7.org/linux/man-pages/man3/inet_pton.3.html

[4] Domain Name System (DNS) Parameters

URL = https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

[5] GETHOSTBYNAME()

URL = https://man7.org/linux/man-pages/man3/gethostbyname.3.html