

Manuál

DNS Resolver

Obsah

1	Úvod	1
1.1	Charakteristika DNS	1
1.2	Funkcie a účel programu DNS resolver	1
2	Vytvorenie spustiteľného súboru a automatizácia úloh	1
2.1	Požiadavky na systém	2
3	Spúšťanie programu	2
3.1	Príklady výstupov	2
4	Analýza a interpretácia výstupov	3
4.1	Rekurzívne a nerekurzívne dotazy	3
4.2	Reverzné dotazy	3
4.3	Typy dotazov AAAA a A	3
4.4	Výstupy a ich interpretácia	3
4.5	Využitie výsledkov	3
5	Riešenia problémov a chybové hlášky	3
6	Architektúra programu	4
6.1	Hlavný súbor	4
6.2	Triedy	4
6.3	Knižnice	5
6.4	Manuál	5
7	Testovanie programu	5
8	Rozšírenia	5
9	Referencie	6
10	Prílohy	7
10.1	Výpis ./dns -h na štandardnom výstupe	7
10.2	Tabuľka chybových kódov	8
10.3	Diagram tried programu	9
10.4	Príklad výstupu make test	10

1 Úvod

DNS (Domain Name System) resolver je kľúčovou súčasťou internetovej infraštruktúry, umožňujúcou prevod ľuďom čitateľných doménových mien (ako je napríklad `www.fit.vut.cz`) na IP adresy, ktoré používajú pre sieťovú komunikáciu. Údaje ukladá do databázy na serveroch a vďaka tomu pomáha ľuďom pristupovať k údajom pomocou názvov domén, a nie iba pomocou IP adries. Tento dokument poskytuje prehľad funkcií a implementácie programu `dns`, ktorý bol navrhnutý na zasielanie dotazov na DNS servery a na zobrazenie prijatých odpovedí na štandardnom výstupe.

1.1 Charakteristika DNS

DNS je hierarchický a decentralizovaný systém, ktorý umožňuje užívateľom na internete nájsť webové stránky a iné služby prostredníctvom ľuďom čitateľných názvov, tzv. doménových mien. DNS servery sú zodpovedné za prevod týchto názvov na IP adresy (alebo naopak), ktoré sú nevyhnutné pre nadviazanie sieťového spojenia. Packet DNS sa skladá z hlavičky, otázky, odpovedi, autorizovanej odpovedi a zvyšných (doplňujúcich) dát.

Hlavička obsahuje ID packetu, QR flagy (rozlišujú medzi otázkou a odpoveďou), TC, TD, RA, Z, OPCODE, QDCOUNT, ANCOUNT, NSCOUNT a ARCOUNT. Hlavička má vždy veľkosť 12B.

Otázka sa skladá z NAME (meno domény alebo IP adresa), TYPE (typ záznamu) a CLASS (typ komunikácie).

Odpoveď obsahuje odpoveď zo servera a skladá sa z NAME, TYPE, CLASS, TTL (čas platnosti odpovede), DL (dĺžka ďalšieho záznamu) a z dát odpovede.

Autorizovaná odpoveď je odpoveďou zo servera, ktorý je autorizovaný.

Doplňujúce dáta obsahujú dáta, ktoré aj keď nie sú priamou odpoveďou na otázku, tak môžu mať s ňou niečo spoločné.

1.2 Funkcie a účel programu DNS resolver

Program `dns` je nástroj navrhnutý na zasielanie dotazov na DNS servery a analýzu prijatých odpovedí. Umožňuje používateľom vykonávať dotazy v čitateľnej podobe a poskytuje informácie o odpovediach od DNS serverov. Priamo v programe je implementované zostavenie a odoslanie packetov, bez potreby nástrojov. Program podporuje komunikáciu pomocou UDP.

2 Vytvorenie spustiteľného súboru a automatizácia úloh

Prebieha pomocou Makefile na automatizáciu kompilácie, testovania a generovania dokumentácie.

- `make`: Kompiluje zdrojové súbory projektu a pomocou kompilátora `g++` vytvára spustiteľný súbor `dns`.
- `make run`: Spustí skompilovaný program `dns` s preddefinovanými argumentmi, čím demonštruje jeho funkčnosť.
- `make test`: Spustí testovacie skripty definované v adresári `tests`, čím overuje správnosť implementácie programu.
- `make doc`: Generuje dokumentáciu projektu vo formáte PDF pomocou nástroja `pdflatex`. Všetky súborové operácie sú vykonávané v rámci adresára `doc` a výsledná dokumentácia je presunutá do objektového adresára `obj` pre udržanie poriadku.
- `make clean`: Odstráni všetky objektové a dočasné súbory, ktoré boli vytvorené počas kompilácie, čím udržiava projektový adresár čistý.

- `make cleanall`: Rozširuje funkcionality `clean` o odstránenie spustiteľného súboru `dns`, vygenerovanej dokumentácie v PDF a všetkých výstupov z testov.

2.1 Požiadavky na systém

Pre úspešnú kompiláciu a fungovanie programu je potrebné zabezpečiť, aby bol program spustený v operačnom systéme Linux.

3 Spúšťanie programu

Použitie príkazu `dns` je nasledovné:

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

Poradie parametrov je ľubovoľné, ak nie je nastavená premenná prostredia `POSIXLY_CORRECT`. Každý parameter má špecifický význam:

- `-h`: Vypíše vysvetlenie použitia parametrov do štandardného výstupu. Príklad výstupu je v prílohe 10.1
- `-r`: Požadovaná rekurzia (Recursion Desired = 1), inak bez rekurzie.
- `-x`: Reverzný dotaz namiesto priameho.
- `-6`: Dotaz typu AAAA namiesto preddefinovaného A.
- `-s server`: IP adresa alebo doménové meno servera, kam sa má odoslať dotaz.
- `-p port`: Číslo portu, na ktorý sa má odoslať dotaz, preddefinovaný je 53.
- `adresa`: Adresa na zistenie.

3.1 Príklady výstupov

Odpoveď na príkaz `./dns -r -s kazi.fit.vutbr.cz www.fit.vut.cz`

```
Authoritative: Yes, Recursive: 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 (0)
Additional section (0)
```

Odpoveď na príkaz `./dns -r -6 -s kazi.fit.vutbr.cz www.fit.vut.cz`

```
Authoritative: Yes, Recursive: Yes, Truncated: No
Question section (1)
  www.fit.vut.cz., AAAA, IN
Answer section (1)
  www.fit.vut.cz., AAAA, IN, 14400, 2001:67c:1220:809::93e5:91a
Authority section (0)
Additional section (0)
```

Odpoveď na príkaz `./dns -x -s kazi.fit.vutbr.cz 147.229.9.26`

```

Authoritative: Yes, Recursive: No, Truncated: No
Question section (1)
  26.9.229.147.in-addr.arpa., PTR, IN
Answer section (1)
  26.9.229.147.in-addr.arpa., PTR, IN, 14400, www.fit.vutbr.cz.
Authority section (4)
  9.229.147.in-addr.arpa., NS, IN, 14400, rhino.cis.vutbr.cz.
  9.229.147.in-addr.arpa., NS, IN, 14400, gate.feec.vutbr.cz.
  9.229.147.in-addr.arpa., NS, IN, 14400, guta.fit.vutbr.cz.
  9.229.147.in-addr.arpa., NS, IN, 14400, kazi.fit.vutbr.cz.
Additional section (0)

```

4 Analýza a interpretácia výstupov

Táto sekcia rozoberá, čo možno zistiť vďaka použitiu programu `dns`.

4.1 Rekurzívne a nerekurzívne dotazy

Umožňuje určiť, či sa má vykonať rekurzívny (-r) alebo nerekurzívny dotaz. Rekurzívne dotazy žiadajú DNS server o kompletnú odpoveď, zatiaľ čo nerekurzívne vrátia len informácie, ktoré server už má.

4.2 Reverzné dotazy

Program umožňuje vykonávať reverzné dotazy (-x), vďaka čomu možno zistiť doménové meno priradené k určitej IP adrese.

4.3 Typy dotazov AAAA a A

Poskytuje možnosť vykonávať dotazy typu AAAA (-6) pre získanie IPv6 adries alebo typu A pre IPv4 adresy, čo umožňuje flexibilitu v získavaní sieťových informácií.

4.4 Výstupy a ich interpretácia

Výstupy programu sú prehľadné a informatívne, poskytujú podrobné údaje o odpovediach DNS serverov. Umožňujú získať dôležité informácie vrátane autoritatívnosti odpovede, rekurzívnych vlastností a detailov o dotazovanej adrese.

4.5 Využitie výsledkov

Na základe analýzy výsledkov možno získať užitočné informácie pre sieťovú diagnostiku, administráciu a plánovanie.

5 Riešenia problémov a chybové hlášky

Problémové stavy sú v programe riešené pomocou chybových kódov a výpisov chýb na štandardný chybový výstup. Príloha 10.2 obsahuje tabuľku chybových stavov a ich kódov. Trieda, ktorá obsluhuje chyby, má nasledujúci kód:

```

// Class will print error message and exit program
class Error {
public:

    // Constructor with an optional variable parameter

```

```

Error(int errCode = ERR_UNDEFINED, std::string message = "undefined", std::string
    variable = "") {
    // If variable is provided, replace %s in the message with variable
    if (!variable.empty()) {
        size_t pos = message.find("%s");
        if (pos != std::string::npos) {
            message.replace(pos, 2, variable);
        }
    }

    // Print the error message
    std::cout << ANSIREDF;
    std::cerr << "ERROR: ";
    std::cout << ANSI_RESET;
    std::cerr << message + "!\n";

    exit(errCode);
}
};

```

Listing 1: Kód triedy Error

6 Architektúra programu

Aplikácia je navrhnutá s viacerými modulmi, vrátane hlavného vstupného bodu, tried a knižníc, čo umožňuje modularitu a ľahšiu údržbu kódu. Diagram tried programu je v prílohe 10.3.

6.1 Hlavný súbor

Súbor `main.cpp` slúži ako hlavný vstupný bod programu, kde sa inicializujú a spúšťajú triedy aplikácie.

```

#include "libraries/main.hpp"

/*
 * Main function
 */
int main(int argc, char *argv[]) {

    DnsRequestSender *request = new DnsRequestSender;
    DnsResponseReceiver *response = new DnsResponseReceiver;
    int dnsResponseSize;

    // Parse arguments, send packet and parse response
    Arguments *arg = Arguments::parse_arguments(argc, argv);           // Parse command-line
                                arguments into an 'Arguments' object
    char *dnsResponse = request->execute(arg, &dnsResponseSize);        // Send DNS request
    response->parse(dnsResponse, &dnsResponseSize);                     // Parse DNS answer

    // Free memory
    delete[] dnsResponse;
    return 0;
}

```

Listing 2: Kód hlavného súboru

6.2 Triedy

V rámci adresára `classes` sú implementované nasledujúce triedy:

- **Arguments:** Obsahuje implementáciu triedy `Arguments`, ktorá spracováva argumenty príkazového riadku.
- **DNS:**
 - `DnsRequestSender`: Táto trieda je zodpovedná za odosielanie DNS požiadaviek a čaká na príchod odpovede.
 - `DnsResponseReceiver`: Trieda na spracovanie DNS odpovedí.
- **Error:** Trieda `Error` a jej hlavičkový súbor `Error.hpp` sa zaoberajú spracovaním a reportovaním chýb v programe.

6.3 Knížnice

- *ansi_colors.hpp*: Obsahuje farby, ktoré sú použité na štandardnom výstupe.
- *dns_constants.hpp*: DNS konštanty potrebné pre chod programu.
- *dns_structures.hpp*: Obsahuje štruktúru `dnsHeaders`, `DnsQuestions`, `DnsAnswers` a `SoaHeaders`.
- *dns_constants.hpp*: Importuje všetky kódy tried, knižníc a hlavičkových súborov.

6.4 Manuál

V adresári man sa nachádza súbor dns . 1, ktorý obsahuje manuál pre tento program.

7 Testovanie programu

Testovanie prebieha pomocou overovania rozdielov medzi očakávaným výstupom programu a výstupom programu po spustení testov. Testy je možné spustiť pomocou `make test`.

Príklad výstupu `make test` je možné vidieť v prílohe 10.4

```
for (( i=1; i<=$numOfTests_A; i++ )); do
    read -r args < $root_dir/$sub_dir/$i.in

    # Using the variables in the paths
    ./dns $args > "$root_dir/$sub_dir/$i.programOut"
    result=$(diff -b "$root_dir/$sub_dir/$i.expectedOut" "$root_dir/$sub_dir/$i.
        programOut")

    if [ -n "$result" ]; then
        echo -e "[\033[0;31mERR\033[0m] _TEST_$i: _Unexpected_result"
        echo "$result\n"
        counterError=$((counterError + 1))
    else
        echo -e "[\033[0;32mOK\033[0m] _TEST_$i: _Correct_result"
    fi
done
```

Listing 3: Ukážka kódu pre A testy

8 Rozšírenia

- **Chybové kódy a detailný výpis chyby na STDERR:** Obsahuje jedinečný chybový kód pre každý typ chyby. Na štandardný chybový výstup vypíše opis chyby aj s ukážkou chybového kódu.

- **man dns:** V `src/man/dns.1` je kód pre manuál tohto programu.
- **Spracovanie viacerých typov RDATA:** Program dokáže okrem typov CNAME, A, AAAA spracovať aj NS, MD, MF, SOA, MB, MG, MR, NULL, WKS, PTR, HINFO, MINFO, MX, TXT a SRV.
- **Kontrola vstupných argumentov:** Konroluje, či vstupné argumenty majú správny formát (napr. či je argument, kde má byť IPv4 adresa naozaj IPv4 adresa, takisto aj pre IPv6,...)

9 Referencie

Mockapetris, P. (1987). "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION". Network Working Group, Request for Comments: 1035, ISI, November 1987, <https://datatracker.ietf.org/doc/html/rfc1035>

Thomson, S., Huitema, C., Ksinant, V., Souissi, M. (2003). "DNS Extensions to Support IP Version 6". Network Working Group, Request for Comments: 3596, Cisco, Microsoft, 6WIND, AFNIC, October 2003, <https://datatracker.ietf.org/doc/html/rfc3596>

Cloudflare. (n.d.). "What is DNS? — How DNS works". <https://www.cloudflare.com/learning/dns/what-is-dns/>

10 Prílohy

10.1 Výpis `./dns -h` na štandardnom výstupe

```
./xlizic00@merlin: ~/skola/isa$ ./dns -h
DNS Resolver
Usage: ./dns [-r] [-x] [-6] -s server [-p port] address
-r: Recursion desired (Recursion Desired = 1), otherwise no recursion.
-x: Reverse query instead of direct query.
-6: Query type AAAA instead of default A.
-s: IP address or domain name of the server to which the query should be sent.
-p: The port number to which the query should be sent, default 53.
address: Queried address.
```

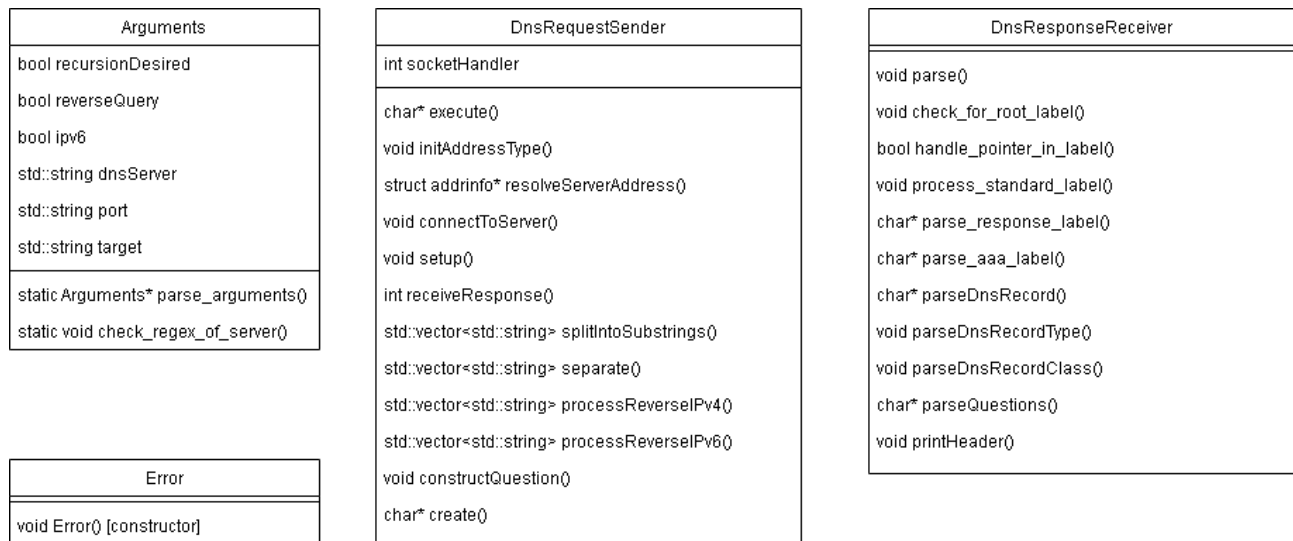
Obr. 1: Príklad výstupu `./dns -h` na štandardnom výstupe. Príklad bol realizovaný na serveri `merlin.fit.vutbr.cz`

10.2 Tabuľka chybových kódov

Názov chyby	Chybový kód
ERR_NONE	0
<i>Build Errors</i>	
ERR_BUILD	1
<i>Argument Errors</i>	
ERR_ARG_INVALID_ARGUMENT	10
ERR_ARG_UNUSUAL_PARAMETER	11
ERR_ARG_MISSING_SERVER	12
ERR_ARG_CAN_NOT_BE_X_AND_IPV6	13
ERR_ARG_IS_NOT_IPV4_ADDRESS	14
ERR_ARG_IS_NOT_IPV6_ADDRESS	15
ERR_ARG_IS_NOT_WEBSITE	16
<i>DNS Request Sender Errors</i>	
ERR_DNS_S_UNABLE_LOAD_SERVER	30
ERR_DNS_S_SOCKET_WAS_NOT_CREATED	31
ERR_DNS_S_DNS_CONNECTION_FAILED	32
ERR_DNS_S_UNABLE_LOAD_DNS_SERVER	33
ERR_DNS_S_IP_ADDRESS_NOT_FOUND	34
ERR_DNS_S_RESPONSE_FAILED	35
<i>DNS Response Receiver Errors</i>	
ERR_DNS_T_UNDEFINED	70
<i>Tests Errors</i>	
ERR_TESTS_UNDEFINED	80
<i>Documentation Errors</i>	
ERR_DOC_UNDEFINED	90
<i>Other</i>	
ERR_UNDEFINED	100

Tabuľka 1: Zoznam chybových stavov a ich chybových kódov

10.3 Diagram tried programu



Obr. 2: Diagram tried

10.4 Příklad výstupu make test

```
xlizic00@merlin: ~/skola/isa$ make test
g++ -std=c++20 -Wall -pedantic -Werror -static-libstdc++ src/main.cpp -o dns
bash ./tests/run_tests.sh

*****
*          TESTS START          *
*****
****    BASIC TESTS: A    ****
[ OK ] TEST 1: Correct result
[ OK ] TEST 2: Correct result

****    BASIC TESTS: AAAA    ****
[ OK ] TEST 1: Correct result
[ OK ] TEST 2: Correct result

****    ERROR TESTS    ****
ERROR: the target `www.fit.vut.cz` is not IPv4!
[ OK ] TEST 1: Correct result
ERROR: the target `notIPv4` is not IPv4!
[ OK ] TEST 2: Correct result
ERROR: the target `notIPv6` is not IPv6!
[ OK ] TEST 3: Correct result
ERROR: the target `notAWebsite` is not a website url!
[ OK ] TEST 4: Correct result
ALL TESTS PASSES (8/8)

xlizic00@merlin: ~/skola/isa$
```

Obr. 3: Výstup make test