



# Filtrujúci DNS resolver

Sieťové aplikácie a správa sietí

2020/2021

18. novembra 2020

**Autor:** Martin Fekete ([xfeket00@stud.fit.vutbr.cz](mailto:xfeket00@stud.fit.vutbr.cz))

# Obsah

<b>1</b>	<b>Zadanie</b>	<b>1</b>
<b>2</b>	<b>Problematika</b>	<b>1</b>
2.1	DNS	1
2.2	Štruktúra DNS paketu	1
2.2.1	Hlavička	1
2.2.2	Otázka	2
2.2.3	Odpoveď	2
<b>3</b>	<b>Implementácia</b>	<b>3</b>
3.1	Základné jadro programu	3
3.2	Filtrovanie domén	3
3.3	Podpora IPv4 a IPv6	3
3.4	Chybové stavy	4
<b>4</b>	<b>Preklad a spustenie</b>	<b>4</b>
4.1	Preklad	4
4.2	Spustenie	4
4.3	Príklad spustenia	4
<b>5</b>	<b>Testovanie</b>	<b>5</b>
5.1	Automatické testovanie	5

# 1 Zadanie

Zadaním projektu bolo implementovať DNS resolver, ktorý filtruje požiadavky typu A smerujúce na domény a ich poddomény vrámci dodaného zoznamu. Domény, ktoré sa v tomto zozname nenachádzajú sú preposiellané špecifikovanému DNS serveru. Odpoveď od serveru bude následne preposlaná pôvodnému žiadateľovi. Zoznam blokovovaných domén je dodaný v textovom ASCII súbore.

## 2 Problematika

### 2.1 DNS

Systém doménových mien (DNS) je hierarchický a decentralizovaný menný systém, ktorého základnou úlohou je mapovanie doménových adries na IP adresy [4]. DNS tak významne uľahčuje prácu pre ľudí, nakoľko pre človeka je oveľa jednoduchšie zapamätať si názov zariadenia alebo domény ako IP adresu.

### 2.2 Štruktúra DNS paketu

V tejto časti je popísaná základná štruktúra DNS paketu a blžšie popísané časti hlavička, otázka a odpoveď. Informácie k tejto časti pochádzajú z [5].

Header	hlavička
Question	otázka
Answer	odpoveď
Authority	autorita (autorizovaná odpoveď)
Additional	odpovede naviac

#### 2.2.1 Hlavička

DNS hlavička je súčasťou každého DNS paketu a má fixnú veľkosť 12B. Hlavička obsahuje nasledujúce prvky:

- ID - identifikačné číslo paketu (16b)
- QR - príznak identifikujúci či sa jedná o otázku alebo odpoveď (1b)
- OPCODE - označuje variantu paket (4b)
- TC - príznak identifikujúci poškodený paket (1b)
- RD - príznak identifikujúci, či je vyžiadaná rekurzia (1b)
- Z - rezerované miesto (vždy by tam mala byť 0) (1b)
- RA - príznak identifikujúci či server dokáže vykonať rekurziu (1b)

- QDCOUNT - počet otázok (16b)
- ANCOUNT - počet odpovedí (16b)
- NSCOUNT - počet autorizovaných odpovedí (16b)
- ARCOUNT - počet odpovedí navyiac (16b)

### 2.2.2 Otázka

Sekcia premenlivej veľkosti, ktorá obsahuje QDCOUNT otázok (zvyčajne 1). Ďalej obsahuje časti TYPE a CLASS.

- QNAME - meno domény, ktorá má byť preložená (premenlivá veľkosť). Časť obsahujúca doménové meno reprezentované ako sekvenciu štítkov (labels). Každý štítok obsahuje číslo, ktoré vyjadruje počet oktetov, ktorým bude nasledovaný.
- QTYPE - typ záznamu (16b). 16 bitové pole obsahujúce informáciu o type otázky. V projekte bol braný do úvahy iba typ QNAME, teda záznam obsahujúci IPv4 adresu. Medzi ďalšie typy patria napríklad AAAA (záznam obsahujúci IPv6 adresu), PTR (ukazateľ na doménové meno) alebo MX (mail exchange).
- QCLASS - trieda komunikácie (16b)

Ak je otázok v tejto sekcii viac, hore spomenuté časti budú opakované pre každú otázku zvlášť.

### 2.2.3 Odpoveď

Podobne ako otázka je aj sekcia odpoveď premenlivej veľkosti. Obsahuje nasledujúce časti:

- NAME - meno domény, ktorá má byť preložená (premenlivá veľkosť)
- TYPE - typ záznamu, podmnožina typu QTYPE (16b)
- CLASS - trieda komunikácie, napr. IN (internet) (16b)
- TTL - time-to-live, dĺžka platnosti odpovede (32b)
- RDLENGTH - dĺžka záznamu RDATA (16b)
- RDATA - dáta odpovede (premenlivá veľkosť)

Rovnako ako pri otázke, v prípade viacerých odpovedí budú spomenuté časti zopakované pre každú jednu odpoveď.

## 3 Implementácia

Program je implementovaný v programovacom jazyku C++. Použité boli systémové knižnice, STL a knižnice pre prácu so sieťovými nástrojmi.

### 3.1 Základné jadro programu

Na začiatku sú spracované argumenty pomocou funkcie `getopt` a vzápätí je skontrolovaná ich validita (teda napr. či boli zadané všetky povinné argumenty).

Ďalej je spustený UDP server, ktorý beží implicitne na porte 53, čo môže byť upravené argumentom `-p`. Po prijatí novej správy vo funkcii `recvfrom` je vykonaný fork programu. Rodičovský proces pokračuje v prijímaní nových správ a child proces spracováva prijatý UDP paket.

Po prijatí paketu je skontrolované, či sa jedná o DNS typu A. Ak nie, je klientovi odoslaná odpoveď s chybovým kódom 4, teda *Not Implemented* [5]. Následne je z DNS správy prečítaná časť `QNAME` a porovnaná s doménami zo súboru obsahujúceho zoznam blokovaných domén. Ak doména nie je blokovaná, je celá DNS žiadosť preposlaná špecifikovanému serveru pomocou funkcie `sendto`. Po prijatí odpovedi zo strany serveru je celá odpoveď preposlaná pôvodnému klientovi. V prípade, že resolver neprijme odpoveď od špecifikovaného DNS serveru do 3 sekúnd, klientovi je odoslaná odpoveď s chybovým kódom 2, teda *Server failure*. Napríklad program `dig` má túto hodnotu nastavenú implicitne na 5 sekúnd [1].

### 3.2 Filtrovanie domén

Zoznam blokovaných domén je najprv uložený do vektoru `blacklist` a tento vektor je ešte pomocou funkcie `sort` usporiadaný, čo umožňuje pri ďalšej práci využiť binárne vyhľadávanie. Kontrola, či má byť doména vyfiltrovaná alebo nie prebieha vo funkcii `is_filtered`, kde je každý štítok domény najprv vložený do vektoru. Potom je postupne doména z tohto od konca porovnávaná s obsahom s vektorom blokovaných domén. Teda napríklad v prípade domény `docs.google.com` je najprv vyhľadávaná v zozname blokovaných domén doména `com`, potom `google.com` a nakoniec `docs.google.com`. Vyhľadávanie vo vektore blokovaných domén je implementované pomocou binárneho vyhľadávania, čo tento proces značne urýchľuje. Ak bola doména alebo jej časť nájdená vo vektore blokovaných domén, tak je z funkcie vrátená hodnota `true` a klientovi je odoslaná odpoveď s chybovým kódom 5 (*Refused*).

### 3.3 Podpora IPv4 a IPv6

Pre možnosť spracovania ako IPv4, tak aj IPv6 žiadostí je na ich príjem vytvorená IPv6 schránka, kde je pomocou funkcie `setsockopt` vypnutý príznak `IPV6_V6ONLY`. Toto umožňujú tzv. IPv4-namapované adresy [3]. Na väčšine súčasných systémov je takáto možnosť povolená, na niektorých je dokonca príznak `IPV6_V6ONLY` vypnutý implicitne. Nájdu sa však systémy, ktoré takéto mapovanie nepodporujú a je pre spracovanie IPv4 a IPv6 adries nutné vytvoriť separátne schránky [2].

### 3.4 Chybové stavy

Sk nastala chyba pri spustení samotného serveru, je program ukončený s nenulovým chybovým návratovým kódom a chybovou hláškou vypísanou na štandardný chybový výstup. Návratové kódy sú:

- 1 - použitý neznámy argument pri spúšťaní programu
- 2 - chýbajúci povinný argument pri spúšťaní programu
- 3 - chyba pri otvorení súboru obsahujúceho zoznam blokovaných domén
- `errno` - chyba pri nejakej zo systémových sieťových funkcií

## 4 Preklad a spustenie

### 4.1 Preklad

Preklad programu prebieha pomocou príkazu `make`:

1. Pomocou príkazového riadku je nutné otvoriť priečinok, v ktorom je uložený zdrojový súbor `dns.cpp` a príslušný `Makefile`.
2. Spustenie príkazu `make` vytvorí spustiteľný súbor `dns`.

### 4.2 Spustenie

Spustenie programu vyzerá nasledovne:

```
./dns -s server [-p port] -f filter_file [-h]
```

kde:

- `-s server` je IP adresa alebo doménové meno DNS serveru (resolveru), kam sa má zaslať požiadavok.
- `-p port` je číslo portu, na ktorom bude program očakávať požiadavky (implicitne port 53).
- `-f filter_file` je meno súboru obsahujúceho zoznam blokovaných domén.
- `-h` pomocou prepínaču program iba vypíše stručné informácie s návodom, ako ho spustiť.

Prepínače `-s server` a `-f filter_file` sú povinné.

### 4.3 Príklad spustenia

V nasledujúcom príklade bude server spustený na porte 53, DNS požiadavok bude odoslaný na Google DNS server a nežiaduce domény budú čítané zo súboru `filter`.

```
./dns -s google.dns -f filter
```

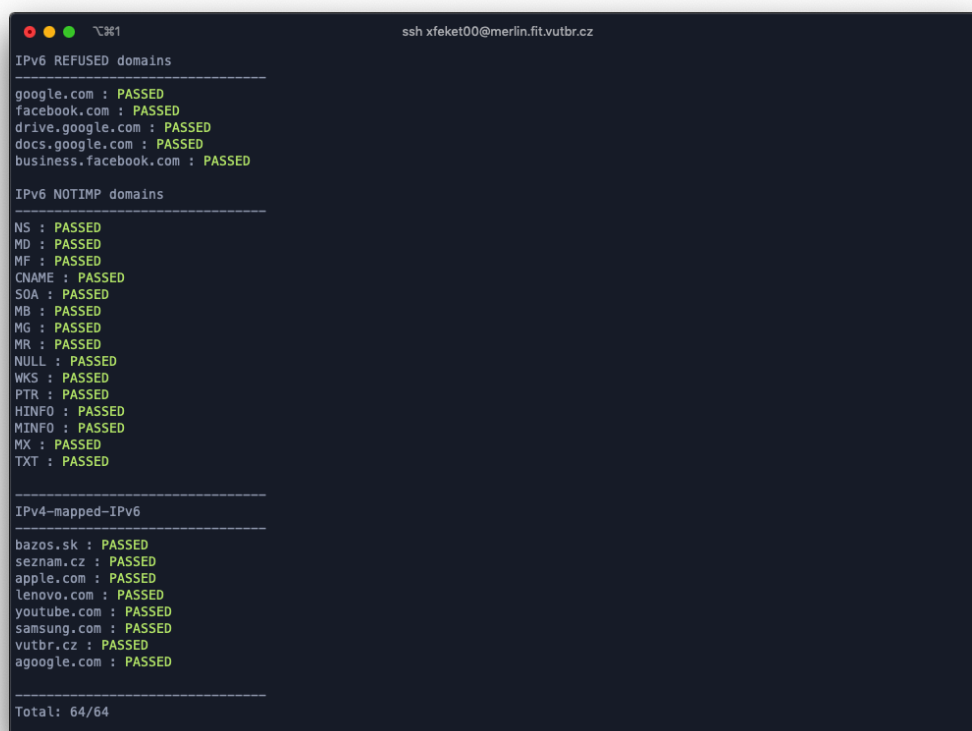
Server samotný nemá nijakú odozvu, je iba odoslaná odpoveď klientovi.

## 5 Testovanie

Funkčnosť programu bola testovaná na operačných systémoch MacOS a serveroch merlin.fit.vutbr.cz a eva.fit.vutbr.cz.

### 5.1 Automatické testovanie

Na účely automatického testovania bol implementovaný testovací skript v jazyku Python 3, ktorý je možné spustiť príkazom `make test`. Skript obsahuje 64 testov a odosiela požiadavky na IPv4 adresu serveru, IPv6 adresu serveru a takisto IPv4-mapovanú adresu serveru. Posielané požiadavky sú požiadavky na preklad validných (nefiltrovaných) adries, filtrovaných adries a takisto typu iného ako A. V prípade validných požiadavkov porovnáva testovací príznaky odpovede a IP adresu odpovede s výsledkom programu `dig`. V prípade nevalidných požiadavkov je pomocou programu `awk` a `grep` z odpovede serveru ponechaná iba hláška `status` a tá porovnávaná s obsahom súboru obsahujúceho očakávanú hlášku (teda buď `NOTIMP` alebo `REFUSED`).



```
ssh xfeket00@merlin.fit.vutbr.cz

IPv6 REFUSED domains
-----
google.com : PASSED
facebook.com : PASSED
drive.google.com : PASSED
docs.google.com : PASSED
business.facebook.com : PASSED

IPv6 NOTIMP domains
-----
NS : PASSED
MD : PASSED
MF : PASSED
CNAME : PASSED
SOA : PASSED
MB : PASSED
MG : PASSED
MR : PASSED
NULL : PASSED
WKS : PASSED
PTR : PASSED
HINFO : PASSED
MINFO : PASSED
MX : PASSED
TXT : PASSED

IPv4-mapped-IPv6
-----
bazos.sk : PASSED
seznam.cz : PASSED
apple.com : PASSED
lenovo.com : PASSED
youtube.com : PASSED
samsung.com : PASSED
vutbr.cz : PASSED
agoogle.com : PASSED

Total: 64/64
```

Obr. 1: Ukážka výstupu testovacieho skriptu

# Literatúra

- [1] *dig(1)* - *Linux man page*.
- [2] *inet6(4)* – *OpenBSD Kernel Interfaces Manual*.
- [3] R. Gilligan, Intrinsa Inc., S. Thomson, Cisco, J. Bound, J. McCann, Hewlett-Packard, and W. Stevens. *Basic Socket Interface Extensions for IPv6*. RFC 3493, IETF, Február 2003.
- [4] P. Matoušek. *Síťové aplikace a jejich architektura*. VUTIUM, 2014.
- [5] P. Mockapetris. *Domain Names - Implementation and Specification*. RFC 1035, IETF, November 1987.